



Blogger – Multiple Instances

1.0 Contents

1.0	Contents	2
2.0	Introduction	3
2.1	About the Blogger	3
2.2	Additional Notes	4
3.0	Configuration	5
3.1	Creating a New Blog Instance.....	5
3.1.1	Copy the directory	5
3.1.2	Move the directory	5
3.1.3	Create blogger data directory	5
3.1.4	Set permissions for blogger directories	5
3.1.5	Modify blog instance properties	5
3.1.6	Modify blojsom instance properties	5
3.1.7	Modify configuration for Academus integration	5
3.1.8	Copy AcademusApps sample configuration for blogger integration.....	6
3.1.9	Modify new AcademusApps configuration for blogger integration	6
3.1.10	Modify AcademusApps permissions XML for blogger permissions.....	6
3.1.11	Modify AcademusApps Portlet XML	6
3.1.12	Modify AcademusApps Web XML	7
3.1.13	Restart tomcat.....	8
3.1.14	Add blogger instance permissions	8
3.1.15	Create new channel publication.....	8

2.0 Introduction

This document outlines the required steps to publish multiple Blojsom instances within Academus.

2.1 About the Blogger

What is Blojsom?

Blojsom is...

A Java-based, full-featured, multi-blog, multi-user software package that was inspired by Bloxom (www.bloxom.com). Blojsom aims to retain simplicity in design while adding flexibility in areas such as the flavors, templating, plug-ins, and the ability to run multiple blogs with a single Blojsom installation.

For further information about Blojsom visit
<http://wiki.blojsom.com/wiki/display/blojsom/About%2Bblojsom>.

Why Blojsom?

Currently, Blojsom appears to be the most flexible and cleanest solution for a blogger to be packaged with the Academus product for the following reasons:

1. It is free BSD.
2. Documentation is sufficient and installation and implementation is quick and simple.
3. Pluggable architecture at all levels of the system including, data storage, authorization, authentication, and content rendering.
4. Supports chaining of plug-in functionality which allows components to interact in a variety of ways. This also allows for event handling and notifications at any point of the process to be quickly plugged in depending upon client needs.
5. System is very lightweight that is easy to understand with extensible and highly configurable architecture.
6. Ships with plug-in for LDAP authentication.
7. Due to the flexibility of the architecture, implementing SSO requires minimal effort.
8. Supports the most common and flexible templating languages for rendering right out of the box (e.g. JSP, Velocity, Freemarker, Groovy) and provides a straight forward way of adding new templating technologies as needed via plug-in.
9. Supports the most common Blogger APIs and formats, including MovableType API, for blogger client integration and data syndication right out of the box.
10. Default install and usage does not require a database for use but other types of data storage mechanisms can be integrated via plug-in.
11. Most common functionality of bloggers is already implemented and has a simple and easy to understand interface and workflow.

12. This product is used by various corporations such as, Apple, Cisco, JBoss, Adobe, Hewlett-Packard, InfoMinder, IRIS, OpenMLS, etc... Further, this product is also used by a number of Universities and is gaining popularity. In viewing the state and functionality of this product it is apparent that Blojsom is full featured and has a simple but well thought out flexible design which has already been through appropriate testing and adjustments.

2.2 Additional Notes

<NAME> is used to represent the desired short name of your new blog instance. This should not contain any spaces or special characters.

<TOMCAT_HOME> is used to represent the location of Apache Tomcat on your system. This is the directory that contains the 'webapps' subdirectory.

<PORTAL_DATA> is used to represent the portal data directory.

3.0 Configuration

3.1 Creating a New Blog Instance

The following steps can be used to create a new blogger instance:

3.1.1 Copy the directory

Copy `<TOMCAT_HOME>/webapps/blojsom/WEB-INF/bootstrap` to
`<TOMCAT_HOME>/webapps/blojsom/WEB-INF/<NAME>`

3.1.2 Move the directory

Move `<TOMCAT_HOME>/webapps/blojsom/WEB-INF/<NAME>/resources` to become
`<TOMCAT_HOME>/webapps/blojsom/resources/<NAME>`

3.1.3 Create blogger data directory

Create `<PORTAL_DATA>/blojsom/<NAME>`

3.1.4 Set permissions for blogger directories

Recursively change the owner (if needed) to the user that runs Academus on the following directories

```
<TOMCAT_HOME>/webapps/blojsom/resources/<NAME>  
<PORTAL_DATA>/blojsom/<NAME>  
<TOMCAT_HOME>/webapps/blojsom/WEB-INF/<NAME>
```

3.1.5 Modify blog instance properties

Modify the file `<TOMCAT_HOME>/webapps/blojsom/WEB-INF/<NAME>/blog.properties` and set/add the following properties:

```
blog-home -- This should be set to <PORTAL_DATA>/blojsom/<NAME>  
blog-base-url -- This should be the full URL to blojsom.  
blog-url -- This should be the full URL to blojsom, with /blog/<NAME>' appended.
```

3.1.6 Modify blojsom instance properties

Modify the file `<TOMCAT_HOME>/webapps/blojsom/WEB-INF/blojsom.properties` and change the property 'blojsom-users' to include `<NAME>`. This is a comma-separated list.

3.1.7 Modify configuration for Academus integration

Modify the file `<TOMCAT_HOME>/webapps/blojsom/WEB-INF/academus_config.xml` to include a new access-broker entry:

Replace all instances of `<NAME>` appropriately.

```
<access-broker handle="blog-<NAME>"  
    impl="net.unicon.alchemist.access.permissions.PermissionsAccessBroker">  
  <access impl="net.unicon.academus.apps.blogger.BlojsomAccessType"/>  
  <permissions>  
    <access-broker handle="blog-<NAME>-p"  
      impl="net.unicon.alchemist.access.rdbms.RdbmsAccessBroker"  
      needsDataSource="true">  
      <access impl="net.unicon.academus.apps.blogger.BlojsomAccessType"/>  
    </access-broker>  
  </permissions>
```

```
</access-broker>
```

3.1.8 Copy AcademusApps sample configuration for blogger integration

Copy the file:

```
<TOMCAT_HOME>/webapps/AcademusApps/WEB-INF/classes/config/blogs/blog-EXAMPLE.xml
```

to

```
<TOMCAT_HOME>/webapps/AcademusApps/WEB-INF/classes/config/blogs/blog-<NAME>.xml
```

Make sure you copy the latest blog-EXAMPLE.xml. Do not try to re-use an older blog-EXAMPLE.xml as the configuration file format may have been changed since.

3.1.9 Modify new AcademusApps configuration for blogger integration

Modify the file you just copied, <TOMCAT_HOME>/webapps/AcademusApps/WEB-INF/classes/config/blogs/**blog-<NAME>.xml**

Replace all occurrences of the token EXAMPLE with <NAME>.

3.1.10 Modify AcademusApps permissions XML for blogger permissions

Modify the file <TOMCAT_HOME>/webapps/AcademusApps/WEB-INF/classes/config/**permissions-portlet.xml** to include a new portlet entry:

Replace all instances of <NAME> appropriately.

```
<portlet handle="blog-<NAME>">
  <label>Blog: <NAME></label>
  <description>Modify permissions for the blog instance: <NAME></description>
  <portlet-access handle="access-blog-<NAME>">
    <import src="/config/blogs/blog-NAME.xml"
      select="//access-broker[@handle='blog-<NAME>']"/>
    <access impl="net.unicon.academus.apps.blogger.BlojsomAccessType"/>
  </portlet-access>
</portlet>
```

3.1.11 Modify AcademusApps Portlet XML

Modify the file <TOMCAT_HOME>/webapps/AcademusApps/WEB-INF/**portlet.xml** to include a new portlet definition:

Replace all instances of <NAME> appropriately.

```
<portlet>
  <portlet-name>Blog<NAME></portlet-name>
  <portlet-class>net.unicon.academus.apps.gateway.GatewayPortlet</portlet-class>
  <init-param>
    <name>Id</name>
    <value>blog-<NAME></value>
  </init-param>
  <init-param>
    <name>configPath</name>
    <value>/WEB-INF/classes/config/blogs/blog-<NAME>.xml</value>
  </init-param>
  <expiration-cache>0</expiration-cache>
  <supported-locale>en-US</supported-locale>
  <portlet-info>
```

```
<title>Blog: <NAME></title>  
<short-title><NAME></short-title>  
<keywords>Blog</keywords>  
</portlet-info>  
</portlet>
```

Make sure the value in the tag:

```
<portlet-name>Blog<NAME></portlet-name>
```

Matches the value after the “AcademusApps” in web.xml for the tag:

```
<param-value>AcademusApps.Blog<NAME></param-value>
```

3.1.12 Modify AcademusApps Web XML

Modify the file `<TOMCAT_HOME>/webapps/AcademusApps/WEB-INF/web.xml` to include a new servlet definition and mapping:

Replace all instances of `<NAME>` appropriately.

```
<servlet>  
  <servlet-name>Blog<NAME></servlet-name>  
  <display-name>Blog<NAME> Wrapper</display-name>  
  <description>Automated generated Portlet Wrapper</description>  
  <servlet-class>org.apache.pluto.core.PortletServlet</servlet-class>  
  <init-param>  
    <param-name>portlet-class</param-name>  
    <param-value>net.unicon.academus.apps.gateway.GatewayPortlet</param-value>  
  </init-param>  
  <init-param>  
    <param-name>portlet-guid</param-name>  
    <param-value>AcademusApps.Blog<NAME></param-value>  
  </init-param>  
</servlet>  
  
<servlet-mapping>  
  <servlet-name>Blog<NAME></servlet-name>  
  <url-pattern>/Blog<NAME>/*</url-pattern>  
</servlet-mapping>
```

Make sure the value after the “AcademusApps” in for the tag:

```
<param-value>AcademusApps.Blog<NAME></param-value>
```

Matches the value in portlet.xml for the tag:

```
<portlet-name>Blog<NAME></portlet-name>
```

3.1.13 Restart tomcat

3.1.14 Add blogger instance permissions

Log in as an administrator and go to the Portlet Permissions Manager. Add new permissions for the new blog instance for the desired user.

3.1.15 Create new channel publication

Create a new channel publication with the following parameters:

Channel Type:	Portlet
Channel Title:	Blog: <NAME>
Channel Name:	Blog: <NAME>
Channel Functional Name:	BlojsomGatewayPortlet
Channel Description:	Blog: <NAME>
Channel Timeout:	60000ms
Channel Secure:	false
Portlet Definition ID:	AcademusApps.Blog<NAME>
Portlet Preferences:	None
Category:	At your discretion
Group:	At your discretion

The Channel Functional Name should always be left as “BlojsomGatewayPortlet”. This allows for the Academus help system to correctly identify the channel and provide blogging assistance to your users.

Place the new channel in a user's layout as desired.