Changing the Appearance of the 2.2/2.3/2.4/2.5 Portal and Tools

Introduction

After the release of Sakai 2.1 and it's minor versions a group got together to tackle issues related to the surface layer of the rendering of the portal and to the tools.

Recommendations were collected regarding for example:

- what markup the portal needed for maximum flexibility
- what markup the tools needed for consistency, semantic value, usability, accessibility

The results are reflected in 2.2 – and they have an impact on how an institution customizes the rendering to fit their needs. The following documentation intends to help people transition skins created for 1.5, 2.0, 2.1 to 2.2, and 2.3 as well as help guide newcomers through the process of creating a new skin from scratch.

Unlike previous guides this one will separate reference materials from step by step how-to materials. Regarding the later we realized that some institutions would want a simple modification of a simple model, others would want more of a divergence, others a total transformation – so the how-to parts of the following are targeted accordingly.

How to proceed

Even if you have substantial work invested in a skin based on Sakai 2.1 or earlier you are better off starting from scratch and retrofitting the previous design into the 2.2 framework.

We will try to make this as painless as possible by mapping the 2.1 to the 2.2 locales and elements. If this is your case – skip ahead to the reference material at the end – specially the 2.1 > 2.2 conversion chart. If you are updating from 2.3 skip ahead to Appendix J

1) Download the source and build or get and build the 2.4 demo version. How to do this at this time has not been documented – we will point here at the relevant documents when they become available.
2) The file structure and the mechanisms by which sites are associated with skins has not changed since 2.1. See https://source.sakaiproject.org/svn/branches/sakai_2-1-1-x/docs/architecture/ for the details (2 docs with "skin" in their names)
3) Edit the css in the deployed version.
4) **Important:** save these files frequently. You do not need to stop and restart tomcat or the demo to see your changes.
5) This guide assumes knowledge of xhtml, css.

Further help

Join the WG: Default Skin site in http://collab.sakaiproject.org – you may need to create an account first. Then you can send and receive the site email and post any questions – suggestions, patches, etc.

Portal

The portal is where you will probably do most of the work. See the reference for the block diagram.

How to

Easy does it (a few easy changes for maximum effect)

Changing logos and images

There are 2 logos that appear in the markup in the site navigation area. The name of these files is hard coded – but the path is relative to the skin in the skin directory. You can edit these logos to be what you need – one is intended as a institutional logo, the other as an additional banner – this is reflected in the names – but you can used them for whatever you need to use them of course.

```
<skin path>/images/logo_inst.gif
<skin path>/images/banner_inst.gif
```

Alternatively – you can hide these either by making them 1px transparent images or by specifying in *portal.css*:

```
#mastLogo img{
      display:none
}
```

and then specifying the background of the parent block:

```
#mastLogo{
      background: #yourColor url (images/newlogo.jpg) top left no-repeat
}
```

this will give you more flexibility (do remeber that you will need to specify the height and width of the parent so that the background image will have a canvas). Or you can use both – a background image specified in the css and a foreground image (logo_inst.gif for example) – but beware of this if you need pixel precision placement of one on another. Experiment with the text resizing of your browsers...

The only other portal images worth noting are the reload and help icons as well as the “user is in the chat room” icon. These are rendered as background images – so (in *tool.css* as this is a “tool” thing):

```
.portletTitle .title a{
      background: url(images/reload.gif) center left no-repeat;
}
.portletTitle .action a{
      background: url(images/help.gif) center right no-repeat;
}
```

you can see that you can point at any image you care to use – specifying it in the definition of the “a” and the “a:hover” (and the “a:active” too) – you may need to adjust the sizes of the parent elements if your new images are very different in size from the default.

The “user is in chat room” icon is rendered as a background image of a classed list element (in *tool.css* again):

```
.presenceList li.inChat {
      background: url(images/chatpresicon.gif) -.1em center no-repeat;
}
```

Changing background colors

The background color for the whole portal in the default 2.4 skin is given in the *body* and inherited from there – so there will be nothing in your way. Some elements that you may want to change (see block diagram at end for what effect each will have):

```
#siteNavWrapper
```

```
            #mastHead
            .siteNavWrap
                  #siteNav
      #container
            #content
                  .col1
                  .col1of2
                  .col2of2
                        .portlet
            #toolMenuWrap
                  #toolMenu
                        #toolMenu ul
      #footer (and all its children)
```

Indentation denotes hierarchy/inheritance - so a background color for *#mastHead* will lay over the background color of *#siteNavWrapper*

If you need to you can also give any of the above elements borders. The default does this with *#toolMenuWrap, #footer, #sitenav-log* (site navigation iframe when not logged in), *.siteNavWrap, .portletTitleWrap*

Changing fonts

The font-family for the portal is specified once in the default – in the body, and inherited from there. Override as needed in the parent element of the text you want to change. For example – if you want the tool links to stand out:

```
      #toolMenu ul{
            font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
            width:auto;
            list-style: none;
            margin: 0;
            padding-right: 0;
      }
```

Whenever a specific size was prefered – this is indicated in the parent element as a size relative (via em, % or named size) to the user's browsers default. Beware of inheritance – font sizes are always relative the the parent's font size.

A special font issue is the one of the portlet titles. These display as part of the portal – but they are contained in a tool – hence they are rendered via *tool.css*:

```
      .portletTitle .title h2{
            display: inline;
            margin: 0 0 0 .2em;
            padding: 0;
            color: #555;
            font-weight: bold;
            font-size: 1em;
      }
```

As you can see the title is an <h2> (this is to help screen readers negotiate the iframed content). But it is an <h2> we want displayed as a bit of inline text, not a block level element – and the font size and weight (and margins and paddings) are there to override the default rendering of the <h2>.

Adjust sitenav iframe

A side effect of changing the images in the mast head will be that now your site navigation document may be bigger or smaller than the default iframe size it is displayed in. Adjust height of the three possible states of this iframe: *.sitenav-max,.sitenav-min,.sitenav-log*

Browsers will vary as to how iframes are sized. You can specify Firefox first and then use the tan hack to adjust for Internet Explorer. So – for example:

```
* html body #sitenav-max{
      height:specialheightforinternetexplorer
}
```

this will cover most browser/os combinations. Although it is always a good idea to shy away from relying on pixel precision where iframes are concerned. Give yourself plenty of wiggle room – do not rely on having the border of an iframe perfectly match the border of another element.

Getting more involved

Changing style of elements

Some of the most important elements in the portal are the navigational site and tool links. The structure of the markup there is designed to permit as many rendering variations as possible. The site links are a set of nested blocks whose inner block is an unordered inline list.

```
<ul id="siteLinkList">
      <li class="selectedTab">
            <a href="#">
                  <span>
                        My Workspace
                  </span>
            </a>
      </li>
      <li>
            <a>
                  <span>
                        Site title
                  </span>
            </a>
      </li>
      . . .
</ul>
```

Beginning in Sakai 2.5, the site tabs are by default rendered as literal tabs, as opposed to the more figurative tabs before 2.5.

But there are lots of ways that you might style the site tabs. The HTML used for the site tabs is very flexible and can support almost any tab design, literal or figurative, that you might come up with.

For more tab ideas, check out how other institutions have styled Sakai:

http://confluence.sakaiproject.org/confluence/display/UI/Sakai+screenshots

To learn how to create tabs in the new *literal* style, see *Appendix K – Sakai 2.5 modifications* below.

The tool links can also be customized since they are also an unordered list of the same sort:

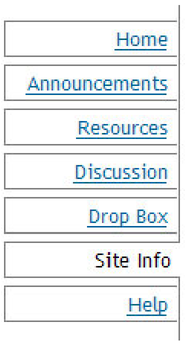
```
<div id="toolMenu">
      <ul>
            <li class="selectedTool">
                  <a class="selected" href="#">
                        <span> Home </span>
                  </a>
            </li>
            . . .
      <ul>
</div>
```

So – taking the default skin, you can

1) give a right border to *#toolMenuWrap*, make *padding-right:0*
2) give a border on all sides (except the right) to *#toolMenu li*
3) give *#toolMenu li.selectedTool* a

```
      margin-right:-2px;
      background: #fff;
```

And you will end up with something like this:

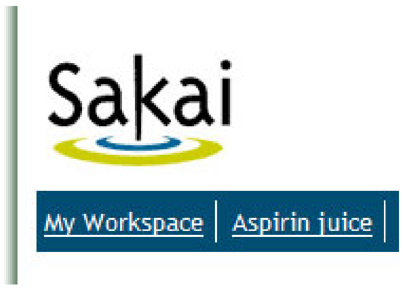


Which is part of the way to the tabbed metaphor.

Adjusting the link font and text-decoration, including some background images, etc. will get us the rest of the way there.

Make invisible visible

As mentioned before – there are a lot of style hooks available in the portal in 2.2 – adapted from folks that were forced to modify the servlet to meet their needs. The default skin does not make use of any of them. The comments in the *portal.css* make clear which these are. By way of an example – the following will give you a gradient on left and right (only left pictured below).



```
#portalOuterContainer{
      background:#fff url(images/port-bord-
      left.gif) top left repeat-y;
      border:1px solid #fff;
      padding-left:30px }
#portalContainer{
      background:#fff url(images/port-bord-
      right.gif) top right repeat-y;
      border:1px solid #fff;
      padding-right:30px }
```

Decorative edges can be applied to almost all the elements of the portal, as there is an abundance (an embarrasement?) of nested containers. Since a tool consists of:

```
<div id="content">
```

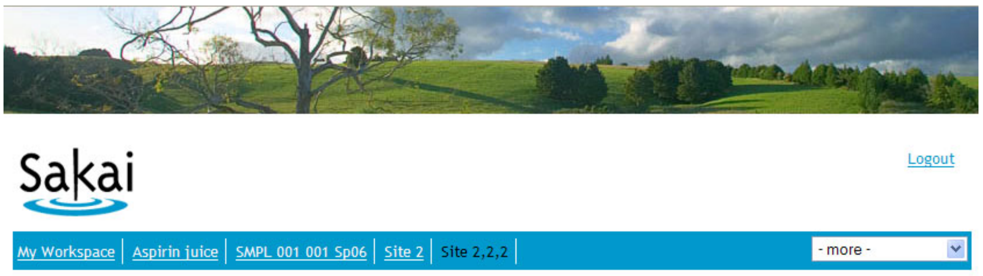
```
            <div id="col1">
                  <div class="portlet">
                        <div class="portletTitleWrap">
                              <iframe />
                        </div>
                        <div class="portletMainWrap">
                              <iframe />
                        </div>
                  </div>
            </div>
      </div>
```

The same techique above can be applied here to all borders of the "thing" that makes up a portlet if needed, or drop shadows, etc.

The abundance of leave-or-take-it blocks also provides loci for displaying images above and beyond what the actual 2 <img /> links provide you with. Applied to the default portal skin this:

```
      #portalOuterContainer{
            background:#fff url(images/top-back.jpg) top left no-repeat;
            padding-top:92px;
      }
```

Will get you this horror:



Extreme makeover

Since Sakai 2.2 is the first totally nested/floated container Sakai the most extreme thing that can be done is rearranging the blocks. The possible layout combinations are outlined here:

http://bugs.sakaiproject.org/confluence/download/attachments/10726/portal-blocks.pdf

and exercised here:

http://bugs.sakaiproject.org/confluence/x/5ik (links midway down the page – search for "the same structural markup")

Other settings

Login details

Sakai provides many different ways by which the user logs in. The following outline the possibilities. Left column indicates the setup – right where to change things.

Internal login – user logs into Sakai, using a Sakai account.	
Login form is the portal	In *portal.css*

	`#loginForm`
	`#eid`
	`#pw`
	`#loginForm label`
	`#submit`

Login form is in a separate page	In *`tool.css`*
	`.login`
	`.loginform`
	`. . . and their children`

If in separate page, login link in portal is a link or an image	In *`portal.css`*
	`#mastLogin`
	`#mastLogin img`
	`#mastLogin a`
	`#loginLinks`
	`. . . and their children`

External login – user logs into Sakai by login into an enterprise system	

Login link in portal is a link or an image	In *`portal.css`*
	`#mastLogin`
	`#mastLogin img`
	`#mastLogin a`
	`#loginLinks`
	`. . . and their children`

Some systems will have both internal and external login options active – check with your team.

Tool

Intro

Not much has really changed in the tool skin setup. Some of the older tools have been refurbished, some elements have been introduced, some definitions to bypass jsf issues added, some elements have been renamed. Consult the reference (the tool conversion chart and glossary) about these if you have a pre-existing skin

The mechanism via which a given tool shares the same definitions as the rest are the same as for 2.1. The mechanism by which a given tool skin shares a common set of characteristics as other skins in the same server is the same as well.

The `tool.css` and `tool_base.css` have been substantially commented as well, and formatted in a more readable format than in 2.1

Getting started

The following is meant for newcomers.

Participanting tools will use 2 css files of a given skin: `tool_base.css.` and `tool.css.`

The first is meant to hold the definitions common to all skins in an institution. The second one holds the particulars of a given skin. Some tools will make use of a css of their own which may or may not override the other 2.

As with the portal you can work in the file deployed to tomcat and there is no need to restart the server to see changes (although if you have a very aggressive browser cache – you may need to empty that periodically).

How to

Easy does it

Change font. The tools specify the font family and size in the *body* and override both as needed. Below is the list in tool.css where these overrides take place:

- .discTria
- h2,h3,h4,h5,h6
- .listHier th h3,.listHier th h4,.listHier th h5,.listHier th h6,.listHier td h3,.listHier td h4,.listHier td h5,.listHier td h6
- label
- textarea
- .navIntraTool
- #submitnotif

Aside from the *hx* used inside of the *`.listHier`* table and the *`textarea`* there is no crucial reason for the overrides. In the first case wanted to provide a hierarchy for screen readers but maje the *hx* display as normal text for sighted users. In the *`textarea`* case we found that in some browsers this value did not inherit adequately. In the case of *`label`* just wanted to differentiate the input label from the surrounding info without resorting to bold etc.

Beware of inheritance regarding font size. *`tool.css`* has in the comments an indication of the hierarchy.

Change background colors

The following is a list of items that either have a background color or may benefit from one. It is not complete – but if you want to refer or complement the portal scheme these will be some places to do so.

Name	**Seen in …**
`.listHier th`	The table header of any listing – see any populated list of resources, announcements, etc.
`table.lines tr:hover`	The hover color for medium complex tabular data. See populated list of resources
`.discTria`	The open to disclose hidden section twiggle. Create and preview an annoncement.
`.navIntraTool`	The generic toolbar of the tool (displays at top)
`.navPanel`	The generic navigation panel – can have floated navigation children. It is meant as a general float breaker – but there is no reason it cannot have a background. Adjust the padding if you do this. It is invisible in the default – but you can see it in the resorce listing, for example.
`.highlightPanel`	When there is some important, high stakes info. Create an assignment and respond to it as a student – see the “honor code” box
`.portletTitle`	The title of a portlet.

.act	The contained for formish actions. Create an announcement – look in the bottom. Invisible in default.

More involved

Change style of elements

Aside from changing the background color you can also modify display of the most common widgets with borders, background images, etc. If you feel like rewarding CSS2 compliant browsers the range of possibilities increases dramatically.

This for example

```
      .listHier th:first-child{
            background:#ddd url (images/tab_th_lt.gif) top left no-repeat
      }
      .listHier th:last-child{
            background:#ddd url (images/tab_th_rt.gif) top right no-repeat
      }
```

will give top rounded corners (if the referenced images are round) to the generic tabular data table.

Some other elements that can be modified are links, for example. If you want the link children of *.navIntraTool* to look like buttons, instead of links -

```
      navIntraTool a {
            border-width:1px
            border-style:solid
            border-color:toplight, rightshadow, bottomshadow, leftlight
      }
```

Substituting the colors with a complementary set. You should probably also address the :hover and :active pseudoclasses to gain the “buttony” affordance fully.

And so forth...

Gotchas

Background images

Beware of complex superimposition of background images that must render precisely – position is tricky and will vary between browser, OS, and even within the same browser with different settings.

Boilerplate content

Sakai comes out of the box with a collection of static xhtml pages – you will see these if you click through the links on the left before login in. Your institution will want to change these, add others, etc.

If you reference the default tool stylesheets of your institution in these pages, use a <div class=”portletBody”> as the sole child of <body />, use xhtml (*without* the xml declaration that bungs up things in Internet Explorer) and clear any floats, you will have a page that looks good and like it belongs. If you see scrollbars something is off. Validate to see what.

Conversely – you can use the javascript used in workspace_info.html in Sakai 2.3 to climb up the DOM ,get the reigning css links from the portal document and associate this static file with whatever css is in play in that site.

Iframes

You may have noticed that Sakai has some iframes. Iframes are a convinient way of doing what it does – but they do have some disadvantages that bear on making a usable skin.

Internet Explorer, for example, is fairly mulish about sizing them correctly horizontally. It will tend to respond well only when the inmediate outer container tells it what to do. If you do not modify these (`#header, #presenceWrapper, .portletTitleWrap, .portletMainWrap`) you will be OK.

The vertical resizing of the tool portlets is accomplished via javascript. Skinwise the only consideration is *`.portletBody`* (in *`tool_base.css`*) – it is being used to pad the content so that the resize is appropriate. If you see scrollbars, something is off.

Floats

Since all content in Sakai 2.2 (except tabular data and some JSF produced markup) is arranged in positioned boxes instead of tables – if you change any of the default box properties concerned you might find your skin dropping a float. You can adjust the padding, margin, width, float to get it back in shape. A nice schematic explanation of the problems can be found here:

http://nemesis1.f2o.org/aarchive?id=11

Margins

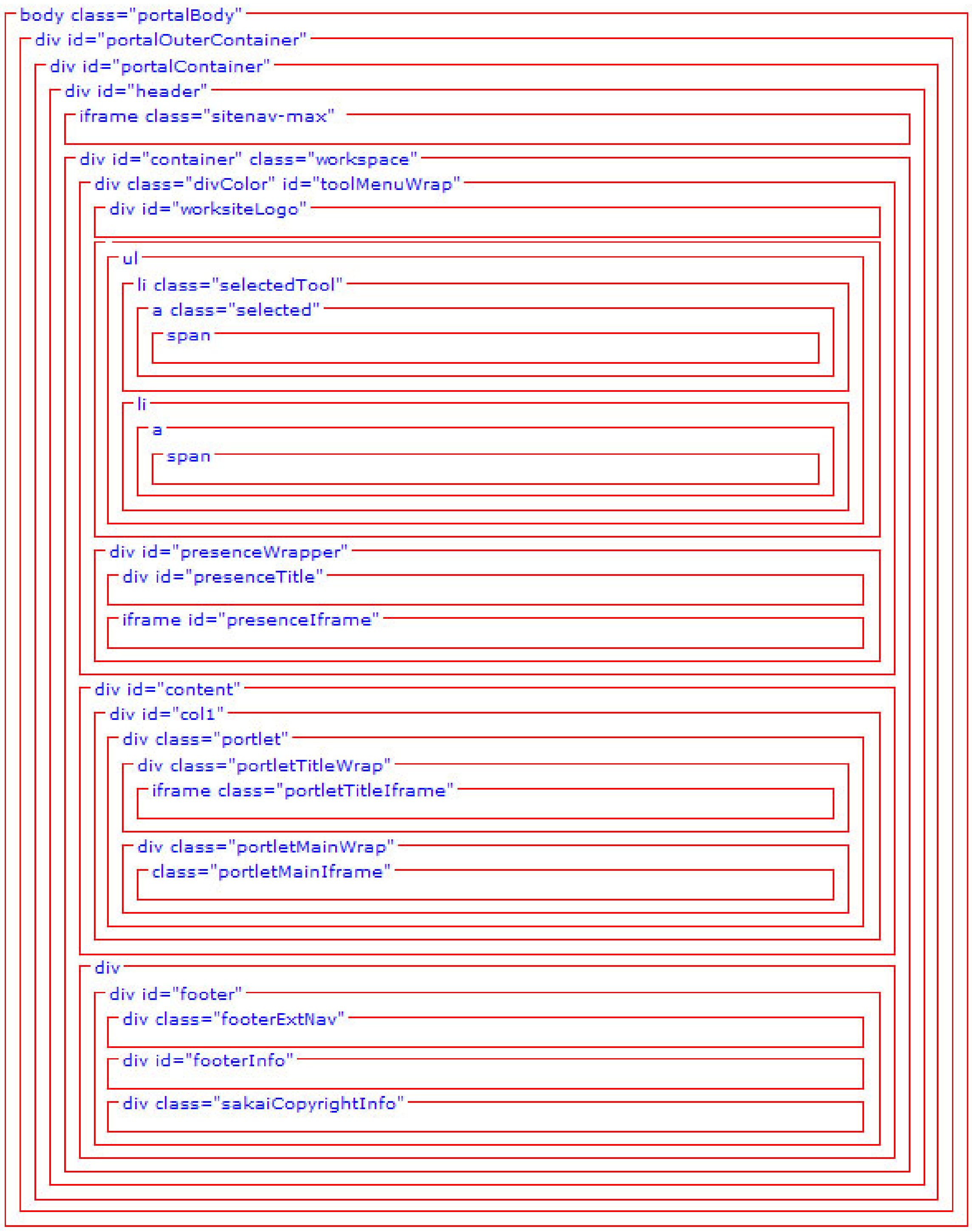
More of a tool issue. The default skin does not use any margins or paddings for the *body* or the *`.portletBody`* and relies instead on the white space between portlets and other portlet content and portal elements.

If you do decide to give either of the 2 elements above some padding – maybe because you gave them or their parent a border - keep in mind that some of the child elements (such as the toolbar [*`.navIntraTool]`*) look best when flush with the portlet edge – you may need to resort to negative margins in these cases.

Reference

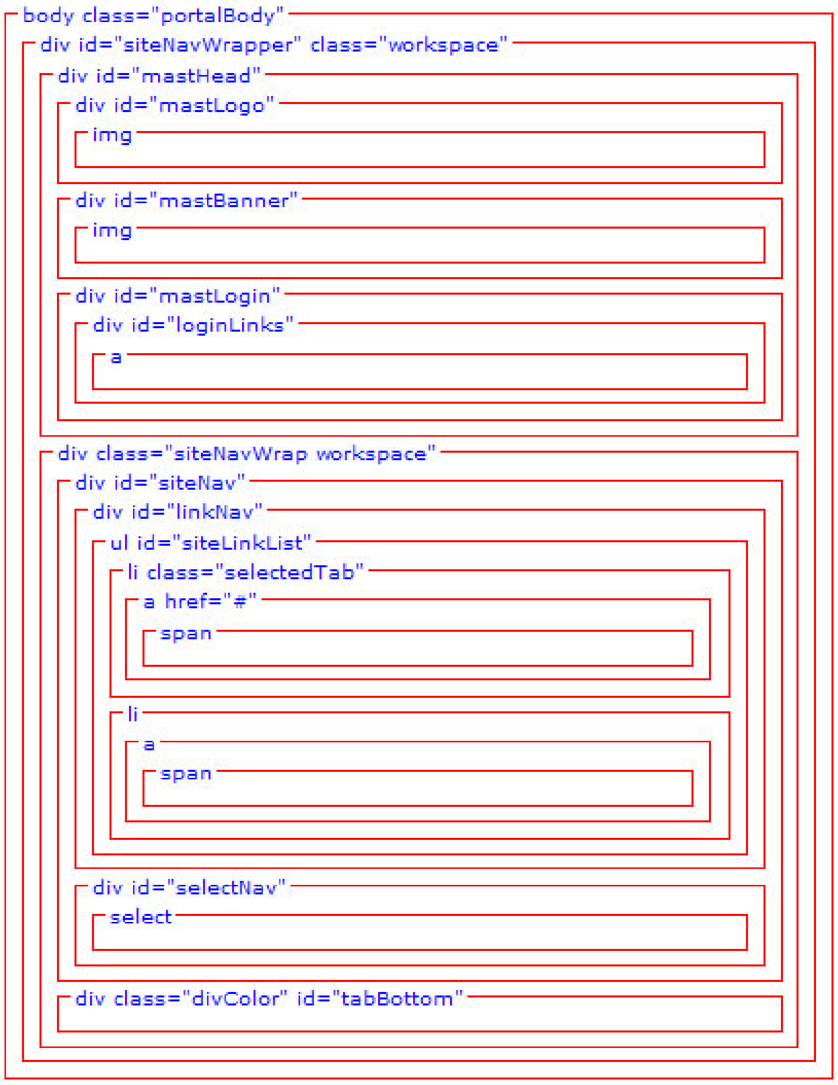
Appendix A -- 2.2 block diagram - portal structure

This is the dehydrated structure of the portal. Yikes.

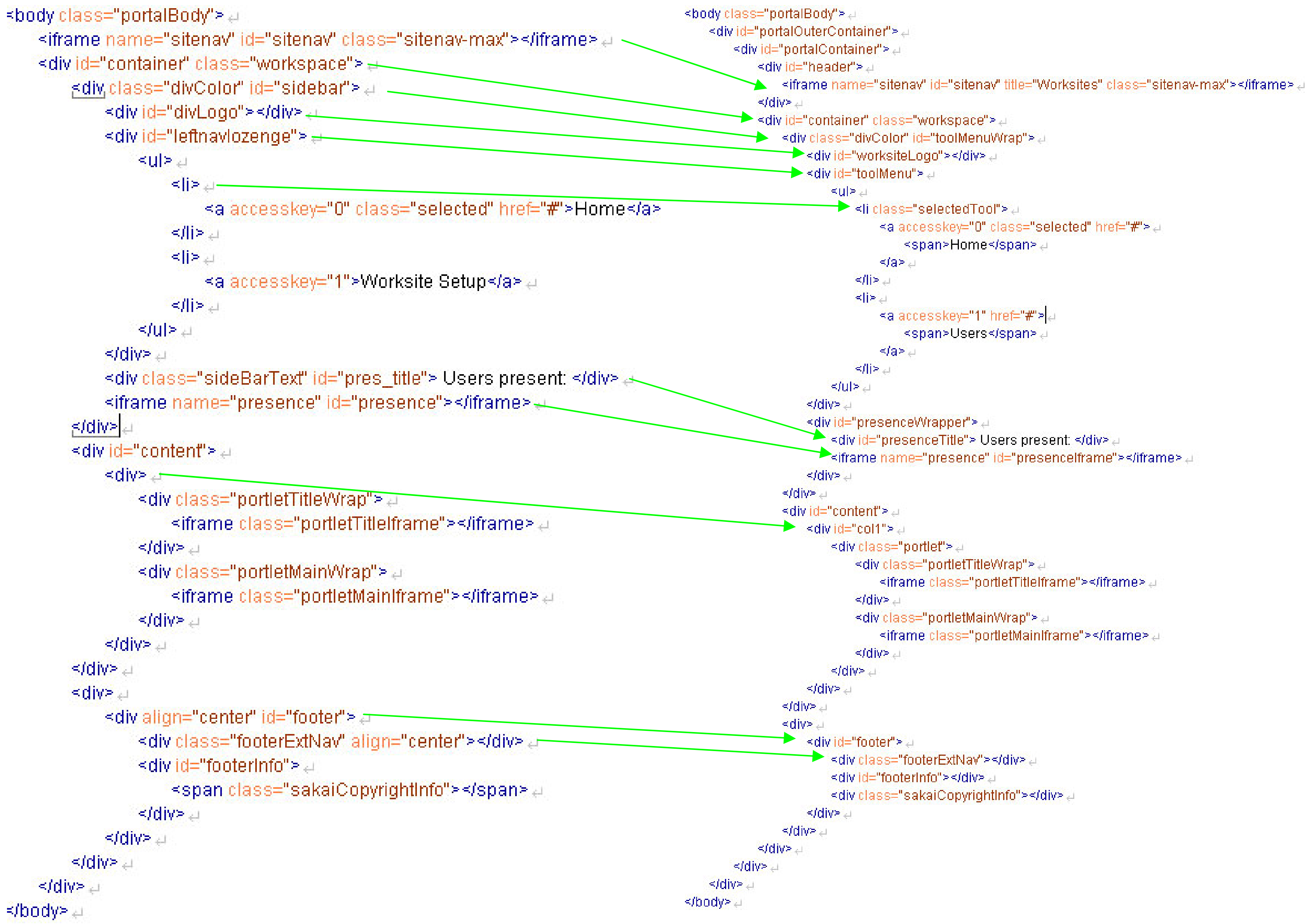

body class="portalBody"
div id="portalOuterContainer"
div id="portalContainer"
div id="header"
iframe class="sitenav-max"
div id="container" class="workspace"
div class="divColor" id="toolMenuWrap"
div id="worksiteLogo"
ul
li class="selectedTool"
a class="selected"
span
li
a
span
div id="presenceWrapper"
div id="presenceTitle"
iframe id="presenceIframe"
div id="content"
div id="col1"
div class="portlet"
div class="portletTitleWrap"
iframe class="portletTitleIframe"
div class="portletMainWrap"
class="portletMainIframe"
div
div id="footer"
div class="footerExtNav"
div id="footerInfo"
div class="sakaiCopyrightInfo"

Appendix B -- 2.2 Block diagram - site navigation -

This is the dehydrated structure of the site navigation (contents of top iframe). **Note:** Sakai 2.3 has done away with this iframe – the contents now live in the main doc – see Appendix I)



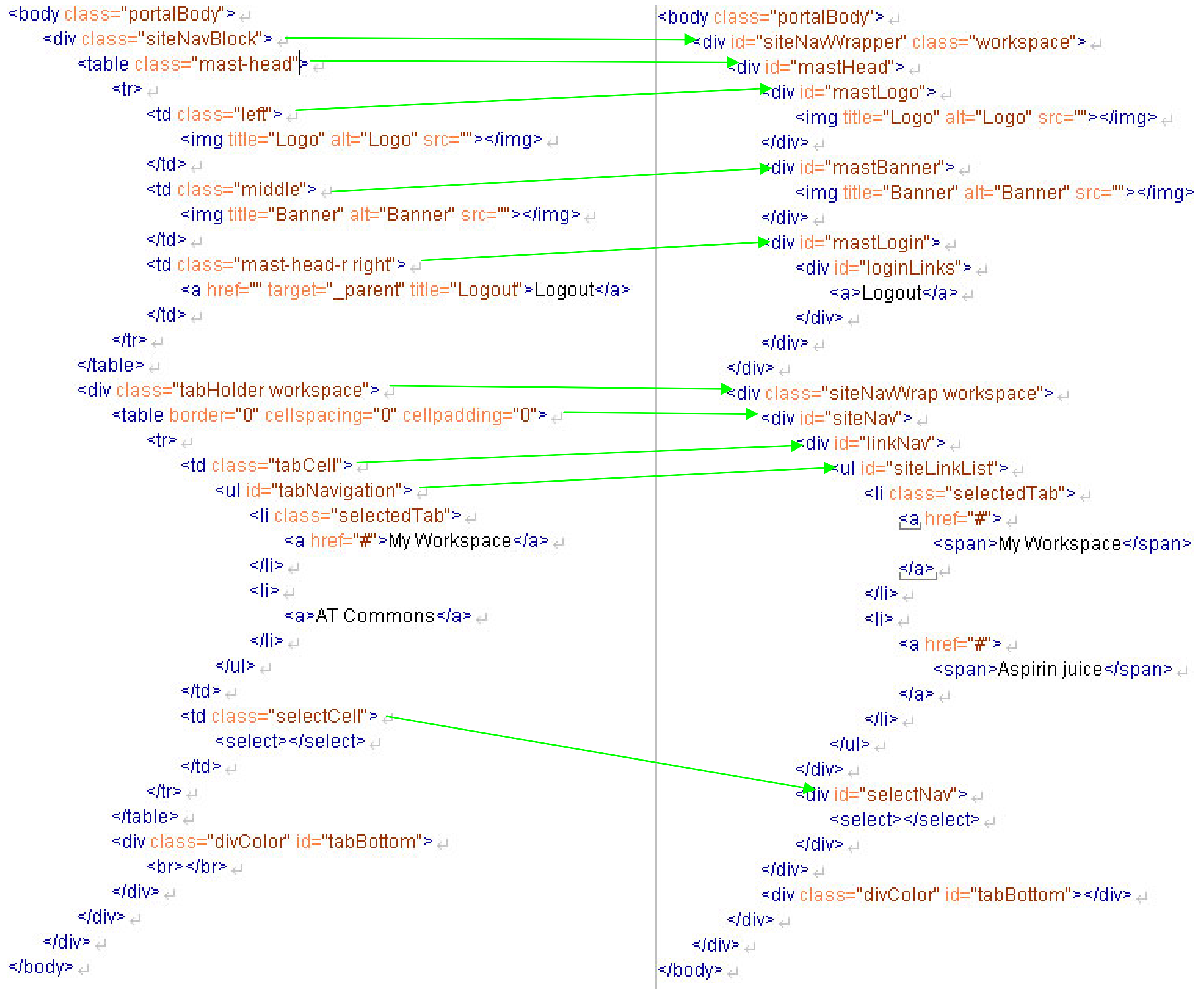
Appendix C -- Conversion chart – portal (2.1.x → 2.2)



```
<body class="portalBody">
    <iframe name="sitenav" id="sitenav" class="sitenav-max"></iframe>
    <div id="container" class="workspace">
        <div class="divColor" id="sidebar">
            <div id="divLogo"></div>
            <div id="leftnavlozenge">
                <ul>
                    <li>
                        <a accesskey="0" class="selected" href="#">Home</a>
                    </li>
                    <li>
                        <a accesskey="1">Worksite Setup</a>
                    </li>
                </ul>
            </div>
            <div class="sideBarText" id="pres_title"> Users present: </div>
            <iframe name="presence" id="presence"></iframe>
        </div>
        <div id="content">
            <div>
                <div class="portletTitleWrap">
                    <iframe class="portletTitleIframe"></iframe>
                </div>
                <div class="portletMainWrap">
                    <iframe class="portletMainIframe"></iframe>
                </div>
            </div>
        </div>
        <div>
            <div align="center" id="footer">
                <div class="footerExtNav" align="center"></div>
                <div id="footerInfo">
                    <span class="sakaiCopyrightInfo"></span>
                </div>
            </div>
        </div>
    </div>
</body>
```

```
<body class="portalBody">
    <div id="portalOuterContainer">
        <div id="portalContainer">
            <div id="header">
                <iframe name="sitenav" id="sitenav" title="Worksites" class="sitenav-max"></iframe>
            </div>
            <div id="container" class="workspace">
                <div class="divColor" id="toolMenuWrap">
                    <div id="worksiteLogo"></div>
                    <div id="toolMenu">
                        <ul>
                            <li class="selectedTool">
                                <a accesskey="0" class="selected" href="#">
                                    <span>Home</span>
                                </a>
                            </li>
                            <li>
                                <a accesskey="1" href="#">
                                    <span>Users</span>
                                </a>
                            </li>
                        </ul>
                    </div>
                    <div id="presenceWrapper">
                        <div id="presenceTitle"> Users present: </div>
                        <iframe name="presence" id="presenceIframe"></iframe>
                    </div>
                </div>
                <div id="content">
                    <div id="col1">
                        <div class="portlet">
                            <div class="portletTitleWrap">
                                <iframe class="portletTitleIframe"></iframe>
                            </div>
                            <div class="portletMainWrap">
                                <iframe class="portletMainIframe"></iframe>
                            </div>
                        </div>
                    </div>
                </div>
                <div>
                    <div id="footer">
                        <div class="footerExtNav"></div>
                        <div id="footerInfo"></div>
                        <div class="sakaiCopyrightInfo"></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
```

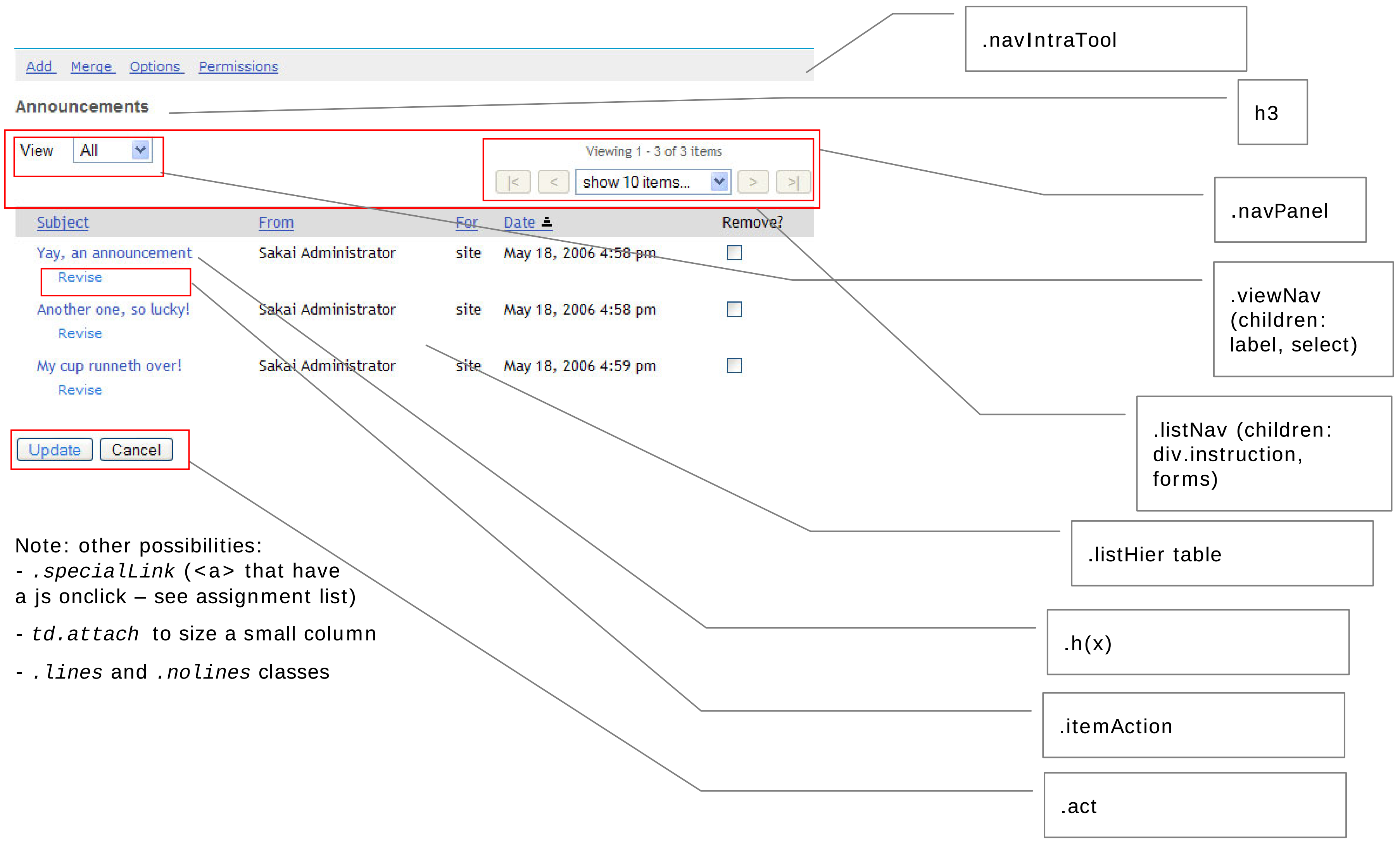
Appendix D -- Conversion chart – site navigation iframe (2.1.x → 2.2)



```
<body class="portalBody">
    <div class="siteNavBlock">
        <table class="mast-head">
            <tr>
                <td class="left">
                    <img title="Logo" alt="Logo" src=""></img>
                </td>
                <td class="middle">
                    <img title="Banner" alt="Banner" src=""></img>
                </td>
                <td class="mast-head-r right">
                    <a href="" target="_parent" title="Logout">Logout</a>
                </td>
            </tr>
        </table>
        <div class="tabHolder workspace">
            <table border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td class="tabCell">
                        <ul id="tabNavigation">
                            <li class="selectedTab">
                                <a href="#">My Workspace</a>
                            </li>
                            <li>
                                <a>AT Commons</a>
                            </li>
                        </ul>
                    </td>
                    <td class="selectCell">
                        <select></select>
                    </td>
                </tr>
            </table>
            <div class="divColor" id="tabBottom">
                <br></br>
            </div>
        </div>
    </div>
</body>
```

```
<body class="portalBody">
    <div id="siteNavWrapper" class="workspace">
        <div id="mastHead">
            <div id="mastLogo">
                <img title="Logo" alt="Logo" src=""></img>
            </div>
            <div id="mastBanner">
                <img title="Banner" alt="Banner" src=""></img>
            </div>
            <div id="mastLogin">
                <div id="loginLinks">
                    <a>Logout</a>
                </div>
            </div>
        </div>
        <div class="siteNavWrap workspace">
            <div id="siteNav">
                <div id="linkNav">
                    <ul id="siteLinkList">
                        <li class="selectedTab">
                            <a href="#">
                                <span>My Workspace</span>
                            </a>
                        </li>
                        <li>
                            <a href="#">
                                <span>Aspirin juice</span>
                            </a>
                        </li>
                    </ul>
                </div>
                <div id="selectNav">
                    <select></select>
                </div>
            </div>
            <div class="divColor" id="tabBottom"></div>
        </div>
    </div>
</body>
```

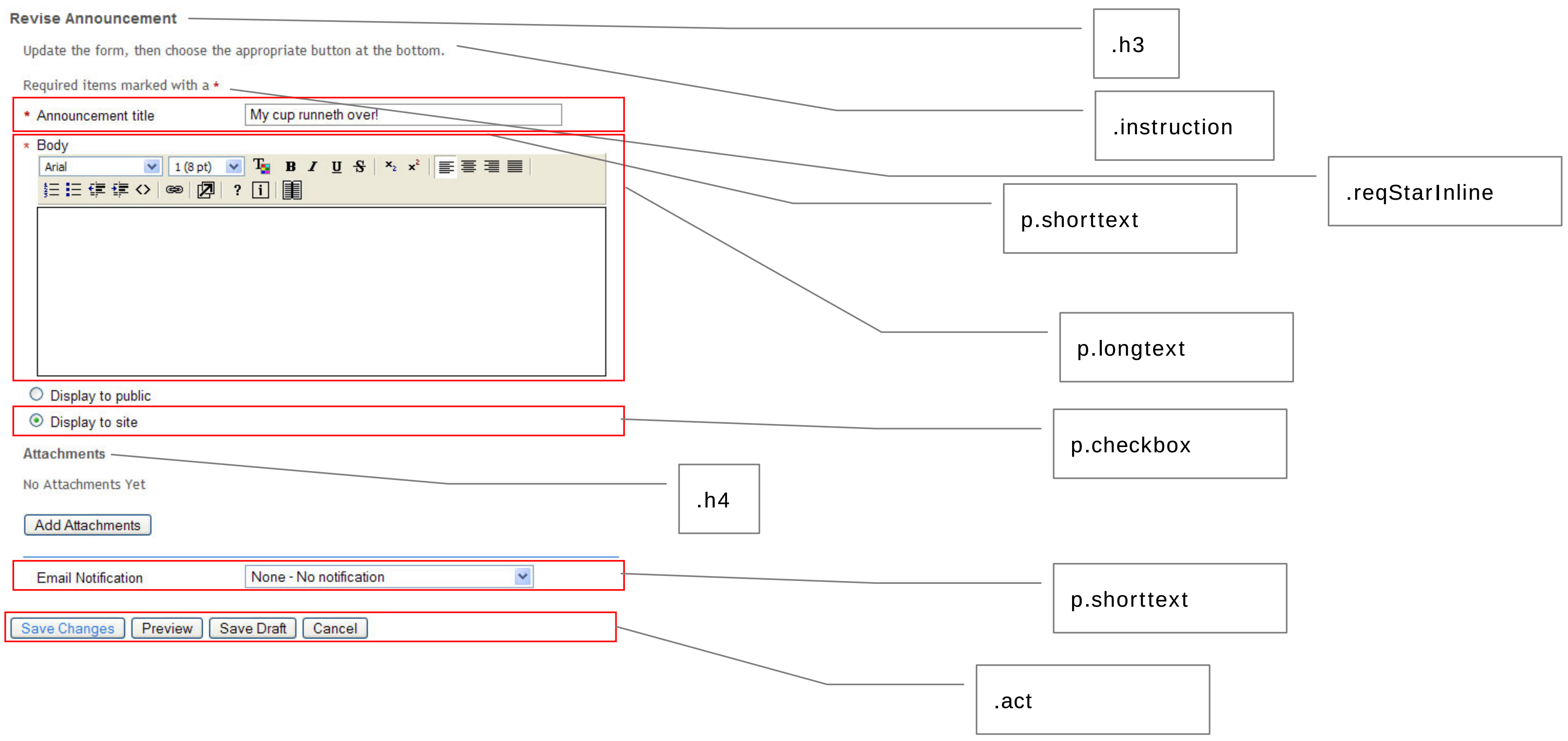
Appendix E -- 2.2 Block diagram –typical tool – list



Subject	From	For	Date	Remove?
Yay, an announcement Revise	Sakai Administrator	site	May 18, 2006 4:58 pm	☐
Another one, so lucky! Revise	Sakai Administrator	site	May 18, 2006 4:58 pm	☐
My cup runneth over! Revise	Sakai Administrator	site	May 18, 2006 4:59 pm	☐

Note: other possibilities:
- *.specialLink* (<a> that have a js onclick – see assignment list)
- *td.attach* to size a small column
- *.lines* and *.nolines* classes

Appendix F -- 2.2 Block diagram –typical tool – form



Revise Announcement

Update the form, then choose the appropriate button at the bottom.

Required items marked with a *

* Announcement title	My cup runneth over!
* Body	Arial, 1 (8 pt)
○ Display to public	
◉ Display to site	

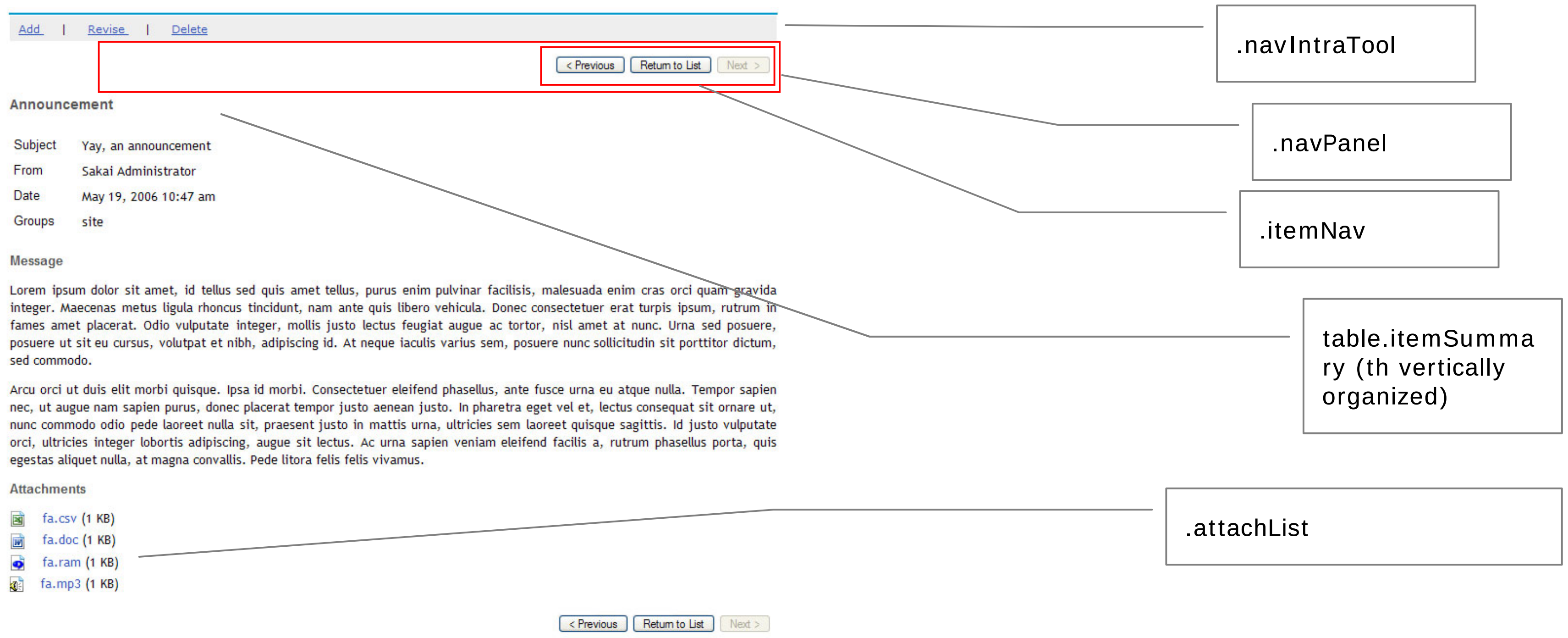
Attachments

No Attachments Yet

Add Attachments

Email Notification	None - No notification

Appendix G -- 2.2 Block diagram –typical tool – item



Announcement

Subject	Yay, an announcement
From	Sakai Administrator
Date	May 19, 2006 10:47 am
Groups	site

Message

Lorem ipsum dolor sit amet, id tellus sed quis amet tellus, purus enim pulvinar facilisis, malesuada enim cras orci quam gravida integer. Maecenas metus ligula rhoncus tincidunt, nam ante quis libero vehicula. Donec consectetuer erat turpis ipsum, rutrum in fames amet placerat. Odio vulputate integer, mollis justo lectus feugiat augue ac tortor, nisl amet at nunc. Urna sed posuere, posuere ut sit eu cursus, volutpat et nibh, adipiscing id. At neque iaculis varius sem, posuere nunc sollicitudin sit porttitor dictum, sed commodo.

Arcu orci ut duis elit morbi quisque. Ipsa id morbi. Consectetuer eleifend phasellus, ante fusce urna eu atque nulla. Tempor sapien nec, ut augue nam sapien purus, donec placerat tempor justo aenean justo. In pharetra eget vel et, lectus consequat sit ornare ut, nunc commodo odio pede laoreet nulla sit, praesent justo in mattis urna, ultricies sem laoreet quisque sagittis. Id justo vulputate orci, ultricies integer lobortis adipiscing, augue sit lectus. Ac urna sapien veniam eleifend facilis a, rutrum phasellus porta, quis egestas aliquet nulla, at magna convallis. Pede litora felis felis vivamus.

Attachments

- fa.csv (1 KB)
- fa.doc (1 KB)
- fa.ram (1 KB)
- fa.mp3 (1 KB)

Appendix H -- Conversion chart for tool

Cannot provide a tree version as for portal – as every tool, every state of a tool, is different, of course. Instead here is a side-by-side + notes. I will only include an item if the **structural** underpinning has changed.

Note: addding/editing your 2.1.2 stylesheets to bring them forward will drive you mad. Start with 2.2 and add the styling from 2.1.x. Below is just a note of things of note that have changed

In tool.css

Some things to note: content has been reorganized. Read the comments of the new default placeholder – it will tell you what it is, where to see an example.

2.1.2	**2.2**	**Notes**
table.toolTitle	div.toolTitle	Was a table, now is a set of nested divs – the div children have the same classname as of old (title/action)
.portletBody	.portletBody	Only change here – no h padding, no h margins – you may need to adjust this.
	tr.highLightRow	New
.chefEditItem,.itemSummaryForm	N/A	Gone (as well as the defs for their children)
.leftNav	.viewNav	Name change (was doubled up in 2.1.2)
.rightNav	.listNav	Name change (was doubled up in 2.1.2)
.rightNav span.instruction	.listNav div.instruction	Span made more sense as a div

Final note: the styling of some tools was either hard coded or tool specific – these tools have been brought into the fold – and their definitions are now in tool.css. The style elements are mostly about positioning –so they can be ignored. If you need to deal with them they are in SECTION 11 of *tool.css*. The tools affected are Chat, Discussion, Profile, Roster.

In tool_base.css

2.1.2	2.2	Notes
.listView, .leftNav	.listView	.leftNav is gone
.nav	N/A	Gone – UI elements have been reclassified (.viewNav, .listNav., .itemNav) etc.
	.*Panel	Added a set of panels – for information, to highlight, to indicate that a block is “look but do not touch” etc.
	H(x).textPanelHeader, .textPanelFooter, .textPanel	An item in a list. Read the comments in the css.
	. hierItemBlock	Wrapper for items above when the list is hierarchical.

Appendix I – Sakai 2.3 modifications

For Sakai 2.3 the Charon portal now displays the site navigation as well as the tool titles in the main doc (thanks Chuck!).

This greatly increases the accessibility of the portal and decreases the iframe troubles. But the 2.2 skins will need to be adjusted.

Below are the adjustments made to the default 2.2 portal.css to make it work in this new environment. This is a minimal list, other adjustments may be needed or wanted, it is just an example.

A) - Add some floats and clears.

```
#container{
        clear:both; <<<<<
        width: 98%;
        margin: 1em auto;
        border:1px solid #fff
}

#linkNav{
        float:left; <<<<
}
#selectNav{
        float:right; <<<<
        padding: .2em .4em .2em .2em;
        text-align: right;
}
```

B) – New site navigation container

Instead of the outermost sitenav container being the iframe, now it is #headerMax (for logged in) and #headerMin (fornot logged in). Edit accordingly - no need to specify a height - yahoo!

```
#headerMax,#headerMin{
        width: 98%;
        margin: 1em auto 0 auto;
}
```

C - take advantage of the fact that the site navigation is no longer in a hole and can wrap

```
#siteLinkList{
        white-space: nowrap;*/ >>>>>  out! out!  - tabs and select now will slide over each other
        list-style: none'
        font-size: .8em;
        margin: 0;
        padding: 4px 0;
        width: auto;
}
#siteLinkList li{
        display: inline;
        line-height: 2em; <<<<< in (so - when they slide over each other there is enough vspace)
}
```

D - Bring over to portal.css the .portletTitle and its children from tool.css

Since the title bars are no longer in an iframe, they need to be defined in the portal.css instead.

Adjust the font size of .portletTitle .title h2

Give the img children of .portletTitle a "border:none"

E. Other things

Finally - there was some work done in the Resources tool (sorting, hiding, timed resources) that needed some new selectors. These are in tool.css - add the tr#selectedReorder and all it's children to your 2.2 tool.css

Appendix J – Sakai 2.4 modifications

Not much change. By file below:

portal.css

No changes of substance.

tool_base.css

A lot of niggly bits – see below

Line number	**2.3**	**2.4**

109c109	`*html #wciframe{`	`* html #wciframe{`
116c116	`.itemSummary{`	`.itemSummary, .itemSummaryLite{`

122c122	`.itemSummary th, .itemSummary td.header{`	`.itemSummary th, .itemSummary td.header,`
		`.itemSummaryLite td.header{`

131c131	`table.itemSummary td{`	`table.itemSummary td, td.itemSummaryLite{`
137c137	`table.itemSummary caption{`	`table.itemSummary caption, caption.itemSummaryLite {`
457c457	`background-position: .3em;`	`background-position: 5px 5px;`
474c474	`background-position: .3em;`	`background-position: 5px 5px;`
490c490	`background: #fff url(images/warn.gif) .3em center no-repeat;`	`background: #fff url(images/warn.gif) 5px 5px no-repeat;`
509c509	`background: #fff url(images/warn.gif) .3em .3em no-repeat;`	`background: #fff url(images/warn.gif) 5px 5px no-repeat;`
613c614	`.shorttext,.longtext{`	`.shorttext,.longtext,.filepicker{`
620c621	`.shorttext label{`	`.shorttext label,.filepicker  label{`
627c628,629	`.shorttext input{`	`.shorttext input[type="text"]{`

737,740c739,745	`.loginform td #submit:hover{`
	`border: 1px solid #888;`
	`background-color: #eee;`
	`}`
		`/*Buttons dont have hover anymore`
		`.loginform td #submit:hover{`
		`border: 1px solid #888;`
		`background-color: #eee;`
		`}`
		`*/`

tool.css

How to proceed? I would edit your 2.3 *tool.css* – search the new selectors below in a 2.4 and move them over to the 2.3

IMPORTANT: Resources tool was redone and many new selector were added. Copy the whole “makeMenu” family to your css and modify.

A series of changes to some basic elements that the DG-UI felt were needed for usability. Since the changes were to general elements and there was little review of the general impact YMMV.

Links – essentially underline all links everywhere as you can see

```
/*SECTION 2 LINKS*/
a:link{
        color: #35b;
        text-decoration: underline !important;
}

a:visited{
        color: #53b;
        text-decoration: underline !important;
}
```

Headers –adjusting

```
/*SECTION 6 HIERARCHY*/

/*headers are used in many places for structuring things semantically - these are the plain headers - below there are others
contextually defined*/
h2,h6{
        color: #555;
        padding: 0;
        font-weight: bold;
        font-family: Arial,Helvetica,sans-serif;
        margin: 1em 0;
        background-color: transparent !important;
}


h2{
        font-size: 130%;
}

h3{
        font-size: 110%;
}

h4,h5,h6 {
        font-size: 100%;
}
h3,h4,h5{
        color: #555;
        padding: 0;
        font-weight: bold;
        font-family: Arial,Helvetica,sans-serif;
        margin: 1em 0 0 0;
        background-color: transparent !important;
}
```

New needs from new tools

```
/*highlight a row representing an item that has been added*/
tr.highLightAdded{
        background:#afa;
}
```

Note: Chat has been totally redone – the look is the same but some new selectors were needed. Search in the 2.4 file for “chat”
Note: The last 2 sections in *tool.css* are new. One is intended for *mneme* – a new test taking engine, the other is crucial for Resources.

Widget refinements

Some definition for separators and inactive elements in the toolbar

```
.navIntraTool span.separator {
        left: -.5em;
        position: relative;
}
.navIntraTool span.inactive {
        margin-right:1em;
        white-space: nowrap;
        display:inline-block
}
```

The submit notification text used a mozilla only feature for blinking – substituted a blicking icon instead

```
#submitnotif{
        background: #fff url(../images/warn-a.gif) .3em .3em no-repeat;
        border: 1px solid #b11;
        font-family: Verdana,Arial,Helvetica,sans-serif;
        clear: none;
        color: #b11;
        font-size: small;
        vertical-align: text-top;
        margin: 0;
        padding: 5px 5px 5px 25px;
}
```

A disabled button that is classed as “active” needs to be made to look disabled

```
.act .active[disabled="disabled"] {
        color:gray
}
```

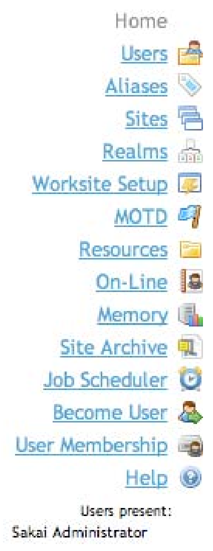
Appendix K – Sakai 2.5 modifications

The default look and feel of Sakai changed in three ways for Sakai 2.5

- The addition of icons for all the core tools
- Restyling the default Site tabs to appear as literal tabs
- The optional presentation of the More Site dropdown as an Pop-up window organized by Site type

The addition of icons for all the core tools

The Tool navigation bar now contains icons* for all core tools.



The icons are displayed as background images and are defined towards the end of the portal.css file, under **`/* PART 9 - Tool Icons */`**.

Example:

```
.icon-sakai-forums{
  background-image: url(../../image/silk/comments.png );
}

.icon-sakai-gradebook-tool{
  background-image: url(../../image/silk/report.png);
}

.icon-sakai-mailtool {
  background-image: url(../../image/silk/email_go.png );
}
```

How icons are associated with tools:

Every tool in the tool navigation has a tool icon class attribute added to it whether or not a tool icon class has been defined for that tool in the portal.css.

```
<li>
   <a class="icon-sakai-sitesetup" href="…"><span>Worksite Setup</span></a>
</li>
```

The tool icon class name is derived from the tool id by replacing any dots in the tool id [**`sakai.sitesetup`**] with a dashes [**`sakai-sitesetup`**] and prepending “icon-” [**`icon-sakai-sitesetup`**].

How to add an icon to a tool that lacks one:

Find or create a image for your tool approximately 16 by 16 pixels square in either gif or png format and add that image to the images directory for your skin, or select an image from the default icon images in `/reference/library/src/webapp/image/silk/`.

If your image has different dimensions you may need to modify the tool item css [`#toolMenu li a:link, #toolMenu li a`] to get your image to display correctly.

Define the tool icon class for your tool in portal.css, using the pattern above and add the path for the icon you’ve choosen in the definition of the background-image attribute.

Redeploy your instance of Sakai.

How to change a the icon assigned to a tool:

Use the same steps as indicated above in *How to add an icon to a tool that lacks one*, except instead of adding a new tool classname to the portal css, simply redefine the background-image attribute for the existing tool.

How to remove an icon from a tool:

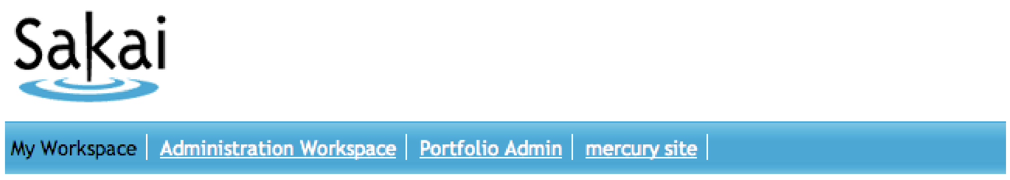
Simply delete or comment out the tool class name for the tool that you wish to appear with out an icon in portal.css.

How to remove all the icons from the tool area:

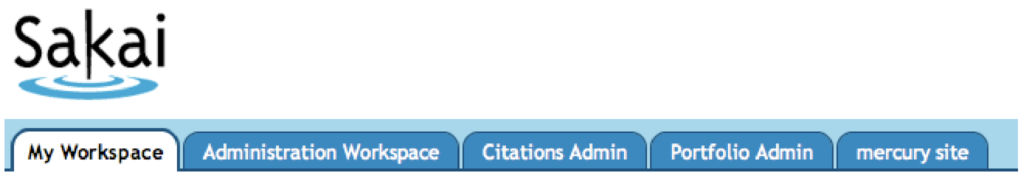
Delete or comment out all the tool class names in portal.css. You may also want to modify the tool item css to make use of the area that was taken up by the tool icon [`#toolMenu li a:link`, `#toolMenu li a`].

* Sakai's tool icons have been selected from the excellent Silk icon family by Mark James. The entire Silk family is available for use in Sakai.

Restyling the default Site tabs to appear as literal tabs



2.4 Default Site Tabs



2.5 Default Site Tabs

Sakai 2.5 sports literal tabs instead of the more figurative tabs of Sakai 2.4.

> Literal tabs are just one of many ways to style the tabs in Sakai. The HTML used for the site tabs is very flexible and can support any tab design, literal or figurative, that you might come up with.
>
> For more tab ideas, check out how other institutions have styled Sakai:
>
> http://confluence.sakaiproject.org/confluence/display/UI/Sakai+screenshots

The new Sakai tab style is achieved using the Douglas Bowman's *Sliding Doors of CSS* technique from *A List Apart*.

How to change to the look of the default site tabs:

Before restyling these tabs, you might want to review the A List Apart article to better understand the technique.

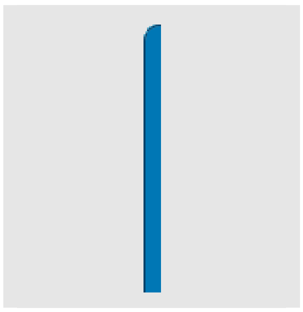
http://www.alistapart.com/articles/slidingdoors/

As you can see above, there are two different kinds of tabs, unselected tabs and selected tabs. There is also a third type as well, which you'll only need if you've enabled the More Sites option, which we'll cover in the next section.

The HTML mark-up which defines the tabs didn't change from Sakai 2.4 to 2.5, what changed was the CSS and the images. Each tab is still defined by the following markup in the **`<ul id="siteLinkList">`** element:

```
<li>
  <a href="…" title="My Workspace"><span>My Workspace</span></a>
</li>
```

The basic trick is that the tab is composed of two images that slide past each other to create a single variable-width image.


Left side of the tab

Right side of the tab

The left part of the tab image, `tab-left.gif`, is attached to ***a*** tag and the right part, `tab-right.gif`, is attached to the ***span*** tag.

```
#siteLinkList li a {
      background:transparent url(images/tab-left.gif) no-repeat scroll left top;
      float:left;
      …
}
#siteLinkList li span {
      background:transparent url(images/tab-right.gif) no-repeat scroll right top;
      …
}
```

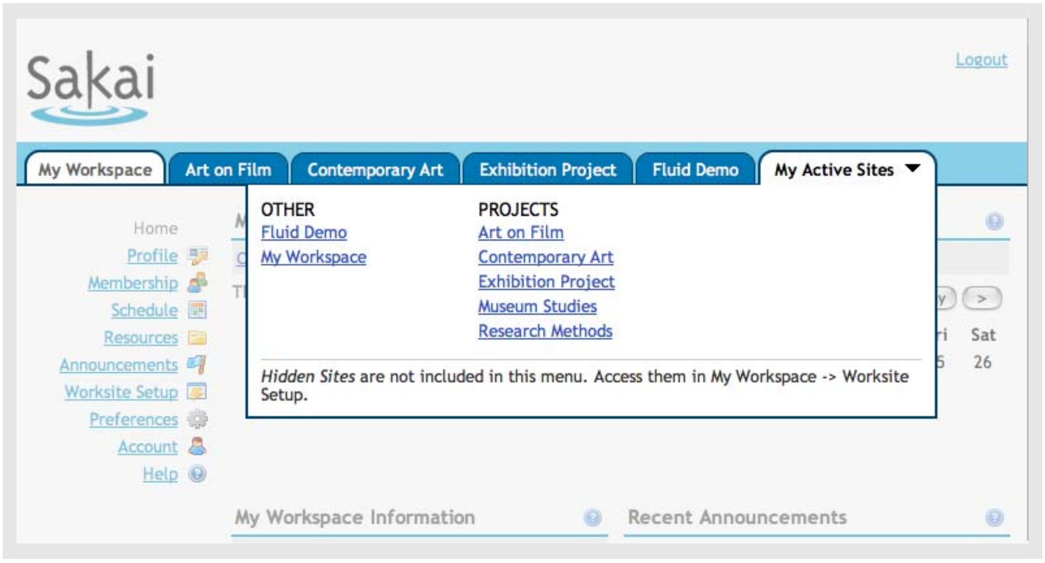
Image templates to assist in creating your own tabs are available in both Photoshop and Fireworks formats at:

https://source.sakaiproject.org/svn//reference/trunk/docs/image-templates/sakai-2.5-site-tabs/

More Sites DHTML Drop-down

Sakai 2.5 introduces a different way to display the *active sites* that don't fit in the currently visible tabs. In Sakai 2.4 these sites where displayed a drop-down menu. The optional More Sites enhancement organizes all the users active sites by site type and term and displays them in a DHTML drop-down accessed through an additional tab.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.



Hidden Sites are not included in this menu. Access them in My Workspace -> Worksite Setup.

More Sites Drop-down Menu

Optional *My Active Sites* DHTML Drop-down in 2.5

To turn on the More Sites option in Sakai, add the following to `sakai.properties`:

```
# turn on More Sites
portal.use.dhtml.more=true
```

To customize the look of the More Sites DHTML drop-down, edit the CSS in `portal.css`.

Please note that just looking at the rendered HTML will not give you a full picture of what is going on with the drop-down. For example, the sizing and positioning of the dropdown is done dynamically using Javascript in `portalscripts.js`.

All the DHTML Drop-down styles are proceeded by the class `.dhtml_more_tabs`.

The primary styling of the More Sites drop-down is a bold, blue, three-sided box defined thusly in this CSS markup:

```
.dhtml_more_tabs {
	position: absolute;
	top: 2.5em;
	right: 4px;
	line-height: 1.5em;
	background-color: #FFFFFF;
	border: 2px solid #013F68;
	border-top-width: 0;
	width: 75%;
	z-index: 9999;
}
```

The other `.dhtml_more_tabs` styles define the formatting and layout of the elements within the box.

Two classes used by the DHTML Drop-down not proceeded by`.dhtml_more_tabs`, are `#more_tabs_instr` and `#portalMask`, used respectively to format the help content at the bottom of the More Sites drop-down and create the mask over the portal when the drop-down is displaying.

```
#more_tabs_instr  {
   color: #333;
   font-size: 1em;
   margin: 0.4em 0.8em 0.8em;
   padding-top: 0.4em;
   clear: both;
   border-top: 1px solid #CCC;
   text-align: left;
   line-height: 1.1em;
}
```

```
#portalMask {
   background-color: #EEEEEE;
   position: absolute;
   z-index: 9000;
   top:0;
   left:0;
   filter:alpha(opacity=50);
   opacity: 0.50;
   -moz-opacity: 0.50;
}
```