



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 Test Rule

Edit

New Page

Terry Brady edited this page on Aug 18, 2014 · 21 revisions

The File Analyzer Tool walks a directory tree and performs a "File Test" on each file that is encountered. The application framework allows new File Tests to be quickly developed and deployed into the application. The results of each File Test are compiled into a table that summarizes the results of the analysis.

A File Test is a simple set of actions that are performed upon a single file such as filename validation, file size statistical analysis, checksum calculation, file type extraction. Depending on the action, the content of the file may or may not be read. Each File Test is configured with filters that determine which files will be processed by the File Test (i.e. only image files).

Each File Test will generate a table of results. The number of columns and the definition of the columns will vary from test to test. For example, a file type analysis will report the file extension and the number of files discovered with that extension. The checksum file tests will report the name of a file and the checksum string associated with that file.

The File Analyzer tool can be run as a GUI in which the results are displayed in a table. The File Analyzer can also be run in batch mode. In batch mode, the results will be written to a tab-separated file. The GUI version of the application allows the results of multiple executions to be merged. The merged information can be filtered to display matching values and mismatched values.

Pages 46

- 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

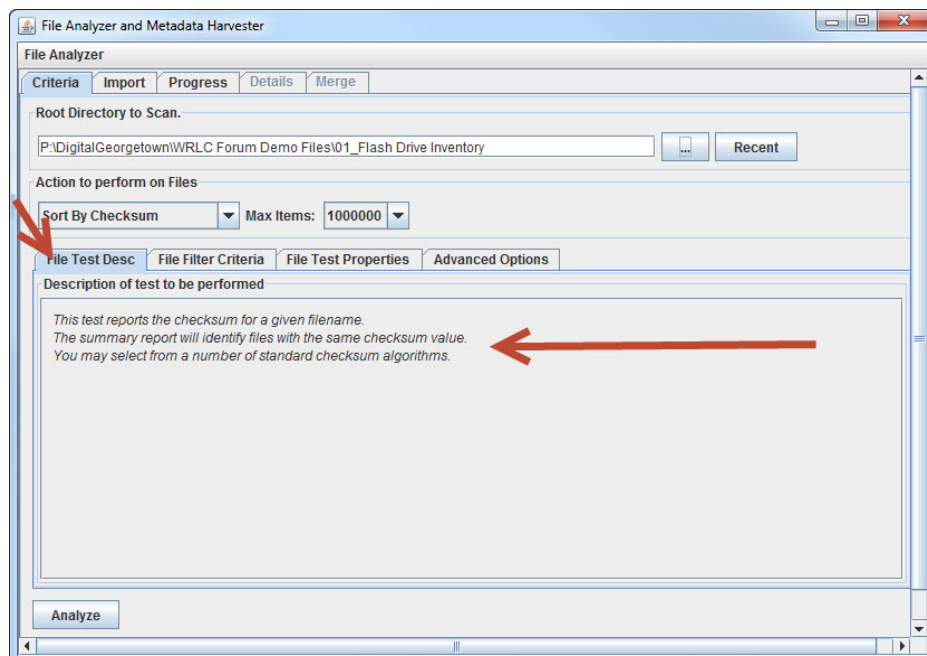
Clone this wiki locally

<https://github.com/Georgetown-University-Libraries/File-Analyzer/wiki/File-Test-Rule>

Clone in Desktop

Components of a File Test

Name and description: explains the File Test to a user



```

public class NameChecksum extends DefaultFileTest {

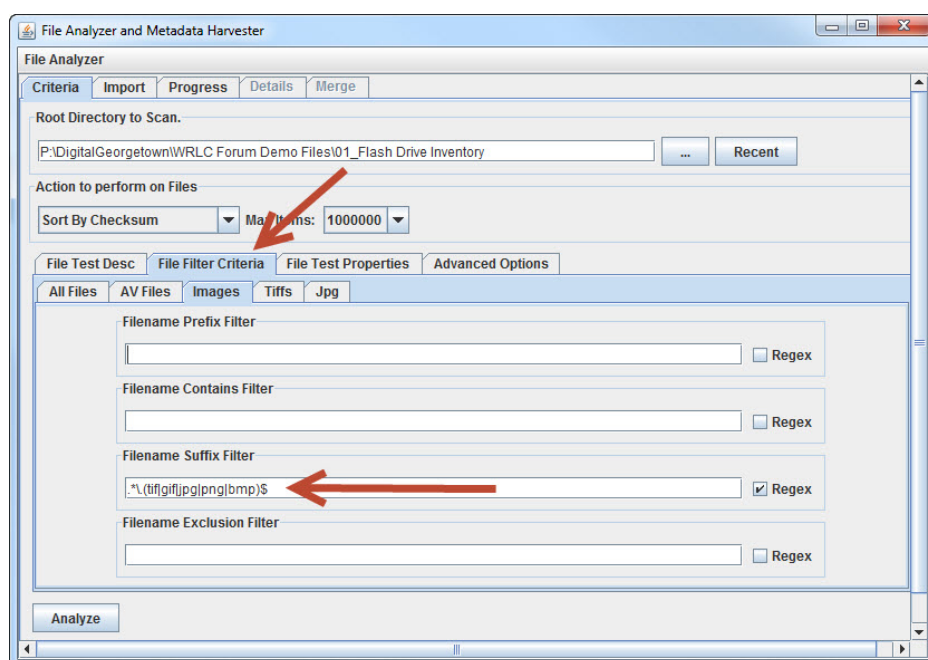
    public String toString() {
        return "Sort By Checksum";
    }

    public String getShortName(){return "Checksum";}

    public String getDescription() {
        return "This test reports the checksum for a given filename.\n" +
            "The summary report will identify files with the same checksum value.\n" +
            "You may select from a number of standard checksum algorithms.";
    }
}

```

Filters: determine the files that the test will operate upon



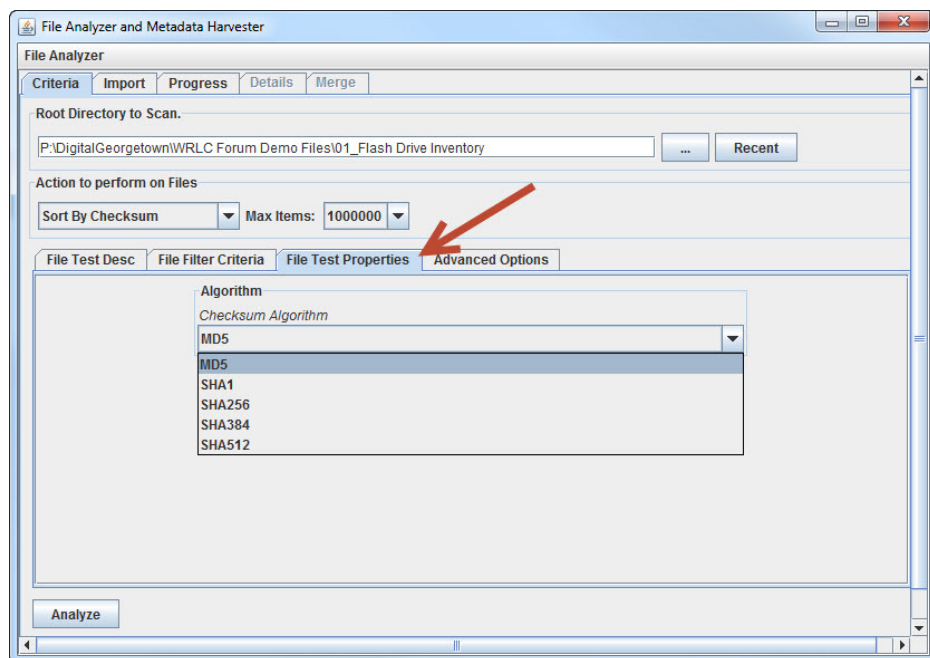
```

public void initFilters() {
    initAllFilters();
}

/* from DefaultFileTest.java */
public void initAllFilters() {
    filters.add(new DefaultFileTestFilter());
    filters.add(new AVFileTestFilter());
    filters.add(new ImageFileTestFilter());
    filters.add(new TiffFileTestFilter());
    filters.add(new JpegFileTestFilter());
}

```

Properties: runtime parameters that the user can pass to the File Test



```

public static final String ALGORITHM = "Algorithm";
static enum Algorithm {
    MD5("MD5"),
    SHA1("SHA-1"),
    SHA256("SHA-256"),
    SHA384("SHA-384"),
    SHA512("SHA-512");
    String algorithm;
    Algorithm(String s) {algorithm = s;}
    MessageDigest getInstance() throws NoSuchAlgorithmException {
        return MessageDigest.getInstance(algorithm);
    }
}

public NameChecksum(FTDriver dt) {
    super(dt);
    keymap = new HashMap<String, List<ChecksumStats>>();
    this.ftprops.add(new FTPPropEnum(dt, this.getClass().getName(), ALGORITHM, "algor
        "Checksum Algorithm", Algorithm.values(), Algorithm.MD5));
}

```

Result Stats: defines the resulting information that will be displayed to the user (as a table)

File Analyzer and Metadata Harvester

File Analyzer

Criteria Import Progress Details Merge Checksum1

Sort By Checksum for ...WRLC Forum Demo Files\01_Flash Drive Inventory

Remove Tab

Is Duplicate Duplicate Stat

Summary Counts

Key	Data	Is Duplicate	Duplicate Stat	Num of Matches
Checksum Tests2adapting-to-america.jpg	1f92844e221a335751be909766252190	Y	Duplicate	2
Checksum Tests2catholic-universities-church-society.jpg	291382b7e15e60e2d494b5c039a5d48c	Y	Duplicate	2
Checksum Tests2catholic-university-promise-project.jpg	d8bcaa2894f2b8cc74879ae1c798e082	Y	Duplicate	2
Checksum Tests2enhancing-religious-identity.jpg	09f0bfc1e65f6f1cec7be2b4ca8c88bb	Y	Duplicate	2
Checksum Tests2footnotes-to-history.jpg	7c5dd7a8cea24d57f31177c37020b59	Y	Duplicate	2
Checksum Tests2for-that-i-came.jpg	c2561a3cda70879f50a0b0933c20c3	Y	Duplicate	2
Checksum Tests2georgetown-at-200.jpg	313107907d22891f293dbfa259e009c	Y	Duplicate	2
Checksum Tests2higher-education-moral-enterprise.jpg	319c5453eb7e98d17ad9f673140bb398	Y	Duplicate	2
Checksum Tests2horace-priest-poor.jpg	1943834a96f53d5c1996a4a605b08142	Y	Duplicate	2
Checksum Tests2jesuit-education-cultivation-virtue.jpg	c89c572f2dcd9e3df6723a0d9f41e98d	N	Unique	1
Checksum Tests2labor-of-god - Copy.jpg	dcdeb437da81d3a5cdda466311cf1e62	Y	Duplicate	4
Checksum Tests2labor-of-god.jpg	dcdeb437da81d3a5cdda466311cf1e62	Y	Duplicate	4
Checksum Tests2let-justice-roll-down.jpg	1e2c0aac6c0808349dc0895e68d4181	Y	Duplicate	2
Checksum Tests2memoirs-of-a-yukon-priest.jpg	91281acbc9d06093ba45c5e8705b87	Y	Duplicate	2
Checksum Tests2minding-the-time.jpg	218c93da1150ae0f2e7d534812e9efc3	Y	Duplicate	2
Checksum Tests2riding-time-like-a-river.jpg	60fe2a259a7d398703800cd9a33e2d45	Y	Duplicate	2
Checksum Tests2splendor-and-wonder.jpg	f5c50091b61dd1bd2090cbd49667cb47	Y	Duplicate	2
Checksum Tests2swift-potomacs-lovely-daughter.jpg	22c6b022471b06dbda97bfea1f1011e8	Y	Duplicate	2
Checksum Tests2adapting-to-america.jpg	1f92844e221a335751be909766252190	Y	FirstFound	2
Checksum Tests2catholic-universities-church-society.jpg	291382b7e15e60e2d494b5c039a5d48c	Y	FirstFound	2
Checksum Tests2catholic-university-promise-project.jpg	d8bcaa2894f2b8cc74879ae1c798e082	Y	FirstFound	2

36 items. 2.237 seconds

Export Table

```

public Stats createStats(String key){
    return ChecksumStats.Generator.INSTANCE.create(key);
}

public StatsItemConfig getStatsDetails() {
    return ChecksumStats.details;
}

}

/*from ChecksumStats.java*/
public class ChecksumStats extends Stats {
    public static enum DUP {Unique, FirstFound, Duplicate;}
    public static enum ChecksumStatsItems implements StatsItemEnum {
        Key(StatsItem.makeStringStatsItem("Key", 400)),
        Data(StatsItem.makeStatsItem(Object.class, "Data", 300).setInitVal("")),
        IsDuplicate(StatsItem.makeEnumStatsItem(YN.class, "Is Duplicate").setInitVal(
        DuplicateStat(StatsItem.makeEnumStatsItem(DUP.class, "Duplicate Stat").setInitVal(
        MatchCount(StatsItem.makeIntStatsItem("Num of Matches").setInitVal(1));

        StatsItem si;
        ChecksumStatsItems(StatsItem si) {this.si=si;}
        public StatsItem si() {return si;}
    }
    public static enum Generator implements StatsGenerator {
        INSTANCE;
        public ChecksumStats create(String key) {return new ChecksumStats(key);}
    }
}

```

Result Key: defines the unique key value that will be saved for each file (or set of files) that is processed

File Analyzer and Metadata Harvester

File Analyzer

CriteriaImportProgressDetailsMergeChecksum1

Sort By Checksum for ...JWRLC Forum Demo Files\01_Flash Drive Inventory

Remove Tab

Is DuplicateDuplicate Stat

Summary Counts	Key	Data	Is Duplicate	Duplicate Stat	Num of Matches
	C:\checksum Tests2\adapting-to-america.jpg	192844e221a335751be909766252190	Y	Duplicate	2
	C:\checksum Tests2\catholic-universities-church-society.jpg	291382b7e15e60e2d494b5c039a5d48c	Y	Duplicate	2
	C:\checksum Tests2\catholic-university-promise-project.jpg	d8bcaa2894f2b8cc74879ae1c798e082	Y	Duplicate	2
	C:\checksum Tests2\enhancing-religious-identity.jpg	09f5bfc1e85b91cec7be2b4ca8c8bb	Y	Duplicate	2
	C:\checksum Tests2\footnotes-to-history.jpg	7c5dd7a8cea24d57f31177c37020b59	Y	Duplicate	2
	C:\checksum Tests2\for-that-i-came.jpg	c2561a3cda070879f50a0b0933c20c3	Y	Duplicate	2
	C:\checksum Tests2\georgetown-at-200.jpg	3131079d7d22891f293d8fa259e009c	Y	Duplicate	2
	C:\checksum Tests2\higher-education-moral-enterprise.jpg	319c5453eb7e98d17ad99f73140bb398	Y	Duplicate	2
	C:\checksum Tests2\horace-priest-poor.jpg	1943834a96f53d5c1996a4a605b08142	Y	Duplicate	2
	C:\checksum Tests2\jesuit-education-cultivation-virtue.jpg	c89c572f2dc09e3df6723a0d9f41e98d	N	Unique	1
	C:\checksum Tests2\labor-of-god - Copy.jpg	dcdeb437da81d3a5cdda466311cf1e62	Y	Duplicate	4
	C:\checksum Tests2\labor-of-god.jpg	dcdeb437da81d3a5cdda466311cf1e62	Y	Duplicate	4
	C:\checksum Tests2\et-justice-roll-down.jpg	1e2c0aaac08808349dc0895e8d4181	Y	Duplicate	2
	C:\checksum Tests2\memoirs-of-a-yukon-priest.jpg	91281acbd9f6e6933ba45c5e8705b87	Y	Duplicate	2
	C:\checksum Tests2\minding-the-time.jpg	218c93da1150ae0fe7d534812e9efc3	Y	Duplicate	2
	C:\checksum Tests2\riding-time-like-a-river.jpg	60fe2a259a7d398703800d9a33e2d45	Y	Duplicate	2
	C:\checksum Tests2\splendor-and-wonder.jpg	f5c50091b61dd1bd2090cbd49667cb47	Y	Duplicate	2
	C:\checksum Tests2\swift-potomacs-lovely-daughter.jpg	22c6b022471b06dbda97bfea1f1011e8	Y	Duplicate	2
	C:\checksum Tests2\adapting-to-america.jpg	192844e221a335751be909766252190	Y	FirstFound	2
	C:\checksum Tests2\catholic-universities-church-society.jpg	291382b7e15e60e2d494b5c039a5d48c	Y	FirstFound	2
	C:\checksum Tests2\catholic-university-promise-project.jpg	d8bcaa2894f2b8cc74879ae1c798e082	Y	FirstFound	2

36 items. 2.237 seconds

Export Table

```
public String getKey(File f) {
    return getRelPath(f);
}

/*from DefaultFileTest.java*/
public String getRelPath(File f) {
    return f.getAbsolutePath().substring(getRoot().getAbsolutePath().length());
}
```

Code the FileTest

In the example displayed above, a checksum is generated on the file using the algorithm provided by the user.

```
public String getChecksum(File f) {
    Algorithm algorithm = (Algorithm)getProperty(ALGORITHM);
    FileInputStream fis = null;
    try {
        MessageDigest md = algorithm.getInstance();
        fis = new FileInputStream(f);
        byte[] dataBytes = new byte[1204];
        int nread = 0;
        while((nread = fis.read(dataBytes)) != -1){
            md.update(dataBytes, 0, nread);
        }
        byte[] mdbytes = md.digest();
        StringBuffer sb = new StringBuffer();
        for(int i=0; i<mdbytes.length; i++){
            sb.append(Integer.toString((mdbytes[i] & 0xFF) + 0x100, 16).substring(1,
        )
        return sb.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fis!=null)
            try {
                fis.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
    }
}
```

```

    return null;
}

public Object fileTest(File f) {
    return getChecksum(f);
}

```

Indicate how to handle directories and files (beyond the filter settings)

```

public boolean isTestable(File f) {
    return true;
}

public boolean isTestDirectory() {
    return false;
}

public boolean processRoot() {
    return false;
}

public boolean isTestFiles() {
    return true;
}

```

Provide an initial task and a summary task (if needed)

```

@Override public void init() {
    keymap.clear();
}

@Override public void refineResults() {
    for(List<ChecksumStats> matches: keymap.values()) {
        if (matches.size() == 1) continue;
        int count = 0;
        for(ChecksumStats match: matches) {
            match.setVal(ChecksumStatsItems.IsDuplicate, YN.Y);
            if (count == 0) {
                match.setVal(ChecksumStatsItems.DuplicateStat, ChecksumStats.DUP.Fir
            } else {
                match.setVal(ChecksumStatsItems.DuplicateStat, ChecksumStats.DUP.Dup
            }
            count++;
            match.setVal(ChecksumStatsItems.MatchCount, matches.size());
        }
    }
}

```

Register the FileTest with the File Analyzer

```

public class ActionRegistry extends Vector<FileTest> {

    private static final long serialVersionUID = 1L;
    boolean modifyAllowed = true;

    public ActionRegistry(FTDriver dt, boolean modifyAllowed) {
        this.modifyAllowed = modifyAllowed;
    }
}

```

...

```
add(new NameChecksum(dt));
```

+ Add a custom footer

