Facilitating Multi-Site Management with Oracle WebCenter

Karla Schommer, Fishbowl Solutions

Muli-Site Management Overview

Simply put multi-site management is the creation and management of multiple websites within a single toolset. While a multitude of tools exist to enable businesses to manage multiple websites, the distinguished advantage of using a content management system is the ability to reuse both assets and content. *Assets* control the look and feel of the site and include components such as page templates, JavaScript, and CSS files. These building blocks are usually modified by a developer. *Content* refers to user contributed data that varies across a website such as text and images. The multi-site management philosophy encourages the reuse of content and assets whenever possible. However, even if assets cannot be reused directly they can still be copied or utilized in such a way as to avoid recreating them from scratch.

Businesses can use a multi-site management tool to maintain both individual websites and micro sites. Oftentimes organizations have separate, unrelated websites that they create as distinct sites within the same tool. On the other hand, an effective multi-site management tool can also be used with *micro sites*, subsections of a larger web site that are treated like individual websites.

What Multi-Site Management is Not

When defining multi-site management, it is also helpful to distinguish what multi-site management is not. For example, although you are managing multiple websites throughout a single infrastructure, it does not mean that the same template should be reused on every page within a website or even that the same general look-and-feel must be applied to every website. On the contrary, a single website can contain several different templates or layouts and still reuse assets. Additionally, multi-site management tools should not be used to create many uncontrolled and unrelated sites that lack a clear content management strategy. Multi-site management systems are best used with websites that serve specific and dedicated purposes for the business, and where end user contribution is clearly defined and limited. Finally, a good multi-site management approach should not overcomplicate the infrastructure or require constant IT involvement to design, create, and update pages. In fact, when properly deployed, nontechnical users should be able to easily update web content once the website framework has been created by the developers.

Business Benefits

An effective multi-site management tool is essential to your business because by consolidating the administration of numerous websites into a single tool, your business will be able to reduce overall costs in several key areas. One of the upfront savings is in the number of required licenses to purchase. From an IT perspective, the ability to reuse site assets and content results in faster implementations and by extension cost savings. This could mean hiring less developers, or it could mean developers spend more time on other value-add projects. IT resources are not only needed less on the frontend, but are also conserved administration tasks. Since contributors can change content without IT intervention, site administration time is drastically reduced. This reduction in resource needs can be a huge cost savings for the business. A final money saver is training. Since only one tool is required to develop the presentation and content of the site as well as maintain future changes, users only need to be trained on a single technology. This not only reduces training costs, but also decreases ramp-up time.

Common Struggles

While multi-site management is not a new concept, organizations continue to struggle with effectively implementing and maintaining a multi-site infrastructure. Most organizations that implement web sites using multi-site management tools have unrealistic expectations of the product. Everyone wants a silver bullet. Businesses expect that the products are easy to use and will cover a variety of possibilities out of the box with little to no customizations. These beliefs are usually based on perceptions formed during the sales cycle. Unlike what organizations would like to believe, even with the best of multi-site management tools, proper web site implementations require planning, development time, training, and cohesiveness between all parties involved.

One of the biggest struggles organizations run into is a lack of planning. Anxious to see an end result, businesses often don’t take the time to thoroughly think through topics such as metadata, taxonomy, and security modeling. Before any development begins, these important questions need to be asked: What data needs to be collected about each piece of the website? Who should have access to the website, and what permissions should each user have? As a result of poor preparation, IT often continues to be the bottleneck and organizations are not able to save time by offloading tasks to the appropriate subject matter experts. Designing a good contribution model up front can help reduce IT involvement down the road.

Another common downfall of companies trying to implement multiple websites is unrealistic timelines. It is not uncommon for implementations to take longer than expected, even when it is possible to reduce time by reusing assets. Business requirements change; resources are not as available as expected; unforeseen problems must be resolved. There will always be unpredictable timeline impacts. Sometimes this gives the impression that the project was not as successful as hoped, but a missed deadline should not be confused with project failure. There are very few situations where having to wait another day, week, or even month for a website to be completed will significantly impact the bottom line. It’s important to set realistic timelines for multi-site implementations. Taking the proper time to plan and execute your website will reap benefits in the long term by creating greater flexibility and reusability of content and assets.

Sometimes organizations make the mistake of thinking that just because their websites don’t have a similar look-and-feel that they can’t employ reusability practices. While it is true that the easiest and most obvious way to reuse assets is when websites have a comparable look or layout, this doesn’t mean that reusability is out when dealing with dissimilar website layouts. The concept behind separating the content of a website from the presentation means that you can easily change the look-and-feel. The Tips and Tricks section gives some useful suggestions for ways to implement this. It is important to keep in mind that multi-site management is as much about process management as it is about code management. This means that even if you are limited in the ways you can reuse site assets, you will still be able to make use of the planning previously done to implement metadata, taxonomy, security, and workflow models. Those time savings alone will be a considerable cost saver to your organization in subsequent website implementations.

A final obstacle organizations must overcome in order to successfully implement their websites is to coordinate the efforts of their developers. Development teams are often siloed, meaning that each person has their individual role and does not work outside of their function. While each person is an expert at what they do, this structure adds challenges because it’s difficult to get everyone on the team in sync. Additionally, developers are often quick to re-implement new code instead of reusing the assets that already exist. This is often because developers don’t know that the code had already been created or because the site was not designed in a way that facilitates reuse. While reuse isn’t always possible, it should at least be taken into consideration for both current and future projects. Good planning can help prevent developers from creating duplicate code by failing to reuse assets that have already been created.

Choosing WebCenter Content

With the plethora of multi-site management tools available today, there are many strategic advantages of choosing Oracle WebCenter Content and the Site Studio web content management component. The underlying benefit is that the system’s backend is the Oracle Content Server. This content management tool is a mature, enterprise level system. The benefit of having a tool that has been around for years cannot be emphasized enough. Oracle WebCenter Content has gone through many phases of testing and has proven to be stable and scalable. Additionally, as an enterprise level system, the same backend system can be used across many diverse areas of your business beyond web content management.

By choosing WebCenter Content and Site Studio, your business is able to take advantage of several key features of WebCenter Content including workflows, security, and content history. WebCenter Content includes an easy-to-use workflow engine that is available out-of-the-box. Organizations can quickly and easily implement a simple workflow model. Another feature of WebCenter Content is security. Both security groups and accounts are available within WebCenter Content and can be configured to allow for a flexible security model. Accounts allow administrators to assign access differently based on criteria such as department, or geography. Security Groups are groups of content items typically ordered with classifications such as Public, Secure, Guest, etc based on a user’s role. A final advantage of using a multi-site management tool based on a content management system such as WebCenter Content is the content history functionality. All revisions of each item are saved, which means that changes can be recovered or reverted at any point in time – nothing is lost and should a change break a page, the previous working version can be restored.

One of the biggest advantages of Site Studio is the ease of separating the presentation layer from the content layer. Site Studio was designed so that templates contain all of the code related to the display of the website while the content itself is stored in contributor data files. This functionality is built into the tool, eliminating the IT bottleneck with little extra effort. This also leads to quicker implementations.

It is relatively easy to set up a simple functioning website. For example, in one hour you can create a one page website, with all the assets stored in WebCenter Content, in a workable and testable environment. This is not just a demo, but an actual functioning website. Additionally, Site Studio out of the box comes with sample fragments that can be used as a starting point for code snippets. While the fragments themselves should not be used as is, they do provide examples of how to develop basic items such as navigation, searches, and dynamic lists. These samples can be extremely helpful when starting to develop a website. The ease of website creation is just one of the many reasons why Oracle WebCenter Content is an excellent choice for a multi-site management tool.

Multi-Site Management with WebCenter Content

This paper has discussed in length the importance of asset separation, that is splitting up a web page into pieces so that they can be best reused throughout the website and future websites. This section discusses ways to do that and considerations that should be made. Making the correct decisions about how to divide web assets is one of the most important tasks to aid the reuse of assets.

Division of Assets and Content

The first division that needs to be made is between the web assets and the web content. Web assets are generally the portion of the website that will be controlled by developers. Assets control both the look and feel of the website as well as the contribution model. Certain assets like templates, CSS, and JavaScript files determine how the site looks and behaves, while element, region, and placeholder definitions determine what data the contributor is able to edit and what actions he or she will be able to perform while editing.

When dividing your page into different assets, start by thinking about which pieces will be part of the frontend and which will be part of the backend. The header, footer, and navigation are pieces that will be a part of nearly every website. These should each be put in a separate web asset, possibly more if it makes sense. The header will contain the code for items such as logos, images, and searches. Depending on the website, the code snippets that are part of the html head tags may be part of this asset, or may be included in the page template. A common option is to create a separate Subtemplate containing only these snippets. The footer asset typically contains any links for images that are standard across most pages in the website. The navigation code is often stored in a Subtemplate or two depending on if the site has horizontal or vertical navigation or both.

The backend of the site will consist of three main types of assets: Element, region, and placeholder definitions all determine how the contributor will interact with the data on the page. When creating these assets, think about what pieces of the page the contributor should be able to change and what actions the contributor will need. How does this compare with other pages in the site as well as pages from other websites within the infrastructure?

The second type of assets is templates. Region, Sub, and Page Templates all are used to divide up the page into different pieces of code. Page templates should be the most general and Region and Subtemplates more specific. Avoid putting too much code in a page template. These should simply be the skeleton of a page with placeholders for other templates. In general, the number and type of page templates in your website should be based on the variety of different page layouts. For example, you may have one two-column Page Template and another three-column Page Template. This practice will lend to greater reusability. Region Templates and Subtemplates are more likely to pertain to a specific website, but by splitting these into smaller sections, you will find it easier to construct a website from these building blocks.

Fragments are the third type of backend web asset. While fragments are less common and can often be completely avoided in the 10gR4 architecture, they do make sense in certain situations. The nice thing about fragments is their ability to store assets, such as images and CSS files. By storing both images and CSS as fragment assets, the images can be referenced with relative paths. As with templates, be cautious about putting too much code into a fragment.

In addition to web assets, the website will also include web content. Web content is the data in the website that is managed by contributors. The main text on the page is usually web content, but other items such as images, navigation, and dynamically generated content could be controlled by contributors. Thus, these types of files usually contain the bulk of the website content. Web content items can be stored as either XML data files or native documents (Microsoft Word, PDF, etc). Depending on their usage in the website, images can also be used as web content items.

Reusing Assets and Content

The effort put into dividing a website into different assets and content becomes worthwhile when separate items are easier to reuse throughout an environment. Although theoretically any type of web asset or content can be reused, certain types lend themselves more easily to reuse. Web content can easily be reused since both Data Files and Native Documents can be ported to another website with little effort.

Web assets can be reused both in the same website and in different websites. Reusing web assets throughout the same website is quite common and the easiest to visualize. For example, the same Page Template will often be reused throughout the website for a consistent look-and-feel. Another example is the use of a Data File on multiple pages throughout the website. This is especially useful for right-column lists or non-dynamic navigations. When reusing Data Files, it is important to be aware of search engine optimization rules and to be careful to avoid duplicate content throughout your websites. Web content can also be reused throughout different websites. For example, a content file could be used in multiple websites. The Privacy and Terms pages are often the same across multiple websites, so the text can be reused. By reusing the Privacy and Terms Data File on multiple sites, an organization would only need to update one item when they want to make a change. That update would be reflected across all sites sharing that Data File. An example of asset reuse in different websites is page templates. As long as the pages have a similar layout, the same page template can be reused by dynamically swapping the CSS file for a varied look-and-feel.

Additional Considerations

Several other items must be taken into consideration when deciding the best way to reuse assets and content between websites. First, does sharing content between websites make sense? For example, an update to a shared Data File will result in content changes on multiple pages and/or websites. Is this a desired behavior? If not, the Data File should not be shared. Secondly, is the security model appropriate for shared content? Should users from different websites be able to edit the same content? Finally, is the approval process appropriate for shared content? Is the workflow the same for different websites? These items are important to consider because even though the content item to be reused will be used in two different websites, it will still be treated as one document in regards to revisions, security, and workflow. If these applications are not the same, it may be advisable to copy the item so there are two separate content items.

Tips and Tricks

This section details some of the best practices that Fishbowl recommends based on our experiences implementing WebCenter Content websites with SiteStudio. Following these tips will make reuse within your environment easier and clearer.

1. **Make separations logical**

When splitting up a page into different assets, be aware of the frame of the page and make separations that make sense. The most common page layouts are the C-frame (header, footer, left column, and main content area) and the I-frame (header, footer, and main content area).

1. **Create reusable building blocks**

Pieces of code that occur repeatedly throughout the environment usually should be separated into assets so that they can be easily reused. Breaking these down into appropriate building blocks will easily lend to reusability. Common building blocks include the header (containing the logo, search, etc), navigation, and footer. Breaking these down into chunks of code that is always together allows you to easily arrange the particular blocks of code that are used on any given page.

1. **Follow standard HTML best practices**

One of the most effective ways to lend reusability to your websites is to follow HTML best practices while keeping the goal of reuse in mind. For example, style the navigation in your website with CSS so that you can reuse the navigation logic and still easily change the styles. Another common need is the use of meta tags. These tags can be used, but the data should be dynamically supplied (from document metadata, custom section properties, etc) so that the templates can be reused.

1. **Avoid hard coding**

One of the biggest roadblocks organizations run into when trying to reuse site assets is that their developers have hard coded pieces of the page. This is especially common in navigation code. Although it requires a more sophisticated logic, utilize a dynamic navigation built from the site hierarchy if at all possible. Another trick to avoid hard coding is to make certain pieces of the header and footer contributable when appropriate. A final suggestion is to use custom section properties to provide dynamic content. Common uses for custom section properties include title and meta tags. Since custom section properties can be edited in site studio manager, the developer no longer has the responsibility of updating these files.

1. **Avoid duplicates**

Duplicate content is a frequent problem, especially in environments with many developers. Images are a common example. Multiple copies of the same image get checked in when users either aren’t aware that it exists or aren’t able to easily find it. Good planning promotes consistency and helps to avoid duplicate content. Custom components, such as Fishbowl’s Duplicate File Checker, can also assist companies with this problem.

1. **Naming conventions**

One of the keys to good planning is naming conventions. This alone will help greatly with reusability. If others do not know what a content item contains, reuse is extremely difficult and duplicates become common. It is important to decide on a naming convention and be consistent with it throughout your entire environment. Make the naming scheme known throughout your organization and include it as part of the training for both developers and contributors so that everyone is on the same page. Fishbowl suggests the following naming convention for Site Studio assets: <site prefix>\_<description>\_<asset suffix>. For example, FB\_Home\_PT may be used as the content id for the page template assigned to the home page in a Fishbowl website. While specific naming conventions are not always used for content files, keep secondary pages in mind if friendly URLs are desired.

1. **Element Definitions**

Start by creating a basic set of reusable elements. A simple text, image, and WYSIWYG element definition will be used in nearly every website. Copy and create new versions of these reusable assets to implement site-specific features. For example, it may be desirable to have a site-specific WYSIWYG element definition so that the image and link settings can be customized. Having a basic set of reusable elements simplifies the task of setting up new element definitions for new websites.

1. **Secondary pages**

Secondary pages are an important feature of Site Studio that should not be disregarded. By swapping content files into the same template, secondary pages allow the same look-and-feel to be used for an unlimited number of pages. This greatly reduces the number of total website sections that must be created and maintained and allows you to easily reuse the same templates across many pages.

These principles often are difficult for businesses to put into practice. When followed, they can greatly enhance the website implementation process and empower businesses to easily facilitate multi-site management within their infrastructure.

Conclusion

Although multi-site management is not a new concept, organizations continue to struggle with effectively implementing and maintaining a multi-site infrastructure. Multi-site management allows businesses to administer multiple websites in a single tool, saving them time and money. Despite the simplification created by using a single tool such as Oracle WebCenter Content with SiteStudio, organizations have problems implementing websites that make use of the reusability benefits. Reasons for this vary from lack of planning and unrealistic timelines to siloed development teams. It is important to carefully consider factors including security and workflow before determining the best way to separate a site into reusable assets and content. While every website is different and must be considered individually, following clearly defined best practices within your business will ultimately lead to effective reuse within websites and ease your multi-site management headaches.