

## The Sakai WorkGroup Portal

Charles Severance

DRAFT 1.0 - comments to csev@sakaifoundation.org

December 31, 2006

### Introduction

The basic idea of the Sakai workgroup portal capability is to allow Sakai's tools and content management capabilities to be used to produce a basic public web site for an organization such as Sakai. This is a new feature for Sakai 2.4.

This is called the "workgroup portal" because it is not expected to be usable as a one-stop shopping portal for a medium to large enterprise. This is because Sakai effectively is a groupware system, and its basic organizing principle is "sites" and groups of users with roles in those sites. All of Sakai's workflow and user interfaces are centered on these basic business objects.

To create the workgroup portal, we simply make it convenient to permit a set of sites to anonymous users and give anonymous users roles and permissions (like content.read) in those sites. Enterprise portals think in terms of channels, pushed content, enterprise notifications, and the workflow of enterprise portals is wrapped completely around content and the enterprise workflow is designed around creating and publishing, and presenting content.

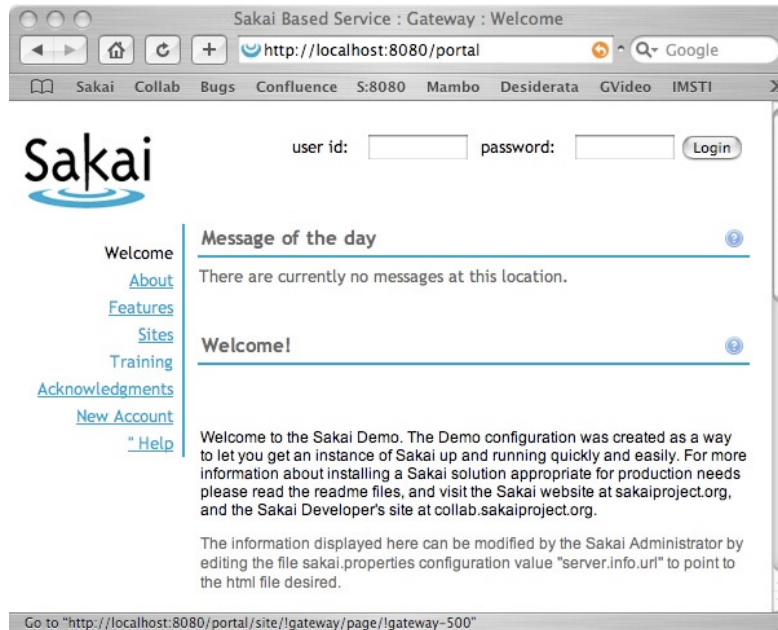
Sakai has always had a "workgroup portal" in the of the box configuration. When the Sakai portal/aggregator (Charon) detects that a user is not logged in Charon displays a site configured in **sakai.properties** which defaults to **!gateway**. Displaying the gateway site will continue to be the default for Sakai 2.4. To use the new workgroup portal features, a Sakai administrator must enable the new feature through new properties.

A key effect of these new features will be to allow Google to increasingly search content within Sakai that is stored in these anonymously viewable sites. It will take some further effort for tools to re-factor themselves to provide nice URLs that reference objects like mail messages directly. All resource content already does have nice URLs and the existing Sakai tools will get a user \*close\* to the data element that the search has found.

The remainder of this document describes Sakai's portal support for anonymous users.

### The Pre-2.4 Capabilities

Most developers and users of Sakai are familiar with the Sakai login screen:



This is simply a Sakai site that is specified by **sakai.properties**:

```
# gateway site id
gatewaySiteId=!gateway
```

This site is generally edited by the **admin** user using the **Sites** tools - an important key when working with the gateway site is to remember that not all Sakai tools present a clean user interface for anonymous users. So there are several tools that are "safe" to use on the gateway:

```
sakai.motd
sakai.iframe
sakai.iframe.service
sakai.sitebrowser
sakai.createusers
```

Wise administrators simply limited themselves to these tools and the gateways functioned to provide basic capabilities - a common approach is to use **Resources** to upload an HTML file, mark the resource as publicly viewable and then place an iframe on the **!gateway** site to place content.

If you were clever there are additional things that can be done but we will leave the slick stuff to the next section.

### Required Functions for Tools

A critical feature for the workgroup portal was introduced in Sakai 2.3 as part of a requirement (REQ-#####). Sakai 2.3 allows a tool (and its associated page) to

appear or disappear based on whether a user has a particular security function or not.

This is controlled by a new property on the tool called **functions.require** which is part of the tool registration.

```
<registration>
  <tool    id="sakai.rutgers.linktool"
          title="Link Tool"
          description="A tool to link to external applications.">

    <category name="course" />
    <category name="project" />

    <!-- Allow this to be set -->
    <configuration name="functions.require" />
  </tool>
</registration>
```

In Sakai 2.3 there were very few tools that made use of this capability - for the Rutgers link tool (above) the key was to allow a tool (and its page) to be suppressed for students if it was a link to a campus-wide grading system. In this example the tool registration does *not* specify which function was required to include the tool in the buttons on the left hand side of the Sakai display. For the link tool, often a function like **site.upd** was used as an indicator of highly powered users - any function will do - it does not have to be a function for the tool itself. Another possible function might be **content.new** if this function were only given to **instructor** or **maintain** roles.

In Sakai 2.4 required functions were added to a number of tools to make them suitable for use in sites open to anonymous users. The following were added:

- sakai.announcements and sakai.synoptic.announcement require **annc.read**
- sakai.mailbox (Email Archive) requires **mail.read**
- sakai.schedule (Calendar) and sakai.summary.calendar require **calendar.read**
- sakai.chat and sakai.synoptic.chat requires **chat.read**
- sakai.discussion and sakai.synoptic.discussion require **disc.read**
- sakai.resources requires **content.read** (this needs to be thought through and validated)
- sakai.rwiki requires **rwiki.read**
- sakai.messagecenter and sakai.synoptic.messagecenter require **messagecenter.read**

Several tools work fine for anonymous users so they do not need a particular function to be required to function properly - but like link tool we add the property

so that site admins can set functions to hide the tools by adding a function. This is done for: sakai.iframe, sakai.sitebrowser, and sakai.motd.

\*\*\*\*\* Several need to be thought through: sakai.resources and sakai.news and assignments.

As we explore the use of these features, remember that the key value of **functions.require** is to make tools disappear when a user is missing a permission. So this allows a site to have many tools that are all there when people are logged in and then have those buttons disappear for anonymous users or for users with or without a particular role.

### The .anon and .auth Roles

There are two particular hard-coded roles that have been in Sakai since 1.0 - these are:

- .auth - This allows the assignment of permissions (like content.read) to any authorized user (i.e. after they log in)
- .anon - This allows the assignment of permissions to any anonymous user

If you use the **Realms** tool as the admin user you can see that the **!gateway** site gives only **site.visit** to both .anon and .auth users.

#### Users

These are the Roles granted to individual users in this Realm. Click on an Id to edit.

No Roles have been granted to any users.

Complete the Realm Edit

#### Roles

These are the Roles defined within this Realm. Click on an Id to edit.

Role Id	Description
<a href="#">.anon</a>	
<a href="#">.auth</a>	

This is why you are never presented with a login screen when the gateway site is displayed. Other sites like lerror are also set up with this permission.

These .anon and .auth roles can be used in any site and given any combination of permissions including write/modify permissions in order to allow either anonymous or all logged in users to perform certain operations in the site.

This coupled with the **functions.require** capability and the ability to have multiple gateway sites in Sakai 2.4 allows a relatively high-functionality and flexible anonymous user portal to be deployed.

## New Capabilities

The primary new feature in Sakai 2.4 is to allow the Sakai deployment to specify multiple gateway sites in **sakai.properties**:

```
# gateway site list - comma separated values indicating site IDs -
# these sites must include the .anon role and at a minimum give
# the .anon role site.visit - and any other permissions you want
# (usually *.read) permissions to the .anon role.
# if the .anon role does not have site.visit - these sites will not
# appear in the gateway - even if they are in this list.
# in the example below, mercury does *not* have site.visit
# for .anon so it does not appear
# if this is not set, the portal simply displays a single gateway site
# specified by gatewaySiteId
# gatewaySiteList=!gateway,mercury

# The number of tabs to display before adding the "More" dropdown
# the default for this is four tabs
# gatewaySiteListDisplayCount=99
```

The **gatewaySiteList** specifies both which sites are shown as the gateway as well as the order of the tabs. There were several design alternatives to configuring the gateway sites which were rejected which are discussed below.

The portal will not display a site to anonymous users in this list *unless* the site grants **site.visit** to the .anon role.

## Configuring Workgroup Portal

To configure the anonymous portal in Sakai, you create the desired sites. It is often more convenient to create the sites in the administrator's **Sites** tool rather than worksite setup because you can set the actual site Id and have something more convenient than a long GUID.

You need to also configure the realm associated with the site using the realm tool - if you used the Sites tool to make the site - you can find the realm quickly using the siteId. Once in the Realm, add the .anon role and assign the role permissions - make sure to include **site.visit**. Also assign the rest of the permission functions (chat.read, content.read, rwiki.read, etc) for the tools that you will add to the site.

Remember that if you add a tool to the site and do not give a permission to the .anon user, it will appear normally for the **access** or **maintain** users but not anonymous users. This provides nice flexibility for the site owner - as an example, you might use the wiki to edit web pages for a site, but only display those pages with the iframe tool. You can put in the wiki tool but *not* grant rwiki.read to .anon users - this way regular members of the site will be able to work with wiki and the anonymous users will never see the wiki tool but the anonymous users will see the iframe tools that point to wiki pages.

A special case is the **Site Setup** tool. Lets assume that you want the normal users in the site to see **Site Setup** but not anonymous users. We must give anonymous users **site.visit** because they need to see the site. So you cannot use **site.visit** as the **functions.require** to cause the **Site Setup** tool to be hidden for anonymous users. Much like the link tool example, simply pick another function like **site.upd** or **content.new** that access/maintain users will have but anonymous users will not have.

#### Tool Configuration

Title:	<input type="text" value="Site Info"/>
LayoutHints	<input type="text"/>

row,column (zero-based)

#### Configuration

defaultSiteType:	<input type="text" value="project"/>
privateSiteTypes:	<input type="text"/>
functions.require:	<input type="text" value="site.upd"/>
publicChangeableSiteTypes:	<input type="text" value="project"/>
publicSiteTypes:	<input type="text" value="course"/>
siteTypes:	<input type="text" value="course,project"/>

You can apply this approach to any tool placement because the function does not have to be directly related to the tool. Note that if the tool registration does not mention **functions.require**, you cannot set this property - to solve this simply add an entry in the tool registration (like in the linktool above) with no value to let the value be set in the admin interface.

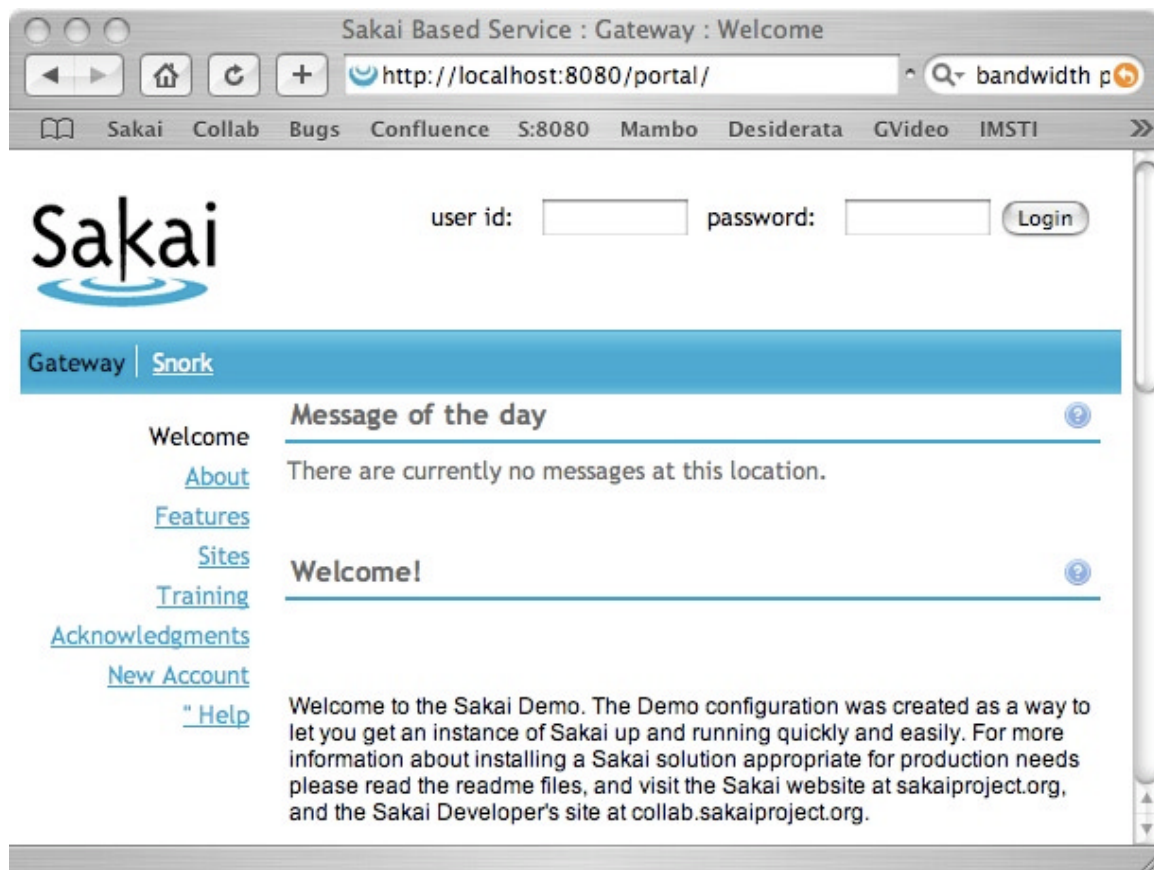
You also have to set up the site list in the **sakai.properties**:

```
# sakai.properties - the demo edition
# Testing 123
gatewaySiteListDisplayCount=99
```

```
gatewaySiteList=!gateway,snork,zorch
```

You do *not* have to include the gateway site - it is just convenient to use for testing because it is already there - in a normal situation you would likely not use the gateway site. Also remember that if the site does not have **site.visit** for .anon users it will be ignored. If you set this property and none of the sites have **site.visit** - your page will look weird and a warning message will be issued.

If all goes well, you can start up your system and see multiple sites on the gateway page.



In this example, the site **zorch** does not yet exist - this is a non-fatal error that is not logged. This allows you to set up the list of site ids in advance and then create, configure, and test your sites dynamically by simply logging in and out or using two browsers.

### New Horizontal Buttons Skin

Depending on the end user experience you are looking for, you may or may not want the default skin for these gateway sites - you probably want to have the same skin for all gateway sites.

If the gateway site is mostly for people who will also log into Sakai you might want to keep the default "tabs and columns" layout for the gateway sites to simplify user-experience.

But most workgroup portals use an approach that is two layers of tabs rather than tabs and columns - this also allows more of the horizontal space to be used for content particularly when multiple columns of tools are being used.

A new skin is added for Sakai 2.4 called **default-horiz** that is like the default skin (named **default**) except with two layer buttons. This can be set in the Sites tool or in Site Setup:

Description:	<input type="text" value="The Gateway Site"/>
Published:	<input type="radio"/> unpublished <input checked="" type="radio"/> published
Joinable?:	<input type="radio"/> yes <input checked="" type="radio"/> no
Role to assign to joiners:	<input type="text"/>
Worksite Icon URL:	<input type="text"/>
Worksite Info URL:	<input type="text"/>
Skin:	<input type="text" value="default-horiz"/>
Public View?:	<input type="radio"/> yes <input checked="" type="radio"/> no
Custom Page Order?:	<input checked="" type="radio"/> yes <input type="radio"/> no

You also will likely want to set the **Custom Page Order** for the site as well to give you flexibility in the button ordering.

Once this is set your gateway site will look as follows:



Sakai Based Service : Snork : Announcements

http://localhost:8080/portal/site/snork/page/14

Sakai Collab Bugs Confluence S:8080 Mambo Desiderata GVideo IMSTI Pluto GS

**Sakai** user id:  password:

Gateway | Snork

Announcements [Wiki](#) [Help](#)

**Announcements**

View  Viewing 1 - 1 of 1 items

Subject	From	For	Date
<a href="#">KJKHHJK</a>	Sakai Administrator	site	Dec 28, 2006 1:09 pm

Default

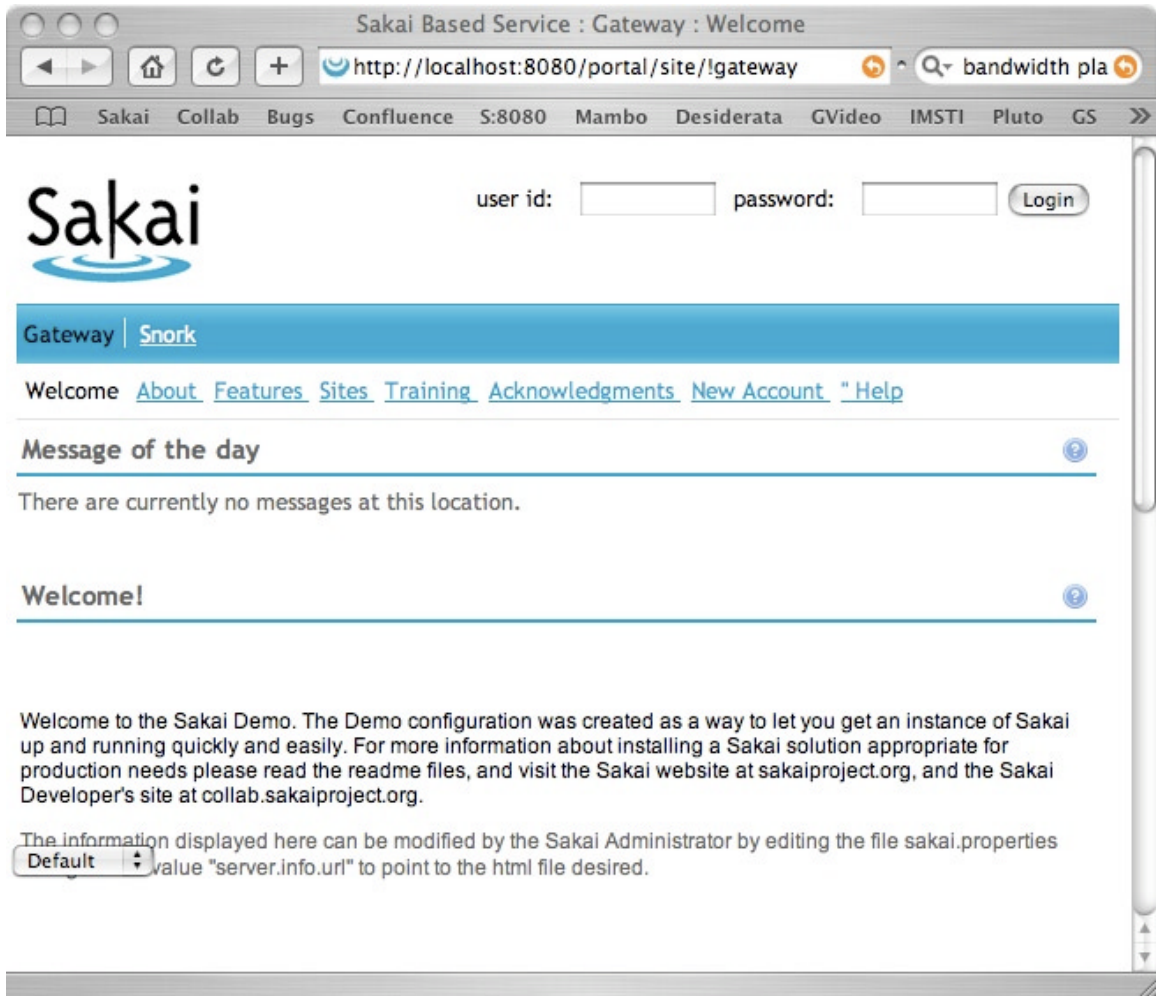
Powered by Sakai

Copyright 2003-2006 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.  
Sakai Based Service - dev - Sakai dev - Server charles-severances-computer.local

The tool pages are now horizontal just below the site tabs. This provides a more "portal-like" navigation style that some may prefer. The presence is suppressed in this view as there is no logical place to put it at this time.

Ignore the "Default" drop-down - that is just there for testing.

By setting the skin to be the same on all of the gateway sites, the navigation is consistent from site to site.



## Design Considerations

There were several alternative designs considered but rejected - in the future we may decide that some of these make sense but here I just capture thoughts/notes.

Simply display all sites that grant **.anon site.visit**

This was rejected because any site can do this and the gateway needs to be under the strict control of the Sakai administrator.

Simply display all sites that are marked as **Public View**

This was rejected because any site can do this and the gateway needs to be under the strict control of the Sakai administrator.

Create a user called **gateway** and display whatever sites that user is enrolled in in the order specified in that user's preferences

This is less convenient than you might think - we have to add gateway user to each site and give them the right roles. Setting the effective user id to gateway might open holes up. We need to suppress the "My Workspace" for the gateway user. Much of the display code detects if a log in is necessary by noting that there is no user - setting effective ID to gateway might break some of this code and/or introduce security holes - particularly if people can "make" the gateway user.

So all in all, while this properties approach seems a little clunky - it assures 100% that the only way to mess with the gateway is to edit sakai.properties.

## Looking Forward/TODO

Show list of things to consider/do

We will need a way to get required functions for JSR-168 portlets - they register differently - but the simple thing is to simply put in a blank **functions.require** for each portlet registration and let the admin set functions on a placement by placement.

Allow an or capability in addition to an and capability. Here is a possible syntax:

```
or,content.read,chat.read  
and,content.read,chat.read
```

If the string "or" is encountered the evaluation switches to an "or" evaluation and the evaluation stops as soon as a "true" is detected. This is processed left to right (i.e. no parenthesis). And is implied by default. You can combine this to some degree by carefully ordering the functions. For example:

```
and,content.read,chat.read,or,site.upd,content.create
```

Means that if you have content.read and chat.read - you are golden. If you are missing one or the other, then site.update or content.create gives you the "green light".

Something to think about. This was part of the original **functions.require** requirement and perhaps it should be done by Sakai 2.4.

Another general area will be to go through tools and make them suitable for use on the gateway page. One set of tasks is to simply put tools on the page and see if they function and file JIRAs if they do not. As an example, Wikis does not check wiki.edit before putting up the "Edit" button - if you press Edit without having wiki.edit, wiki does not allow you to edit - but it would be nicer if the button was simply not there.

Another set of tasks is to build/change tools so that they function well on the gateway - as an example - make sure that a poll tool does not introduce security holes even when we give .anon a poll.participate role.

There also might be a set of new tools that are written paying particular attention to the anonymous portal like games, publishing, etc.

## **Conclusion**

This is the start of a work in progress aimed at supporting a workgroup's needs to create and publish a basic public web site using Sakai as their content management system. It is not intended nor expected to compete with real content management systems or real enterprise portals.

As we gain experience with this approach it can evolve to meet our needs.

A general to-do is to redesign the overall navigation of the portal looking at things like breadcrumbs, metatabs like "Courses", "Projects", etc. Also the portal is moving to a Velocity based markup generation. We might also want to look into an XSLT based skin as well.