

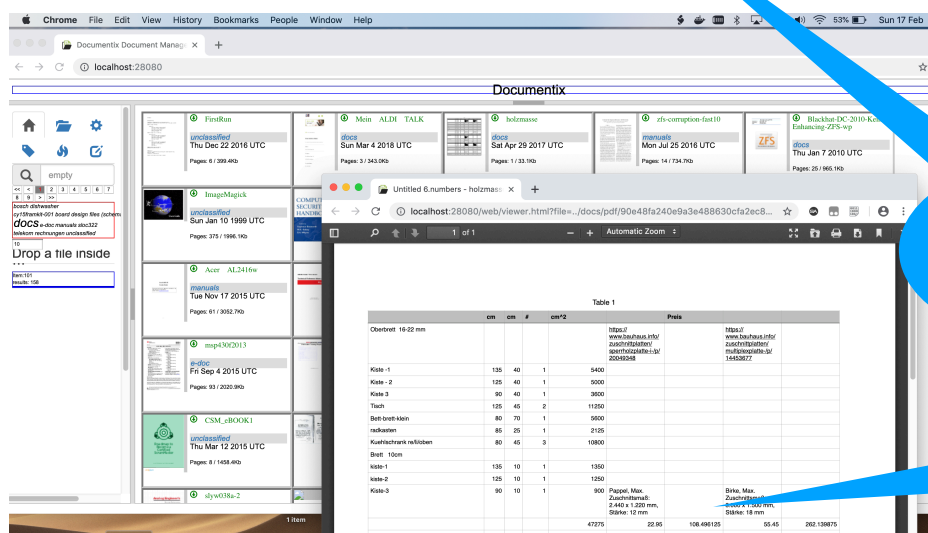
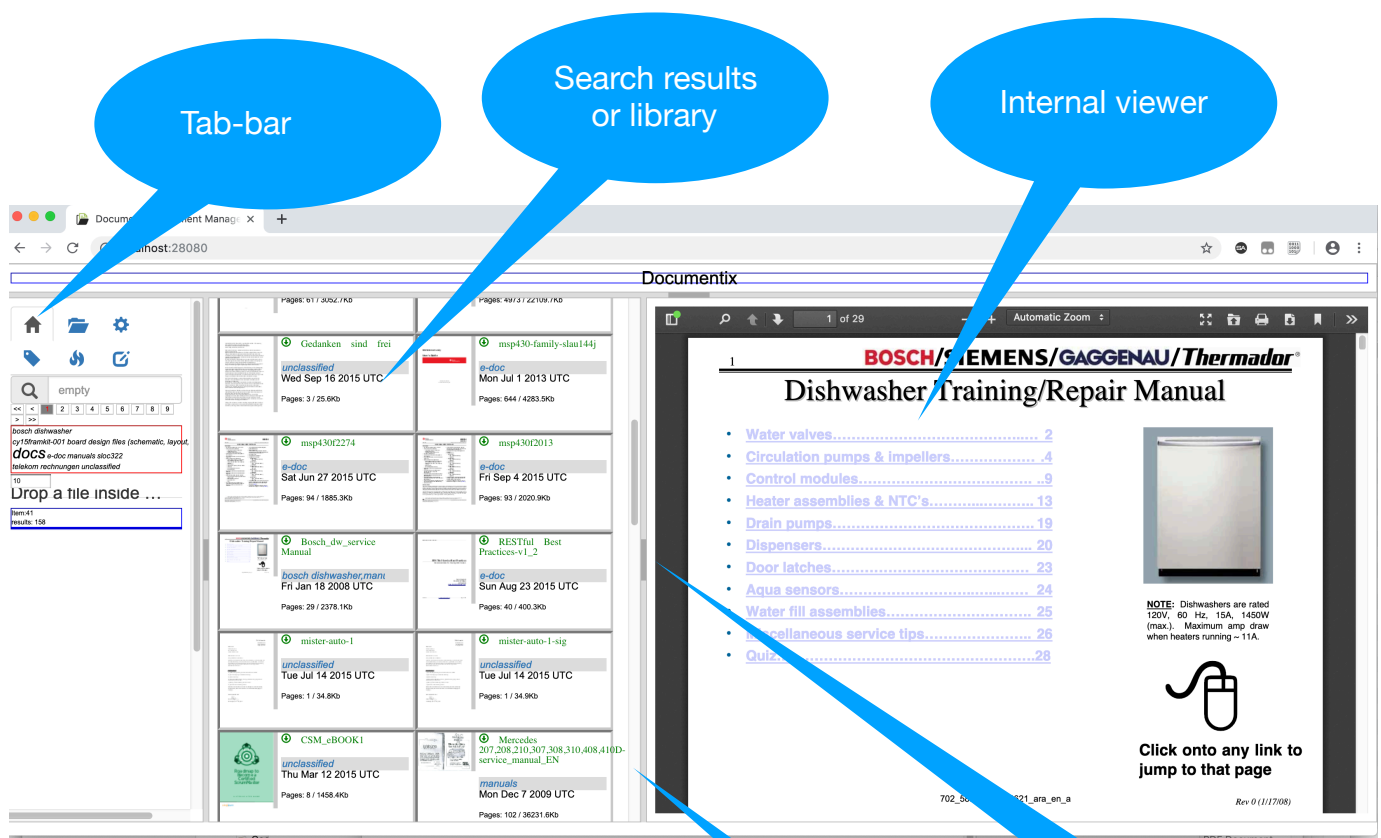
Documentix

the Document manager

Overview

Documentix quickly searches your collection using a full-text search engine
Documentix lets you classify documents
Documentix automatically classify new documents based on previous training
Documentix automatically OCR's new documents to enable searching
Documentix manages multiple types of documents (pdf/word/ppt/jpeg..)
Documentix presents all archived documents through a web-interface
Documentix allows easy sharing of documents

User interface

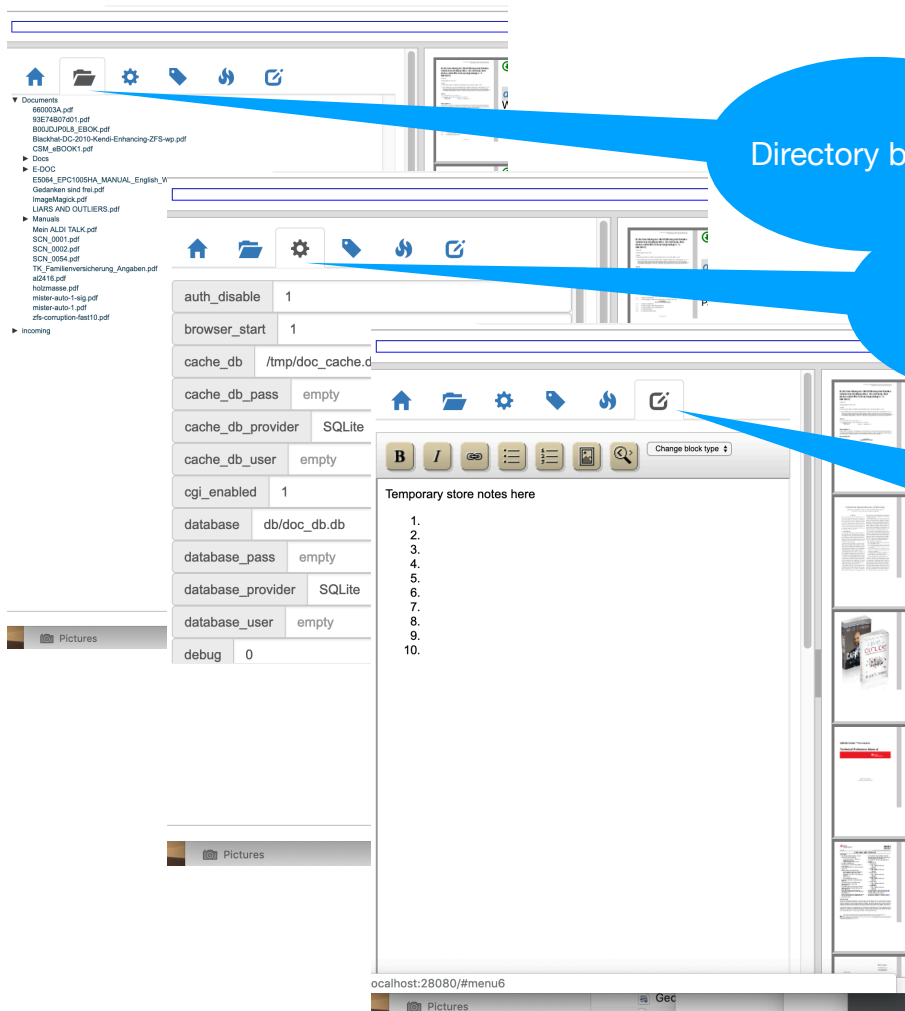
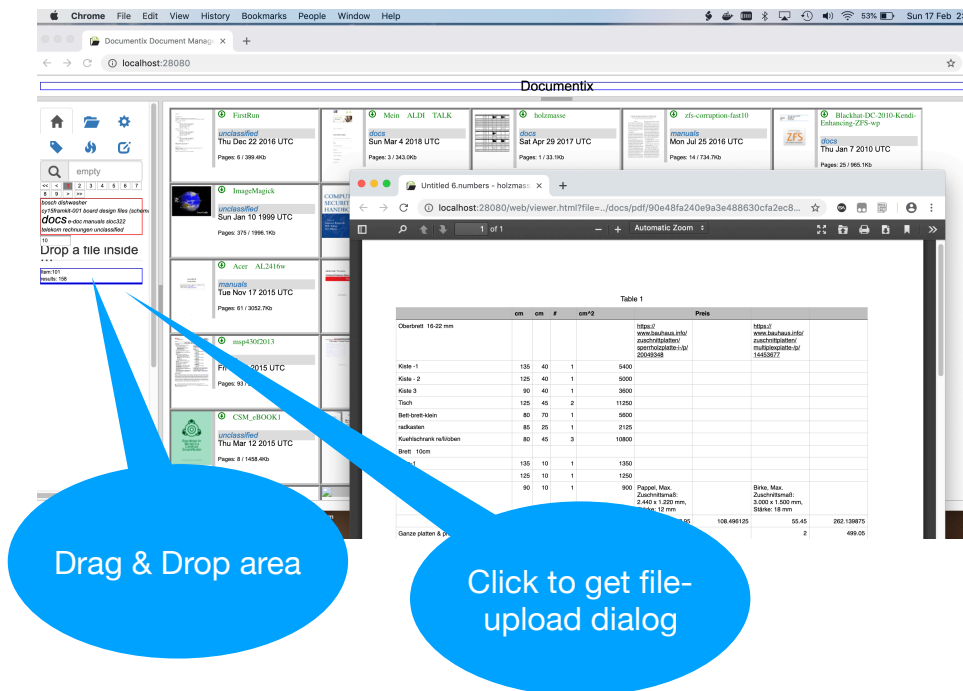


Search function

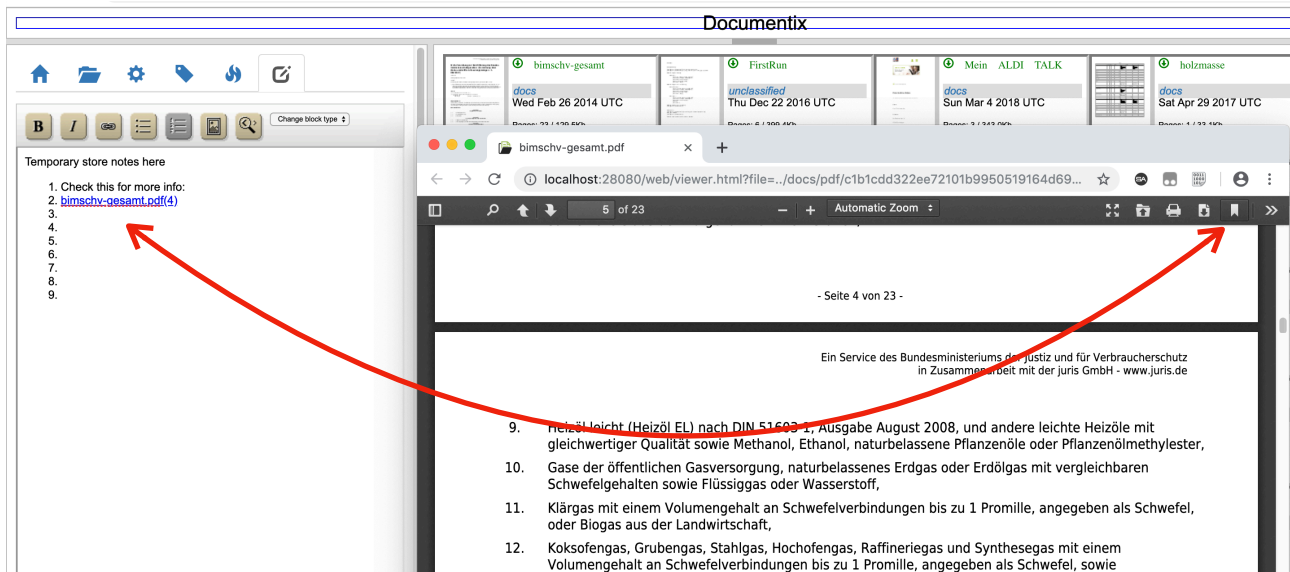
The image shows a screenshot of a document management system interface. On the left, a sidebar contains navigation icons (home, folder, settings, tag, flame, edit) and a search bar. The search bar contains the text "linux". Below the search bar, a red box highlights the text "docs e-doc manuals unclassified". A blue callout bubble points to the search bar with the label "Search string".

On the right, a document preview is shown. The document title is "Lagebericht2014". Below the title, the text "docs" is highlighted. A blue callout bubble points to a green download icon with the label "Download button". Another blue callout bubble points to the text "docs" with the label "Document tags". Below the document title, the text "Thu Dec 11 2014 UTC" and "Pages: 44 / 1617.0Kb" are visible. A blue callout bubble points to a snippet of text from the document preview with the label "Preview of found text". The snippet includes the text "...32", "3.3.4", and "Operation Windigo - **Linux**-Schadpr".

Upload new documents



Drag & Drop Bookmark-icon to capture document & location in scratch-pad



Internals

Documentix uses SQLite and the full-text-search engine as a database backend. All documents are stored using the MD5 hash as an index. (Don't tell me it's broken - it does not matter here)

Documents send to the processing engine are converted to PDF files (if not already). If the PDF does not contain more than a certain number of words, tesseract as an OCR engine is started on the document and a searchable PDF is created.

An interface to a Canon-mx870 exists but is not supported in the docker image.

It should be easy to create a curl command-line to upload new scans like:
> curl -F file=@o-page-14.pdf -H x-file-name:MyPage.pdf <http://localhost:28080/upload>

Which will upload the file "o-page-14.pdf" and name it on the server "MyPage.pdf"

Retrieving files is also possible:

If you know the MD5 has of a file.

Lets assume the MD5="009914b6b6a4cb4a37f2fc5bdbc561ca" is.
And the server is reachable under: <http://localhost:28080>

Then:

<http://localhost:28080/docs/raw/009914b6b6a4cb4a37f2fc5bdbc561ca/MyPage>
Will return the original document as it was uploaded

<http://localhost:28080/docs/pdf/009914b6b6a4cb4a37f2fc5bdbc561ca/MyPage>
Will get the file as a PDF (converted if necessary including OCR)

<http://localhost:28080/web/viewer.html?file=../docs/pdf/009914b6b6a4cb4a37f2fc5bdbc561ca/MyPage>
Will show the file as a PDF in the html viewer.

<http://localhost:28080/docs/ico/009914b6b6a4cb4a37f2fc5bdbc561ca/MyPage>
Will return the original document as it was uploaded