Deployment Template

Versioning

- Choose a versioning scheme for the new releases. The versioning scheme should be documented and all members of the project organization should know and understand the scheme. Some ‘well-known’ examples are http://apr.apache.org/versioning.html and http://semver.org/. A maven plugin exists to enforce a versioning scheme (http://mojo.codehaus.org/versions-maven-plugin/version-rules.html)

General Guidelines

- When you start a new 'release' after a previous release has gone to T, A or P always start with a new 'Version Number' of the project. Avoid deployment of snapshot versions !
- Create new Release Notes/Deployment instructions for each version (e.g.) The deployment instructions might contain information for several 'revisions’ or versions

Environment

Date	What Environment (D/T/A/P) ?

What release ?

Current release version	New release version

Important Deployment Information

Information on upgraders / namespace visitors?	**yes/no**
-Drepo.bootstrap	**yes/no**
Are there manual steps for this release?	**yes/no**
Jcr Runners ?	**yes/no**
Groovy Updaters ?	**yes/no**
Hippo Upgrade ?	**yes/no**

Environment

Create new deployment instructions for D,T,A and P with the date of the deployment.

What release ?

Describe the version of current release and the new release which is deployed on the server

Important Deployment Information

This set of questions gives the System Adminan overview of the most important steps during the deployment.Always fill in all the fields marked with a (*), if there is nothing to fill in, just fill in 'N/A'

Information on upgraders / namespace visitors?

This information is important for Hippo versions before 7.8 or when upgrading a ‘strict’ dcoumenttype, and the namespace prefix of the document needs an update. The namespace visitor used for the procedure does not work in a clustered environment. This means that this upgrade can only occur on a single cluster. The deployment notes should describe this in detail, because this type of upgrade is very riskfull.

More info:

http://blog.jeroenreijn.com/2010/06/introduction-to-hippo-cms-7-updater.html

http://www.onehippo.org/7_7/library/concepts/upgrade/upgrading-and-updating.html

http://www.onehippo.org/7_8/library/about/release-notes/7.8.0-release-notes-updater-improvements.html

https://forge.onehippo.org/svn/hippo-utilities/hippo-utilities/trunk/repository/src/main/java/org/onehippo/forge/utilities/repository/updater/BaseCndUpdater.java

-Drepo.bootstrap

In order to load initialize items during the startup of the CMS, the system property -Drepo.bootstrap = true is needed as JVM argument. Make sure that the system property is true for only 1 cms cluster ! System admin need to know if the property should be on true.

More info:

http://www.onehippo.org/library/development/reload-on-startup-initialize-items.html http://www.onehippo.org/7_7/library/concepts/content-repository/content-bootstrap.html

Are there manual steps for this release?

It is important for the system admin to know if during the deployment manual steps are needed, because these steps are easily overlooked or misinterpreted. The are an extra risk during the deployment.

Groovy Updaters ?

The system admin has to know if the Groovy Updaters are needed for this release, because the extra time needed for the execution of these updaters. Releases with Groovy updater are more riskfull and time-consuming than releases without updaters.

Hippo Upgrade ?

Hippo upgrades are - in general - more complex and take more time and effort. Hence the System Admin should know if the release contains a new version of Hippo.

Steps BEFORE release:

#	Step	Who (Role)	Description	Status	Notes

Estimated Time:

Steps DURING release:

#	Step	Who (Role)	Description	Status	Notes

Estimated Time:

Steps AFTER release:

#	Step	Who (Role)	Description	Status	Notes

Estimated Time:

Fields in the deployment notes:

Field	Description	Example
#	The # is the order or number of the step.	1
Step	The title or short description of the step. For instance ‘start the cms’.	Remove sitemenus from the hst:workspace
Who (Role)	Who is doing the step ?	John Doe (Functional Admin)

	In between the brackets the role of the person doing the step. Do not forget the elease managers, testers, developers and functional admins !	
Description	A detailed description of the step	For all .org channels (i.e. /hst:hst//hst:hst/hst:configurations/7_X): move the 'hst:sitemenus' node into the 'hst:workspace' node. For example: /hst:hst/hst:configurations/7_1/hst:sitemenus should move to /hst:hst/hst:configurations/7_1/hst:workspace/hst:sitemenus
Status	When the person responsible for the step has executed the step, he/she updates status field.	Done
Notes	Additional notes about the step	On test no sitemenus are removed.
Estimated Time	An estimation of the time needed to execute the steps helps to release manager to make a valid planning of	2:30

	the release. This is especially important for acceptance and production releases	

Additional remarks

- Add to the deployment instructions the links to the issues of the 'Revision or microversion numbers'.
- Make sure release notes are always updated for every release of every version.