

Upgrades & Customization Maintenance 2

We have developed an upgrade strategy intended to make it as simple as possible for you to maintain your customizations as you upgrade KFS to future versions. The keys to working within this strategy are...

- Using the provided override hooks to avoid modification of delivered files whenever possible
- Choosing the appropriate location for the files that you add in order to override or extend delivered functionality
- Tagging any delivered files that you absolutely have to modify with the appropriate phrase (#INSTITUTIONAL_CUSTOMIZATION#) within a comment appropriate for that file type

The Upgrade Process

1. Obtain the new distribution (Vanilla KFS)
2. Point it at your prior version (Customized KFS) via a build property
3. Run an upgrade target that will copy the following from Customized KFS to Vanilla KFS
 - Java Source Code
 - All files under work/src but not under work/src/org/kuali
 - All files under work/src/org/kuali that contain #INSTITUTIONAL_CUSTOMIZATION#
 - Test Source Code
 - All files under test/src but not under work/src/org/kuali
 - All files under test/src/org/kuali that contain #INSTITUTIONAL_CUSTOMIZATION#
 - Web Files
 - all files under work/web-root that contain #INSTITUTIONAL_CUSTOMIZATION#
4. Copy any content that you added to work/web-root from Customized KFS to Vanilla KFS



It would be best for you to decide on a clear separation strategy from the start. For example, you could always add your content in a sub directory named after your institution, so that when you go to upgrade you know to copy over work/web-root/WEB-INF/tags/my_institution and work/web-root/WEB-INF/lib/my_institution.

5. Run a target to generate the upgrade script appropriate for your database platform from the upgrade XML delivered with the distribution
6. Review the generated upgrade script, modify as necessary, and execute against your database

Extra Care with Modifications to Delivered Files

It is also recommended that you use version control or other comparison tools to assist with merging of modifications to delivered files. For example, when you get a new version of vanilla kfs...

1. Create a branch off of your current development code base
2. Delete the customized code and put the vanilla distribution in its place
3. Do a merge / synchronize and apply changes in the vanilla distribution to the files that you customized after you've run through the process above

Unable to render {include} The included page could not be found.