

VITAL - Configure a new Vital instance

Purpose

This document provides instructions on how to configure a new instance of the ARROW VITAL software.

Audience

RUBRIC Tech Team and Partner System Administrators

Requirements

Access to the server on which the Vital instance will live

Server already installed with ARROW VTLS Vital

References

None

Conventions

ARROW VTLS Vital is referred to simply as 'Vital'

[vital-dir] is the directory in which you have Vital installed (eg /usr/vtls/vital/)

1 Setup

1.1 Place entire access (Portal) directory under subversion control

1. Place the entire access directory under subversion control by creating a directory in subversion:

```
svn mkdir https://subversion_server_hostname.domain/path/to/dir/access
```
2. Check out the just-created access directory 'over' the access directory locally (if it is empty on the subversion server, it will check out an “empty” access directory locally, not touching any files locally):

```
svn co https://subversion_server_hostname.domain/path/to/dir/access
```
3. All the local files / directories will then need to be added to the local “working copy” that was just checked out:

```
svn add filename.extension
```

or

```
svn add directory/
```
4. Sync this all up to the subversion server:

```
svn ci -m “Adding original subversion files”
```

1.2 Create a new skin for your specific instance

5. In the [vital-dir]/access/assets directory, copy the default Vital skin (Template):

```
cp -r Template/ [your_skin_name]
```

6. Change the configuration so that it looks for the new skin:

Edit [vital-dir]/access/configuration/configuration.xml by changing the <DefaultSkin> tag to reflect the name of the skin you created above.

1.3 Place new skin into subversion

7. Add the contents of the new skin to the working copy of subversion:

```
svn add your_skin_name
```

8. Sync the contents of your new directory up to the server:

```
svn ci -m ""
```

2 Configure Vital instance

2.1 Add custom logos to the Access Portal

9. Place your logo in the directory [vital-dir]/access/assets/[your_skin_name]/images (create the directory "images")
10. In the directory [vital-dir]/access/assets/[your_skin_name] open the file called skin.css
11. Find the div called "div#navigation"
12. Edit the file so that the name of your logo replaces "master.png" with images/[your_logo_name]
13. Save your changes
14. Check the appearance of your logo in the Portal
15. NOTE: Depending on your logo, you may need to alter the background-color in the "div#header" as well as "div#navigation li" and "div#navigation".
16. NOTE: Depending on your logo, you may need to create an image that can sit behind the navigation buttons. One way of doing this is copying "master.png" (the default VTLS Vital header image) and modifying the copy (using image editing software) so that the lower banner in this image is the colour you wish it to be.

2.2 Create custom header

17. If the header required has an image on both the left and right sides, the following step is a way to do this:
18. In access/assets/[skin_name]/skin.css, remove the reference to an image from div#navigation

19. In `div#header` modify the “background” value to reflect the following:


```
background: url("directory/image.gif") no-repeat right bottom;
```
20. In the same div, make sure the “background-color” value reflects the colour that you need the header to be.
21. In `index.php`, find the following code:


```
<div id="header"></div>
```
22. Insert the following code (customised to your requirements) inside the div above the call to “`top_navigation`”, to place the left side header image:


```
<ul>
  <li><a
href="#"></a></img></li>
</ul>
```
23. Check the browser to see that the images have placed correctly (<http://hostname:8888/access>)
24. The code above will need to be added to all files that call `<div id="header">`:
 - about.php
 - adv_search.php
 - browse.php
 - contact.php
 - contribute.php
 - copyright.php
 - detail.php
 - disclaimer.php
 - help.php
 - highlights.php
 - index.php
 - journals.php
 - navigator.php
 - privacy.php
 - results.php
 - view.php

2.3 Create custom footer

25. If the footer requires more than what Vital already provides, the following steps may help:
26. In `access/assets/[skin_name]/skin.css`, above “`div#footer`”, create a new set of divs that follow the same format as `div#footer`, for example as follows:

```
div#footer_mq_info
{
  font-weight: bold;
  float: right;
  padding: 3px 10px 0 0;
  background-color: #fff;
}
```

```
div#footer_mq_info ul
{
  margin: 16px;
  margin-top: 8px;
  margin-bottom: 4px;
  padding: 0px;
  width: auto;
}
```

```
div#footer_mq_info li
{
  display: block;
  color: #686868;
  margin: 0;
  padding-left: 3px;
  padding-right: 3px;
}
```

```
div#footer_mq_info li.first
{
  border-left: none;
  padding-left: 0;
}
```

27. Note that each of the values above may or may not be relevant to the type of custom footer you are trying to build.
28. In the appropriate locale file (eg localization/locale-en-us.xml), in the Footer Navigation section, add the links that you need for your footer, for example:

```
<Text_1_name>[text_1_value]</Text_1_name>
<Text_2_name>[text_2_name]</Text_2_name>
```

```
<Link_1_name>[link_1_name]</Link_1_name>
```

29. In common.php, near the code for function bottom_navigation, create a new function as follows (example only) (note the difference between a non-linking name and a link):

```
function bottom_mq_info() {
    ?>
    <ul id="footer_mq_info">
        <li class="first"><?=get("Text_1_name");?></li>
        <li><?=get("Text_2_name");?></li>
        <li><a href="[link_target_1]"><?=get('Link_1_name');?></a></li>
    </ul>
```

30. In index.php, find “<div id="footer">” and straight after the opening tag (before the call to “bottom_navigation”) enter the following code (customised to your specific code):

```
<div id="footer_mq_info">
    <?=bottom_mq_info();?>
</div>
```

31. The above code will need to be placed in each appropriate file (as listed in step 2.2)

2.4 Customise text in the Access Portal

32. All text changes can be done by editing the “locale-en-us.xml” file, located in [vital-dir]/access/localization/

33. To alter the name of the site (that displays at the top of the browser window):

Edit the <Title> tag so that it contains your own text.

34. To alter the text of the Title on the home page:

Edit the <AboutTitle> tag so that it contains your own text.

35. To alter the text on the home page, edit locale-en-us.xml:

Edit the <About> tag so that it contains your own text.

36. To add new fields, find the section of locale-en-us.xml that corresponds and create the new tag, keeping in mind that the text will not appear until you have configured the other appropriate files to call that tag.

2.5 Configure Browse views

37. To make any configuration changes to the Access Portal, edit the “configuration.xml” file located in [vital-dir]/access/configuration

38. To turn off all letters displaying in all browse views (to make only letters that have objects “in” them display):

- a. Edit [vital-dir]/access/configuration/configuration.xml by changing the <ShowAllBrowseLetters> value to false (instead of true).

39. To change the order of the Browse Views menu:

- b. Edit [vital-dir]/access/localization/locale-en-us.xml by changing the order of the values in the <browseBy> tag.
 - c. NOTE: Any indexed field can be added as a Browse option
40. To add a search type for the Advanced Search
- d. Edit [vital-dir]/access/localization/locale-en-us.xml by adding a copy of the lines between the <zones> tags that start with <attribute type="....>. For example, add in a line:


```
<attribute type="keyword" label="Resarch Unit" value="researchunit" />
```
 - e. NOTE that the search type you add MUST BE IN THE INDEXES.
41. To remove a Browse option from the Browse menu:
- f. Edit [vital-dir]/access/localization/locale-en-un.xml by commenting out the line between the <browseBy> tags that starts with <attribute label="...>

2.6 Link to Valet

42. Edit the [vital-dir]/access/localization/locale-en-us.xml file by creating a new tag for <ContribContent> that contains the following (comment the old one out):
- g. <ContribContent>Click to submit a resource using Valet.</ContribContent>

2.7 Change colours of tables

43. Edit the [vital-dir]/access/assets/[skin_name]/skin.css file by changing the following values in the following places. Note that some divs are repeated with different values. Ensure that you are using the appropriate div for that value.
- h. For the following code, change the "border-bottom: solid #xxxxxx 1px" to change the border below the navigation bar:

```
div#navigation
{
  background-image: url(images/master_newc_midblue_lib.png);
  background-repeat: repeat-x;
  background-position: 0px -142px;
  border-bottom: solid #597786 1px;
}
```

- i. Change the colour codes in the following to change the colour in the browse menu:

```
ul#browse li a
{
  background-color: #597786;
  border: solid #597786 1px;
  margin: 1px;
}
```

- j. Change the color code in the following to change the colour in the browse menu:

```
html>body ul#browse li a
{
    width: auto;
    background-color: #597786;
}
```

- k. Change the color code in the following to change the colour of the table headers:

```
thead tr th
{
    background-color: #5A8263;
    border: solid #6c9554 1px;
}
```

- l. Change the color code in the following to change the colour of the browse table:

```
ul#browse li a.selected
{
    border: solid #5A8263 1px;
    background-color: #5A8263;
}
```

2.8 Sync altered content up to subversion

44. From the access directory, sync up the altered content:

```
svn ci -m "Syncing altered content"
```

3 Check out an already-configured access directory

45. If you are wanting to deploy code and configuration from a test instance to a production instance, this can be done by checking out the already-configured code from subversion to your local (production) instance.

46. Move the old access directory to user's home directory:

```
mv access/ /home/username/
```

47. Check out a new copy of the access directory (already configured):

```
svn co https://subversion_server_hostname/path/to/dir/access
```

[you will need to authenticate – firstly as the user on the production server, secondly as the user in subversion]

48. Make any required local modifications:

In configuration/configuration.xml, change all occurrences of the hostname

In localization/locale-en-us.xml, change all occurrences of the hostname
49. Your access directory should now be ready for use.

4 Create base collections

4.1 Create first collection

NOTE: This functionality does not always work in release 2.1, so may not succeed.

50. Open the Vital Manager and login

51. From the Create/Import tab, select “Create New”

52. In the “Description” tab provided, enter the details for the new Collection

53. When complete, click on “Apply” to save the changes.