



[www.hannonhill.com](http://www.hannonhill.com)



# Cascade CMS 101

## Welcome to Cascade CMS!

This introduction is the perfect place to start for technical users experiencing Cascade CMS for the first time. In presenting the basics of the system, we will walk you through a step-by-step process for creating and publishing a very simple website from scratch. After the initial learning curve has been mastered, creating and managing content in Cascade CMS is a breeze.

### **Prerequisites:**

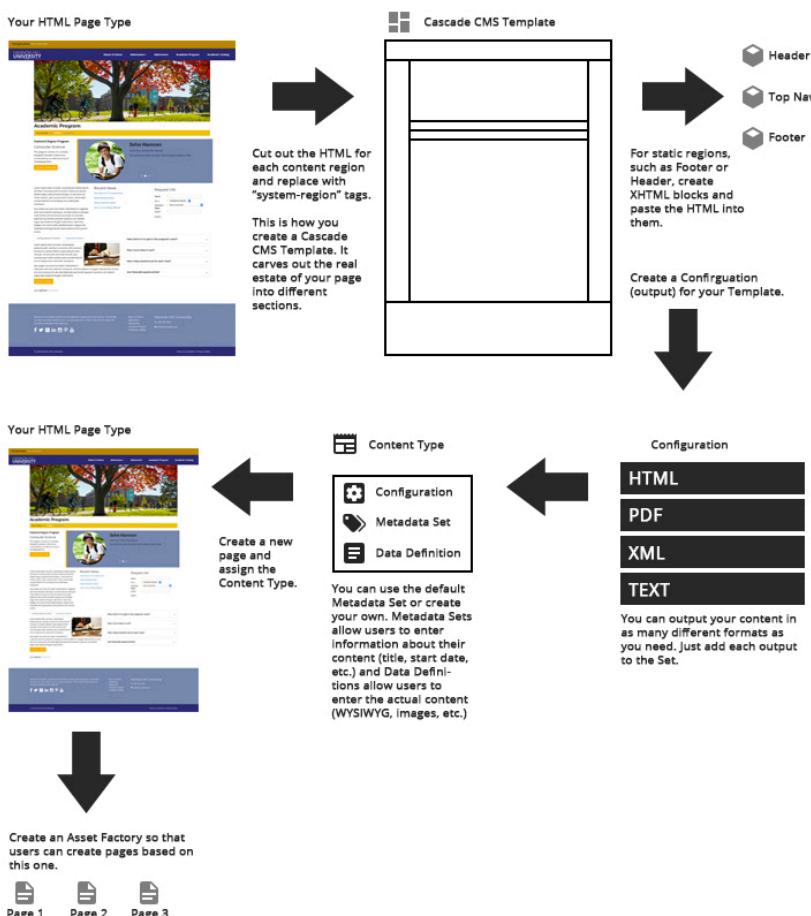
- A running instance of Cascade CMS (a non-production instance is recommended).
- A fresh copy of the Default Database provided by Hannon Hill.
- An Administrator level user account.
- A text editor of your choice (Notepad,TextEdit, Coda, Eclipse, etc.).
- A basic knowledge of XHTML.
- About two hours of time - because you're reading and creating for the first time.

# Cascade CMS 101

## Overview

Cascade CMS is an Enterprise Web Content Management System, meaning that its infrastructure is built in a way that provides a maximum level of scalability and flexibility. The purpose of an enterprise system is to accommodate fast growth (of contributors and sites) and be flexible enough to suit each organization's, department's, and contributor's needs. For an Enterprise WCMS to help you achieve your goals, it is crucial that it fully supports effective multi-channel distribution of content. The following provides an overview of the pieces that Cascade CMS has in place that make it such a powerful system, while making it easy for contributors to create and edit content.

This document is a step-by-step guide on how to integrate pages into Cascade CMS. In order to get yourself acquainted with the terminology and how the pieces will fit together, check out the diagram below.



### Cascade CMS Terminology:

#### **Template:**

Lays out the different regions on a Page.

#### **Output:**

An output of a page (such as HTML or PHP).

#### **Configuration:**

A collection of Outputs for a Page.

#### **Metadata Set:**

Fields that allow the user to capture information about an asset (such as title, keywords, review date, etc.).

#### **Data Definition:**

Fields that allow the user to enter Page content in a structured way.

#### **Content Type:**

A Page type defined by its Configuration, Metadata Set, and Data Definition.

#### **Asset Factory:**

A cookie cutter of an asset.



# Cascade CMS 101

## Login to Cascade CMS

We will begin by logging into Cascade CMS. Cascade CMS is supported by all major browsers. In your browser, navigate to the login screen for your instance using your favorite browser, supply valid credentials, and click Login.

A screenshot of a Cascade CMS login page. The page has a dark background with the Cascade CMS logo at the top. Below the logo is a white login form with the word "Login" in bold. The form contains two input fields: "Username" and "Password", both with placeholder text " ". Below the password field is a checkbox labeled "Remember me" and a blue "Login" button. At the bottom of the page, there is copyright information: "© 2020 Hannon Hill Corporation - v8.15.cloud1" and "Demo Sandboxes - Cascade CMS Web Content Management System - Hannon Hill".

Username

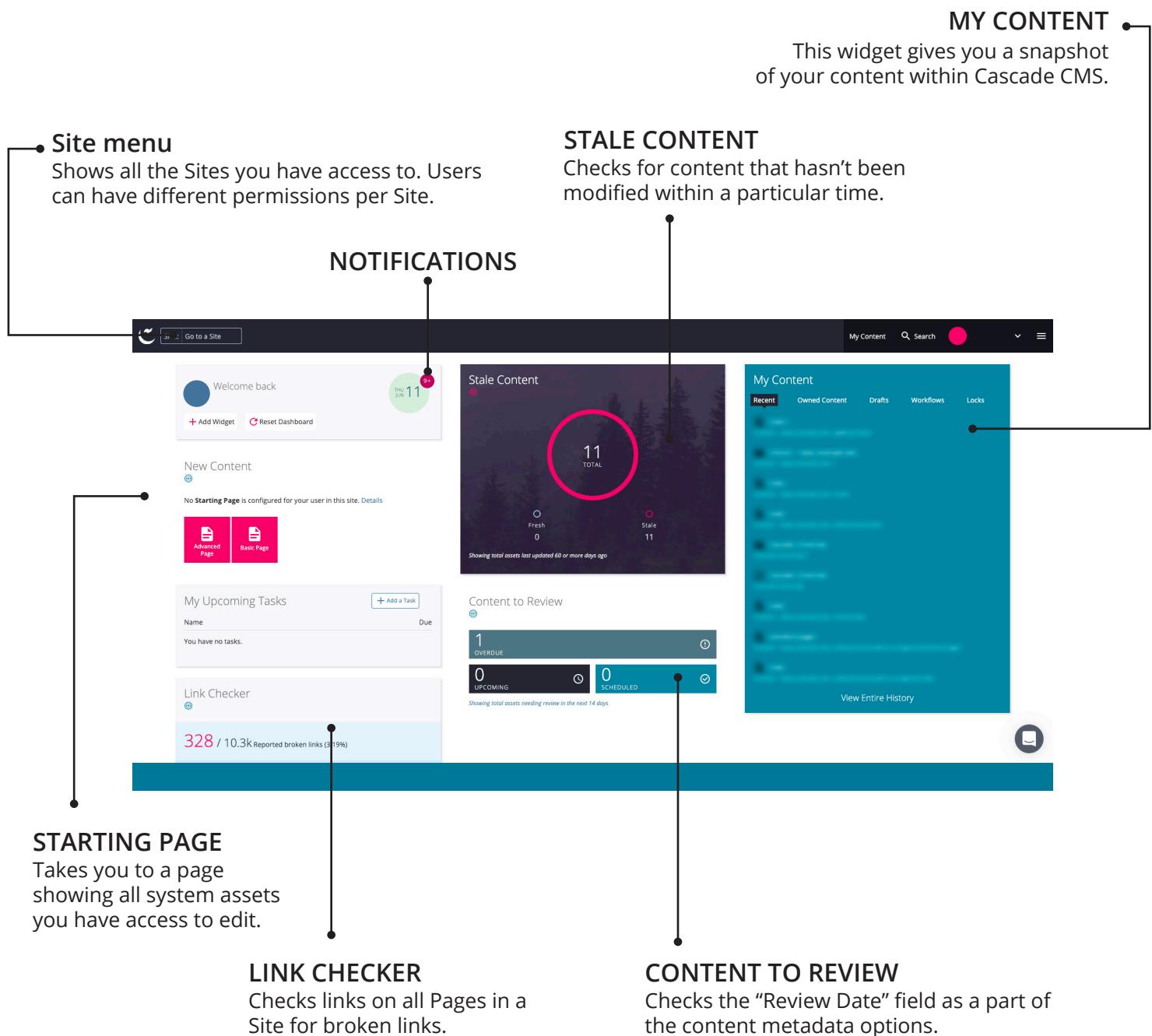
Password

Remember me   **Login**

© 2020 Hannon Hill Corporation - v8.15.cloud1  
Demo Sandboxes - Cascade CMS Web Content Management System - Hannon Hill

# Cascade CMS 101

After logging in to Cascade CMS, you will be directed to your Dashboard. The Dashboard can be personalized for each user and provides easy access to your Sites, upcoming tasks, Asset Factories, content you own, stale content, notifications, content to review, and a link checker.



## Prepare a Template

We will create a simple Template that will serve as the framework for all of our future Page assets. Templates contain one or more Regions that specify areas where content can be applied. In other words, Templates carve out the different regions on a page. We will create a plain Template in a text editor and then import it into Cascade CMS. Templates can also be manually created inside Cascade CMS, but more often than not implementations begin by importing an existing Template, so we'll show you how this can be accomplished.

### To create a Template:

Open your favorite text editor and paste the following XHTML structure (shout out to HTML5 support):

```
<!DOCTYPE html>
<html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title><system-page-title/></title>
</head>
<body>
    <header>
        <system-region name="HEADER"/>
    </header>
    <nav>
        <system-region name="NAVIGATION"/>
    </nav>
    <div id="content">
        <system-region name="DEFAULT"/>
    </div>
    <footer>
        <system-region name="FOOTER"/>
    </footer>
</body>
</html>
```

Save this file to your local machine as an HTML file: "basic.html".

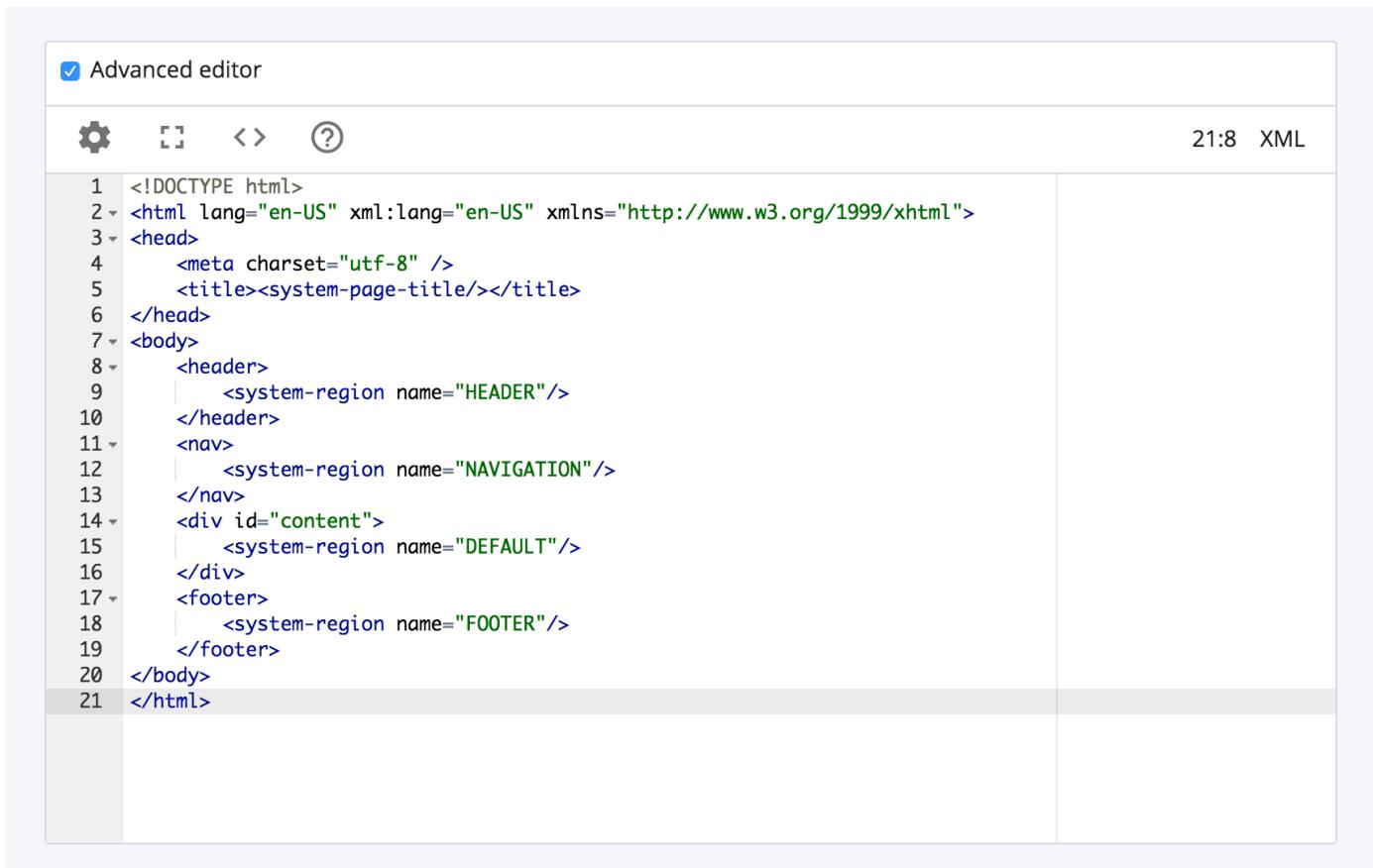
The `<system-region>` tag is the most basic and essential system tag that Cascade CMS provides. This tag defines regions within a Template where content can be inserted. In our example, we have created regions named **HEADER**, **NAVIGATION**, **DEFAULT**, and **FOOTER** which we will dynamically populate with page content. All Templates should include a **DEFAULT** region, which serves as the primary editable region of a page.

The `<system-page-title/>` tag pulls in the title as defined in each Page's Title metadata field.

## Creating a Template

Once you have created the Template and saved it to your local machine, it is time to import it into Cascade CMS.

- Navigate to your base folder **Introduction** > **\_cms** > **templates**
- Click **Add Content** > **Default** > **Template**.
- In the Template Name field type "Basic".
- To import the Template you can either paste the XHTML code into the Advanced Code Editor (shown below) or drag and drop / choose the file from your computer.
- Click **Save & Preview** > **Submit**.



The screenshot shows the Cascade CMS Advanced editor interface. At the top, there is a checkbox labeled "Advanced editor" which is checked. Below the editor area, there are several icons: a gear (settings), a square (new), a double arrow (refresh), and a question mark (help). On the right side of the editor area, the status bar displays "21:8 XML". The main area contains the following XML code:

```
1  <!DOCTYPE html>
2  <html lang="en-US" xml:lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4      <meta charset="utf-8" />
5      <title><system-page-title/></title>
6  </head>
7  <body>
8      <header>
9          <system-region name="HEADER"/>
10     </header>
11     <nav>
12         <system-region name="NAVIGATION"/>
13     </nav>
14     <div id="content">
15         <system-region name="DEFAULT"/>
16     </div>
17     <footer>
18         <system-region name="FOOTER"/>
19     </footer>
20 </body>
21 </html>
```

This Template will serve as the structure for your Cascade CMS Pages. It will provide the design around your content and allow you to create many different Pages, with potentially different styles, that use the same Page structure and design.

The **Template** is the first of five assets that is created in order for Page assets to be used.

## Creating a Configuration

When creating Pages inside Cascade Server, Configurations are just one of the assets within the CMS that help with the separation of content from design. Configurations help users manage and edit a large number of Pages that share common characteristics. One of Cascade CMS's most powerful features is the ability to output a single piece of content in as many different ways as you need, including HTML (even multiple versions of HTML if you have a separate mobile site), PDF, XML, or text. In simple terms, a Configuration is a collection of those different outputs.

**Configurations** 

Collection of page outputs (e.g. JSON, HTML, PHP) defining the type of content and the file extension among other things.

[Go to Configurations](#)

### To create a Configuration:

- In the Menu Bar click **Manage Site > Configurations**.
- Click **Add > Configuration**.
- In the Configuration Name field type "Simple Page".
- In the Output Name field type "Standard".
- For Type of Data choose HTML.
- In the File Extension field type ".html".
- For Template click **Choose Template**, search/browse to the Basic template, and click **Choose**.
- Check the Publishable checkbox.
- Click **Submit**.

The **Configuration** is the second of five assets that is created in order for Page assets to be used.

# Cascade CMS 101

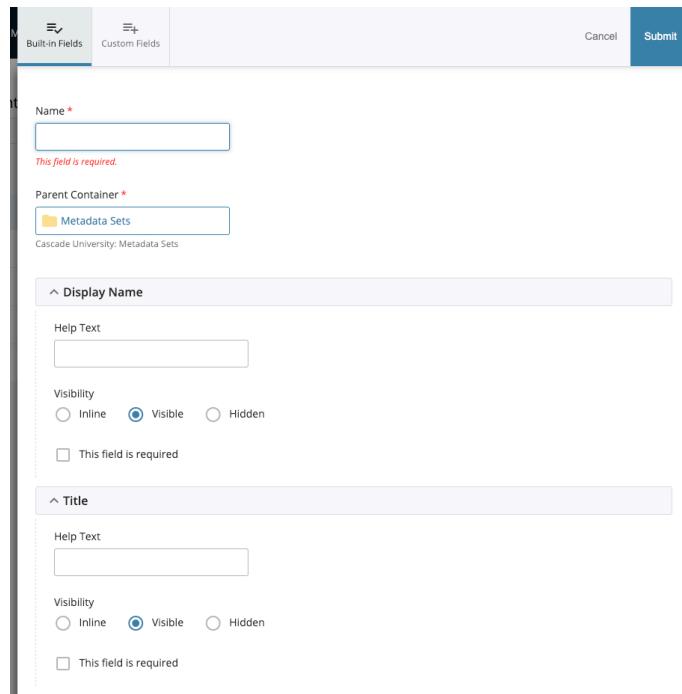
## Creating a Metadata Set

Metadata is data within the CMS that contains information about an asset. Fields such as title, summary, and keywords provide quick information about the content contained inside a particular asset. These Metadata Sets can be applied to all Home Area assets inside the CMS. A Metadata Set is simply a custom combination of the available fields and options for an asset type.

We will create a Metadata Set for our Simple Page so that we can manage how metadata is captured without having an impact on other Pages.

### To create a Metadata Set:

- In the Menu Bar click **Manage Site > Metadata Sets**.
- Click **Add > Metadata Set**.
- In the Name field type "Simple Page".
- Toggle the Title field to **Inline**.
- Toggle all other fields to **Hidden**.
- Click **Submit**.



*We recommend that unused fields be hidden to ensure that other users don't interact with fields that might change the way the asset is published and indexed around the system. Note that we did not choose to make any of these fields required.*

The **Metadata Set** is the third of five assets that is created in order for Page assets to be used.

# Cascade CMS 101

## Creating a Data Definition

A Data Definition enables Cascade CMS users that have little or no knowledge of web formatting standards to enter content via simple form fields. Content can be output into a structured page that already fits the overall website look and feel. The Data Definition is attached to a Content Type for a structured method of capturing and displaying content.

The Data Definition Builder is a drag and drop interface that can create many different types of content capturing fields. Each field has options for identifiers, labels, values, and advanced options.

### To create a Data Definition:

- In the Menu Bar click **Manage Site > Data Definitions**.
- Click **Add > Data Definition**.
- In the Name field type "Simple Page".
- Click and drag the **WYSIWYG** element into the content area.
- In the Identifier field type "body-content".
- In the Label field type "Body Content".
- Click **Save**.
- Click **Submit**.

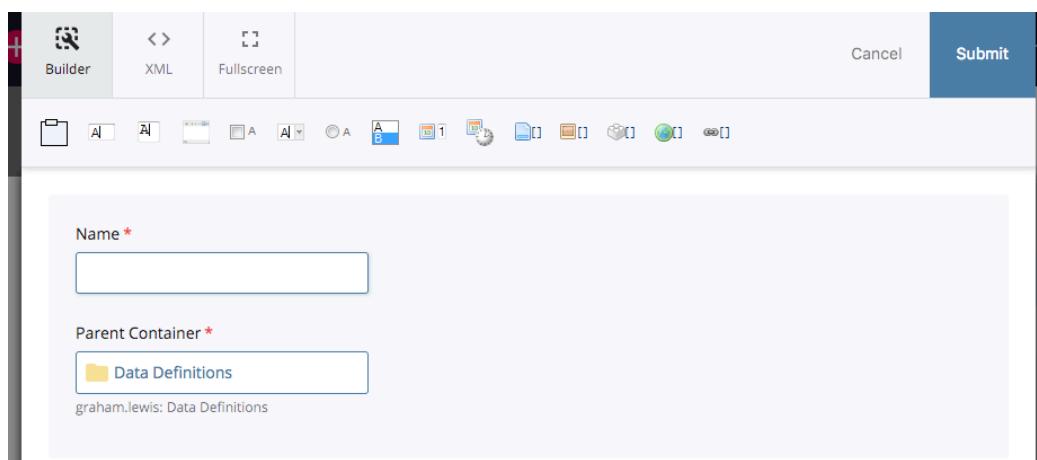


**Data Definitions**

Collections of fields for use when editing or creating pages and blocks.

[Go to Data Definitions](#)

The **Data Definition** is the fourth of five assets that is created in order for Page assets to be used.



## Creating a Content Type

Content Types provide a way for formally aggregating Configurations, Metadata Sets, and Data Definitions into a single administrative component that is applied to a Page. In other words, Content Types are page types and are defined by the page outputs (Configuration), the information about the page (Metadata Set), and the actual content (Data Definition). Content Types provide a logical way for users to identify the proper type to associate a Page with by grouping together the necessary settings and content fields.

### To create a Content Type:

- In the Menu Bar click **Manage Site > Content Types**.
- Click **Add > Content Type**.
- In the Name field type "Simple Page".
- In the Configuration field search/browse to the Simple Page Configuration.
- In the Metadata Set field search/browse to the Simple Page Metadata Set.
- In the Data Definition field search/browse to the Simple Page Data Definition.
- Click **Submit**.

The **Content Type** is the fifth asset that is created in order for Page assets to be used.

**Content Types** 

Page types consisting of a Configuration, a Metadata Set and a Data definition

[Go to Content Types](#)

## Creating Static Content Blocks

Managing content from a single asset and having that content be reused in multiple locations, and potentially in multiple Sites, is a huge differentiator for Cascade CMS. Static XHTML Blocks are one of the many ways to manage a piece of content and use it in several different locations. We will create basic XHTML Blocks that are used in the HEADER and FOOTER regions of our Template.

### Header Block

- Visit <http://www.hannonhill.com> and download the Hannon Hill logo.
- Navigate to your base folder **Introduction > \_files > images**
- Click **Add Content > Default > File**.
- Drag and drop or choose the image file from your computer. Note that this automatically populates the File Name field.
- Click **Save & Preview > Submit**.

### To create an XHTML Block:

- Navigate your base folder **Introduction > \_cms > blocks > xhtml**
- Click **Add Content > Default > Block**.
- For the Block Type select XHTML/Data Definition.
- In the XHTML/Data Definition Block Name field type "site logo".
- In the WYSIWYG editor click the **Insert/Edit Image** toolbar button.
- For Image Source choose Internal.
- For Image click **Choose File**, search/browse to the Hannon Hill logo, and click **Choose**.
- In the Image Description field type "Hannon Hill Logo".
- Click **OK**.
- Click **Save & Preview > Submit**.

### Footer Block

- Navigate to your base folder **Introduction > \_cms > blocks > xhtml**
- Click **Add Content > Default > Block**.
- For the Block Type select XHTML/Data Definition.
- In the XHTML/Data Definition Block Name field type "site footer".
- In the WYSIWYG editor click the **Source Code** toolbar button and enter the following code:  

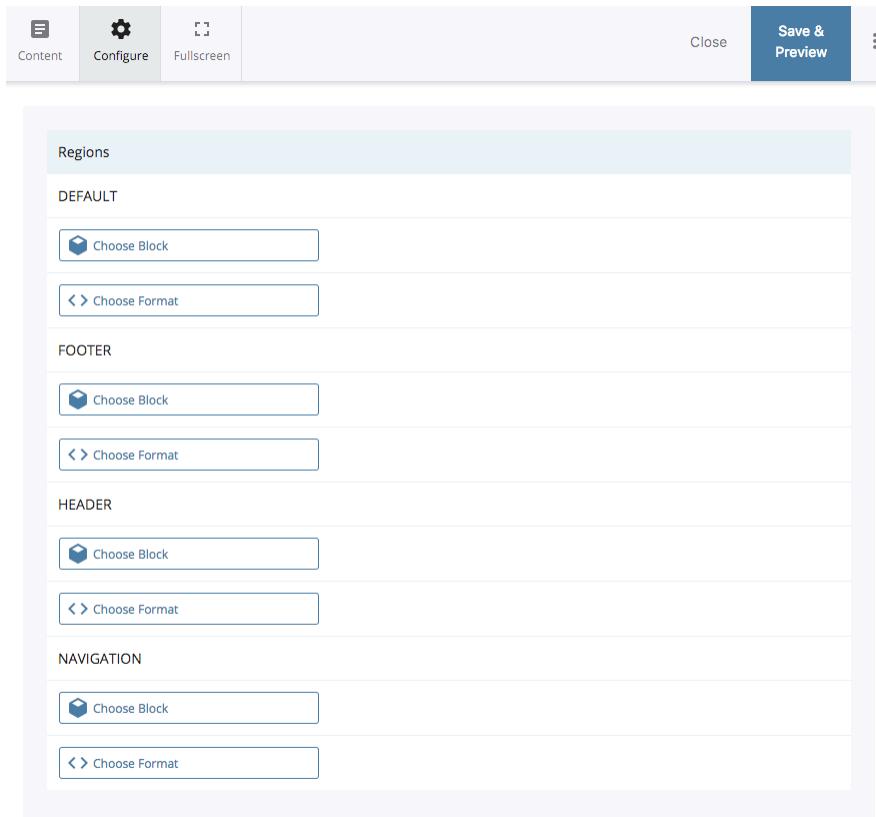
```
<p>Copyright Hannon Hill. All rights reserved.</p>
```
- Click **OK**.
- Click **Save & Preview > Submit**.

## Assigning Static XHTML to Page Regions

With our static blocks of code for our HEADER and FOOTER Regions created, it is time to assign those Blocks to Template Regions. The reason we will be assigning these Blocks directly to the Template, as opposed to an individual Page, is so they can be inherited by the Page regions on a large scale. All Page assets that make use of a particular Template will inherit Block and Format information in the Regions that have assignments.

### To assign Blocks to a Template:

- Navigate to your base folder [Introduction > \\_cms > templates](#)
- Click the Basic Template and then click [Edit](#).
- Click the [Configure](#) tab.
- Find the HEADER and FOOTER Regions.
- Click [Choose Block](#) and search/browse for their respective Blocks.



## Creating a Page Asset

Now that we have gone through the process of creating a Template (with assigned Blocks), Configuration, Metadata Set, Data Definition, and Content Type, we can create a Page asset. We still have a few more steps in order for our Page to be fully functional, but we will go ahead and create the Page. We do this in order to take advantage of the XML output. This raw XML data set will be beneficial for writing our Format as it points out in plain text the XML nodes that we must reference for accessing our content for display purposes.

The initial Page that we create we will used as the basis for other Pages that can be created quickly and effectively. We will use this first Page as the "Base Asset" for an Asset Factory. An Asset Factory is a simple cookie cutter for creating a new asset.

### To create a Page:

- Navigate to your base folder **Introduction > \_cms > base-assets**
- Click **Add Content > Default > Page**.
- For the Page Type select Simple Page and click **Choose**.
- In the Page Name field type "simple-page".
- Click **Save & Preview > Submit**.

*Note that we left all other fields blank and submitted the form. It is usually recommended that base assets are left completely empty so that when an end user interacts with the Asset Factory, a blank version of the page is presented to them for filling in their own original content.*

## Dynamic Content Blocks

With the base asset in place, we can start working on a way to have the Page content dynamically placed into the Page. As each Page is loaded, they may share the same Template and possible content Regions. We want the content to change as we move from Page to Page. One of the major benefits of Cascade CMS is the separation of content from design. The content is stored as XML in the database and is inserted into Page Regions (which are defined in the Template) through the use of static and/or dynamic Index Blocks.

The first dynamic Index Block that we will create will serve the purpose of loading the Page content into the DEFAULT Region of the Template. As a user clicks to view different Pages, the underlying Template will remain the same while the body content in the DEFAULT Region changes based on the Index Block's rendering. This Index Block, which is usually called the Calling Page Block is one of the most used Blocks in Cascade. To further explain the purpose behind its functionality, this Block will access the stored content for whatever Page you are currently looking at and load it into the Region this Block is assigned to.

### To create an Index Block:

- Navigate to your base folder [Introduction > \\_cms > blocks > index](#)
- Click **Add Content > Default > Block**.
- For the Block Type select Index and click **Choose**.
- In the Index Block Name field type "calling-page".
- For the Index Type field choose Folder Index.
- For the Rendering Behavior field select Render normally, starting at the indexed folder.
- In the Depth of Index field type "0".
- For the Indexed Asset Content field check Append Calling Page Data, Regular Content, and User Metadata.
- For the Indexed Asset Types field check Pages.
- For the Page XML field select Render page XML inline.
- Click **Save & Preview > Submit**.

The different sections of options correspond to a number of different things:

**Index Type:** This describes how the Block will gather information in the system. It will use the folder structure (Asset Tree).

**Rendering Behavior:** This is the behavior (logic) that is used to traverse the folder structure. "Render normally" follows a brute force path to determine which assets should be indexed relative to the Page you are currently viewing. It also takes into account the "Depth of Index" specified to limit how many levels forward (deep) to gather.

## Dynamic Content Blocks

**Depth of Index:** This dictates how many levels forward (deep) the Block will search for gathering information. Relative to the asset you are currently working with, siblings are level 1, nieces/nephews (children of siblings) are level 2, and so on.

**Indexed Asset Content:** These correspond to different aspects of asset information that can be included or excluded from the returned data set. "Regular Content" will provide you with the Display Name and path information of the asset. "User Metadata" will include anything collected from the Metadata Set of the asset - Title, Summary, Keywords, Description, Author, Start Date, etc. "Append Calling Page Data" is how the information for the Page you are currently viewing is included. As a by-product of the "Depth of Index" being 0, information for the current Page is ignored in addition to information from sibling assets.

**Indexed Asset Types:** You can limit the types of assets that the Block will gather information from.

**Page XML:** This option corresponds to whether or not you would like to have access to content gathered from the Data Definition level. The reason you can choose to not return that information is that Index Blocks can be used for a variety of other purposes besides just rendering content for a single Page. In the case of dynamically creating a navigation based on the Pages that exists contextually, Page content is not something that is used when building an XHTML list of Page titles. Therefore it is not advised to include the Page content in the set of XML data that is made available in a region.

# Cascade CMS 101

## Format Transformations

Paired with an Index Block, a Format in Cascade CMS parses XML and helps transform raw data into another format. XML data can be accessed and then wrapped inside of XHTML tags, PDF FO tags, or even RSS tags. Again, the system separates content from design by storing asset information as XML. This leads to greater flexibility when it is time to output the information. Page fields that have no style information (text inputs, drop-down menus, etc.) store just the core content while allowing you to define the style aesthetics every time that piece of information should be output into the Template design.

Simply speaking, all your content is stored as XML. But XML is not pretty:

6.jpgsite://Andy - www.example.edu/\_assets/images/slider/homepage-banner-6.jpgAndy - www.example.eduhomepage-banner-6.jpgHeading for Slider 1//\_assets/images/slider/homepage-banner-4.jpgsite://Andy - www.example.edu/\_assets/images/slider/homepage-banner-4.jpgAndy - www.example.eduhomepage-banner-4.jpgHeading for Slider 2Caption for Slider 2White White Button/\_assets/images/slider/homepage-banner-5.jpgsite://Andy - www.example.edu/\_assets/images/slider/homepage-banner-5.jpgAndy - www.example.eduhomepage-banner-5.jpgHeading for Slider 3Caption for Slider 3Yellow Yellow Button/Journey With UsAt Hannon Hill University, we pride ourselves at being not only affordable but upfront with all costs and fees. No surprises./With over 25 undergraduate programs and 13 graduate programs, we are one of the finest academic institutions inside of the Atlanta Tech Village./academic-program/indexpriority0.5changefreqalwaysnavYessite://Andy - www.example.eduacademic-program/indexAndy - www.example.eduindexAcademic Program DemoAug 15, 2014 5:38 PMThere are so many things to do on our Campus that you will not know where to start. From our Coffee Shop to our Game Room, we have everything you could want!/Register for Open HouseRegister Now!/admissions/family-eventpriority0.5changefreqweeklynavYessite://Andy - www.example.eduadmissions/family-eventAndy - www.example.edufamily EventJan 1, 2018 12:00 AMhttps://www.youtube.com/watch?v=S6oTeto43Tw&t=1s

So by applying a Format, you can style the content exactly how you want to:

### JOURNEY WITH US

 COSTS & AID

At Hannon Hill University, we pride ourselves at being not only affordable but upfront with all costs and fees. No surprises.

 ACADEMICS

With over 25 undergraduate programs and 13 graduate programs, we are one of the finest academic institutions inside of the Atlanta Tech Village.

 VISIT CAMPUS

There are so many things to do on our Campus that you will not know where to start. From our Coffee Shop to our Game Room, we have everything you could want!

[LEARN MORE ▶](#)

## Format Transformations

The Format uses the XML structure of the paired Index Block for accessing the XML data. Each Index Block can have a different XML structure, so it is important to understand how Formats process the XML node structure.

For this example, the Index Block we just created has a core structure as follows:

```
<system-index-block current-time="1337619570322" name="current page" type="folder">
    <calling-page>
        <system-page current="true" id="36d151ae7f000001006c8343adff4a59">
            <name>page-name</name>
            <!-- Metadata Set fields listed here as XML nodes -->
            <path>/path/to/page-name</path>
            <site>site name</site>
            <link>site://site name/path/to/page-name</link>
            <system-data-structure definition-path="Simple Page">
                <!-- Custom Data Definition fields listed here as XML nodes -->
            </system-data-structure>
        </system-page>
    </calling-page>
</system-index-block>
```

Our Format needs to traverse this specific XML node structure for accessing node data and using it to output XHTML. As you create more Index Blocks for other purposes, knowing the XML data structure is important as it will dictate how you will create the Format. It is also worth noting that a single Index Block can be used multiple times across different Page types and have a different Format paired with it each time. The same single source of content can be repurposed in different ways depending on the Page type and region.

## Format Transformations

There are two languages that are supported within Cascade CMS for doing XML data transformations: Velocity and XSLT.

You are not locked in to using only one language or the other. Both languages can be used throughout the system. You can even use both Format types on the same Page.

Example #1 - To create a Format with Velocity and Index Block XML:

- Navigate to your base folder [Introduction > \\_cms > formats](#)
- Click [Add Content > Default > Format](#).
- For the [Format Type](#) select [Velocity](#) and click [Choose](#).
- In the Velocity Format Name field type "simple page content".
- Paste the following in the content area:

```
## Reference to the system-page XML node
#set ( $page = $_XPathTool.selectSingleNode($contentRoot, "/system-index-block/
calling-page/system-page") )
## Reference to the system-data-structure XML node
#set ( $data = $page.getChild("system-data-structure") )
## Reference to the title XML node
#set ( $title = $page.getChild("title") )
## Reference to the body-content XML node
#set ( $content = $data.getChild("body-content") )
## XHTML code for the region
<h1>$_EscapeTool.xml($title.value)</h1>
$_SerializerTool.serialize($content, true)
```

- Click [Save & Preview > Submit](#).

## Format Transformations

Example #2 - To create a Format with Velocity and the Cascade CMS API (no Index Block required):

- Navigate to your base folder **Introduction > \_cms > formats**
- Click **Add Content > Default > Format**.
- For the **Format Type** select **Velocity** and click **Choose**.
- In the Velocity Format Name field type "simple page content".
- Paste the following in the content area:

```
## Reference to the current page's body-title structured data node
#set ( $title = $currentPage.metadata.title )

## Reference to the current page's body-content structured data node
#set ( $content = $currentPage.getStructuredDataNode("body-content") )

## XHTML code for the region
<h1>$_EscapeTool.xml($title)</h1>
$content.textValue
```

- Click **Save & Preview > Submit**.

## Assigning Block and Format for Structured Page Content

Pages with Data Definitions require Block and/or Format assignments for accessing and displaying the content. We have created a Block and Format for gathering and displaying the content for our Simple Page.

We will assign the Block and/or Format to the DEFAULT Region of the Page. Next, we assign these assets at the Configuration level so that only Pages using the Simple Page Configuration are affected. This means that as new Page types are created, they can share the same Template but use the Template Regions differently.

### To assign a Block and/or Format to a Configuration:

- In the Menu Bar click **Manage Site > Configurations**.
- Click the **Simple Page Configuration** and then click **Edit**.
- Locate the **DEFAULT Region** of the **Standard Output**.
- If you created a Format using Example #1, click Choose Block and search/browse for the calling-page Block.
- Click **Choose Format** and search/browse for the simple-page-content Format.
- Click **Submit**.

### Configurations



Collection of page outputs (e.g. JSON, HTML, PHP) defining the type of content and the file extension among other things.

[Go to Configurations](#)

## Creating an Asset Factory

Asset Factories provide a framework which allows users to create new assets with all of the necessary options, configuration, and content fields already in place. An end user does not need to worry about the Template being assigned or the Data Definition fields as Administrators have created a blueprint asset that already contains all of the important assignments that they will need. Users will select from a predefined list of Asset Factories the asset type they would like to create.

### To create an Asset Factory:

- In the Menu Bar click **Manage Site > Asset Factories**.
- Click **Add > Asset Factory**.
- For the Asset Factory Type select Page and click **Choose**.
- In the Name field type "Simple Page".
- For the Base Asset field click **Choose Page** and search/browse for simple-page in the /\_cms/base-assets directory.
- Click the **Plugins** tab.
- In the Available Plugins drop-down menu, select the Title to System Name Plugin and click **Add Plugin**.
- Click **Submit**.

*Note that we still have all optional fields in our new Asset Factory asset. This is because we created a "base asset" earlier that needed to remain blank. At this point you should now revisit the Metadata Set and Data Definition for marking certain important fields required.*

We have now created a custom Asset Factory for users to interact with. You should now see your custom Asset Factory listed as a selectable menu item called Simple Page in the Add Content drop-down menu in the Menu Bar. When users select this option, a new Page will be initialized for them that already has the appropriate Content Type attached.

### To create a Page:

- Navigate to your base folder **Introduction**.
- Click **Add Content > Simple Page**.
- In the Title field type "index".
- Fill in some content for the Body Title and Body Content.
- Click **Save & Preview > Submit**.

With a custom Asset Factory available, you should be able to create basic Pages like these in the Asset Tree to build out an example Site. Our next step is to publish it out to a web server. We will now set up the components needed for publishing.

## Publishing a Site

### You'll need the following pieces of information to publish:

- The publishing URL: Where on the web will this Site be viewed.
- Login credentials to a web server: A username/password for writing/removing files through (S)FTP.

The first piece of information is publishing URL. Earlier in the process, when we created a new Site object, we provided a URL for the Site (<http://www.example.edu>). We need to update this piece of information, which will allow Cascade CMS to know exactly what URL should be used during the publish process for creating the links/path to each asset.

### To update the Site URL:

- Make sure you are still within your Site. Check that the Site drop-down menu in the Menu Bar displays your Introduction Site. If not, select it from the menu.
- In the Menu Bar click **Manage Site > Site Settings**.
- Update the URL field.
- Click **Submit**.

The login credentials to the web server you want to publish to will be attached to an asset we call the Transport. The Transport is how you connect to where you want to publish. The other half of the publishing asset is called the Destination. The Destination is where on the web server (which directory) you want to publish the files in. Separating these two assets can be beneficial when you are managing multiple Sites that are stored on the same web server. One Transport can be used by multiple Destinations for publishing several Sites into different directories on the same server.

### To create a Transport:

- In the Menu Bar click **Manage Site > Transports**.
- Click **Add > Transport**.
- For the Transport Type select FTP and SFTP and click **Choose**.
- Fill in the required fields and provide the appropriate information for connecting to your web server.
- Click **Submit**.

Transports

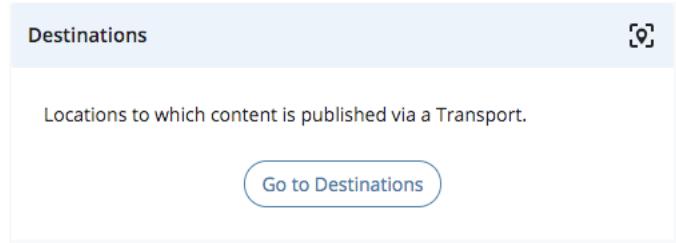
FTP and SFTP servers, local file system destinations, or MySQL databases to transfer content to.

[Go to Transports](#)

## Publishing a Site

### To create a Destination:

- In the Menu Bar click **Manage Site > Destinations**.
- Click **Add > Destination**.
- For the Transport field click **Choose Transport** and search/browse for your Transport.
- Make sure Enable Destination is checked.
- In the Web URL field type the same publishing URL used in the Site Management area.
- Click **Submit**.



We will now publish out the Site to the web server.

### To publish a Site:

- In the Menu Bar click **Site Content > Publish**.
- Verify that your Destination is checked.
- Click **Submit**.

You can check the progress of your publish job in real time by clicking the **System Menu** button > **Publish Queue**. When the publish job is complete, you should also receive a Notification detailing the publish job.

Check your (S)FTP server and verify that all of the assets published out correctly.

# Cascade CMS 101

## Wrapping Up

### Congratulations!

You have created a Site entirely from scratch and published that Site to a web server. Hopefully this guide has laid a foundation for understanding Cascade CMS and increased your confidence and abilities.