TikaProposal

Tika, a content analysis toolkit

Abstract

Tika is a toolkit for detecting and extracting metadata and structured text content from various documents using existing parser libraries.

Proposal

The Tika content analysis toolkit will include features for detecting the content types, character encodings, languages, and other characteristics of existing documents and for extracting structured text content from the documents.

The toolkit is targeted especially for search engines and other content indexing and analysis tools, but will be useful also for other applications that need to extract meaningful information from documents that might be presented as nothing else than binary streams.

Instead of implementing its own document parsers, Tika will use existing parser libraries like Jakarta POI and PDFBox.

Background

The initial idea for the Tika project was voiced in April 2006 by Jérôme Charron and Chris A. Mattman on the Nutch mailing list. The Nutch parser framework and other content analysis features were seen as value-added components that would benefit also other projects. The idea received positive feedback, but lacked the momentum.

The idea was revisited in August 2006 when Jukka Zitting from the Jackrabbit project contacted Nutch for possible cooperation with similar ideas. The original Tika idea gained extra momentum and a Google Code project was set up as a staging area for prototype code before deciding how to best handle the setup of a new project. After a few initial commits the activity again declined.

In January 2007 the idea started gaining more momentum when Rida Benjelloun offered to contribute the Lius project to Apache Lucene and when Mark Harwood also started looking for a generic toolkit like Tika.

This proposal is the result of the above efforts and related discussions both in private and on various public forums. Some alternatives to incubation, like Apache Labs or Jakarta Commons, came up during the discussions but we believe that taking the project to the Incubator is the best way to start growing a viable community to sustain the Tika toolkit.

Rationale

There is ever more demand for tools that automatically analyze and index documents in various formats. Search engines, content repositories, and other tools often need to extract metadata and text content from documents given as nothing or little else than a simple octet stream. While there are a number of existing parser libraries for various document types, each of them comes with a custom API and there are no generic tools for automatically determining which parser to use for which documents. Currently many projects end up creating their custom content analysis and extraction tools.

The Tika project attempts to remove this duplication of efforts. We believe that by pooling the efforts of multiple projects we will be able to create a generic toolkit that exceeds the capabilities and quality of the custom solutions of any single project. A generic toolkit project will also provide common ground for the developers of parser libraries and content applications to interact.

Initial Goals

The initial goals of the proposed project are:

- Viable community around the Tika codebase
- Active relationships and possible cooperation with related projects and communities
- Generic parser API for extracting structured text content from various document formats
- Flexible metadata detection and extraction API
- Java implementations of the metadata standards mentioned below

Current Status

Meritocracy

All the initial committers are familiar with the meritocracy principles of Apache, and have already worked on the various source codebases. We will follow the normal meritocracy rules also with other potential contributors.

Community

There is not yet a clear Tika community. Instead we have a number of people and related projects with an understanding that a shared toolkit project would best serve everyone's interests. The primary goal of the incubating project is to build a self-sustaining community around this shared vision.

Core Developers

The initial set of developers comes from various backgrounds, with different but compatible needs for the proposed project.

Alignment

As a generic toolkit the Tika will likely be widely used by various open source and commercial projects both together with and independent of other Apache tools like Lucene Java or Jakarta POI. Other Apache projects like Nutch and Jackrabbit are potential candidates for using Tika as an embedded component.

Known Risks

Orphaned products

There are a number of projects at various stages of maturity that implement a subset of the proposed features in Tika. For many potential users the existing tools are already enough, which reduces the demand for a more generic toolkit. This can also be seen in the slow progress of this proposal over the past year.

However, once the project gets started we can quickly reach the feature level of existing tools based on seed code from sources mentioned below. After that we believe to be able to quickly grow the developer and user communities based on the benefits of a generic toolkit over custom alternatives.

Inexperience with Open Source

All the initial developers have worked on open source before and many are committers and PMC members within other Apache projects.

Homogenous Developers

The initial developers come from a variety of backgrounds and with a variety of needs for the proposed toolkit.

Reliance on Salaried Developers

Some of the developers are paid to work on this or related projects, but the proposed project is not the primary task for anyone.

Relationships with Other Apache Products

Tika is related to at least the following Apache projects. None of the projects is a direct competitor for Tika, but there are many cases of potential overlap in functionality.

- Apache Lucene - The analysis part of Lucene contains code that might overlap with some of the potential Tika functionality. There might also be some overlap regarding the Document model in Lucene.
- Lucene Nutch - The Nutch project already contains a parser framework that does many of the things that Tika is designed to do.
- Apache Jackrabbit - The Jackrabbit project contains a text extraction component that also implements a subset of the proposed Tika features.
- Apache UIMA - The UIMA project provides a framework and pluggable tools for analyzing text content and extracting information. Example tools include language identification, sentence boundary detection and "entity extraction" - finding references to people, places and organisations. Tika could be used by UIMA to parse text but Tika should be careful not to duplicate the subsequent text analysis features UIMA offers.

A Excessive Fascination with the Apache Brand

All of us are familiar with Apache and we have participated in Apache projects as contributors, committers, and PMC members. We feel that the Apache Software Foundation is a natural home for a project like this.

Documentation

There are bits and pieces of design discussions and other documentation around, see for example the following:

- August 2006 nutch-dev: Parser design
- September 2006 nutch-dev: Content type detection
- October 2006 Lius tutorial
- February 2007 Tika wiki: Design discussion

Standards and conventions related to Tika include the Dublin Core metadata set, the Shared MIME information draft specification from freedesktop.org, and of course RFCs 2046 and 3066 for identifying media types and languages.

See also the potential parser libraries listed below for details on the various document formats that Tika plans to support.

Initial Source

Tika will start with a combination of seed code from the efforts listed below:

- The Apache Nutch project that contains a parser framework and various content analysis tools
- The Lius project, an indexing framework for Apache Lucene

- The Apache Jackrabbit project that contains a text extraction component

No existing codebase is selected as "the" starting point of Tika to avoid inheriting the world view and design limitations of any single project.

Source and Intellectual Property Submission Plan

All seed code and other contributions will be handled through the normal Apache contribution process.

We will also contact other related efforts for possible cooperation and contributions.

External Dependencies

Tika will depend on a number of external parser libraries with various licensing conditions. An initial list of potential dependencies is shown below.

Library	License
Jakarta POI	ASLv2
PDFBox	BSD
NekoHTML	CyberNeko (like ASL)
JTidy	W3C

There are also some LGPL parser libraries that would be useful. Whether and how such dependencies could be handled will be discussed during incubation. No such dependencies will be added to the project before the legal implications have been cleared.

Cryptography

Tika itself will not use cryptography, but it is possible that some of the external parser libraries will include cryptographic code to handle features like DRM in various document formats.

Required Resources

Mailing lists

- tika-dev@incubator.apache.org
- tika-commits@incubator.apache.org
- tika-private@incubator.apache.org

Subversion Directory

- https://svn.apache.org/repos/asf/incubator/tika

Issue Tracking

- JIRA Tika (TIKA)

Other Resources

- none

Initial Committers

Name	Email	CLA
Rida Benjelloun	rida dot benjelloun at doculibre dot com	yes
Mark Harwood	mharwood at apache dot org	yes
Chris A. Mattmann	mattmann at apache dot org	yes
Sami Siren	siren at apache dot org	yes
Jukka Zitting	jukka at apache dot org	yes

Affiliations

Name	Affiliation
Rida Benjelloun	Doculibre inc.

Chris A. Mattmann	NASA Jet Propulsion Laboratory
Jukka Zitting	Day Management AG

Sponsors

Champion

- Jukka Zitting (jukka at apache dot org)

Nominated Mentors

- Doug Cutting (cutting at apache dot org)
- Bertrand Delacretaz (bdelacretaz at apache dot org)
- Jukka Zitting (jukka at apache dot org)

Sponsoring Entity

- Apache Lucene