



Deployment Best Practices

James Min

Sr. Consultant / Sales Engineer

July 2011

Issues and Strategies

What do I need to worry about?

Some important questions you need to ask yourself:

- What app server or servlet container am I running on?
- Am I deploying only web content, or plugins as well?
- Do I have a window of down time to work with? How long?
- Am I deploying to a cluster?
- Am I using Liferay staging publish to remote feature?
- Do I have documents in Liferay's Document Library?
- How recent a copy is Staging/UAT to production?
- Is doing a DB export/import an option (simplicity)?
- Do you use portal-ext.properties heavily?

App Servers & Servlet Containers

- Tomcat (servlet container) is the simplest.
 - Deploying plugins is as simple as using the auto deploy directory, or using the UI to deploy
- Java applications servers may require extra steps when deploying Liferay Plugins.
 - This may require using the app server's admin console to deploy the plugins as web apps in a separate fashion.
 - Do not use auto deploy or the UI to deploy plugins

Web Content vs. Plugins

- Web Content is stored in the database
 - You can export/import via LAR
 - You may also use Liferay Staging
 - Brute force DB export/import is also an option
- Plugins are WAR files that get deployed as Java web apps
 - For Tomcat, this will end up being a physical directory in tomcat/webapps
 - For other Java app servers, they also be physical directories but this varies from vendor to vendor.

Do I Get Any Down Time?

- If Yes,
 - GREAT!
- If No (or minimal),
 - Deploy to a staging that is as close to production as possible. After testing, switch over your router to point to the staging server (new prod), or bring down the prod and bring up staging (as your new prod).
 - Swap out nodes in a cluster, one by one.

Am I Clustering?

In a cluster, always remember:

- All portal, page configuration, and web content is stored in the DB
- Each node in the cluster reads portal-ext.properties locally
- Documents in the Doc Lib are stored locally (by default)
- Plugins and the Portal (root) web app is local to each node in the cluster. In other words, make sure you physically deploy each plugin to each node.
- Indexes are local to each node by default (turn on clusterLink)

Staging: Publish to Remote

Publish to Remote is a great feature to use if all you are doing is deploying new web content or page configurations.

- You still need to deploy WARs (plugins) to the prod server
- You still need to deploy portal-ext.properties to each server
- New users are not exported with this feature

Document Library

Document Library stores documents locally in the file system by default.

- The database hold the meta-data about the document
- The physical file needs to be moved to prod as well (data/document_library/)
- In a cluster, the path to the doc lib needs to be centralized (i.e. - SAN, NAS), and storage hook needs to be AdvancedFileSystemHook

Staging / UAT

Keep Staging or UAT as updated as possible.

- Refresh from the prod database often
- Refresh the plugins/web apps

Database Export / Import

Perhaps just doing a DB dump and import is an option

- Works best if only content was changed
- Useful if you need to get new users or other non-content settings into production
- You still need to take care of Document Library and any plugins (web apps)

portal-ext.properties

It is best to store as much in portal-ext.properties as possible.

- Most portal setting that are set via UI are in portal-ext.properties as well
- Settings in the DB take precedence over what is in the properties file
- Allows for a more “scriptable” deployment
- Local to each node in a cluster

Other Considerations & Resources

- JVM Tuning
- Using a CDN, or simply offloading static content to a web server

“Advanced Liferay Architecture-Clustering and High Availability” :

<http://www.liferay.com/documentation/additional-resources/whitepapers>

Questions?