

A paper in the InfoGlue Series

InfoGlue - Administrative Manual

Author: Mattias Bogeblad
Version 2.9.6

© 2006 Formedia and Mattias Bogeblad. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of The GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled "GNU Free Documentation License".

InfoGlue – Administrative Manual

Sponsors

We wish to thank Nominet UK (especially Jay Daley) for sponsoring the creation, maintenance and release of this document to the community. You can read more about Nominet UK on www.nominet.org.uk.

Note

All information in this document is the property of Formedia and Mattias Bogebblad. Use and distribution of the document, or of the information in it, is defined by the terms in the license "GNU Free Documentation License".

InfoGlue – Administrative Manual

INTRODUCTION.....	5
FEEDBACK	5
GENERAL CONCEPTS	6
<i>What is InfoGlue.....</i>	6
<i>What parts are there in InfoGlue.....</i>	6
INSTALLATION.....	7
STEP 1: INSTALLING JAVA.....	7
STEP 2: INSTALLING APACHE TOMCAT	7
STEP 3: INSTALLING MYSQL	7
STEP 4: INSTALLING INFOGLUE	8
DETAILED DESCRIPTIONS OF THE FIELDS IN THE INSTALLER / UPGRADER:	9
ALTERNATIVE DEPLOY CONFIGURATION.....	13
ADDING APACHE HTTP-SERVER AS A FRONT-END	17
APPLICATION SETTINGS	19
<i>Common settings.....</i>	20
<i>Cache settings.....</i>	22
<i>CMS Connection</i>	25
<i>Webserver settings</i>	26
<i>Encoding settings.....</i>	27
<i>Mail settings</i>	28
<i>Protection settings part 1.....</i>	29
<i>Protection settings part 2.....</i>	31
<i>Protection settings part 2.....</i>	32
<i>Security settings CMS.....</i>	34
<i>Security settings Deliver:.....</i>	35
<i>CMS GUI settings.....</i>	36
<i>Special settings</i>	38
CONFIGURATION FILES	39
CMS-APPLICATION IS CONTROLLED IN THE CMS.PROPERTIES	40
DELIVER-APPLICATION IS CONTROLLED IN THE DELIVER.PROPERTIES.....	43
REPOSITORY-SPECIFIC ERROR AND LOGIN PAGES	45
INTEGRATION BETWEEN INFOGLUE PARTS	47
CACHE UPDATES AND BROADCASTS.....	47
INFOGLUE DELIVER APPLICATIONS CALLING CMS	47
DATABASES AND MAINTENANCE	48
SETTING UP THE DATABASE	48
SEPARATION AND SCALABILITY ON THE DATABASES.	48
SCALING THE APPLICATIONS ONLY.....	50
PORTLET (JSR 168) SUPPORT AND PITFALLS	51
NICE URI.....	52
HOW DOES IT WORK.....	52
LIMITATIONS AND PITFALLS.....	52
<i>Encoding</i>	52
SETUP INSTRUCTIONS.....	53
AUTHENTICATION AND AUTHORIZATION	54
INFOGLUE BASIC AUTHENTICATION / AUTHORIZATION	54
SSO THROUGH CAS	54
J2EE AUTHENTICATION	54
JDBC AUTHORIZATION	54
LDAP	54

InfoGlue – Administrative Manual

HOW TO COMBINE SEVERAL AUTHORIZATION MODULES	56
HOW TO SET UP LDAP	57
DETAILED DESCRIPTION OF JNDI-CONFIGS	58
<i>Sample config for JNDIBasicAuthorizationModule</i>	58
<i>Sample config for SimplifiedJNDIBasicAuthorizationModule</i>	61
ENCODING SETTING	63
NICEURI EFFECTS:	63
CONFIGURING THE CONTENT TREE WITH CUSTOM ICONS	64
CONFIGURING THE WYSIWYG EDITOR	65
EXAMPLE CONFIGURATION	66
PERFORMANCE TIPS	68
DIAGNOSTICS VIEWS AND FEATURES	70
STATISTICS (2)	70
VIEWAPPLICATIONSTATE (1)	70
LOG VIEWER	74
MAIL NOTIFICATIONS	74
REFERENCES	75
APPENDIX A GNU FREE DOCUMENTATION LICENSE	76

InfoGlue – Administrative Manual

Introduction

This document was written to give administrators as well as some developers a useful guide to the underlying setting and mechanism of InfoGlue. Hopefully the information herein will help clarify how to set up some aspects of InfoGlue and function as a reference document to all settings available in the system. Whatever the use, I hope the information is valuable and easy to understand.

The document will not include information about pure user aspects or functional development aspects as there are other documents that specialize in these matters. The aim is to give the administrator the best reference document as possible.

Feedback

This document is always a work in progress so mail any suggestions to bogeblad@yahoo.com and I will try to improve it as the platform evolves.

InfoGlue – Administrative Manual

General concepts

What is InfoGlue

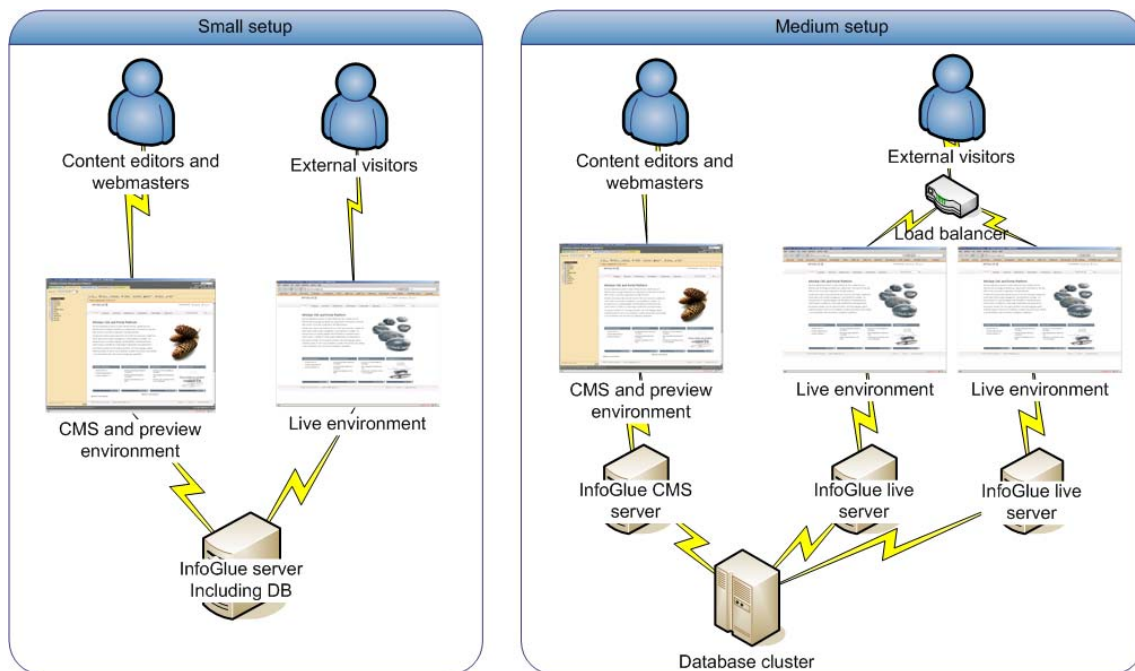
InfoGlue is a Content Management System. Content Management is very much another word for how to manage a company's or organization's information internally and externally. As content management is such a broad term we often say InfoGlue is a Web Content Management System. That is – InfoGlue helps organizations manage their information mainly targeted for the Web both internally, in intranets etc, and externally in public websites and extranets.

What parts are there in InfoGlue

InfoGlue is a pure Java platform. It is completely database driven which means both the management tool and the public sites are using information from a database. The platform consists of several different applications but from a users perspective the two important ones are the administrative tool and the different delivery engines.

The administrative tools are discussed in detail in the InfoGlue User Manual and are where you manage all aspects of your site. The delivery engines are specialized in presenting sites to users based on the data managed by the tools. By default InfoGlue installs 3 delivery engines. The first is the working version which presents the working version of the site. There is also a version called staging site which shows the site in a preview mode so the publisher can check that the site will look good after publication. The last delivery engine is the one that shows the live site to visitors.

Here are two sketches that shows first how a simple setup could look and then a more complex example with better redundancy:



Installation

The InfoGlue platform is build on top of standard components and are therefore both easy to maintain and easy to upgrade. This reference will take you through the process of installing it for the first time. This reference is written for Windows and Linux-based systems. If you run a different OS the steps should be the same but some minor details may differ.

Step 1: Installing Java

As the platform is written purely in Java you need the Java SDK installed. InfoGlue has from version 2.4.5 taken a step up and now only supports Java 5 (1.5.x) or later. Since Java 6 is still very young we have not started recommending it although many installations run InfoGlue on it already. The software is collected on www.java.sun.com. When you got it just install the software according to Sun's installations instructions. After installation complete it by adding the following environment variables to the system:

```
SET JAVA_HOME={InstallationPath} (for example c:\j2sdk1.5.5_05 )  
PATH=%PATH%;{InstallationPath}\bin (for example c:\j2sdk1.5.5_05\bin )
```

On windows, this is done by right-clicking on the My Computer-icon and selecting Properties. Then choose the tab Advanced and then Environment variables. In that dialog these settings are made. If you know how you can also make them at the command-prompt. To verify that the installation worked, start a new command prompt and type "java -version". The result should be something like this:

```
java version "1.5.0_05" Java(TM) Runtime Environment, Standard Edition Java HotSpot(TM) Client  
VM (build 1.5.0_05-b06, mixed mode)
```

We also strongly recommend that you set up some environment parameters:

```
SET JAVA_OPTS=-XX:MaxPermSize=256m -Xms256M -Xmx1024M  
SET CATALINA_OPTS=-XX:MaxPermSize=256m -Xms256M -Xmx1024M
```

The same environment-variables should be set up on Linux (export them as usual vars).

Step 2: Installing Apache Tomcat

The application has to have a servlet 2.3 compatible environment to run in and InfoGlue is known to run on several although the installer only supports Apache Tomcat 5.x or above. Recommended is Tomcat 5.5 or above. This software is of course free and can be found at <http://tomcat.apache.org>. When you have the right software just install it as instructed. To check that the installation was successful you might want to start tomcat from the start-menu and then browse to <http://localhost:8080>. There you should see a tomcat information screen. After this check please stop tomcat and proceed to step 3.

Step 3: Installing MySQL

All information in the InfoGlue platform is stored in a database. This database could be almost any JDBC-2 compliant database and the installer default supports MySQL, SQL-Server, DB2 and Oracle. This tutorial will instruct how to install the fast, competent and free MySQL database. You can fetch it from www.mysql.com. InfoGlue supports ver. 3.23 or later but strongly suggests version 5.0 or above as a number of limitations has been removed in that version. Install the database as instructed by the MySQL documentation. If you want to use large attachments on your sites like large images or movies you must follow the instructions on http://www.mysql.com/doc/en/Package_too_large.html. The standard limit is about 1Mb so you will probably want to do that.

Step 4: Installing InfoGlue

Normal way

Time has come to install the InfoGlue platform itself. The installer comes in a zip-file which you just unpack with your favorite unzip-program. In the installation directory there is a visual installer you start by running the install.bat(windows) / install.sh (Linux) file. If you prefer a command based installation run installcmd.bat (Windows) / installcmd.sh (Linux) instead. The installer will try to guide you as much as possible and if you move the mouse over the labels in the visual installer more help will be shown as tool tips. If you want to run it from Unix/Linux-systems you have to set the rights on the new directory so it can be executed. This is done by "chmod 777 * -Rf" in the newly unzipped directory. Below is a small sequence for Linux-users and it assumes you are located in the directory where you downloaded InfoglueInstaller2.x.xFinal.zip:

```
unzip "InfoglueInstaller2.x.xFinal.zip"
chmod 777 "InfoglueInstaller" -R
cd "InfoglueInstaller"
./installcmd.sh
```

Silent option

If you want to you can specify all setting for the installer in an xml file named installconf.xml and run the installer in silent mode. This is convenient sometimes for a hosted environment or if you wish to automate the installation. Just modify the existing config example file and run InfoGlue with silent option.

```
Example windows: "infogluecmd.bat silent"
Example linux/unix: "./installcms silent"
```

Upgrading older versions of InfoGlue

The installer can also upgrade existing versions of InfoGlue including updating the database model if needed. Up to now InfoGlue can upgrade versions so far back as 1.0 and we hope to keep it that way in the future. That way upgrades should be pretty easy. When upgrading from 1.3.2 or earlier there may be some problems with access rights, packet structure and other backward compability issues which can be resolved but you should know about. Since 2.0 InfoGlue has settled much more into a more modern setup.

InfoGlue – Administrative Manual

Detailed descriptions of the fields in the installer / upgrader:

Database dialog:

InfoGlue 2.8.0
CMS and Portal platform

InfoGlue is released under the GNU General Public License

Please visit us at:
<http://www.infoglue.org>
for documentation, upgrades, contribution, support etc.

The core developers behind InfoGlue:
Mattias Bogeblad, Stefan Sik, Patrik Nyborg,
Per Jonsson, L.B. Jensen, Frank Febrarro,
Jed Prentice, Magnus Güvenal, Robert Lerner

Special thanx to these organisations:
Modul1, Korsnäs Frövi, Tripnet, MSU
Univ. of Gothenburg, OMX GROUP, WM-Data
SLU and many, many more

INFOGLUE

Database setup

For the system to work you will have to state which database you wish to use. All fields have a help texts if you click on the question mark next to it.

Database type	MySQL	?
Database name	infoglue	?
Database suffix		?
Database server	localhost	?
Database port	3306	?
Database instance		?
InfoGlue username	INFOGLUE_CMS	?
InfoGlue password	*****	?

The installer can automatically create a database and set up some basic data to make the basic functionality and examples work. If you would like the installer to do these tasks you mark the options you want below and then supply the installer with an administrative db-account. The account must have rights to create databases and tables on the server stated above.

Create database	<input type="checkbox"/> Yes	?
Create tables and initial data	<input type="checkbox"/> Yes	?
Create example sites	<input type="checkbox"/> Yes	?
Upgrade to utf-8?	<input type="checkbox"/> Yes	?

DBA name	root	?
DBA password		?

<< Back Next >>

Database type: The database brand you wish to install or upgrade InfoGlue on. Valid are "MySQL", "SQL Server", "DB2" and "Oracle".

Database name: The name of the InfoGlue database – either an existing one you wish to upgrade or the one you wish to create.

Database suffix: This field gives you the option to add a suffix to the database name so you can have more than one InfoGlue installation in the same database. Used primarily in hosting services.

Database server: The hostname of the database server on which the database is located. Can be an IP or a hostname.

Database port: The port for the database server if different from default.

Database instance: Use this for SQLServer setups where you have several instances running.

InfoGlue – Administrative Manual

InfoGlue username: The username InfoGlue is to use for connecting to the database. This username is not related to user's logins.

InfoGlue password: The password InfoGlue is to use for connecting to the database. This password is not related to user's logins.

Create database: Do you wish to install a new InfoGlue database? If you already have an InfoGlue installation and wish to keep the data you don't want to. This feature is only available for MySQL and SQLServer. **If you are upgrading an old install don't select this.**

Create tables and initial data: Do you wish to InfoGlue to fill the database with the needed tables etc. If you don't want to script the creation of the database yourself you should choose this. **If you are upgrading an old install don't select this.**

Create example sites: This installs a couple of example sites for you to look at. **If you are upgrading an old install don't select this.**

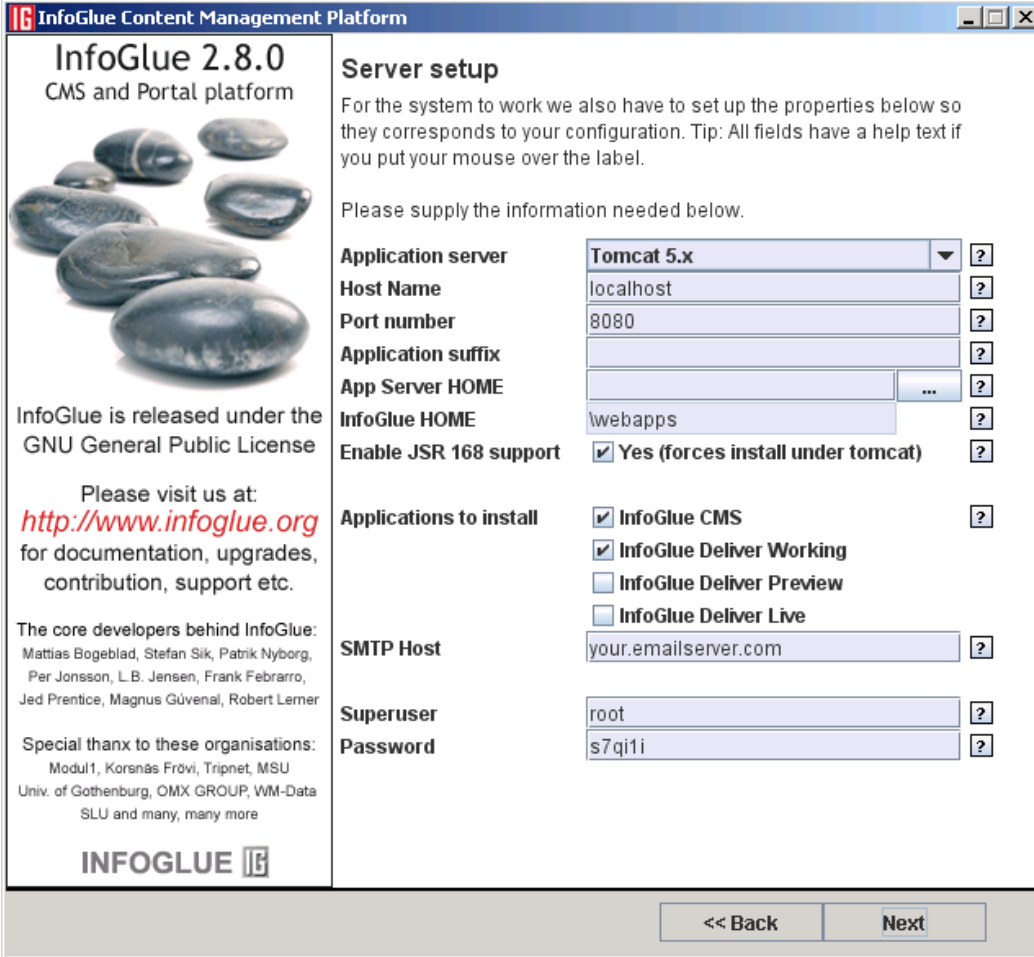
Upgrade to utf-8: Use this option if you have tried to upgrade and international characters gets messed up. It fixes a problem in the 1.2-version. **DO NOT USE AFTER 1.2.**

DBA name: The root (or similar) username to the database. InfoGlue needs it if it's to create the database in a new install.

DBA password: The root (or similar) password to the database. InfoGlue needs it to be able to create the database.

InfoGlue – Administrative Manual

Server dialog:



InfoGlue 2.8.0
CMS and Portal platform

InfoGlue is released under the GNU General Public License

Please visit us at:
<http://www.infoglue.org>
for documentation, upgrades, contribution, support etc.

The core developers behind InfoGlue:
Mattias Bogeblad, Stefan Sik, Patrik Nyborg,
Per Jonsson, L.B. Jensen, Frank Febrarro,
Jed Prentice, Magnus Güvenal, Robert Lerner

Special thanx to these organisations:
Modul1, Korsnäs Frövi, Tripnet, MSU
Univ. of Gothenburg, OMX GROUP, WM-Data
SLU and many, many more

INFOGLUE

Server setup

For the system to work we also have to set up the properties below so they corresponds to your configuration. Tip: All fields have a help text if you put your mouse over the label.

Please supply the information needed below.

Application server	Tomcat 5.x	?
Host Name	localhost	?
Port number	8080	?
Application suffix		?
App Server HOME		?
InfoGlue HOME	webapps	?
Enable JSR 168 support	<input checked="" type="checkbox"/> Yes (forces install under tomcat)	?
Applications to install	<input checked="" type="checkbox"/> InfoGlue CMS <input checked="" type="checkbox"/> InfoGlue Deliver Working <input type="checkbox"/> InfoGlue Deliver Preview <input type="checkbox"/> InfoGlue Deliver Live	?
SMTP Host	your.emailserver.com	?
Superuser	root	?
Password	s7qi1i	?

<< Back Next >>

Host Name: The hostname or IP-address of the application server you are installing InfoGlue on. If you want people to access your site from other computers this is important to set correctly.

Port number: The port number tomcat operates on.

Application suffix: The suffix will be added to the name of the InfoGlue applications. This feature is mostly used in shared environments like hosting services. You can leave it out if this is a one-client installation.

Tomcat HOME: Point out the home directory of your tomcat installation. Often something like c:\Program Files\Apache Group\Tomcat 4.1

Tomcat VERSION: We now support both tomcat 4.x and 5.x but it is important you state which version it is as they have organized their files a bit different.

InfoGlue HOME: Point out the home directory of your InfoGlue directory. As InfoGlue will create a couple of applications/subdirectories you are wise to create a common InfoGlue directory somewhere on the disc which you then supply.

Enable JSR 168 support: This sets InfoGlue up support the JSR-168 portlet specification. Choosing this forces install under tomcat webapps.

InfoGlue – Administrative Manual

Applications to install: InfoGlue consists of several applications and two are mandatory. You are also able to select if you wish to install a preview-deliver and a live-deliver. The preview-deliver is used for preview at publication and the live site shows only published data and are aimed to be the deliver used for the public. Both options are recommended.

SMTP Host: Your email server used for outgoing mail.

Super user: The account name of the super user you want to create.

Password: What password the super user should have.

After installation you should start Tomcat again and open the address stated by the installation program. The default login name is **"administrator"** and the password is **"changeit"**. If images does not show and you are running Linux you probably have to chmod the directory digitalAssets and logs under each installer application so the applications are allowed to create files there.

WARNING: InfoGlue puts the root account in the cms.properties under the cms webapp. This account is used when security settings have been set up invalid or you must override something without having the rights as an administrator. Be aware to use a good password and username here as it will be a security risk if you don't. To secure the application even more you can remove the username and password altogether.

Now you are ready to go!! Good luck in building you websites.

InfoGlue – Administrative Manual

Alternative deploy configuration

Instead of using the installer some people and projects prefer to set up a deployment environment which let's them redeploy a configuration easily with just an ant-task. This is a recommended way when working with a development server which you want to be able to update easily and we also use this way for some live installations as the approach let's us get the newest InfoGlue versions automatically with one command. One should however know how CVS works and only fetch the latest stable InfoGlue versions if it's a production environment.

I thought we should be nice and describe a simple example setup.

First we create a Build-folder:

```
cd /root (can be anywhere – this is just an example)
mkdir BuildProject
mkdir software
```

Download Apache Ant

```
cd software
wget http://www.infoglue.org/downloads/apache-ant-1.6.5-bin.tar (or get it from the official apache)
tar -xvf apache-ant-1.6.5-bin.tar
cd ../BuildProject
```

Fetch the latest version of InfoGlue:

```
cvs -d :pserver:anonymous@www.infoglue.org:2401/home/cvsroot checkout infoglue
```

or fetch a tagged version

```
cvs -d :pserver:anonymous@www.infoglue.org:2401/home/cvsroot checkout infoglue241final infoglue
```

Create redeployAll.sh in BuildProject containing:

```
/etc/init.d/tomcat stop
rm /usr/local/tomcat5.5/logs/* -f
cd infoglue
#Enable this if you wish to have always the latest development version of InfoGlue to be fetched.
#/root/software/apache-ant-1.6.5/bin/ant --noconfig fetchInfoGlue
/root/software/apache-ant-1.6.5/bin/ant --noconfig deploy
/root/software/apache-ant-1.6.5/bin/ant --noconfig deploy-pluto
```

Chmod it:

```
chmod 755 redeployAll.sh
```

InfoGlue – Administrative Manual

Now create and edit infoglue/build.properties

Set all paths and database settings so it suits your needs. Here is a template with the most used parameters:

```
rm infoglue/build.properties
vi infoglue/build.properties
```

Infoglue/build.properties

```
##
# $Id: build.properties,v 1.27 2005/11/20 20:59:46 mattias Exp $
#####
project.version=2.5.0
#build.compiler=jikes

##
# deployment directories for each version of the InfoGlue webapp
##

deploy.dir.cms=/usr/local/tomcat5.5/webapps/infoglueCMS
deploy.dir.working=/usr/local/tomcat5.5/webapps/infoglueDeliverWorking
deploy.dir.preview=/usr/local/tomcat5.5/webapps/infoglueDeliverPreview
deploy.dir.live=/usr/local/tomcat5.5/webapps/infoglueDeliverLive

##
# database
##
#Oracle
database.driver.class=com.mysql.jdbc.Driver
database.driver.engine=mysql
database.url=jdbc:mysql://localhost/infoglue?autoReconnect=true&useUnicode=true&characterEncoding=UTF-8
database.user=infoglue
database.password=xxxxxxx
database.minLimit=5
database.maxLimit=200
database.maxConnections=200
database.mapping=mapping.xml
database.encoding=UTF-8
database.use.unicode=true

##
# Hibernate
##
hibernate.dialect=net.sf.hibernate.dialect.MySQLDialect
hibernate.show_sql=false

masterServer=
slaveServer=

##
# Which mode does the engine run in... 0=working(first preview), 2=publish(staging preview), 3=published(live)
##
operatingMode.cms=0
operatingMode.working=0
operatingMode.preview=2
operatingMode.live=3

##
# portlet
##
enablePortal=true
actionDispatcher=org.infoglue.deliver.portal.dispatcher.PortalServletDispatcher

portletBase=/usr/local/tomcat5.5/webapps/
portletBase.working=/usr/local/tomcat5.5/webapps/
portletBase.preview=/usr/local/tomcat5.5/webapps/
portletBase.live=/usr/local/tomcat5.5/webapps/
deploy.dir.shared=/usr/local/tomcat5.5/shared/
```

InfoGlue – Administrative Manual

```
##
# web-specific settings, user to locate images, etc.
##
externalWebServerAddress=
webServerAddress=http://xxx.xxx.xx
liveWebServerAddress= http://xxx.xxx.xx
context.root.cms=infoglueCMS
context.root.working=infoglueDeliverWorking
context.root.preview=infoglueDeliverPreview
context.root.live=infoglueDeliverLive
#digitalAssetPath=/usr/local/tomcat5.5/webapps/digitalAssets
#requestArgumentDelimiter=&
URIEncoding=ISO-8859-1
workflowEncoding=UTF-8
#errorUrl=http://localhost:8080/error.html
errorUrl=/error.jsp
errorBusyUrl=/errorBusy.jsp

enableNiceURI=true
niceURIEncoding=UTF-8
niceURIAttributeName=NiceURIName

#Cache settings
useSelectivePageCacheUpdate=true
cachePath=/tmp
compressPageCache=true
#siteNodesToRecacheOnPublishing=
#pathsToRecacheOnPublishing=/
#recachePublishingMethod=requestAndMetaInfoCentric
recacheUrl=http://localhost/ViewPage.action

##
# default tree
##
tree=html

#settings for how the content tool should work
showContentVersionFirst=true

#settings for how the structure tool should work
showComponentsFirst=true

#Setting for if content types can be protected
protectContentTypes=false

#Setting for if categories can be protected
protectCategories=false

# Protect workflows
protectWorkflows=false

#Setting for which HTML editor to use FCKEditor | HTMLArea
wysiwygEditor=FCKEditor

#SET THIS TO TRUE IF YOU ARE RUNNING ORACLE OR DB2
useShortTableNames=false

# Show all workflows or only the one you owns
showAllWorkflows=false

##
# edition pagination
##
edition.pageSize=10

##
# AutoUpdate Settings
# ( Should point to masterserver, which provide a mirror list later
#  For now, point up2dateUrl directly to a xml updates list )
##
up2dateUrl=http://www.infoglue.org/infoglueDeliver/ViewPage.action?siteNodeId=23
```

InfoGlue – Administrative Manual

```
##
# security settings - basic
##

#loginUrl=Login.action
loginUrl=https://localhost:8080/infoglueDeliverWorking/Login.action
invalidLoginUrl=LoginInvalidLogin.action
authenticatorClass=org.infoglue.cms.security.InfoGlueBasicAuthenticationModule
authorizerClass=org.infoglue.cms.security.InfoGlueBasicAuthorizationModule
serverName=localhost:8080
authConstraint=cmsUser
extraParametersFile=InfoGlueAuthorizationParameters.properties
casValidateUrl=
casServiceUrl=

##
# mail
##
mail.smtp.host=smtp.xxx.xx
mail.smtp.auth=false
mail.smtp.user=
mail.smtp.password=
mail.contentType=text/html
systemEmailSender=mattias.bogeblad@modul1.se
warningEmailReceiver=mattias.bogeblad@modul1.se

##
# Sets up a superuser
##
administratorUserName=root
administratorPassword=xxxxxx
administratorEmail=mattias.bogeblad@modul1.se
```

When save run redeployAll.sh i /root/BuildProject
./redeployAll.sh

The result shall be BUILD SUCCESSFUL.

Adding Apache HTTP-Server as a front-end

Many linux/unix-servers have Apache HTTP-server installed by default and it may even be used by several previous projects. If so there are many InfoGlue-installations which utilize the web server as a front-end to Apache Tomcat. Even though this is not an InfoGlue-specific task we have decided to include a small and simple how-to here.

First make sure Apache HTTP-server is installed. If you don't know how you should start on www.apache.org before continuing.

We will use the Apache mod_jk-module for connecting Apache and Tomcat. Download and install from <http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/> or download and compile yourself.

When installed you configure it by adding this or similar last to the httpd.conf (often in /etc/httpd/conf) or in an included file:

```
#
# Mod_jk settings
#

AllowEncodedSlashes On

JkWorkersFile "/usr/local/tomcat5.5/conf/workers.properties"
JkLogFile "/usr/local/tomcat5.5/logs/mod_jk.log"

JkLogLevel error

JkMount /infoglueCMS default
JkMount /infoglueCMS/* default
JkMount /infoglueDeliverWorking default
JkMount /infoglueDeliverWorking/* default
JkMount /infoglueDeliverPreview default
JkMount /infoglueDeliverPreview/* default
JkMount /infoglueDeliverLive default
JkMount /infoglueDeliverLive/* default

# Use SetEnvIf if you want to let apache take care of all digitalAsset calls so Tomcat only
# get app-calls. Improves performance quite a bit actually. Use only for the live instances.
SetEnvIf Request_URI "/digitalAssets/*" no-jk

# Now /home will go to the dir below - Use only for the live instances.
Alias /digitalAssets /usr/local/tomcat5.5/webapps/infoglueDeliverLive/digitalAssets

JkMount /* default
# End of mod_jk settings
```

Then create workers.properties in the dir pointed out in the conf (in this case /usr/local/tomcat5.5) with the following content:

```
workers.tomcat_home=/usr/local/tomcat5.5
workers.java_home=/usr/java/jdk1.5.0_04
ps=/
worker.list=default
```

```
worker.default.port=8009
worker.default.host=localhost
worker.default.type=ajp13
worker.default.lbfactor=1
```

Then restart Apache HTTP-server and make sure Tomcat is running. Then call the server through apache. Usually that means calling the same address but on port 80 instead of 8080.

InfoGlue Configuration

In InfoGlue not much is hard coded. Almost all settings affecting the behaviour of the system can be configured either in the tools or in the properties files on disk.

We have tried to let administrators keep as much as possible of their configurations in the database as that very much eases the upgrade process in the future as well as let's the users reconfigure and manage the application settings through the management interfaces.

To allow this InfoGlue is coded so it first checks if an application setting exists in database-driven application properties. If not InfoGlue checks in the properties-files on disk and lastly, if not defined there either, it defaults to preset values. The next sections describe these settings.

InfoGlue – Administrative Manual

Application settings

This part describes the settings that can be handled through the management tool. If you wish to have different settings for different servers you can add server nodes in this view as well. Just be sure to call the new nodes what the servers are called (InetAddress) as the names are used when looking for server specific properties. This feature is not used by many but now you know it exists.

InfoGlue fetched any property it needs like this:

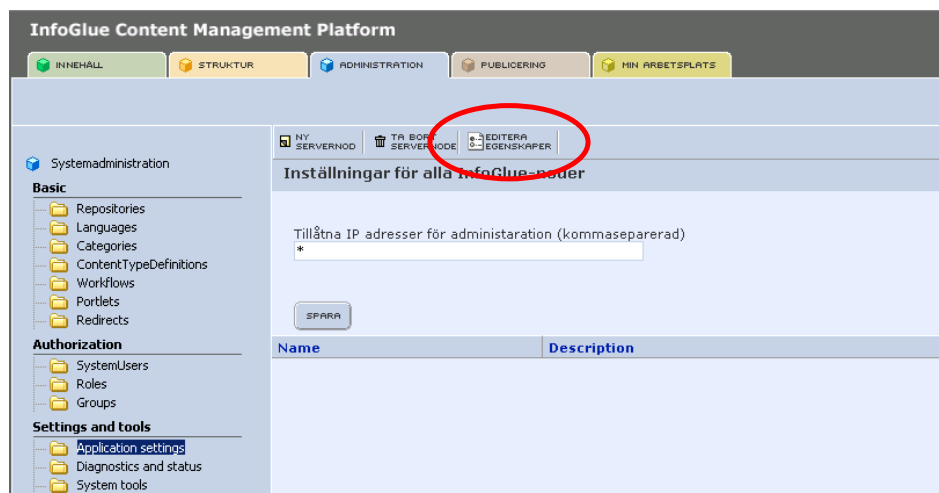
1. Check if there is a server node matching the server running the InfoGlue instance wanting the property/setting. If so InfoGlue gets the application setting in question filled in under that node.
2. If no value or no server node was found InfoGlue looks at the default application settings and fetches the value for that attribute.
3. If no value was specified in either local server node or in general application settings InfoGlue fallbacks to the property files on disk. Mainly deliver.properties and cms.properties. It looks for the value there.
4. If still no property was found a default value is used which is hard coded in the application.

Since InfoGlue 2.3.5 we recommend all app settings to be handled in the management tool with one or two exceptions which should be handled in the property files as they are unique to each deliver-instance.

If you can access the management tool (the blue tab) just press "Application settings". Before we head into the detailed properties we would like to point the only field in this screen:

Allowed admin IP-addresses: States the IP-addresses that are allowed to call some admin actions. This is very important as **you must allow the cms-server's IP if you got a distributed environment as the cache update calls are handled like this. If not the cache will not get updated correctly.**

To edit the actual properties press the "Edit Properties"-button.



Below is a thorough walkthrough of the sections and fields under the edit properties button.

InfoGlue – Administrative Manual

Common settings

Common settings		Cache settings	CMS connection	Webserver settings	Encoding settings	Mail settings
Protection settings		Security settings CMS	Security settings Deliver	CMS GUI Settings	Special settings	
Common settings						
Session timeout	<input type="text" value="180000"/>					
A commaseperated list of sitenode id's to recache	<input type="text" value="2"/>					
Recache method	<input type="text" value="requestCentric"/>					
Recache URL	<input type="text" value="http://working.officestand2.com:80"/>					
Enable Nice/Friendly URI in working	<input type="text" value="Inherit"/>					
Enable Nice/Friendly URI	<input type="text" value="Yes"/>					
Nice/Friendly URI encoding	<input type="text" value="iso-8859-1"/>					
Nice/Friendly URI attribute	<input type="text" value="NiceURIName"/>					
Nice/Friendly URI Character mappings	<div><div>a=a ä=ae ö=oe</div></div>					
Default replacement character (if not mapped) and non ascii.	<input type="text" value="none"/>					
Nice/Friendly URI force lowercase	<input type="text" value="Yes"/>					
Request argument delimiter (& default)	<input type="text"/>					
Error page URL	<input type="text"/>					
Use short table names (Oracle and DB2 should have yes)	<input type="text" value="Inherit"/>					
Number of content versions (for each language) to keep (-1 = all).	<input type="text" value="5"/>					
State how much time must have passed between two versions for the cleanup routine to keep both versions (leave empty to skip this logic).	<input type="text" value="24 hours"/>					
Duplicate assets on version change (no default)	<input type="text" value="No"/>					
Asset upload transformations (XML - read in developer manual about syntax).	<div><div><transformations> <transformation assetKeyPattern="(minibild)" inputFilePattern=".* (jpeg jpg gif png).*" keepOriginal="false"> <transformResult type="scaleImage" width="50" height="50" outputFormat="jpg" assetSuffix="small"/> </transformation> <transformation assetKeyPattern="(?!minibild)" inputFilePattern=".*</div></div>					
<div><input type="button" value="SAVE"/> <input type="button" value="SAVE & EXIT"/> <input type="button" value="CANCEL"/></div>						

Name / description

Session timeout: How long until a user session are invalidated? Controls when a user is automatically logged out etc. Set in milliseconds.

A commaseperated list of sitenodes to recache: This is to enable one of the recache strategies to know what to recache after a publication.

Recache method: Can be one of these:

- **requestCentric** (recommended) – before flushing all caches the strategy makes a few requests to specified pages so a lot of data get's reached. Minimizes the load after a publication.
- **requestAndMetaInfoCentric** – before flushing all caches the strategy makes a few requests to specified pages and indexes all pages meta info so a lot of data get's reached. Minimizes the load after a publication – a bit slower.
- **contentCentric** – before flushing all caches the strategy indexes some contents. Minimizes the load after a publication – not recommended.

Recache URL: Used by the recache strategies which calls pages. Should point out the ViewPage.action with a full url.

InfoGlue – Administrative Manual

Enable Nice/Friendly URI in working: Set to Yes if you want the working deliver engine to render url:s like /products/infoglue_cms instead for /ViewPage.action?siteNodeId=7 etc. Overrides next field.
Enable Nice/Friendly URI: Set to Yes if you want all the deliver engines to render url:s like /products/infoglue_cms instead for /ViewPage.action?siteNodeId=7 etc.
Nice/Friendly URI encoding: What encoding should be used if encoding takes place due to i18n-characters.
Nice/Friendly URI attribute: What meta info attribute should be used to generate the URI:s. Can be any page meta info attribute or SiteNode.name which uses the non localized name from the structure three.
Nice/Friendly URI Character mappings: This field allows you to state what characters/strings should replace non-ascii-characters in nice uri:s. Below is an example on how we map some special swedish characters: <div><pre>â=a ä=ae ö=oe</pre></div>
Default replacement character (if not mapped) and non ascii: You can state here what character should be used for non ascii-chars in the nice uri. Use "none" if you want such chars to just be removed without replacement. Default replacement is "_" (underscore).
Nice/Friendly URI force lowercase: States if you wish the system to force lowercase on all nice uri-names.
Request argument delimiter: Should we use & or & or ; as the argument separator?
Error page URL: If you want a custom error page you can set that here. When an error occurs the user will be shown the information and the url can be relative or full.
Use short table names: The value should be "Yes" if you run InfoGlue against a DB/2 or Oracle database. Set by the installer usually but can be set here as well.
Number of content versions (for each language) to keep (-1 = all): This field sets up how many versions the content cleanup routine should keep. By default it keeps all. The cleanup routine runs each day and if you want to save space and think you don't need all the old versions in the system you can state how many in each language you want to keep.
State how much time must have passed between two versions for the cleanup routine to keep both versions (leave empty to skip this logic): This field makes it possible to have the cleanup routine be more selective about what it cleans.
Duplicate assets on version change (no default): Here you can state if you want the old behaviour where all assets were duplicated between versions or if the system should use the new method (default) which shares assets between versions unless changed.
Asset upload transformations: This field lets you define if there should be any global transformations of images. This feature lets you force so images can be resized to predefined sizes upon uploads. Below is the syntax for transformation setups: <div><pre><transformations> <transformation assetKeyPattern="(minibild)" inputFilePattern=".*(jpeg jpg gif png).*" keepOriginal="false"> <transformResult type="scaleImage" width="50" height="50" outputFormat="jpg" assetSuffix="small"/> </transformation> <transformation inputFilePattern=".*(jpeg jpg gif png).*" keepOriginal="false"> <transformResult type="scaleImage" width="100" height="100" outputFormat="png" assetSuffix="medium"/> <transformResult type="scaleImage" width="50" height="50" outputFormat="jpg" assetSuffix="small"/> </transformation> </transformations></pre></div>
As you can see you can state 0..n transformations which will be checked for each upload (only new assets – not updates). For each transformation you can state regular expressions to control which images and asset keys it triggers on and transforms. You can also state if the original file should be kept or not. For now there is only a scaleImage transform result type but that could change in the future. Let us know what other results would be nice to have.

InfoGlue – Administrative Manual

Cache settings

Common settings

Cache settings

CMS connection

Webserver settings

Encoding settings

Mail settings

Protection settings

Security settings CMS

Security settings Deliver

CMS GUI Settings

Special settings

Cache settings

Enable Page Cache

Inherit

Selective page cache update

Inherit

Expire cache automatically (interval below)

Inherit

Cache expiration interval

Request timeout (in deliverWorking and preview only). 0 = infinite

Request timeout (in deliverLive only). 0 = infinite

Kill live request after timeout (No means infoglue only sends a warning email)

Inherit

Limit the number of active threads (sleeps others until the load lessens). This can result in max threads being reached if the site is under very heavy load but will hopefully keep the application alive.

Inherit

Max number of concurrent threads (makes new request sleep until others have finished). Only active if the above field is Yes.

Max time for requests before starting sleeping new threads. Use when you have a huge amount of traffic but each request is very fast normally and you only want the new threads to be waiting if the processing time gets big.

Compress page cache

Inherit

Compress page response

Inherit

Disable decorate final rendering (if you have good templates you can disable this)

Inherit

Use IP-security on application actions

Inherit

Allow X-Forwarded IP addresses

Inherit

Allowed admin IP addresses (comma separated)

Default page cache key

Default component cache key

Cache settings

SAVE

SAVE & EXIT

CANCEL

Name / description

Enable Page Cache: Enable or disable the page cache possibility globally. You must have a very good reason for setting it to "No" as it has a big performance impact and the cache on pages can be disabled individually instead where needed.

Selective page cache update: Enable or disable if pages should be updated selectively in deliverWorking – do not set to "No" unless you have a special reason.

Expire cache automatically: Enables one to set that the page caches and other caches are thrown away every x millisecond (x defined below). This is not a common setup as it's quite ineffective and there are better options if such behaviour is needed.

Cache expiration interval: How many milliseconds to wait until refreshing the caches. Only used if Expire cache automatically is set to yes.

Request timeout (in deliverWorking and preview only): If set to a positive value the working/preview deliver will kill off request-threads that take longer than the set value. This is to avoid faulty test templates etc to block/disrupt the system. A value between 30000 and 120000

InfoGlue – Administrative Manual

could be suitable if you have problems with blockage/hangs – otherwise keep empty.
Request timeout (in deliverLive only): If set to a positive value the live deliver will either report or kill off requests that take longer than the set value. This is to avoid stability issues when issues arises. In some situations blocked threads can linger forever. A value between 30000 and 120000 could be suitable if you have problems with blockage/hangs – otherwise keep empty. Very nice to have if you have surrounding systems which are behaving badly like taking to long for example. Whether it kills the thread after the time specified here or just sends a warning mail depends on the next field.
Kill live request after timeout: Decides if the live deliver should kill threads or just warn the sysadmin when threads take to long.
Limit the number of active threads: This feature lets you tell InfoGlue to queue up threads if to many comes in at once so we limit the synchronization costs. OBS: Use only after you have tested very much in a controlled environment so you know what is best for your site. Not many needs this setting.
Max number of concurrent threads: Lets you state how many threads can be active at once. Only active if you set the previous setting to true. A value of 20 to 40 is not uncommon if used.
Max time for requests before starting sleeping new threads: Lets you specify how long the request must become before InfoGlue should start to limit the incoming active requests. Only active if you set the previous setting to true. A value of 3000 (milliseconds) is not uncommon if used.
Compress page cache: States if the page cache should be compressed before stored in memory. Saves quite a lot of RAM.
Compress page response: States if the resulting pages should be gzip:ed before sent to the visitor. Saves bandwidth and speeds delivery a bit.
Disable decorate final rendering (if you have good templates you can disable this): Disables last rendering in velocity after the components are rendered. If you have nice components which behaves you may be able to turn this off. Makes some sites a bit faster.
Use IP-security: States if some application actions like UpdateCache and ViewApplicationState should be protected on which IP-addresses the call come from. If set to Yes you must specify the ip-addresses in the field below.
Allowed admin IP-addresses: States the IP-addresses that are allowed to call some admin actions. OBS: This settings are also on the application settings startpage – use them there. Take care in allowing the cms-server to do this as the cache update calls are handled like this. If not cache will not get updated correctly.
AllowXForwardedIPCheck: Allow you to specify that it is ok for Apache X-forwarded IP-addresses in IP-protected actions like ViewApplicationState.action.
Default page cache key: States what parameters should be taken into account when creating or fetching data from the page cache. The complexity of this affects how large the page cache is in memory and how often the page cache holds what the user wants. The default key is: <i>siteNodeId_languageId_contentId_userAgent_queryString</i> which means for example which browser the user has affects the hit rate on the cache. If one knows one will not make components which cares of the browser on the server side the page cache key can be shorter which is good. If one need cookies and session parameters can be used in the key.
Default component cache key: States what parameters should be taken into account when creating or fetching data from the component cache (cache containing complete rendered component on a page). The complexity of this affects how large the component cache is in memory and how often the cache holds what the user wants. The default key is: <i>{pageCacheKey}_componentId_componentSlotName_componentContentId_isInherited</i> which means here as well for example which browser the user has affects the hit rate on the cache. If one knows one will not make components which cares of the browser on the server side the page cache key can be shorter which is good. If one need cookies and session parameters can be used in the key.

InfoGlue – Administrative Manual

Cache settings: This is a feature which allows you to state individual sizes for each cache. The syntax is for example:

CACHE_CAPACITY_contentVersionCache=10000
CACHE_CAPACITY_componentPropertyCache=5000
CACHE_CAPACITY_contentAttributeCache=50000

This means that the cache called "contentVersionCache" as named in the ViewApplicationState.action-view will use a LRU-algorithm when inserting things in the cache. When the cache capacity we have defined as 10000 above reaches it's limit it will throw away the item used least recently. This way one can control memory usage quite a bit. Especially the page cache and some caches above can be very large after a while. If you start running without these settings you can check out how much memory is used in ViewApplicationState.action and limit the memory if needed. Remember – less memory caches means worse performance so don't take it to far.

InfoGlue – Administrative Manual

CMS Connection

Common settings Cache settings **CMS connection** Webservice settings Encoding settings
Mail settings Protection settings Security settings CMS Security settings Deliver CMS GUI Settings
Special settings

CMS connection

CMS Full Base URL

CMS Base URL

Component editor URL

Component renderer URL

Component renderer action

Edit On Sight URL

Name / description
CMS Full Base URL: The full URL to the cms. This is used by the MyDesktop-tool as the return address when invoking workflows from there.
CMS Base URL: The path to the cms. Could be a full url or a relative one to the domain. Used to construct action-urls in structure tool for example to avoid relative urls which might be a problem.
Component editor URL: Points to the action that edits pages (add / remove components etc). Used by the structure tool component editor.
Component renderer action: The component editor action.
Edit On Sight URL: Points out the editor action URL.

InfoGlue – Administrative Manual

Webserver settings

This section let's you define what addresses meet the user or what paths the applications should put files in. Leave empty most of the time.

The screenshot shows the 'Webserver settings' tab selected in a settings interface. The interface has a top navigation bar with tabs: 'Common settings', 'Cache settings', 'CMS connection', 'Webserver settings' (active), 'Encoding settings', 'Mail settings', 'Protection settings', 'Security settings CMS', 'Security settings Deliver', and 'CMS GUI Settings'. Below the tabs, the 'Webserver settings' section contains several input fields, each with a trash icon for clearing the value: 'Webserver address', 'Application base URL', 'Digital asset base URL', 'Images base URL', and 'Digital asset directory'. Below these fields is a checkbox labeled 'Store assets on disk (not in database as default)' and a dropdown menu currently set to 'Inherit'. At the bottom of the form are three buttons: 'SAVE', 'SAVE & EXIT', and 'CANCEL'.

Name / description
Web server address: Never use this really except if you know what you are doing. Leaving it blank let's InfoGlue operate with relative URLs only which is more flexible.
Application base URL: The cms base URL.
Digital asset base URL: What URL the assets are found at. Defaults to the dir under the webapp and is relative the context by default.
Images base URL: The URL to the standard InfoGlue image directory. Leave it empty.
Digital asset path: The file system path to where assets should be stored. Defaults to a folder called digitalAssets under the webapp dir – could be set to a SAN or something.
Edit On Sight URL: Points out the editor action URL.

InfoGlue – Administrative Manual

Encoding settings

This section controls all encodings in the system.

Encoding settings		
Input character encoding (cms)	<input type="text"/>	
Input character encoding (deliver)	<input type="text"/>	
URI encoding	<input type="text"/>	
Workflow encoding	<input type="text"/>	
Property forms encoding	<input type="text"/>	
Upload asset key from encoding	<input type="text"/>	
Upload asset key to encoding	<input type="text"/>	
Asset key from encoding in WYSIWYG asset dialog	<input type="text"/>	
Asset key to encoding in WYSIWYG asset dialog	<input type="text"/>	

SAVE SAVE & EXIT CANCEL

Name / description
Input character encoding (cms): What input character encoding should we expect? Seldom needed to be changed – only if you experience encoding problems when entering data in InfoGlue’s standard GUI:s.
Input character encoding (deliver): What input character encoding should we expect when data is entered on pages on the site? Important when we use portlets etc and can depend on what encoding the site is written in, how the web server/tomcat is configured etc.
URI-encoding: What encoding should be used when InfoGlue deals with i18n strings when creating url’s inside cms?
Workflow encoding: What input character encoding should be used for the workflow definitions. Defaults to utf-8.
Property forms encoding: Not used at the moment.
Upload asset key from encoding: Just use this if asset keys(i18n chars) get scrambled. Should default to correct encoding since 2.5.0.
Upload asset key to encoding: Just use this if asset keys(i18n chars) get scrambled. Should default to correct encoding since 2.5.0.
Asset key from encoding in WYSIWYG asset dialog: Just use this if asset keys when using the wysiwyg with i18n chars in names get scrambled. Should default to correct encoding since 2.5.0.
Asset key to encoding in WYSIWYG asset dialog: Just use this if asset keys when using the wysiwyg with i18n chars in names get scrambled. Should default to correct encoding since 2.5.0.

InfoGlue – Administrative Manual

Mail settings

Here you define all settings concerning mailservers etc.

Common settings Cache settings CMS connection Webserver settings Encoding settings **Mail settings** Protection settings Security settings CMS Security settings Deliver CMS GUI Settings Special settings

Mail settings

SMTP host name

SMTP port

Use SMTP authentication

User name

Password

System email sender email

Surveillance email receiver

Name / description
SMTP host name: Here you can define the mail server so InfoGlue can send e-mails.
SMTP port: Port to use (default 25).
Use SMTP authentication: If the mailserver requires authentication set this to Yes and fill in user name and password below.
User name: The user name for the mail server.
Password: The password for the mail server.
System email sender email: The email address which should be used as sender address.
Surveillance email receiver: The email address to which problems should be reported. OBS: Very important now for debugging situations – a lot of info comes here.

InfoGlue – Administrative Manual

Protection settings part 1

Common settings	Cache settings	CMS connection	Webserver settings	Encoding settings	Mail settings
Protection settings	Security settings CMS	Security settings Deliver	CMS GUI Settings	Special settings	
Protection settings					
Use password encryption	Yes				
Disable image editor	Inherit				
Protect content types	Inherit				
Protect workflows	Inherit				
Protect categories	Inherit				
Allow Publication event filter	Yes				
Hide restricted component properties	Inherit				
Default Publication event filter	All items for this site				
Protect working deliver	No				
Protect preview deliver	No				
Force identity check (for non protected pages)	Yes				
Enforce content access check (for extra hard security)	Inherit				
Allow submit to publish to propose other repositories pages and contents	Yes				
Max rows shown on content search					
Default number of years before content/pages expire	10				
Enable datefield editing	Yes				
Show content version first	Inherit				
Tree version					
Disable custom icons	No				
Show components first					
Show all workflows	Inherit				
Enable edit on sight	Inherit				

Name / description

Use password encryption: Since InfoGlue 2.8.0 we have added a possibility to use encrypted passwords in the database.

Disable image editor: Not used at the moment.

Protect content types: Set to yes if you want to access control what content types users should see. Take care as when you switch to Yes you will have to go through all content types and set rights. If enabled none of the content types are usable by anyone by default.

Protect workflows: Set to yes if you want to access control what workflows users should see. Take care as when you switch to Yes you will have to go through all workflows and set rights. If enabled none of the workflows are visible to anyone by default.

Protect categories: Set to yes if you want to access control what categories users should see. Take care as when you switch to Yes you will have to go through all categories and set rights. If enabled none of the categories are visible to anyone by default.

InfoGlue – Administrative Manual

Allow publication event filter: If set to yes the users are able to notify the publishers in their group about what they have sent to them. More on that feature in the user manual.
Hide restricted component properties: This setting allow you to state if the system should hide properties not editable by the user in the structuretool/component-editor.
Default publication event filter: Sets the default choice for the feature.
Protect working deliver: Set to yes if you want to force login to the working deliver. Usable to avoid people to authorized to peek at the working site.
Protect preview deliver: Set to yes if you want to force login to the preview deliver. Usable to avoid people to authorized to peek at the preview site.
Force identity check: States if the site should look if the user is logged into SSO even though the page does not require login.
Enforce content access check: Set that the site should enforce extra hard content access control. Some overhead but more checks for unauthorized reads.
Max rows shown on content search: How many found items should be shown after a search.
Allow submit to publish to propose other repositories pages and contents: If set to yes a user publishing a content or page are also suggested to publish items from other repositories as well.
Max rows shown on content search: How many found items should be shown after a search.
Default number of years before content/page expire: You can now control the default expire date jump which I proposed when you create new content/pages. Default is 50 years.
Enable datefield editing: You can choose to let users edit the publish/expire dates directly in the textfields as a complement to using the calendar. Useful sometimes.
Show content version first: When clicking on a content InfoGlue shows the master language version first instead of the content cover. If you don't want this behavior set this to "No".
Tree version: Allows you to change the tree implementation. Default the value is "html" but it can be "applet" as well.
Disable custom icons: Allows you to turn on/off the new feature which allows custom icons for each content type but is slow on some OS/Browsers. Defaults to true.
Show components first: By default InfoGlue shows the working version of a page when you click on a site node in the structure tool. If you rather want the page cover to be shown set this to "Yes"
Show all workflows: Set to Yes if you want the users to see all workflows even if they are not involved.
Enable edit on sight: No good reason to disable it completely but it's possible here.

Protection settings part 2

The screenshot displays the 'Protection settings part 2' configuration page. It features a light blue header and a main content area with five distinct sections, each with a title bar and a trash icon for deleting entries:

- Preview deliver url:** A single text input field.
- Staging deliver url:** A single text input field.
- Internal deliver urls (index based):** A large text area for multiple entries.
- Public deliver urls (index based):** A large text area for multiple entries.
- Deployment servers (index based):** A large text area for multiple entries.
- Available Tool languages (index based):** A large text area for multiple entries.

Name / description

Working deliver urls: Here you register all working delivery-engines. These [url:s](#) are used when the cms broadcasts cache messages to working and preview url:s. Format (index based):

0=http://xxx.xxx.xx/infoglueDeliverWorking

1=http://yyy.yyy.yy/infoglueDeliverPreview

Public deliver urls: Here you register all live delivery-engines. These [url:s](#) are used when publication cache messages are broadcasted. Format (index based):

0=http://xxx.xxx.xx/infoglueDeliverLive

1=http://yyy.yyy.yy/infoglueDeliverLive

Deployment servers: This is part of the new deployment architecture. You register all the deployment servers you want to communicate with here:

0=http://xxx.xxx.xx/infoglueCMS

1=http://yyy.yyy.yy/infoglueCMS

Available Tool languages: Here you can state what languages should be available in the cms tools. If you state only one language there will be no drop down in the tool header. Format based on ISO language code(index based):

0=sv

1=en

Protection settings part 2

Edition page size	<input type="text"/>	
Content tree sort	<input type="text"/>	
Structure tree sort	<input type="text"/>	
Structure tree hidden property	<input type="text"/>	
Hide protected contents in tree	Inherit	
Hide protected pages in tree	Inherit	
Disable empty urls	Inherit	
Cache update action	<input type="text"/>	
Log path	<input type="text"/>	
Log transactions	Inherit	
Enable extranet cookies	Inherit	
Use alternative language check (not recommended)	Inherit	
Case sensitive redirects	Inherit	
Use DNS name in URI	Inherit	
Extranet cookie timeout	<input type="text"/>	
Webservice base url	<input type="text"/>	
Live publication thread class	<input type="text"/>	
Publication thread delay	<input type="text"/>	
Paths to recache on publishing	<input type="text"/>	
Disable template debug	Inherit	
Export format (1 for original or 2 for new optimized)	<input type="text"/>	
Upload path	<input type="text"/>	

Name / description

Edition page size: Allows you to specify how many publications are shown at a time.

Content tree sort: Here you can state how the tree in content tool should be sorted. Normally sorted by content-name but you can state other attributes like id, publishDateTime or other content properties.

Structure tree sort: Here you can state how the tree in structure tool should be sorted. Normally sorted by site node name but you can state other sort-attributes as well. Like for content tree you can use id, publishDateTime and other SiteNode-properties but you can actually also use the site nodes meta info attributes by adding extra: first. This example sort first on pages meta info attribute sortOrder and if not there by name. "**extra:sortOrder,name**"

Structure tree hidden property: You can state if some meta info attribute should signal that the page is hidden.

Hide protected contents in tree: You can filter the tree so only accessible contents are shown in the content tree.

Hide protected pages in tree: You can filter the tree so only accessible pages are shown in the structure tree.

Cache update action: Don't set this yourself.

Log path: Not used at the moment.

Log transactions: Not used at the moment.

Enable extranet cookies: Do you want the users to be remembered through a cookie between sessions so they do not have to login again and again.

Use alternative language check: Do not change this.

Case sensitive redirects: By default the redirects are not case sensitive.

Use DNS name in URI: Set this to **Yes** if you want all [url:s](#) to be full – opposed to relative which are default.

InfoGlue – Administrative Manual

Extranet cookie timeout: Set the expiration period of the extranet cookie (if enabled).
Webservice base URL: If you build stuff where users can edit things from the site through the IG-webservices you must point out the services here.
Live publication thread class: Can be org.infoglue.deliver.util.SelectiveLivePublicationThread instead of the default and if set to this the live environments are also updated dynamically and selectively. Improves performance but is still in Beta-mode.
Publication thread delay: How long should the system wait until the publication process starts. This is used only when you have a setup where you replicate the database between the working and live servers and the replication can take a minute or two.
Paths to recache in publishing: Enter paths on the live servers that should be called after a publication. To increase performance and delay after a publication.
Disable template debug: If you don't wish the working deliver to output errors etc in the site you set this to Yes and only look at the logs for debug.
Export format: Since 2.4 InfoGlue's export and Import system is much more advanced and needs a new format. Only use 1 for old systems.
Upload path: Not used.

InfoGlue – Administrative Manual

Security settings CMS

These sections let's you define how authentication and authorization are handled in the cms. Most of the field names speaks for themselves but more information can be found under the chapter on Authentication and Authorization.

A new thing since 2.4.6 is that one can specify more than one role name as auth constraint for the tools. Just use the ";" as separator and complement with more names. For example one can state the constraint "cmsUser;helpdesk" to specify that users who have the role "cmsUser" and/or "helpdesk" are allowed. This only applies to the cms login of course – not to deliver logins.

The screenshot shows the 'Security settings CMS' configuration page. At the top, there are several tabs: 'Common settings', 'Cache settings', 'CMS connection', 'Webserver settings', 'Encoding settings', 'Mail settings', 'Protection settings', 'Security settings CMS' (which is selected), 'Security settings Deliver', 'CMS GUI Settings', and 'Special settings'. Below the tabs, the 'Security settings CMS' section is active. It contains a list of configuration fields, each with a text input and a trash icon for deletion:

- Login URL
- Logout URL
- Invalid login URL
- Success login base URL
- Authenticator class
- Authorizer class
- Server name
- Authentication constraint

Below these fields is a section titled 'Extra security parameters' with a large text area and a trash icon. At the bottom of the form, there are four more fields:

- CAS validate URL
- CAS Proxy validate URL
- CAS service URL
- CAS logout URL

At the very bottom of the page, there are three buttons: 'SAVE', 'SAVE & EXIT', and 'CANCEL'.

Security settings Deliver:

These sections let's you define how authentication and authorization are handled in the delivery engine (extranet for example). Most of the field names speaks for themselves but more information can be found under the chapter on Authentication and Authorization.

The screenshot shows the 'Security settings Deliver' configuration window. At the top, there are tabs for 'Common settings', 'Cache settings', 'CMS connection', 'Webserver settings', 'Encoding settings', and 'Mail settings'. Below these are 'Protection settings', 'Security settings CMS', and 'Security settings Deliver' (which is selected). The 'Security settings Deliver' tab contains a sub-tab 'Security settings Deliver'. The main area has a list of settings, each with a text input field and a trash icon for deletion:

- Login URL
- Logout URL
- Invalid login URL
- Success login base URL
- Authenticator class
- Authorizer class
- Server name
- Authentication constraint
- Anonymous username (anonymous default)
- Anonymous password (anonymous default)

Below these is a section for 'Extra security parameters' with a large text area and a trash icon. At the bottom, there are four more settings, each with a text input field and a trash icon:

- CAS validate URL
- CAS Proxy validate URL
- CAS service URL
- CAS logout URL

At the very bottom, there are three buttons: 'SAVE', 'SAVE & EXIT', and 'CANCEL'.

InfoGlue – Administrative Manual

CMS GUI settings

These sections let's you define shortcuts etc.

Common settings Cache settings CMS connection Webserver settings
Encoding settings Mail settings Protection settings Security settings CMS
Security settings Deliver **CMS GUI Settings** Special settings

CMS GUI Settings

Tree item working style information

Tree item final style information

Tree item publish style information

Tree item published style information

Custom content type definitions icons (for tree)

```
3=/infoglueCMS/images/icons/articleIcon.png
3_folderIcon=/infoglueCMS/images/icons/articleFolderIcon.png
3_openFolderIcon=/infoglueCMS/images/icons/articleOpenFolderIcon.png
3_protectedFolderIcon=/infoglueCMS/images/icons/articleProtectedFolderIcon.png
3_protectedOpenFolderIcon=/infoglueCMS/images/icons/articleProtectedOpenFolderIcon.png
3_protectedIcon=/infoglueCMS/images/icons/articleProtectedIcon.png
```

Name / description

Tree item working style information: State any css-inline-style you wish to use to indicate that the content/page is in working state in the content/structure tree.

Tree item final style information: State any css-inline-style you wish to use to indicate that the content/page is in final state in the content/structure tree. **Not used.**

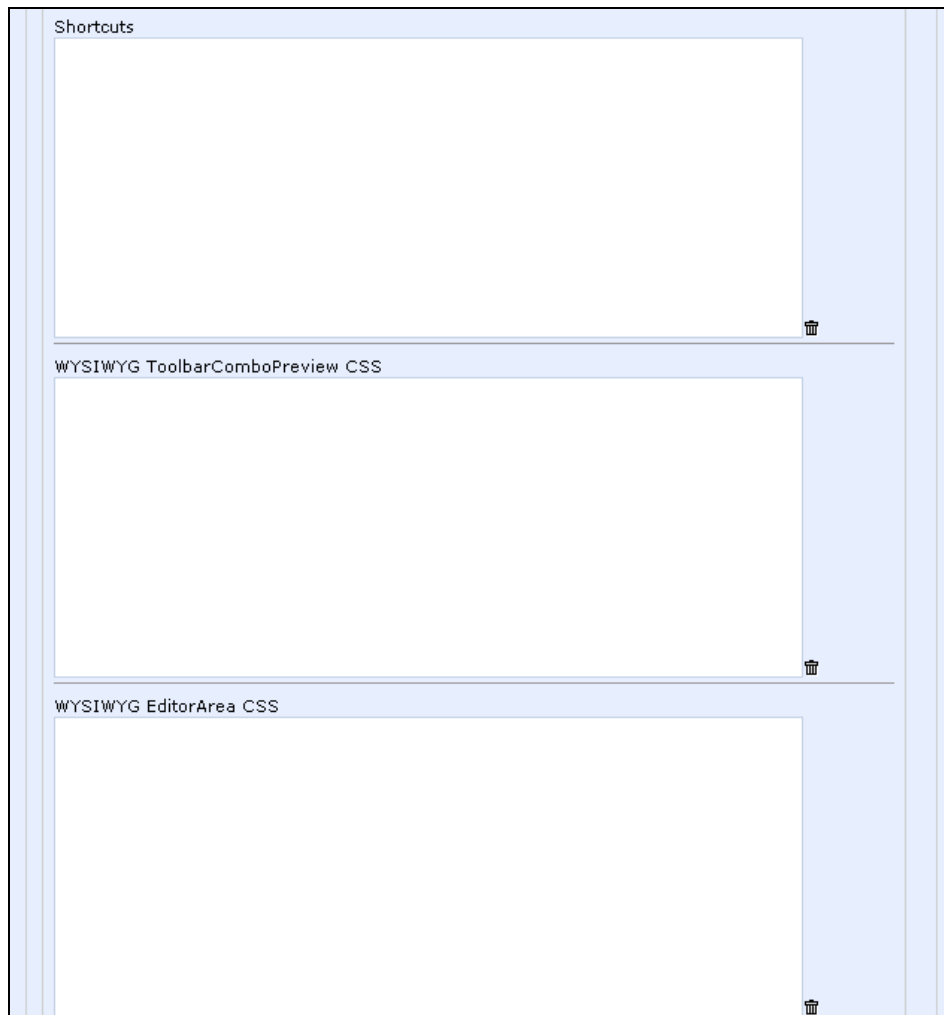
Tree item publish style information: State any css-inline-style you wish to use to indicate that the content/page is in published state in the content/structure tree.

Tree item published style information: State any css-inline-style you wish to use to indicate that the content/page is in published state in the content/structure tree.

Custom content type definitions icons: In the example below we have stated urls to icons for different states of the content type with id 3 (the id can be found in the management tool). OBS: You will have to deploy the custom icon files yourself to the path you reference. The system does not include any deploy mechanism for this.

```
3=/infoglueCMS/images/icons/articleIcon.png
3_folderIcon=/infoglueCMS/images/icons/articleFolderIcon.png
3_openFolderIcon=/infoglueCMS/images/icons/articleOpenFolderIcon.png
3_protectedFolderIcon=/infoglueCMS/images/icons/articleProtectedFolderIcon.png
3_protectedOpenFolderIcon=/infoglueCMS/images/icons/articleProtectedOpenFolderIcon.png
3_protectedIcon=/infoglueCMS/images/icons/articleProtectedIcon.png
```

InfoGlue – Administrative Manual



Name / description
<p>Shortcuts: Here you can define application/task shortcuts which will be shown in the myDesktop tool.</p> <p>The format is xml-based and here is an example:</p> <pre><?xml version='1.0' encoding='UTF-8'?> <shortcuts> <shortcut> <name>InfoGlue Official homepage</name> <url>http://www.infoglue.org</url> <popup>true</popup> </shortcut> </shortcuts></pre>
<p>WYSIWYG ToolbarComboPreview CSS: Here you enter a CSS for how the styles should look in the toolbars in FCKEditor. Usually it's nice if you state the same classes as the external site uses so users feel at home when selecting formats etc in the drops.</p>
<p>WYSIWYG EditorArea CSS: Here you enter the CSS you wish the FCKEditor to use when you edit. Usually it's nice if you state the same classes as the external site uses so users feel at home.</p>

InfoGlue – Administrative Manual

Special settings

This section defines some settings which is kind of deprecated or only for advanced users.

Name / description
Use Freemaker as the default template engine instead of velocity: Tells InfoGlue to use freemaker as the default template engine if not JSP was declared in the component/template. The developer docs explains in details how this works.
Busy page URL: Not used at the moment.
External thumbnail generator: The full path to image magic you have that installed and want to use it rather than the ordinary Java2D-resizing.
Enable JSR-168: Controls if you want the possibility to deploy portlets. Should probably always be set to "Yes" as we don't see any downside to have it enabled even though you don't use portlets at the moment.
Portlet base: Should point to the tomcat home directory at the moment as Apache Pluto demands this.
Log database transactions: Not used at the moment.
Enable site statistics: Not used at the moment.
Statistics logs path: Not used at the moment.
One statistics log file per day: Not used at the moment.
Statistics logger mode: Not used at the moment.
Use improved content category search (default yes): Should always be true.
DB Release: Not used
DB Release: Not used
DB User: Not used
DB Password: Not used
Master server: Not used
Slave server: Not used
Build name: Not used
Admin tools path: Not used
Db script path: Not used

InfoGlue – Administrative Manual

Configuration files

All the configuration files are located under WEB-INF/classes under respective webapp, both cms and deliver. There is also a web.xml for each application which contains some settings for filters. The most central files are:

cms.properties	Used only by the infoglueCMS-webapp although located in all applications. Contains most of the settings for the CMS-tools. Described in a separate section.
deliver.properties	Used only by the infoglueDeliver-webapp although located in all applications. Contains most of the settings for the Deliver-engines. Described in a separate section.
database.xml	This file contains the database connection settings used by the O/R-mapper Castor which InfoGlue uses for all interaction with the database.
hibernate.cfg.xml	This file contains the database connection settings used by the O/R-mapper Hibernate which InfoGlue uses for all interaction with the database concerning Workflows.
propertyset.xml	This file contains all setting used by the property-framework OS PropertySet which InfoGlue uses for some properties.
oscache.properties	This file contains cache settings for the cache-engine OSCache which InfoGlue uses for page-caching and more further on.
log4j.xml	The log4j settings file for InfoGlue. Controls logging completely.
webwork.properties	This file contains some minor settings for how webwork behaves. Most important the maximum file upload size allowed by WebWork.
quartz.properties	This file contains settings for the scheduler Quartz.
jobs.xml	Contains all jobs defined that are to be scheduled to run at certain times. Contains a cleanup job for example but are meant to be filled with custom jobs in an installation.
JNDIParameters.properties	Settings to use when authenticating with custom authentication and authorization frameworks. More on that later.
InfoGlueAuthorizationParameters.properties	Settings to use when authenticating with custom authentication and authorization frameworks. More on that later.

CMS-application is controlled in the cms.properties

Below "infoglueCMS{suffix}/WEB-INF/classes" there are a file called "cms.properties". It contains all settings for the CMS-application if not defined in the application settings in management tool and the possible setting are listed below:

Cache settings for the component editor environment		
Field name	Typical values	Comment
isPageCacheOn	true / false	Allways use true except for debugging purposes or for a completely dynamic site.
expireCacheAutomatically	true / false	false is default.
cacheExpireInterval	1800000	Only set this value if you set expireCacheAutomatically=true – should not be used in cms really
session.timeout	1800	This sets the timeout for the user session in the tools. Defaults to 30 minutes (1800 seconds)
WEB-SPECIFIC SETTINGS (USED TO LOCATE IMAGES ETC)		
Field name	Typical values	Comment
webServerAddress	http://localhost:8080	Never use this really except if you know what you are doing.
digitalAssetBaseUrl	infoglueCMS/digitalAssets	Defaults to the dir under the webapp.
imagesBaseUrl	infoglueCMS/images	Only set this value if you set expireCacheAutomatically=true – should not be used in cms really
digitalAssetPath	/usr/local/tomcat/webapps/infoglueCMS/digitalAssets	Defaults to the webapp dir – could be set to a SAN or something.
Thumbnail generation		
Field name	Typical values	Comment
externalThumbnailGeneration		Could point to image magic if installed.
Encoding settings		
Field name	Typical values	Comment
URIEncoding	UTF-8	Descides in which way the urls in the tools are generated. If you have internation characters in role/user or group names this will be important.
Logging		
Field name	Typical values	Comment
logTransactions	true / false	This descides if to log activity or now into the table cmTransactionHistory. Default true.
logDatabaseMessages	true / false	Old setting – not used any more.
logToFile	true / false	false is default – old setting we now uses log4j.
logPath	/tmp/logs/tool.log	Old setting
logLevel	INFO, WARNING, ERROR	Old setting
Preview settings		
Field name	Typical values	Comment
previewDeliveryUrl	An url to the working version of InfoGlue.	Used allways except if NiceUri:s are configured with a working-url.
stagingDeliveryUrl	An url to the preview	Used allways except if NiceUri:s are

InfoGlue – Administrative Manual

	version of InfoGlue.	configured with a working-url.
Replication / Synchronization messaging		
Field name	Typical values	Comment
internalDeliverUrl.0	An url to the deliver context.	The index starts with 0 but can be any number of additional urls.
internalDeliverUrl.x		
Publication settings		
Field name	Typical values	Comment
publicDeliverUrl.0	An url to the public deliver context.	The index starts with 0 but can be any number of additional urls.
publicDeliverUrl.x		
Cache update settings		
Field name	Typical values	Comment
cacheUpdateAction	UpdateCache.action	Allways use this.
Language settings		
Field name	Typical values	Comment
0.toolLanguageCode	sv, en, fr	The index first starts with 0 but can have any number of additional languages.
x.toolLanguageCode		
Tree settings		
Field name	Typical values	Comment
Tree	Applet/html	Sets if the system should use the applet or the html tree.
treemode	Dynamic/static	The default is the new dynamic tree.
content.tree.sort	Id, name etc	The sort order in the tree
structure.tree.sort	Id, name etc	The sort order in the tree
Misc		
Field name	Typical values	Comment
showContentVersionFirst	true / false	Decides if to show the master content version directly upon selection of a content.
showComponentsFirst	true / false	Decides if to show the page component view directly upon selection of a sitenode.
protectContentTypes	true/false	Selects if to protect content types or not from visibility to all users.
protectCategories	true/false	Selects if to protect categories or not from visibility to all users.
wysiwygEditor	FCKEditor / HTMLArea	Selects which html-editor to use.
helpUrl		The url the help-button opens.
Search settings		
Field name	Typical values	Comment
maxRows	100	How many rows should be shown in search.
Publishing settings		
Field name	Typical values	Comment
edition.pageSize	10	How many editions should be visible in the publishing tool.
Database setting		
Field name	Typical values	Comment
useShortTableNames	false	SET THIS TO TRUE IF YOU ARE

InfoGlue – Administrative Manual

		RUNNING ORACLE OR DB2.
masterServer	cms.infoglue.org	Set this value only if you are running in a replicated mysql-environment.
slaveServer	www.infoglue.org	Set this value only if you are running in a replicated mysql-environment.
Mail settings		
Field name	Typical values	Comment
mail.smtp.host		The mail server
mail.smtp.auth		If to use authentication
mail.smtp.user		Username if using auth
mail.smtp.password		Password if using auth
systemEmailSender		The system email address, sender.
Update settings		
Field name	Typical values	Comment
up2dateUrl		Do not change
Component editor settings		
Field name	Typical values	Comment
componentRendererUrl		Do not change
componentRendererAction		Do not change
operatingMode		Do not change.
Superuser settings		
Field name	Typical values	Comment
administratorUserName		The root account name
administratorPassword		The root account password
administratorEmail		The root account email.

There are a few other settings in the file but those are not to be changed.

Deliver-application is controlled in the deliver.properties

Below "infoglueDeliver{Working/Preview/Live}{suffix}/WEB-INF/classes" there are a file called "deliver.properties". It contains all settings for the Deliver-application if not specified in the application settings in management tool and the possible setting are listed below:

Cache settings for the component editor environment		
Field name	Typical values	Comment
isPageCacheOn	true / false	Allways use true except for debugging purposes
useSelectivePageCacheUpdate	true / false	This sets so only affected pages are discarded during updates, can be set to false for debugging purposes.
session.timeout	3600	Sets the timeout on the extranet sessions.
expireCacheAutomatically	true / false	false is default.
cacheExpireInterval	1800000	Only set this value if you set expireCacheAutomatically=true – should not be used in cms really
Operating mode		
Field name	Typical values	Comment
operatingMode	0, 2 or 3	This sets 0 for working, 2 for preview and 3 for live.
Edit On Sight		
Field name	Typical values	Comment
editOnSite	True / false	
editOnSiteUrl		
WEB-SPECIFIC SETTINGS (USED TO LOCATE IMAGES ETC)		
Field name	Typical values	Comment
webServerAddress		
applicationBaseAction		
digitalAssetBaseUrl		
imagesBaseUrl		
digitalAssetPath		
enableNiceURI		
niceURIEncoding		
disableEmptyUrls		
requestArgumentDelimiter		
Encoding settings		
Field name	Typical values	Comment
URIEncoding	UTF-8	Descides in which way the urls in the tools are generated. If you have internation characters in role/user or group names this will be important.
Logging		
Field name	Typical values	Comment
logDatabaseMessages	true / false	Allways use true except for debugging purposes
logToConsole	true / false	false is default.
logToFile	true / false	false is default.
logPath		
logLevel		
statisticsLogPath		

InfoGlue – Administrative Manual

statisticsLogOneFilePerDay		
statisticsLogger		
Database setting		
Field name	Typical values	Comment
useShortTableNames	false	SET THIS TO TRUE IF YOU ARE RUNNING ORACLE OR DB2.
Portal setting		
Field name	Typical values	Comment
enablePortal	true/false	
portletBase		
Mail settings		
Field name	Typical values	Comment
mail.smtp.host		
mail.smtp.auth		
mail.smtp.user		
mail.smtp.password		
systemEmailSender		
Update settings		
Field name	Typical values	Comment
up2dateUrl		
Component editor settings		
Field name	Typical values	Comment
componentEditorUrl		
componentRendererUrl		
componentRendererAction		

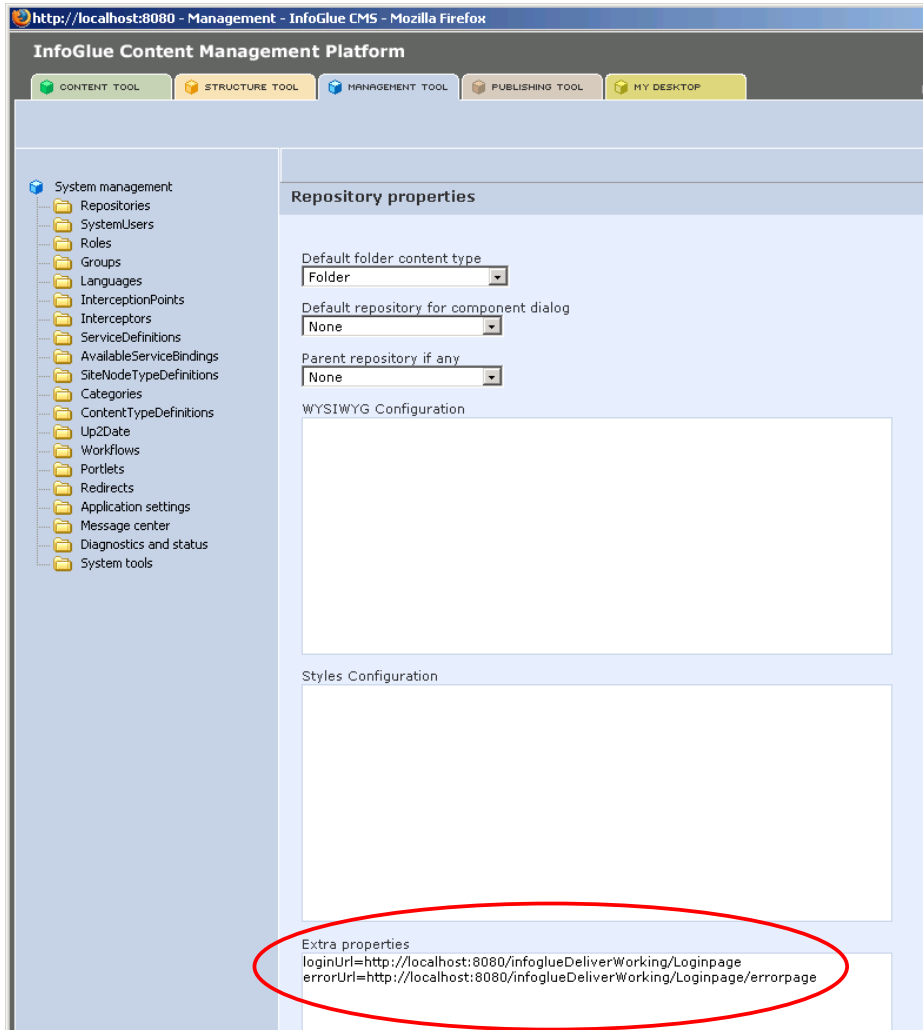
InfoGlue – Administrative Manual

Repository-specific error and login pages

Since InfoGlue 2.6.0 it is possible to define different error and login pages for each repository you have in the installation. It can of course fall back to the default screens but if you wish to specify it just add the properties in the repository detail view:

Enter the "Repository properties"-view and enter a name-value pair in the Extra properties field:

```
loginUrl=http://mysite.com/loginPage  
errorUrl=http://mysite.com/errorpage
```



InfoGlue – Administrative Manual

The InfoGlue applications

InfoGlue follows the ordinary J2EE standard for web applications in it's file system layout. See more in the servlet specification available at <http://java.sun.com/products/servlet/download.html>.

InfoGlue does not have to put files anywhere else unless you want the JSR 168 support (Portlets). Then it needs to insert some components in the shared class loading area of the runner. In tomcat's case the installer copies two files from Apache Pluto into the shared/lib folder.

Integration between InfoGlue parts

Cache updates and broadcasts

InfoGlue uses caching heavily. It not only can cache whole pages or parts of pages but also caches most of the model in memory. This gives a superior performance compared to querying the database all the time. The downside is that there needs to be some kind of notifications between the different applications or a common cache. While it is possible to achieve a common cache InfoGlue by default does not support it. Instead we use broadcasts from the CMS-application which is the only app that are to write data to the InfoGlue-tables and the delivery applications. So when a change has been written a simple HTTP-request goes out to all deliver instances and tells them to update what was changed.

Which instances to call are defined in the cms.properties file and if you add a new instance of infoglueDeliverXXX you must also add a reference to it in the appropriate section there. It is described in detail in the cms.properties section.

This is the first place to look if you feel your caches are not updating and reflecting the changes made. If a restart of the appserver results in the right info being shown 99% of the times the reference to the deliver instance in cms.properties are wrong or missing. Remember to test so the address you type actually reached the application. You should try it with a simple wget or similar.

A simple check is to look at the status page on **/infoglueDeliverXXX/ViewApplicationState.action** and see if there is a lot of things cached. If the information comes up on the site after you press clear all caches in that view you probably have a configuration problem.

InfoGlue Deliver applications calling CMS

In InfoGlue one can of course want to insert content or edit content from forms etc in the live site. For example one perhaps would want to build a simple blogger-tool as a site. That is fine and InfoGlue only requires that the one doing the saves are the cms-tools as they have to be the master at all times so no business rules are broken. This is achieved by using the InfoGlue Web services API:s provided over SOAP by Apache Axis. The deliver application and the site templates/logic calls the internal cms api:s over HTTP and inserts content or whatever that way. The only important thing when using such API:s is that you have an infrastructure that allows the deliver apps to reach the cms-application over HTTP.

Databases and maintenance

As all databases are a bit different in how they should be maintained we will not try to describe this here. What we will do instead is to discuss different areas which should be discussed or addressed when set up by a DBA.

Setting up the database

The InfoGlue installer supports creating the database for MySQL and SQLServer while it does not for Oracle and DB2. This is because most DBAs have told us it is not a good idea. If that is so is probably dependent on how the installation of InfoGlue will operate and how much data and transactions will burden the database. So – if you feel that you want some custom setup for the InfoGlue database you should create the schema and tables first and only let the installer insert the needed data.

What DBAs have told me is to take special care when dealing with tables which will grow fast in size as table spaces etc should be dimensioned accordingly. Not being a DBA we only suggest you understand what tables will contain much data and change often and make your own decisions based on that.

Key tables:

cmContent – will probably never reach over 10.000 rows for a medium site and there will be no especially long data types for your regular site.

cmContentVersion – could reach up to 100.000 rows for a regular site and it contains a clob which represents the actual texts written and presented on the site so it could contain a fair amount of data. Perhaps up to a one or two hundred megabytes after a while.

cmDigitalAsset – this is the heaviest table as it's only for storing blobs. So all pdf-files, programs, images and other binary content the users wish to publish through InfoGlue will be stored there. If a site will contain a lot of high-res images or movies the size will quickly be very large. We have seen everything from 10Mb in total to over 100Gb and growing.

cmTransactionHistory – this table will contain a record of all writes to the database (create/update/delete). The amount of rows will be very big but the size in Kb will not. It is also possible to archive and truncate this once in a while to save space.

Separation and scalability on the databases.

In a high profile site with a high workload there are often a need to separate the live database and the cms-database. This will relieve the live sites database from any transactions done in the working area of the tools leaving that database instance to only serve visitors to the sites. It will also speed up the internal tools as the live sites traffic will not slow down the administrative tools. It is also excellent from a maintenance point of view as you then can upgrade the cms-tools including database schema without bringing down the live sites as well if you turn off replication during that time.

We will not discuss how to set up such a separated environment with replication here but for people wanting to replicate between two MySQL servers they should look at <http://dev.mysql.com/doc/mysql/en/replication-howto.html>.

Other database vendors have similar possibilities which your DBA can tell you about.

To achieve this you probably also have separated the servers which deliver the applications and then you only need to install the InfoGlue deliver live on the live server

InfoGlue – Administrative Manual

and point it to the replicated database instead of to the master db. You can do it yourself afterwards as well by modifying the following files jdbc-url under infoGlueDeliverLive:

database.xml	This file contains the database connection settings used by the O/R-mapper Castor which InfoGlue uses for all interaction with the database.
hibernate.cfg.xml	This file contains the database connection settings used by the O/R-mapper Hibernate which InfoGlue uses for all interaction with the database concerning Workflows.
propertyset.xml	This file contains all setting used by the property-framework OS PropertySet which InfoGlue uses for some properties.

Scaling the applications only

There are cases when you do not wish to scale on the database side by setting up replication or a clusters but instead want to share database but split the applications on different servers. This can be very effective in the following situations:

- If you for example have a low transaction public site but with extreme load and then a large number of content editors putting a lot of load on the tool server.
- It can also be of interest if you just want to separate the apps out of mainainence reasons. Perhaps you do not want to take down the public site just because you have to upgrade the tools and need to restart tomcat in order to do so.

If this seems attractive to you below are some general guidelines to how to set this up:

1. Install the cms, infoglueDeliverWorking and infoglueDeliverPreview on server1. Let the installer create the database if you don't already have it set up. When it comes to the db just remember to put it on a server which both the internal and external server will be able to reach. After that normal installation make sure the apps are working correctly and that the working and preview engines gets updated by the tools when changes occur.
2. Install the infoglueDeliverLive only on server2 using the same installer. When the database questions come point it to the database set up in step 1 or earlier. Do not tick in so it creates anything in the db.
3. Go into server1 and the cms-applications cms.properties and edit/add the property publicDeliverUrl.x (where x is 0,1,2,3 etc for as many live servers you have). The url should be a network acceptable url to the deliver application **which is reachable from the cms**. An example could look like this:
publicDeliverUrl.0=http://server2.mycomp.com:8080/infoglueDeliverLive or similar depending on what your setup and servers look like.
4. After a restart of the cms application it will now notify the live server whenever a new publication is made either through the publishing tool or through the direct publishing feature.

If you have nice uri enabled you should also look at that section to find out what more needs to be set up for that to work on the live server. If you have trouble with the live server not getting updated by the cms look in the tips section in this document.

Portlet (JSR 168) Support and pitfalls

InfoGlue supports JSR 168 by including Apache Pluto and integrating it in the InfoGlue core. Apache Pluto is very good but there are a few pitfalls most users end up investigating.

- InfoGlue Deliver and CMS must be located under the tomcat webapps directory directly if portlets are to be deployed. That is a limitation of Pluto today.
- There must never be more than one of pluto-1.0.1.jar and portlet-api-1.0.jar and they must be located in the shared lib directory under tomcat. If one of them are duplicated in one of the webapps Pluto will not work correctly and have trouble finding the portlets.
- If you want to share data between your portlet and InfoGlue you have to have a context for the portlet with the attribute crosscontext="true" as well as for the InfoGlue applications which is default. Otherwise it won't be visible for each other.

How to deploy new portlets and how to use them as components are described in the new InfoGlue User Manual.

Nice URI

One of the most important features in InfoGlue is Nice-URI or friendly URL's. What it's all about is that instead of referencing a page with a cryptic parameter string like in /ViewPage.action?siteNodeId=22&languageId=11&contentId=-1 your users can find the page by using /Products/Boats or some other understandable URL.

How does it work

Nice URI can work as InfoGlue tries to match all incoming Uri's to a particular site Node/page. All [URL:s](#) created are also translated to a readable URL in the reverse way. This is not that easy and there are two tricks to this:

The first trick is for InfoGlue to find out which repository the page the user tries to look at is located in (since there may be many repositories). In theory several sites can have the same page structure and therefore InfoGlue would not know where to go to find /contact/about_us for example if both repository1 and repository2 has the same structure. This is solved by using different dns-names for all repositories. So if you have two repositories, InfoGlue will first match the name of the dns coming in from the users browser request (<http://www.infoglue.org> for example) against the dnsNames registered on each repository and pick the repository which has the first match. So if the repository "InfoGlue" should be picked it has to have www.infoglue.org in the dnsNames-attribute while you use other names for other repositories.

The second trick is for InfoGlue to find the pages. The /products/boats path given in the browser is matched against the site structure (defined in the structure tool) for that repository. It is possible to set which attribute on each site Node to match against.

Limitations and pitfalls

Encoding

As urls are not supporting anything else than ASCII in them any pages with international characters in them needs to be encoded/decoded. This works if you set up InfoGlue NiceURI encoding correctly to fit your platform but the [url:s](#) are far from readable as the Uri's will contain a lot of special characters etc.

InfoGlue – Administrative Manual

Setup Instructions

This part assumes that Nice URI is turned on – look at the settings section for info.

For each repository create a dnsName for each webapp (working/preview/live). This may require you to register quite some dns-entries if you want all sites to have a unique dns name and have nice uri enabled in both working and live. It may be a better idea to use paths for the different repositories in working which is also supported.

As an example we would have the following dns-names for www.infoglue.org pointing to the same machine:

The main site

- www.infoglue.org
- preview.infoglue.org
- working.infoglue.org

A marketplace site

- market.infoglue.org
- preview.market.infoglue.org
- working.market.infoglue.org

Enter the dnsNames for each repository. Completing the example we would add the following string to dnsName for the main www.infoglue.org-repository:

```
working=working.infoglue.org,preview=preview.infoglue.org,live=www.infoglue.org
```

For the marketplace repository the dnsName-string would be:

```
working=working.market.infoglue.org,preview=preview.market.infoglue.org,live=market.infoglue.org
```

Luckily you can use the form with assistance so you don't have to remember the cryptic format. Also – there is a great "Explain"-link next to the DNSName-field which explains your current settings and their effect. The working=, preview= and live= - arguments are just for the tools to know what URL to use when launching the site from different places in the administrative interfaces. For example – the preview button on a site node in the www.infoglue.org-repository will check out the working=-keyword and launch working.infoglue.org thereby letting InfoGlue find the right repository in NiceURI-mode. You can also add more entries but without any keyword before. If you for example add www.infoglue.com last to the www.infoglue.org-repository people coming in with that domain will come to the right site so you can have many aliases for the same site but only one "working=", "preview=" and "live=" keyword.

Since InfoGlue 2.4 we have added a feature which allows a simple way to map several repositories to a single domain. So www.infoglue.org could serve X number of repositories where each repository would have to have a subdir like www.infoglue.org/market or www.infoglue.org/plugins.

Now the users calling <http://www.infoglue.org/marketplace/Components> will come to the component-page directly under the marketplace repository.

NiceURI are pretty flexible in 2.0. It allows you to influence the algorithm for creating the Uri's pretty much. By default the name used in the Uri's are the NavigationTitle for the sitenodes involved but you can easily set it to use for example the site node name(the non-localized one in structure tree) or a node properties attribute of your choice. You set this up in application settings.

Authentication and Authorization

InfoGlue has a very modular authentication and authorization system. The first thing to understand is the difference between Authentication which is about identifying who a user are during login etc and Authorization which is about deciding what the authenticated user has right to see/do in the system. When it comes to the latter InfoGlue supports setting access rights on user level as always it's recommended using roles and groups for access control to avoid cumbersome maintenance.

InfoGlue supports custom modules for both Authentication and Authorization. By default InfoGlue is installed with the module that uses the infoglue-database for both authentication and authorization. That means that when a user logs on his/her login info is compared to the cmSystemUser table and associated tables by the default Authentication Module and then access rights are controlled by the InfoGlue Authorization Module against cmSystemUser, cmRole and cmGroup tables. This is all fine in some situations but sometimes the organization using InfoGlue would want to reuse the users/roles/groups available in their global user directory. Many organisations also has some kind of Single Sign On system they want to use. This is why you in InfoGlue can set up references to other Authentication/Authorization modules and even write your own if the ones shipped are not enough. These moduls are included:

InfoGlue Basic Authentication / Authorization

InfoGlue itself can act as the source of authentication and authorization. This is the default setup.

SSO through CAS

Apart from the InfoGlue default authentication module InfoGlue ships with a single sign on module written to support CAS(Central Authentication Service). CAS is a open source authentication system developed by the university world. Those of you who have worked with CAS will recognize the validate-service etc under application settings and hopefully know how to configure it. Later a more detailed example will be posted.

J2EE Authentication

If InfoGlue runs inside a J2EE container InfoGlue can be told to operate with J2EE-security and accept the principal coming in from the container.

JDBC Authorization

An module which can be configured against any JDBC-datasource. Queries can be defined in extra security settings.

LDAP

Both Authentication and Authorization can be directed to use the included LDAP-modules. They were written for integration with Active Directory but should work with many more LDAP-servers. These modules let you use users/groups and roles in InfoGlue which are fetched directly from your LDAP-directory thereby reusing that information and avoiding redundant information. The lookups are made online so there is no periodic export into InfoGlue or anything like that. In InfoGlue there is no difference between users/roles/groups coming from InfoGlue's own user-db or from an external resource.

Several combinations

There are also combinations of these modules.

List of modules

Here is a full list of authentication modules available:

- `org.infoglue.cms.security.InfoGlueBasicAuthenticationModule` (default)
Uses the InfoGlue database for authentication.
- `org.infoglue.cms.security.CASBasicAuthenticationModule`
This module makes it possible to use CAS (Central Authentication Services) authentication both to cms and / or deliver. Open source SSO solution.
- `org.infoglue.cms.security.JNDIBasicAuthenticationModule`
Uses an LDAP for authentication.
- `org.infoglue.cms.security.InfoGlueJ2EEAuthenticationModule`
An authentication module which trusts the ordinary J2EE container authentication. That means it accepts a user found in the request as authenticated.
- `org.infoglue.cms.security.CombinedJNDIBasicAuthenticationModule`
Can be used to combine an LDAP authentication and the ordinary InfoGlue authentication module as a fallback.
- `org.infoglue.cms.security.InfoGlueExtranetAuthenticationModule` (not used).

Here is a full list of authorization modules available:

- `org.infoglue.cms.security.InfoGlueBasicAuthorizationModule` (default)
Uses the InfoGlue database for users, roles and groups.
- `org.infoglue.cms.security.GenericCombinedAuthorizationModule` (new)
Allows a user to combine any number of other authorization modules. So some users/roles/groups can come from InfoGlue, some from LDAP servers and so on. The limitation is that a user can only have roles / groups from the same adaptor.
- `org.infoglue.cms.security.SimplifiedJNDIBasicAuthorizationModule` (new)
From now on the default LDAP-adaptor. Very flexible and configurable.
- `org.infoglue.cms.security.WebServiceAuthorizationModule` (new)
The new SOAP-based (Apache Axis) webservice way of getting all authorization information.
- `org.infoglue.cms.security.InfoGlueJDBCAuthorizationModule`
Allows you to have almost any JDBC source as the authentication source.
- `org.infoglue.cms.security.CombinedJDBCBasicAuthorizationModule`
Allows you to combine the JDBC and basic authorization modules.
- `org.infoglue.cms.security.CombinedJNDIBasicAuthorizationModule`
Allows you to combine the JDBC and basic authorization modules.
- `org.infoglue.cms.security.JNDIBasicAuthorizationModule` (use the `SimplifiedJNDIBasicAuthorizationModule` module instead)

How to combine several authorization modules

Quite often one wants to combine several different authorization modules. For example we often have infoglue's own database and combine it with one or several LDAP/JNDI modules. The benefit is that you can have as many sources as you like and InfoGlue will combine the results. An example is when a customer wants all external users in one database/LDAP or similar and the internal users should be fetched from one or more internal sources.

To make this work we have created a `GenericCombinedAuthorizationModule`. It doesn't do much more than orchestrating the modules involved. The way to set up authorization to make this work is this:

In application settings:

- Set "Authenticator class" to `org.infoglue.cms.security.GenericCombinedAuthorizationModule`
- Add the following first in the "Extra security parameters" field.

```
#####  
#Properties for the generic combined auth  
#####  
0.authorizerClassName=org.infoglue.cms.security.SimplifiedJNDIBasicAuthorizationModule  
1.authorizerClassName=org.infoglue.cms.security.SimplifiedJNDIBasicAuthorizationModule  
2.authorizerClassName=org.infoglue.cms.security.InfoGlueBasicAuthorizationModule
```

- The syntax is simple and index-based. You just add as many modules by setting up a index starting with 0. Then you state what module corresponds to each index like in the example. The example means I will have 3 modules activated. First two JNDI-modules which I use to connect and fetch users, roles and groups from two different LDAP's. The third is the common InfoGlue module so this setup will combine all three.
- When using this module this way it's quite normal to ask how to set up the different properties for the same module. It is simple. If you have a property which you need to diversify between two modules of the same kind just state "{index}." In front of the property and the underlying module will get just that property. So for example "0.connectionURL=someURL" in extra security parameters will be fed to only the first `SimplifiedJNDIBasicAuthorizationModule` and the property will, when it reaches that module, be stripped of the index info.

InfoGlue – Administrative Manual

How to set up LDAP

When deciding to use LDAP as the resource for authorization and/or authentication there are a lot of configuration to be considered. We could have decided upon a directory structure the users have to adjust to but instead we choose to make it configurable instead so InfoGlue can adjust to the users existing setup. So InfoGlue now asks a couple of different questions to the LDAP to get users, roles and groups or when authenticating to log a user on. Below is an example configuration which enables JNDI as both auth and authorization for CMS (the same exists for deliver further down that screen). The red area is where you paste your ldap config. It is described in detail later on in this chapter.

The settings are entered in management tool under application settings:

The screenshot shows the 'InfoGlue Content Management Platform' management tool. The left sidebar contains a tree view with categories like 'System management', 'Repositories', 'SystemUsers', 'Roles', 'Groups', 'Languages', 'InterceptionPoints', 'Interceptors', 'ServiceDefinitions', 'AvailableServiceBindings', 'SiteNodeTypeDefinitions', 'Categories', 'ContentTypeDefinitions', 'Up2Date', 'Validation', 'Workflows', 'Portlets', 'Redirects', 'Application settings', 'Message center', and 'Diagnostics and status'. The main panel is titled 'ServerNode properties' and contains a 'Security settings CMS' section. This section includes fields for 'Login URL', 'Login.action', 'Logout URL', 'Login!logout.action', 'Invalid login URL', 'Login!invalidLogin.action', 'Success login base URL', 'inherit', 'Authenticator class', 'org.infoglue.cms.security.JNDIBasicAuthorizationModule', 'Authorizer class', 'org.infoglue.cms.security.JNDIBasicAuthorizationModule', 'Server name', 'inherit', 'Authentication constraint', and 'cmsUser'. Below these is the 'Extra security parameters' field, which is highlighted with a red box and contains the following text:

```
#####  
# Roles settings section  
#  
# This part is used by InfoGlue JNDI module to get information about roles which  
# are to  
# be used in InfoGlue. The most striking place is to look under roles in management  
# tool.  
#####  
  
#####  
# Role base - this states which part of the directory structure we want to search for  
# roles  
#####  
roleBase=dc=infoglue,dc=org  
#####
```

Below the red box, there are fields for 'CAS validate URL', 'https://www.infoglue.org:8433/cas/serviceValidate', 'CAS service URL', 'https://www.infoglue.org/infoglueCMS24/ViewCMSTool', 'CAS logout URL', and 'https://www.infoglue.org:8433/cas/logout'. The footer of the application shows 'InfoGlue CMS (EN) Version build: 2.4 Final (2006-11-13)' and 'Klar'.

Detailed description of JNDI-configs

Sample config for JNDIBasicAuthorizationModule

```
#####
# Properties for the JNDI SETUP
#####

#####
# Authentication method - can be "none" or "simple".
#
# "simple" binds a user
# "none" works only if anonymous bind is allowed.
#####
authenticationMethod=simple

#####
# LDAP-version (2 or 3)
#####
ldapVersion=3

#####
# Enable if you want to run ldaps
#####
#socketFactory=org.infoglue.cms.security.DummySSLSocketFactory

#####
# The ldap url, user name and password to bind with (can be ldaps as well).
#####
connectionURL=ldap://ldap.infoglue.org:389
connectionName=InfoGlue LDAP User
connectionPassword=password

#####
# Roles settings section
#
# This part is used by InfoGlue JNDI module to get information about roles which are to
# be used in InfoGlue. The most striking place is to look under roles in management tool.
#####

#####
# Role base - this states which part of the directory structure we want to search for roles
#####
roleBase=dc=infoglue,dc=org

#####
# Role search attribute - which conditions should we use to find a specific user.
# Can be any ordinary query.
# Default is: roleSearch=(cn=*)
# More advanced example: #roleSearch=(&(objectclass=ldapsubentry)(cn=InfoGlue*))
#####
#roleSearch=(cn=*)

#####
# Roles search - find all roles InfoGlue should be aware of when assigning rights.
#####
rolesFilter=(cn=*)

#####
# How long (in seconds) should the roles be cached to avoid to much queries against the LDAP
#####
roleCacheTimeout=10

#####
# Should the search be recursive in the tree or on the base dn only.
# ONELEVEL_SCOPE, SUBTREE_SCOPE and OBJECT_SCOPE possible
#####
roleSearchScope=SUBTREE_SCOPE

#####
# Filter roles on what search criteria.
#####
roleFilter=*
```

InfoGlue – Administrative Manual

```
#####
# What attributes should be returned for each role - commaseperated list
#####
rolesAttributesFilter=cn,uniquemember

#####
# Set which attribute should be used to fetch the roleName
#####
roleNameAttribute=cn

#####
# The attribute containing references to the users who has this role
#####
usersAttributesFilter=uniquemember

#####
# Do you want to remove the role base dn from the role name? Easier on the eye but less
# correct.
#####
removeRoleBasedDN=false

#####
# Groups settings section
#
# This part is used by InfoGlue JNDI module to get information about groups which are to
# be used in InfoGlue. The most striking place is to look under groups in management tool.
#####

#####
# Group base - this states which part of the directory structure we want to search for groups
#####
groupBase=ou=groups,dc=infoglue,dc=com

#####
# Group search attribute - which conditions should we use to find a specific group.
# Can be any ordinary query.
# Default is: groupSearch=(cn=*)
# More advanced example: groupSearch=(&(objectclass=ldapsubentry)(cn=InfoGlue*))
#####
groupSearch=(cn=*)

#####
# Filters what groups are read
#####
groupsFilter=(cn=*)

#####
# How long (in seconds) should the groups be cached to avoid to much queries against the LDAP
#####
groupCacheTimeout=10

#####
# Should the search be recursive in the tree or on the base dn only.
# ONELEVEL_SCOPE, SUBTREE_SCOPE and OBJECT_SCOPE possible
#####
groupSearchScope=SUBTREE_SCOPE

#####
# Filter roles on what search criteria.
#####
groupFilter=*

#####
# What attributes should be returned for each group - commaseperated list
#####
groupsAttributesFilter=cn,uniquemember

#####
# Set which attribute in LDAP should be used for the groupName
#####
groupNameAttribute=cn

#####
# The attribute containing references to the users who has this groups
```

InfoGlue – Administrative Manual

```
#####
usersAttributesFilter=uniquemember

#####
# Do you want to remove the group base dn from the group name? Easier on the eye but less
# correct.
#####
removeGroupBaseDN=false

#####
# Users settings section
#
# This part is used by InfoGlue JNDI module to get information about users which are to
# be used in InfoGlue. The most striking place is to look under users in management tool.
#####

#####
# User base - this states which part of the directory structure we want to search for users
#####
userBase=ou=People,dc=infoglue,dc=com

#####
# Anonymous user base - this states which part of the directory structure we want to search
# for the anonymous user which are used when no user is logged on (on a ordinary website).
#####
#anonymousUserBase=cn=IGUsers,cn=specialusers,dc=infoglue,dc=com

#####
# User search attribute - which conditions should we use to find a specific user.
# Can be any ordinary query. The user name will be inserted where 1 is.
# Default is: userSearch=(uid={1})
#####
userSearch=(uid={1})

#####
# How long (in seconds) should the groups be cached to avoid to much queries against the LDAP
#####
userCacheTimeout=10

#####
# Should the search be recursive in the tree or on the base dn only.
# ONELEVEL_SCOPE, SUBTREE_SCOPE and OBJECT_SCOPE possible
#####
userSearchScope=SUBTREE_SCOPE

#####
# Group search attribute - which conditions should we use to find a specific group.
# Can be any ordinary query.
# Default is: userListSearch=(cn=*)
#####
userListSearch=(cn=*)

#####
# What attributes should be returned for each user - commaseperated list
#####
userAttributesFilter=cn,givenName,sn,mail

#####
# Set which attribute in LDAP should be used for the userName
#####
userNameAttributeFilter=cn

#####
# Set which attribute in LDAP should be used for the users first name
#####
userFirstNameAttributeFilter=givenName

#####
# Set which attribute in LDAP should be used for the users last name
#####
userLastNameAttributeFilter=sn

#####
```

InfoGlue – Administrative Manual

```
# Set which attribute in LDAP should be used for the users mail
#####
userMailAttributeFilter=mail

#####
# The attribute containing references to the roles this user is part of
#####
memberOfAttributeFilter=memberOf

#####
# The ??????????????????
#####
usersAttributeFilter=cn

#####
# Do you want to remove the user base dn from the user name? Easier on the eye but less
# correct.
#####
removeUserBasedDN=false

#####
# This part is for if you use LDAP as authentication as well. You state here what bind user
# name should be used. infoglue.user is replaced with the user name the user states at login.
#####
userNamePattern=CN=infoglue.user,ou=People,dc=infoglue,dc=org

#####
# This part is for if you use LDAP as authentication as well. You state here what bind the
# anonymous user name should be used. infoglue.user is replaced with the user name the user
# states at login.
#####
anonymousUserNamePattern=CN=infoglue.user,ou=SpecialPeople,dc=infoglue,dc=org
```

Sample config for SimplifiedJNDIBasicAuthorizationModule

```
#####
#Properties for the JNDI SETUP
#####
#####
# Authentication method - can be "none" or "simple".
#
# "simple" binds a user
# "none" works only if anonymous bind is allowed.
#####
authenticationMethod=simple

#####
# LDAP-version (2 or 3)
#####
ldapVersion=3

#####
# Enable if you want to run ldaps
#####
#socketFactory=org.infoglue.cms.security.DummySSLSocketFactory

#####
# The ldap url, user name and password to bind with (can be ldaps as well).
#####
connectionURL=ldap://ldap.infoglue.org:389
connectionName=InfoGlue LDAP User
connectionPassword=password

#####
# Roles settings
# roleBase - can be several ";"-separated paths
#####
roleBase=OU=Groups,DC=infoglue,DC=org;OU=Groups,OU=External,DC=infoglue,DC=org
roleSearch=(CN=*)
rolesFilter=(CN=IG_*)
```

InfoGlue – Administrative Manual

```
roleCacheTimeout=1800
#Should the search be recursive in the tree or on the base dn only. ONELEVEL_SCOPE,
SUBTREE_SCOPE and OBJECT_SCOPE possible
roleSearchScope=SUBTREE_SCOPE
roleFilter=*
rolesAttributesFilter=cn,displayName,distinguishedName,description,member
roleNameAttribute=distinguishedName
roleDisplayNameAttribute=cn
roleDescriptionAttribute=description
usersAttributesFilter=member

#####
# Groups settings
# groupBase - can be several ";"-separated paths
#####
groupBase=OU=Department,OU=Groups,DC=infoglue,DC=org
groupSearch=(CN=*)
#groupsFilter=(!(cn=InfoGlue*))
groupsFilter=(!(CN=IG*))
groupCacheTimeout=1800
#Should the search be recursive in the tree or on the base dn only. ONELEVEL_SCOPE,
SUBTREE_SCOPE and OBJECT_SCOPE possible
groupSearchScope=SUBTREE_SCOPE
groupFilter=*
groupsAttributesFilter=cn,displayName,distinguishedName,description,member
groupNameAttribute=distinguishedName
groupDisplayNameAttribute=cn
groupDescriptionAttribute=description
usersAttributesFilter=member

#####
# Users settings
# userBase - can be several ";"-separated paths if needed
# if the username coming in has a domain on it which we want to remove before searching use
#####
userBase=OU=Users,OU=infoglue,DC=org
#userSearch=(CN={1})
userSearch=(|(CN={1})(mail={1})(proxyAddresses={1})(sAMAccountName={1}))
#samAccountDomainName.0=mycorp\
userListSearch=(CN=*)
userCacheTimeout=1800
#Should the search be recursive in the tree or on the base dn only. ONELEVEL_SCOPE,
SUBTREE_SCOPE and OBJECT_SCOPE possible
userSearchScope=SUBTREE_SCOPE
userAttributesFilter=distinguishedName,CN,displayName,givenName,sn,mail,memberOf
userNameAttributeFilter=distinguishedName
usersAttributeFilter=CN
userFirstNameAttributeFilter=givenName
userLastNameAttributeFilter=sn
userMailAttributeFilter=mail
memberOfAttributeFilter=memberOf
#memberOfAttribute=memberOf
userFirstNameDummy=Undefined
userLastNameDummy=Undefined
userMailDummy=noreply@infoglue.org

userNamePattern=CN=infoglue.user,OU=Users,DC=infoglue,DC=org
```

Encoding setting

Encoding of characters are one of the most important aspects of a global content management system. InfoGlue supports Unicode and suggests that UTF-8 is used as the default charset.

It's important to understand how InfoGlue works when receiving data from forms etc both in deliver mode and in the tools. It is also important to understand the issues caused by Apache Tomcat and some other application servers and browsers. The fact is that many appservers and browsers do not behave correctly when sending/receiving data. The browsers sometimes does not send which encoding the data is coming in and Apache Tomcat for example therefore does not decode it correctly. This is a known issue and the solution is to use a servlet filter which sets the encoding in the request object before InfoGlue gets it.

We use this technique both in the cms and deliver applications and the filter (SetCharacterEncoding) is defined in the web.xml. If you're having problems with the wrong encoding coming in you may want to try to set a different encoding in that file than the default UTF-8.

Outgoing encoding is totally managed in the management tool under languages. There you set what charset to use when presenting data to the users browser. We suggest using utf-8 as much as possible.

NiceURI effects:

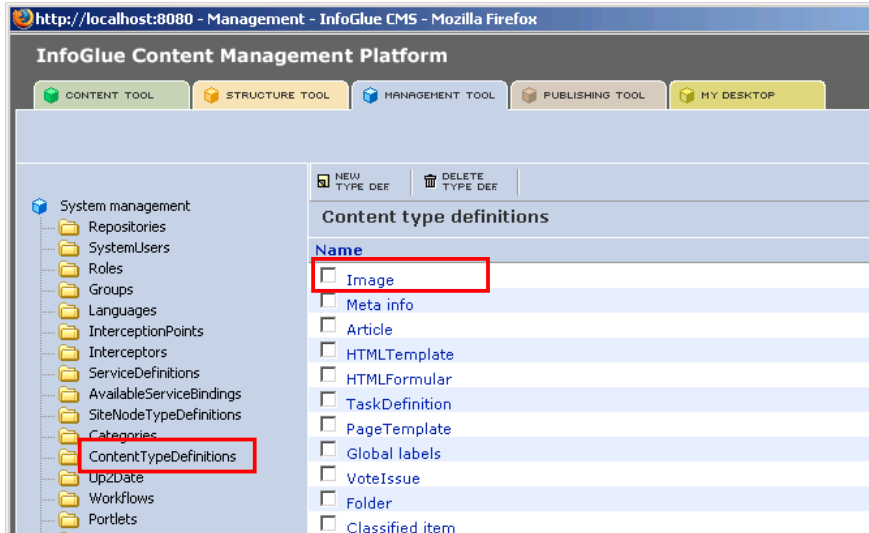
If you have turned on NiceURI – you may want to change the filter SetCharacterEncoding to use a different match in web.xml as there are no longer any action in the url. Set it to “/*” instead of “.action”. Otherwise get/posts to the deliver engine may not turn up rightly encoded.

```
<filter-mapping>
  <filter-name>Set Character Encoding</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Configuring the Content tree with custom icons

To use the feature with custom icons in the tree which was introduced in 2.5.0 you just have to do two things:

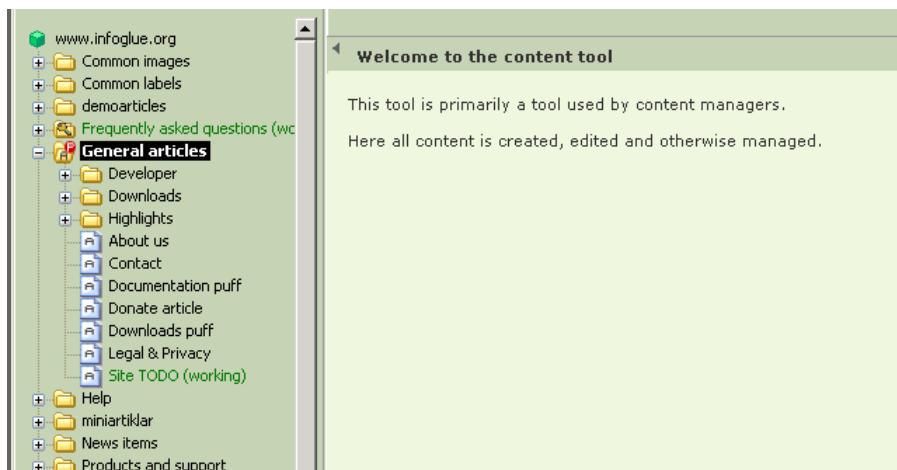
1. Set "Disable custom icons" to "No" in application settings.
2. Check what id the content types you want to add a custom icon for have. Enter the management tool and drag the mouse over the list of content types. In the example below we have opened the content type listing and drag the mouse over the "Image" content type. The url contains the contentTypeDefinitionId which in our case is 1.



Now we want all contents of type Article to have custom icons. We have stated urls in Application settings → custom icons (look at the cms gui-section) to icons for different states of the content type with id 3

```
3=/infoglueCMS/images/icons/articleIcon.png  
3_folderIcon=/infoglueCMS/images/icons/articleFolderIcon.png  
3_openFolderIcon=/infoglueCMS/images/icons/articleOpenFolderIcon.png  
3_protectedFolderIcon=/infoglueCMS/images/icons/articleProtectedFolderIcon.png  
3_protectedOpenFolderIcon=/infoglueCMS/images/icons/articleProtectedOpenFolderIcon.png  
3_protectedIcon=/infoglueCMS/images/icons/articleProtectedIcon.png
```

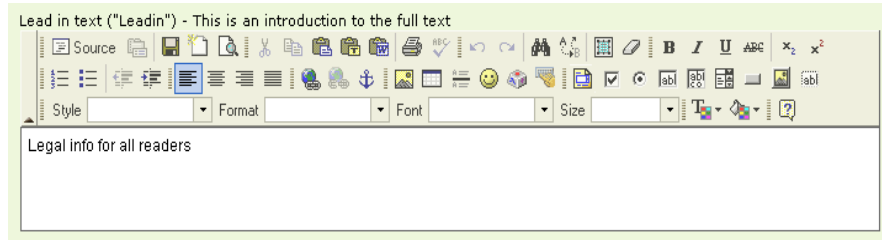
Below is an example where we styled some content types icons:



Configuring the WYSIWYG editor

This information also resides in the User Manual but since it's a pretty central thing in a CMS we thought we add it here as well.

As stated before InfoGlue comes with FCKEditor as the default WYSIWYG-editor. It is a very nice and customizable editor. Default in InfoGlue it looks like this with all buttons enabled:



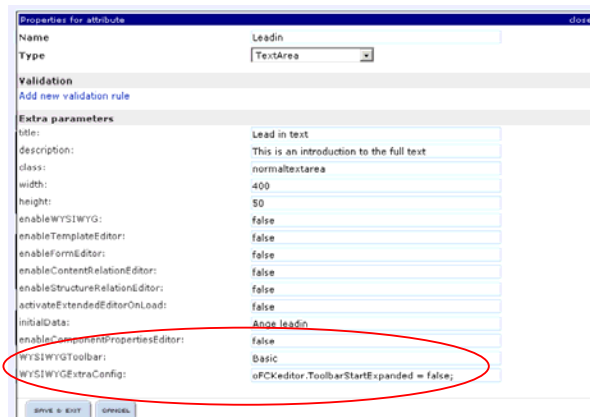
The common thing is of course to configure the palette and some behavior to better fit your organization and your users. This is mostly done in a JavaScript-based configuration file. However – in InfoGlue we decided that it is important to be able to set the configuration not only for the system but also give organizations the possibility to give different repositories, users, groups or roles different setups so more advanced users are allowed more than new users etc.

So here is the possible places to put the configurations:

1. Under each repository's properties. As shown in the repository section of the management tool there was two fields available for the WYSIWYG there.
2. For SystemUser, Roles and Group you can create two property-attributes called WYSIWYGConfig and StylesXML and add the configuration there.

If the properties are available on the system user that information is used, otherwise the editor looks in the roles / groups the user is part of and uses the information there if it exists. If not it uses properties put on the repository in question and if not even defined there it uses the default setup.

In addition to the configurations it is possible to choose toolbar for each attribute in the content types. You can also state other settings for each attribute. For example if we want to state that the "Leadin"-attribute in the content type "Article" should use the toolbar defined as "Basic" in the configuration (javascript shown below) you just state that in the content type.



Example configuration

WYSIWYGConfig

```
FCKConfig.CustomConfigurationsPath = '' ;

//Here you can define a CSS you want the editor to use, usually the same URL as the site.
//FCKConfig.EditorAreaCSS = '$request.contextPath/ViewPage.action?siteNodeId=100130' ;
FCKConfig.EditorAreaCSS = '/info glueDeliverWorking/ViewPage.action?siteNodeId=100130';

FCKConfig.BaseHref = '' ;
FCKConfig.FullPage = false ;
FCKConfig.Debug = false ;
FCKConfig.SkinPath = FCKConfig.BasePath + 'skins/default/' ;
FCKConfig.PluginsPath = FCKConfig.BasePath + 'plugins/' ;
FCKConfig.DefaultLanguage = 'en' ;
FCKConfig.ContentLangDirection = 'ltr' ;
FCKConfig.EnableXHTML = true ;
FCKConfig.EnableSourceXHTML = true ;
FCKConfig.FillEmptyBlocks = true ;
FCKConfig.FormatSource = true ;
FCKConfig.FormatOutput = true ;
FCKConfig.FormatIndentator = ' ' ;
FCKConfig.GeckoUsesSPAN = true ;
FCKConfig.StartupFocus = false ;
FCKConfig.ForcePasteAsPlainText = true ;
FCKConfig.ForceSimpleAmpersand = false ;
FCKConfig.TabSpaces = 0 ;
FCKConfig.ShowBorders = true ;
FCKConfig.UseBROnCarriageReturn = false ;
FCKConfig.ToolbarStartExpanded = true ;
FCKConfig.ToolbarCanCollapse = true ;

//This is a InfoGlue addition which let's you turn off styling in the dropboxes styles/formats
FCKConfig.StyleFontFormats = 'false';
FCKConfig.StyleStyles = 'false';
FCKConfig.AllowImageSizes = 'false';

//This part in for InfoGlue 2.1 - not in 2.0 - let's you define available classes on img-tag.
FCKConfig.ImageClasses =
'img_left_portrait,img_left_letter,img_right_letter,img_right_portrait,img_scaled';
FCKConfig.ImageClassesNames = 'St&aring;ende v&auml;nster,Liggande v&auml;nster,Liggande
h&ouml;ger,St&aring;ende h&ouml;ger,Fullbredd';

FCKConfig.LinkClasses = 'pdf,word,excel,ppt,image,link ';
FCKConfig.LinkClassesNames = 'PDF,Word,Excel,PowerPoint,Image,Link';

FCKConfig.MaxUndoLevels = 15 ;

//This is where you define the toolbar. Look at the FCKEditor-site for the full list
FCKConfig.ToolbarSets["Default"] = [
    ['Cut','Copy','Paste','PasteText','Print','Undo','Redo','Find','Replace','Remove
Format'],
    ['Bold','Italic'],
    ['OrderedList','UnorderedList'],
    ['Link','Unlink','Anchor'],
    ['Image','Table','Rule','SpecialChar','Source'],
    ['Style','FontFormat']
] ;

FCKConfig.ToolbarSets["Basic"] = [
    ['Bold','Italic'],
    ['OrderedList','UnorderedList'],
    ['Link','Unlink','Anchor'],
    ['Image','Source']]
] ;

FCKConfig.ContextMenu =
['Generic','Link','Anchor','Image','Select','Textarea','Checkbox','Radio','TextField','HiddenF
ield','ImageButton','Button','BulletedList','NumberedList','TableCell','Table','Form'] ;

FCKConfig.FontFormats = 'h2;h4;h3;h5;p' ;
```

InfoGlue – Administrative Manual

```
FCKConfig.StylesXmlPath =
'$request.contextPath/WYSIWYGProperties!viewStylesXML.action?repositoryId=$!request.getParameter("repositoryId")&contentId=$!request.getParameter("contentId")&languageId=$!request.getParameter("languageId")' ;

FCKConfig.SpellChecker = 'ieSpell' ; // 'ieSpell' |
'SpellerPages'
FCKConfig.IeSpellDownloadUrl = 'http://www.iespell.com/rel/ieSpellSetup211325.exe'
;
FCKConfig.LinkBrowser = true ;
FCKConfig.LinkBrowserURL =
'$request.contextPath/ViewLinkDialog!viewLinkDialogForFCKEditor.action?repositoryId=$!request.getParameter("repositoryId")&contentId=$!request.getParameter("contentId")&languageId=$!request.getParameter("languageId")' ;
FCKConfig.LinkBrowserWindowWidth = "770" ;
FCKConfig.LinkBrowserWindowHeight = "660" ;
FCKConfig.ImageBrowser = true ;
FCKConfig.ImageBrowserURL =
'$request.contextPath/ViewContentVersion!viewAssetsDialogForFCKEditor.action?repositoryId=$!request.getParameter("repositoryId")&contentId=$!request.getParameter("contentId")&languageId=$!request.getParameter("languageId")' ;
FCKConfig.ImageBrowserWindowWidth = "750" ;
FCKConfig.ImageBrowserWindowHeight = "600" ;
```

StylesXML

```
<?xml version="1.0" encoding="utf-8" ?>
<Styles>
  <Style name="Blockcitat" element="Blockquote"/>
  <Style name="Bild stående vänster" element="img">
    <Attribute name="class" value="img_left_portrait" />
  </Style>
  <Style name="Bild liggande vänster" element="img">
    <Attribute name="class" value="img_left_letter" />
  </Style>
  <Style name="Bild liggande höger" element="img">
    <Attribute name="class" value="img_right_letter" />
  </Style>
  <Style name="Bild stående höger" element="img">
    <Attribute name="class" value="img_right_portrait" />
  </Style>
  <Style name="Fullbredd" element="img">
    <Attribute name="class" value="img_scaled" />
  </Style>
</Styles>
```

As stated these are just examples of configurations we have used. Read more on FCKEditor homepage about configuration options.

Performance tips

This is a hard area to give advice on but in general there are a couple of ground rules.

- Design your website so that dynamic pages are kept to a minimum. They are of course needed and we don't advise against them but don't get used to always turning page cache off as it severely reduces scalability. If you can get by with creating a separate page with your dynamic logic instead of putting everything in one super dynamic component which sends around a lot of parameters that is often justifiable for a high load site.
- Use JSP instead of velocity or freemarker. Faster and consumes less memory.
- Never use recursive macros in Velocity. Under load macros in velocity does not seem to be thread safe. Could be better since 2.4.6 which includes Velocity 1.5.
- Make sure what component takes the most time and analyse what in it is expensive. Perhaps it's a badly written component or perhaps we could optimize InfoGlue. Let us know.
- Make sure to use the new component cache available from InfoGlue 2.0 and forward. It lets you cache parts of a page and leave other parts dynamic.
- Check the database so the indexes in InfoGlue are ok. If one index is broken it can affect performance big.
- Set up a load balanced set of live servers or cms-servers. Also make sure you separate the live database and the cms-database.
- Use the `ViewApplicationState.action` view to analyze which components are slow. Focus on those when optimizing. More info in the diagnostics chapter on that topic.

Errors and how to find them

InfoGlue has some tricks for looking for errors. Error tracking takes some experience but here is a few basic tips:

- First – if you find the system to behave strange – run the validation service. It's located under Management tool → System tools → Validation tool. Run that and you should not get any errors.
- If you wish to see how the delivery engine is doing check the Diagnostics and status page under Management tool. There you have links to all `ViewApplicationState.action` which is extremely useful and even allows you to turn on debugging during runtime.
- The first thing to know is the root account. In the `cms.properties` file there is a entry described above which let you log in without authenticating against the database or normal repository. This will let you do anything in the tools and if you are looking for errors in user / role handling that is a huge help.
- If you log in and the tools work you can also use the Validation service in the management tool menu. It verifies that database access is ok, that digital assets works and that the file system is configured so InfoGlue can write where it needs to.
- If you still get errors or even get errors before this the log-files are the place to look. InfoGlue uses log4j and in `log4j.xml` you can set the debug levels in appropriate ways to get whatever information you want logged. Normally everything is set to WARN so only important things end up in the logs.

Normally all errors are found in the tomcat-logs either for Catalina itself or for the particular webapp. That error message found there together with it's stack trace is often the only thing one needs to find the error if one knows how InfoGlue works.

- If you get problems with memory exception in the tomcat logs look at the `ViewApplicationState.action` and the linked subpage in it. It will report which heap is full. You can also get a list of all Threads being executed at the moment. Often shows what code parts are locked or takes long.
- Use the new Inconsistencies-tool under management → System tools → Resolve inconsistencies. Here you can find if there are inconsistencies in the system that may cause errors or warnings in the logs.

Diagnostics views and features

If you happen to have problems with the site being able to diagnose this is very important. You got several tools reachable from the management tool.

Management - InfoGlue CMS - Mozilla Firefox

http://localhost:8080/infogluCMS/ViewCMSTool.action

InfoGlue Content Management Platform

CONTENT TOOL STRUCTURE TOOL MANAGEMENT TOOL PUBLISHING TOOL MY DESKTOP

MY SETTINGS HELP LOG OUT NEW WINDOW

System management

Basic

- Repositories
- Languages
- Categories
- ContentTypeDefinitions
- Workflows
- Portlets
- Redirects

Authorization

- SystemUsers
- Roles
- Groups

Settings and tools

- Application settings
- Diagnostics and status
- System tools
- Message center

Extra

- InterceptionPoints
- Interceptors
- ServiceDefinitions
- AvailableServiceBindings
- SiteNodeTypeDefinitions
- TransactionHistory
- Up2Date

Diagnostic-central

Application states:

Context/URL	Actions
CMS Context/URL infogluCMS	Clear caches Logs
Working/Preview Context/URL http://localhost:8080/infogluDeliverWorking	Clear caches Logs
Live Context/URL	

Statistics

Number of siteNodes	951
Number of siteNodeVersions	1168
Number of contents	5839
Number of content versions	5523
Number of digitalAssets	3089

Number of unused digitalAssets (could potentially be removed): 1005
SQL to remove unused assets:

MySQL and SQLServer:

```
delete from cmDigitalAsset da where da.assetKey NOT LIKE '%portiententyregistry.xml%' AND da.assetContentType NOT LIKE '%application%' AND da.digitalAssetId not in (select digitalAssetId from cmContentVersionDigitalAsset) AND da.digitalAssetId not in (select digitalAssetId from cmGroupPropertiesDigitalAsset) AND da.digitalAssetId not in (select digitalAssetId from cmRolePropertiesDigitalAsset) AND da.digitalAssetId not in (select digitalAssetId from cmUserPropertiesDigitalAsset)
```

Oracle:

```
delete from cmDigAsset da where da.assetKey NOT LIKE '%portiententyregistry.xml%' AND da.assetContentType NOT LIKE '%application%' AND da.digAssetId not in (select digAssetId from cmContVerDigAsset) AND da.digAssetId not in (select digAssetId from cmGroupPropDigAsset) AND da.digAssetId not in (select digAssetId from cmRolePropDigAsset) AND da.digAssetId not in (select digAssetId from cmUserPropDigAsset)
```

InfoGlue CMS (SE) Version build: 2.9.6.3 Final (2008-12-28)

DU ÄR INLOGGAD SOM administrator

Klar

Statistics (2)

Here you can get some basic information of what is stored in your system and also if you can clean something.

ViewApplicationState (1)

The ViewApplicationState.action can be reached through the management tool and each application has one view.

InfoGlue – Administrative Manual

Each application then has a status view and the sections are described below (opened in new window here):

InfoGlue Statu	
Version	Release date
2.8.0 Final	2008-01-27
Item	Value
Application started	2008-01-27 20:33
Maximum memory (MB)	1016
Used memory (MB)	292
Free memory (MB)	215
Total memory (MB)	508
Number of sessions	185
(remains for 60 minutes after last request)	
Number of request being handled now	0
Number of active request being handled now	0
Total number of requests handled	57093
Average processing time per request (ms)	28
Slowest request (ms)	59596

The first section describes some release information, memory information and rough statistics on how the site is performing and how the load is at the moment. This is all live information so reloading this view is often quite useful. Pay close attention to the memory state when doing load tests for example or when you are developing new heavy components. The average processing time is also of great interest during load testing or during production. A low value here but slow site performance for users would for example indicate that there is an issue with your infrastructure or bandwidth.

Latest publications	
Date:	2008-02-08 02:00:01
Date:	2008-02-08 12:00:01
Date:	2008-02-08 22:00:01
Date:	2008-02-09 02:00:01
Date:	2008-02-09 12:00:01
Individual components (in milliseconds)	
IG Component list component(527) - 42 hits	894
IG References iterator(190) - 19 hits	856
IG References component(916) - 59 hits	741
IG Add Company component(972) - 26 hits	590
RSS Feed JSP(374) - 74 hits	427
IG Sitemap component(539) - 41 hits	408
IG Services component(917) - 97 hits	363
IG Thumbnail viewer component(1006) - 43 hits	297
IG Component detail component(529) - 544 hits	258
IG Omnifind integration component(896) - 200 hits	210
IG Add Content(387) - 33 hits	201
IG News Iterator(173) - 38 hits	193
Company iterator(264) - 54 hits	159
IG Latest news module component(531) - 336 hits	130
IG News Archive component(906) - 121 hits	79
Search result component(319) - 24 hits	64
IG Wide Main CSS(1019) - 150 hits	53
IG HTML Base component(524) - 7004 hits	45
Sellable Items Iterator component(215) - 91 hits	41
IG Left navigation component(538) - 3539 hits	38
Reset component stats	
Decrease active user count (only in special situations)	

The next section (exemplified above) first shows a list of the 5 latest publications made (if it's a deliver engine). The section after that shows a full list of the components rendered since the application was started. It also shows the average processing time for each component and this is really great if you want to optimize your site. In this example the application has run for a while and it has becomes quite clear that the top 6 components are in fact quite heavy and should be considered first when optimizing. However the average page response time on the previous section was quite low and the number of hits the components have had are mostly limited to under 30-100 per component. This indicates that the pages are well cached and that the component average is perhaps not very interesting as they are hit so seldom. Look instead for components with a lot of hits and slow average time or extremely long response times. Below the list you can reset the stats.

InfoGlue – Administrative Manual

Caches section

The section above is the list of all caches InfoGlue controls:

InfoGlue cache name	Cached items Statistics		Actions
propertySetCache	4	No additional statistics	Clear
localeCache	3	No additional statistics	Clear
serverNodePropertiesCache	41	No additional statistics	Clear
boundContentCache	165	No additional statistics	Clear
serviceDefinitionCache	3	No additional statistics	Clear
redirectCache	1	No additional statistics	Clear
masterRepository	2	No additional statistics	Clear
parentRepository	9	No additional statistics	Clear
componentEditorCache	252	No additional statistics	Clear
pageCache	193	Added 193, Approximate size 1731 KB, Cache Flushed 0 Hit count = 440, stale hit count = 0 and miss count = 194	Clear
pageCacheSiteNodeTypeDefinition	71	No additional statistics	Clear
pageCacheParentSiteNodeCache	74	No additional statistics	Clear
userCache	1	No additional statistics	Clear
pageCacheLatestSiteNodeVersions	74	No additional statistics	Clear
languageCache	6	No additional statistics	Clear
matchingContentsCache	2	Added 2, Approximate size 13 KB, Cache Flushed 0 Hit count = 0, stale hit count = 0 and miss count = 1	Clear
parentSiteNodeCache	87	No additional statistics	Clear
NavigationCache	74	No additional statistics	Clear
digitalAssetCache	131	No additional statistics	Clear
siteNodeLanguageCache	127	No additional statistics	Clear
contentVersionCache	5637	Added 5637, Approximate size 2342 KB, Cache Flushed 0 Hit count = 30312, stale hit count = 0 and miss count = 1853	Clear
rootSiteNodeCache	4	No additional statistics	Clear
componentPropertyCache	2948	Added 2948, Approximate size 502 KB, Cache Flushed 0 Hit count = 10149, stale hit count = 0 and miss count = 8742	Clear
contentAttributeCache	4450	Added 4450, Approximate size 5164 KB, Cache Flushed 0 Hit count = 23322, stale hit count = 0 and miss count = 4449	Clear
sortedChildContentsCache	22	No additional statistics	Clear
availableServiceBindingCache	2	No additional statistics	Clear
childSiteNodesCache	50	No additional statistics	Clear
principalCache	1	Added 1, Approximate size 0 KB, Cache Flushed 0 Hit count = 120, stale hit count = 22 and miss count = 0	Clear
assetUrlCache	199	No additional statistics	Clear
pageCacheExtra	193	Added 193, Approximate size 33 KB, Cache Flushed 0 Hit count = 440, stale hit count = 0 and miss count = 194	Clear
pagePathCache	193	No additional statistics	Clear
Clear All Recache sitenodes			

This view is very useful when optimizing cache sizes to minimize memory consumption etc. As mentioned in the section about controlling different cache sizes one should look at the big caches and consider putting a limit on them. To low limit means less performance so it's a delicate balance. The statistics for the different caches helps here. A high miss count in comparison to hit count means a lot of performance loss and suggest you increase that cache size if you limited it. Here you can also clear all the caches if you have cache-problems and want to debug.

Below the normal caches we have added a view of the 3:rd party taglib OSCache so you can clear those caches as well.

OSCache cache name	Cached items Statistics		Actions
com.opensymphony.oscache.web.ServletCache@453fa1	0	No additional statistics	Clear All

We have also added a part were you can clear castor persistence caches as well as application-attributes individually.

Clear individual castor caches	
ClassName: <input type="text" value="SmallContentImpl"/>	<input type="button" value="Submit"/>
Application attributes	
Show/Hide	

InfoGlue – Administrative Manual

Log settings

ClassName: Log level:

[Logfiles viewer](#)

[Restore workflows](#)

Analysis tools

- [Suspected thread locks \(On Java 5-setups only\)](#)
- [List of all active threads \(On Java 5-setups only\)](#)
- [Detailed memory info \(On Java 5-setups only\)](#)

The section above lets you do a few things. The first input form lets you control log4j-logging during runtime on this application. Very useful for debugging issues and the concept is simple. You just state the full class or package and then set the log level. For example – if you want to get all available debug info from the class “org.infoglue.deliver.applications.actions.ViewPageAction” you just type it in and set log level to debug. When you have gotten the debug you need do the same but set it back to warning and it stops verbose logging again.

The link below the fields leads to the log viewer tool (described later).

The last links let you get to some more technical views. The first ("Suspected thread locks") shows the stacktraces for threads that have taken too long to be good and the second (List of all active threads) shows all currently active threads. Here is an example of the suspected threads:

```

Long Threads
All threads which has been running for more than 30 seconds

Elapsed time 10391
Thread id 83
Original url http://infofuse.data.slu.se/infofuseDeliverWorking/ViewPage/renderDecorateIdPage.action?siteNodeId=102&languageId=1&contentType=1&cmsUserName=eussich)
Killed thread
null stack framejava.lang.Thread.sleep(Native Method)
null stack frameorg.apache.jsp.Template_005f1243355042_jsp._jspService(Template_005f1243355042_jsp.java:102)
null stack frameorg.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:97)
null stack framejavax.servlet.http.HttpServlet.service(HttpServlet.java:802)
null stack frameorg.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:334)
null stack frameorg.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:314)
null stack frameorg.apache.jasper.servlet.JspServlet.service(JspServlet.java:264)
null stack framejavax.servlet.http.HttpServlet.service(HttpServlet.java:802)
null stack frameorg.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:252)
null stack frameorg.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:173)
null stack frameorg.apache.catalina.core.ApplicationDispatcher.include(ApplicationDispatcher.java:672)
null stack frameorg.apache.catalina.core.ApplicationDispatcher.doInclude(ApplicationDispatcher.java:574)
null stack frameorg.apache.catalina.core.ApplicationDispatcher.include(ApplicationDispatcher.java:499)
null stack frameorg.infofuse.deliver.util.VelocityTemplateProcessor.dispatchJSP(VelocityTemplateProcessor.java:193)
null stack frameorg.infofuse.deliver.util.VelocityTemplateProcessor.renderTemplate(VelocityTemplateProcessor.java:89)
null stack frameorg.infofuse.deliver.invokers.DecorateComponentBase.doHTMLPageInvoker(decorateComponentBase.doHTMLPageInvoker.java:404)
null stack frameorg.infofuse.deliver.invokers.DecorateComponentBase.doHTMLPageInvoker(decorateComponentBase.doHTMLPageInvoker.java:156)
null stack frameorg.infofuse.deliver.invokers.PageInvoker.deliverPage(PageInvoker.java:179)
null stack frameorg.infofuse.deliver.application.action.ViewPageAction.doRenderDecorateIdPage(ViewPageAction.java:537)
null stack framejavax.servlet.ServletException.invokeMethod(AccessorImpl invokeMethod(Native Method)

Thread MXBean
All suspicious Threads

TP-Processor2
Thread id = 83 TIMED_WAITING(773.634 - 1.1) - lock ownerId - 1 null
Killed thread
null stack framejava.lang.Thread.sleep(Native Method)
null stack frameorg.apache.jsp.Template_005f1243355042_jsp._jspService(Template_005f1243355042_jsp.java:102)
null stack frameorg.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:97)

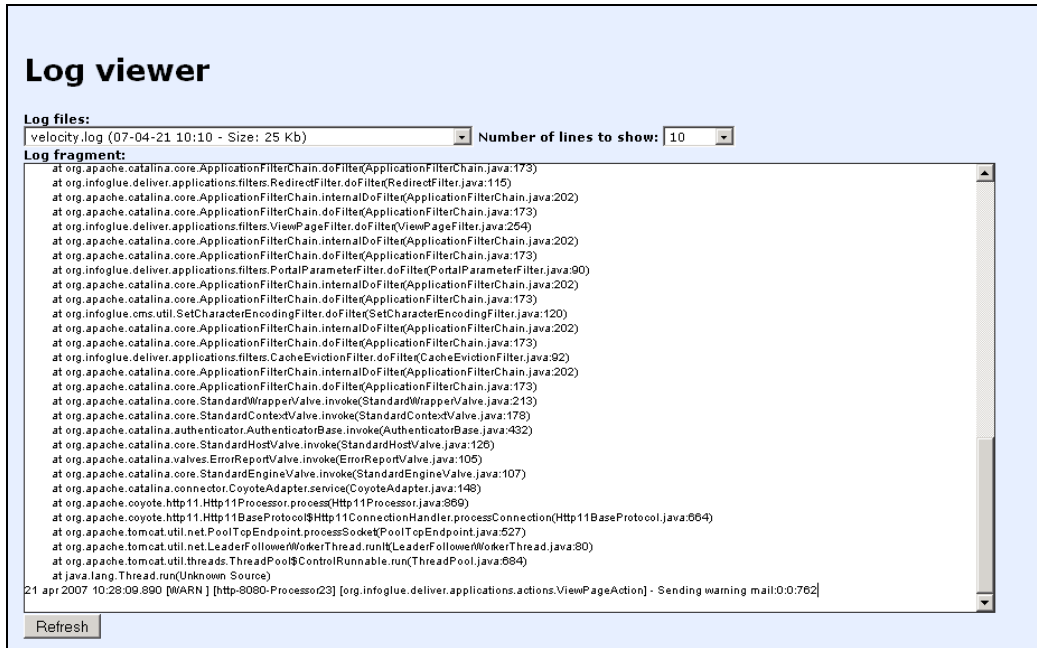
```

These views may seem a waste and to technical but they actually contains very valuable information. However you need to be familiar with java stacktraces to be able to read them. Most interesting here is the original url which caused the long response time and then the first lines in the stacktrace. It shows exactly where in the java-code the thread is now and that in combination with the call history below is all information a developer needs to find what stalls. If you reload the view and the thread does not progress you know what to debug.

InfoGlue – Administrative Manual

Log viewer

This tool is like the tail-program in linux. That is – you can choose how many of the latest added lines you want to view from a certain file. Very useful if you don't have access to the server yourself.



Just select the file you want to show and how many lines and press the refresh-button.

Mail notifications

In InfoGlue 2.4.6 we have added a feature that senses if the system is getting in trouble or if some pages takes to long. This is a very, very important debugging feature and quite often the best way to start debugging any production instabilities. The concept is quite simple. You set up the warning email receiver address in application settings and InfoGlue will then send emails with full thread information when pages takes to long or memory issues arises. Simplifies debugging and error tracking hugely. USE IT!!!

References

1. InfoGlue User Manual and Developer Manual available at www.infoglue.org.
2. MySQL reference website at www.mysql.com
3. Information about FCKEditor – the WYSIWYG-editor used in InfoGlue – can be found at www.fckeditor.net

Appendix A GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly

InfoGlue – Administrative Manual

within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a

InfoGlue – Administrative Manual

section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that

InfoGlue – Administrative Manual

edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers

InfoGlue – Administrative Manual

- or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections

InfoGlue – Administrative Manual

Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

InfoGlue – Administrative Manual

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.