

MedCommons HomeSite Templating System

Overview

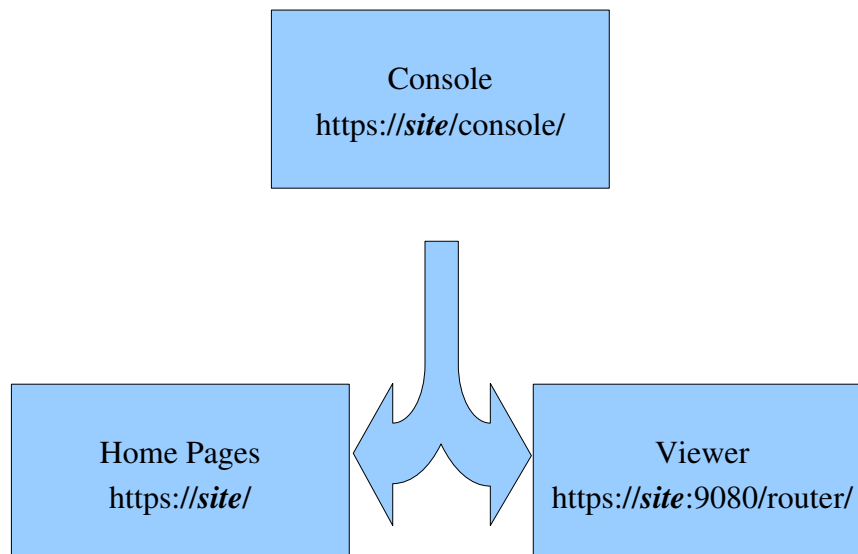
A MedCommons Appliance has three major components. All three components are dynamic web sites.

1. The Home Pages – The home page, sign in and registration, account management;
2. The Viewer – manages patient data; and
3. The Console – solely for administrators of the machine.

Each of these three systems is designed to be customized to a very fine degree by the appliance owner. The MedCommons main web site <http://www.medcommons.net> is in fact a standard appliance customized by the MedCommons webmasters.

Three things to keep in mind when customizing the Appliance:

1. DRY – Don't Repeat Yourself. If you need to make one change to many files, there's almost certainly a single file that needs to be changed, and all other pages will “inherit” or be affected by the change.
2. Different Technologies – MedCommons uses the best technology for the purpose. Some parts are written in PHP, some parts are written as Java Servlets, some parts are written in Python.
3. The Console consolidates it all – despite the different technologies, the Administrator Console has control of the look and feel of the other components.



Right now, the Console edits selected web pages (**/var/www/html**), selected PHP templates in **/var/www/mc_templates**, it edits the PHP settings file **/var/www/php/local_settings.php**, and it edits the Gateway/Router configuration file **/opt/gateway/conf/LocalBootParameters.properties**.

Accessing the Appliance Console

The console lives on the appliance host's main webserver:

<http://site/console>

The console can only be accessed by specially authenticated users.

Adding Media to Your Site

You can add **pages, images and other media** to your site by copying them to the Apache webserver's top-level document root.

```
> scp myfile.html site:/var/www/html
```

This file will now be accessible as `http://site/myfile.html`

And, of course, if you create **subdirectories** and place files there, they'll be accessible as well.

```
> scp -r subdir site:/var/www/html
```

All the pages in `subdir` will be accessible as `http://site/subdir/`

Modifying the Built-in Appliance Home Page

Most of the appliance's web pages have consistent styles, headers, and footers. Some parts of the page, like the logo or your company name, can be changed by the console, as described above in **Changing Owner Variables**.

If you want to *really* customize a set of pages, you can edit the HTML in a single file, and have that HTML published to all the other pages.

Let's look at the files.

```
> ssh site
```

```
...
```

```
$ ls /var/www/console/templates/www
```

base.html	fq.html	password.tpl.php	root.html	ud.html
base.js	ft.html	pd.html	style.css	
dd.html	index.html	receipt.tpl.php	th.html	
forgot.tpl.php	login.tpl.php	register.tpl.php	tu.html	

Let's look at a single file. The simplest file is `ud.html`.

```
$ cd /var/www/console/templates/www
```

```
$ cat ud.html
```

```
{% extends "www/base.html" %}

{% block main %}
<div id='content'>

    <div id='loggedon'>
        <p>
<a href='{{ site }}/secure/gwredir.php?a=ImportCCR' target='ccr'>Import</a> CCR
<iframe name='homeframe' id='homeframee1'
src='/acct/home.php' width='98%' allowtransparency='true'
frameborder='0' scrolling='no' height='550px' ></iframe>
        </p>
    </div>
</div>

{% endblock main %}
```

The first line, the bit that says `{% extends "www/base.html" %}` is a template instruction. The file “base.html” is HTML, with holes in it. Each hole is named, and can be filled in later by other files. In this case, “ud.html” fills in the hole named “main” -- look for `{% block main %}`. Suppose you modify ud.html, and run the templating system. Great, you've modified your appliance! When MedCommons releases a new version of the software, your modification will be lost: ud.html is part of the MedCommons software distribution.

The console looks for templates in two directories: the first directory it looks in is the customize directory, which will never be overwritten by MedCommons installation software. This directory is typically `/var/www/console/customize`.

So, instead of editing `/var/www/console/templates/www/ud.html`, we make a copy of it and place it in `/var/www/console/customize/www`. Then any changes we make will override any MedCommons-supplied **ud.html**.

When you are done editing the HTML, you can 'publish' by clicking on the “**Publish All**” button on the Console's Configuration page: <http://site/console/config/>

Changing Headers or Footers

If we change a file that other HTML 'extends' – extends using the template system – then all those pages will be modified.

The **www/root.html** file is extended by all MedCommons-supplied HTML. Changing this file can dramatically change the entire site.

Before making any changes, take a copy and place it in `/var/www/console/customize/www`. This prevents any changes we make from being overwritten by newer versions of the Console.

Now edit the `/var/www/console/customize/www` versions.

When you are done editing the HTML, you can 'publish' by clicking on the “**Publish All**” button on the Console's Configuration page: [**http://site/console/config/**](http://site/console/config/)

Changing Outgoing Email

All emails that are sent are parameterized by templates in the directory **/var/www/mc_templates/EmailTemplates**. For each email, there are two varieties: the text, and the HTML. The text is visible to recipients with text-only email clients. The HTML is visible to recipients with graphical email clients.

For example, the email sent if a user forgets his/her password has two flavors:

/var/www/mc_templates/EmailTemplates/forgotText.tpl.php

/var/www/mc_templates/EmailTemplates/forgotHTML.tpl.php

Fact Sheet

HTML templates live in '**/var/www/console/templates**'

Customers need to make changes, and these changes should not be overwritten by new versions of software provided by MedCommons.

These changes can live in '**/var/www/console/customize**'

Summary: The console looks in these directories, in order, for rendering a template:

/var/www/console/customize

/var/www/console/templates

The top level of the template path is for console pages. Look in **/var/www/console/templates/index.html** for the home page of the console.

The console can also produce the web pages for 'central' – the set of web pages that include the home page, the login and registration pages, and the gadgets/widgets.

The **www** subdirectory is where the web page templates live.

Summary: The main web pages are created by the console. The templates are located, in order:

/var/www/console/customize/www

/var/www/console/templates/www

If you want to change the home page for all appliances, you must edit

/var/www/console/templates/www/index.html

This needs to be checked out of and checked back into Subversion.

If you want to change the home page for a single appliance, say on **cisco.myhealthspace.com**, then copy **/var/www/console/templates/www/index.html** to **/var/www/console/customize/www**. Edit the file in **customize/www**, and then the console will use that one instead of the one in **templates/www**.

Templates have a nice feature: Template Inheritance. This works a lot like object-oriented inheritance. The first line of **www/index.html** is:

```
{% extends "www/base.html" %}
```

The inherited or parent templates define the shape of the resulting file, with named 'holes' that can be filled in by the original or child template.

The file **www/base.html** turns right around and inherits **www/root.html**:

{% extends "www/root.html" %}

If you want to change the look of all files, on all appliances, you must edit

/var/www/console/templates/www/root.html

This file, as above, need to be checked out of and checked back into Subversion.

If you want to change the look of all files on a single appliance, say on **cisco.myhealthspace.com**, then copy **/var/www/console/templates/www/root.html** to **/var/www/console/customize/www**.

The template language is documented here:

<http://www.djangoproject.com/documentation/templates/>

The list of files that get processed is in

/var/www/console/templates/www.xml

And, to prevent customizations from being overwritten by MedCommons, this can be overridden by:

/var/www/console/customize/www.xml

Each file is listed as a **<template in='filename' />** XML element in **templates/www.xml** or **customize/www.xml**.

Any attributes in that **template** element are used as template variables.

For example:

<template in='foo.html' title='Sample Code' />

If the file **foo.html** contains the template bits **{{ title }}**, that is replaced with **'Sample Code'**

Groups of template elements are placed within **<group>** XML elements in **templates/www.xml** or **customize/www.xml**.

Any attributes in that **group** element are used as template variables for all files within that **group**.

Group elements can be nested.

All the properties that begin with **'ac'** in the **mcproperties** table are used as template variables, as well.

Example: **acLogo** in **mcproperties** will replace **{{ Logo }}** in any HTML file.