

Deploying Magnolia from start to finish



↕

Your Rating:  Results:  114 rates

Refer to the documentation for formal instructions for installing Magnolia CMS.

Deploying Magnolia From Start to Finish

So what does it take to go from start to finish with Magnolia? Hold on tight, here we go. This document assumes you have no familiarity with Magnolia or J2EE, but that you are a developer and know HTML, CSS (a must), some shell scripting, and are willing to at least learn some Java and JSP.

There are many more things you can do than what is listed on this page, but hopefully this lists the list of things that just about everyone would want to do.

The Ground Work

There is a lot you must do before you can even begin to create content. It isn't hard or take too long, especially if you do the "default" thing. If you have a really custom configuration where Magnolia needs to fit into an existing system, well, it could be more difficult. It may be best to set it up on a test box using the default Tomcat just so you can get started. You can always author content on a test system and move it over to a production system using subscriptions and activation.

Know what you are getting into

System requirements

Java, Tomcat/JBoss/etc, maybe Apache

Windows, or any Unix variant that runs J2EE application servers like Linux or Mac OS X.

Download Magnolia

[Download](#)

Community Edition is free, but does not include support other than wiki and mail lists. Does not include built-in upgrade ability. Magnolia 2.1.x community edition does not include the Documents module. Magnolia 3.x community edition does but it does not include Sitedesigner and other features.

You can download a version of Magnolia that ships with Tomcat, or a version that only includes the war files if you already have a J2EE deployment server.

Install Magnolia

If you downloaded Magnolia with Tomcat, you can put it anywhere on your hard disk that you like.

If you downloaded the war files, put them in your existing server webapps folder.

See the Setup and Installation wiki section of the version of Magnolia you are installing for more information. For example, see [Deploying Magnolia from start to finish^Magnolia 2.1](#) or [Deploying Magnolia from start to finish^Magnolia 3.0](#).

Change names of webapps

In webapps folder, change magnoliaAuthor and magnoliaPublic to the names you want to appear after your domain name (or see next section for root). For example:

```
www.example.com/en (public)
www.example.com/author (author)
```

Change "magnoliaPublic" to "en" and "magnoliaAuthor" to "author". The author url is what site editors will use. The public is what the public will use.

Setup ROOT or Virtual hosts

If you want your public site to be at the root, like this:

```
www.example.com
```

you have to [run Magnolia at the root of the host](#).

If you want something like

```
public.example.com (public)
author.example.com (author)
```

You have to setup [virtual hosts](#) as well.

Set up Magnolia to run at startup

To start Magnolia (assuming you are using the Tomcat server provided with Magnolia), you need to execute the startup.sh (Unix) or startup.bat (Windows) scripts in the tomcat/bin/ folder. To startup Magnolia when the computer starts up, just execute the script at startup. Linux will use some /etc/rc script or something. The best thing for [Mac OS X is a Startup Item](#).

You also want to shut down Magnolia before the computer goes off or you could corrupt your site content (repository).

Startup Magnolia

Restart your computer, execute the startup item, or execute the startup script in tomcat/bin/. It will take a few minutes for Magnolia to startup the first time. The startup progress is saved to the file tomcat/logs/catalina.out. On Unix you can watch it in a terminal with:

```
tail -f /path/to/tomcat/logs/catalina.out
```

You will know it is finished when you see the text:

```
May 30, 2006 2:29:28 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 38654 ms
```

Then you will have to wait for the server to expand the war files. (Magnolia is the author and public war files, not Tomcat and by default a war file is a zipped file that Java uses).

Make sure subscribers work.

If you changed the webapp name, or setup virtual hosts or other, you will need to change the author instance subscriber settings.

Change passwords, add users

Once started, you can open the following location in a web browser (do not use Safari on Mac OS X, use Firefox):

<http://localhost/author> (or whatever the name of your author webapp is)

enter the default username and password, then you will get Admin Central.

In Magnolia 2.1.x you will see "Users" on the left side. Click on it to get the list of users. In Magnolia 3.0.x you will see "Security" on the left side. Click on it to see the "Users" option. Click on that.

You should now see the list of users who can login to the author instance. First thing you should do is make a new user with the same permissions as superuser. Do that by editing the user and set the role to "superuser". Then activate the user! Then logout as "superuser" and login as the new user. Now you can safely change the password of "superuser" and activate it and not see things blow up (users are stored in both the author and public instance and if they have different passwords, things don't work).

Then, change all the default passwords. Right click on each user and in the dialog change the password. You can delete the default users if you would like. I would leave "superuser" though. Just be sure to change the password.

Set up backups

[Repository backup and restore](#)

Content

Now we actually look at site content

Edit your templates

The templates determine the layout of your page. You can use the samples already provided. If you do, you may want to change the names from "samples", but that is actually quite a bit of work. Or create new ones, which is even more work. It might be wise just to use the samples as they are.

Templates are located in the "template" directory in your webapp. You have to manually sync any template changes from author to public. Sometimes when you copy the jsp files from author to public, Tomcat will continue to use the old template (cached version). If you activate the page in question, it will renew the cache for that page.

If you are satisfied with the templates that ship with Magnolia for pages (One Column, Two Column, etc) and paragraphs (Text Image, Table, Mail Form, etc), you really don't need to edit templates at all. If you want to change the way the site looks, that can be done totally by editing the CSS file. However, if you want to have custom paragraphs, like an [image gallery paragraph](#), you will need to edit templates.

See the [templating overview](#) for more information about templates.

Edit your CSS

In your webapp, there is a folder docroot/samples/css. The file main.css controls the look of your site. In Magnolia 2.0.x, there is only one css file. In Magnolia 3.0.x the main.css file loads other files for the menu and such. If you remove your CSS file, you get a text view. This type of design conforms to web standards and is accessible to many devices from computers to cell phones. It is a good idea to customize your site looks by editing the CSS file, and not the templates, unless you have no other choice. For more information on CSS standards, see how powerful and flexible css can be with the same html content, such as [Deploying Magnolia from start to finish^www.csszengarden.com](#).

Default home page

The default home page is features.html. It is very easy to [change the default page](#).

Set up 404 page

Again, very easy to [set up 404 page](#).

Set up /robots.txt

(Magnolia 3.0.x only)

This is assuming your webapp is named ROOT or using virtual hosts to put it at / (otherwise, there is no point in /public/robots.txt as robots wont look for the file there!).

Go to Configuration -> Default URI (in left hand menu). Create a new node "robots". Give it these values:

fromURI = "/robots.txt"

toURI = "forward:/docroot/robots.txt"

Now your robots.txt file in /docroot will be used.

Add content using author instance (easy part)

Using a web browser, add new pages and add paragraphs to the pages. Insert content! This is the easy part!

If you have trouble moving, deleting, or renaming pages, then you have an activation error (since moving, deleting and renaming all activate content to the public instance automatically).

Activate

This is a beautiful thing. Activation is the process of moving the data from the author server to the public server.

In AdminCentral just left click on any page with a Yellow or Red bullet and choose activate. Green means it has been activated. Red means it is not activated. Yellow means that it has been activated but is different.

There are some things you don't want to activate, like subscribers.

However, with all beautiful things, it can go bad very easy. There are several things to watch out for. Be sure you are logged in with a user that exists in the public war and that the passwords are the same.

In Magnolia 3.0.x, you will have to click on the "Inbox" button on the left hand menu. You should see a list of activation requests. If there is nothing there, click on "Workflow" in the "Group by" section in the right hand column. When you see the activation request, click on it and then select "Proceed".

You should now have a live site ready for use!

*In its previous incarnation on JspWiki, this page was last edited on Mar 2, 2007 1:14:53 AM by Magnolia
Other known authors include :*

- *JamesReynolds*