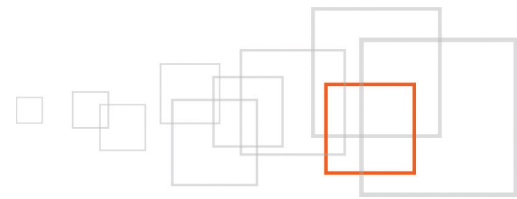


Lots of Websites, One eZ Publish Installation

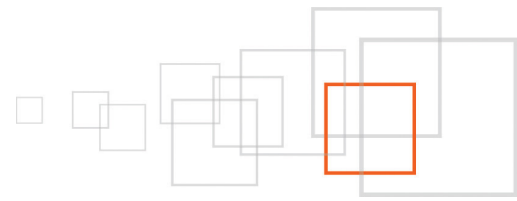
Adding Siteaccesses in eZ Publish

By Greg McAvoy-Jensen
Granite Horizon - <http://granitehorizon.com/>



Index

| | |
|--|----|
| 1 Goal description..... | 3 |
| 2 Introduction..... | 3 |
| 3 Prerequisites and target population..... | 3 |
| 3.1 Target population..... | 3 |
| 3.2 Prerequisites..... | 3 |
| 4 Use Cases..... | 3 |
| 5 What Is a Siteaccess, Anyway?..... | 4 |
| 6 How-To Part I: Preparation of content structure..... | 5 |
| 7 How-To Part II: Settings..... | 7 |
| 8 How-To Part III: Updating Templates and Content..... | 8 |
| 8.1 Top menu adjustment..... | 8 |
| 8.2 Access control..... | 10 |
| 8.3 Populating with content..... | 10 |
| 8.4 Using the search feature..... | 11 |
| 9 Additional Considerations..... | 12 |
| 9.1 Site Map and Tag Cloud..... | 12 |
| 9.2 Site Settings..... | 13 |
| 9.3 Quality Assurance..... | 13 |
| 9.4 Design Extension Override Rules..... | 13 |
| 9.5 Shared Content Classes..... | 14 |
| 9.6 Content Sharing..... | 14 |
| 9.7 Security..... | 14 |
| 9.8 User and access control management..... | 14 |
| 10 How-To Part III: Making Your Site Multilingual..... | 15 |
| 11 Conclusion..... | 17 |
| 12 Resources..... | 17 |
| 13 About the author : Greg McAvoy-Jensen..... | 17 |



1 Goal description

This tutorial gives developers the power to extend the features and flexibility of eZ Publish beyond a single website, to having an unlimited number in one installation. At the end of this tutorial you will have learned how to best leverage eZ Publish's siteaccess concept to create multiple websites. It will let you apply this concept through the concrete case of building two different, multilingual websites on one single eZ Publish instance.

2 Introduction

Users of eZ Publish sometimes wonder if it is possible to make use of a single installation of eZ Publish for multiple websites. The answer is yes—through the power of the “siteaccess”. In fact, an understanding of siteaccesses gives one the ability to do much more than just host multiple sites in one installation. With this technique you can:

- make it easier to share content between databases
- copy and add locations to objects
- use a single Premium agreement and one eZ Network extension for multiple sites with the same or different content
- simplify eZ Publish upgrades by consolidating several sites on one installation

3 Prerequisites and target population

3.1 Target population

This article written for intermediate-level eZ Publish developers, who already know how to install eZ Publish, create a design, program in the template language, and make use of the settings cascade. These skills can be gained by taking an eZ Publish Developer Basics training course, reading eZ Publish Basics or other eZ Publish books, or learning from a solid eZ Publish developer. As well, the community is a vast source of knowledge and exchange. It can be found here : <http://share.ez.no>.

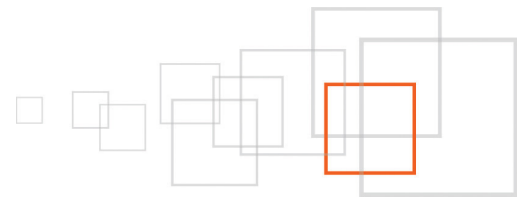
3.2 Prerequisites

The tutorial example in this article is based on this installation:

- eZ Publish 4.3
- ezwebin extension version 1.6
- out-of-the-box installation (your customizations may require additional attention)

4 Use Cases

Let's discuss a variety of reasons one might want several siteaccesses, explore what one is, and then see how



it's done. Here are some typical scenarios for using different siteaccesses in addition to the admin and public siteaccess:

| Case | Siteaccesses | Content-Sharing Capabilities | Details |
|--|--|---|---|
| typical (e.g. a single web shop, social networking site, or corporate site) | site_public site_admin | n/a (only one real website) | <i>both share the same database and var directory but use different designs and extensions</i> |
| multilingual (e.g. a corporate website with three translations plus an administrative interface) | nor slk chi admin | all sites can access the same content provided there's an available translation | <i>same database and var directory and same design (except admin), but different language settings in each</i> |
| several completely different websites (e.g. a company which sells websites with similar features but different design and content to companies in an industry) | site_1_public site_1_admin site_2_public site_2_admin ... | Content (and user accounts) are kept separate so there is no danger of one customer accessing another customer's data. | <i>different databases, var directories, and designs</i> |
| one site which gets customized for different populations (for example, a magazine publisher with different versions for various cities) | city_1_public city_1_admin city_2_public city_2_admin ... | Content is kept separate for each edition. | <i>design may be shared but databases and var directories may be distinct</i> |
| Several sites revolving around a brand/company/idea cross-sharing part of their content, functionalities, and design with each other. | brand_site_1_public brand_site_1_admin brand_site_2_public brand_site_2_admin | Content is partly kept separate for common editors: one subtree per site, but can be shared easily for enabled editors. | <i>Design may be shared, extensions too, database and var directories as well. One content subtree is one site.</i> |

Feel free to add your own creative uses of siteaccesses in this forum : <http://share.ez.no/forums/discussions/lots-of-websites-one-ez-publish-installation-adding-siteaccesses-in-ez-publish>.

5 What Is a Siteaccess, Anyway?

Each siteaccess isn't exactly a different website, though it can certainly create that effect. A siteaccess has two main elements:

- a specific set of configuration files used for it alone, and
- a way of using it, by specifying it in the URL.

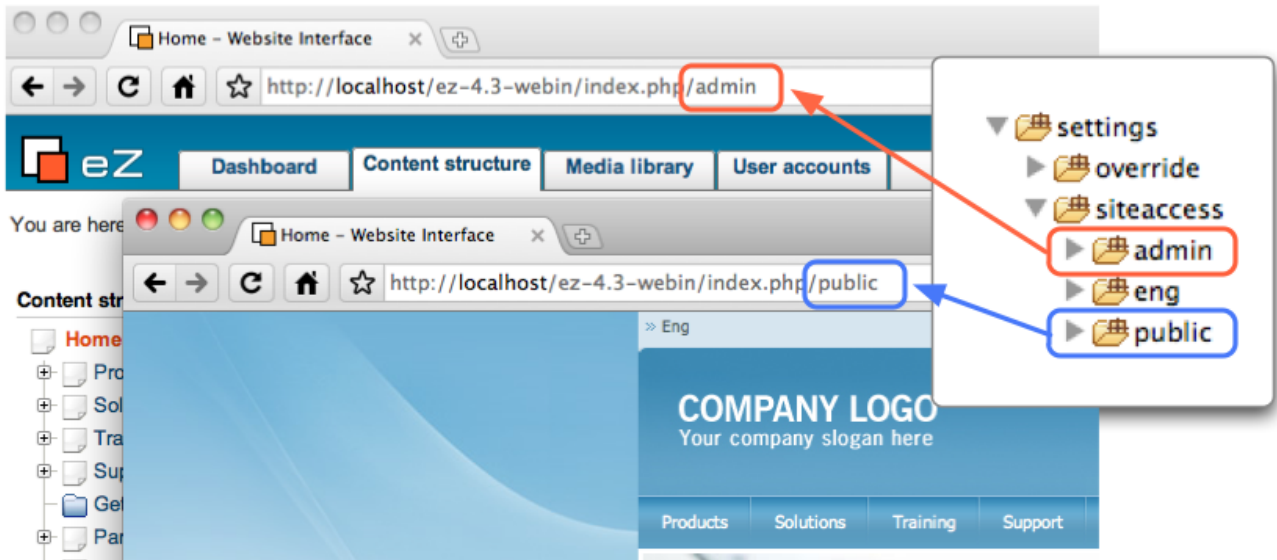
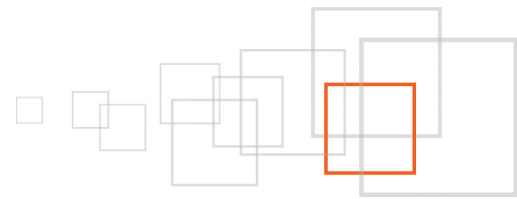


Illustration 1: The concept of siteaccess

Let's say you already have an eZ Publish installation. You're used to addressing the homepage as `http://www.example.com/index.php/eng`. The "eng" is your siteaccess. Perhaps you've rid yourself of the "eng" in your url by making it your default. Or instead of URI-based access, you've switched to host-based access, and the "www" specifies your siteaccess. Either way, the siteaccess is specified in your URL.

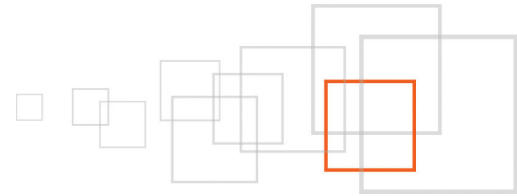
You already have a second siteaccess. When you log in to the admin interface, you use `http://www.example.com/index.php/admin`, or perhaps `http://admin.example.com/index.php`. In both cases, the siteaccess requested is "admin".

At first glance you might have thought the admin interface is really a separate piece of software. But it's simply another siteaccess. In other words, it's just a way of accessing your installation by specifying a different set of configuration files. Remember—in eZ Publish, configuration files let you control everything you want, plus scores of things you had never even thought about. So one siteaccess can produce very different results from another.

For example, you could have different designs used, different language settings, or different extensions with different modules, workflows, datatypes, etc. Moreover, you can even specify that you want a different database used, so that none of the content or even class definitions would be shared.

6 How-To Part I: Preparation of content structure

This is an example of how to change your existing ezwebin eZ Publish installation from one public and one admin siteaccesses (named "public" and "admin") to two public and one admin siteaccesses. Each public site will have its own landing page node. We will specify those node ID's in settings, but first we must create the landing page (home page) for each site in the content node tree. The current "Home" object (of the 'Frontpage' content class) will be reused as one of the two home pages, but we will need to create the second one.



1. Create a new folder called "Websites" (for example). This will be the container to hold the landing pages.
2. Swap it with the top node

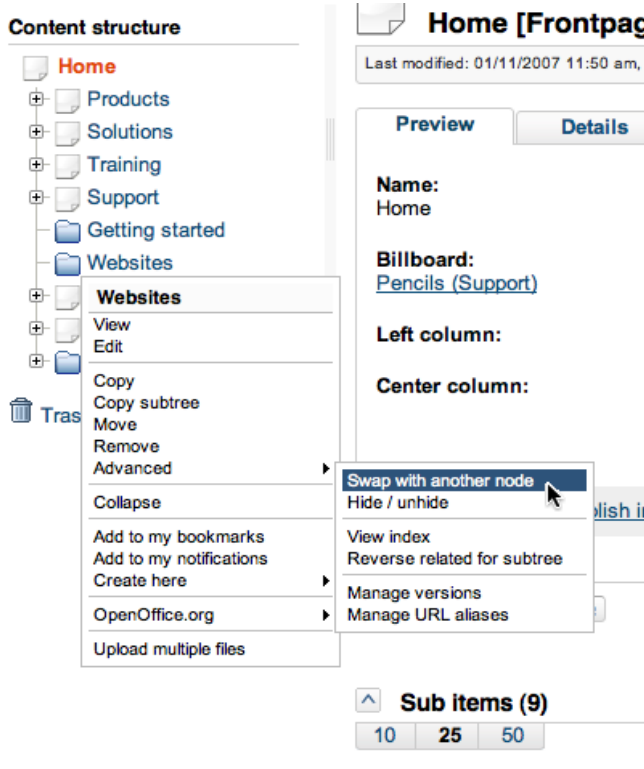
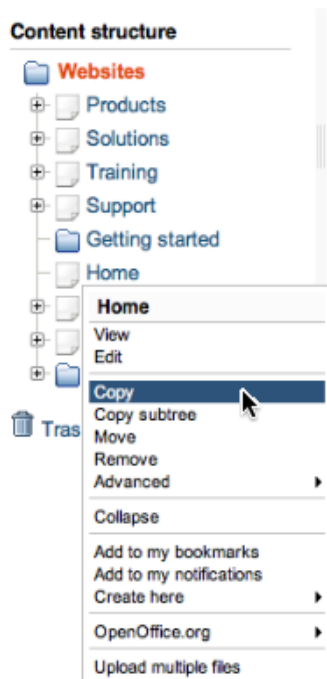


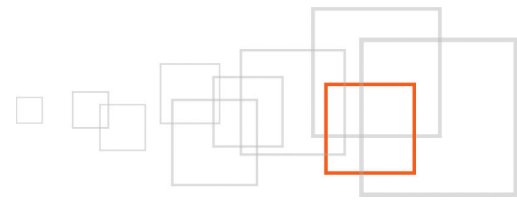
Illustration 2: Swap the new root with the initial top node



3. Clean up the former home page you just swapped (called "Home"), by editing it and removing the various unnecessary elements.
4. Create the second public website's homepage by copying the "Home" object
5. Finally, give the two websites' home pages more meaningful names. For our example we will use:
 - node 175: "Public site one"
 - node 176: "Public site two"

For simplicity of demonstration, we will remove all content than the "Websites" folder and the two home pages created above.

Illustration 3: Copy Home object



7 How-To Part II: Settings

Now we may edit our settings. Node two, previously the landing page node for the one public siteaccess, has become the parent (container) for the two new landing page nodes (see how-to part I). Let's say existing siteaccess 'public' happens to have as its landing page node 175, and siteaccess 'new_site' has landing page node is 176.

1. Edit settings/siteaccess/public/content.ini.append.php :

```
[NodeSettings]
RootNode=175
```

2. Edit settings/siteaccess/public/site.ini.append.php :

```
[SiteSettings]
SiteName=Public site one
SiteURL=localhost/ez-4.3-webin/index.php/public
SiteURL=example.com/index.php/public # (or public_site.example.com) ?
IndexPage=/content/view/full/175
DefaultPage=/content/view/full/175
RootNodeDepth=2
# customize the metadata by site if you like :
MetaDataArray[author]=John Doe
MetaDataArray[copyright]=John Doe co.ltd
MetaDataArray[description]=This is my public website number one, out of two on the same
eZ instance
MetaDataArray[keywords]=Public website one, multi-siteaccesses with eZ Publish
```

This is the beginning of leveraging eZ Publish's Search Engine Optimization (SEO) features. Feel free to do more. Extensions which may be of help include Meta Data Datatype, Google Sitemaps, Google News Sitemap, and feZ Meta Data.

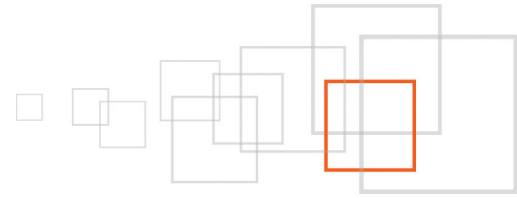
```
[SiteAccessSettings]
PathPrefix=Public-site-one # (name of node 175 if you don't want its name appearing in
the URL)
# Comment out the following array reset line, will ease the addition
# of new siteaccesses in our tutorial.
# RelatedSiteAccessList[]
```

3. Copy settings/siteaccess/public to settings/siteaccess/new_site

4. Repeat steps 1 and 2 for new_site (replacing 175 with 176, and using the proper url)

5. Edit settings/override/site.ini.append.php :

Make sure none of the above settings is in override/site.ini.append.php. If any is, comment it out and make the appropriate addition to settings/siteaccess/admin/site.ini.append.php. Also in the override site.ini.append.php, add :



```
[SiteSettings]
SiteList[]=new_site

[SiteAccessSettings]
AvailableSiteAccessList[]=new_site
RelatedSiteAccessList[]=new_site
PathPrefixExclude[]
PathPrefixExclude[]=Media
PathPrefixExclude[]=Users
```

Note that in a site using host-based access method, we'd add something like:

```
HostMatchMapItems[]=new_site.example.com;new_site
```

8 How-To Part III: Updating Templates and Content

8.1 Top menu adjustment

If you're using Website Interface (ezwebin) 1.6 and have the top menu bar, edit menu/flat_top.tpl so it knows where to find the list of pages below the landing page node:

```
{def $root_node=fetch( 'content', 'node', hash( 'node_id', $indexpage ) )
```

to

```
{def $root_node=fetch( 'content', 'node', hash( 'node_id', ezini( 'NodeSettings',
'RootNode', 'content.ini' ) ) )
```

The change above leads to the following visual change :

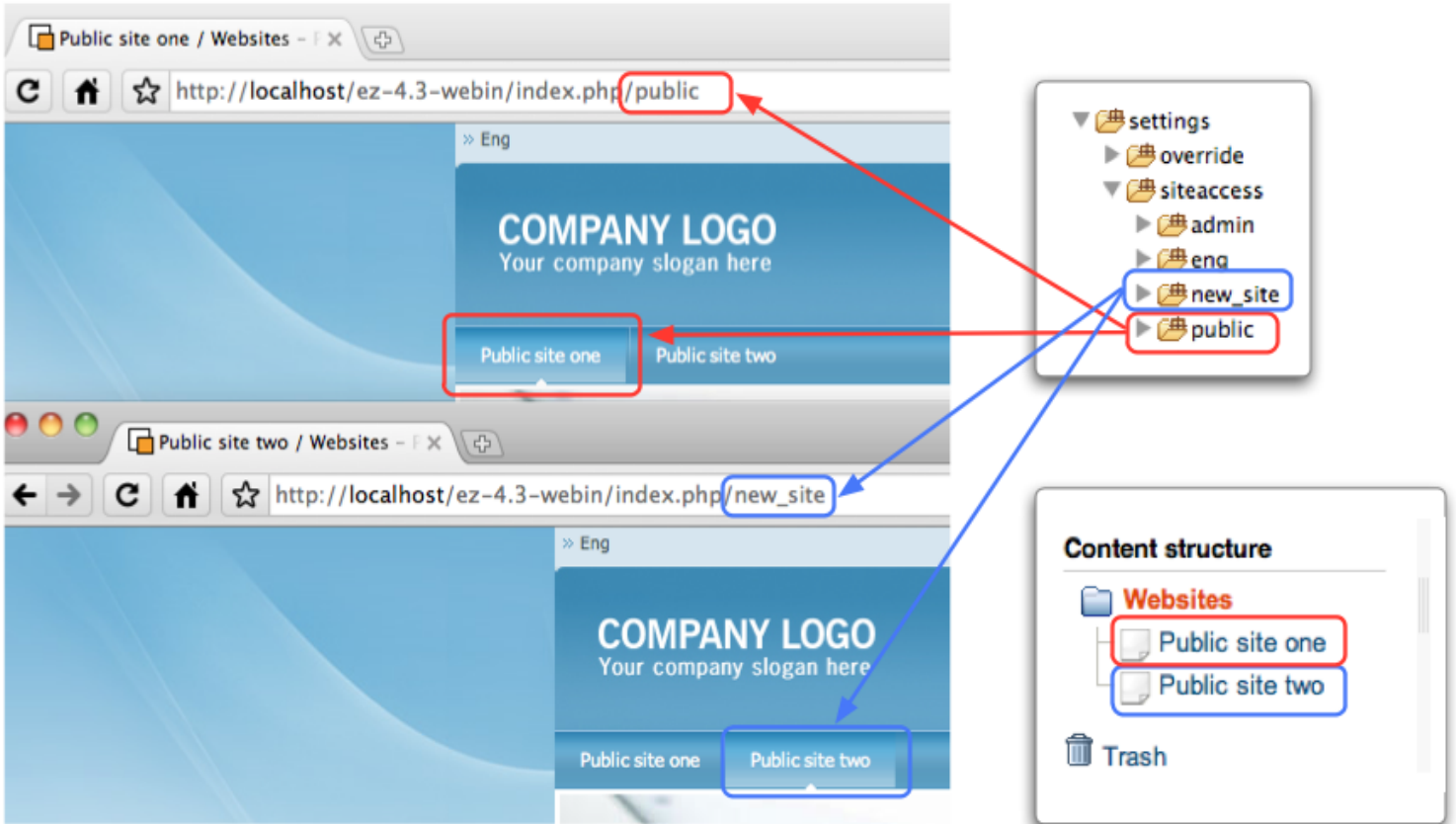
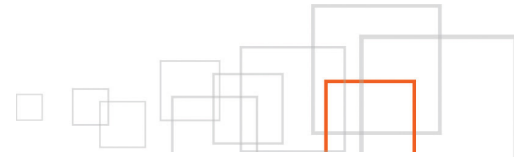


Illustration 4: Before altering the template building the menu

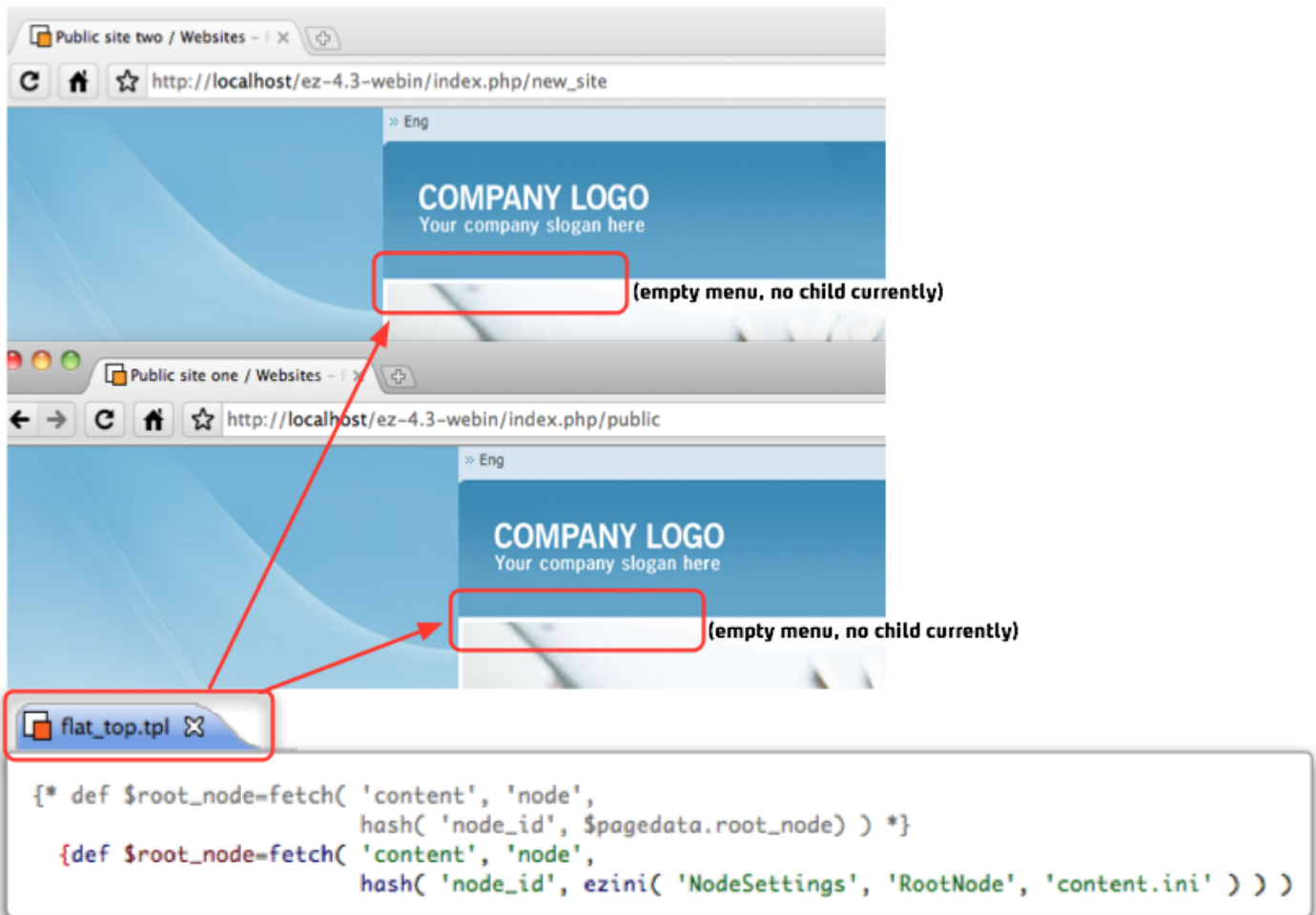
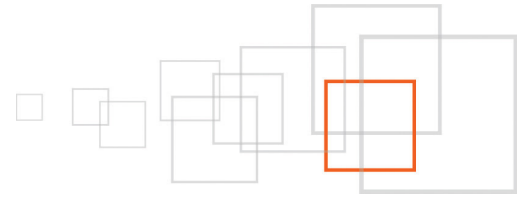


Illustration 5: After having altered flat_top.tpl, the template building the menu



8.2 Access control

Edit permissions for the anonymous user to allow login to the new siteaccess.



Anonymous [Role]

Name:
Anonymous

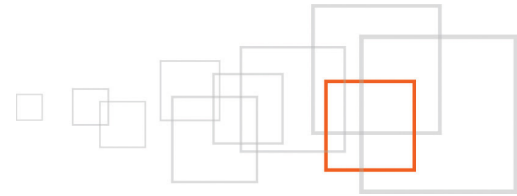
Policies (6)

| Module | Function | Limitation |
|---------|----------|------------------------------------|
| content | read | Section(Standard) |
| content | pdf | Section(Standard) |
| rss | feed | No limitations |
| user | login | SiteAccess(public , new_site) |
| user | login | SiteAccess(eng) |
| content | read | Class(Flash , Image , Quicktime , |

Illustration 6: Update anonymous role

8.3 Populating with content

Now it is possible to add content under each home page. It will then appear only on the correct website.



The screenshot shows the eZ Publish 4.3 interface. At the top, there are navigation tabs for 'Company', 'Community', and 'Products'. Below them is a breadcrumb trail: 'Public site one / Community / Developers blog / Discovering siteaccesses in eZ Publish 4.3'. The main content area displays an article titled 'Discovering siteaccesses in eZ Publish 4.3' by Administrator User, dated 01/04/2010 6:50 pm. The article text includes a greeting 'Aloa,' and a paragraph about a tutorial by Greg MacAvoy Jensen. Below the article is a 'Comments' section with one comment titled 'Timeless feature indeed' by Administrator User, dated Thursday 01 April 2010 6:55:39 pm. A 'New comment' button is visible at the bottom of the comments section. On the right side, a 'Content structure' tree is shown, highlighting the 'Developers blog' node under the 'Community' folder of 'Public site one'.

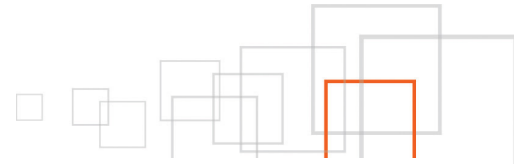
Illustration 7: Added content in one of the websites

8.4 Using the search feature

If you'd like the search function to only search the subtree below each siteaccess's own landing page node, add the following to pagelayout.tpl's search form:

```
<input type="hidden" name="SubTreeArray[]" value="{ $indexpage}" />
```

Do similarly for templates/content/advancedsearch.tpl and templates/content/search.tpl, though you'll want to use something more like "value="{ezini('NodeSettings', 'RootNode', 'content.ini')}" since \$indexpage isn't available there.



```

page_header_searchbox.tpl
1 <div id="searchbox">
2   <form action="{/content/search}|ezurl">
3     <label for="searchtext" class="hide">{'Search text:'|li18n('design/ezwebin/pagelayout')}</label>
4     {if $pagedata.is_edit}
5     <input id="searchtext" name="SearchText" type="text" value="" size="12" disabled="disabled" />
6     <input id="searchbutton" class="button-disabled" type="submit" value="{Search'|li18n('design/ezwebin/pagelayout')}>
7     {else}
8     <input id="searchtext" name="SearchText" type="text" value="" size="12" />
9     <input id="searchbutton" class="button" type="submit" value="{Search'|li18n('design/ezwebin/pagelayout')}>
10    {if eq( $ui_context, 'browse' )}
11    <input name="Mode" type="hidden" value="browse" />
12    {/if}
13  {/if}
14  <input type="hidden" name="SubTreeArray[]" value="{ezini( 'NodeSettings', 'RootNode', 'content.ini' )}" />
15  </form>
16</div>

advancedsearch.tpl
84 {foreach $search_sub_tree as $sub_tree}
85 <input type="hidden" name="SubTreeArray[]" value="{ $sub_tree}" />
86 {/foreach}
87 <input type="hidden" name="SubTreeArray[]" value="{ezini( 'NodeSettings', 'RootNode', 'content.ini' )}" />

search.tpl
29 <p>
30   <input class="halfbox" type="text" size="20" name="SearchText" id="Search" value="{ $search_text|wash}" />
31   <input class="button" name="SearchButton" type="submit" value="{Search'|li18n('design/ezwebin/content/search')}>
32   <input type="hidden" name="SubTreeArray[]" value="{ezini( 'NodeSettings', 'RootNode', 'content.ini' )}" />
33 </p>

```

Illustration 8: Update search template code

Clear caches for changes to take effect if you have not turned your eZ Publish instance into development mode.

9 Additional Considerations

9.1 Site Map and Tag Cloud

You can also then have each siteaccess produce a sitemap of only its own site. Edit the template showing the series of links at the top-right corner of the page to make sure the generated links take the new content root into account. It is called `page_header_links.tpl` in the 'ezwebin' design.

The generated sitemap URLs will say something like :

`http://www.example.com/index.php/public/content/view/sitemap/175` (or 176 for the other siteaccess)

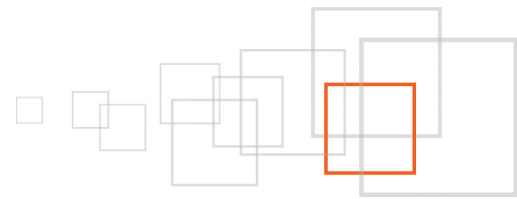
If you have a Google sitemap or Google news sitemap generator, make the appropriate adjustment there as well.

```

page_header_links.tpl
3   {if $pagedesign.data_map.tag_cloud_url.data_textline('')}
4     <li id="tagcloud"><a href="{concat("/content/view/tagcloud/", ezini( 'NodeSettings', 'RootNode', 'content.ini' ) )}|ezurl" title="
5     {/if}
6   {if $pagedesign.data_map.site_map_url.data_textline('')}
7     <li id="sitemap"><a href="{concat("/content/view/sitemap/", ezini( 'NodeSettings', 'RootNode', 'content.ini' ) )}|ezurl" title="
8     {/if}

```

Illustration 9: Update sitemap and tagcloud templates



9.2 Site Settings

If you are making use of the “site settings” feature (which may be accessed on the public site by a link near “registration” and “login”, or in the administration interface's Design/Look and Feel), it will work well for websites which are multiple translations of the same data. This feature is useful for demo's and small sites. However, it will not work by default for different websites with different data (whether using multiple databases or different subtrees of the same content tree). In this case, it is generally best to modify the templates which use this data to draw it from a content nodes you specify for each site separately.

9.3 Quality Assurance

This how-to was certainly not exhaustive. Your own site may have other features which get broken by trying to switch from one root node to two landing page nodes beneath a folder. The example code above should give good hints as to how to fix your custom template code.

9.4 Design Extension Override Rules

Your next step might be to create a custom design for each site by modifying the DesignSettings block in settings/siteaccess/<siteaccess-name>/site.ini.append.php. Note that if you keep your design settings in an extension, you'll have to list that extension in settings/override/site.ini.append.php; it's not enough to simply list it as an ActiveAccessExtension in settings/siteaccess/<siteaccess-name>/site.ini.append.php. The ActiveAccessExtension setting is useful because it exposes only the necessary URL's and API's. It allows each site to use a different design extension. Alternately, siteaccesses could share one common design extension, and each have their own specific design extension cascading over the common design extension.

Putting this all together, if we created a “publicdesign” design in a “publicdesign” extension, and a “newsitedesign” in a “newsitedesign” extension, and also an “ourshareddesign” in an “ourshareddesign” extension, we could see some example ini settings. For the “public” siteaccess this would look like:

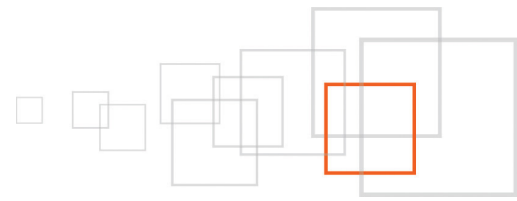
```
[ExtensionSettings]
ActiveAccessExtensions[]=publicdesign
ActiveAccessExtensions[]=ourshareddesign

[DesignSettings]
SiteDesign=publicdesign
AdditionalDesignList[]=ourshareddesign
AdditionalDesignList[]=ezwebin
AdditionalDesignList[]=base
AdditionalDesignList[]=standard
```

On “new_site” we would do this:

```
[ExtensionSettings]
ActiveAccessExtensions[]=newsitedesign
ActiveAccessExtensions[]=ourshareddesign

[DesignSettings]
SiteDesign=newsitedesign
AdditionalDesignList[]=ourshareddesign
AdditionalDesignList[]=ezwebin
AdditionalDesignList[]=base
AdditionalDesignList[]=standard
```



9.5 Shared Content Classes

In this example we've used the same database and var directory for both public sites. Though the content is in separate subtrees, the content classes they use are shared. Be careful not to change attributes in a class to suit one site if they're needed by the other site. If this might become a problem, you could use a separate database and var directory—or more simply, each site could use different content class groups.

9.6 Content Sharing

Copying an object or subtree is of course a simple way of getting content into a second subtree (website) when the database is shared by two sites, but then changes to one copy will not appear on the other. In the same scenario, you can easily share content by using secondary/additional locations. That's an easy way of sharing content, but it must be done for each individual object (the children of the shared object do not automatically get shared as well). A workflow (“after publishing”) could be used to publish all articles in one subtree in another subtree (on the second site) as well. Another way to pull content from one database to another would be using an RSS feed.

9.7 Security

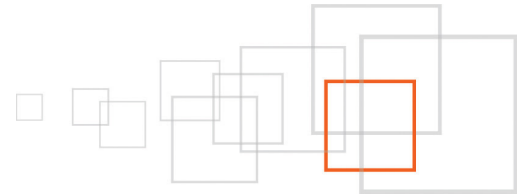
It is more secure to use separate databases and var directories for separate sites. It may be best to always use separate databases and var directories unless the siteaccesses need access to some of the same content or users in an easy way. If you do have a database and var directory for each public siteaccess, each public siteaccess will need a corresponding admin siteaccess.

9.8 User and access control management

A Single Sign On (SSO) capability is sometimes required, letting a visitor log in with one single set of access credentials on a series of websites. This can be easily achieved when the content is shared between siteaccesses (one subtree per site), for the users (stored as other content objects) are natively shared. When the content base is not shared, dedicated SSO solutions, such as LDAP or Active Directory, must be used. eZ Publish natively supports LDAP as an SSO system, and its architecture can make use of any SSO-plugin. The latter take the form of an extension, and can interface eZ Publish with basically any SSO system. More on SSO extensions here : <http://share.ez.no/articles/ez-publish/using-a-sso-in-ez-publish> .

Concerning access control, user roles can be applied with subtree limitations, and we now know each website can be a subtree. In our example, care would need to be given to ensure the node 2 is not tampered with, and that the name of the landing page nodes is not changed without an appropriate configuration file change. This can be assured by changing user roles and policies in the administration interface. Having roles applied with subtree limitations eases access control administration. This generic benefit makes a whole lot of sense in our example : the few user profiles are identified, across the two public sites we built, then materialized as roles. Typically, a logged-in user on both site 1 and site 2 can read content from the standard section, the media library, and create & edit blog posts. However, they should not be able to read or edit content from the other website (ie: from the other subtree). To satisfy this :

- One single role is crafted, containing the handful of policies described above
- Two user groups are created : “Editors site one” and “Editors site two”, self explanatory.
- The role is applied to both user groups, with subtree limitation on their respective website's subtree.



10 How-To Part III: Making Your Site Multilingual

Imagine that your new, award-winning website (siteaccess "new_site") has been well so well received that your boss now wants to have it available in Chinese as well. We simply add another siteaccess ("chinese_new_site") which will use the same data, but with different language settings.

1. Copy the 'new_site' directory under settings/siteaccess, and rename it to 'chinese_new_site'.
2. Edit settings/siteaccess/chinese_new_site/site.ini.append.php to show only Chinese:

```
[RegionalSettings]
TextTranslation=enabled
Locale=chi-CN
ContentObjectLocale=chi-CN
SiteLanguageList[]=chi-CN
ShowUntranslatedObjects=disabled
```

3. Alter access-related settings :

```
[SiteSettings]
SiteName=Public site two in CHINESE
SiteURL=localhost/ez-4.3-webin/index.php/chinese_new_site
SiteURL=example.com/index.php/ chinese_new_site # (or chinese_new_site.example.com) ?
IndexPage=/content/view/full/176
DefaultPage=/content/view/full/176
RootNodeDepth=2
MetaDataArray[author]=John Doe
MetaDataArray[copyright]=John Doe co.ltd
MetaDataArray[description]=This is my public website number two in CHINESE, out of two
on the same eZ instance
MetaDataArray[keywords]=Public website two in CHINESE, multi-siteaccesses with eZ
Publish
```

4. Edit settings/override/site.ini.append.php, adding

```
[SiteSettings]
SiteList[]=chinese_new_site

[SiteAccessSettings]
AvailableSiteAccessList[]=chinese_new_site
RelatedSiteAccessList[]=chinese_new_site
HostMatchMapItems[]=chinese.example.com;chinese_new_site
```

5. Add chinese as a language for editors : navigate to Setup > Languages from the administration interface.

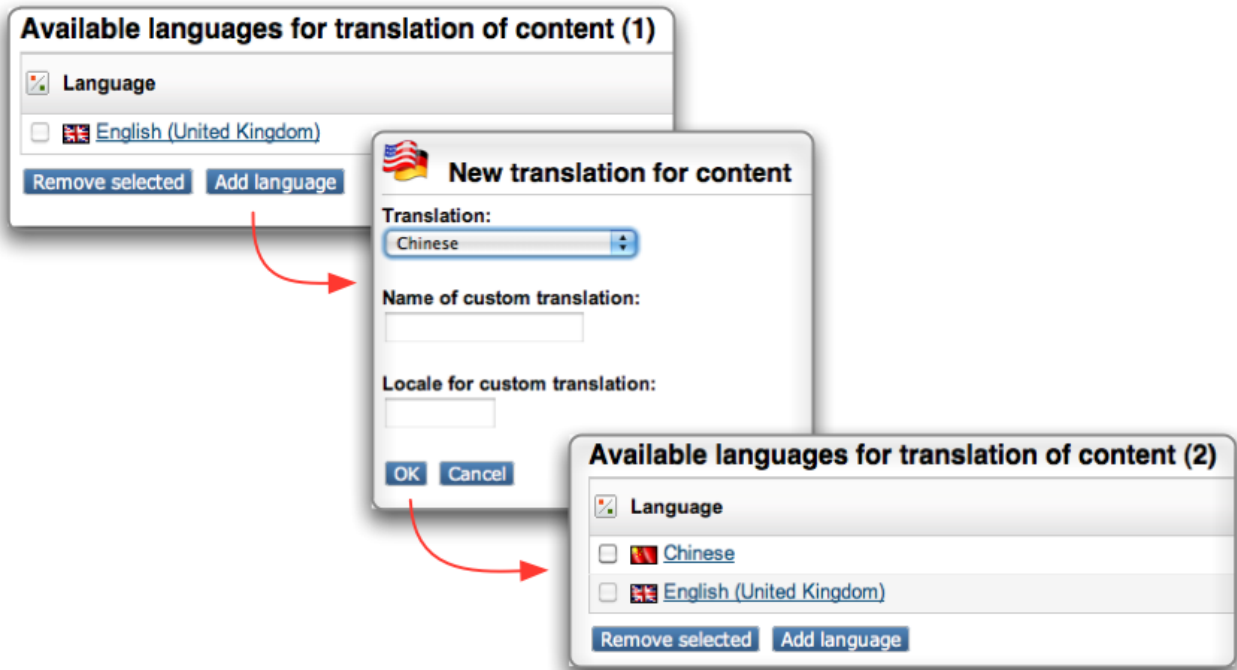
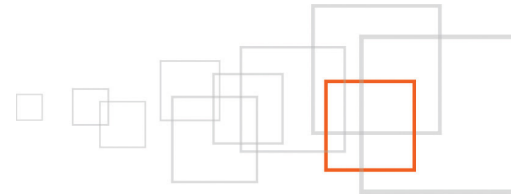


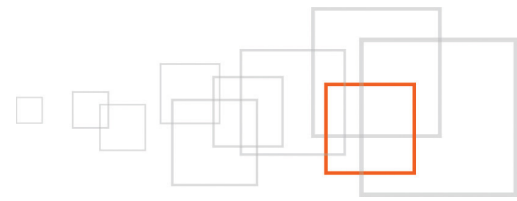
Illustration 10: Adding the Chinese language

6. Allow anonymous login to this new siteaccess (make a ref to same procedure earlier above)
7. Clear the caches

Here is the final result after having added some content :



Illustration 11: Content in Chinese was added



11 Conclusion

eZ Publish is already an incredibly configurable framework / content management platform. Siteaccesses allow one to offer multiple sets of configurations in one installation, using the same or different data. This allows one site to be offered in different languages, or multiple sites to be run in one installation, or different looks to be offered with the same data. Adding an additional siteaccess requires careful planning; the basic steps for an example site were provided here.

12 Resources

Here are some links to other reading you might find helpful.

- [siteaccesses](#)
- [siteaccess configuration settings \(see also the comments in settings/site.ini\)](#)
- [ezwebin top menu bar](#)
- [multilingual sites](#)
- [sitemaps](#)
- [forum thread](#)
- [forum thread](#)
- [See also the comments in settings/site.ini.](#)

13 About the author : Greg McAvoy-Jensen



Greg is the executive director of Granite Horizon, one of the largest eZ Publish development firms in the United States of America. A certified eZ Publish developer, he has served clients and trained developers on four continents since he started working with eZ Publish in 2003. His wife Heather and he are proud parents of three wonderful children.

[@granitegreg](#)