Efficient component analysis with FOSSology

Overview: Contents

1. **Introduction and overview**
   What is FOSSology
2. **Introduction and Basic Work Flow**
   What FOSSology is designed for
3. **Features to make you efficient**
   How to work efficiently
4. **Show time**
   Workflow demo
5. **Take aways**

What is FOSSology?

A Web server application for license and copyright compliance of software components.

FOSSology Project
https://www.fossology.org/

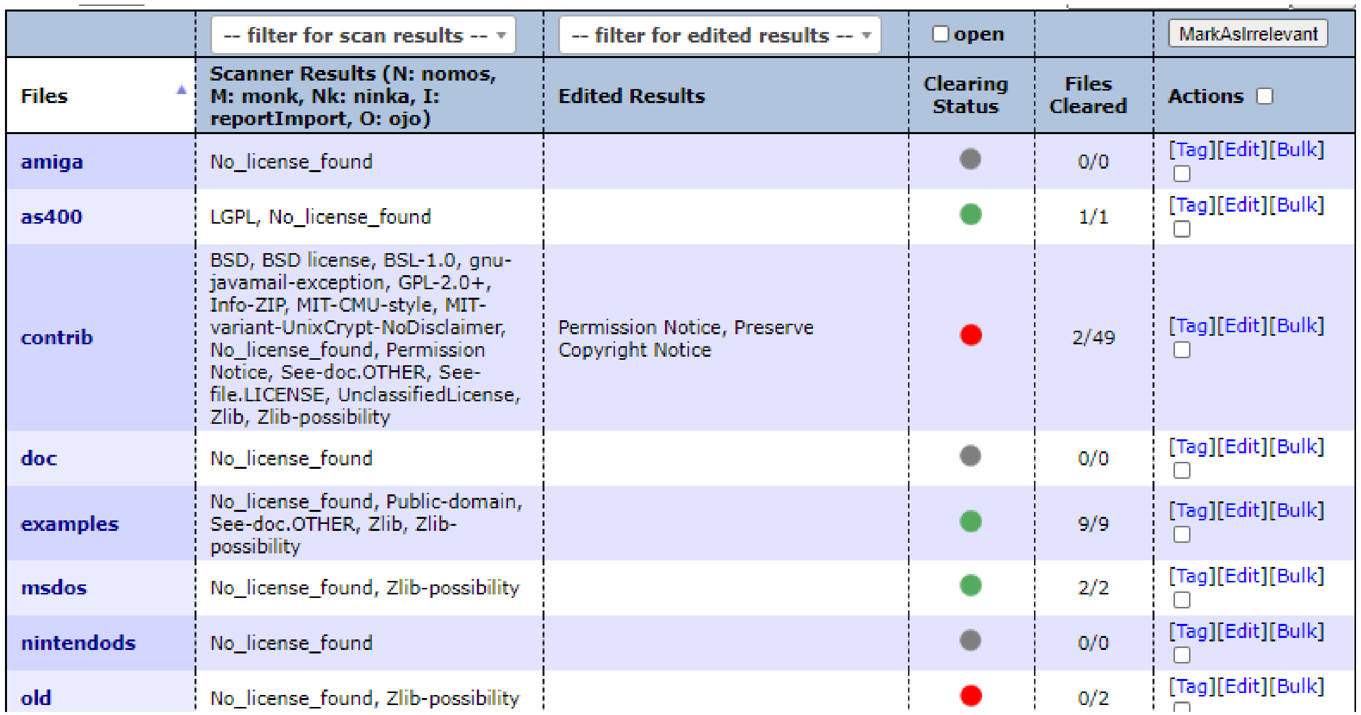
- Published first in 2008, GPL-2.0
- 2015: Linux Foundation collaboration project
- Web server based, command line interfaces and REST API
- Scanning agents searching for license and copyright relevant hits (and more …)
- A multi-user / multi-tenant Web UI for review

FOSSology Development
https://www.github.com/fossology/fossology

- Standard Web application stack:
  - Linux, Apache 2, PostgreSQL, PHP,
- Web-based UI in PHP, but scanners written in C / C++

FOSSology’s license analysis support

- FOSSology provides:
  - Finding license relevant texts
  - Aggregation in a hierarchy
  - Highlighting text occurrences
  - Identifying wording differences compared with reference texts
  - Searching for licensing phrases
  - Reporting of found licenses

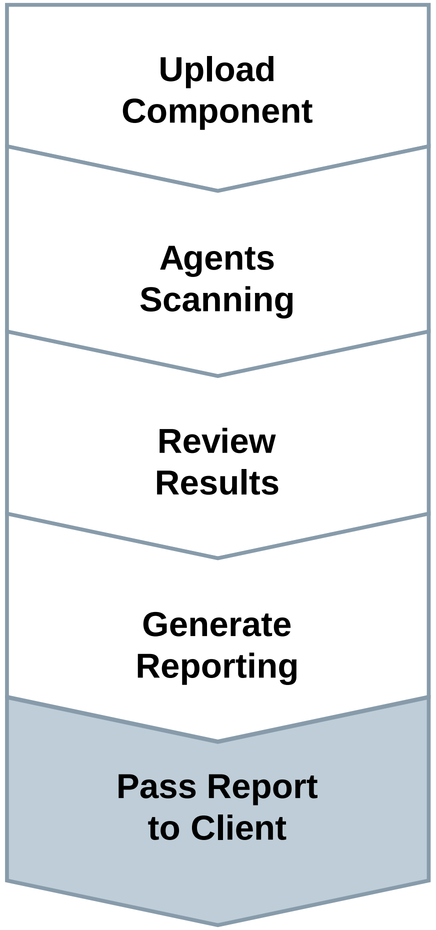


	-- filter for scan results --	-- filter for edited results --	☐ open		MarkAsIrrelevant
Files	Scanner Results (N: nomos, M: monk, Nk: ninka, I: reportImport, O: ojo)	Edited Results	Clearing Status	Files Cleared	Actions ☐
amiga	No_license_found		●	0/0	[Tag][Edit][Bulk] ☐
as400	LGPL, No_license_found		🟢	1/1	[Tag][Edit][Bulk] ☐
contrib	BSD, BSD license, BSL-1.0, gnu-javamail-exception, GPL-2.0+, Info-ZIP, MIT-CMU-style, MIT-variant-UnixCrypt-NoDisclaimer, No_license_found, Permission Notice, See-doc.OTHER, See-file.LICENSE, UnclassifiedLicense, Zlib, Zlib-possibility	Permission Notice, Preserve Copyright Notice	🔴	2/49	[Tag][Edit][Bulk] ☐
doc	No_license_found		●	0/0	[Tag][Edit][Bulk] ☐
examples	No_license_found, Public-domain, See-doc.OTHER, Zlib, Zlib-possibility		🟢	9/9	[Tag][Edit][Bulk] ☐
msdos	No_license_found, Zlib-possibility		🟢	2/2	[Tag][Edit][Bulk] ☐
nintendods	No_license_found		●	0/0	[Tag][Edit][Bulk] ☐
old	No_license_found, Zlib-possibility		🔴	0/2	[Tag][Edit][Bulk] ☐

Overview: Contents

1. **Introduction and overview**
   What is FOSSology
2. **Introduction and Basic Work Flow**
   How to interact with FOSSology
3. **Features to make you efficient**
   How to work efficiently
4. **Show time**
   Workflow demo
5. **Take aways**

How does FOSSology work?



Upload Component

Agents Scanning

Review Results

Generate Reporting

Pass Report to Client

- Uploading source code archive (*.zip, *.tar.gz, etc)

- Agents scan for license relevant text (nomos, monk, ojo)
- Copyrights, ECC, your keywords to look for etc.

- Review scanner results for wrong license classification
- Review other scanner findings (copyrights, ECC)

- Result of the “clearing”
  - SPDX reporting
  - Generated notice or readme file
  - Debian-copyright
  - Document based report

Basic End-to-End Workflow

Functionality

1. **Using FOSSology End-to-End**
   - From uploading …
   - … to generating report: SPDX
2. **Uploading - offers a variety of selections**
3. **Review the uploaded file in the license browser**
4. **Review the found licenses in the aggregated view**
5. **Do the clearing work**
6. **Review the copyrights**
7. **Review the Export Control and Customs (ECC) phrases**
8. **Generate desired report output**

Basic End-to-End Workflow

Functionality

1. **Using FOSSology End-to-End**
   - From uploading …
   - … to generating report: SPDX
2. **Uploading - offers a variety of selections**
3. **Review the uploaded file in the license browser**
4. **Review the found licenses in the aggregated view**
5. **Do the clearing work**
6. **Review the copyrights**
7. **Review the Export Control and Customs (ECC)**
8. **Generate desired report output**

License analysis can’t be fully automated

```
/*******************************************************************************
* Copyright (C) 2008 - 2015 ***, Inc.  All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* Use of the Software is limited solely to applications:
* (a) running on a *** device, or
* (b) that interact with a *** device through a bus or interconnect.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHAI
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EV
* ***  CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABIL
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
* OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*
*******************************************************************************/
```

Real world example:

- It is actually based on an MIT license text
- MIT license: very popular and permissive
- Added two conditions inside the original license text
  - (not so permissive)
- Very hard to identify with regular expression matching

License analysis can’t be fully automated

```
/*************************************************************************************
* Copyright (C) 2008 - 2015 ***, Inc.  All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
* copies of the Software, and to permit persons to whom the Software is
* furnished to do so, subject to the following conditions:
*
* The above copyright notice and this permission notice shall be included in
* all copies or substantial portions of the Software.
*
* Use of the Software is limited solely to applications:
* (a) running on a *** device, or
* (b) that interact with a *** device through a bus or interconnect.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHAI
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EV
* ***  CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABIL
* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
* OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
* SOFTWARE.
*
*************************************************************************************/
```

Real world example:

- It is actually based on an MIT license text
- MIT license: very popular and permissive
- Added two conditions inside the original license text
  - (not so permissive)
- Very hard to identify with regular expression matching

fossology

There is much creativity in expressing licenses

(from zlib-1.2.8.tar/ zlib-1.2.8/ contrib/ amd64/ amd64-match.S)

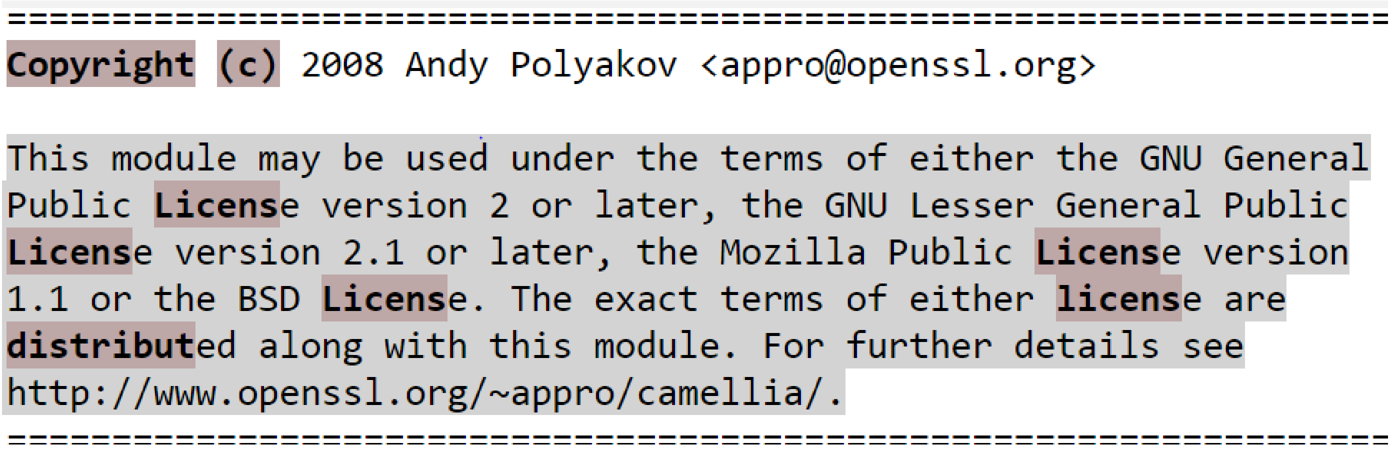
```
/*
 * match.S -- optimized version of longest_match()
 * based on the similar work by Gilles Vollant,
 * and Brian Raiter, written 1998
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the BSD License.
 * Use by owners of Che Guevarra
 * parafernalia is prohibited, where possible,
 * and highly discouraged
 * elsewhere.
 */
```

…

Another real world example:

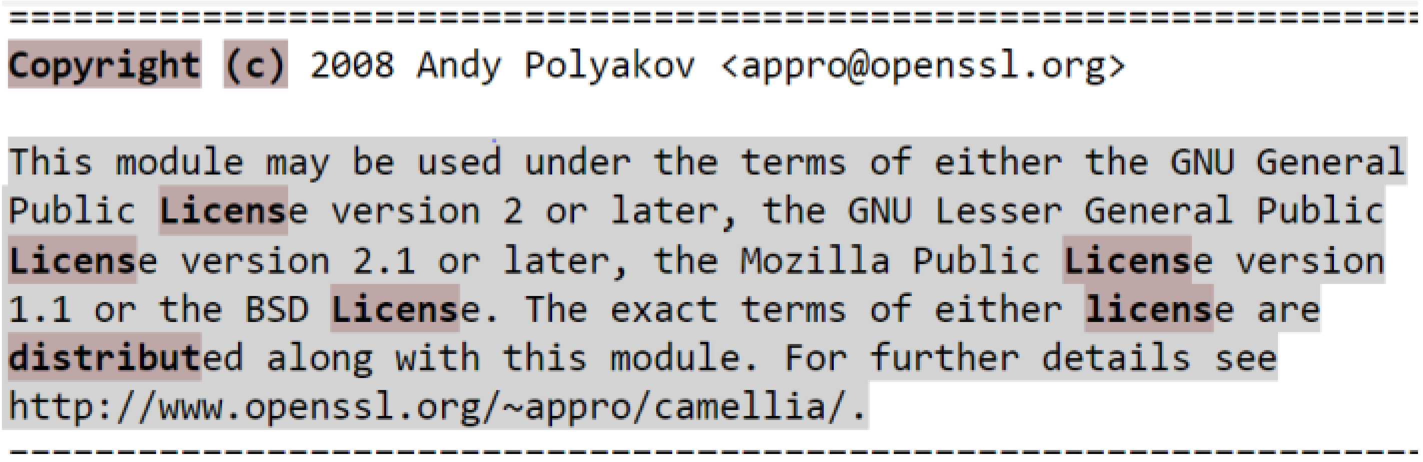
- What was meant to be fun (or a political statement), is difficult for license analysis
- *Question: Can this be ignored or shall the origination check for ownership of referred parafernalia?*

Determining the correct license requires additional work



```
==================================================================
Copyright (c) 2008 Andy Polyakov <appro@openssl.org>

This module may be used under the terms of either the GNU General
Public License version 2 or later, the GNU Lesser General Public
License version 2.1 or later, the Mozilla Public License version
1.1 or the BSD License. The exact terms of either license are
distributed along with this module. For further details see
http://www.openssl.org/~appro/camellia/.
==================================================================
```

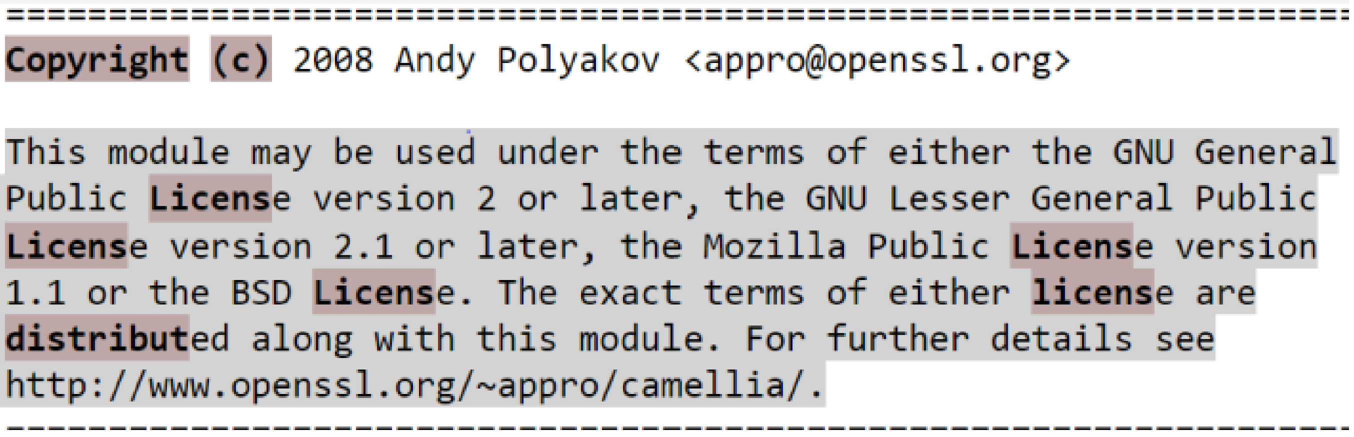
Determining the correct license requires additional work



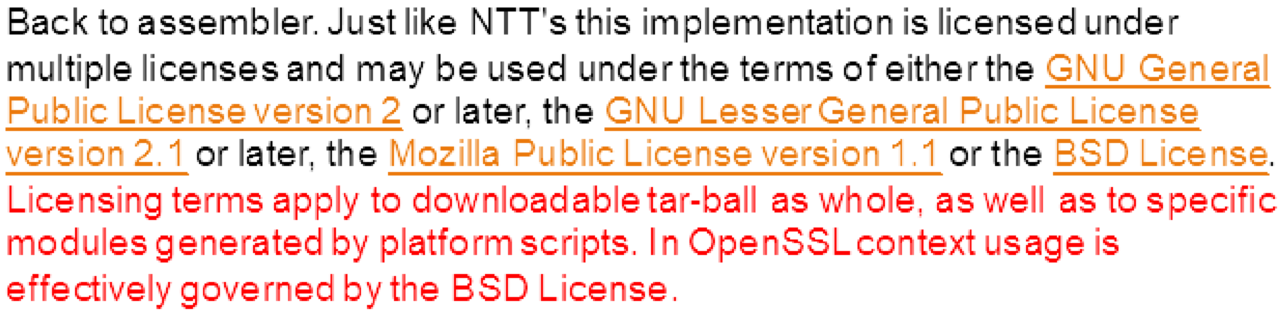
```
=====================================================================
Copyright (c) 2008 Andy Polyakov <appro@openssl.org>

This module may be used under the terms of either the GNU General
Public License version 2 or later, the GNU Lesser General Public
License version 2.1 or later, the Mozilla Public License version
1.1 or the BSD License. The exact terms of either license are
distributed along with this module. For further details see
http://www.openssl.org/~appro/camellia/.
---------------------------------------------------------------------
```

Back to assembler. Just like NTT's this implementation is licensed under multiple licenses and may be used under the terms of either the GNU General Public License version 2 or later, the GNU Lesser General Public License version 2.1 or later, the Mozilla Public License version 1.1 or the BSD License. Licensing terms apply to downloadable tar-ball as whole, as well as to specific modules generated by platform scripts. In OpenSSL context usage is effectively governed by the BSD License.

Determining the correct license requires additional work

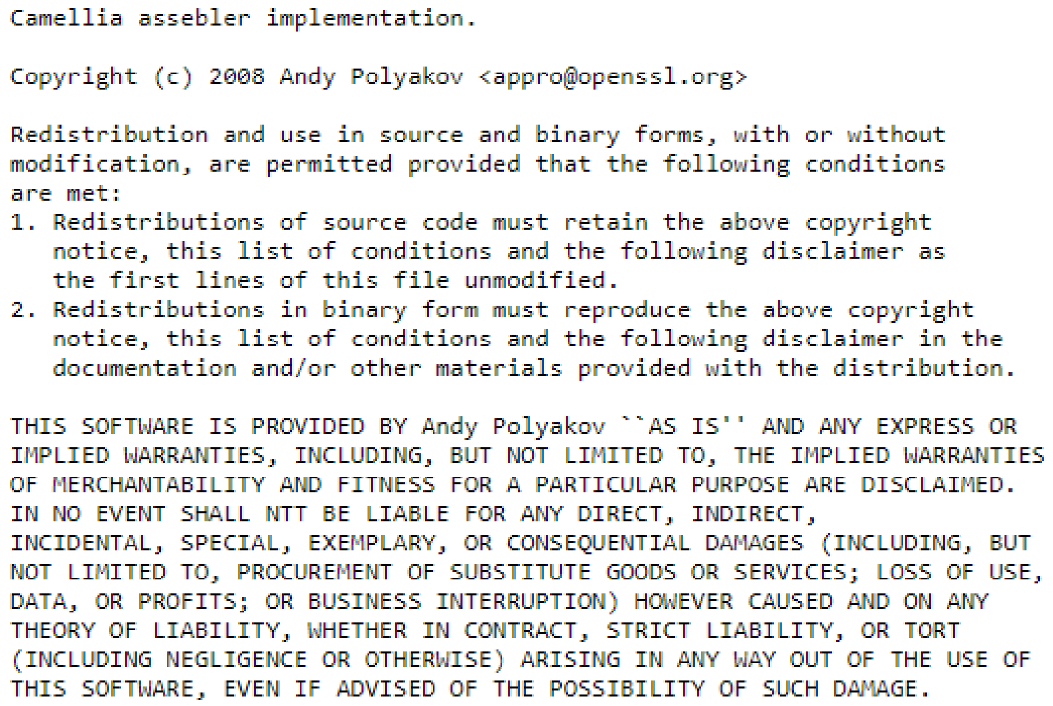


```
====================================================================
Copyright (c) 2008 Andy Polyakov <appro@openssl.org>

This module may be used under the terms of either the GNU General
Public License version 2 or later, the GNU Lesser General Public
License version 2.1 or later, the Mozilla Public License version
1.1 or the BSD License. The exact terms of either license are
distributed along with this module. For further details see
http://www.openssl.org/~appro/camellia/.
--------------------------------------------------------------------
```



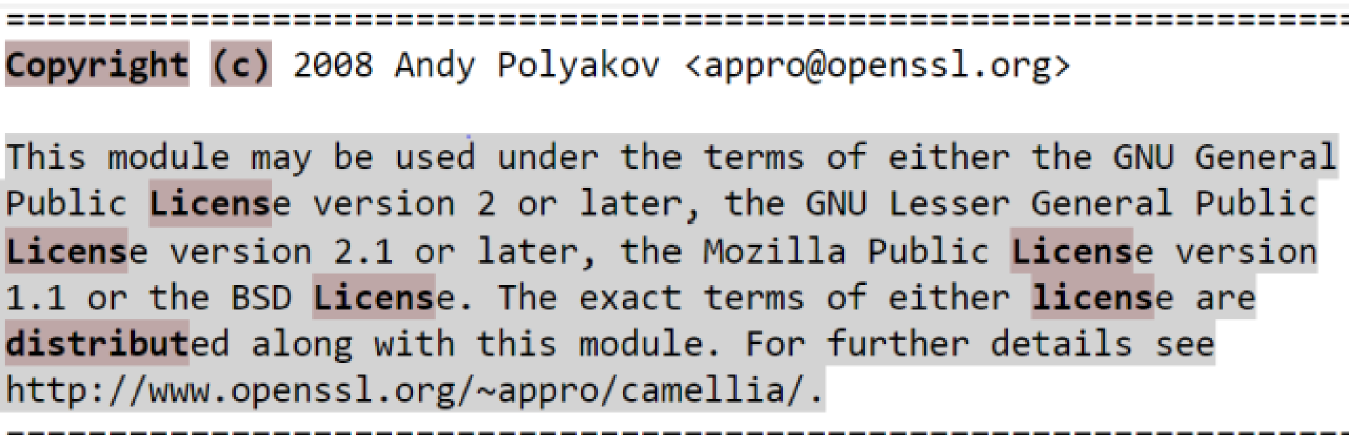
Back to assembler. Just like NTT's this implementation is licensed under multiple licenses and may be used under the terms of either the GNU General Public License version 2 or later, the GNU Lesser General Public License version 2.1 or later, the Mozilla Public License version 1.1 or the BSD License. Licensing terms apply to downloadable tar-ball as whole, as well as to specific modules generated by platform scripts. In OpenSSL context usage is effectively governed by the BSD License.

https://www.openssl.org/~appro/camellia/dist/BSD_license.txt



```
Camellia assebler implementation.

Copyright (c) 2008 Andy Polyakov <appro@openssl.org>

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer as
   the first lines of this file unmodified.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY Andy Polyakov ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL NTT BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

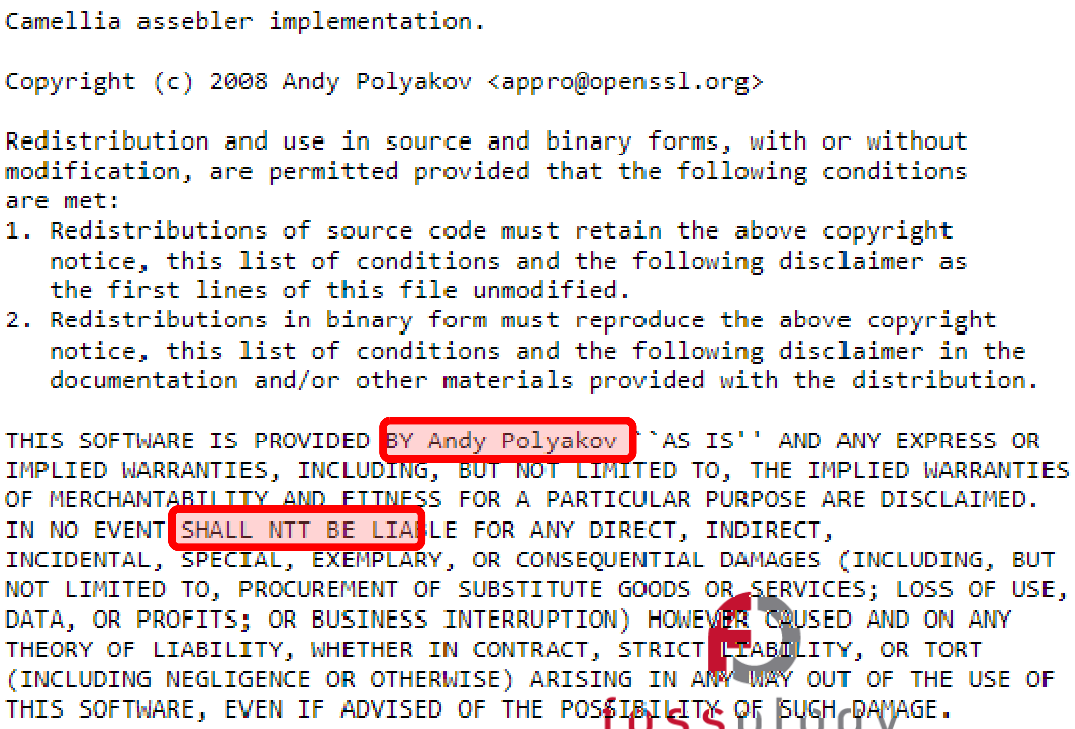
Determining the correct license requires additional work



```
====================================================================
Copyright (c) 2008 Andy Polyakov <appro@openssl.org>

This module may be used under the terms of either the GNU General
Public License version 2 or later, the GNU Lesser General Public
License version 2.1 or later, the Mozilla Public License version
1.1 or the BSD License. The exact terms of either license are
distributed along with this module. For further details see
http://www.openssl.org/~appro/camellia/.
--------------------------------------------------------------------
```

Back to assembler. Just like NTT's this implementation is licensed under multiple licenses and may be used under the terms of either the GNU General Public License version 2 or later, the GNU Lesser General Public License version 2.1 or later, the Mozilla Public License version 1.1 or the BSD License. Licensing terms apply to downloadable tar-ball as whole, as well as to specific modules generated by platform scripts. In OpenSSL context usage is effectively governed by the BSD License.

https://www.openssl.org/~appro/camellia/dist/BSD_license.txt



```
Camellia assebler implementation.

Copyright (c) 2008 Andy Polyakov <appro@openssl.org>

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer as
   the first lines of this file unmodified.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY Andy Polyakov ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL NTT BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Dual licensing expressions

(TrueCrypt 7.1a Source.zip/ Crypto/ AesSmall.h)

The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that:

1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials;
3. the copyright holder's name is not used to endorse products built using this software without specific written permission.

ALTERNATIVELY, provided that this notice is retained in full, this product may be distributed under the terms of the GNU General Public License (GPL), in which case the provisions of the GPL apply INSTEAD OF those given above.

DISCLAIMER
This software is provided 'as is' with no explicit or implied warranties in respect of its properties, including, but not limited to, correctness and/or fitness for purpose.

Another real world example:

- How does the organization decide which license to choose
- There may be an external reason for choosing either one or the another

fossology

Overview: Contents

1. **Introduction and overview**
   What is FOSSology
2. **Introduction and Basic Work Flow**
   What FOSSology is designed for
3. **Features to make you efficient**
   How to work efficiently
4. **Show time**
   Workflow demo
5. **Take aways**

Feature: Edit License Conclusions in License Browser

Edit is about a short cut at license review

Use Case

- Within a software component, I know files / folders already.
- I know that files are irrelevant.
  - Unused architecture
  - Test files, code examples
  - Build infrastructure
- **Do I need to look at the files?**

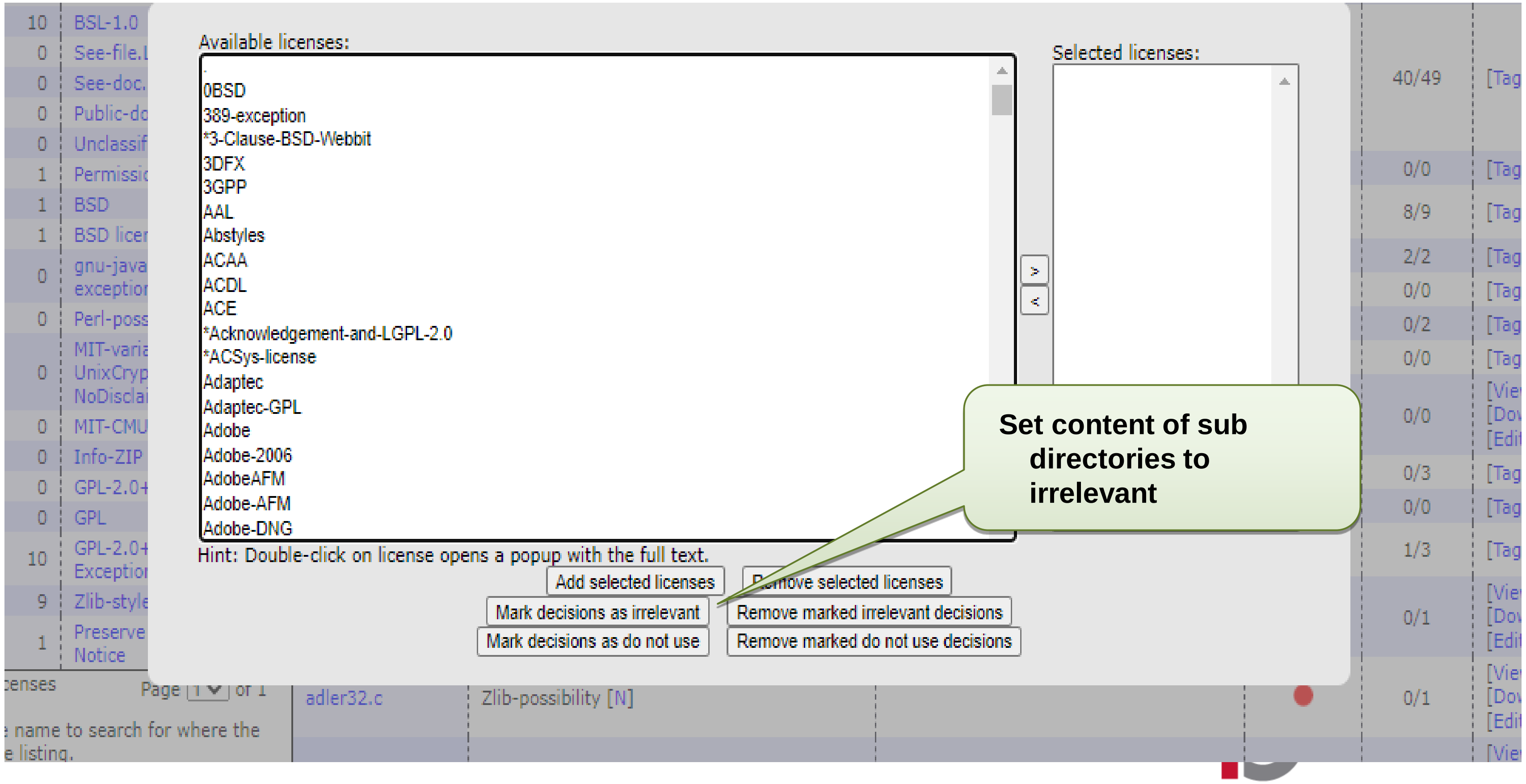
Solution

- FOSSology allow to set / unset license on folder level
- In license browser, the user can set the licenses on folders
  - Set or confirm licenses
  - Remove scan results from the clearing decision
  - Mark files as irrelevant
- What happens actually?
  - The “Edit” creates clearing decisions
  - The scan results are not touched (and preserved)
  - No clearing decision final on scan result conflicts

Feature: Edit License Conclusions in License Browser

User Interface

- At aggregated license browser view select link [Edit]
- Select either licenses for decision …
- … or mark file trees irrelevant (for distribution)



Available licenses:

0BSD
389-exception
*3-Clause-BSD-Webbit
3DFX
3GPP
AAL
Abstyles
ACAA
ACDL
ACE
*Acknowledgement-and-LGPL-2.0
*ACSys-license
Adaptec
Adaptec-GPL
Adobe
Adobe-2006
AdobeAFM
Adobe-AFM
Adobe-DNG

Selected licenses:

Hint: Double-click on license opens a popup with the full text.

Set content of sub directories to irrelevant

Feature: Reviewing Many Files at Once: Bulk Scan

The FOSSology software cannot know all possible license relevant text phrases

Use Case

- Finding standard and known license texts is straightforward
- ***Do I need to create a clearing decision for every file?***
- ***If the scanner yields wrong results, do I need to go into every file and correct then?***

Solution

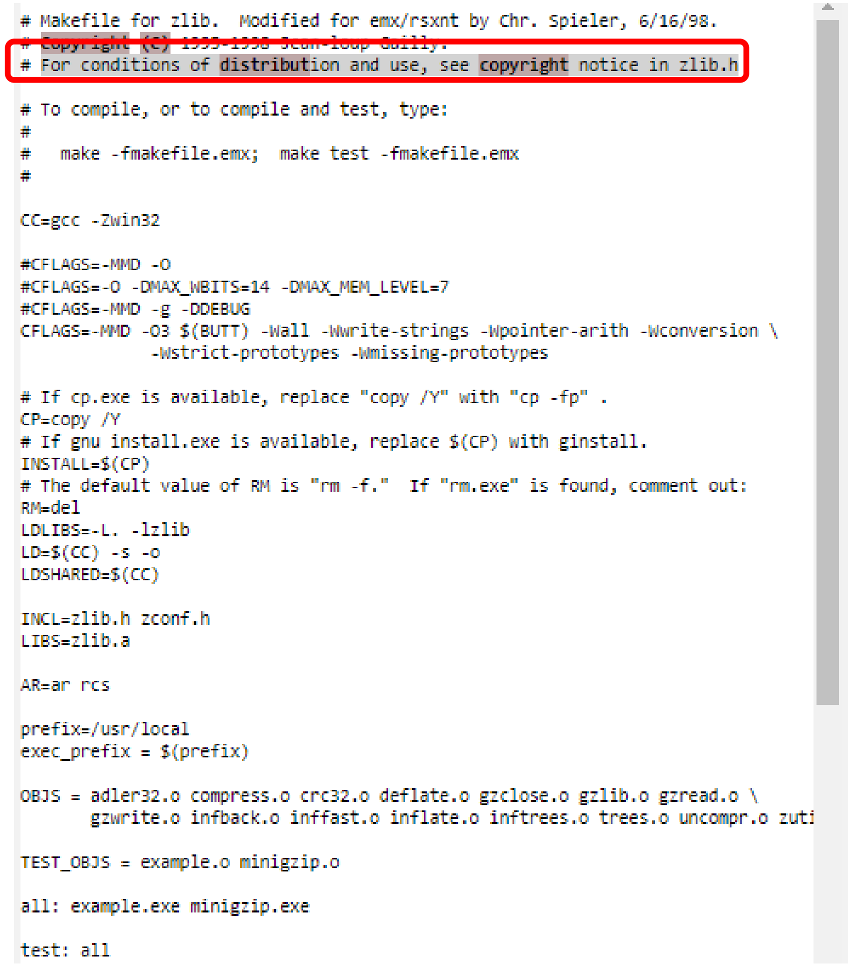
- FOSSology allows the user to define text phases …
  - **For confirmation …**
  - **… or correction**
- And assign license confirmations or corrections to it
  - Every time a file is found with this text phrase (100%) …
  - … the confirmation or correction is applied to it.
- No need to go through every file again.
- Does not apply with scan result conflicts

fossology Open Source Software

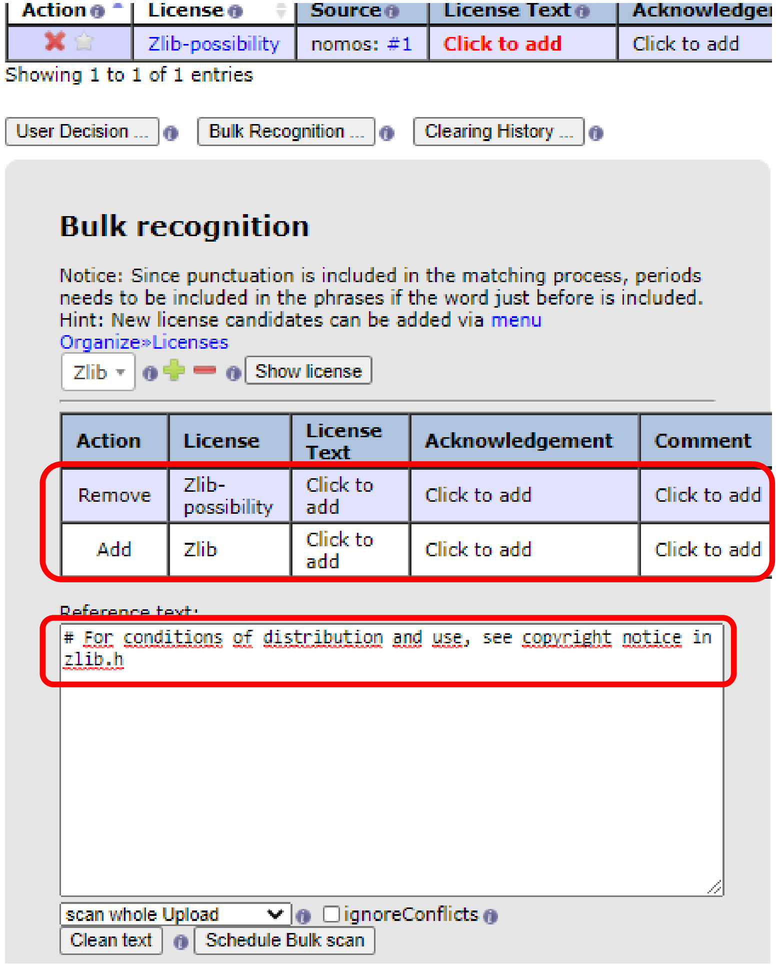
Feature: Reviewing Many Files at Once: Bulk Scan

User Interface

- Copy a characteristic text phrase from file view left (1)
- Paste into the bulk scan text field (2)
  - The application will search for file with this text
- Define scanner license findings (3)
  - For correction / removal
  - For confirmation, creating a clearing decision
- Bulk scan will run over all files of the package and apply clearing decisions or scanner corrections where matching (4)



```
# Makefile for zlib.  Modified for emx/rsxnt by Chr. Spieler, 6/16/98.
# Copyright (C) 1995-1998 Jean-loup Gailly.
# For conditions of distribution and use, see copyright notice in zlib.h

# To compile, or to compile and test, type:
#
#   make -fmakefile.emx;  make test -fmakefile.emx
#

CC=gcc -Zwin32

#CFLAGS=-MMD -O
#CFLAGS=-O -DMAX_WBITS=14 -DMAX_MEM_LEVEL=7
#CFLAGS=-MMD -g -DDEBUG
CFLAGS=-MMD -O3 $(BUTT) -Wall -Wwrite-strings -Wpointer-arith -Wconversion \
             -Wstrict-prototypes -Wmissing-prototypes

# If cp.exe is available, replace "copy /Y" with "cp -fp" .
CP=copy /Y
# If gnu install.exe is available, replace $(CP) with ginstall.
INSTALL=$(CP)
# The default value of RM is "rm -f."  If "rm.exe" is found, comment out:
RM=del
LDLIBS=-L. -lzlib
LD=$(CC) -s -o
LDSHARED=$(CC)

INCL=zlib.h zconf.h
LIBS=zlib.a

AR=ar rcs

prefix=/usr/local
exec_prefix = $(prefix)

OBJS = adler32.o compress.o crc32.o deflate.o gzclose.o gzlib.o gzread.o \
       gzwrite.o infback.o inffast.o inflate.o inftrees.o trees.o uncompr.o zuti

TEST_OBJS = example.o minigzip.o

all: example.exe minigzip.exe

test: all
```



Action	License	Source	License Text	Acknowledge
✖ ☆	Zlib-possibility	nomos: #1	Click to add	Click to add

Showing 1 to 1 of 1 entries

Bulk recognition

Notice: Since punctuation is included in the matching process, periods needs to be included in the phrases if the word just before is included.
Hint: New license candidates can be added via menu Organize»Licenses

Action	License	License Text	Acknowledgement	Comment
Remove	Zlib-possibility	Click to add	Click to add	Click to add
Add	Zlib	Click to add	Click to add	Click to add

Reference text:

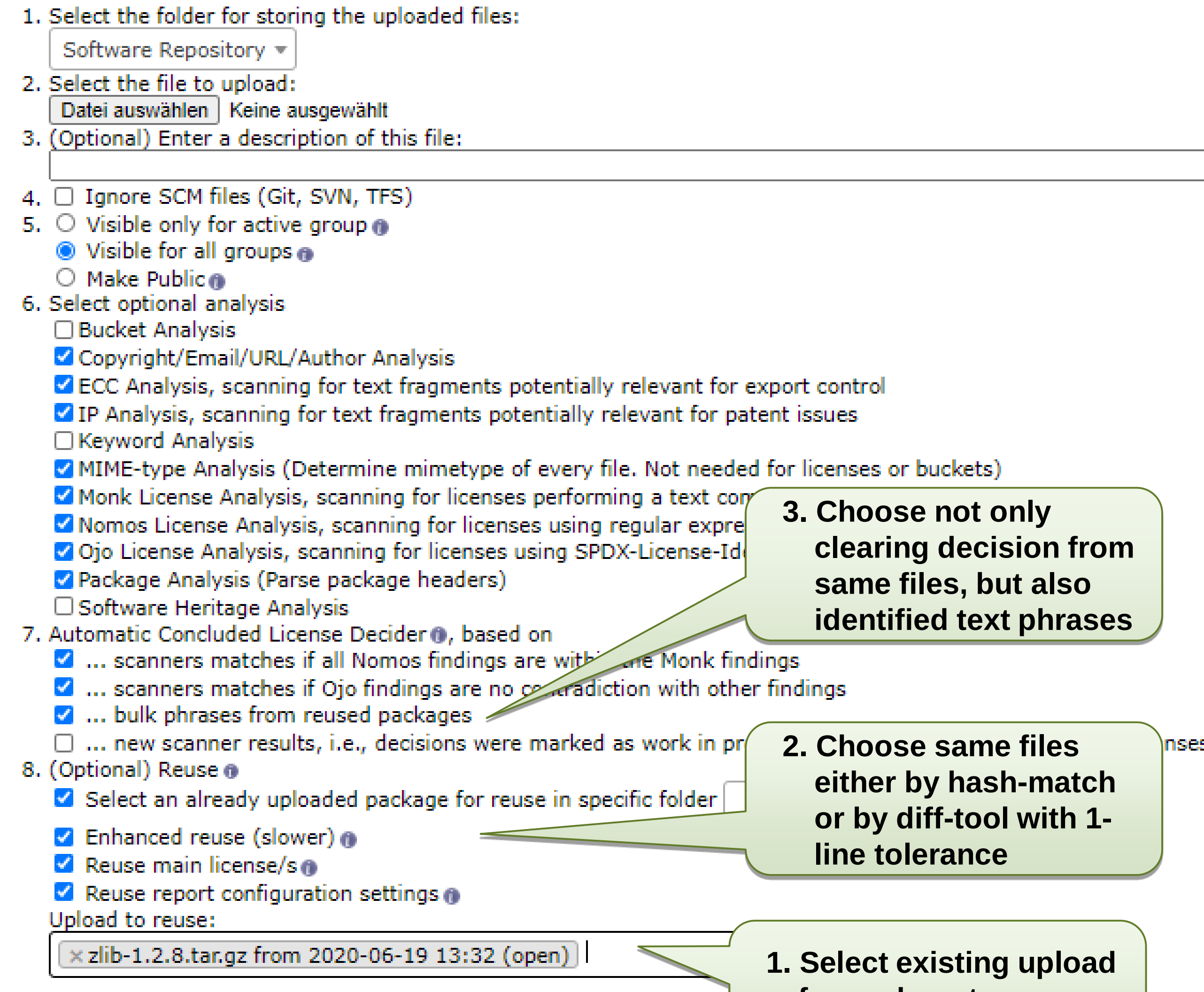
```
# For conditions of distribution and use, see copyright notice in zlib.h
```

scan whole Upload □ ignoreConflicts

Feature: Using Reuse of License Corrections

User Interface

- At upload you can select another existing package on server for reusing license review data
- Three main reuse options given:
  - Reuse license review data at same hash value calculated for files (1)
  - Reuse license review data at one-line tolerance using the diff tool (2)
  - Reuse bulk scan tasks entered for selected existing package also for new package (3)



1. Select the folder for storing the uploaded files:
   Software Repository ▾
2. Select the file to upload:
   Datei auswählen Keine ausgewählt
3. (Optional) Enter a description of this file:
4. □ Ignore SCM files (Git, SVN, TFS)
5. ○ Visible only for active group
   ● Visible for all groups
   ○ Make Public
6. Select optional analysis
   □ Bucket Analysis
   ☑ Copyright/Email/URL/Author Analysis
   ☑ ECC Analysis, scanning for text fragments potentially relevant for export control
   ☑ IP Analysis, scanning for text fragments potentially relevant for patent issues
   □ Keyword Analysis
   ☑ MIME-type Analysis (Determine mimetype of every file. Not needed for licenses or buckets)
   ☑ Monk License Analysis, scanning for licenses performing a text com
   ☑ Nomos License Analysis, scanning for licenses using regular expre
   ☑ Ojo License Analysis, scanning for licenses using SPDX-License-Id
   ☑ Package Analysis (Parse package headers)
   □ Software Heritage Analysis
7. Automatic Concluded License Decider, based on
   ☑ ... scanners matches if all Nomos findings are within the Monk findings
   ☑ ... scanners matches if Ojo findings are no contradiction with other findings
   ☑ ... bulk phrases from reused packages
   □ ... new scanner results, i.e., decisions were marked as work in pr nse
8. (Optional) Reuse
   ☑ Select an already uploaded package for reuse in specific folder
   ☑ Enhanced reuse (slower)
   ☑ Reuse main license/s
   ☑ Reuse report configuration settings
   Upload to reuse:
   × zlib-1.2.8.tar.gz from 2020-06-19 13:32 (open)

3. Choose not only clearing decision from same files, but also identified text phrases

2. Choose same files either by hash-match or by diff-tool with 1-line tolerance

1. Select existing upload from where to reuse license decisions

Overview: Contents

1. **Introduction and overview**
   What is FOSSology
2. **Introduction and Basic Work Flow**
   What FOSSology is designed for
3. **Features to make you efficient**
   How to work efficiently
4. **Show time**
   Workflow demo
5. **Take aways**

Overview: Contents

1. **Introduction and overview**
   What is FOSSology
2. **Introduction and Basic Work Flow**
   What FOSSology is designed for
3. **Features to make you efficient**
   How to work efficiently
4. **Show time**
   Workflow demo
5. **Take aways**

Take aways

- License analysis cannot be fully automated – expert knowledge is required
- Think about how to organize your folder structure
- Use edit feature on folder level
- Use bulk scans very extensively
- Consider also to reuse bulk phrases of different uploads
- It is Open Source – no vendor locking and you can adapt it to your needs
- Do not forget to see the fun side

Thank you for your attention

CC-BY-SA 4.0
https://creativecommons.org/licenses/by-sa/4.0/

Internet
https://www.fossology.org

Github
https://github.com/fossology/fossology

fossology
Open Source License Compliance by Open Source Software