



This repository Search

Explore Gist Blog Help



terrywbrady



Georgetown-University-Libraries / File-Analyzer

forked from terrywbrady/File-Analyzer

Unwatch 5

★ Unstar 13

Fork 7

File Analyzer Training Code4Lib 2014

Terry Brady edited this page 4 days ago · 24 revisions

Edit

New Page

Pre-conference Preparation Tasks

1. Install and build the File Analyzer (required): [Installation instructions](#)
2. Send Terry a quick note confirming that you were able to complete the installs. At the end of the pre-conference session, we will code a custom File Analyzer rule. In your email, indicate your level of experience/comfort programming in Java. This portion of the session will be tailored to the experience of the audience.
3. A Java IDE is recommended for last portion of the pre-conference. If you do not already have a Java IDE available, consider installing the Eclipse Standard Edition: <https://www.eclipse.org/downloads/>

Training Outline

- File Analyzer Overview
- Try it yourself
- Demonstration of highly customized File Analyzer Rules
- Your ideas for future customizations
- Coding a File Analyzer rule

Overview Documentation

- [File Analyzer Documentation](#)

Demonstration of basic tasks

User documentation is available at the link listed above.

- [User Interface - Search the File System](#)
- [User interface - viewing results](#)
- [Sorting results](#)
- [Filtering results](#)
- [Exporting results](#)
- [User interface - import records from a file](#)
- [User interface - Merging and Comparing Results](#)

Try it yourself

Sample data files corresponding to these exercises will be provided at the start of the pre-conference session. Download the [exercise test files](#) from [GitHub](#). Extract the contents of the zip file after you download it.

▼ Pages 46

Find a Page...

[Bagit automation](#)[Batch processing](#)[Coding new File Test Rules and new File Import Rules](#)[Command line interface](#)[Convert ETD Metadata \(in DSpace\) to MARC](#)[Core File Import Rules](#)[Core File Test Rules](#)[Core package contents](#)[Counter compliant reports](#)[Create Ingest Folders for a Set of Files](#)[Demo package contents](#)[DSpace Institutional Repository Ingest](#)[File Analyzer at Georgetown Solutions for Access Services](#)[File Analyzer at Georgetown Solutions for Acquisitions](#)[File Analyzer at Georgetown Solutions for Digital Services](#)[File Analyzer at Georgetown Solutions for E Resources](#)[File Analyzer at Georgetown Solutions for Library Administration](#)[File Analyzer at Georgetown Solutions for Metadata Services](#)[File Analyzer at Georgetown Solutions for Preservation](#)[File Analyzer at Georgetown Solutions for Special Collections](#)[File Analyzer at Georgetown Solutions for Library Systems](#)[File analyzer component packages](#)[File analyzer stories](#)[File Analyzer Training Code4Lib 2014](#)

Exercises to try

Run "Count Files by Type" on the "01_Flash Drive Inventory" folder.

- Sort the results from highest count to lowest count. What file type occurs most frequently?

Run "Match by Name" on the "01_Flash Drive Inventory" folder.

- Which file names have been duplicated?
- Remove your open tabs

Run "Match by Base Name"

- on the PDFs folder
- run it again on the Word Docs folder
- Which word document does not have a corresponding PDF?

Remove the tabs from all of your prior tests.

Run "Sort by Checksum" looking only at image files

- on the Checksum Tests folder.
- run it again on the Checksum Tests2 folder.
- Which files are not identical between the 2 folders?
- Remove the tab for your test on the Checksum Tests2 folder.
- Export the results from your first "Sort by Checksum" task as a tab-delimited file.
Export only the key and data fields.
- Import your checksum results using "Import Delimited File"
- Use the merge tool to compare your imported file to the results from your checksum test
- No differences should exist

Customized imports

Regular Expression Parser

- Sample text: <https://www.nga.gov/collection/anA5.htm>
- Regex: `^([\^,]+), ([\^,]+)\t([\^,]+).*(\d\d\d\d\d).*(\d\d\d\d\d).*$`
- Sample text: http://en.wikipedia.org/wiki/Internet_media_type - save source as text
- Regex: `^.*<code>(.*?)</code>:.*$`

Count Key

- <http://catalog.data.gov/dataset/public-library-survey-pls-2011> (US Public libraries, 2011)
- Key column 1,2,8

Demonstration of Customized File Analyzer Rules

- Image Properties
- Page Count
- Counter compliant report validation
- Output to Bursar processing*

[File Analyzer Training Code4Lib 2015](#)

[File Analyzer Use Cases at Georgetown University](#)

[File Analyzer Use Cases at State Library of South Carolina](#)

[File Import rule](#)

[File Test Rule](#)

[Generate DSpace Ingest folders from ProQuest ETD files](#)

[Home](#)

[Identify digital derivatives](#)

[Installation instructions](#)

[Latest Features](#)

[MARC Encoding Check](#)

[Marc File Analyzer Package Contents](#)

[MARC Inventory and Serializer](#)

[MARC Record Validator](#)

[Outsourced MARC Record Validation](#)

[Read checksum files](#)

[User interface import records from a file](#)

[User interface Merging and Comparing Results](#)

[User Interface Search the File System](#)

[User interface viewing results](#)

[User interface watching a task in progress](#)

[User interface overview](#)

- [Home](#)
- [File Analyzer Component Packages](#)
- [Installation instructions](#)
- [File Analyzer Stories](#)
- [File Analyzer Use Cases at Georgetown University](#)
- [Latest Features](#)
- [User Interface Overview](#)
- [Command Line Interface](#)
- [Batch Processing](#)
- [Coding new File Test Rules and new File Import Rules](#)

- [File-Analyzer-Training-Code4Lib-2015](#)

Clone this wiki locally

<https://github.com/Georgetown>



Clone in Desktop

- Invoice processing*
- Identify digital derivatives
- ETD Processing for DSpace ingest

**institution specific solution*

Discussion: Your ideas for future enhancements

Creating a File Test Rule or a File Import Rule

- [Coding new File Test Rules and new File Import Rules](#)

Coding a File Test Rule or File Import Rule

The project to be implemented will be determined by the interest of the group.

Parse MARC records and validate custom business logic

- [MARC-File-Analyzer](#)
- Sample MARC files: https://archive.org/details/unc_catalog_marc

Analyze Digital Image Properties

- [Enhance the Image Properties Task](#)
- Some sample images: <http://commons.wikimedia.org/wiki/Libraries> (click through and download a handful)

PDF Introspection

- [Enhance the Page Count Task](#)
- Some sample PDF's: <http://code4lib.org/conference/2009/schedule> (download a handful from the bottom of the page)

+ Add a custom footer

