Migrating to Tika 2.0.0

Major breaking changes

- OCR is now triggered automatically for PDFs if tesseract is on the user's path see (TikaOCR#disable-ocr) for how to disable OCR.
- Removed deprecated Metadata keys/properties and moved some commonly used keys from Metadata to TikaCoreProperties (such as TikaCoreProperties.RESOURCE_NAME_KEY) (TIKA-1974). See below for a list of changed keys.
- We upgraded from `log4j` to `log4j2` in tika-app, tika-server and anywhere else we used to use `log4j`.
- The `tika-parsers` package has been split into several sub packages, inluding: `tika-parsers-standard-package`, `tika-parser-scientific-package` and `tika-parser-sqlite3-package`. You will need the `tika-parsers-standard-package` for complete detection of container-based formats such as .doc, .ppt, .xls, .docx, .pptx, .xlsx and others.
- `tika-app` only includes parsers in `tika-parsers-standard-package`; users have to add `tika-parser-scientific-package` and `tika-parser-sqlite3-package` if desired.
- `tika-server` is now `tika-server-standard` and only includes parsers in `tika-parsers-standard-package`
- `tika-server` is now run in `--spawnChild` mode by default.
- Removed deprecated PDFPreflightParser (TIKA-3437).
- Parsers are now configured via `tika-config.xml` on instantiation. We have moved away from configuration via .properties files because of confusion among users. This affects the PDFParser, TesseractOCRParser and the StringsParser. See below for links to the specific parsers.
- Changed namespaces of translator implementations (e.g. `org.apache.tika.language.translate.impl`) to avoid split-package with tika-core.

For more details on changes in `tika-server` in 2.x, please see: TikaServer in Tika 2.x.

Metadata

Breaking Metadata Key Changes Between 1.x and 2.x

These are changes in locations of keys, not in the key names that consumers/clients will see:

- `Metadata.RESOURCE_NAME_KEY` has been renamed `TikaCoreProperties.RESOURCE_NAME_KEY`.
- `TikaCoreProperties.KEYWORDS` has been removed in favor of `Office.KEYWORDS`

Changed Metadata Keys

There are a few other subtle changes in key names listed below:

Tika 1.x	Tika 2.x
X-Parsed-By	X-TIKA:Parsed-By
X-TIKA:EXCEPTION:runtime	X-TIKA:EXCEPTION:container_exception

Removed duplicate/triplicate keys

Background: In early 1.x, we had basic metadata keys that were created somewhat *ad hoc*. We then added metadata keys based on standards such as Dublin Core, or we at least tried to add namespaces to the metadata keys for specific file formats. To maintain backwards compatibility, we kept the old keys and added new keys. This led to quite a bit of metadata bloat, where we'd have the same information two or three times. In Tika 2.x, we slimmed down the metadata keys and relied only on the standards-based or name-spaced keys. In the table below, we document the mappings. If you notice any missing, please let us know or update the wiki.

Tika 1.x	Tika 2.x
Author, meta:author, dc:creator	dc:creator
Last-Author, meta:last-author	meta:last-author
title, dc:title	dc:title
Creation-Date, date, dcterms:created	dcterms:created
Last-Modified, modified, dcterms:modified	dcterms:modified
Last-Save-Date, meta:save-date	meta:save-date
w:comments	w:Comments
Application-Name, extended-properties:Application	extended-properties:Application
Character Count, meta:character-count	meta:character-count

Company, extended-properties:Company	extended-properties:Company
Edit-Time, extended-properties:TotalTime	extended-properties:TotalTime
Keywords, meta:keyword, dc:subject	meta:keyword, dc:subject
Page-Count, meta:page-count	meta:page-count
Revision-Number, cp:revision	cp:revision
subject, cp:subject, dc:subject	dc:subject
Template, extended-properties:Template	extended-properties:Template
Word-Count, meta:word-count	meta:word-count
identifier	dc:identifier
publisher	dc:publisher
dc:description, subject (as in MSG files)	dc:description (dc:subject was added back in 2.4.0).

tika-parsers – Configuring via tika-config.xml

In 2.x, we've moved all configuration into a `tika-config.xml` file. Two popular parsers used to rely on *.properties files; see their individual pages for details: PDFParser and TesseractOCRParser.

See other individual parser pages for available configurations: TikaParserNotes. If you notice any missing parsers, please help us document configurations for all parsers.

tika-parsers module

In Tika 2.x, we separated the 1.x `tika-parsers` module into three modules and packages:

1. tika-parsers-standard – the most common parsers – should not require rest calls nor native libs (**NOTE**: despite the goal of this package, we do include the TesseractOCR parser which will run Tesseract if you have that installed)
2. tika-parsers-extended – may include native libs and/or dependencies that not everyone wants (e.g. netcdf)
3. tika-parsers-ml – may include heavy dependencies (e.g. dl4j) or parsers that rely on rest calls and external services

The goal is to allow users to select only the parsers (and dependencies) that they want.

When using `tika-parsers` in your project, you need to change the dependencies from:

pom.xml from 1.27

```
<dependency>
  <groupId>org.apache.tika</groupId>
  <artifactId>tika-parsers</artifactId>
  <version>1.27</version>
</dependency>
```

to at least `tika-parsers-standard-package`. If you want `netcdf` parsing and/or `sqlite3` parsing – both of which were included in `tika-parsers` in 1.x – you'll need to include `tika-parser-scientific-package` and/or the `tika-parser-sqlite3-package`.

pom.xml for 2.0.0+

```
<dependency>
  <groupId>org.apache.tika</groupId>
  <artifactId>tika-parsers-standard-package</artifactId>
  <version>2.7.0</version>
</dependency>
<dependency>
  <groupId>org.apache.tika</groupId>
  <artifactId>tika-parser-scientific-package</artifactId>
  <version>2.7.0</version>
</dependency>
<dependency>
  <groupId>org.apache.tika</groupId>
  <artifactId>tika-parser-sqlite3-package</artifactId>
  <version>2.7.0</version>
</dependency>
```

NOTE: As of Tika 2.7.0, we have added `tika-parser-nlp-package` to our release artifacts.

NOTE: As in Tika 1.x, if you need detection on container formats (e.g. OLE2: .doc, .ppt, .xls or zip-based: .xlsx, .pptx, .docx or .ogg based), you need to include the underlying Tika parsers that will parse the container files and make the detection based on the information in the container. In Tika 2.x, this means that you need to include `tika-parsers-standard-package`!

Lesser parser notes that may only affect early versions of 2.x

Also, there's a small transitive dependency issue with jcl-over-slf4j between tika-parsers-standard-package 2.0.0 and tika-parser-scientific-module:2.0.0. So if you are using maven enforcer plugin, you will need to fix it by adding this:

pom.xml

```
<!-- Fix tika-parsers-standard-package 2.0.0 vs tika-parser-scientific-module:2.0.0 transitive dependency -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>jcl-over-slf4j</artifactId>
    <version>1.7.31</version>
</dependency>
```

If you are checking for CVEs (recommended), the tika-parser-scientific-module:2.0.0 comes with a transitive dependency on quartz 2.2.0 which should be fixed like this:

quartz

```
  <dependency>
    <groupId>edu.ucar</groupId>
    <artifactId>netcdf4</artifactId>
    <version>${netcdf-java.version}</version>
    <exclusions>
      ...
      <exclusion>
        <groupId>org.quartz-scheduler</groupId>
        <artifactId>quartz</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>org.quartz-scheduler</groupId>
    <artifactId>quartz</artifactId>
    <version>2.3.2</version>
  </dependency>
```

Language Detection

When using lang detection, you need to use:

pom.xml 2.0.0

```
<dependency>
  <groupId>org.apache.tika</groupId>
  <artifactId>tika-langdetect-optimaize</artifactId>
  <version>2.1.0</version>
</dependency>
```

Also note that `org.apache.tika.langdetect.OptimaizeLangDetector.getDefaultLanguageDetector` has moved to `org.apache.tika.langdetect.optimaize.OptimaizeLangDetector.getDefaultLanguageDetector`.

For OCR, you can not use anymore `TesseractOCRConfig.setTesseractPath(String)` and `TesseractOCRConfig.setTessdataPath(String)` methods. They moved to the `TesseractOCRParser` class.

tika-parsers-module optional dependencies

zstd

The zstd dependency includes native libs and is not packaged with the tika-parsers-module. If you'd like to parse zstd files, include:

zstd-jni

```
    <dependency>
      <groupId>com.github.luben</groupId>
      <artifactId>zstd-jni</artifactId>
      <version>1.5.0-4</version>
    </dependency>
```

TIFF and JPEG2000

If you plan to write TIFFs with Tika (rendering of PDF pages for OCR), and if the BSD-3 with nuclear disclaimer license is acceptable to you, include:

jai-imageio-core

```
<dependency>
  <groupId>com.github.jai-imageio</groupId>
  <artifactId>jai-imageio-core</artifactId>
  <version>1.4.0</version>
</dependency>
```

If you plan on processing JPEG2000 images (most common use case would be rendering PDF pages for OCR), and if the BSD-3 with nuclear disclaimer license is acceptable to you, include:

jpeg2000

```
<dependency>
  <groupId>com.github.jai-imageio</groupId>
  <artifactId>jai-imageio-jpeg2000</artifactId>
  <version>1.4.0</version>
</dependency>
```

Note! In 2.x, Tika will not warn you if a PDF page that you're trying to render has a JPEG2000 in it. PDFBox will log a warning.

tika-app

tbd

tika-server

General

- enableFileUrl has been removed in favor of two separate fetchers, one for files and one for URLs (see tika-pipes#FetchersInClassicServerEndpoints).
    - `FileSystemFetcher` (which is packaged with tika-core) for files
    - `HttpFetcher` (requires an external jar from https://mvnrepository.com/artifact/org.apache.tika/tika-fetcher-http) for URLs.

Configuration

tika-pipes

See the tika-pipes page.

tika-eval

tika-langid

In the 1.x branch, the default (hardwired) language identification component was the wrapper around Optimaize. If you used the following in 1.x:

pom.xml 1.27

```
<dependency>
  <groupId>org.apache.tika</groupId>
  <artifactId>tika-langdetect</artifactId>
  <version>1.27</version>
</dependency>
```

In 2.x, change this to:

optimaize-lang-detect

```
<dependency>
  <groupId>org.apache.tika</groupId>
  <artifactId>tika-langdetect-optimaize</artifactId>
  <version>2.0.x</version>
</dependency>
```

The original language id component that was built by Tika devs and that used to be in tika-core is now in the tika-langdetect-tika module.