



Strigi

Searching files in KDE4

Jos van den Oever



What is searching about?



So, list your "Crazy Ideas" here: (<http://strigi.sf.net>)

- give me all e-mails that have more than one xsl attachment and are overall bigger than 1 MB
- give me all e-mails from user xyz about music but without any music file attached
- show me all music files rated better than 90 % (in amarok) which I played in March
- show me all kopete/IM talks with contact xyz with links to kde.org
- give me all documents related to a scientific reference e.g. "A. Manz, J. C. T. Eijkel, Pure Appl. Chem. 2001, 73, 1555-1561"
- display all files larger than 5kB that I have downloaded in march
- give me all documents related to a specific chemical compound AND a specific author
- find all my social-bookmarked pages on strigi (eg. on del.icio.us or connotea.org)
- give me from all music-related rss-feed posts those containing artist names which are also in my amarok collection
- show me all konqueror-visited locations (local/remote/http/whatever) Dever



A search interface should

- show the user files or parts of files that match the query,
- match the current context
- and open entries from the search result in the right program

Strigi

Nepomuk



Java has nice streaming base class

```
public StreamDemo(URL url) throws IOException {  
    InputStream filestream = url.openStream();  
    ZipInputStream zipstream = new ZipInputStream(filestream);  
    ZipEntry entry = zipstream.getNextEntry();  
    while (entry != null) {  
        handleEntry(zipstream, entry);  
    }  
}
```

1



```

class StreamBase<T> {
    ...
public:
    ...
    virtual int32_t
        read(const T*& start,
             int32_t min,
             int32_t max) = 0;
    virtual int64_t
        reset(int64_t pos) = 0;
    ...
};

```

```

void
readdemo() {
    int32_t nread;
    const char* data;
    nread = jstream->read(data, 1, 0); // read at least 1 byte
    jstream->reset(0); // reset to start of stream
    nread = jstream->read(data, 3, 3); // read exactly 3 bytes
}

```

Simple abstract class

- templated class
- one read function
- passes a pointer to an internal buffer
- only two functions need to be implemented



```
class BufferedStream<T> {  
    ...  
public:  
    ...  
    virtual int32_t  
        fillBuffer(T* start,  
                  int32_t space) = 0;  
    ...  
};
```

Stream with a buffer

- most common use case
- implement one simple function
- called when the buffer is empty

Examples

- FileInputStream
- BZ2InputStream
- GZipInputStream
- InputStreamReader
- ProcessInputStream



```
class SubInputStream<T> {  
    ...  
public:  
    SubInputStream(  
        StreamBase<char>* input,  
        int32_t size);  
    ...  
};
```

SubInputstream

- a size limited version of another stream

```
class SubInputStream<T> {  
    ...  
public:  
    SubInputStream(  
        StreamBase<char>* input,  
        int32_t size);  
    ...  
};
```

StringTerminatedSubStream

- a size limited version of another stream



```
class SubStreamProvider {  
    ...  
public:  
    SubStreamProvider(  
        StreamBase<char>*  
input);  
    virtual StreamBase<char>*  
        nextEntry() = 0;  
    const EntryInfo&  
        getEntryInfo() const;
```

Examples

- TarInputStream,
ZipInputStream
- ArInputStream,
RpmInputStream
- MailInputStream

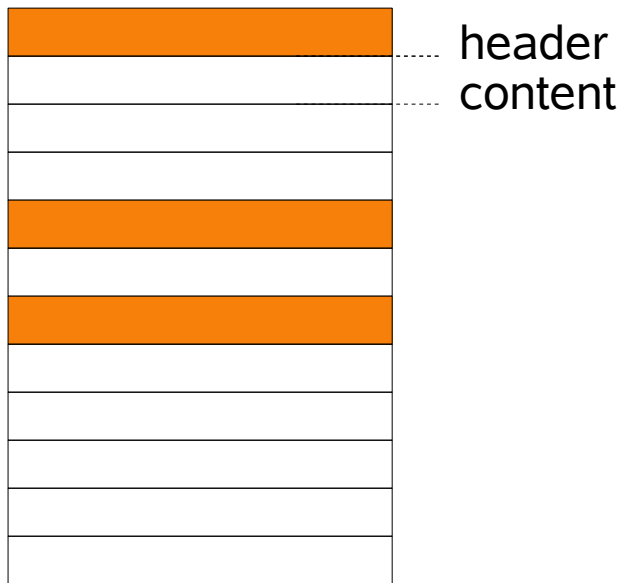
Split a stream up

- access parts of files
- implement one simple
function
- called to get streams one
after the other



Tar file format

512 byte blocks



Simple SubStreamProvider

- fixed size blocks
- no additional buffer required
- parse the header into the `EntryInfo` object
- Position the stream at the start of the content and create a `SubInputStream` with the given size of the stream



A simple JStreams program

```
sub listDir(path):  
    dir = open(path)  
    for entrypath in dir:  
        print entrypath  
        if isDir(entrypath):  
            listDir(entrypath)  
        else:  
            ssp = openStreamProvider(  
                entrypath)  
            if ssp.isOk():  
                listStream(ssp)
```

- normal `find` without arguments just list files and directies
- `deepfind` also lists all files contained in other files



```
sub listStream(ssp, path):
    stream = ssp.nextEntry()
    while stream:
        entrypath = path + '/' +
            ssp.getEntryInfo().filename
        print entrypath
        ssp = openStreamProvider(
            entrypath)
        if ssp.isOk():
            listStream(ssp)
```

A simple JStreams program

- normal `find` without arguments just list files and directories
- `deepfind` also lists all files contained in other files

Improving 'grep -r' is left as an exercise for the audience




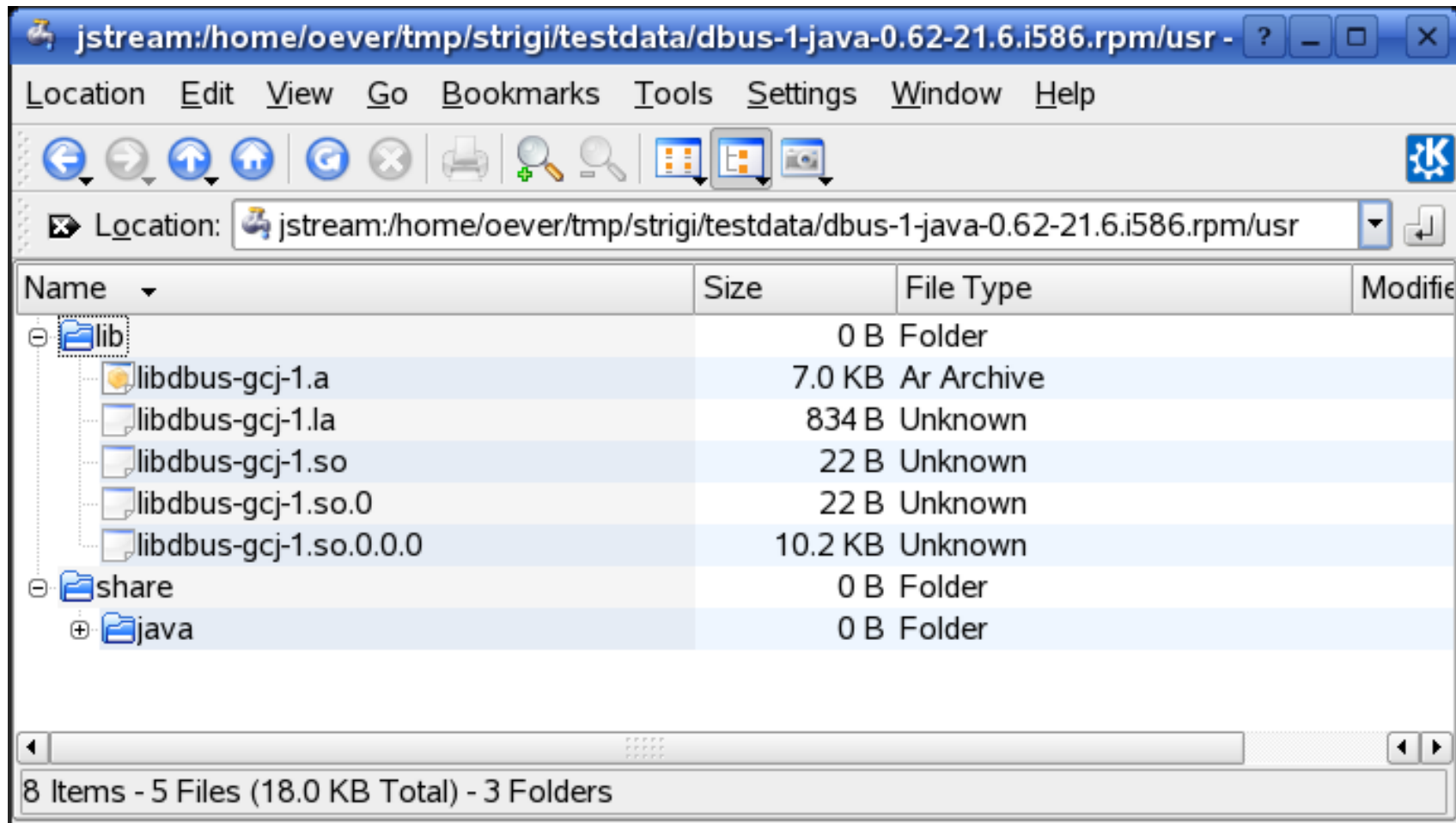
Browsing nested files



qclient

Name	Size	Type	Modified
doc	96 bytes	Directory	2006-06-22 00:07:33
src	680 bytes	Directory	2006-09-21 23:01:24
testdata	472 bytes	Directory	2006-09-13 21:28:28
a.bz2	214 bytes	bz2 File	2006-06-22 00:07:34
a.gz	158 bytes	gz File	2006-08-22 01:12:58
a.tar	10 KB	tar File	2006-06-22 00:07:34
a.zip	275 bytes	zip File	2006-06-22 00:07:34
all.zip	83 KB	zip File	2006-08-25 23:55:23
chinese_ucs2.txt	1 KB	txt File	2006-06-22 00:07:34
chinese_utf8.txt	1 KB	txt File	2006-06-22 00:07:34
dbus-1-java-0.62-21.6.i586.rpm	15 KB	rpm File	2006-07-10 09:50:23
home.de.html	15 KB	html File	2006-06-22 00:07:34
libpdfstream.a	138 KB	a File	2006-08-25 20:35:31
pdfinputstream.o	52 KB	Unknown	2006-08-25 20:35:31
pdfparser.o	82 KB	Unknown	2006-08-23 00:24:40
mail	21 KB	File	2006-06-22 00:07:34
p.zip	82 KB	zip File	2006-08-25 21:32:16
data2	0 bytes	Directory	1970-01-01 01:00:00
kdesvn	0 bytes	Directory	1970-01-01 01:00:00
qt4	0 bytes	Directory	1970-01-01 01:00:00
qt-copy	0 bytes	Directory	1970-01-01 01:00:00
demos	0 bytes	Directory	1970-01-01 01:00:00
affine	0 bytes	Directory	1970-01-01 01:00:00
arthurplugin	0 bytes	Directory	1970-01-01 01:00:00
bq1.jpg	23 KB	jpg File	2006-08-23 04:18:56







```
cd /usr/bin  
rm find  
ln -s deepfind find
```

find + jstreams = deepfind
deepfind + locate = deeplocate

'grep -r' + jstreams = deepgrep
deepgrep + X = deepX



What is X?



Kat



Beagle



Four problems when finding X

- 1 Beagle is designed to index files, not streams
- 2 Kat is more or less dead
- 3 JStreams indexes more than one file at once
- 4 Kat and Beagle are not just indexing text



Solution:

create X and call it Strigi



4) Extracting more than text



One can extract more than just text from files

- subject, author, modification time, sha1, title, links, etc
- multiple analyzers can add to the object at once

Strigi extracts from each file an “Indexable object”

- file path (URL)
- mtime
- size
- mimetype
- key, value metadata

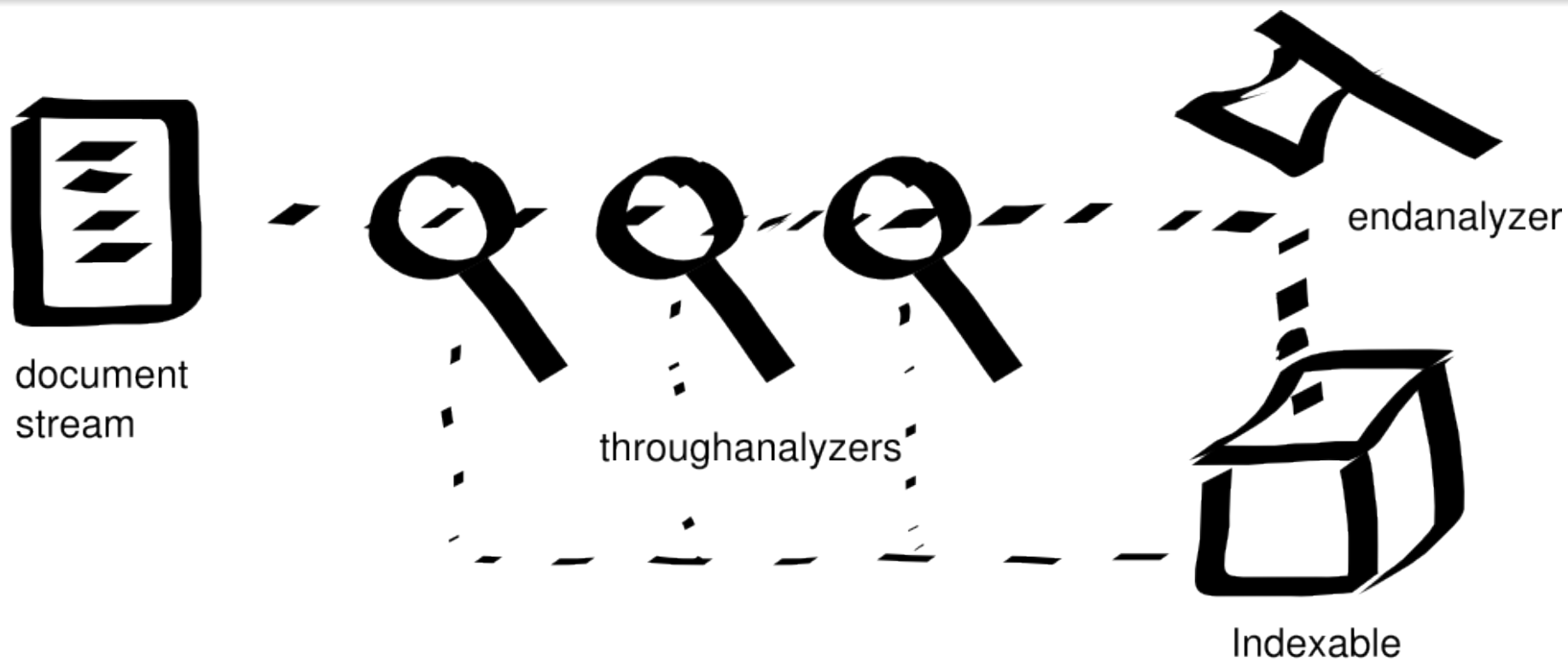


EndStreamAnalyzer

- there can be only one per stream
- reads the stream by pulling (calling `read()`) on the stream

ThroughStreamAnalyzer

- there can be many
- reads the stream by passing along `read()` calls and looking at the passing bytes



- loadable modules for both types
- work on windows and linux



TextEndAnalyzer

- splits the text up and passes it to the Indexable

MimeTypeThroughAnalyzer

- uses libmagic to determine the mimetype and encoding

KFileThroughAnalyzer

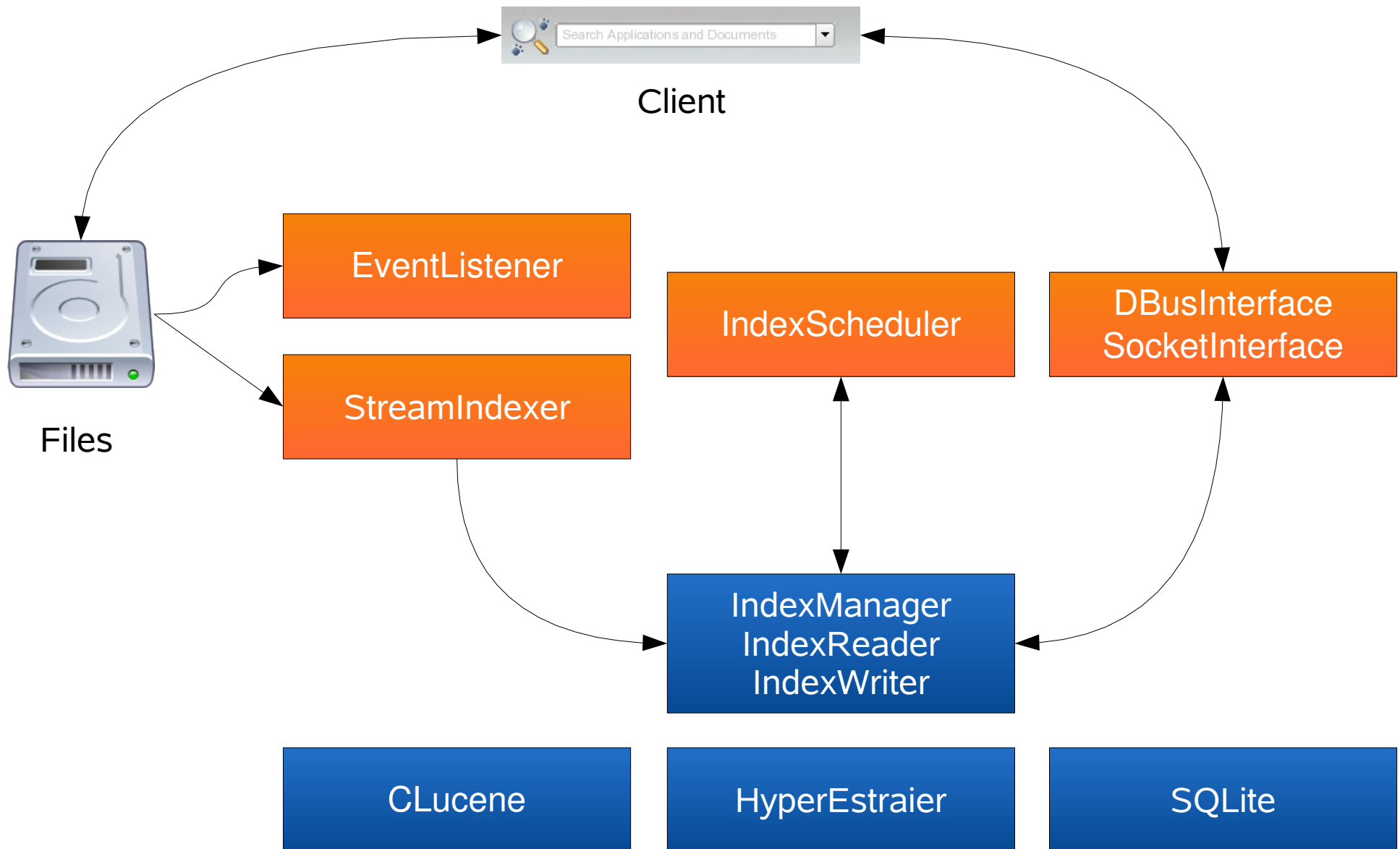
- uses KFileMetaData to get metadata

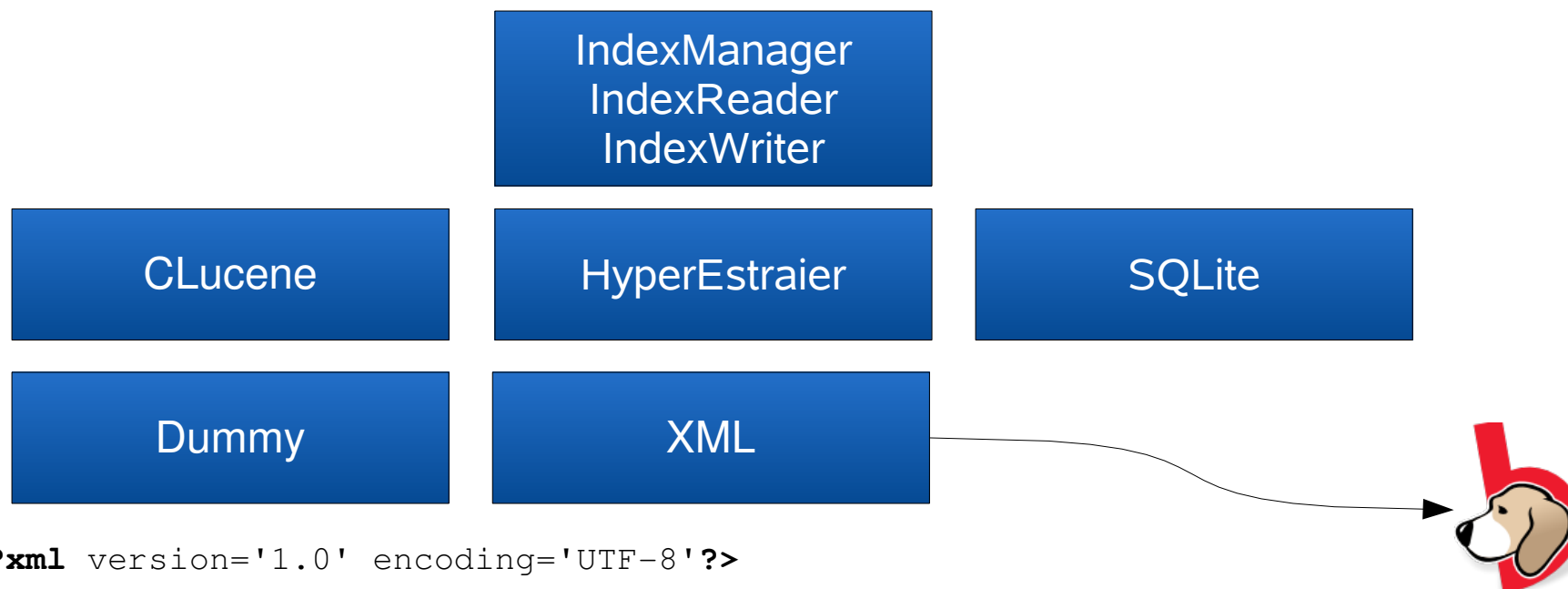
SaxEndAnalyzer

- extracts the text content from xml files

MailEndAnalyzer

- analyzes the mail headers and indexes the attachments





```

<?xml version='1.0' encoding='UTF-8'?>
<metadata>
  <file uri='testdata/.svn/text-base/all.zip.svn-base/a.zip' mtime='1150927654'>
    <value name='mimetype'>application/x-zip</value>
    <value name='sha1'>25da41e3282f81b8289ed63da8a534c15d9fee9b</value>
    <value name='size'>275</value>
  </file>
  <file uri='testdata/.svn/text-base/all.zip.svn-base/p.zip/data2/kdesvn/qt4/qt-copy/demos/affine/bg1.jpg' mtime='1156299536'>
    <value name='mimetype'>image/jpeg</value>
    <value name='sha1'>a25141506f894bd6e963283d758d7ff21aeee516</value>
    <value name='size'>23771</value>
  </file>

```

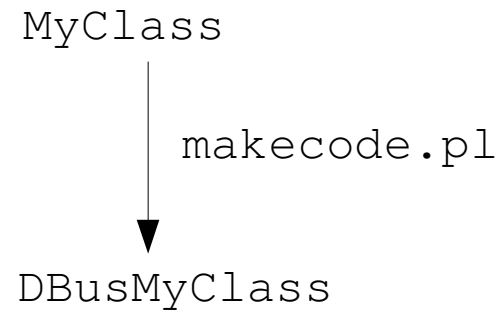


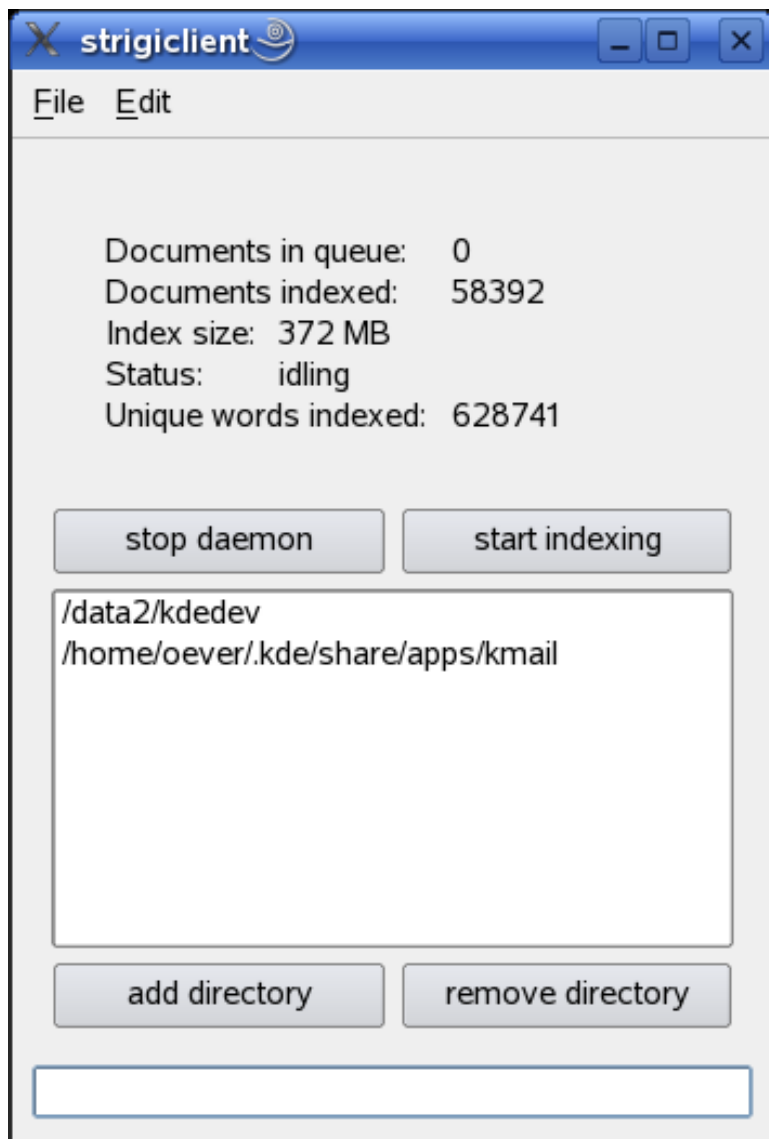

DBus support using the C API

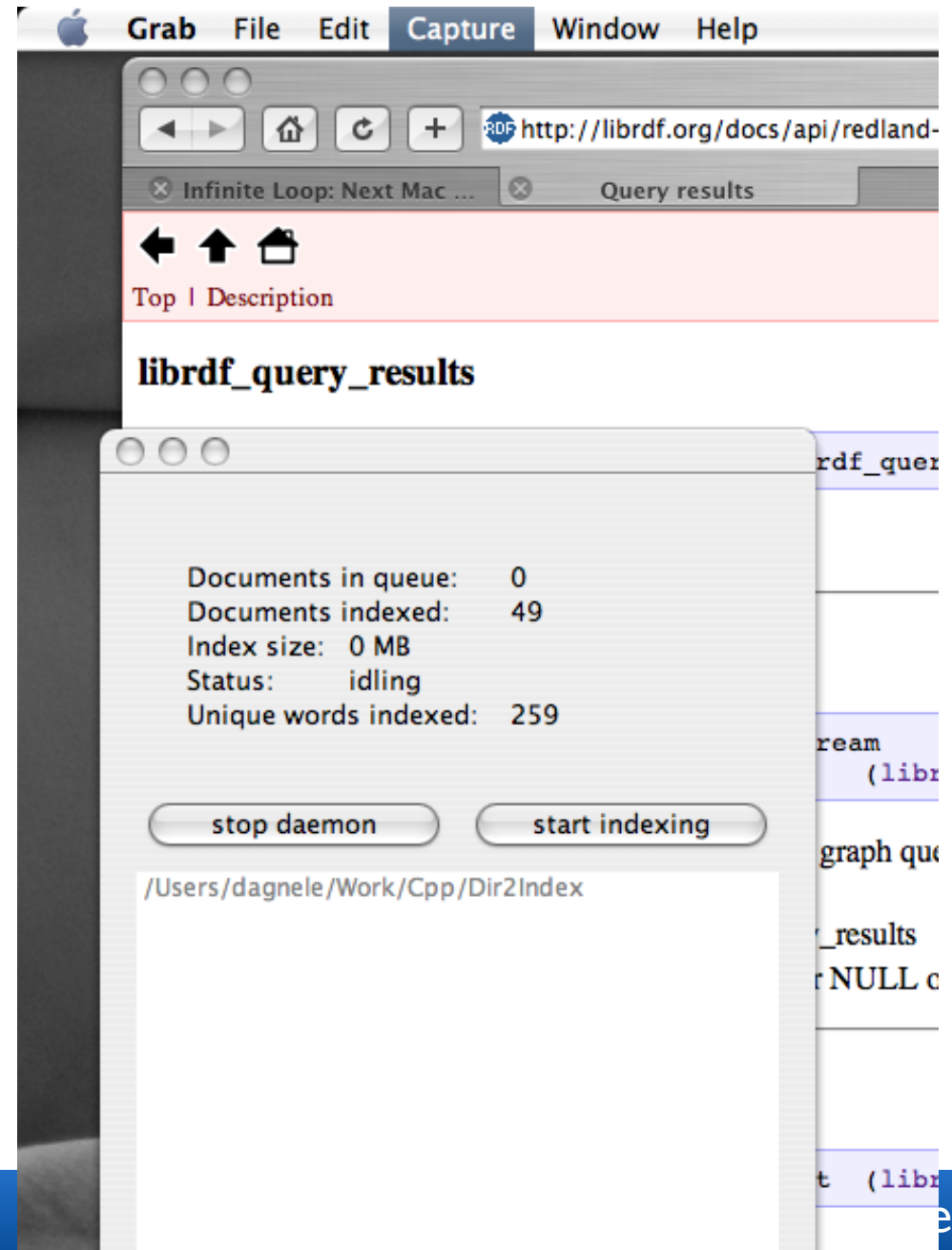
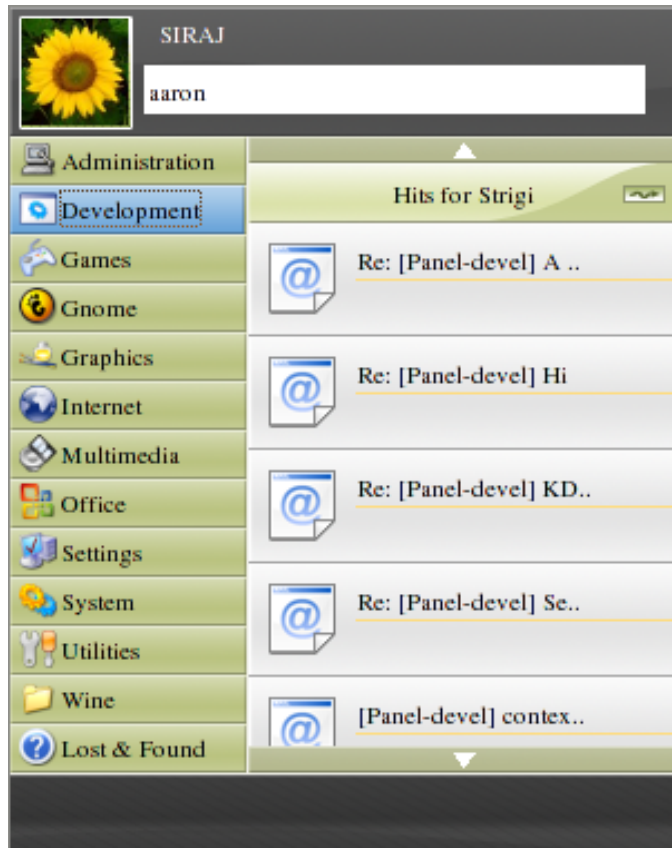
- ideal for a daemon
- only 1 dependency
- high performance
- not easy to get it running

Simple C++ Dbus code generator

- generate Service class from a C++ header file
- support for introspection









strigiapplet



#strigi - Konversation

File Edit Insert Bookmarks Settings

Qt 4.2: How to Report a Bug Home
AllClasses MainClasses

bugft.desktop [Desktop Entry]
Encodina=UTF-8 Kevs=buaft

bugno.desktop [Desktop Entry]
Encodina=UTF-8

bugft.desktop [Desktop Entry]
Encodina=UTF-8 Kevs=buaft

bugno.desktop [Desktop Entry]
Encodina=UTF-8

bug.png

bug.png

bug.png

bug.png

bug.png

bug

11:38 London

vandenoever

#freenode #kat #kde-devel #kde4-c

Ready.

chann
t sets
t sets
et, no
ated c
n 'topi
annel
annel
this c
s chan
rigi ha
protect
n 'topi
rigi ha
protect
n 'topi
d this



#strigi - Konversation

File Edit Insert Bookmarks Settings

Qt 4.2: How to Report a Bug Home
AllClasses MainClasses

- bugft.desktop [Desktop Entry]
Encodina=UTF-8 Kevs=buaft
- bugno.desktop [Desktop Entry]
Encodina=UTF-8
- bugft.desktop [Desktop Entry]
Encodina=UTF-8 Kevs=buaft
- bugno.desktop [Desktop Entry]
Encodina=UTF-8
- bug.png
- bug.png
- bug.png
- bug.png
- bug.png
- bug.png

bug

11:38 vandenoever

London

FreeNode #kat #kde-devel #kde4

Ready.





Strigi Desktop Search - Konqueror

Location Edit View Go Bookmarks Tools Settings Window Help

Location: strigi/?q=filename:bug*&t=Images

search status preferences help about

filename:bug* search Images (8) Text (5) Web (1)

- 
bug.png
 /data2/kdedev/install/share/apps/khtml/icons/crystalsvg/128x128/actions/bug.png
 - 11k - PNG Image
- 
bug.png
 /data2/kdedev/install/share/apps/khtml/icons/crystalsvg/16x16/actions/bug.png -
 992 bytes - PNG Image
- 
bug.png
 /data2/kdedev/install/share/apps/khtml/icons/crystalsvg/22x22/actions/bug.png -
 2k - PNG Image
- 
bug.png
 /data2/kdedev/install/share/apps/khtml/icons/crystalsvg/32x32/actions/bug.png -
 2k - PNG Image

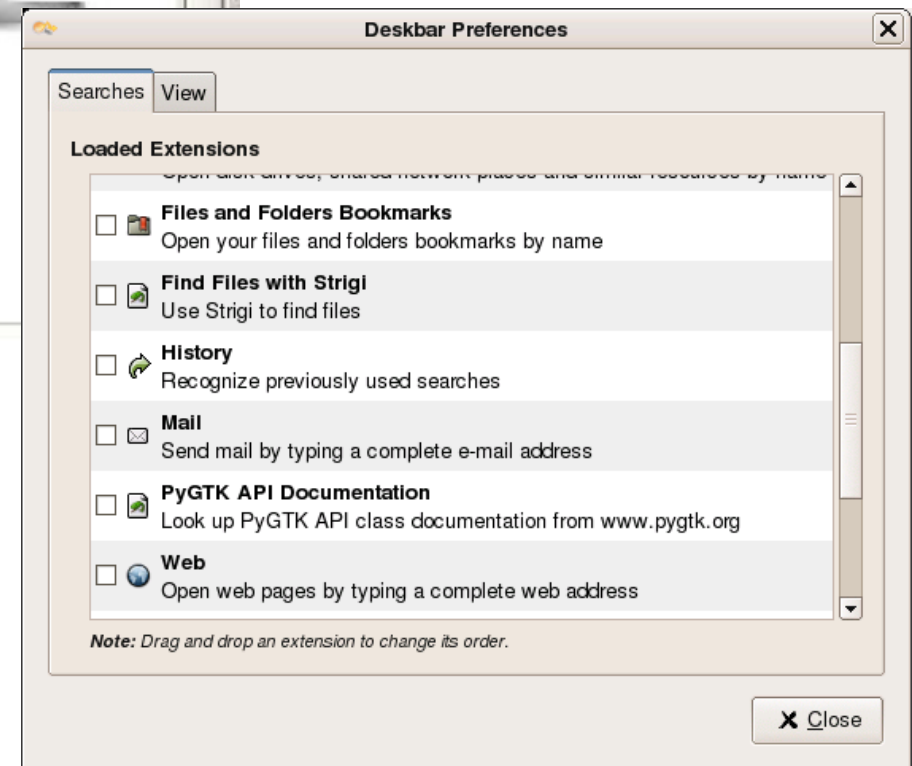


GNOME DeskBar



Added a Strigi adaptor

- written in Python
- communicates over DBus





Can we rely on Strigi staying there?

- code core is small
- most of the code is the implementation of various interfaces
- unit tests for implementations of jstreams are easy
- xmlstreamwriter can be used by other indexers



Ben van Klinken (CLucene developer)

- ported plugin architecture to Windows
- JStreams testing and discussions
- advises Strigi as the indexer of choice for CLucene

Flavio Castelli

- Inotify support
- Selective filtering by indexing on filename
- Logging framework

Egon Willighagen

- KFileThroughAnalyzer

Fathi Boudra

- .deb packaging



- KMail filter bar filters on entire mails and colours the mail by search score
- Calender entries can be found in the calender file and these entries can be opened directly in Kontact
- Search results are displayed on a timeline or on a sizerline
- File dialog filters the directories based on whether the desired mimetype is somewhere in the hierarchy
- Entry of keywords in the file dialog does a search instead of an error message
- The konqueror context menu gets a menu item for finding duplicate files
- Email on an imap server are indexed and can be opened



Integration

- Implement a JStream that can split up your multipart files
- Write a stream analyzer that extracts the data you want to have indexed
- Teach you app how to handle the URL that Strigi gives you (usually `jstream:/` will take care of this)

Resources

- `#strigi`
- <http://strigi.sf.net>
- `trunk/playground/base/strigi`
- `trunk/playground/base/strigiapplet`



A search interface should

- show the user files or parts of files that match the query,
- match the current context
- and open entries from the search result in the right program

Strigi

Nepomuk



- integration into KDE4 (svn, dependencies, releases)
- enable multiple repositories
- enable indexing of remote files like http and imap
- think about metadata standards
- come up with more search ideas
- generalize the jstream:/ kioslave
- write (yet another) backend
- porting analyzers from other indexers (mp3, jpg, ogg, kfilemetadata)