Skinning Sakai

SKINNING SAKAI 1

BACKGROUND 1

CREATING A SKIN 2

GETTING STARTED 2
THE PORTAL 4
THE TOP AREA 4
The mast head 5
The tabs.......... 7
THE SITE AREA 9
Structure 9
Site navigation..........10
Content..........14
Tool title 15
Specifying the unit..........16

APPENDIX A: PORTAL PARTS 19

APPENDIX B: BLOCK DIAGRAM 21

Background

This document is intended for a simple purpose: modifying Sakai to reflect a look and feel consonant with your institution. It will try to guide you through the job of modifying the default look of Sakai. By way of example we will be creating a new skin on this basis for a hypothetical university. It assumes only knowledge of HTML and CSS.

There are two types of skins: portal skins and tool skins. This document addresses only portal skins, as this is were the need to brand an installation will probably manifest itself more clearly. Tool skins are also now in a state of flux as we consolidate and normalize the Sakai legacy tools and the new tools like Gradebook, Samigo, etc. However, since some of the definitions for the portal live in the tool css, here is an explanation of how the 2 interact and how things work in general.

A site will be associated with a skin via some mechanism:

1. the sakai.properties default skin setting points to that skin
2. the admin interface has been used to point to that skin
3. (for course sites) the course site creation process involves selecting a skin or having a skin selected automatically via association with a unit

This association consists of a string: “architecture”, “oncourse”, “foothills”, etc.

The template files that construct the portal will output a link to the corresponding CSS in the file system.

```
<link href="${sakai_portalskin}skin/${sakai_skin_id}.css"
type="text/css" rel="stylesheet" media="all" />
```

Where `${sakai_portalskin}` concatenates the directory where portal skins live and the name of the skin and `${sakai_skin_id}` is the skin string that that particular site is associated with. So:

sakai-portal/css/portalskins/sakai_core/skin/sakai_core.css

Tools function similarly:

```
<link href="$sakai_skin" type="text/css" rel="stylesheet"
media="all" />
```

Where `$sakai_skin` concatenates the path of the tool CSS directory as well as the name of the CSS file used by that skin. So:

sakai-portal/css/sakai_core.css

This is the structure of the skin directory simplified:

```
🗁  css
     skin1.css (tool skin)
    🗁portalskins
         🗁skin1
              🗁icons
              🗁institutional
                   banner_inst.gif
                   logo_inst.gif
              🗁skin
                   skin1.css (portal skin)
```

If you have worked with uPortal this should be familiar…

Creating a skin

In the Sakai layout framework all elements are addressable via CSS definitions, either through their ID, CLASS, or via inheritance because that element is a child of an addressable parent element.

There will be gaps in this coverage that we hope will be identified as new skins with new design needs are developed. Please call attention to these by notifying gsilver-at-umich.edu

What we will do is go through the steps required to create a new skin on the basis of the default Sakai skin.

Getting started

Copy the portalskins/sakai_core directory and the tool skin and rename as below. The locations here are the same in the source and the deployed version.

Before

- 📁css
  - **sakai_core.css** (tool css)
  - 📁portalskins
    - 📁**sakai_core**
      - 📁institutional (logos)
      - 📁icons
      - 📁skin (css and some images)
        - **sakai_core.css** (portal css)

After

- 📁css
  - **new_skin.css** (tool css)
  - 📁portalskins
    - 📁**new_skin**
      - 📁institutional (logos)
      - 📁icons
      - 📁skin (css and some images)
        - **new_skin.css** (portal css)

Modify sakai.properties to point to `new_skin.css` as the default skin as the default skin for unpublished new sites.

Again – if you see gaps in the “addressability” issue, and we will assume you will, or if you have any suggestions for improvements (I assume you will too!) – do contact me (gsilver-at-umich.edu). Thanks!

Note – if your local Sakai instance is compiled from CVS source, work in the deployed version, not in the source. If you work in the source and compile and deploy after every small change to see how things look you will go melancholy mad. Just make sure that you save your changes elsewhere so if you do run `maven` to recompile and redeploy you do not loose your work.

Note - your Sakai version. There were some changes to the basic framework post-1.5 that affect how things are skinned – I will make a note of these when they come up. If you are stuck with an older version I will note what changes are needed to what files to make things work well.

The Portal

The portal consists of 3 main areas, the top area, the site area, the footer. We will deal with each in turn.

The Top Area

The Top Area lives at present inside an iframe. The contents of this iframe respond to 3 different states: user not logged in, user logged in (show mast head), user logged in (mast head does not show). The iframe then has three corresponding classes:

```
sitenav-max (user logged in - show institutional branding)
sitenav-min (user logged in - do not show institutional branding)
sitenav-log (user not logged in - show login form)
```

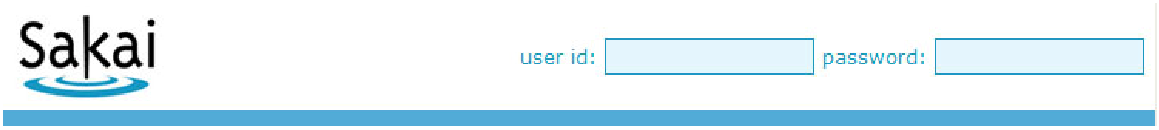
This is true for Sakai after 1.5.1 – previous versions did not address the different states of the top area iframe

The children of the iframe are

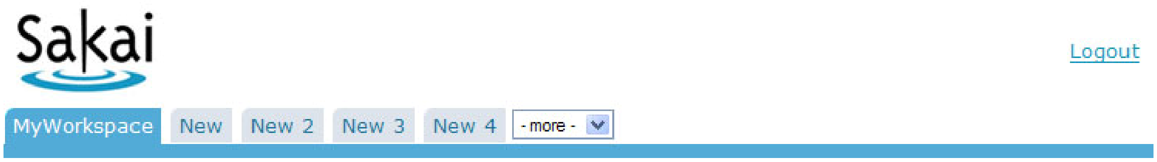
```
<table class="mast-head">
    <tr>
        <td class="left">
            <img /> Institutional logo image
        </td>
        <td class="middle">
            <img /> Institutional banner image
        </td>
        <td class="right">
            Login form or log out link
        </td>
    </tr>
</table>
<div class="tabHolder">
    <table border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td>
                <ul id="tabNavigation">
                    <li><a>Link</a></li>
                    <li><a> Link </a></li>
                    <li><a> Link </a></li>
                    <li class="selectedTab"><a>Link</a></li>
                </ul>
            </td>
            <td>
                Select item if more that the default max tab no.
            </td>
        </tr>
    </table>
</div>
<div class="divColor" id="tabBottom"><br/></div>
```

What we will do with this framework is change it from:

Login Form version

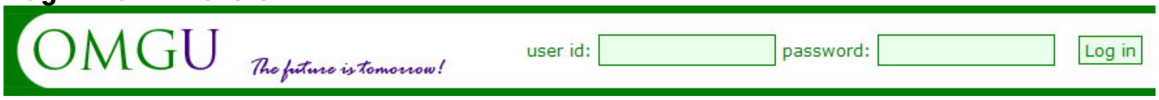


Logged in version

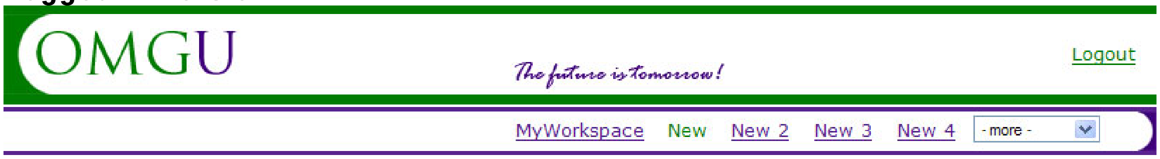


To this (I am sure you can do better!):

Login Form version



Logged in version



The top area that lives inside of the iframe has 2 parts (see structure above) – the **.mast-head** table and the **.tabholder** block.

Note: there is also another possible state of the top area iframe: demo mode. In this mode the whole **.mast-head** block is hidden. The only thing to worry about then is the height of the iframe, addressable in the CSS through its corresponding class – "sitenav-min"

The mast head

We will give the **.mast-head** table a top and bottom border and change the text color as well

Before (portalskins/….sakai_core.css)	After (portalskins/….new_skin.css)

.mast-head{
width:100%;
background-color:#fff;
margin:0;
font-size:.8em;
color:#1997BB;}
	.mast-head{
	width:100%
	background-color:#fff
	margin:0
	font-size:.8em
	color:**#007B00**
	border-width:.5em 0;border:color:#007B00;
	border-style:solid
	}

We will also change the color of links in the **.right** cell of the table.

Before	After
.mast-head .right a,.mast-	.mast-head .right a,.mast-head-r

head-r a:visited{color:#1997BB}	a:visited{color:**#007B00**}
.mast-head .right a:hover{color:#1997BB;text-decoration:underline}	.mast-head .right a:hover{color: **#007B00**;text-decoration:underline}

Note that some selectors are doubled up above (**.mast-head .right** and **.mast-head-r**) some older css still refer to the cell by the second name.

Now we will change the institutional logo `portalskins/new_skin/institutional/logo_inst.gif` to this horror:

And the `banner at portalskins/new_skin/institutional/banner_inst.gif` to this other one:

And adjust their respective parent cells

Before	After

N/A (sakai_core.css accepts defaults for these)	.mast-head .left {
	    padding:0 !important
	    }

ditto	.mast-head .left img {
	    border:0;
	    margin:0;
	    float:left
	}

ditto	.mast-head .middle {
	    vertical-align:bottom;
	    padding:0 !important;
	}

I added the img selector to the core – to make IE behave, the image needs to be floated. The reason for the **!important** is that some older Sakai have those definitions set in the template files.

Some things that can be added here – 1) a background color or image for **mast-head** (and then adjusting the logos to have that as background); 2) doing away with the banner (keep in file system but set it's **display:none**); 3) alignment of the images with text-align and vertical-align; etc.

You may not need a banner. Either all the info you need has been put in the logo, or the logo is way too big. Either serve a 1px big banner, or hide with

```
    .mast-head .right img {display:none}
```

Now – let's change how things look like on the right (where the login form or the logout links appear. We have already changed the color of the text for the **.mast-head** table above, as well as the link color.

Then the login input and the button:

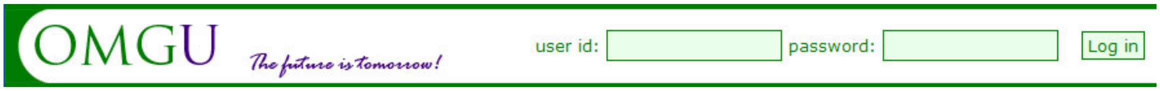
Before	After

input{
border:1px solid #1D97C2;
background-color:#E4F5FC;
padding:3px;
font-family:Verdana,Geneva,Arial,Helvetica,sans-serif;
color:#1D97C2
}
	input{
	border:1px solid #**007B00**;
	background-color:#**EAFFEA**;
	padding:3px;
	font-family:Verdana,Geneva,Arial,Helvetica,sans-serif;
	color:#**007B00**
	}

.button{
border:1px solid #1D97C2;
background-color:#E4F5FC;
margin-left:1em;
padding:1px;
color:#1D97C2
}
	.button{
	border:1px solid #**007B00**;
	background-color:#**EAFFEA**;
	margin-left:1em;
	padding:1px;
	color:#**007B00**
	}

Other things that can be done – modifying the **display** properties of the 2 above, adding floats, to move them around the right hand cell.

This is where we are at the moment, more or less (ugh!)



The tabs

The original basis for the tabs is a simplified version of what Terence Ordona did for uPortal (http://www.imaputz.com/uportal/tabstyles.html).

```
<div class="tabHolder">
    <table border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td>
                <ul id="tabNavigation">
                    <li><a>Link</a></li>
                    <li><a> Link </a></li>
                    <li><a> Link </a></li>
                    <li class="selectedTab"><a>Link</a></li>
                </ul>
            </td>
            <td>
                Select item if more that the default max tab no.
            </td>
        </tr>
    </table>
</div>
<div class="divColor" id="tabBottom"><br/></div>
```

At the core – it is simply an unordered list. In Sakai the parent of this list is a table, it’s parent a block (**. tabHolder**).

Let’s put the tabs to the right by changing the **.tabHolder** parent default text-align, give the whole thing a top/bottom border as well as a small top margin and a background image: and adding two new selectors to counteract various effects of a table living inside a block level element.

Before	After

.tabHolder{
width:100%;
height:auto;
padding-top:1px;
background:#fff}
	.tabHolder{
	width:100%;
	padding-top:1px;
	background:#fff;
	display:block;
	border-width:.2em 0;
	border-color:#59228D;
	border-style:solid;
	margin-top:2px;
	text-align:right;
	background: url(tab-back.gif) center right no-repeat;}

N/A	**.tabHolder table{**
	display:inline;
	}
	.tabHolder table td{
	padding:0 .5em
	}

Now the tabs. We will just "linkify" them by removing the background image of the <a>, changing the background color to white and setting text-decoration to underline (as well as removing various non-needed borders)

Before	After

#tabNavigation a,#tabNavigation a:link,#tabNavigation a:visited{
background-image:url(topleft-tab.gif);
background-position:top left;
background-repeat:no-repeat;
background-color:#DDE3EB;
border-bottom:1px solid #C0C0C0;
border-right:0px solid #fff;
border-top:0px solid red;
color:#25769C;
height:.8em;
margin:0px 0 0px 0;
padding:2px 6px 2px 6px;
text-decoration:none
}
	#tabNavigation a,#tabNavigation a:link,#tabNavigation a:visited{
	background-color:#fff;
	color:#59228D;
	height:.8em;
	margin:0px 0 0px 0;
	padding:2px 6px 2px 6px;
	text-decoration:underline
	}

And doing similar surgery to the :hover pseudo-class.

Before	After

#tabNavigation a:hover{
background-color:#FC0;
color:#25769C;
background-image:url(topleft-tab_hov.gif);
background-position:top left;
background-repeat:no-repeat
}
	#tabNavigation a:hover{
	background-color:**#fff**;
	color:**#007B00**;
	}

Now the selected tab, similar surgery. To indicate that this is the site you are in – no underline for it (this selector grouns the a and all the pseudoclasses)

Before	After

#tabNavigation .selectedTab a,#tabNavigation .selectedTab a:link,#tabNavigation .selectedTab a:visited,#tabNavigation .selectedTab a:hover{
background-image:url(topleft-
	#tabNavigation .selectedTab a,#tabNavigation .selectedTab a:link,#tabNavigation .selectedTab a:visited,#tabNavigation .selectedTab a:hover{
	background-color:#fff;
	color:#007B00;
	text-decoration:none;

`    curtab.gif);`
`  background-position:top left;`
`  background-repeat:no-repeat;`
`  background-color:#52AAD6;`
`  border-right:0px solid #fff;`
`  border-top:0px solid #C0C0C0;`
`  color:#fff;`
`  padding:3px 5px 5px 5px`
`}`
	`  padding:3px 5px 5px 5px`
	`}`

Other things that can be done – changing the select item (addressable as `.tabHolder table td select`) – many different things with the tabs (see Terence's excellent site for uPortal: http://www.imaputz.com/uportal/tabstyles.html), etc.

We no longer need **.tabBottom** so we will make it go away

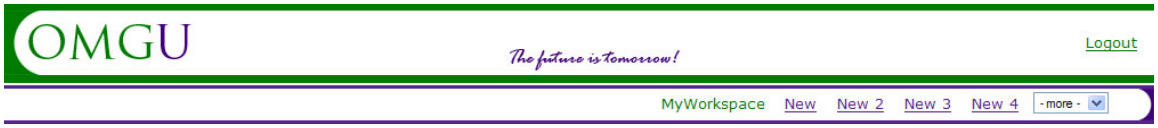
```
#tabBottom{display:none}
```

The last remaining thing to be done is adjusting the height of the top iframe to make things fit now.

```
.sitenav-max{height:6.6em !important}
.sitenav-log{height:4.3em !important}
```

Use whatever measurement option you feel comfortable with above (px, em, etc.)

Here is the end result in all it's hideousness



The site area

Structure

The structure is simple:

```
<div id="container ">
       <div class="divColor " id="sidebar ">
              site navigation
       </div>
       <div id="content ">
              tools
       </div>
</div>
```

The basic Sakai skin had the navigation on the left, the content on the right. Since OMG U makes use of some interesting tools with very wide content, lets stack the two areas, freeing up some horizontal space.

In other words – we are going from this layout

Logo Banner	Links/Form
Tab | Tab | Tab | Tab | Tab | Tab | Tab	

Tool Tool Tool Tool Tool Tool Tool	Content

To this

Logo Banner Links/Form
Tab | Tab | Tab | Tab | Tab | Tab | Tab
Tool | Tool | Tool | Tool | Tool | Tool | Tool |
Content

To do so:

We no longer float the site navigation block (**#sidebar**), we want it at the top. The width also changes

Before	After

#sidebar{
float:left;
width:8.4em;
line-height:17px;
color:#fff
}
	#sidebar{
	width:100%;
	line-height:17px;
	color:#fff
	}

The content (**#content**) area no longer needs a left margin to account for the tool navigation…

Before	After

#content{
padding:.3em .3em .3em .3em;
margin-left:8.5em;
background-color:#fff
}
	#content{
	padding:.3em;
	margin-left:.5em;
	background-color:#fff;
	}

The container (**#container**) for both above does not change

```
        #container{}
```

Site navigation

To review – the site navigation container (#leftnavlozenge) is the child of the following nodes:

```
<div id="container">
       <div id="sidebar">
              <div id="leftnavlozenge">
              <div>
       </div>
       <div id="content">
              tools
       </div>
</div>
```

The site link container (**#leftnavlozenge**) lives inside the **#sidebar** box – or our purposes we will increase the width to 100% and margins and padding change – we also override the default background-color. In the template this box is classed as id="leftnavlozenge" class="divColor" – the later is used to class elements that all need to be the same color. This is convenient to make a change in one place and show up in many and you can use it when you are dealing with sub unit skins below, right now we want to override it.

Before	After

#leftnavlozenge{
width:11em;
padding:0;
margin-bottom:1em;
margin-right:0em;
margin-left:0px;
font-size:70%;
color:#fff;
clear:both;
}
	#leftnavlozenge{
	width:100%;
	padding:0;
	margin-bottom:.2em;
	margin-right:0;
	margin-left:0;
	font-size:70%;
	color:#fff;
	clear:both;
	background-color:#fff !important
	}

The unordered list inside this container changes as well: the list style set to "none none none" for an inline list. It will also be aligned left.

Before	After

#leftnavlozenge ul{
list-style:none;
margin:0;
padding:0;
border:none
}
	#leftnavlozenge ul{
	list-style:none none none;
	margin:0;
	padding:0 2em;
	border:none;
	text-align: left
	}

As do the list items – with a generous line height to take care of wrapping issues.

Before	After

#leftnavlozenge li{
margin:0;
width:auto
}
	#leftnavlozenge li{
	display:inline;
	line-height:2em;
	margin:0;
	width:auto
	}

And the links inside those list items and the css hack for them

Before	After

#leftnavlozenge li a{
display:block;
padding:5px 5px 5px 0.5em;
text-decoration:none;
	#leftnavlozenge li a{
	display:inline;
	padding:0 .3em;
	text-decoration:underline;

width:100%;
color:#fff;
}
html>body #leftnavlozenge li a{
width:auto;
border-width:1px 0 1px 1px;
border-color:transparent;
border-style:solid
}
	width:100%;
	color:#59228D;
	}
	
	html>body #leftnavlozenge li a{
	width:auto;
	}

Then – the link that indicates the site page you are on, shared with the :hover of the normal links

Before	After

#leftnavlozenge li a.selected,#leftnavlozenge li a:hover{
padding:5px 5px 5px 0.5em;
width:100%;
background-color:#fff;
color:#52AAD6
}
	#leftnavlozenge li a.selected,#leftnavlozenge li a:hover{
	padding:.3em;
	width:100%;
	background-color:#fff;
	color:#487B48;
	text-decoration:none
	}

Let's take out the iframe that indicates who is on the site. This actually may be already an option in your institution, set in the properties file.

```
#presence{
       display:none
       }
```

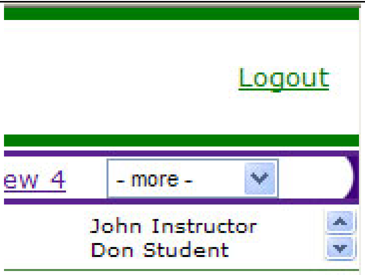
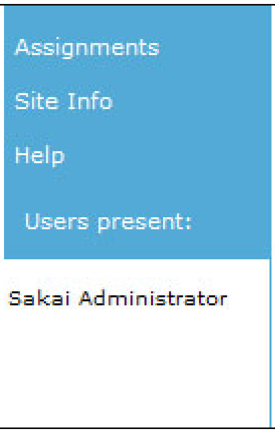
Or we could position it somewhere like at the top right, underneath the tabs:

```
#presence{
       position:absolute;
       top:100px;
       right:0;
       width:auto;
       height:1.8em;
       overflow:hidden
       }
```

And then modify the contents of the iframe (in the tool css at `css/new_skin.css`) to:

```
.presenceList{
       margin:0;
       padding:0;
       height:26px;
       width:auto;
       overflow:auto;
       }
```

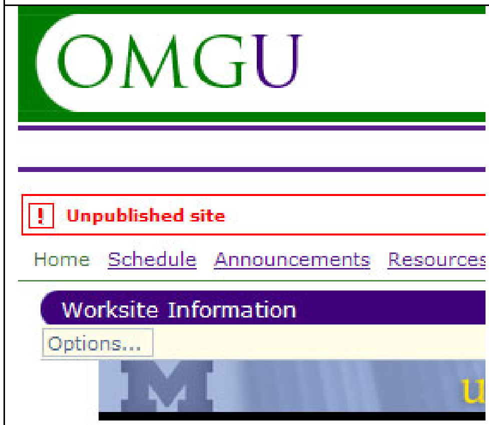
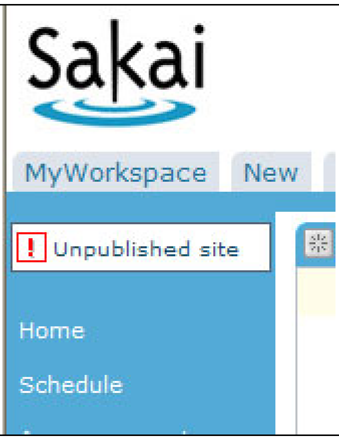
To give is a presence area like this:	Instead of this:

 Logout ew 4 - more - John Instructor Don Student	 Assignments Site Info Help Users present: Sakai Administrator

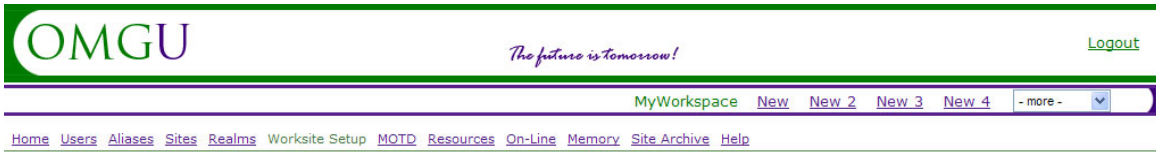
The alert box that displays notifications on a per site basis used to live above the site links on the left. This will put it at the top of the new location, 100% wide, obtrusive, but hard to miss.

Before	After

`.sakaiAlertBox /*used in alert for unpub sites*/{`
`background-color:#fff;`
`background-image:url(/sakai-portal/image/warn.gif);`
`background-position: 3px 3px;`
`background-repeat:no-repeat;`
`border:1px solid #536377;`
`clear:both;`
`color:#1A506A;`
`font-size:x-small;`
`margin:5px 2px 15px 2px;`
`padding:3px 3px 3px 20px`
`}`
`html .sakaiAlertBox/* */{background-position:3px center/* - adj for ie - */}`
	`.sakaiAlertBox{`
	`background-color:#fff;`
	`background-image:url(/sakai-portal/image/warn.gif);`
	`background-position: 3px 3px;`
	`background-repeat:no-repeat;`
	`clear:both;`
	`color:red;`
	`font-weight: bold;`
	`font-size:x-small;`
	`margin:5px 2px 2px 2px;`
	`padding:5px 5px 5px 25px`
	`}`
	`html .sakaiAlertBox/* */{background-position:3px center/* - adj for ie - */}`

 OMGU Unpublished site Home Schedule Announcements Resources Worksite Information Options...	 Sakai MyWorkspace New Unpublished site Home Schedule

Here is the site nav + the previous top. No alerts or presence areas.



Footer

Before dealing with the content of the **#content** block we will deal with the footer. All that has changed below is the background color as well as the text color and link color

```
#footer{
      background-color:#fff;
      color:#000;font-size:xx-small;
      margin:0px;
      padding:0;
      width:auto;
      clear:both;
      top:0;
      border-top:1px solid #007B00
      }
.footerExtNav a,.footerExtNav A:link,.footerExtNav A:visited{
      color:#007B00;
      text-decoration:none;
      margin:1em
      }
.footerExtNav a:hover{
      color:#007B00;
      text-decoration:
      underline;margin:1em
      }
```

Content

To review – the content area is the child of the following:

```
<div id="container">
      <div id="sidebar">
      </div>
      <div id="content">
            tools
      </div>
</div>
```

The content area is a set of nested divs that lays out the tools in a grid, in a two column or in a one column layout depending on the page.

Each tool is blocked out as follows:

```
<div (properties depend on page layout, are set)>
     <div class="channeltitlewrap ">
          <iframe class="channelTitleIframe "/>
     </div>
     <div class="uportalchannelcontent ">
          <iframe class="channelMainIframe "/>
     </div>
</div>
```

The first iframe holds the title of the tool, as well as links to the different actions that can be performed in this channel. The second iframe folds the tool's content.

There are any number of things that could be changed here, as most everything is addressable. We will just get rid of the box bounding the tool content. Never mind the name of this class – it is a left over from some experimental work using uPortal as the Sakai portal engine.

```
.uportalchannelcontent{
      width:99%;
      margin-bottom:.5em
      }
```

Tool title

Since the tool title lives in a tool inside of an iframe, and not on the portal, we need to modify the tool css, which can be found at `css/new_skin.css`

The title content is contained in a table with this structure:

```
<table class="chefPortletStyleClass">
      <tr>
            <td class="chefTitleStyleClass">
                  Links, title
            </td>
            <td class="chefActionTitleStyleClass">
                  More links
            </td>
      </tr>
</table>
```

So – to modify the title bar we will act on those classes by changing the background color, adjusting the padding and pointing to 2 new background images.

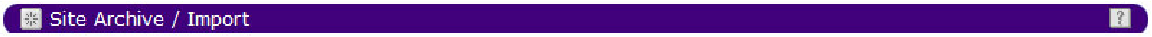
Before	After

td.chefTitleStyleClass{
background-color:#52AAD6;
color:#FFF;
padding:2px 4px 2px 4px;
background-image:url(portalskins/sakai_core/skin/topleft-p-title.gif);
background-position:top left;
background-repeat:no-repeat;
height:16px;
border-bottom:3px solid #52AAD6;
}
td.chefTitleStyleClass img{margin:0;
vertical-align: text-bottom;
}
td.chefActionTitleStyleClass{
background-image:url(portalskins/sakai_core/skin/topright-p-title.gif);
background-position:top right;
background-repeat:no-repeat;
background-color:#52AAD6;
color:#FFF;
padding:3px 4px 2px 2px;
text-align:right;
	td.chefTitleStyleClass{
	background-color:#40007B;
	color:#FFF;
	padding:2px 4px 2px 12px;
	background-image:url(portalskins/new_skin/skin/left-ptitle.gif);
	background-position:top left;
	background-repeat:no-repeat;
	height:16px;
	}
	td.chefActionTitleStyleClass{
	background-image:url(portalskins/new_skin/skin/right-ptitle.gif);
	background-position:top right;
	background-repeat:no-repeat;
	background-color:#40007B;
	color:#FFF;
	padding:3px 12px 2px 2px;
	text-align:right;
	vertical-align:top
	}

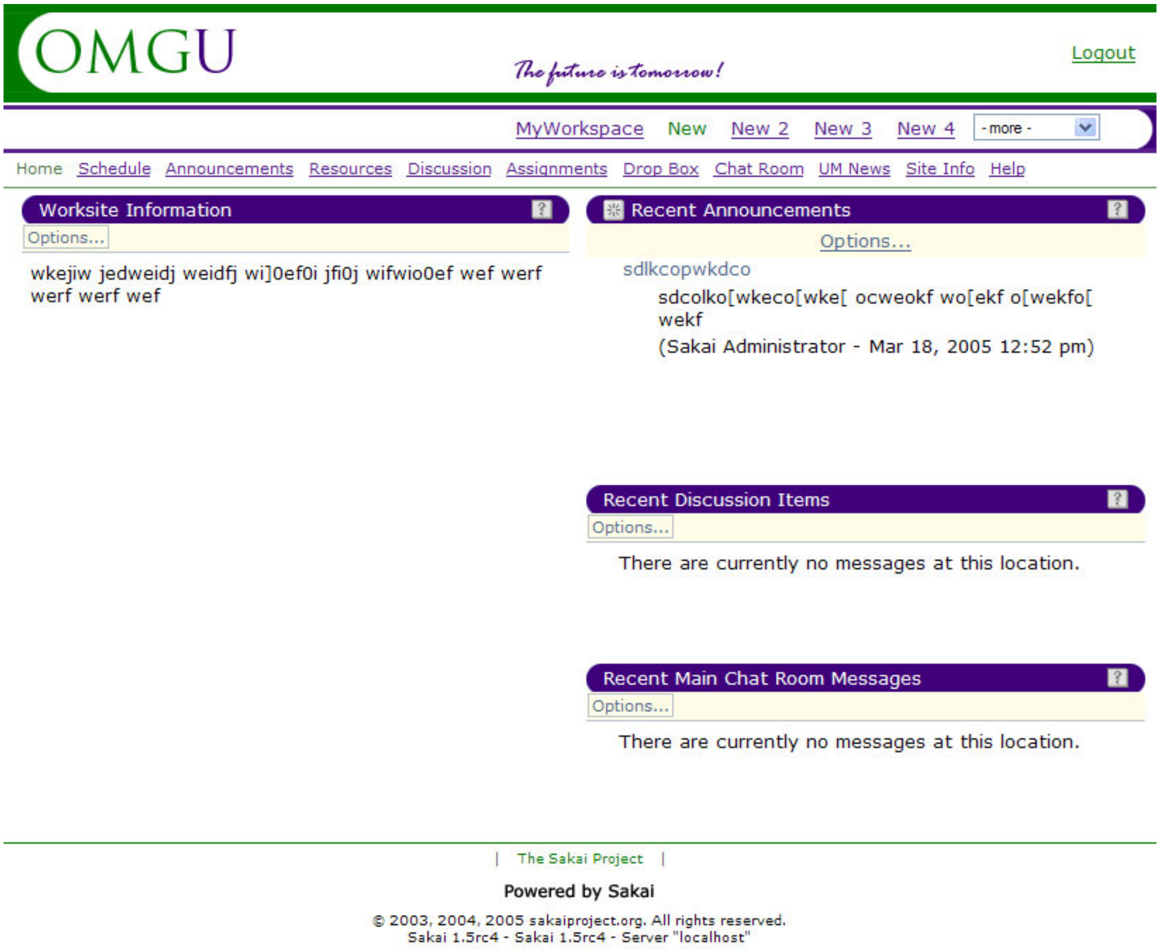
`        vertical-align:top`
`}`	

Note the url to the background image – it will expect to find it in the associated portalskin directory.

Here is the new one:



And here is the whole grizzly thing. All that remains is the tool skin.



wkejiw jedweidj weidfj wi]0ef0i jfi0j wifwio0ef wef werf werf werf wef

sdlkcopwkdco

sdcolko[wkeco[wke[ocweokf wo[ekf o[wekfo[wekf

(Sakai Administrator - Mar 18, 2005 12:52 pm)

There are currently no messages at this location.

There are currently no messages at this location.

© 2003, 2004, 2005 sakaiproject.org. All rights reserved.
Sakai 1.5rc4 - Sakai 1.5rc4 - Server "localhost"

Specifying the unit

We are revisiting how this is done for version 2.0. For right now if you need to serve a “special” skin to particular sites within your installation – but the differences are minor, this is done via a stub that calls a core (so that the main CSS always lives within one file for maintenance).

Say all that needs to change for site X is the addition to a logo indicating a school or sub-unit affiliation. In most Sakai core skins this logo would normally show up on the left, above the site navigation. Since in our example we have done away with that we will seek an alternate spot for it.

Note: this is the circumstance where the **.divColor** comes in handy – setting the color in the stub css serves a background color for several elements for sub-skins.

First – the stub business. As you did before, select the portalskin/new_skin directory and duplicate it, renaming the new directories and the new file.

Before	**After**
🗁css	🗁css
new_skin.css (tool css)	**new_skin_sub.css** (tool css)
🗁portalskins	🗁portalskins
🗁**new_skin**	🗁**new_skin_sub**
🗁institutional (logos)	🗁institutional (logos)
🗁icons	🗁icons
🗁skin (css and some images)	🗁skin (css and some images)
new_skin.css (portal css)	**new_skin_sub.css** (portal css)

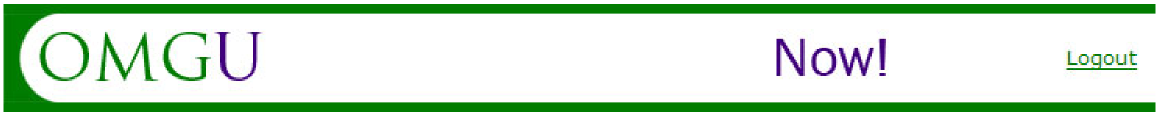
The content of tool css at `css/new_skin_sub.css` will be simply:

```
@import url("new_skin.css");
```

The content of portal skin css at `css/portalskins/new_skin_sub/skin/new_skin_sub.css` will be:

```
1   @import url("../../new_skin/skin/new_skin.css");
2   .mast-head .middle img{display:none}
3   .mast-head .middle {width:6em;
    background:url(../institutional/div_logo.gif) center left no-repeat;}
```

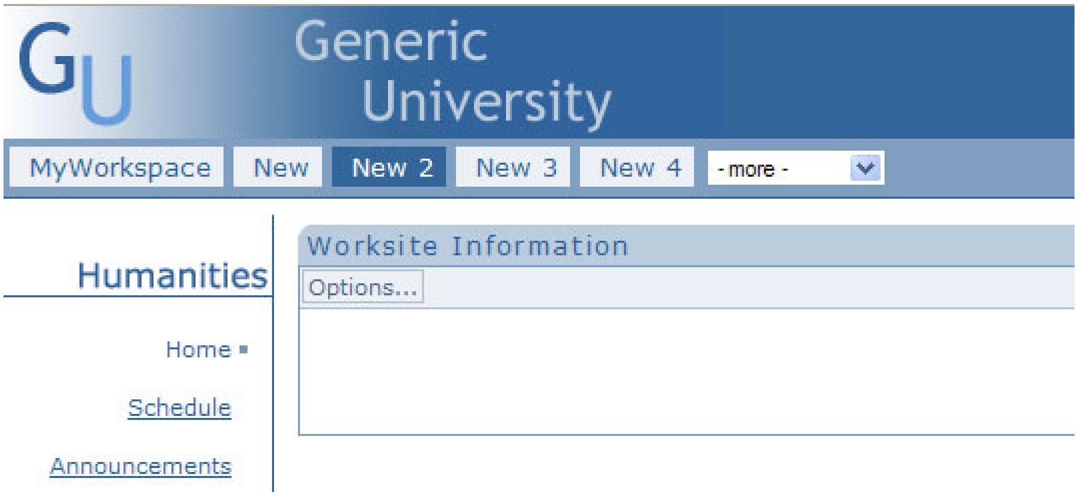
The first line imports the main stylesheet for the portal. In the second we hide any <img> children of the cell that would normally display the inst_banner.gif image. In the third line we display a new image via the background of the **.middle** cell of the **.mast-head** table, and given the cell some width. So (the *Now!* Text):



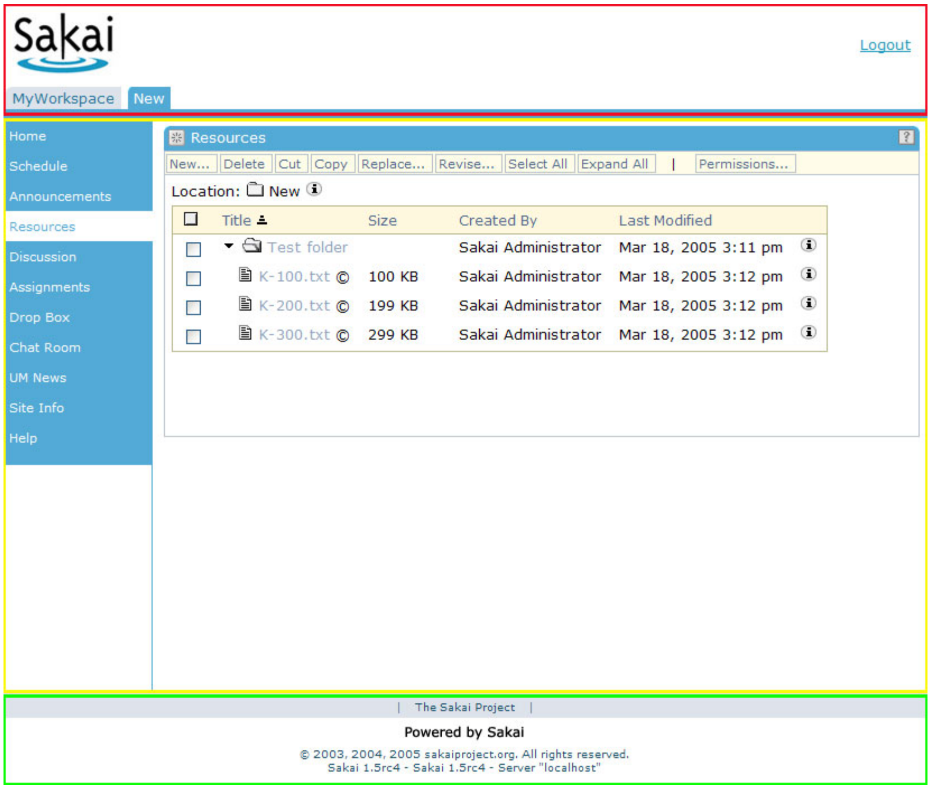
In left aligned site navigation situations the logical place would have been above the site navigation, as I mentioned. An example:

```
#divLogo{
        background-image: url(../institutional/div_logo.gif) top no-
repeat;
        height:41px;
        display:block;
        text-align: right;
        }
```

This will produce something like:


GU
Generic University
MyWorkspace
New
New 2
New 3
New 4
- more -
Humanities
Home
Schedule
Announcements
Worksite Information
Options...

Appendix A: Portal Parts



Location: New

☐	Title	Size	Created By	Last Modified
☐	Test folder		Sakai Administrator	Mar 18, 2005 3:11 pm
☐	K-100.txt ©	100 KB	Sakai Administrator	Mar 18, 2005 3:12 pm
☐	K-200.txt ©	199 KB	Sakai Administrator	Mar 18, 2005 3:12 pm
☐	K-300.txt ©	299 KB	Sakai Administrator	Mar 18, 2005 3:12 pm

| The Sakai Project |

Powered by Sakai

© 2003, 2004, 2005 sakaiproject.org. All rights reserved.
Sakai 1.5rc4 - Sakai 1.5rc4 - Server "localhost"

Here is the dehydrated structure of the portal, color coded in reference to the above. If the node has no descendants other than text nodes, it will show as closed (so: <stuff />)

```
<xhtml>
    <body>
        <iframe class="sitenav-max or sitenav-log or sitenav-min">
            <iframe-html>
                <body>
                    <table class="mast-head">
                        <tr>
                            <td class="left"><img /></td>
                            <td class="middle"><img /></td>
                            <td class="right"><form /></td>
                        </tr>
                    </table>
```

Holds the logos and on the right the login form or the logout link

```
                    <div class="tabholder">
                        <table>
                            <tr>
                                <td>
                                    <ul id="tabNavigation">
                                        <li>
                                            <a/>
                                        </li>
                                        <li class="selectedTab">
                                            <a/>
                                        </li>
                                    </ul>
```

The tabs. “selectedTab{“ is where you are at.

```
                </td>
                <td>
                  <select>                                If more than 5 tabs,
                    <option/>                             select item for the rest of
                  </select>                               sites
                </td>
              </tr>
            </table>
          </div>
          <div class="divColor" id="tabBottom"/>
        </body>
      </iframe-html>
    </iframe>
    <div id="container">
      <div class="divColor" id="sidebar">
        <div id="divLogo"/>
        <div id="leftnavlozenge">
          <ul>                                          Links to tools
            <li>
              <a/>
            </li>
            <li class="selected">
              <a/>
            </li>
          </ul>
        </div>
        <div id="pres_title"/>                          Title for presence iframe
        <iframe id="presence">
          <iframe-html>
            <body>
              <div class="presenceList">                  Users present list inside
                <span class="chefPresenceListItem"/>      this iframe
              </div>
            </body>
          </iframe-html>
        </iframe>
      </div>
      <div id="content">
        <div>
          <div class="channeltitlewrap">                Tool title
            <iframe class="channelTitleIframe"/>
          </div>
          <div class="uportalchannelcontent">           Tool
           <iframe class="channelMainIframe"/>
          </div>
        </div>
      </div>
      <div align="center" id="footer">                  Footer stuff
        <div class="footerExtNav" align="center">
          <a/>
        </div>
        <div id="footerInfo">
          <a/>
          <br/>
          <span class="sakaiCopyrightInfo"/>
          <br/>
        </div>
      </div>
    </div>
  </body>
</xhtml>
```

Appendix B: Block Diagram

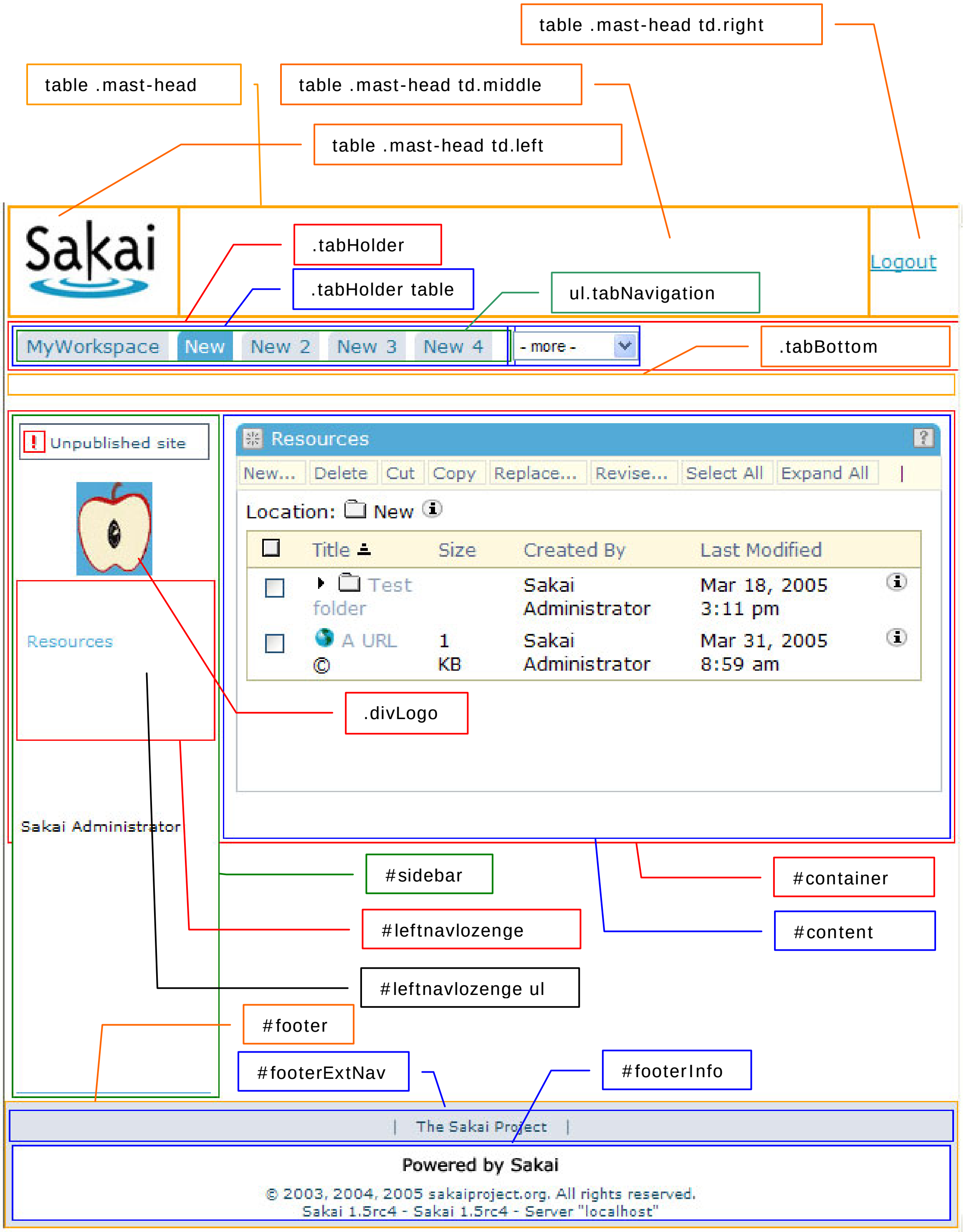


table .mast-head td.right
table .mast-head
table .mast-head td.middle
table .mast-head td.left
Sakai
Logout
.tabHolder
.tabHolder table
ul.tabNavigation
MyWorkspace
New
New 2
New 3
New 4
- more -
.tabBottom
Unpublished site
Resources
New... Delete Cut Copy Replace... Revise... Select All Expand All
Location: New
Title
Size
Created By
Last Modified
Test folder
Sakai Administrator
Mar 18, 2005 3:11 pm
A URL
1 KB
Sakai Administrator
Mar 31, 2005 8:59 am
Resources
.divLogo
Sakai Administrator
#sidebar
#container
#leftnavlozenge
#content
#leftnavlozenge ul
#footer
#footerExtNav
#footerInfo
| The Sakai Project |
Powered by Sakai
© 2003, 2004, 2005 sakaiproject.org. All rights reserved.
Sakai 1.5rc4 - Sakai 1.5rc4 - Server "localhost"