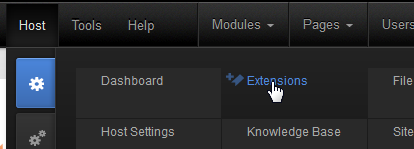
DNN Blog Module Manual

# Installation

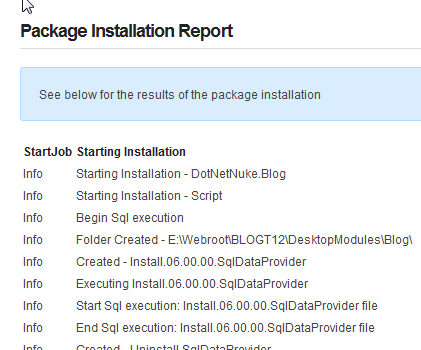
First, let’s identify what you’ve downloaded. Normally you’ll have downloaded a zip file with a name along these lines: Blog\_06.00.00\_Install.zip. The name explains what it is you have. “Blog” is the name of the module, “06.00.00” is the version and “Install” is the package type.

The module can be found in two “flavors”: Install and Source. These are two different zip files. The Install zip file (or “package”) includes just those files necessary to make the module work, while the Source package includes the source code files that will enable you to change the behavior of the module to your liking and recompile it. Normally you’d install the “install package”, i.e. in this case Blog\_06.00.00\_Install.zip.

This zip file needs to be uploaded to DNN. You do that by logging in as host user and going to Host > Extensions



Click “Install Extension Wizard” which will bring up a pop up screen that will start the installation process. Choose the downloaded zip file and click “Next”. Keep clicking “Next” to walk through each stage of the installation process until you see the final installation report.



If you see anything in red or bold on this report, or anything else suspicious: now’s the time to copy the contents of this report so it can be referred to at a later time to check if the installation was OK. Most importantly the report shows the result of the SQL scripts that were run and the writing of the files to your DNN installation. Installation errors commonly are a result of SQL scripts failing or files not being written to the server’s hard disk.

Click “Return” once you’re satisfied the installation was OK. You’ve now installed the module.

# Upgrades

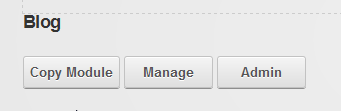
You can skip this part if this was the first time you’ve installed the module. If you’ve been using the DNN Blog module before, then you need to pay attention. DNN takes care of upgrading modules. There’s a built in mechanism to do this which is based on the version number of the module. The module makers make sure the package contains all necessary details to upgrade older versions of the same module. The procedure for upgrading is no different than for a regular first install of the module. I.e. you log in as host and upload the zip file as described above. However, I urge you to do one thing: backup your installation. Why? Is it because we’ve taken a cavalier approach to your upgrade? No. It’s just that DNN is unable to recover from a bad upgrade. And this inevitably means you’ll lose your data. So a prudent DNN administrator will back up his/her installation before upgrading a module.

Secondly: pay extra close attention to the installation report mentioned above. This is not trivial. But again: DNN can’t recover from an error and sometimes errors begin to appear later on. Make sure that (1) the SQL scripts that were run are all labeled with a higher version nr than the previous version you had installed, (2) that they didn’t produce any errors, (3) that dlls were actually written to the bin folder and (4) that other files were written to the module folder. Then do a “sanity check” on the module. Is it still working? Try various functions to be sure. Only then give the green light that all’s well.

## Pre version 6 to version 6 upgrades

For version 6 the module has been completely rewritten. And in doing so there have been a number of paradigm shifts in how the module works. The first thing you’ll notice is that the module looks very differently. What you need to realize at this point is that the module is no longer a group of modules with each a specific function (i.e. blog list, blog view, category list, etc). Rather it is one single module that is set to display a “template”. Templates are included for most of the previous submodules. I.e. the default template shows the blog posts, there’s a template for a category list, one for a calendar, etc.

A second major shift in paradigm is that the module manages its content per module, and no longer per portal. In the old days, if you'd add a second blog module to another page it’d display the same data. This is no longer the case. The module encases its own data. This can still be multiple blogs per module, but the data is locked to the module. So how can you make another module show data from the first module? That’s done in the module’s settings. Each module has the option to point to a data source Blog module anywhere on the site. The upgrade tries to do set this up as best as it can. But if you see Blog module parts that are empty on your migrated site, check out this setting. And find out the primary module where the data is now kept. Usually it should show up with the default template and the management buttons at the top.

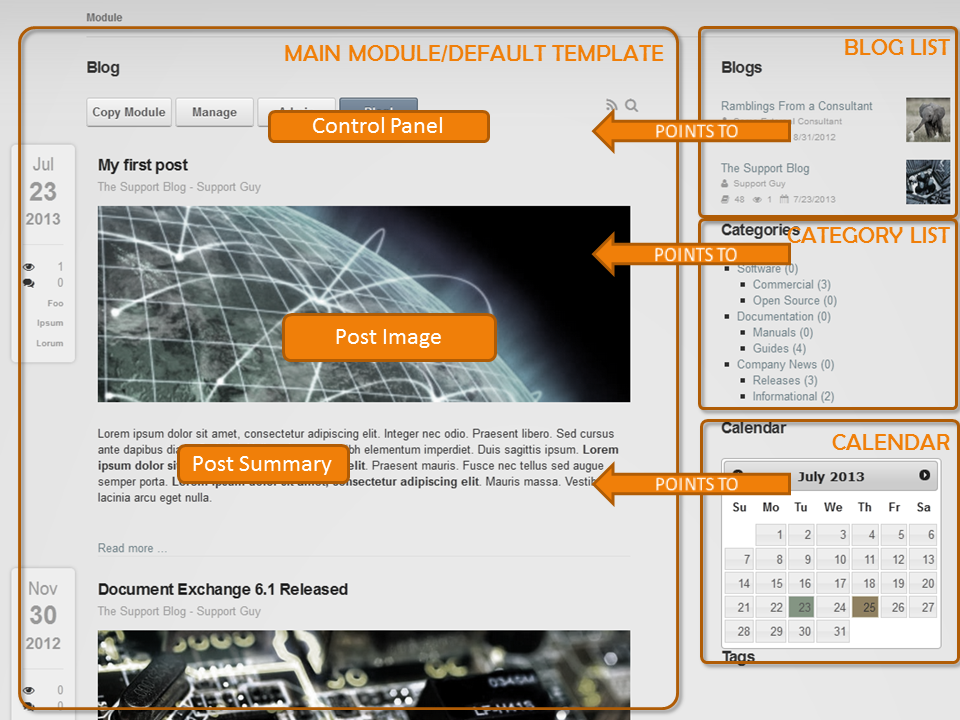


# Quick Start

If you’re itching to start blogging here’s the quick path to start:

1. Create or go to the page you want to use for your blog
2. Add the blog module to the page
3. Click the “Manage” button on the module that has now appeared
4. On the “Blogs” tab, click “Add Blog”
5. Add a title for your blog. Under “Permissions” select the checkbox “All Users/View Comments” and “All Users/Add Comment”. Click “Update”. Click “Return” to return to the main page.
6. You should now see a big button “Blog!”. Click that and start blogging.

# Overview



Version 5 and older of the DNN Blog module consisted of multiple so-called “module definitions”. This meant that when you put the module on a page, you didn’t see just one module appear. Instead you got a bunch of modules, each with a different role in the module. Although this mechanism has some merit, it was confusing at best. Plus: there is no good way in the DNN framework to manage the individual module “parts”. Finally the different module definitions were mostly different representations of the same data (i.e. a list of blogs, a list of categories, etc).

Instead, the module is now more straightforward in that it consists of just one module. This module can be set to render in any number of ways which will allow the amount of representations to keep growing. Out of the box the module comes with 11 templates that each show the module’s data in a different way. This includes the old ways data was shown, plus a number of new ways.

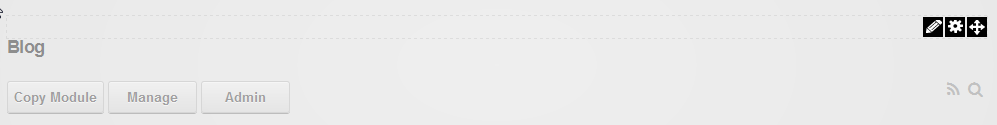
In the screenshot above you can see the main representation (i.e. the default template) shown on the left. On the right you’ll see several module instances showing other templates: a blog list, a category list and a calendar. The “secondary” modules don’t even need to be on the same page. You just need to “point” them to the main blog module so they know where the data comes from (you can have multiple blog modules on your site that each hold its own set of blogs). This “pointing” is done by specifying the parent blog module in the module settings of the secondary modules.

Another thing to note is that the main module shows the control panel. Whether it is shown is also set in the module settings. The idea is that you set it to show on your main blog module.

Finally you’ll see the default template includes a number of features in the way it renders a list of posts. Notably there is a small side panel with metadata (date of publication, nr of view/comments, tags and categories), a banner image (the image specified in the meta data of the post) and the summary of the post. If you find that something doesn’t render as above or as desired, note that most likely it is a result of the interplay of the module’s HTML and CSS with the skin that you are using. The templates make it as easy as possible for you to change these. The packaged templates are found under DesktopModules/Blog/Templates. You can copy and change templates under Portals/[id]/Blog/Templates. Both sets of templates will show up on the template selector.

# The Main Screen

By default, when you add the blog module to a page you’ll see a line of buttons at the top (this is in edit mode):



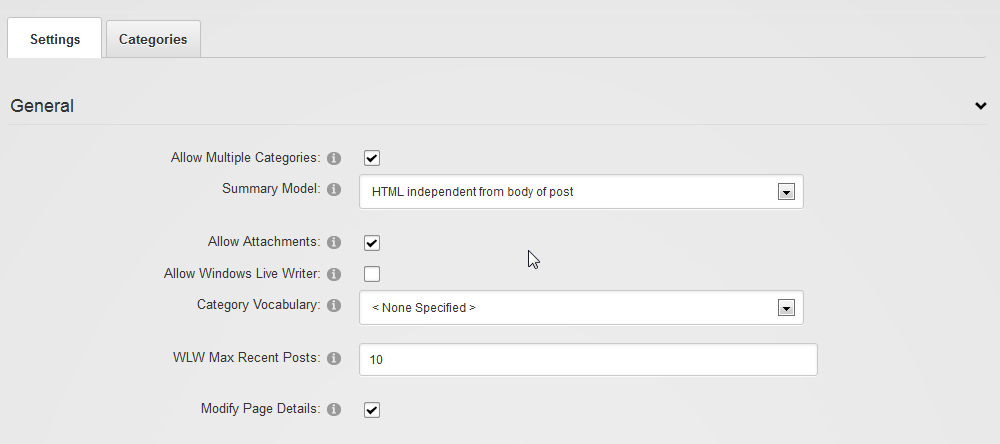
One of the goals of the rewrite of the module has been to offer more flexibility but yet to simplify operations. This is no easy task and I can't say for sure whether we’ve hit the mark. Time will tell. The buttons are there to provide entry points to the primary operations. Admins will see all buttons. Bloggers will see “Manage” and “Blog!” (once a blog has been created). Regular users will only see the grey icons to the right (RSS and Search).

Note that the module menu is still significant. So for some things you’ll need to switch to Edit mode.

|  |  |
| --- | --- |
| The gear icon will give you access to the “Module Settings” | The pencil icon leads to both the “Manage” screen (the same as through the “Manage” button on the main view) and “Template Settings”. More about templates later. |

# Admin Screen

The admin screen shows settings that apply to all managed content of this module. I.e. all blogs. This is why it’s only accessible to those with Edit permissions on the module. The Admin page is divided over two tabs: Settings and Categories. Because categories are also shared between all blogs, this is also managed on this page. Let’s examine the settings first.



### Allow Multiple Categories

If selected then bloggers can select more than one category for their posts.

### Summary Model

We’ve tried to make more explicit how summaries are being used in blogging. It turns out this varies across platforms and they differ significantly. The basic difference is between having it as an introduction to the main text (in Windows Live Writer this is achieved using a “Split Post”) or as a completely independent entity. In the former, the complete post is a concatenation of the summary text and the body text. In the latter the complete post is just the body text. In this case the summary is really a summary of the body text.

A second distinction is between plain text or HTML. In some applications it is useful to force users to provide a summary in plain text. I.e. without the possibility to add markup, images, etc. This is useful when you need total control over the presentation of the text on aggregated views. I.e. when you’re displaying a list of post summaries, you may not want your bloggers to be able to inject HTML which would potentially ruin the list’s appearance on the web page. Similarly, a scientific blog may require a more academic style no frills abstract as summary that can be emitted over RSS without the risk of upsetting presentation elsewhere due to faulty HTML.

For these reasons there are 3 models for the summary: a summary preceding the main post (by definition this would be HTML as the rest is HTML), an independent HTML summary and an independent plain text summary. The default is an independent HTML summary.

### Allow Attachments

Allowing attachments allows Windows Live Writer to add images and so forth to a blog post and upload them to the Blog module. If you disallow this no images can be embedded in a blog post sent using WLW.

### Allow Windows Live Writer

Allow bloggers to use Windows Live Writer (see separate chapter about this tool for more details).

### Category Vocabulary

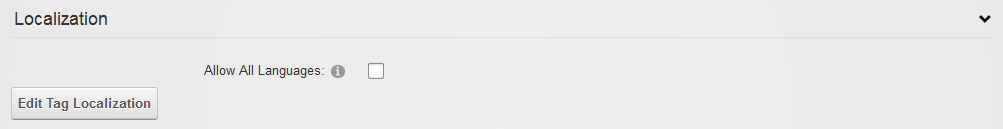
DNN organizes categories and tags in so-called vocabularies. These are part of what is called the Taxonomy feature of DNN. You can select a vocabulary to use from the dropdown or you can create a new vocabulary on the categories tab on this page.

### WLW Max Recent Posts

When bloggers connect using Windows Live Writer, the program retrieves a list of last posts which allow the blogger to edit those. Here you specify how many posts will be retrieved at most.

### Modify Page Details

When selected the module attempts to change the page title and inject the post title whenever it can. There is no guarantee as other pieces (the DNN framework or other modules) may attempt to do the same.

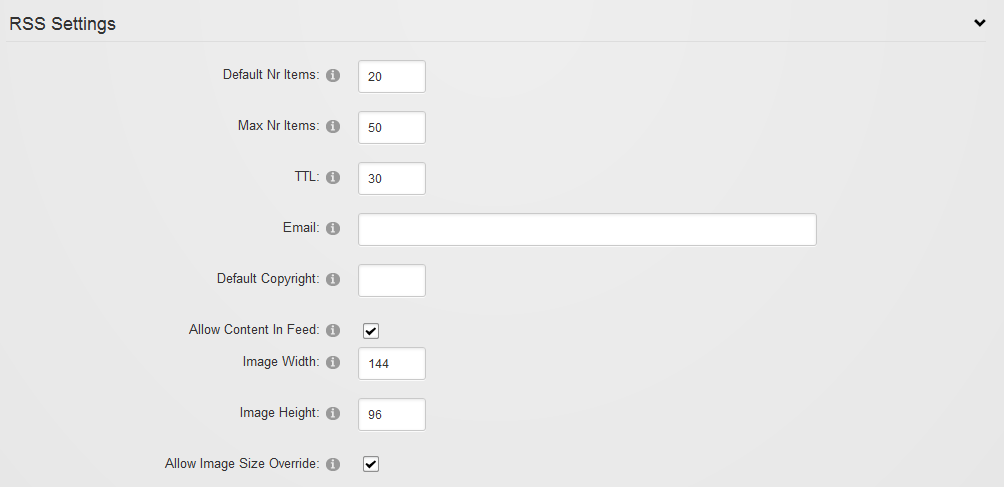


### Allow All Languages

Bloggers can now indicate for each post in their blog what language the post is in. This allows us to filter at a later stage if desired. This option allows users to indicate any language for their post. The default is not selected, which means that only one of the portal languages is allowed.

### Edit Tag Localization

This button takes you to an editor where you can specify translations for the tags used in various posts for this module.



The RSS engine for the module has been rewritten as well and now includes a number of new options. As before, the context of your view (i.e. if you’re viewing just a single blog or viewing an aggregated view of the blogs of the module) determines what the RSS feed will show. This is all done through querystring parameters. The querystring determines which posts will be selected for output and the format of the output.

### Default Nr Items

If nothing’s specified in the querystring, this is the number of posts that will be included in the feed. Consumers of the feed can specify the number of items using “recs” in the querystring. I.e. recs=5 will tell the blog module to only send 5 posts.

### Max Nr Items

The maximum amount of items to send. This is to protect against an overload if someone asks for a feed with recs=10000 for instance.

### TTL

The “Time To Live” is a value indicating how many minutes the feed will be cached before it is refreshed. Consumers are expected to cache for this length of time as well to avoid the feed being requested again and again.

### Email

This is the email address sent in the feed as managingEditor. Note this is overridden by a blog’s email address if a single blog is requested. This email address is only used in aggregated feeds. If left blank no managingEditor is included.

### Default Copyright

The text included as “copyright” in the feed. If left blank no copyright is included.

### Allow Content In Feed

If selected, consumers can ask for the complete content of the blog posts as well. Consumers will need to specify “body=true” in the querystring. Potentially it allows you to mirror the content of the blog somewhere else.

### Image Width/Height

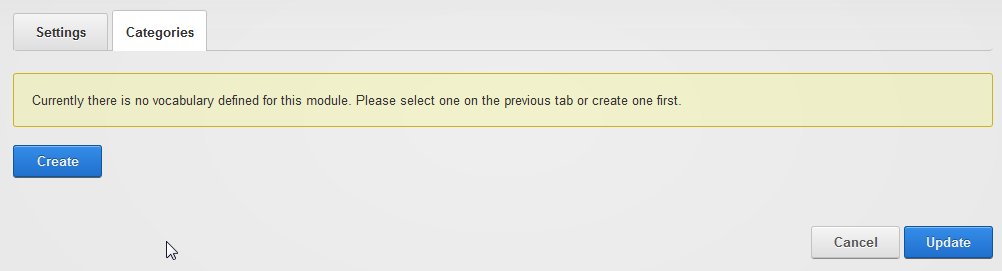
Since blog version 6, a post can have an associated image. You can specify what the image width and height will be for the RSS feed. Note the image will still be retrieved from your own blog module. The image itself is not included in the RSS feed as it is binary information.

### Allow Image Size Override

You can allow the consumer to override image width and height using “w=240” and “h=140” for instance in the querystring.

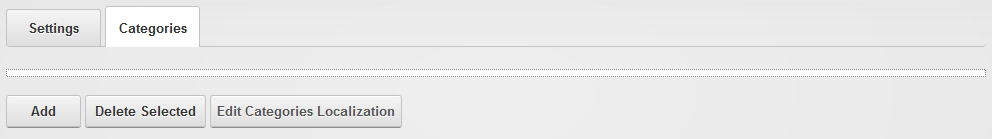
# Categories

As said before, categories are stored in DNN’s taxonomy system. This system allows you to create multiple “vocabularies” for various applications in your site. You can opt to select an existing vocabulary of your site, or you can create a new one for the categories of your Blog module:

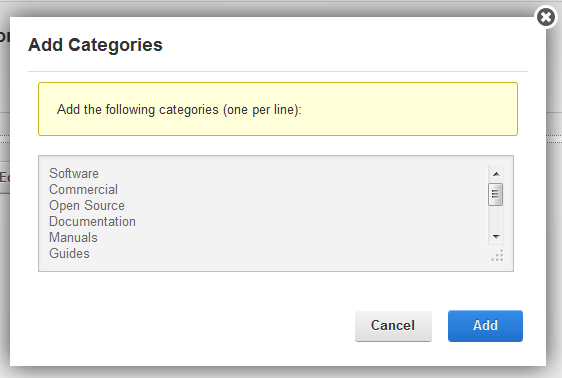


Categories are managed by those with edit rights to the module and the idea is that they serve as a “rigid” structuring mechanism for the content of the module. This in contrast with tags that can be created by bloggers and provide a more fluid structuring mechanism.

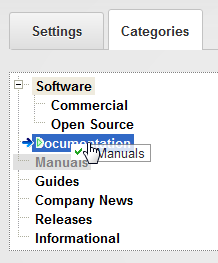
After you click “Create” you’ll see this:



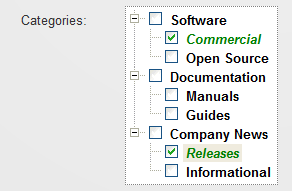
Adding categories is done in a pop-up screen that allows you to add categories in bulk.



Once added, you can click and drag items around to rearrange them in a hierarchy.

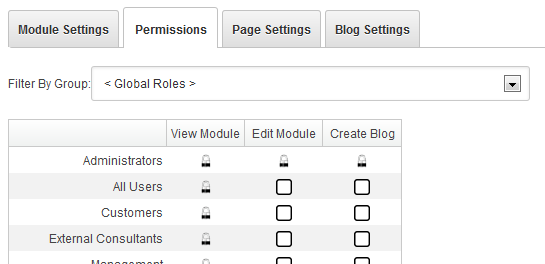


The resulting tree will show up in the post edit screen under metadata

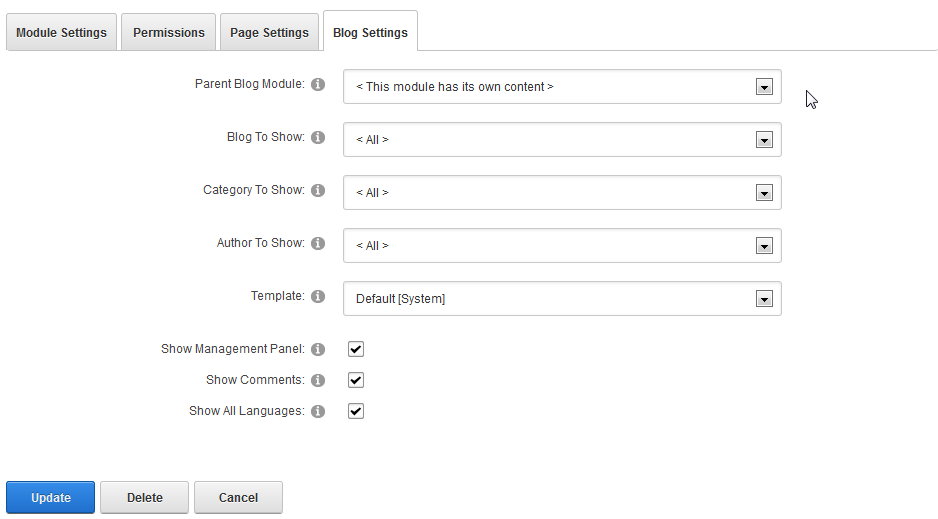


# Module Settings

You access the module settings through the module menu when you’ve switched to Edit mode on the page (described earlier). You’ll notice a new permission type under the Permissions tab:



Module editors can edit settings of the module. Those with “Create Blog” permission can create blogs in the module. The latter will see the “Manage” button on their screen when they log in.



### Parent Blog Module

Given that this is the first module you’ve stuck on the page, you’ll leave this as it is. But once you add another instance of the Blog module to the page (or another page), you can point that instance to this module by selecting it from the dropdown. Then, that “parent” blog module will be used to draw the data from. Typically you use this to show the data in another fashion somewhere, like an author list, a blog list, or a tag cloud for instance.

### Blog/Category/Author To Show

You can opt to have the module show just a single blog, category or author. Select that here. This way you could have a single themed page somewhere on your site which just shows posts relevant to that page.

### Template

Selection of the template to use. The blog module comes with a number of templates “out of the box” that you can use. You can have your own templates stored in the portal home directory as well. More about templates later on.

### Show Management Panel

If selected this shows the buttons at the top of the module and provides an entry point for bloggers. The suggestion is to only use this for the parent blog module which manages its own content.

### Show Comments

Comments are shown based on a set of criteria. On some blog modules (e.g. a tag cloud or an author list) comments would be out of place, so you’d deselect this option.

### Show All Languages

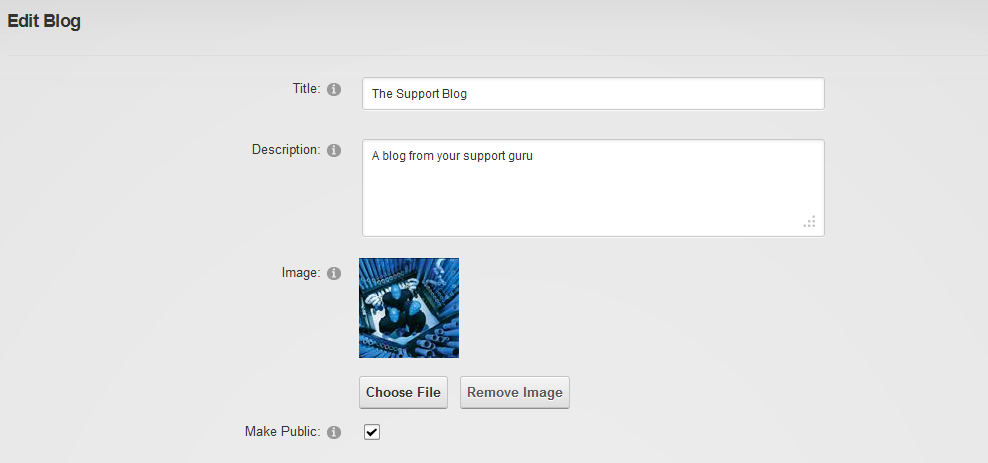
Posts are labeled with the language they’re written in. When showing a list of posts, you can opt to either show all posts or only those posts labeled with the language currently selected in DNN.

# Manage Screen

The manage screen is the entry point to managing content of the module you have access to. You’ll find a list of your blogs and posts on this screen.

## Adding Your First Blog

Click on Add Blog on the management screen and you’ll be taken to the Blog edit screen:



### Title/Description

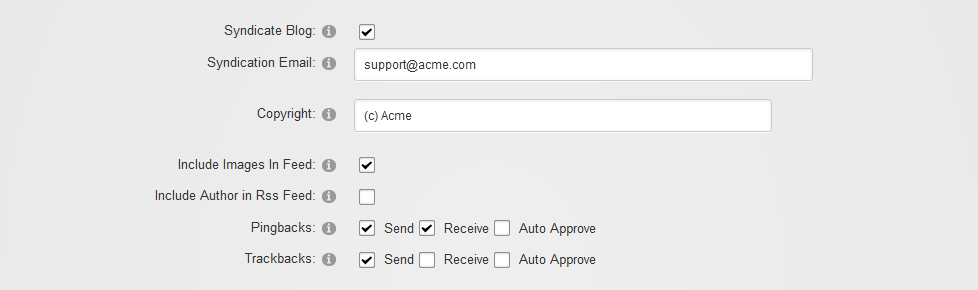
The title is compulsory. The description is not. The latter

### Image

A blog can have an associated image, like a logo, which can be used in templates. The blog list template, for instance, will show this image.

### Make Public

If selected then this blog is for public viewing. That is: for all those with VIEW permissions to the module. Otherwise it is only visible to the owner of the blog.



### Syndicate Blog

This option determines whether or not to include this blog in aggregated feeds and whether to allow users to draw an rss feed from this blog at all.

### Syndication Email

The syndication email is the email address used for managingEditor in the RSS feed if specified.

### Copyright

The copyright, if specified, will be included in the feed. It can also be used in templates.

### Include Images In Feed

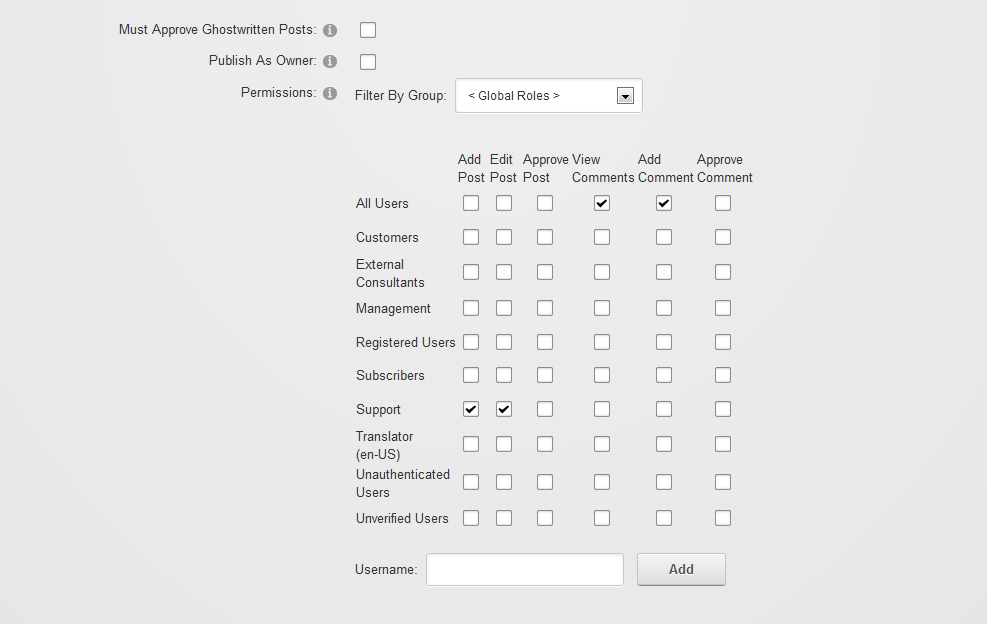
If selected, the images associated with posts will be included in the RSS feed.

### Include Author in Rss Feed

If selected the actual author of the post (not necessarily the owner of the blog) will be looked up in DNN and his/her email will be included in the feed as author of the post.

### Pingbacks/Trackbacks

Specify the behavior of the module for this blog here with regards to ping- and trackbacks. Ping- and trackbacks are mechanisms whereby, during the publishing of a new blog post, the blogging software scans the post for links to other blogs. Those links are then tested to see if the other blogs support the ping/trackbacks. If they do, a ping/trackback is sent and this will result in a comment being placed underneath that blog with a link to your new post. For some bloggers this is immensely important as it raises your Google ranking (and ranking within your blogging community), plus it is can be very useful. The downside is that these mechanisms are prone to spam. Pingbacks especially are prone to spam. Trackbacks include an extra check to see if the referring post really exists and includes the link. For more details I encourage you to Google on pingbacks and trackbacks.



The blog module includes powerful features to allow for shared authoring of blogs. Not only does the module support the concept of multiple blogs that are presented in aggregated form. Every blog can be opened up to be shared by others to write for.

### Must Approve Ghostwritten Posts

If selected, the owner must approve every post that is made to the blog. Approving can be delegated to others in the permissions.

### Publish As Owner

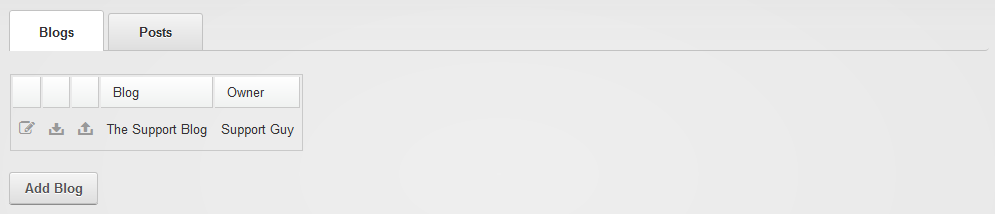
If selected then every post made to the blog appears to be written by the owner of the blog (i.e. the user that created the blog). If not selected, then the actual user that submitted the post is recorded as the author.

### Permissions

The permission grid set permissions of various aspects of the blog. Currently we have 6 permissions:

|  |  |
| --- | --- |
| Add Post | The ability to author a post on this blog. If none are selected here, then only the owner can write to this blog. |
| Edit Post | The ability to edit (any) post on the blog. |
| Approve Post | The ability to approve ghostwritten (i.e. not by the owner) posts. |
| View Comments | Can view (all) comments. Note that you’ll want to grant this permission to all that can add comments at the very least. |
| Add Comment | Ability to comment on posts. Note that each post still has a setting to shut down comments which overrides this permission setting. |
| Approve Comment | The ability to approve or delete comments. |

## Posts List



Having created your blog you’ll see it appear on your management list. With each blog you have access to the following functions: Edit (on the edit screen you can find a delete button if you need that), Import and Export.

### BlogML Import/Export

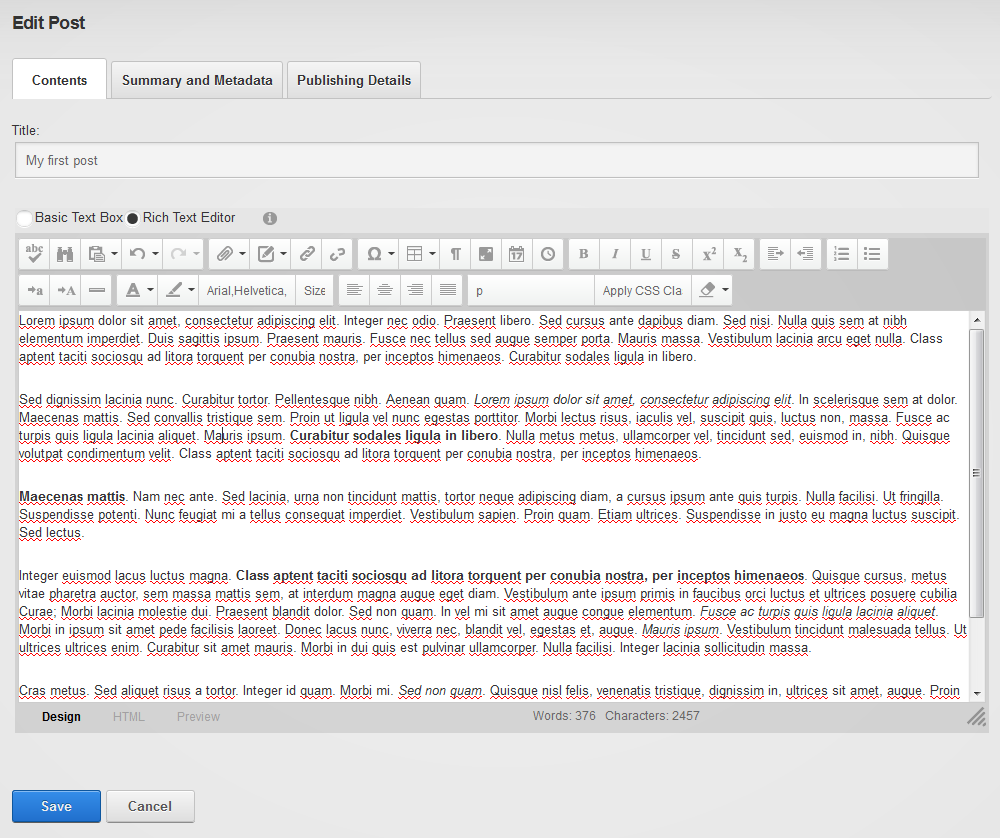
The blog module includes the ability to import and export a blog in the standard BlogML format. This is an XML standard designed for blogs. Note this includes embedded images, so expect these files to grown in size if you’ve used images extensively.

Both the import and the export buttons next to the blog will take you to a popup wizard that will perform the actions.

# Post Edit Screen

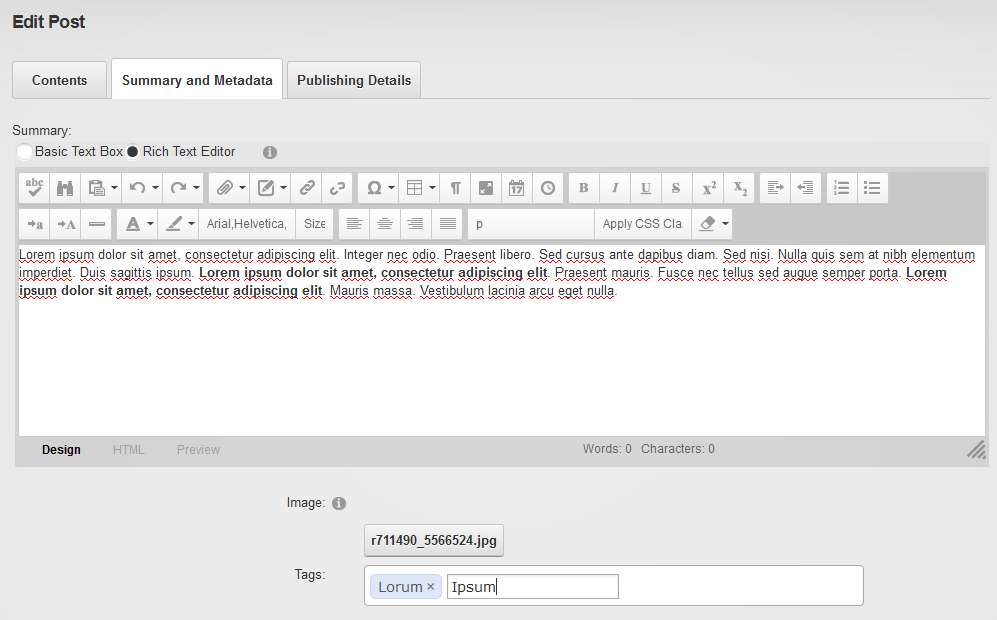
The post edit screen (you get there by clicking either “Blog!” or “Edit Post”) is divided in three parts: contents, summary and metadata and publishing details.

## Contents



The contents tab holds the most important part of your post: the title and the body text of the post. DNN’s default text editor is used, so for any questions about features I’ll refer you to DNN documentation about that. Note that if you’d like to have images embedded in the post, you’ll need to use the appropriate button from the text editor toolbar where you can upload/select images.

## Summary and Metadata

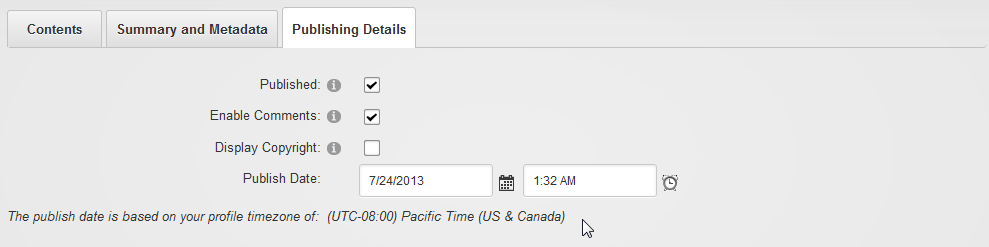


A **summary** serves as en enticer to your post. Adding one makes sense. It is also the text fragment that is sent when some other site requests the blog contents through an RSS feed.

The **image** serves as an identifying image that can be used in templates to make the blog look more professional. For example, news sites always include an image with each article. This allows them to show the image as an attention grabber in various places on their site.

**Tags** and **Categories** serve to reorganize the contents of your blog module. That is: you can select posts based on a single tag or category. The difference between tags and categories is that tags are “free to enter” words by the blogger, whereas categories are managed by the module administrator and serve to “structure” the content of the module. But it is ultimately up to you how you wish to use them. Just be aware that bloggers can add tags but not categories.

## Publishing Details



This is where you specify if and when the post will be published. Note the message about the time zone. Every user in DNN can specify his/her timezone. This is used in the blog module to determine what time it is where you are. By default it will show the value for when you clicked on “Blog!” for the Publish Date. I.e. it will be published immediately. If you don’t see the post appear immediately in the feed, then check your timezone settings. Is that correct? Or did you publish to the future?

### Published

If not selected the post will remain invisible to others. Otherwise the post will be published and added to your Journal in DNN.

### Enable Comments

If unchecked, no comments will be allowed for this post. This overrides the settings for the blog. This can be useful if you feel that a particular post may attract too much trolling and you wish to shut down the comments for that particular post.

### Display Copyright

This switch can be used in the template to optionally show the copyright text set in the blog settings.

# Windows Live Writer

If you’re not already familiar with this tool or you haven’t heard of it, chances are you’re new to blogging. Windows Live Writer (or simply WLW) is a Windows desktop program (it’s part of a suite of free software tools made by Microsoft called “Live Essentials”) that resembles Word, which allows you to upload a written document to your blog at the click of a button. This not only includes text styles (headings, bold, italic, subscript, etc) but also tables and images. It is incredibly easy to write a blog post using this tool as you don’t need to upload images you wish to use individually to the web server. Instead, WLW takes care of uploading these embedded images and sends them to the server for you. With this program, you can blog to various platforms including Blogger and other non-Microsoft platforms. We’ve implemented a protocol known as Metaweblog API to allow WLW to post to the module. If you’re able to blog to the module, you’ll see an icon on screen that will bring up a popup with the Metaweblog API link to the module.