

# C# Blog Templates BEST PRACTICES

## Introduction

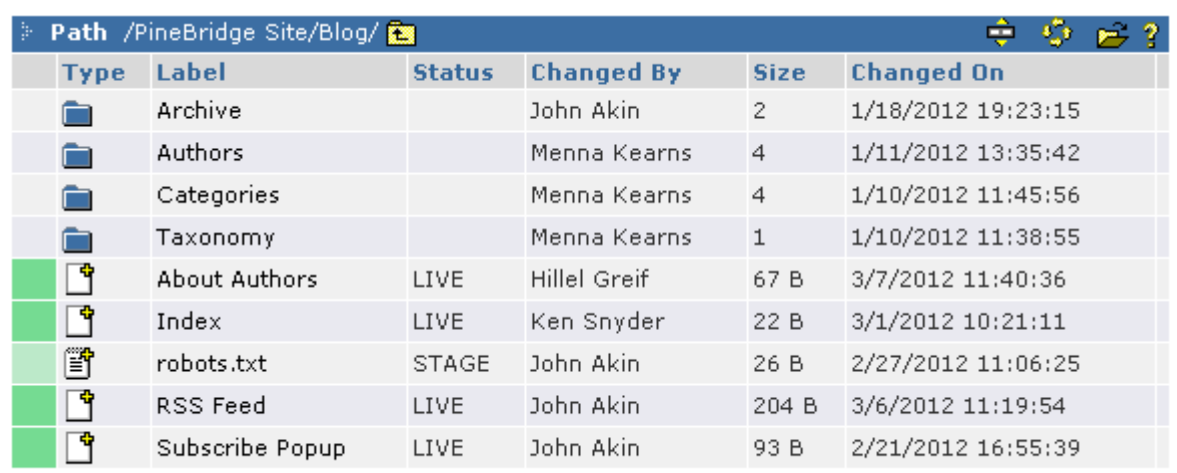
These templates support CrownPeak's blogging platform - which can be configured using C# API.

Standard blog implementations include a separate workflow, so content can be approved to the site in a less restrictive manner.

## Blog Features

- Feeds - atom and xml
- Archive - organized by month
- Blog Index - List of recent post for each author
- Posts - each post will include a date and time stamp, will be organized
- Manage categories - Each post can be linked to categories
- Manage tags - Each post can be linked to tags
- Author Configuration - Manage author information such as bio, titles, and links to all authors posts. Authors create their posts in a folder dedicated to their content.

## Blog Structure








Type	Label	Status	Changed By	Size	Changed On
Folder	Archive		John Akin	2	1/18/2012 19:23:15
Folder	Authors		Menna Kearns	4	1/11/2012 13:35:42
Folder	Categories		Menna Kearns	4	1/10/2012 11:45:56
Folder	Taxonomy		Menna Kearns	1	1/10/2012 11:38:55
File	About Authors	LIVE	Hillel Greif	67 B	3/7/2012 11:40:36
File	Index	LIVE	Ken Snyder	22 B	3/1/2012 10:21:11
File	robots.txt	STAGE	John Akin	26 B	2/27/2012 11:06:25
File	RSS Feed	LIVE	John Akin	204 B	3/6/2012 11:19:54
File	Subscribe Popup	LIVE	John Akin	93 B	2/21/2012 16:55:39

- Authors
  - Author Model which includes Index and Author Config files. Index will list all the author's post. The config allows author and admin to add bio and title information.
  - Authors have the ability to create year sub-folders in their author folder in order to organize content by year. This allows CMS to organize and archive content.
  - Within each year folder, author is able to create new blog post.
- Tags
  - New tags can be created in the Taxonomy folder. Tags are available as checkbox items in the blog post input, under the Taxonomy tab
- Categories
  - New tags can be created in the Categories folder. Categories are available as a dropdowns in the blog post input, under the Taxonomy tab
  - Categories are pre-pended to the title of each post when listed. This is configuration and can be disabled per post.
- Archive

- User is able to create an archive folder for each year and then a archive page for each month. Based on the date of each folder, blog post will be included in the pages.
- About Authors
  - List author name, title, bio
- RSS Feed
  - RSS feed, RSS format (possible atom compatibility)
- Index
  - LAnding page

## Steps to configure

- Templates
  - Author Config - pulled automatically into About Authors page unless selected not to

Type	Label	Status	Changed By	Size	Changed On
	assetfilename.aspx		Menna Kearns	640 B	1/31/2012 10:39:16
	asseturl.aspx		Menna Kearns	604 B	1/31/2012 10:39:44
	filename.aspx		John Akin	625 B	1/30/2012 16:58:15
	input.aspx		Menna Kearns	1 KB	1/10/2012 12:36:57
	output.aspx		Menna Kearns	1 KB	2/7/2012 14:26:29

- Fields: Author Name, Show/Hide from About Authors Page, Author Title, Author Short Title, Author Company, Author image thumbnail, Author Bio (WYSIWYG)

```

<%
// input.aspx: template file to specify content entry input form
context.ShowSaveEditButton = true;
Input.ShowHeader("Author Configuration");
Input.ShowMessage("Enter Author information and click Save");
Input.ShowTextBox("Author Name", "author_name", width:30);
Input.ShowCheckBox("", "author_index_hide", "true", "Check to hide in About Au
Input.ShowTextBox("Author Title", "author_title", width: 70, height:3);
Input.ShowTextBox("Author Short Title", "author_short_title", width: 50);
Input.ShowTextBox("Author Company", "author_company", width:50);

ShowAcquireParams iParams = new ShowAcquireParams();
iParams.AddAdditionalImage("author_thumbnail", 75, 97);
iParams.ShowBrowse = false;
Input.ShowAcquireImage("Headshot", "author_headshot", iParams);

WysiwygParams wParams = WYSIWYGConfig.MediumWYSIWYG();
//Insert WYSIWYG CSS here
Input.ShowWysiwyg("Bio", "author_bio", wParams);
%>

```

- About Authors- pulls Author configs

Type	Label	Status	Changed By	Size	Changed On
	filename.aspx		Menna Kearns	664 B	1/9/2012 14:40:25
	input.aspx		Edward Chan	2 KB	7/6/2012 17:44:55
	output.aspx		Menna Kearns	3 KB	3/6/2012 09:23:04
	post_input.aspx		Menna Kearns	872 B	1/9/2012 13:48:13
	url.aspx		John Akin	537 B	1/30/2012 16:59:17

```

<% String authorsFolderPath = "/" + asset.AssetPath[0] + "/Blog/Authors/";
Asset authorsFolder = Asset.Load(authorsFolderPath);
StringBuilder authorOutput = new StringBuilder("");
if (authorsFolder.IsLoaded) {
    // loop through and output authors
    List<Asset> authorFolders = authorsFolder.GetFolderList();
    int authorCount = authorFolders.Count;
    int counter = 0;
    foreach(Asset authorFolder in authorFolders){
        Asset authorData = Asset.Load(authorFolder.AssetPath + "/Author Config");
        counter++;
        String lastAuthor = "";
        if (authorCount == counter)
            lastAuthor = " last";
        if (authorData.IsLoaded && !String.Equals(authorData["author_index_hide"], "true"))
        {
            String authorTitleCompany = (String.IsNullOrEmpty(authorData["author_company"]) ? authorData
["author_title"] : authorData["author_title"] + ", " + authorData["author_company"]);
            authorOutput.AppendFormat(@"<div class=""authorItem{4}"">
                <div class=""authorImg"">
                    <img src=""{0}"" />
                </div>
                <div class=""authorDesc"">
                    <h3 class=""authorName"">{1}</h3>
                    <span class=""authorTitle"">{2}</span>
                    {3}
                </div>
                <br class=""clearfloat"" />
            </div>", authorData["author_thumbnail"], authorData["author_name"], authorTitleCompany, authorData["author_bio"], lastAuthor);
        }
    }
    Out.Write(authorOutput.ToString());
}
%>
</div>

```

- Taxonomy - create tag and category assets that can be reused on blog posts

Type	Label	Status	Changed By	Size	Changed On
	filename.aspx		John Akin	662 B	1/18/2012 01:29:38
	input.aspx		Edward Chan	2 KB	7/6/2012 17:51:47
	output.aspx		Menna Kearns	5 KB	3/6/2012 09:27:29
	post_input.aspx		Menna Kearns	834 B	1/10/2012 11:08:34
	url.aspx		John Akin	535 B	1/30/2012 16:54:27

- Fields: Title, Name

```

<%
// input.aspx: template file to specify content entry input form
context.ShowSaveEditButton = true;
context.ShowSaveNewButton = true;

Input.StartTabbedPanel(Util.MakeList("Page Content", "Configuration"));
String folderName = asset.Parent.Label;
switch (folderName)
{
    case "Tags":
        folderName = "Tag";
        break;
    case "Categories":
        folderName = "Category";
        break;
}
Input.ShowHeader(folderName + " Entry");
Input.ShowMessage("Enter configuration and click Save. This page both manages tax

Input.ShowTextBox("Browser Title", "title_tag", width:50);
Input.ShowTextBox(folderName + " Name", "taxonomy_name", width: 50);
Input.ShowTextBox("Header", "taxonomy_hdr", width: 50);
WysiwygParams wParams = WysiwygConfig.medWysiwyg_v2();
// Add WYSIWYG CSS here

```

- Output: stores type and name










```

<div id="mainContentBlog" class="blogMain">
    <%
        String indexType = "";
        String indexName = "";
        Boolean taxonomy = false;
        Boolean category = false;
        Asset topFolder = asset.Parent.Parent.Parent;
        switch (asset.Parent.Label) {
            case "Tags":
                indexType = "with tag";
                indexName = asset["taxonomy_name"];
                taxonomy = true;
                break;
            case "Categories":
                indexType = "in category";
                indexName = asset["taxonomy_name"];
                category = true;
                break;
        }

        if (gConfig.IsLoaded)
            Out.Write(OutputHelper.IndexBuilder(asset, topFolder, gConfig, context, indexType, indexName, taxonomy, category));
        else
            Out.WriteLine("<!--gConfig not loaded-->");
    %>
</div>

```

- Blog Post - create blog post, add tags and categories as desired.

Type	Label	Status	Changed By	Size	Changed On
	assetfilename.aspx		John Akin	640 B	1/30/2012 16:12:22
	asseturl.aspx		CrownPeak Admin	600 B	4/11/2012 09:59:19
	filename.aspx		Menna Kearns	664 B	1/9/2012 15:28:57
	input.aspx		Edward Chan	7 KB	7/6/2012 17:49:56
	output.aspx		Menna Kearns	11 KB	6/13/2012 13:39:40
	post_input.aspx		John Akin	6 KB	3/6/2012 15:01:13
	post_publish.aspx		John Akin	2 KB	3/6/2012 15:01:39
	post_save.aspx		John Akin	873 B	1/24/2012 17:14:19
	url.aspx		John Akin	537 B	1/30/2012 16:13:17

- Fields: Page Label, Title, Sub-Title, Summary, Publish Live Date, Content Option (WYSIWYG with optional Image, Link List, Text List)

Enter content below and click "Save".

Page Label  Leave blank to autogenerate from title and post date.

Title  REQUIRED

Sub Title

Summary Used in RSS feeds.

---

**Posting Dates/Times**

**Current Status: Live**

Publish Live Date  Read Only after a post is "LIVE"

---

**Main Content**

Select a post type and complete post entry below.

Post Type

**Free-form WYSIWYG**

Top Image   Select optional top image.

Image

Image AltText

Image Alignment

Enter free-form blog post.

• I am out traveling this week in balmy Southern California (and later-on chilly

Prepends the Category chosen on the blog post to the title of the blog

```
String postCategory = "";

if (!String.IsNullOrEmpty(asset["post_category"]) && (String.Equals(asset["show_category"], "true")))
{
    Asset category = Asset.Load(asset["post_category"]);
    if (category.IsLoaded)
        postCategory = category["taxonomy_name"];
}

if (!String.IsNullOrEmpty(postCategory))
    postTitle.AppendFormat("{0}: ", postCategory.ToUpper());
```

## Main output

```
switch (asset["post_type"])
{
    case "wysiwyg":
        postOutput.Append(asset["wysiwyg_content"]);
        postOutput.AppendLine("");
        break;
    case "text_list":
        List<PanelEntry> listItems = asset.GetPanels("list_item");
        postOutput.AppendLine("<ul>");
        foreach (PanelEntry listItem in listItems)
        {
            postOutput.AppendLine(String.Format("<li>{0}</li>", listItem["text_item"]));
        }
        postOutput.AppendLine("</ul>");
        break;
    case "link_list":
        List<PanelEntry> referenceItems = asset.GetPanels("reference_item");
        postOutput.AppendLine("<ul>");
        foreach (PanelEntry referenceItem in referenceItems)
        {
            postOutput.AppendLine(String.Format("<li><h4>{0} <a href=\"{1}\">{2}</a></h4>{3}</li>", referenceItem
:e_title", referenceItem["reference_item_link_link"], referenceItem["reference_item_link_title"], referenceItem
:e_description"]));
        }
        postOutput.AppendLine("</ul>");
        break;
}
postOutput.AppendLine("</div>");
```

## Tags added to post

```
String tagOut = ShowTagLinks_v2(asset);

if (!String.IsNullOrEmpty(tagOut))
{
    postOutput.AppendLine("<p class=\"tags\">");

    postOutput.Append(tagOut);
    postOutput.AppendLine("</p>");
}
}
```

- Blog Index- aggregates blog posts, built by IndexBuilder function

Type	Label	Status	Changed By	Size	Changed On
	filename.aspx		John Akin	726 B	1/18/2012 18:38:23
	input.aspx		Edward Chan	3 KB	7/6/2012 17:49:00
	output.aspx		Menna Kearns	5 KB	3/6/2012 09:26:25
	post_input.aspx		John Akin	810 B	1/12/2012 16:50:15
	url.aspx		John Akin	537 B	1/30/2012 16:54:33

## ■ Fields: Posts, Post Limit

Edit - /PineBridge Site/Blog/Index

Page Content Configuration

**Blog Index Page**

Set index options below and click Save.

Posts  Select the number of posts to display per page.

Post Limit  Limits the number of posts that will be returned on main blog index.

## Filter

```
FilterParams fParams = new FilterParams();
fParams.Add(AssetPropertyNames.TemplateLabel, Comparison.Equals, "Blog-PostCS");

fParams.FieldNames = Util.MakeList("post_date", "blog_desc", "post_type", "post_date_month", "post_date_year", "title", "tags",
if (taxonomy)
{
    String tagField = "tags_value_" + asset.Id.ToString();
    fParams.Add(tagField, Comparison.Equals, "true");
}

if (category)
{
    String catField = "post_category";
    fParams.Add(catField, Comparison.Equals, asset.Id.ToString());
}

if (archive)
{
    //calculate a date range from the month asset["select_month"] and year folder (asset.Parent.Label)
    //int dayMax = 31;

    //int dayMin = 1;
    int month = Convert.ToInt32(asset["select_month"]);
    int year = Convert.ToInt32(asset.Parent.Label);
    fParams.Add("post_date_month", Comparison.Equals, asset["select_month"]);
    fParams.Add("post_date_year", Comparison.Equals, asset.Parent.Label);

    //String catField = "post_category";
    //fParams.Add(catField, Comparison.Equals, asset.Id.ToString());
}

if (filterLimit > 0)
{
    //index header
    if (!String.IsNullOrEmpty(indexType))
        postOutput.AppendFormat(@"<div class=""cat-box"">
            <p>Showing posts {0} <strong>{1}</strong> &nbsp;<a href=""{2}"">Show all posts
            </div>", indexType, indexName, gConfig["blog_home_link"]);
}
```






Where indexType and indexName are the assigned tag, category, date, or author name, each generating their own filtered Index page

## ■ Output

```
<div id="mainContentBlog" class="blogMain">
    %> String indexType = "";
    String indexName = "";
    if (!String.Equals(asset.Parent.Label, "Blog", StringComparison.OrdinalIgnoreCase))
    {
        indexType = "by author";
        Asset authorConfig = Asset.Load(asset.Parent.AssetPath+"/Author Config");
        if (authorConfig.IsLoaded)
            if (!String.IsNullOrEmpty(authorConfig["author_name"]))
                indexName = authorConfig["author_name"];
            else
                indexName = "<!-- no author config loaded-->";
    }
    if (gConfig.IsLoaded)
    {
        int filterLimit = 0;
        if (!String.IsNullOrEmpty(asset["post_limit"]))
            filterLimit = Convert.ToInt32(asset["post_limit"]);
        Out.WriteLine(AssetHelper.IndexBuilder(asset, asset.Parent, gConfig, context, indexType, indexName, filterLimit:filterLimit).ToString());
    }
    else
        Out.WriteLine("<!--gConfig not loaded-->");

    // Out.WriteLine(context.PageNumber.ToString());
    %>
</div>
```

- Blog Archive - based on user-created archive folders

Type	Label	Status	Changed By	Size	Changed On
	filename.aspx		John Akin	664 B	1/18/2012 20:27:58
	input.aspx		Edward Chan	4 KB	7/6/2012 17:46:56
	output.aspx		Menna Kearns	2 KB	3/6/2012 09:26:44
	post_input.aspx		John Akin	2 KB	2/13/2012 18:27:02
	url.aspx		John Akin	537 B	1/30/2012 16:58:49

**Page Content**
Configuration

**Archive Index Page**

Set index options below and click Save.

Page Label   
*Set automatically on Save*

Select Month  ▼

Posts  ▼ *Select the number of posts to display per page.*

Fields: Month, Post Limit



Filtered by IndexBuilder function

### ■ Output

```
<div id="mainContentBlog" class="blogMain">
  <%
    String indexType = "";
    String indexName = "";
    Boolean taxonomy = false;
    Boolean category = false;
    Boolean archive = false;
    Asset topFolder = asset.Parent.Parent.Parent;

    indexType = "from";
    // indexName = asset["select_month"];

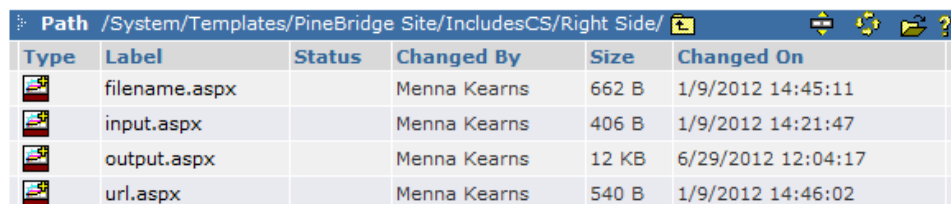
    Dictionary<string, string> dMonths = new Dictionary<string, string>();
    dMonths.Add("01", "January");
    dMonths.Add("02", "February");
    dMonths.Add("03", "March");
    dMonths.Add("04", "April");
    dMonths.Add("05", "May");
    dMonths.Add("06", "June");
    dMonths.Add("07", "July");
    dMonths.Add("08", "August");
    dMonths.Add("09", "September");
    dMonths.Add("10", "October");
    dMonths.Add("11", "November");
    dMonths.Add("12", "December");





    indexName = dMonths[asset["select_month"]] + " " + asset.Parent.Label;

    archive = true;

    if (gConfig.IsLoaded)
      Out.Write(OutputHelper.IndexBuilder(asset, topFolder, gConfig, context, indexType, indexName, archive:archive).ToString());
    else
      Out.WriteLine("<!--gConfig not loaded-->");
  >%
</div>
```

- Right widget - category, archive, and most recent tabs



Type	Label	Status	Changed By	Size	Changed On
	filename.aspx		Menna Kearns	662 B	1/9/2012 14:45:11
	input.aspx		Menna Kearns	406 B	1/9/2012 14:21:47
	output.aspx		Menna Kearns	12 KB	6/29/2012 12:04:17
	url.aspx		Menna Kearns	540 B	1/9/2012 14:46:02

### ■ Sort by most recent

```
<div id="recent-tabbed" class="widget">
  <%
    StringBuilder recentOutput = new StringBuilder("");

    FilterParams fParams = new FilterParams();
    fParams.Add(AssetPropertyNames.TemplateLabel, Comparison.Equals, "Blog-PostCS");

    fParams.FieldNames = Util.MakeList("stored_publish_date", "blog_desc", "title", "sub_title", "post_category", "show_category");
    fParams.SortOrder = SortOrder.OrderByDescending("stored_publish_date");
    fParams.Limit = 5;

    List<Asset> recentPosts = blogHome.Parent.GetFilterList(fParams);
    int postsCount = recentPosts.Count();
  >%
</div>
```

## ■ Sort by category

```
<div id="cat-tabbed" class="widget">
  <%
    StringBuilder catOutput = new StringBuilder("");

    Asset catFolder = Asset.Load("/") + asset.AssetPath[0] + "/Blog/Categories/";

    AssetParams aParams = new AssetParams();
    aParams.SortOrder = SortOrder.OrderBy("taxonomy_name");

    if (catFolder.IsLoaded)
    {
      List<Asset> categories = catFolder.GetFileList();
      int catCount = categories.Count();
      counter = 0;
      last = false;
      foreach (Asset category in categories)
      {
        counter++;
        if (counter == catCount)
        {
          last = true;
        }
        catOutput.Append(postWidgetList(category, context, last));
      }

      if (!String.IsNullOrEmpty(catOutput.ToString()))
      {
        Out.WriteLine(String.Format("<ul>\n{0}\n</ul>\n", catOutput.ToString()));
      }
      else
      {
        Out.WriteLine("<!-- no posts to display-->");
      }
    }
  >%
</div>
```

## ■ Sort by archive month

```
<div id="archive-tabbed" class="widget">
  <%
    StringBuilder archiveOutput = new StringBuilder("");

    Asset archivesFolder = Asset.Load("/") + asset.AssetPath[0] + "/Blog/Archive/";

    last = false;
    AssetParams aFolderParams = new AssetParams();

    aFolderParams.SortOrder = SortOrder.OrderByDescending(AssetPropertyNames.Label);
    // Out.Debug.WriteLine(asset.Label);

    if (archivesFolder.IsLoaded)
    {
      List<Asset> archiveFolders = archivesFolder.GetFolderList(aFolderParams);

      foreach (Asset archiveFolder in archiveFolders)
      {
        aParams = new AssetParams();

        aParams.SortOrder = SortOrder.OrderByDescending("select_month");

        List<Asset> archives = archiveFolder.GetFileList(aParams);

        foreach (Asset archive in archives)
        {
          archiveOutput.Append(postWidgetAsset(archive, context, last));
          archiveOutput.AppendLine("<!--new item-->");
        }

      }

      if (!String.IsNullOrEmpty(archiveOutput.ToString()))
      {
        Out.WriteLine(String.Format("<ul>\n{0}\n</ul>\n", archiveOutput.ToString()));
      }
      else
      {
        Out.WriteLine("<!--no posts to display-->");
      }
    }
    else
    {
      Out.WriteLine("<!--no posts to display-folder not loaded-->");
    }
  <%>
</div>
```