Tika2_0RoadMap

Background

This page is intended for a discussion of changes anticipated in Tika 2.0.

The Work-In-Progress Tika 2.x branch is in Git as *2.x*, see https://github.com/apache/tika/tree/2.x

See Tika2_0MigrationGuide for documentation of changes in the 2.x branch.

Note - not everything list here will be done in 2.0. Please contribute!

Major Planned Changes

- Remove Tika app's server. Users should now use the JAX-RS server available in the tika-server module.
- Remove all deprecated methods and metadata keys
- Move from service loading to config file for parser specification and loading. TIKA-1445 raised this as an important area for improvement within Tika. The current strategy in the AutoDetectParser is to load all parsers and then pick the first parser that matches a given mime type. Tika chooses the "first" by first sorting on whether or not the class name begins with org.apache.tika and then (effectively) by reverse alphabetical order of the class name. It would be great if the user could specify the order of parser selection in the config file. We will be working towards this gradually through Tika 1.8 and 1.9, and we will remove service loading entirely in Tika 2.0.
    - *Not sure this is quite right. We want to allow people full control of parser ordering, combining multiple parsers for fuller metadata etc, but we also want to continue to support the use case of "drop an extra jar on the classpath and automatically have the parser in it loaded+used", which relies on the service loading to find parsers and add them*
    - *Who says this use case *has* to be supported using ServiceLoading - seems like we can also support it without ServiceLoading and with more control over the ordering, etc.*
- Allow users to build composite parsers with configurable strategies via the config file (TIKA-1509 and CompositeParserDiscussion). We will be working towards this gradually through Tika 1.8 and 1.9. By Tika 2.0, however, this will be the default.
- Solve the complex metadata challenge; see: TIKA-1607 and TIKA-1691 and ISO 19115 discussion .... Or at least come to some accommodation that will allow for both easy key/values access and more advanced access for those who know what they're doing.
- Work out how to allow "resetting" or "augmenting" or "rewinding" of the SAX stream, to permit:
    - We tried one parser, got half way through and it failed, and now we want to try another
    - We used on parser, that finished, now we want to run a second one (eg OCR)
    - We finished one paragraph, then did NER on it, and want to update the HTML with the entities
    - We want to mark the last 2 paragraphs as *language=german* or unmark *language=english* on the body now we've found some german text
- Parsers vs Content Handlers vs Decorators - Work out where we want "content enhancement" logic to live (Wrapping Parser? Decorator? Handler? Other). Then, ensure that can be configured in easily (config xml as well as code), can do what it needs, then shift things over to the new model if they're not there already

Major Completed / Mostly-Completed Changes

- Allow for easily configurable parser sub-packages. The tika-app, tika-server and tika-bundle jars are now pushing or are > 50MB. It would be great if users easily could specify a subset of parsers they care about, either a la carte or by category (image, common office files (MSOffice, PDF, etc.), environmental data) and only get the dependencies required for that subset of parsers.

Minor Planned Changes

- Move to Java 1.8 (???)
- Have mail "folder" formats (such as mbox) behave more like other containers, triggering embedded documents for each of their mail messages rather than mushing everything together

Wishes

- Uniform representation of geo information from common files (kml/kmz/exif/others???) in metadata (perhaps WKT)?