Troubleshooting Tika

Troubleshooting Apache Tika

Apache Tika is great when it works, but by default can be silently forgiving of configuration mistakes. Here we'll try to cover some of the main problems, and how to go about diagnosing them

Note that while the underlying cause is often the same no matter how you call Tika, the way of telling what's wrong can vary between them.

- Troubleshooting Apache Tika
  - Wrong Content Extracted
  - No Content Extracted
  - Wrong Parser Used
  - Content Incorrectly Detected
  - Parsers Missing
  - Detectors Missing
  - Mime Type Missing
  - Identifying your Tika Version
    - Tika App
    - Tika Server
    - Tika Facade
    - Tika Java classes
  - Identifying what Mime Types your Tika install supports
    - Tika App
    - Tika Server
    - Tika Facade
    - Tika Java classes
  - Identifying what Parsers your Tika install supports
    - Tika App
    - Tika Server
    - Tika Facade
    - Tika Java classes
  - Identifying what Detectors your Tika install supports
    - Tika App
    - Tika Server
    - Tika Facade
    - Tika Java classes
  - Identifying if any Parsers failed to be loaded
  - Identifying if any Detectors failed to be loaded
  - PDF Text Problems

Wrong Content Extracted

- Make sure you're passing Tika the source file you meant to pass, and it hasn't been corrupted in the transfer process
- Make sure Tika is able to correctly detect your file's type, see **Content Incorrectly Detected**
- Make sure Tika used the parser you meant it to, see **Wrong Parser Used**
- Make sure you're actually using the version of Tika you meant to use! See **Identifying your Tika Version**
- Problems with a PDF? See **PDF Text Problems**

No Content Extracted

- Make sure Tika is able to correctly detect your file's type, see **Content Incorrectly Detected**
- Make sure Tika has the parser for your format, and its dependencies, available and working. See **Parsers Missing**
- Make sure you're actually using the version of Tika you meant to use! See **Identifying your Tika Version**

Wrong Parser Used

- Make sure Tika is able to correctly detect your file's type, see **Content Incorrectly Detected**
- Make sure the parser you wanted to use is available to Tika. See **Identifying what Parsers your Tika install supports**, **Parsers Missing** and **Id entifying is any Parsers failed to be loaded**

Content Incorrectly Detected

Tika detects content types based on mime magic, format (normally container) specific detectors, content type hints and filename hints.

Things to check:

- Does Tika know about your type? See **Identifying what Mime Types your Tika install supports**
- If the mime type isn't listed there, see **Mime Type Missing**
- Does Tika have all its detectors? See **Identifying what Detectors your Tika install supports** and **Detectors Missing**
- Is your file a different version of the format? Check the first few hundred bytes in a hex editor, and compare to the built-in mime type

Parsers Missing

In order for a Parser to be loaded by Apache Tika, it needs:

- The parser class to be on the classpath *at runtime*
- And all of its dependencies
- For most parsers, that means the *tika-parsers* jar and dependencies
- One of:
    - a Tika Config which explicitly lists the parser class
    - a Tika Config (eg default one) which uses DefaultParser *and* a service file for the parser *and* no exclusion of that parser or parser's type

To check what parsers you have, see **Identifying what Parsers your Tika install supports**

To check if any parsers were defined but failed to load see **Identifying if any Parsers failed to be loaded**

To create a service file for auto-loading, see the quickstart guide

Detectors Missing

In order for a Detector to be loaded by Apache Tika, it needs:

- The detector class to be on the classpath *at runtime*
- And all of its dependencies
- For most detectors, that means the *tika-parsers* jar and dependencies (the container detectors are generally stored along with the parsers)
- One of:
    - a Tika Config which explicitly lists the detector class
    - a Tika Config (eg default one) which uses DefaultDetector *and* a service file for the detector

To check what detectors you have, see **Identifying what Detectors your Tika install supports**

To check if any detectors were defined but failed to load see **Identifying if any Detectors failed to be loaded**

Mime Type Missing

- If Tika doesn't out of the box, you need to add a custom mimetypes file. See the quick guide for how
- If you have written a custom mimetypes file, it needs to be present on your classpath at runtime with the *exact* name of `org/apache/tika/mime/custom-mimetypes.xml` . Double check you added it to your classpath, it has exactly that name (no typos, no prefix directories, no suffixes etc), and use **Identifying what Mime Types your Tika install supports** to see if you've loaded it or not

Identifying your Tika Version

Tika App

```
java -jar tika-app-blah.jar --version
```

Tika Server

Go to `http://localhost:9998/version`

Tika Facade

```
// Get your Tika object, eg
Tika tika = new Tika();
// Call toString() to get the version
String version = tika.toString();
```

Tika Java classes

```
// Get your Tika Config, eg
TikaConfig config = TikaConfig.getDefaultConfig();
// Go via the Tika Facade
String version = (new Tika(config)).toString();
```

Identifying what Mime Types your Tika install supports

Tika App

```
java -jar tika-app-blah.jar --list-supported-types
```

Tika Server

Go to http://localhost:9998/mime-types

Tika Facade

This is not directly possible from the Tika Facade class. Instead, follow the **Tika Java classes** route below

Tika Java classes

```
// Get your Tika Config, eg
TikaConfig config = TikaConfig.getDefaultConfig();
// Get the registry
MediaTypeRegistry registry = config.getMediaTypeRegistry();
// List
for (MediaType type : registry.getTypes()) {
   String typeStr = type.toString();
}
```

Identifying what Parsers your Tika install supports

Tika App

```
java -jar tika-app-blah.jar --list-parsers
```

Tika Server

Go to http://localhost:9998/parsers

Tika Facade

```
// Get your Tika object, eg
Tika tika = new Tika();
// Get the root parser
CompositeParser parser = (CompositeParser)parser.getParser();
// Fetch the types it supports
for (MediaType type : parser.getSupportedTypes(new ParseContext())) {
   String typeStr = type.toString();
}
// Fetch the parsers that make it up (note - may need to recurse if any are a CompositeParser too)
for (Parser p : parser.getAllComponentParsers()) {
   String parserName = p.getClass().getName();
}
```

Tika Java classes

```
// Get your Tika Config, eg
TikaConfig config = TikaConfig.getDefaultConfig();
// Get the root parser
CompositeParser parser = (CompositeParser)parser.getParser();
// Fetch the types it supports
for (MediaType type : parser.getSupportedTypes(new ParseContext())) {
   String typeStr = type.toString();
}
// Fetch the parsers that make it up (note - may need to recurse if any are a CompositeParser too)
for (Parser p : parser.getAllComponentParsers()) {
   String parserName = p.getClass().getName();
   if (p instanceof CompositeParser) {
      // Check child ones too
   }
}
```

Identifying what Detectors your Tika install supports

Tika App

`java -jar tika-app-blah.jar --list-detectors`

Tika Server

Go to `http://localhost:9998/detectors`

Tika Facade

```
// Get your Tika object, eg
Tika tika = new Tika();
// Get the root detector
CompositeDetector detector = (CompositeDetector)parser.getDetector();
// Fetch the detectors that make it up (note - may need to recurse if any are a CompositeDetector too)
for (Detector d : parser.getDetectors()) {
   String detectorName = d.getClass().getName();
}
```

Tika Java classes

```
// Get your Tika Config, eg
TikaConfig config = TikaConfig.getDefaultConfig();
// Get the root detector
CompositeDetector detector = (CompositeDetector)parser.getDetector();
// Fetch the detectors that make it up (note - may need to recurse if any are a CompositeDetector too)
for (Detector d : parser.getDetectors()) {
   String detectorName = d.getClass().getName();
   if (d instanceof CompositeDetector) {
      // Check child ones too
   }
}
```

d

Identifying if any Parsers failed to be loaded

When staring your JVM, if you pass in `-Dorg.apache.tika.service.error.warn=true` then you'll get warnings logged if any Parsers or Detectors couldn't be loaded. With the default logging configuration, you'll see things like this printed to your standard output of the JVM:

```
WARNING: Unable to load org.apache.tika.parser.microsoft.OfficeParser
java.lang.NoClassDefFoundError: org/apache/poi/poifs/filesystem/DirectoryEntry
        at java.lang.Class.getDeclaredConstructors0(Native Method)
        at java.lang.Class.privateGetDeclaredConstructors(Class.java:2585)
        at java.lang.Class.getConstructor0(Class.java:2885)
        at java.lang.Class.newInstance(Class.java:350)
        at org.apache.tika.config.ServiceLoader.loadStaticServiceProviders(ServiceLoader.java:315)
        at org.apache.tika.parser.DefaultParser.getDefaultParsers(DefaultParser.java:52)
        at org.apache.tika.parser.DefaultParser.<init>(DefaultParser.java:61)
        at org.apache.tika.parser.DefaultParser.<init>(DefaultParser.java:66)
        at org.apache.tika.config.TikaConfig.getDefaultParser(TikaConfig.java:76)
        at org.apache.tika.config.TikaConfig.<init>(TikaConfig.java:182)
        at org.apache.tika.config.TikaConfig.getDefaultConfig(TikaConfig.java:291)
        at org.apache.tika.Tika.<init>(Tika.java:115)
        at org.apache.tika.cli.TikaCLI.version(TikaCLI.java:629)
        at org.apache.tika.cli.TikaCLI.process(TikaCLI.java:365)
        at org.apache.tika.cli.TikaCLI.main(TikaCLI.java:134)
Caused by: java.lang.ClassNotFoundException: org.apache.poi.poifs.filesystem.DirectoryEntry
        at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
        ... 15 more
```

In this case, the error is telling us that we're missing the Apache POI jars which are a required dependency of Tika Parsers, and of the org.apache.tika.parser.microsoft.OfficeParser parser.

TODO describe how to use a ServiceLoader.LoadErrorHandler.ERROR to trigger an exception

Identifying if any Detectors failed to be loaded

When staring your JVM, if you pass in `-Dorg.apache.tika.service.error.warn=true` then you'll get warnings logged if any Parsers or Detectors couldn't be loaded. With the default logging configuration, you'll see things like this printed to your standard output of the JVM:

```
WARNING: Unable to load org.apache.tika.parser.microsoft.POIFSContainerDetector
java.lang.NoClassDefFoundError: org/apache/poi/poifs/filesystem/DirectoryEntry
        at java.lang.Class.getDeclaredConstructors0(Native Method)
        at java.lang.Class.privateGetDeclaredConstructors(Class.java:2585)
        at java.lang.Class.getConstructor0(Class.java:2885)
        at java.lang.Class.newInstance(Class.java:350)
        at org.apache.tika.config.ServiceLoader.loadStaticServiceProviders(ServiceLoader.java:315)
        at org.apache.tika.detect.DefaultDetector.getDefaultDetectors(DefaultDetector.java:55)
        at org.apache.tika.detect.DefaultDetector.<init>(DefaultDetector.java:66)
        at org.apache.tika.config.TikaConfig.getDefaultDetector(TikaConfig.java:71)
        at org.apache.tika.config.TikaConfig.<init>(TikaConfig.java:183)
        at org.apache.tika.config.TikaConfig.getDefaultConfig(TikaConfig.java:291)
        at org.apache.tika.Tika.<init>(Tika.java:115)
        at org.apache.tika.cli.TikaCLI.version(TikaCLI.java:629)
        at org.apache.tika.cli.TikaCLI.process(TikaCLI.java:365)
        at org.apache.tika.cli.TikaCLI.main(TikaCLI.java:134)
Caused by: java.lang.ClassNotFoundException: org.apache.poi.poifs.filesystem.DirectoryEntry
        at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
        ... 14 more
```

In this case, the error is telling us that we're missing the Apache POI jars which are a required dependency of Tika Parsers, and of the org.apache.tika.parser.microsoft.POIFSContainerDetector detector.

TODO describe how to use a ServiceLoader.LoadErrorHandler.ERROR to trigger an exception

PDF Text Problems

If Tika isn't extracting the right text from a PDF, and/or is giving errors, the first thing to do is identify if this is a Tika issue, or an issue with the underlying Apache PDFBox library used, or an issue with the PDF itself.

To check, grab the latest Apache PDFBox pdfbox-app jar and use the ExtractText command line tool on your problematic PDF:

```
java -jar pdfbox-app.X.Y.jar ExtractText problematicPDF.pdf
```

If PDFBox reports that there are unmapped Unicode characters or other problems, there may be a problem with the PDF itself. Try opening it in, for example, Adobe Reader and "saving as text" or copying and pasting the text.

If the "saved text" is just as errorful as what Tika was extracting, there's a problem with the PDF file itself.

If the "saved text" is in good shape, then there may be a problem in PDFBox. In which case, please file an Apache PDFBox bug report and attach at least one failing file to the bug. When that gets fixed, Tika will pick up the new release and will get the fix.

If PDFBox ExtractText works fine, it may* be a Tika bug. Please report an Apache Tika bug, attach at least one failing file, and mention that PDFBox ExtractText doesn't have the issue.

*PDFBox's ExtractText does not pull text from Annotations or Acroforms, so it is possible that a problem not encountered by PDFBox's ExtractText reveals a bug in Annotations or Acroforms; might be a bug in Tika, too. When in doubt, ask.

See also: PDFParser notes.