University of Alberta:
Cascade CMS Post-Training Resources

Training and Resources provided by:

Sam Skinner
Products Trainer
sam.skinner@hannonhill.com
Mobile: 678-585-7757

If you have follow up questions about the training, please contact me. For other technical questions, please contact support@hannonhill.com. Questions about your account or contracts should be directed to: charlie.holder@hannonhill.com.

Please share the following resources with all training participants:

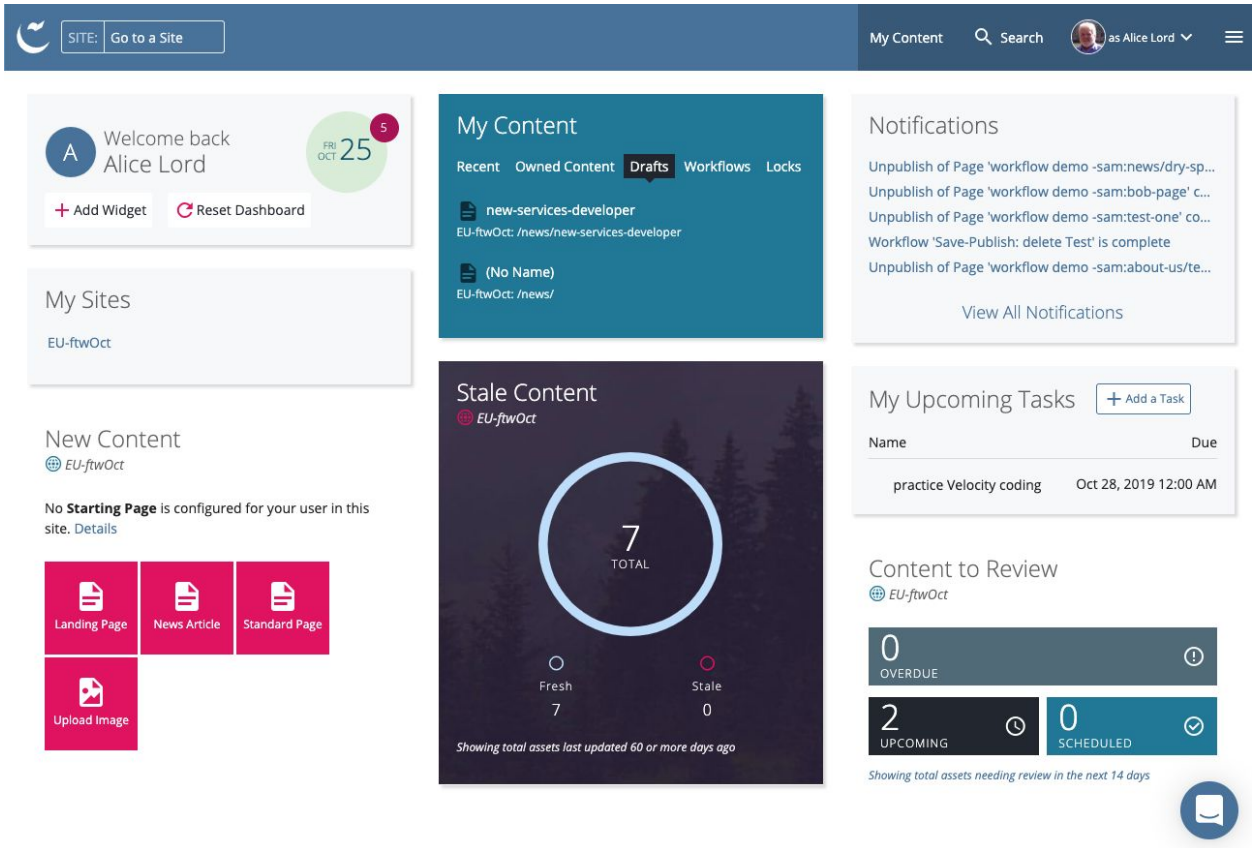
Links to Videos

#1: "Train the Trainer"

Training Schedule with Links

Section 1: Cascade Interface – *Dashboard & Site Content views*

The Dashboard is an organizing point for working in Cascade. Widgets can be added, deleted or moved to configure the Dashboard to each user's needs. Four widgets are general in nature[1]; the others filter by site. Virtually every asset is clickable and has an associated dropdown menu.



In the Site Content view, we see the folder-and-file layout of a site. In the left column, folders can be toggled open and closed with a single click; their dropdown is accessed by a right click or two-finger click. To open a folder, find the arrow to the right of its name or click on the folder name in the main window. The "...More" menu often contains links not seen in the folder-associated dropdowns. Dropdowns vary by user abilities and asset type.

Asset factories are seen via the "+ Add Content" link, while the "Site Content" link always returns the user to the root folder of the site. A "Manage Site" link may or may not be visible (depending on a user's role) but, if present, opens a view where many site assets are created.

[1] *My Sites, My Content, Notifications, My Upcoming Tasks.*

Section 2: Providing Content – *Asset Factories & Controlling the Creating Process*

Asset factories provide a means for quickly creating or up-loading new content. They are custom-built by each institution and for each site, but the overall interface is common. Generally, an asset

factory form consists of four sections: 1) fields to provide a filename and location for the asset, 2) a set of metadata fields, 3) a data definition section, and finally, a "Tags" section for categorizing content.

When a user creates or edits new content, a draft is automatically saved and can always be retrieved. Once a first version has been saved a user can toggle between the saved ("current") and draft versions. Users also can provide versioning comments as each draft is submitted.

The process of creating content can be regulated via the use of Content Checks and Asset Naming Rules along with the use of asset factory Plugins. To simplify modifying certain sections of a page, Editable Fields can be set via the "Content Type" for a page.

Individual pages can be marked to be reviewed at set times or on a schedule. A second, more inclusive, approach is to use the Stale Content report to catch pages that have not been reviewed within a specific time-frame.

When a page is published manually, clicking on the "Live" link in the "...More" menu will take a user to the published version of the page, as it exists on the server, allowing a user to visually determine if publishing was successful.

Section 3: Administration – *Users, Roles, Groups and Workflows*

The Administration area has six panels with links for system-wide actions. The two most-used are Preferences and Security & Authentication. In the Preferences section, one can set up System-Wide tags or add words to the System Dictionary. Additionally, the Preferences link has a tab for "Integrations & Plugins" where one enables connections with Clive, SiteImprove and two Digital Asset Management platforms (WebDam, Widen). A variety of other system-wide settings are actuated here as well.

The Security and Authentication panel provides a means of setting up LDAP Configurations or Custom Authentication. This also is where user accounts can be set up manually.

Regardless of how a user account is established, every user must belong to a Group and be assigned a system-level Role. For most users, this will be a Contributor role that allow them to log into Cascade but do nothing else; for such users, a Site-level role will provide access to the relevant sites and determine the actions they can perform with in the site. An individual with the system-level Administrator role needs no other roles as they have complete access to every feature and asset within Cascade. Overall, what a user can do within Cascade is managed via Permissions and their ability to read or write (edit) content is managed via Access. Generally, the rule is to have few roles but many groups and manage individual users through their group memberships.

Finally, in the Tools panel, one can set up the Logging Configurations and access both Log files and System Information.

Workflows[1] are a means of controlling and monitoring the creation, editing and managing of assets. As such they are part of the process of content management but their successful deployment depends on a good understanding of Permissions, Access and Roles. Hannon Hill provides a set of Sample Workflows on Github. Additionally, custom training on workflows is available as the design and implementation process can be complicated.

[1] *Be sure to view the pdf link at the bottom of the Overview article.*

Section 4: Creating a Page – *Convert an HTML Page to a Cascade Page*

At the core of Cascade is a set of interconnected assets that allow the creation and rendering of pages within the Content Management System. Whether working with a pre-built design template or converting pre-existing pages from a current website, there are four initial steps necessary to work within Cascades' "xhtml" environment. 1) All html tags must be paired with a closing tag or made self-closing. 2) Pathways to files or other assets must begin with a leading slash (and rewritten to reflect new asset locations). 3) Images accessed via the CSS must be wrapped in pseudo-tags, call "system-asset" tags. 4) Passthrough codes (e.g., #protect) are used to manage code display and usage within Cascade especially for non-markup languages.

At the base of the system of assets is a Template (also see here) which provides the HTML scaffolding for generating pages. A template uses a variety of System Tags[1] to pull code and text into various regions of a rendered page. Both code blocks and formats (See Section 5 below.) can be attached to the regions at the template level. A system-region with the name "Default" provides special functions within Cascade starting with inherently providing a WYSIWYG editor for that region of the page.

While a template provides the html scaffolding, a Configuration provides the framework for building pages of different types or providing different ways to output content. The output of a Configuration is based on a specific template and a specific data type. It can inherit the blocks and formats attached to the template but also allows those regions to be re-configured as the basis of a new page or output type. This allows for considerable template reuse. Cascade uses special tags for Linking to Specific Outputs.

Cascade provides two basic ways for users to provide the content of pages. One method is data input via Metadata Fields (also see here) both pre-built and custom. In the set up for building pages, one selects relevant pre-built fields and hides the rest. The hidden fields can be activated at a later time if found useful. It is also possible to create a variety of custom fields as needed. One advantage of using the pre-built metadata fields is that their content can be pulled directly onto a page view (e.g., title and display-name information).
A second, more versatile, method for users to provide content is via a Data Definition. A data definition is basically a hand-constructed form allowing a user to input images, select blocks of content or use a WYSIWYG editor to provide complex content.

The two assets, a Metadata Set and a Data Definition, work in tandem allowing a user to provide all the content for building a page.

Finally, all the preceding assets are tied together in creating a Content Type (aka Page Type). At this point one can actually create and render a page in Cascade. Often the next step is the creation of an Asset Factory to stamp out multiple pages built on the same underlying framework.

[1] *Not to be confused with the Asset Factory tags used for categorizing content.*

Sections 5 - 7: From Static to Dynamic Content – *Coding Velocity Formats for XML Data*

To render content dynamically, Cascade makes use of formats. The content data for a page (text, images, etc.) are retrieved via an XML file. Therefore, Cascade supports two XML languages, Velocity (v 1.7) and XSLT (v 1.0). Both languages are widely used outside of Cascade CMS and are well supported by their respective communities but, of the two, most Cascade CMS users prefer Velocity as it is easier to read, learn and use.

A general approach to working in either language is to first create an Index Block to generate the appropriate XML file for the task at hand and then write the code to pull the relevant data from the XML file and render it on a page. For example, we created an Index Block called "current page" (aka "calling page") which generates an XML file containing both the user-supplied metadata and the

content from a page's data definition form. This is one of the most common blocks used in Cascade as it contains the full information for rendering most any page or page type. We also created different index blocks for pulling information for a navigation system ("nav block"), a breadcrumb trail ("current folder hierarchy") and a content-type index block for pulling a collection of news articles ("news articles")[1].

Examples of these blocks are located in the _common site in whatever instance of Cascade was used for your training. Complete copies of the formats created during the training are also located in the _common site, in the formats folder. From that site, these assets can be used anywhere within Cascade. (The formats also are available on Google Drive). Lastly, remember that as a Site Administrator you can export your site and import it into your own Cascade environment, if desired.

Additionally, in writing a format based on the XML file from an index block, we first used the XPathTool to access the file. Like Velocity, XPath is developed and maintained independent of Cascade. It is a powerful and agile means of pulling information from any XML file.

For users of Velocity, there are a number of Cascade-specific variables and tools allowing pages to be rendered *without* the use of Index Blocks. These Advanced Velocity tools are described in Cascade Velocity Tools (see also Locator Tool). Hannon Hill offers training in the use of these tools and this object-oriented approach to writing formats. We also offer training in the use of XSLT or for converting from XSLT to Velocity.

[1]*Since this block utilizes a specific Content Type, it may be necessary to reset the Content Type for the block to work appropriately in a given site. Also the "Max Rendered Assets" may need to be increased.*

Section 8: Manage Site – *Settings, Publishing, Connectors, WYSIWYG Configurations*

To finish our tour of the Manage Site area, click first on the Site Settings link. Here one can edit the name of a site and provide the url to be used by the "Live" link and links to this site from other sites. (Discussed above; this is separate from the setting for publication though the url may be identical.) The url for connecting a site to SiteImprove is also set here. In addition there are a number of site-level settings that mirror and override the system-level settings in the Preferences panel of the Administration area.

In the Roles tab, the site-level roles provide various groups (or users) access to the site. Generally, a Site Admin role provides broad access to a site including the Manage Site area. Most users and groups will have a Site Contributor role giving them limited abilities within the site. Both roles can be modified to enlarge or reduce the actions available to that role. Remember the ability to create and edit content is set by Access not by the role.

Three panels directly connect to publishing. A Transport holds the credentials for accessing a server (or a database, an S3 bucket or a filesystem). For multiple servers with differing credentials, create multiple Transports. Once created, a Transport can be used across multiple sites. The second part of the key for publishing is a Destination. Here you identify the server directory and the url to access it along with the Transport holding the access credentials. It is possible to publish to multiple Destinations and manage which are optional and which are set as default.

Publish Sets and Publishing Related Content allow further control over publishing. Also see Load Balancing if that is of concern. (Note: Load Balancing requires a special license from Hannon Hill and balances across the entire application, not just publishing.)

Connectors are a related concept in that they hold the credentials for connecting to external resources. These include WordPress, Google Analytics and Twitter. Both WordPress and Twitter connections push selected content to those platforms.

The WYSIWYG editor in Cascade is based on the open-source tinyMCE; it is fully configured. In some cases it may be desirable to limit its functionality for your general content providers as outlined in the preceding link. Here also one can attach a CSS file to pull the relevant CSS classes and ids for styling content in the editor as it will appear on the rendered page. It is also possible to make specific styles available to your content providers for their use. Whenever WYSIWYG features are limited, a link is provided (∞) for Administrators to override the limitations.

Lastly, Tags can be set at the system-level and site-level and both appear in the Tags dropdown at the bottom of Asset Factories for creating pages. In this way a set of broadly-used tags (set at the system-level) can intermingle with tags of specific relevance for a particular site. Each site can have its own set of site-level tags.

Section 9: Deconstructing a Page

Once a page exists, there may arise a need to discern how the page was created. Audits provide a record of "who did what"[1]; Versions provide a means of viewing previously saved versions of a page[2], comparing multiple versions visually and reverting to an earlier version if needed. When the Versions link is clicked, the "... More" menu displays a list of available versioning actions.

To discover the assets on which a page is built, click on the "Details" link at the top of a page and then open the Design tab. This reveals the Content Type, Configuration, MetaData Set and Data Definition that come together to render the page. All are clickable.

An alternative is to click on "Show Regions" in the ...More menu to see a list of the Regions used to assemble the page. Hovering over the name of a region causes Cascade to highlight the content derived from that region. Clicking on the Region name reveals the block and/or format that generates the content.

The above two features greatly facilitate identifying assets involved in rendering a page. The "Render Metrics" link (in ...More) provides the time metrics associated with rendering each region. The times reflect the time taken to assemble and render the page within Cascade, not how the page renders on a live server.

Additionally one can use "Inspect Element" (Chrome), or its equivalent in other browsers, to view both CSS and Javascript issues in rendering. Finally, viewing the XML output of a page can assist in troubleshooting issues.

On a related note, since assets (e.g., blocks, formats, content types) can be pulled from multiple sites within Cascade, it sometimes becomes necessary to disassociate those relationships. In this case open the relevant asset and click on the Relationships link in ...More. All the pages that link to that asset will be listed with an indication of the site where they are located (look at the breadcrumb trail). This also is useful in identifying all pages that use a particular image.

To change an asset underlying all the pages in a folder, open the folder and click on the Bulk Change option in ... More. One can change the underlying Content Type, Metadata Set, or Data Definition for the pages within. **Care is warranted when changing out a Data Definition as data may be lost.**

Lastly, the "Full Screen Preview" link is particularly useful when displaying non-HTML output.

[1] *Audits also show actions of an individual user within Cascade.*

[2] *Versioning also applies to formats and other Cascade assets; access via the asset-associated dropdown menu.*

Additional Resources

Explore our Community Resources: Cascade CMS Community
Modules, Example Sites & Code: Cascade Exchange
Ideas for Cascade Features: Product Ideas Portal