

# ToDo

## Things To Do

### (March 2010) Use Nexus

After graduation, set up to use Apache Nexus; will help when moving to CI, and with releases.

### (March 2010) Use CI

After graduation, set up CI on Hudson

### (March 2010) Use Sonar

See <http://nemo.sonarsource.org/>

Jukka is setting up an Apache-local Sonar server on Hudson (<http://issues.apache.org/jira/browse/INFRA-2140>)

Upgrade to use FOP 0.94, see if it improves anything

Write a script to set all the places in SVN that need changing when going to next release target.

Pick a continuous build environment tool set and configure the project to use it.

### Add Missing Test Cases

Some parts of the UIMA Framework need test cases. We list them here

- Error message for trying to create subtype of AnnotationBase in base CAS - also for JCas
- JCasGen
- CDE (GUI based editor)

### UIMA Framework improvements - future

### Upgrade TableTree widgets (deprecated since 3.1) in CDE - hope it fixes some display issues on Macs

### Redo socket support to use Java NIO primitives

Earlier studies showed some performance gains are possible.

### Improve multi-threading support

Includes better design for interruptability, figuring out what can be used from Java 1.5 (we want to remain 1.4 compatible, though).

### Improved Design for parallelism

Something that substitutes other open source infrastructure code (such as ActiveMQ) for custom code, and moves toward adoption of major standards (like Java's JMS, JMX).

### Other possible cleanups that could be done at this time

Because this transition will have some impact on our users, and we want to minimize the number of times we impact users this way, we're considering some other potential changes.

- (To be listed - what these changes might be)
- [Make JCas\\_Type classes inner classes](#), cutting in 1/2 the number of generated source classes.

### Create separate CAS and CasView interfaces

The following are confusing (or some might say, "broken")

(a) the interface "CAS" can be an interface to either the whole CAS or to a view. Methods like this are poor:  
CAS view = cas.getView(name);

(b) the logic determining which "CAS" (a view or the whole CAS) gets passed to an annotator's process method is needlessly complicated.

We want whatever we do to be easier to document and explain to users than what we currently have. However, we're concerned about doing anything that would break compatibility for existing users, especially for single-view annotators.

A previous proposal that did not achieve consensus can be found here: [CasAndCasViewInterfaceRedesign](#).

## **Something that improves performance.**

## **Historical - things accomplished**

Things in this section have already been accomplished.

## **Redo documentation as Docbook; break into more manageable pieces (4 books).**

## **Rewrite of a few Eclipse modules (Now completed)**

<http://issues.apache.org/jira/browse/UIMA-2> We have a few modules in the Eclipse tooling that are written as derivative works of Eclipse source files. Because of this, the source for these cannot be distributed by Apache under the Apache 2.0 license. These modules need to be rewritten using more standard Eclipse APIs instead of being derivative works of Eclipse source.

## **Tooling to support the Apache namespace movement**

<http://issues.apache.org/jira/browse/UIMA-19> It would be nice to have instructions and tooling that supported users making the transition to Apache UIMA. A conversion tool has been built.

## **Miscellaneous cleanups**

- Redo JCas into an Interface + Impl

## **Preparing for release 2.1**

Jira has many open items that need to be completed for release 2.1. Some of the more major ones are:

## **Docbook (Mostly done)**

<http://issues.apache.org/jira/browse/UIMA-3>

Remaining work as of Jan 2007 includes updating sections of the manual to conform to the 2.1 design level.

Future: Consider upgrading docbook templates to latest stable releases.