



June 3, 2022

Chris:

You asked me to think about how to transfer support of CTP to the RSNA. There are three basic directions you can go:

- You can do nothing. CTP works now, although as DICOM continues to evolve and as support requests go unanswered, CTP will fall behind and slowly die. CTP has had a good run, and letting it die – perhaps motivating someone else with a fresh perspective to replace it – might be good for the community.
- You can find somebody just to provide support. There would be a bit of a learning curve, but if you find a modestly skilled Java programmer with a good knowledge of DICOM, that could keep CTP alive for a good while. I'd probably have to work with them for a few days to get them oriented.
- You can find somebody to work with me on a rewrite to bring CTP up to modern standards and then have them take over the new version. Four or five years ago, I made a list of things that I felt needed to be done to CTP. I have worked on the rewrite (CTP2) sporadically, and I have made good progress, but there is still a lot of work to do. Here are a few notes on this third idea.

Update CTP to use a new DICOM library

CTP is based on dcm4che, which hasn't been officially maintained by anyone for more than 10 years. I finally got the source code and put it in a GitHub repo just so I could update it with new elements as the DICOM standard changes. I also had to adapt it to deal with the problems that were caused when Java 9 eliminated the 32-bit version. That change resulted in the loss of the ImageIO Tools on Windows. CTP depended on the ImageIO Tools for some of the important codecs encountered in medical images

I finally decided to replace dcm4che with a real DICOM library maintained by someone else. There aren't a lot of open source choices for such a library, but fortunately, David Clunie's Pixelmed meets the requirements very well. I have completed a little over half the changes to incorporate Pixelmed.

Note: Changing the DICOM library might affect externally written plugins, AnonymizerExtensions, DatabaseAdapters, and PipelineStages, but that is unavoidable.

Note: This will also involve rewriting the DicomStorageSCP and DicomStorageSCU classes used for communication. This hasn't been done yet.

Redesign the DicomAnonymizer script language

The script language has grown organically for 20 years, starting with something that Guenter Zeilinger included in one of the dcm4che example programs. I finally decided that a complete rewrite of the language syntax and its parser was necessary to add capabilities that many people have requested. I have done a lot of work on this part of the code, but most of it has been in test programs, so there is still a fair amount of work to put it all into CTP.

Improve the ClassLoader mechanism

In the transition from Java 8 to Java 9, the Java maven changed the application class loader, breaking the mechanism by which CTP supports user-written extensions (of which there are quite a few out in the wild). Over the years, I have dealt with the problem in three different ways, but while they have all worked, they have all been embarrassing kludges. I finally came up with a clean approach that works for all Javas. The code is in CTP2, but I haven't put it into the original CTP.

Change default for returnStoredFile

The default value for **returnStoredFile** was set to **yes** for the BasicFileStorageService to make it easy for the NCI to configure NBIA. For historical reasons, I kept that default value in all subsequent storage services. That was a very bad choice. It has caused confusion in the field and resulted in numerous support requests. I have changed the default to **no** in all the stages in CTP2. This sounds like a trivial thing, but it is a significant improvement.

FileStorageService improvements

These would be nice additions to CTP, but they are additions, not conversions of what CTP already has:

Implement JDBM index in FileStorageService

The current index in the FileStorage is an XML structure. Changing it to a B-tree structure would allow development of better searching and viewing features.

Support WADO access to FileStorageService studies

This is probably the best way to improve the viewing capability.

Support {keyword} in the DirectoryStorageService

This would be a new capability. It isn't done. It would be a nice addition.

New pipeline stages and plugins

These have been requested, but they aren't in CTP, so they would be future extensions of CTP2:

- DicomCleaner pipeline stage
- DicomStorageQR plugin

Configuration changes:

Force pipeline root

The configuration element for a Pipeline has an optional **root** attribute. If present, the value defines a directory below which the root directories of its pipeline stages are located (when they are defined as relative paths). CTP2 makes the attribute mandatory. This makes it easier to specify pipeline stages in multi-pipeline configurations and avoids errors.

Show all configuration elements

When CTP starts, it logs everything about the configuration of the system on which it is running, plus the XML of the configuration file that specifies the embedded servlet container, the plugins, the pipelines, and the pipeline stages. Early on, I decided not to log the values of attributes that were missing or had their default values. That was a bad decision. CTP2 fixes that

Java issues

The latest Javas can be deployed with an application, but there are some things to consider. I have dealt with some but not others:

- Build on Java 8. Done.
- Convert to open source Java. Not done.
- Package Java in the CTP2-installer. Not done. This may not be possible if we want to remain compatible with Java 8.
- Consider switching to slf4j. My inclination is not to do this.
- Consider building with Maven instead of Ant. I'm old fashioned, so I use Ant and a really good text editor. This works very well, but younger developers will probably think an IDE and Maven would be better.
- Convert to the Java Base64 class instead of the one I wrote into the Util library. It would probably be best to run on code in the Java class library as much as possible.

Code review

This would all be necessary in any transfer:

- Review all pipeline stages
- Review queue management
- Review private element tag syntax
- Improve VR for private elements if possible