Recoll
Benutzerhandbuch

KOLLABORATOREN			
	TITEL : Recoll Benutzerhandbuch		
AKTION	*NAME*	*DATUM*	*UNTERSCHRIFT*
VERFASSUNG	Jean-Francois Dockes	Dezember 4,2021	

REVISIONSGESCHICHTE			
NUMMER	DATUM	BESCHREIBUNG	NAME

Inhalt

Liste der Tabellen

Abstrakt

Dieses Dokument führt in die Begriffe der Volltextsuche ein und beschreibt die Installation und Verwendung der Anwendung Recoll. Diese Version beschreibt Recoll 1.29.

Kapitel 1

Einführung

Dieses Dokument führt in die Begriffe der Volltextsuche ein und beschreibt die Installation und Verwendung der Recoll-Anwendung. Es wird für Recoll aktualisiert 1.29.

Recoll war lange Zeit für Unix-ähnliche Systeme gedacht. Es wurde erst vor kurzem (2015) auf MS-Windows portiert. Viele Verweise in diesem Handbuch, insbesondere Dateispeicherorte, sind spezifisch für Unix und gelten nicht für Windows, wo einige beschriebene Funktionen auch nicht verfügbar sind. Das Handbuch wird nach und nach aktualisiert werden. Bis dahin können unter Windows die meisten Verweise auf freigegebene Dateien übersetzt werden, indem man im Recoll-Installationsverzeichnis nachsieht (typischerweise `C:/Programme (x86)/Recoll`, insbesondere alles, was in diesem Dokument in `/usr/share` referenziert wird, findet sich im Unterverzeichnis `Share`). Die Benutzerkonfiguration wird standardmäßig unter `AppData/Local/Recoll` im Benutzerverzeichnis gespeichert, zusammen mit dem Index selbst.

1.1 Ein Versuch ist es wert

Wenn Sie nicht gerne Handbücher lesen (wer tut das schon?), aber Recoll ausprobieren möchten, installieren Sie einfach die Anwendung und starten Sie die grafische Benutzeroberfläche (GUI) von **Recoll**, die Sie um Erlaubnis bittet, Ihr Home-Verzeichnis zu indizieren, so dass Sie sofort nach Abschluss der Indizierung suchen können.

Tun Sie dies nicht, wenn Ihr Heimatverzeichnis eine große Anzahl von Dokumenten enthält und Sie nicht warten wollen oder nur wenig Speicherplatz haben. In diesem Fall sollten Sie zunächst die Konfiguration anpassen, um den indizierten Bereich einzuschränken (Shortcut: Gehen Sie in der Recoll-GUI zu: Einstellungen → Indizierungskonfiguration, dann den Abschnitt Top-Verzeichnisse anpassen).

Auf Unix-ähnlichen Systemen müssen Sie möglicherweise die entsprechenden Hilfsprogramme für Dokumenttypen installieren, die sie benötigen (z. B. Antiword für Microsoft Word-Dateien). Das Paket Recoll für Windows ist in sich geschlossen und enthält die meisten nützlichen Hilfsprogramme.

1.2 Volltextsuche

Recoll ist eine Volltextsuchanwendung, d.h. sie findet Ihre Daten anhand des Inhalts und nicht anhand externer Attribute (wie dem Dateinamen). Sie geben Wörter (Begriffe) an, die in dem gesuchten Text vorkommen sollen oder nicht, und erhalten im Gegenzug eine Liste übereinstimmender Dokumente, die so geordnet sind, dass die *relevantesten* Dokumente zuerst erscheinen.

Sie müssen sich nicht daran erinnern, in welcher Datei oder E-Mail-Nachricht Sie eine bestimmte Information gespeichert haben. Sie fragen einfach nach verwandten Begriffen, und das Tool gibt eine Liste von Dokumenten zurück, in denen diese Begriffe vorkommen, ähnlich wie bei Internet-Suchmaschinen.

Volltextsuchanwendungen versuchen zu ermitteln, welche Dokumente für die von Ihnen eingegebenen Suchbegriffe am relevantesten sind. Computeralgorithmen zur Bestimmung der Relevanz können sehr komplex sein und sind im Allgemeinen der Fähigkeit des menschlichen Verstandes zur schnellen Bestimmung der Relevanz unterlegen. Die Qualität der Relevanzermittlung ist wahrscheinlich der wichtigste Aspekt bei der Bewertung einer Suchanwendung. Recoll stützt sich auf die probabilistische Information-Retrieval-Bibliothek Xapian, um die Relevanz zu bestimmen.

In vielen Fällen suchen Sie nach allen Formen eines Wortes, einschließlich Pluralen, verschiedenen Zeitformen eines Verbs

oder Begriffen, die von derselben Wurzel oder demselben *Wortstamm* abgeleitet sind (Beispiel: `floor, floors, floored, flooring...`). Abfragen werden in der Regel automatisch auf alle verwandten Begriffe (Wörter, die auf denselben Wortstamm zurückgehen) erweitert. Dies kann bei der Suche nach einer bestimmten Form verhindert werden.

Das Stemming an sich berücksichtigt keine Rechtschreibfehler oder phonetische Suchen. Eine Volltextsuchanwendung kann auch diese Form der Annäherung unterstützen. Beispielsweise könnte eine Suche nach *aliterattion, die* kein Ergebnis liefert, *alliteration, alteration, alterations oder altercation* als mögliche Ersatzbegriffe vorschlagen. Recoll stützt sich bei seinen Vorschlägen auf den tatsächlichen Inhalt des Index, so dass Vorschläge für Wörter gemacht werden können, die in einem Standardwörterbuch nicht vorkommen würden.

1.3 Übersicht über Recoll

Recoll verwendet die Xapian Information Retrieval Library als Speicher- und Retrieval-Engine. Xapian ist ein sehr ausgereiftes Paket, das ein hochentwickeltes probabilistisches Ranking-Modell verwendet.

Die Xapian-Bibliothek verwaltet eine Indexdatenbank, die beschreibt, wo Begriffe in Ihren Dokumentdateien vorkommen. Sie verarbeitet effizient die komplexen Abfragen, die durch den Abfrageerweiterungsmechanismus von Recoll erzeugt werden, und ist für die äußerst wichtige Relevanzberechnung zuständig.

Recoll bietet die Mechanismen und die Schnittstelle, um Daten in den und aus dem Index zu holen. Dazu gehören die Übersetzung der vielen möglichen Dokumentenformate in reinen Text, die Behandlung von Begriffsvariationen (unter Verwendung von Xapian Stemmers) und Rechtschreibannäherungen (unter Verwendung des aspell speller), die Interpretation von Benutzeranfragen und die Präsentation von Ergebnissen.

Kurz gesagt: Recoll erledigt die schmutzige Arbeit, Xapian kümmert sich um die intelligenten Teile des Prozesses.

Der Xapian-Index kann groß sein (etwa so groß wie der ursprüngliche Dokumentensatz), aber er ist kein Dokumentenarchiv. Recoll kann nur Dokumente anzeigen, die noch an dem Ort existieren, von dem aus sie indiziert wurden.

Recoll speichert alle internen Daten im Unicode UTF-8 Format und kann viele Dateitypen mit unterschiedlichen Zeichensätzen, Kodierungen und Sprachen in denselben Index aufnehmen. Es kann Dokumente verarbeiten, die in andere Dokumente eingebettet sind (z. B. ein PDF-Dokument, das in einem als E-Mail-Anhang gesendeten Zip-Archiv gespeichert ist...), und zwar bis zu einer beliebigen Tiefe.

Stemming ist der Prozess, mit dem Recoll Wörter auf ihre Radikale reduziert, so dass die Suche nicht davon abhängt, ob ein Wort Singular oder Plural ist (floor, floors), oder von einer Verbform (flooring, floored). Da die für das Stemming verwendeten Mechanismen von den spezifischen grammatikalischen Regeln jeder Sprache abhängen, gibt es für die meisten gängigen Sprachen, in denen Stemming sinnvoll ist, ein separates Xapian-Stemmer-Modul.

Recoll speichert die nicht entstammten Versionen von Begriffen im Hauptindex und verwendet Hilfsdatenbanken für die Termexpansion (eine für jede Stemming-Sprache). Das bedeutet, dass Sie die Stemming-Sprachen zwischen den Suchvorgängen wechseln oder eine Sprache hinzufügen können, ohne eine vollständige Neuindizierung vornehmen zu müssen.

Es ist möglich, Dokumente, die in verschiedenen Sprachen verfasst wurden, im selben Index zu speichern, und wird auch häufig gemacht. In diesem Fall können Sie mehrere Stemming-Sprachen für den Index angeben.

Recoll unternimmt derzeit keinen Versuch einer automatischen Spracherkennung, was bedeutet, dass der Stemmer manchmal auf Begriffe aus anderen Sprachen angewendet wird, was möglicherweise zu seltsamen Ergebnissen führt. In der Praxis hat sich dieser Ansatz, auch wenn er zu Verwechslungen führen kann, als sehr nützlich erwiesen, und er ist viel weniger umständlich als die Trennung Ihrer Dokumente nach der Sprache, in der sie geschrieben sind.

Standardmäßig entfernt Recoll die meisten Akzente und diakritischen Zeichen aus den Begriffen und wandelt sie in Kleinbuchstaben um, bevor sie im Index gespeichert werden oder nach ihnen gesucht wird. Infolgedessen ist es unmöglich, nach einer bestimmten Großschreibung eines Begriffs zu suchen (`US / us`) oder zwei Begriffe anhand von diakritischen Zeichen zu unterscheiden (`sake / saké`, `mate / maté`).

Recoll kann optional die rohen Begriffe speichern, ohne Akzententfernung oder Groß- und Kleinschreibung. In dieser Konfiguration verhalten sich die Standard-Suchvorgänge wie bisher, aber es ist möglich, Suchvorgänge unter Berücksichtigung der Groß- und Kleinschreibung durchzuführen. Dies wird im Abschnitt über die Berücksichtigung von Groß- und Kleinschreibung im Index genauer beschrieben.

Recoll verwendet viele Parameter, um genau zu definieren, was indiziert werden soll und wie die Quelldokumente klassifiziert und dekodiert werden sollen. Diese werden in Konfigurationsdateien gespeichert. Eine Standardkonfiguration wird während der Installation in ein Standardverzeichnis kopiert (normalerweise etwas wie `/usr/share/ recoll/examples`). Die Standardwerte, die in den Konfigurationsdateien in diesem Verzeichnis festgelegt sind, können durch Werte überschrieben werden, die Sie in Ihrer persönlichen Konfiguration festlegen. In der Standardkonfiguration indiziert Recoll Ihr Home-Verzeichnis mit allgemeinen Parametern. Die meisten allgemeinen Parameter können über die Konfigurationsmenüs in der Recoll-GUI eingestellt werden. Einige weniger gebräuchliche Parameter können nur durch Bearbeiten der Textdateien

eingestellt werden (die neuen Werte werden von der GUI übernommen).

Der Indizierungsprozess wird automatisch gestartet (nachdem Sie um Erlaubnis gefragt haben), wenn Sie die recoll-GUI zum ersten Mal ausführen. Die Indizierung kann auch durch die Ausführung des recollindex-Befehls durchgeführt werden. Die Indexierung von recoll ist standardmäßig multithreaded, wenn entsprechende Hardware-Ressourcen verfügbar sind, und kann mehrere Aufgaben zur Textextraktion, Segmentierung und Indexaktualisierung parallel ausführen.

Die Suche erfolgt in der Regel über die grafische Benutzeroberfläche von **recoll**, die viele Optionen bietet, um das Gesuchte zu finden. Es gibt jedoch auch andere Möglichkeiten, den Index abzufragen:

- Eine Befehlszeilenschnittstelle.
- Eine Python-Programmierschnittstelle
- Ein KDE-KIO-Slave-Modul.
- Ein Ubuntu Unity Scope Modul.
- Ein Gnome Shell Suchanbieter.
- Eine Webschnittstelle.

Kapitel 2

Indizierung

2.1 Einführung

Die Indizierung ist der Prozess, bei dem die Menge der Dokumente analysiert und die Daten in die Datenbank eingegeben werden. Die Indexierung von Recoll erfolgt normalerweise inkrementell: Dokumente werden nur verarbeitet, wenn sie seit dem letzten Durchlauf geändert wurden. Bei der ersten Ausführung müssen alle Dokumente verarbeitet werden. Ein vollständiger Indexaufbau kann später durch Angabe einer Option für den Indizierungsbefehl (**recollindex** `-z` oder `-Z`) erzwungen werden.

recollindex überspringt Dateien, die bei einem früheren Durchlauf einen Fehler verursacht haben. Dies ist eine Leistungsoptimierung, und die Kommandozeilenoption `-k` kann gesetzt werden, um fehlgeschlagene Dateien erneut zu versuchen, z. B. nach dem Aktualisieren eines Input-Handlers.

Die folgenden Abschnitte geben einen Überblick über verschiedene Aspekte der Indizierungsprozesse und der Konfiguration, mit Links zu detaillierten Abschnitten.

Abhängig von Ihren Daten können während der Indizierung temporäre Dateien benötigt werden, von denen einige möglicherweise recht groß sind. Sie können die Umgebungsvariablen `RECOLL_TMPDIR` oder `TMPDIR` verwenden, um zu bestimmen, wo sie erstellt werden (standardmäßig wird `/tmp` verwendet). Die Verwendung von `TMPDIR` hat die nette Eigenschaft, dass sie auch von Hilfsbefehlen berücksichtigt werden kann, die von **recollindex** ausgeführt werden.

2.1.1 Indizierungsmodi

Die Indexierung von Recoll kann auf zwei Arten durchgeführt werden:

- **Periodische (oder Batch-) Indizierung recollindex** wird zu diskreten Zeiten ausgeführt. Auf Unix-ähnlichen Systemen besteht die typische Verwendung darin, einen nächtlichen Lauf in Ihrer Cron-Datei zu programmieren. Unter Windows ist dies der einzige verfügbare Modus, und der Windows Task Scheduler kann zur Ausführung der Indizierung verwendet werden. In beiden Fällen bietet die grafische Benutzeroberfläche eine einfache Schnittstelle zum System-Batch-Scheduler.
- **Echtzeit-Indizierung** (nur auf Unix-ähnlichen Systemen verfügbar). **recollindex** läuft permanent als Daemon und verwendet einen Dateisystem-Änderungsmonitor (z.B. inotify), um Dateiänderungen zu erkennen. Neue oder aktualisierte Dateien werden sofort indexiert. Die Überwachung eines großen Dateisystembaums kann erhebliche Systemressourcen verbrauchen.

2.1.1.1 Unix-ähnliche Systeme: Auswahl eines Indizierungsmodus

Die Wahl zwischen den beiden Methoden ist meist eine Frage der Vorliebe, und sie können kombiniert werden, indem man mehrere Indizes einrichtet (z.B. periodische Indizierung für ein großes Dokumentationsverzeichnis und Echtzeit-Indizierung für ein kleines Home-Verzeichnis), oder mit Recoll
1.24 und neuer, indem Sie den Index so konfigurieren, dass nur eine Teilmenge des Baums überwacht wird.

Die Wahl der Methode und die verwendeten Parameter können über die GUI von **recoll** konfiguriert werden: Voreinstellungen → Indizierungsplan.

2.1.2 Konfigurationen, mehrere Indizes

Recoll unterstützt die Definition mehrerer Indizes, die jeweils durch ein eigenes Konfigurationsverzeichnis definiert werden. Ein Konfigurationsverzeichnis enthält mehrere Dateien, die beschreiben, was und wie indiziert werden soll.

Wenn **recoll** oder **recollindex** zum ersten Mal ausgeführt wird, wird ein Standardkonfigurationsverzeichnis erstellt. Diese Konfiguration wird für die Indizierung und Abfrage verwendet, wenn keine spezifische Konfiguration angegeben ist. Es befindet sich in `$HOME/.recoll/` für Unix-ähnliche Systeme und in `%LOCALAPPDATA%\Recoll` unter Windows (normalerweise `C:\Users\[me]\Appdata\Local\Recoll`).

Für alle Konfigurationsparameter gibt es Standardwerte, die in systemweiten Dateien definiert sind. Ohne weitere Anpassungen verarbeitet die Standardkonfiguration Ihr komplettes Home-Verzeichnis mit einer vernünftigen Reihe von Vorgaben. Sie kann angepasst werden, um einen anderen Bereich des Dateisystems zu verarbeiten, Dateien auf unterschiedliche Weise auszuwählen und vieles mehr.

In manchen Fällen kann es sinnvoll sein, zusätzliche Konfigurationsverzeichnisse anzulegen, um z. B. persönliche und gemeinsame Indizes zu trennen oder um die Organisation Ihrer Daten zur Verbesserung der Suchgenauigkeit zu nutzen.

Dazu legen Sie ein leeres Verzeichnis an einem Ort Ihrer Wahl an und weisen **recoll** bzw. **recollindex an**, die
durch Setzen einer Kommandozeilenoption (`-c` *`/some/directory`*) oder einer Umgebungsvariablen
(`RECOLL_CONFDIR=/some/directo` Jede durch die Befehle vorgenommene Änderung (z. B. Anpassung der Konfiguration oder Suche durch **recoll** oder Indexerstellung durch **rec- ollindex**) würde dann für das neue Verzeichnis gelten und nicht für das Standardverzeichnis.

Sobald mehrere Indizes erstellt sind, können Sie jeden von ihnen separat verwenden, indem Sie die Option `-c` oder das `RECOLL_CONFDIR` Umgebungsvariable beim Starten eines Befehls, um den gewünschten Index auszuwählen.

Es ist auch möglich, eine Konfiguration anzuweisen, zusätzlich zu ihrem eigenen Index einen oder mehrere andere Indizes abzufragen, indem Sie die Funktion Externer Index in der recoll-GUI oder einige andere Funktionen in der Kommandozeile und den Programmierwerkzeugen verwenden.

Ein plausibles Anwendungsszenario für die Funktion "Mehrere Indizes" wäre, dass ein Systemadministrator einen zentralen Index für gemeinsam genutzte Daten einrichtet, den Sie zusätzlich zu Ihren persönlichen Daten durchsuchen können oder nicht. Natürlich gibt es auch andere Möglichkeiten. So gibt es viele Fälle, in denen Sie die Teilmenge der Dateien kennen, die durchsucht werden soll, und in denen eine Einschränkung der Suche die Ergebnisse verbessern kann. Sie können ungefähr den gleichen Effekt mit dem Verzeichnisfilter in der erweiterten Suche erzielen, aber mehrere Indizes können eine bessere Leistung haben und sind in manchen Fällen die Mühe wert.

Ein fortgeschrittener Anwendungsfall wäre die Verwendung mehrerer Indizes, um die Indizierungsleistung zu verbessern, indem mehrere Indizes parallel aktualisiert werden (unter Verwendung mehrerer CPU-Kerne und Festplatten oder möglicherweise mehrerer Maschinen) und dann zusammengeführt oder parallel abgefragt werden.

Siehe den Abschnitt über die Konfiguration mehrerer Indizes für weitere Details

2.1.3 Dokumenttypen

Recoll kennt eine ganze Reihe verschiedener Dokumenttypen. Die Parameter für die Erkennung und Verarbeitung der Dokumenttypen werden in Konfigurationsdateien festgelegt.

Die meisten Dateitypen, wie HTML- oder Textverarbeitungsdateien, enthalten nur ein Dokument. Einige Dateitypen, wie E-Mail-Ordner oder Zip-Archive, können viele einzeln indizierte Dokumente enthalten, die ihrerseits zusammengesetzte Dokumente sein können. Solche Hierarchien können ziemlich tief gehen, und Recoll kann zum Beispiel ein LibreOffice-Dokument verarbeiten, das als Anhang einer E-Mail-Nachricht in einem E-Mail-Ordner gespeichert ist, der in einer Zip-Datei archiviert ist...

recollindex verarbeitet intern einfache Texte, HTML, OpenDocument (Open/LibreOffice), E-Mail-Formate und einige andere.

Andere Dateitypen (z. B. Postscript, pdf, ms-word, rtf ...) benötigen externe Anwendungen zur Vorverarbeitung. Die Liste befindet sich im Abschnitt über die Installation. Nach jedem Indizierungsvorgang aktualisiert Recoll eine Liste von Befehlen, die für die Indizierung vorhandener Dateitypen benötigt werden. Diese Liste kann durch Auswahl des Menüpunktes File Showing Helpers in der Recoll-GUI angezeigt werden. Sie ist in der `missing` text Datei im Konfigurationsverzeichnis gespeichert.

Nach der Installation eines fehlenden Handlers müssen Sie **recollindex** möglicherweise anweisen, die fehlgeschlagenen Dateien erneut zu durchsuchen, indem Sie die Option `-k` zur Befehlszeile hinzufügen oder das GUI-Menü Datei-Spezial-Indizierung

verwenden. Der Grund dafür ist, dass **recollindex** in seinem Standardmodus Dateien, die bei einem früheren Durchlauf einen Fehler verursacht haben, nicht erneut versucht. In besonderen Fällen kann es sinnvoll sein, die Daten für eine Kategorie von Dateien vor der Indizierung zurückzusetzen. Siehe dazu die Handbuchseite **recollindex**. Wenn Ihr Index nicht zu groß ist, ist es vielleicht einfacher, ihn einfach zurückzusetzen.

Standardmäßig versucht Recoll, jeden Dateityp zu indizieren, den es lesen kann. Dies ist manchmal nicht wünschenswert, und es gibt Möglichkeiten, entweder einige Typen auszuschließen oder im Gegenteil eine Positivliste der zu indizierenden Typen zu definieren. Im letzteren Fall wird jeder Typ, der nicht in der Liste enthalten ist, ignoriert.

Der Ausschluss von Dateien nach Namen kann durch Hinzufügen von Platzhalter-Namensmustern zur skippedNames-Liste erfolgen, was über das GUI-Index-Konfigurationsmenü möglich ist. Der Ausschluss nach Typ erfolgt über die Liste excludedmimetypes in der Konfigurationsdatei (1.20 und später). Dies kann für Unterverzeichnisse neu definiert werden.

Sie können auch eine exklusive Liste von MIME-Typen definieren, die indiziert werden sollen (keine anderen werden indiziert), indem Sie die Konfigurationsvariable indexedmimetypes setzen. Beispiel:

```
indexedmimetypes = text/html anwendung/pdf
```

Es ist möglich, diesen Parameter für Unterverzeichnisse neu zu definieren. Beispiel:

```
[/pfad/zu/mein/verzeichnis]
indexedmimetypes = anwendung/pdf
```

(Wenn Sie solche Abschnitte verwenden, vergessen Sie nicht, dass sie bis zum Ende der Datei oder einem anderen Abschnittskennzeichen in Kraft bleiben).

`excludedmimetypes` oder `indexedmimetypes`, können entweder durch Bearbeiten der Konfigurationsdatei (recoll.conf) für den Index oder mit dem GUI-Indexkonfigurationswerkzeug festgelegt werden.

Hinweis zu MIME-Typen

Wenn Sie die Listen `indexedmimetypes` oder `excludedmimetypes` bearbeiten, sollten Sie die MIME-Werte, die in der mimemap-Datei oder in den Recoll-Ergebnislisten aufgeführt sind, der `Datei -i`-Ausgabe vorziehen: Es gibt eine Reihe von Unterschieden. Die Ausgabe von `file -i` sollte nur für Dateien verwendet werden, die keine Erweiterungen haben oder für die die Erweiterung nicht in `mimemap` aufgeführt ist

---

2.1.4 Fehler bei der Indizierung

Die Indizierung kann bei einigen Dokumenten aus verschiedenen Gründen fehlschlagen: ein Hilfsprogramm kann fehlen, das Dokument kann beschädigt sein, wir können eine Datei nicht dekomprimieren, weil kein Speicherplatz im Dateisystem verfügbar ist, usw.

Der Recoll-Indexer in den Versionen 1.21 und höher versucht standardmäßig nicht, fehlgeschlagene Dateien erneut zu indizieren, da einige Indizierungsfehler sehr kostspielig sein können (z.B. wenn eine große Datei aufgrund von unzureichendem Speicherplatz nicht dekomprimiert werden kann). Ein erneuter Versuch erfolgt nur, wenn eine explizite Option (`-k`) auf der Kommandozeile von **recollindex** gesetzt ist oder wenn ein Skript, das beim Start von **recollindex** ausgeführt wird, dies vorgibt. Das Skript wird durch eine Konfigurationsvariable (`checkneedretryindexscript`) definiert und unternimmt einen ziemlich lahmen Versuch, herauszufinden, ob ein Hilfsbefehl installiert wurde, indem es prüft, ob sich eines der allgemeinen Bin-Verzeichnisse geändert hat.

2.1.5 Erholung

In dem seltenen Fall, dass der Index beschädigt wird (was sich durch seltsame Suchergebnisse oder Abstürze bemerkbar machen kann), müssen die Indexdateien gelöscht werden, bevor ein neuer Indizierungsdurchlauf gestartet wird. Löschen Sie einfach das xapiandb-Verzeichnis (siehe nächster Abschnitt), oder starten Sie alternativ den nächsten **recollindex** mit der Option `-z`, die die Datenbank vor der Indizierung zurücksetzt. Der Unterschied zwischen den beiden Methoden besteht darin, dass die zweite Methode das aktuelle Indexformat nicht ändert, was unerwünscht sein kann, wenn die Xapian-Version ein neueres Format unterstützt.

2.2 Indexspeicher

Der Standardspeicherort für die Indexdaten ist das Unterverzeichnis `xapiandb` des Recoll-Konfigurationsverzeichnisses, normalerweise `$HOME/.recoll/xapiandb/`. Dies kann über zwei verschiedene Methoden (mit unterschiedlichen Zielen) geändert werden:

1. Für ein bestimmtes Konfigurationsverzeichnis können Sie einen vom Standard abweichenden Speicherort für den Index angeben, indem Sie den Parameter `dbdir` in der Konfigurationsdatei setzen (siehe Abschnitt Konfiguration). Diese Methode ist vor allem dann von Nutzen, wenn Sie das Konfigurationsverzeichnis an seinem Standardspeicherort belassen wollen, aber einen anderen Speicherort für den Index wünschen, typischerweise aus Gründen der Plattenbelegung oder der Leistung.

2. Sie können ein anderes Konfigurationsverzeichnis angeben, indem Sie die Umgebungsvariable `RECOLL_CONFDIR` setzen oder den Befehl
   Option `-c` für die Recoll-Befehle. Diese Methode wird normalerweise verwendet, um verschiedene Bereiche des Dateisystems mit unterschiedlichen Indizes zu indizieren. Wenn Sie zum Beispiel den folgenden Befehl eingeben würden:

```
recoll -c ~/.indexes-email
```

   Dann würde Recoll Konfigurationsdateien verwenden, die in `~/.indexes-email/` gespeichert sind, und (sofern nicht anders in `recoll.conf` angegeben) nach dem Index in `~/.indexes-email/xapiandb/` suchen.

   Die Verwendung mehrerer Konfigurationsverzeichnisse und Konfigurationsoptionen ermöglicht es Ihnen, mehrere Konfigurationen und Indizes für jede Teilmenge der verfügbaren Daten, die Sie durchsuchbar machen möchten, anzupassen.

Die Größe des Index wird durch die Größe des Dokumentensatzes bestimmt, aber das Verhältnis kann stark variieren. Bei einem typischen gemischten Dokumentensatz liegt die Indexgröße oft nahe an der Größe des Datensatzes. In bestimmten Fällen (z. B. bei einem Satz komprimierter mbox-Dateien) kann der Index viel größer sein als die Dokumente. Er kann auch viel kleiner sein, wenn die Dokumente viele Bilder oder andere nicht indizierte Daten enthalten (ein extremes Beispiel wäre ein Satz von mp3-Dateien, bei dem nur die Tags indiziert würden).

Natürlich erhöhen Bilder, Ton und Video die Indexgröße nicht, was bedeutet, dass in den meisten Fällen der vom Index belegte Platz im Vergleich zur Gesamtdatenmenge auf dem Computer vernachlässigbar ist.

Das Indexdatenverzeichnis (`xapiandb`) enthält nur Daten, die durch einen Indexlauf vollständig wiederhergestellt werden können (solange die Originaldokumente existieren), und es kann immer sicher zerstört werden.

2.2.1 Xapian-Indexformate

Xapian-Versionen unterstützen normalerweise mehrere Formate für die Indexspeicherung. Eine bestimmte Xapian-Hauptversion hat ein aktuelles Format, das zur Erstellung neuer Indizes verwendet wird, und unterstützt auch das Format der vorherigen Hauptversion.

Xapian konvertiert einen bestehenden Index nicht automatisch von dem älteren Format in das neuere Format. Wenn Sie auf das neue Format aktualisieren wollen oder wenn ein sehr alter Index konvertiert werden muss, weil sein Format nicht mehr unterstützt wird, müssen Sie den alten Index explizit löschen (typischerweise `~/.recoll/xapiandb`) und dann einen normalen Indexierungsbefehl ausführen. Die Verwendung der recollindex-Option `-z` würde in dieser Situation nicht funktionieren.

2.2.2 Sicherheitsaspekte

Der Recoll-Index enthält keine vollständigen Kopien der indizierten Dokumente (nach Version 1.24 ist er fast vollständig). Er enthält jedoch genügend Daten, um eine nahezu vollständige Rekonstruktion zu ermöglichen. Wenn vertrauliche Daten indiziert werden, sollte der Zugriff auf das Datenbankverzeichnis eingeschränkt werden.

Recoll erstellt das Konfigurationsverzeichnis mit einem Modus von 0700 (Zugriff nur durch den Eigentümer). Da das Indexdatenverzeichnis standardmäßig ein Unterverzeichnis des Konfigurationsverzeichnisses ist, sollte dies zu einem angemessenen Schutz führen.

Wenn Sie ein anderes Setup verwenden, sollten Sie sich überlegen, welche Art von Schutz Sie für Ihren Index benötigen, die Zugriffsmodi für Verzeichnisse und Dateien entsprechend einstellen und eventuell auch die bei Indexaktualisierungen verwendete `umask` anpassen.

2.2.3 Besondere Überlegungen für große Indizes

Dies muss Sie nur betreffen, wenn Ihr Index größer als etwa GBytes5 ist. Bei mehr als einem GByte10 wird es zu einem ernsten Problem. Die meisten Leute haben viel kleinere Indizes. Als Anhaltspunkt: GBytes5 sind etwa Bibeln2000, eine Menge Text. Wenn Sie einen großen Textdatensatz haben (denken Sie daran: Bilder zählen nicht, der Textinhalt von PDFs macht in der Regel weniger als 5 % der Dateigröße aus), lesen Sie weiter.

Die Menge der von Xapian während der Indexerstellung durchgeführten Schreibvorgänge ist nicht linear mit der Indexgröße (sie liegt irgendwo zwischen linear und quadratisch). Bei großen Indizes wird dies zu einem Leistungsproblem und kann sogar ein Problem mit dem Verschleiß der SSD-Festplatte sein.

Das Problem kann durch die Einhaltung der folgenden Regeln entschärft werden:

- Partitionieren Sie den Datensatz und erstellen Sie mehrere Indizes von angemessener Größe statt eines großen Indexes. Diese Indizes können dann parallel abgefragt werden (unter Verwendung der Recoll-Funktion für externe Indizes) oder mit **xapian-compact** zusammengeführt werden.

- Stellen Sie viel RAM zur Verfügung und setzen Sie den `idxflushmb` Recoll-Konfigurationsparameter so hoch wie möglich, ohne zu swappen (Experimentieren ist erforderlich). wäre200 in diesem Zusammenhang ein Minimum.
- Verwenden Sie Xapian oder1.4.10 eine neuere Version, da diese Version eine erhebliche Verbesserung der Schreibvorgänge mit sich bringt.

2.3 Index-Konfiguration

Die in den Recoll-Konfigurationsdateien gespeicherten Variablen steuern, welche Bereiche des Dateisystems indiziert werden und wie die Dateien geprüft werden. Die Werte können durch Editieren der Textdateien eingestellt werden. Die meisten der am häufigsten verwendeten Variablen können auch über die Dialoge in der Recoll-GUI angepasst werden.

Wenn Sie recoll zum ersten Mal starten, werden Sie gefragt, ob der Index erstellt werden soll oder nicht. Wenn Sie die Konfiguration vor der Indizierung anpassen möchten, klicken Sie an dieser Stelle einfach auf Abbrechen, um in die Konfigurationsoberfläche zu gelangen. Wenn Sie an dieser Stelle abbrechen, hat `recoll` ein Standardkonfigurationsverzeichnis mit leeren Konfigurationsdateien angelegt, die Sie dann bearbeiten können.

Die Konfiguration ist im Installationskapitel dieses Dokuments oder in der Handbuchseite recoll.conf(5) dokumentiert. Beide Dokumente werden automatisch aus den Kommentaren in der Konfigurationsdatei generiert.

Die unmittelbar nützlichste Variable ist wahrscheinlich topdirs, die die zu indizierenden Teilbäume und Dateien auflistet.

Die Anwendungen, die benötigt werden, um andere Dateitypen als Text, HTML oder E-Mail zu indizieren (z.B.: pdf, postscript, ms-word...), werden im Abschnitt Externe Pakete beschrieben.

Es gibt zwei inkompatible Arten von Recoll-Indizes, abhängig von der Behandlung von Groß- und Kleinschreibung und diakritischen Zeichen. In einem weiteren Abschnitt werden die beiden Typen näher beschrieben. Der Standardtyp ist in den meisten Fällen angemessen.

2.3.1 Mehrere Indizes

Mehrere Recoll-Indizes können durch die Verwendung mehrerer Konfigurationsverzeichnisse erstellt werden, die in der Regel so eingestellt sind, dass sie verschiedene Bereiche des Dateisystems indizieren.

Ein bestimmter Index kann ausgewählt werden, indem die Umgebungsvariable `RECOLL_CONFDIR` gesetzt oder die Option `-c` an **recoll** übergeben wird und
recollindex.

Das Programm **recollindex**, das zum Erstellen oder Aktualisieren von Indizes verwendet wird, arbeitet immer mit einem einzigen Index. Die verschiedenen Konfigurationen sind völlig unabhängig voneinander (beim Indizieren werden keine Parameter zwischen den Konfigurationen ausgetauscht).

Alle Suchschnittstellen (**recoll**, **recollq**, die Python-API usw.) arbeiten mit einer Hauptkonfiguration, von der sowohl Konfigurations- als auch Indexdaten verwendet werden, und können auch Daten von mehreren zusätzlichen Indizes abfragen. Von letzteren werden nur die Indexdaten verwendet, ihre Konfigurationsparameter werden ignoriert. Dies bedeutet, dass einige Parameter zwischen Indexkonfigurationen, die gemeinsam verwendet werden sollen, konsistent sein sollten.

Bei der Suche ist immer der aktuelle Hauptindex (definiert durch `RECOLL_CONFDIR` oder `-c`) aktiv. Wenn dies unerwünscht ist, können Sie Ihre Basiskonfiguration so einrichten, dass ein leeres Verzeichnis indiziert wird.

Indexkonfigurationsparameter können entweder mit einem Texteditor für die Dateien oder, für die meisten Parameter, mit der Indexkonfigurations-GUI von recoll gesetzt werden. Im letzteren Fall ist das Konfigurationsverzeichnis, für das die Parameter geändert werden, dasjenige, das durch `RECOLL_CONFDIR` oder den Parameter `-c` ausgewählt wurde, und es gibt keine Möglichkeit, die Konfiguration innerhalb der GUI zu wechseln.

Eine detaillierte Beschreibung der Parameter finden Sie im Abschnitt Konfiguration

Einige Konfigurationsparameter müssen bei einer Reihe von mehreren Indizes, die gemeinsam für Suchvorgänge verwendet werden, konsistent sein. Am wichtigsten ist, dass alle Indizes, die gleichzeitig abgefragt werden sollen, die gleiche Option für das Entfernen von Groß- und Kleinschreibung und diakritischen Zeichen haben müssen, aber es gibt noch weitere Einschränkungen. Die meisten der relevanten Parameter betreffen die Termgenerierung.

Die Verwendung mehrerer Konfigurationen erfordert einen geringen Einsatz der Befehlszeile oder des Dateimanagers. Der Benutzer muss explizit zusätzliche Konfigurationsverzeichnisse erstellen, die grafische Benutzeroberfläche tut dies nicht.

Damit soll vermieden werden, dass versehentlich zusätzliche Verzeichnisse angelegt werden, wenn ein Argument falsch eingegeben wird. Außerdem muss die grafische Benutzeroberfläche oder der Indexer mit einer bestimmten Option oder Umgebung gestartet werden, um mit der richtigen Konfiguration zu arbeiten.

2.3.1.1 In der Praxis: Erstellung und Verwendung eines zusätzlichen Index

Erstmalige Erstellung der Konfiguration und des Index:

```
mkdir /pfad/zu/mein/neu/konfig
```

Die Konfiguration des neuen Index kann über die grafische Benutzeroberfläche **von recoll** oder über die Befehlszeile mit der Option `-c` erfolgen (Sie können eine Desktop-Datei erstellen, die dies für Sie erledigt), und dann das GUI-Indexkonfigurationswerkzeug zum Einrichten des Index verwenden.

```
recoll -c /pfad/zu/mein/neu/konfig
```

Alternativ können Sie auch einfach einen Texteditor für die Hauptkonfigurationsdatei starten:

```
someEditor /pfad/zu/mein/neu/config/recoll.conf
```

Das Erstellen und Aktualisieren des Indexes kann über die Befehlszeile erfolgen:

```
recollindex -c /pfad/zu/mein/neu/konfig
```

oder aus dem Menü Datei einer mit der gleichen Option gestarteten grafischen Benutzeroberfläche (**recoll**, siehe oben).

Über dieselbe grafische Benutzeroberfläche können Sie auch die Batch-Indizierung für den neuen Index einrichten. Die Echtzeit-Indizierung kann nur über die grafische Benutzeroberfläche für den Standardindex eingerichtet werden (der Menüeintrag ist inaktiv, wenn die grafische Benutzeroberfläche mit der Option `-c` gestartet wurde, die nicht dem Standard entspricht).

Der neue Index kann allein abgefragt werden mit

```
recoll -c /pfad/zu/mein/neu/konfig
```

Oder, parallel zum Standardindex, indem Sie **recoll** ohne die Option `-c` starten und das Menü Preferences External Index Dialog verwenden.

2.3.2 Sensibilität für Groß- und Kleinschreibung und diakritische Zeichen

Ab der Version Recoll 1.18haben Sie die Wahl, ob Sie einen Index mit Begriffen ohne Groß- und Kleinschreibung oder einen Index mit Rohbegriffen erstellen wollen. Für einen Quellbegriff `Résumé` speichert ersterer den `Lebenslauf`, letzterer das `Résumé`.

Jeder Indextyp ermöglicht eine Suche ohne Berücksichtigung von Groß- und Kleinschreibung: Bei einem Rohindex wird die Benutzereingabe so erweitert, dass sie allen im Index vorhandenen Varianten von Groß- und Kleinschreibung entspricht. Bei einem "stripped index" wird der Suchbegriff vor der Suche entfernt.

Ein roher Index ermöglicht die Verwendung von Groß- und Kleinschreibung zur Unterscheidung von Begriffen, z. B. die Ausgabe unterschiedlicher Ergebnisse bei der Suche nach
`US` und `us` oder `Lebenslauf` und `Resümee`. Lesen Sie den Abschnitt über Groß- und Kleinschreibung und diakritische Zeichen für weitere Details.

Die Art des zu erstellenden Index wird durch die Konfigurationsvariable `indexStripChars` gesteuert, die nur durch Bearbeitung der Konfigurationsdatei geändert werden kann. Jede Änderung impliziert einen Index-Reset (nicht automatisch von Recoll), und alle Indizes in einer Suche müssen auf die gleiche Weise gesetzt werden (wiederum nicht von Recoll überprüft).

Recoll erstellt standardmäßig einen gestrippten Index, wenn `indexStripChars` nicht gesetzt ist.

Als Preis für die zusätzlichen Möglichkeiten wird ein Rohindex etwas größer sein als ein bereinigter Index (etwa 10 %). Außerdem sind die Suchvorgänge komplexer und daher wahrscheinlich etwas langsamer, und die Funktion wird relativ wenig genutzt, so dass ein gewisses Maß an Seltsamkeit nicht ausgeschlossen werden kann.

Eine der nachteiligsten Folgen der Verwendung eines Rohindexes ist, dass einige Phrasen- und Proximity-Suchen unmöglich werden können: Da jeder Begriff erweitert werden muss und alle Kombinationen gesucht werden müssen, kann die multiplikative Erweiterung unüberschaubar werden.

2.3.3 Konfiguration der Indizierungs-Threads (Unix-ähnliche Systeme)

Der Recoll-Indizierungsprozess **recollindex** kann mehrere Threads verwenden, um die Indizierung auf Multiprozessorsystemen zu beschleunigen. Die Indizierung der Dateien erfolgt in mehreren Schritten, wobei einige dieser Schritte von mehreren Threads ausgeführt werden können. Die Schritte sind:

1. Dateisystemlauf: Dieser wird immer vom Hauptthread ausgeführt.
2. Dateikonvertierung und Datenextraktion.
3. Textverarbeitung (Splitting, Stemming, etc.).
4. Aktualisierung des Xapian-Index.

Sie können auch ein längeres Dokument über die Umwandlung der Recoll-Indizierung in Multithreading lesen. Die Konfiguration der Threads wird durch zwei Parameter in der Konfigurationsdatei gesteuert.

thrQSizes Diese Variable definiert die Konfiguration der Auftragseingangs-Warteschlangen. Es gibt drei mögliche Warteschlangen für die einzelnen Phasen, 2,3und 4,dieser Parameter sollte die Tiefe der Warteschlange für jede Phase angeben (drei ganzzahlige Werte). Wird für einen bestimmten Schritt der Wert -1 verwendet, wird keine Warteschlange benutzt und der Thread fährt mit dem nächsten Schritt fort. In der Praxis hat sich gezeigt, dass tiefe Warteschlangen die Leistung nicht erhöhen. Ein Wert von für0 die erste Warteschlange weist Recoll an, eine Autokonfiguration durchzuführen (in diesem Fall ist nichts weiter erforderlich, thrTCounts wird nicht verwendet) - dies ist die Standardkonfiguration.

thrTCounts Hier wird die Anzahl der für jede Stufe verwendeten Threads festgelegt. Wird für eine der Warteschlangentiefen ein Wert von -1 verwendet, wird die entsprechende Thread-Anzahl ignoriert. Es ist nicht sinnvoll, einen anderen Wert als für1 die letzte Stufe zu verwenden, da die Aktualisierung des Xapian-Index notwendigerweise single-threaded (und durch einen Mutex geschützt) erfolgt.

Hinweis
Wenn der erste Wert in `thrQSizes thrTCounts`0, ist, wird er ignoriert.

Im folgenden Beispiel würden drei Warteschlangen (mit Tiefe 2) und 4 Threads für die Konvertierung der Quelldokumente, 2 für die Verarbeitung des Textes und einer für die Aktualisierung des Index verwendet. Dies hat sich auf dem Testsystem (Quadroprozessor mit mehreren Festplatten) als die beste Konfiguration erwiesen.

```
thrQSizes = 222
thrTCounts = 421
```

Im folgenden Beispiel würde eine einzige Warteschlange verwendet, und die gesamte Verarbeitung für jedes Dokument würde von einem einzigen Thread durchgeführt (in den meisten Fällen werden dennoch mehrere Dokumente parallel verarbeitet). Die Threads schließen sich gegenseitig aus, wenn sie die Phase der Indexaktualisierung erreichen. In der Praxis wäre die Leistung im Allgemeinen ähnlich wie im vorhergehenden Fall, aber in bestimmten Fällen schlechter (z. B. würde ein Zip-Archiv rein sequentiell verarbeitet), so dass der vorherige Ansatz vorzuziehen ist. YMMV... Die 2 letzten Werte für thrTCounts werden ignoriert.

```
thrQSizes = -12 -1
thrTCounts = 611
```

Das folgende Beispiel würde das Multithreading deaktivieren. Die Indizierung wird von einem einzigen Thread durchgeführt.

```
thrQSizes = -1 -1 -1
```

2.3.4 Die GUI für die Indexkonfiguration

Die meisten Parameter für eine gegebene Indexkonfiguration können von einer Recoll-GUI aus gesetzt werden, die auf dieser Konfiguration läuft (entweder als Standard, oder durch Setzen von `RECOLL_CONFDIR` oder der Option `-c`).

Die Schnittstelle wird über den Menüeintrag→Einstellungen Index Konfiguration aufgerufen. Sie ist in vier Registerkarten unterteilt: Globale Parameter, Lokale Parameter, Webverlauf (der im nächsten Abschnitt erläutert wird) und Suchparameter.

Auf der Registerkarte "Globale Parameter" können Sie globale Variablen festlegen, z. B. die Listen der wichtigsten Verzeichnisse, übersprungene Pfade oder Stemming-Sprachen.

Auf der Registerkarte Lokale Parameter können Sie Variablen festlegen, die für Unterverzeichnisse neu definiert werden können. Diese zweite Registerkarte enthält eine zunächst leere Liste von Anpassungsverzeichnissen, die Sie hinzufügen können. Die Variablen werden dann für das aktuell ausgewählte Verzeichnis gesetzt (oder auf der obersten Ebene, wenn die leere Zeile ausgewählt ist).

Der Abschnitt Suchparameter definiert Parameter, die zur Abfragezeit verwendet werden, aber global für einen Index sind und alle Suchwerkzeuge beeinflussen, nicht nur die GUI.

Die Bedeutung der meisten Einträge in der Schnittstelle ist selbsterklärend und wird durch ein ToolTip-Popup auf dem Textlabel dokumentiert. Weitere Einzelheiten finden Sie im Abschnitt über die Konfiguration in diesem Handbuch.

Das Konfigurationstool respektiert normalerweise die Kommentare und den größten Teil der Formatierung in der Konfigurationsdatei, so dass es durchaus möglich ist, es für manuell bearbeitete Dateien zu verwenden, die Sie vielleicht trotzdem vorher sichern wollen...

2.4 Herausnehmbare Bände

Recoll bot früher keine Unterstützung für die Indizierung von Wechseldatenträgern (tragbare Festplatten, USB-Sticks usw.). Neuere Versionen haben die Situation verbessert und unterstützen die Indizierung von Wechseldatenträgern auf zwei verschiedene Arten:

- Durch Indizierung des Datenträgers im festen Hauptindex und Sicherstellung, daß die Datenträgerdaten nicht gelöscht werden, wenn die Indizierung läuft, während der Datenträger eingelegt ist. (seit Recoll 1.25.2).
- Durch Speicherung eines Datenträgerindexes auf dem Datenträger selbst (seit Recoll 1.24).

2.4.1 Indizierung von Wechseldatenträgern im Hauptindex

Ab Version 1.25.2 bietet Recoll eine einfache Möglichkeit, um sicherzustellen, dass die Indexdaten für einen nicht vorhandenen Datenträger nicht gelöscht werden. Zwei Bedingungen müssen erfüllt sein:

- Der Einhängepunkt des Volumes muss in der Liste `topdirs` enthalten sein.
- Das Einhängeverzeichnis muss leer sein (wenn der Datenträger nicht eingehängt ist).

Wenn **recollindex** beim Starten feststellt, dass eines der `Topdirs` leer ist, werden alle vorhandenen Daten für den Baum durch den Indizierungsdurchlauf erhalten (keine Bereinigung für diesen Bereich).

2.4.2 Selbständige Bände

Seit Recoll 1.24 ist es möglich, in sich geschlossene Datensätze zu erstellen, die ein Recoll-Konfigurationsverzeichnis und einen Index zusammen mit den indizierten Dokumenten enthalten, und einen solchen Datensatz zu verschieben (z. B. auf ein USB-Laufwerk zu kopieren), ohne die Konfiguration für die Abfrage des Index anpassen zu müssen.

Hinweis
Dies ist eine Funktion, die nur zur Abfragezeit gilt. Der Index darf nur an seinem ursprünglichen Speicherort aktualisiert werden. Wenn eine Aktualisierung an einem anderen Ort erforderlich ist, muss der Index zurückgesetzt werden.

Das Funktionsprinzip besteht darin, dass die Konfiguration den Ort des ursprünglichen Konfigurationsverzeichnisses speichert, das sich auf dem beweglichen Datenträger befinden muss. Wenn der Datenträger später an einem anderen Ort gemountet wird, passt Recoll die im Index gespeicherten Pfade um die Differenz zwischen dem ursprünglichen und dem aktuellen Speicherort des Konfigurationsverzeichnisses an.

Um es kurz zu machen, hier folgt ein Skript zur Erstellung einer Recoll-Konfiguration und eines Indexes unter einem bestimmten Verzeichnis (als einzelner Parameter angegeben). Der resultierende Datensatz (Dateien + Recoll-Verzeichnis) kann später auf eine CDROM oder einen USB-Stick verschoben werden. Längere Erklärungen folgen nach dem Skript.

```
#!/bin/sh
fatal()
{
    echo $*;exit 1
}
usage()
{
    fatal "Verwendung: init-recoll-volume.sh <Top-Verzeichnis>"
}
test $# -eq 1 || usage
topdir=$1
test -d "$topdir" || fatal $topdir sollte ein Verzeichnis sein
confdir="$topdir/recoll-confdir"
test ! -d "$confdir" || fatal "$confdir existiert bereits"
mkdir
"$confdir" cd
"$topdir"
topdir='pwd'
cd "$confdir"
confdir='pwd'
(echo topdirs = '"'"$topdir"'"' ; \
 echo orgidxconfdir = $topdir/recoll-config) > "$confdir/recoll.conf"
recollindex -c "$confdir"
```

In den folgenden Beispielen wird davon ausgegangen, dass Sie ein Dataset unter `/home/me/mydata/` haben, wobei die Indexkonfiguration und die Daten in `/home/me/mydata/recoll-confdir` gespeichert sind.

Um nach dem Verschieben des Datensatzes Abfragen durchführen zu können, müssen Sie Folgendes sicherstellen:

- In der Hauptkonfigurationsdatei muss die Variable orgidxconfdir auf den ursprünglichen Speicherort des Konfigurationsverzeichnisses festgelegt werden (`orgidxconfdir=/home/me/mydata/recoll-confdir` muss im obigen Beispiel innerhalb von `/home/me/mydata/recoll-confdir/recoll.conf` gesetzt werden).
- Das Konfigurationsverzeichnis muss zusammen mit den Dokumenten irgendwo unterhalb des zu verschiebenden Verzeichnisses existieren. Wenn Sie z. B. `/home/me/mydata` verschieben, muss das Konfigurationsverzeichnis irgendwo unterhalb dieses Punktes liegen, z. B. `/home/me/mydata/recoll-confdir`, oder `/home/me/mydata/sub/recoll-confdir`.
- Sie sollten die Standardpfade für die Indexelemente beibehalten, die standardmäßig relativ zum Konfigurationsverzeichnis sind (vor allem `tmpdir`). Nur die Pfade, die sich auf die Dokumente selbst beziehen (z. B. topdirs-Werte), sollten absolut sein (im Allgemeinen werden sie ohnehin nur beim Indizieren verwendet).

Nur der erste Punkt erfordert eine explizite Benutzeraktion, die Standardeinstellungen von Recoll sind mit dem dritten Punkt kompatibel, und der zweite Punkt ist selbstverständlich.

Wenn nach dem Verschieben das Konfigurationsverzeichnis aus dem Datensatz kopiert werden muss (z. B. weil das Laufwerk zu langsam ist), können Sie die Variable curidxconfdir in der kopierten Konfiguration setzen, um den Speicherort der verschobenen Konfiguration zu definieren. Wenn zum Beispiel `/home/me/mydata` jetzt auf `/media/me/somelabel` gemountet ist, aber das Konfigurationsverzeichnis und der Index nach `/tmp/tempconfig` kopiert wurden, würden Sie `curidxconfdir` auf `/media/me/somelabel/recoll-confdir` innerhalb von

`/tmp/tempconfig/recoll.conf`. `orgidxconfdir` würde im Original und in der Kopie immer noch `/home/me/mydata/recoll-confdir` lauten.

Wenn Sie die Konfiguration regelmäßig aus dem Datensatz herauskopieren, ist es sinnvoll, ein Skript zu schreiben, um den Vorgang zu automatisieren. Dies kann nicht wirklich innerhalb von Recoll geschehen, da es wahrscheinlich viele mögliche Varianten gibt. Ein Beispiel wäre, die Konfiguration so zu kopieren, dass sie beschreibbar wird, aber die Indexdaten auf dem Datenträger zu belassen, weil sie zu groß sind - in diesem Fall müsste das Skript auch `dbdir` in der kopierten Konfiguration setzen.

Durch dieselben Änderungen (Recoll 1.24) wurde es auch möglich, Abfragen von einem schreibgeschützten Konfigurationsverzeichnis aus zu starten (natürlich mit leicht eingeschränkter Funktion, z. B. ohne Aufzeichnung der Abfragehistorie).

2.5 Unix-ähnliche Systeme: Indizierung besuchter Web-Seiten

Mit Hilfe einer Firefox-Erweiterung kann Recoll die von Ihnen besuchten Internetseiten indizieren. Die Erweiterung hat eine lange Geschichte: Sie wurde ursprünglich für den Beagle-Indexer entwickelt und dann an Recoll und die Firefox XUL API angepasst. Die aktuelle Version der Erweiterung befindet sich im Mozilla Add-ons Repository, verwendet die WebExtensions API und funktioniert mit den aktuellen Firefox-Versionen.

Die Erweiterung kopiert besuchte Webseiten in ein Verzeichnis der Indizierungswarteschlange, die Recoll dann verarbeitet, die Daten in einem lokalen Cache speichert, sie dann indiziert und die Datei anschließend aus der Warteschlange entfernt.

Der lokale Cache ist kein Archiv
Wie bereits erwähnt, speichert Recoll eine Kopie der indizierten Webseiten in einem lokalen Cache (aus dem Daten für Vorschauen oder beim Zurücksetzen des Index geholt werden). Der Cache wird bei einem Index-Reset nicht verändert, sondern nur für die Indizierung ausgelesen. Der Cache hat eine maximale Größe, die in der Indexkonfiguration / Web History eingestellt werden kann (Parameter `webcachemaxmbs` in `recoll.conf`). Sobald die maximale Größe erreicht ist, werden alte Seiten gelöscht, um Platz für neue zu schaffen. Die Seiten, die Sie auf unbestimmte Zeit aufbewahren möchten, müssen explizit an anderer Stelle archiviert werden. Die Verwendung eines sehr hohen Wertes für die Cache-Größe kann das Löschen von Daten vermeiden, aber auf der obigen 'Howto'-Seite finden Sie weitere Details und Probleme.

Die Funktion zur Indizierung besuchter Webseiten kann auf der Recoll-Seite über das GUI-Index-Konfigurationspanel oder durch Bearbeiten der Konfigurationsdatei (`processwebqueue` auf 1 setzen) aktiviert werden.

Die GUI von Recoll verfügt über ein Werkzeug zum Auflisten und Bearbeiten des Inhalts des Webcaches.

(Extras → Webcache-Editor) Der Befehl **recollindex** bietet zwei Optionen zur Verwaltung des Web-

Cache:

- `--webcache-compact` stellt den Platz von gelöschten Einträgen wieder her. Es kann sein, dass doppelt so viel Speicherplatz wie derzeit für den Webcache benötigt wird, verwendet werden muss.
- `--webcache-burst` *`destdir`* extrahiert alle aktuellen Einträge in Paare von Metadaten- und Datendateien, die in *`destdir`* erstellt werden

Weitere Details zur Web-Indizierung, ihrer Verwendung und Konfiguration finden Sie in einem 'Howto'-Eintrag von Recoll.

2.6 Unix-ähnliche Systeme: Verwendung erweiterter Attribute

Benutzererweiterte Attribute sind benannte Informationen, die die meisten modernen Dateisysteme an jede Datei anhängen können.

Recoll verarbeitet alle erweiterten Attribute als Dokumentfelder. Beachten Sie, dass die meisten Felder standardmäßig nicht indiziert sind. Sie müssen sie aktivieren, indem Sie ein Präfix in der Feldkonfigurationsdatei definieren.

Ein Freedesktop-Standard definiert ein paar spezielle Attribute, die von Recoll als solche behandelt werden:

mime_type Wenn gesetzt, hat dies Vorrang vor jeder anderen Bestimmung des MIME-Typs der Datei.

charset Wenn gesetzt, definiert dies den Zeichensatz der Datei (meist nützlich für reine Textdateien).

Standardmäßig werden andere Attribute als Recoll-Felder mit demselben Namen behandelt. Unter Linux wird das Benutzerpräfix aus dem Namen entfernt.

Die Namensübersetzung kann genauer konfiguriert werden, auch in der Feldkonfigurationsdatei.

2.7 Unix-ähnliche Systeme: Importieren externer Tags

Während der Indizierung ist es möglich, durch die Ausführung von Befehlen Metadaten für jede Datei zu importieren. Auf diese Weise können z. B. Tag-Daten aus einer externen Anwendung extrahiert und in einem Feld für die Indizierung gespeichert werden.

Eine Beschreibung der Konfigurationssyntax finden Sie im Abschnitt über das Feld metadatacmds im Kapitel über die Hauptkonfiguration.

Wenn Sie zum Beispiel möchten, dass Recoll die von tmsu verwalteten Tags in einem Feld namens *tags* verwendet, fügen Sie der Konfigurationsdatei Folgendes hinzu

```
[/some/area/of/the/fs]
metadatacmds = ; tags = tmsu tags %f
```

Hinweis

Je nach tmsu-Version müssen/wollen Sie möglicherweise Optionen wie `--database=/some/db` hinzufügen.

Möglicherweise möchten Sie diese Verarbeitung auf eine Teilmenge des Verzeichnisbaums beschränken, da sie die Indizierung etwas verlangsamen kann (`[some/area/of/th` Beachten Sie das erste Semikolon nach dem Gleichheitszeichen.

Im obigen Beispiel wird die Ausgabe von **tmsu** verwendet, um ein Feld namens *tags* zu setzen. Der Feldname ist willkürlich und könnte genauso gut *tmsu* oder *myfield* lauten, aber *tags* ist ein Alias für das Standard-Schlüsselwortfeld von Recoll, und die tmsu-Ausgabe wird lediglich dessen Inhalt ergänzen. Dadurch wird die Notwendigkeit vermieden, die Feldkonfiguration zu erweitern.

Sobald die Neuindizierung erfolgt ist (Sie müssen die Neuindizierung der Datei erzwingen, Recoll erkennt die Notwendigkeit nicht von selbst), können Sie über die Abfragesprache suchen, und zwar über einen ihrer Aliase: *tags:some/alternate/values* oder *tags:all,these,values*.
Die kompakte Komma- oder Schrägstrich-basierte Feldsuchsyntax wird für recoll und1.20 spätere Versionen unterstützt. Bei älteren Versionen müssen Sie den *tags:*-Spezifizierer für jeden Begriff wiederholen, z. B. *tags:some* `OR` *tags:alternate*.

Änderungen an den Tags werden vom Indexer nicht erkannt, wenn sich die Datei selbst nicht geändert hat. Ein möglicher Workaround wäre, die Datei `ctime` zu aktualisieren, wenn Sie die Tags ändern, was der Funktionsweise der erweiterten Attribute entsprechen würde. Ein Paar chmod-Befehle könnte dies bewerkstelligen, oder ein `touch -a`. Alternativ können Sie die Aktualisierung der Tags auch mit einem `recollindex -e -i` */pfad/zu/der/datei*.

2.8 Der PDF-Eingabe-Handler

Das PDF-Format ist für die wissenschaftliche und technische Dokumentation und die Archivierung von Dokumenten sehr wichtig. Es verfügt über umfangreiche Möglichkeiten zur Speicherung von Metadaten zusammen mit dem Dokument, und diese Möglichkeiten werden in der Praxis tatsächlich genutzt.

Der **rclpdf.**py-PDF-Input-Handler hat daher komplexere Fähigkeiten als die meisten anderen, und er ist auch besser konfigurierbar. Im Einzelnen verfügt **rclpdf.py** über die folgenden Funktionen:

- Es kann so konfiguriert werden, dass es bestimmte Metadaten-Tags aus einem XMP-Paket extrahiert.
- Es kann PDF-Anhänge extrahieren.
- Es kann automatisch eine OCR durchführen, wenn der Text des Dokuments leer ist. Dies geschieht durch die Ausführung eines externen Programms und wird nun in einem separaten Abschnitt beschrieben, da das OCR-Framework auch mit Nicht-PDF-Bilddateien verwendet werden kann.

2.8.1 XMP-Feldern

Das Skript `rclpdf.py` in Recoll Version 1.23.2 und höher kann XMP-Metadatenfelder extrahieren, indem es den Befehl **pdfinfo** ausführt (normalerweise mit poppler-utils zu finden). Dies wird durch die Konfigurationsvariable pdfextrameta gesteuert, die angibt, welche Tags extrahiert werden sollen und möglicherweise, wie sie umbenannt werden sollen.

Die Variable pdfextrametafix kann verwendet werden, um eine Datei mit Python-Code zur Bearbeitung der Metadatenfelder

anzugeben (verfügbar für Recoll
1.23.3 und später. hat 1.23.2entsprechenden Code innerhalb des Handler-Skripts). Beispiel:

```
System einführen
        Importware

        class
            MetaFixer(object):
            def init (self):
                Pass

            def metafix(self, nm, txt):
                if nm ==
                'bibtex:pages':
                    txt = re.sub(r'--', '-', txt)
                elif nm == 'someothername':
                    # etwas anderes
                    tun pass
                elif nm ==
                    'stillanother': #
                    usw.
                    Pass

                return txt
            def wrapup(self,
                metaheaders): pass
```

Wenn die Methode "metafix()" definiert ist, wird sie für jedes Metadatenfeld aufgerufen. Für jedes PDF-Dokument wird ein neues MetaFixer-Objekt erstellt (damit das Objekt seinen Zustand beibehalten kann, um z.B. doppelte Werte zu eliminieren). Wenn die Methode "wrapup()" definiert ist, wird sie am Ende der Verarbeitung von XMP-Feldern mit den gesamten Metadaten als Parameter in Form eines Arrays von '(nm, val)'-Paaren aufgerufen, was einen alternativen Ansatz für das Bearbeiten oder Hinzufügen/Löschen von Feldern ermöglicht.

2.8.2 PDF-Anhängen

Wenn pdftk installiert ist und die Konfigurationsvariable pdfattach gesetzt ist, versucht der PDF Input Handler, PDF-Anhänge für die Indizierung als Unterdokumente der PDF-Datei zu extrahieren. Dies ist standardmäßig deaktiviert, da es die PDF-Indizierung ein wenig verlangsamt, selbst wenn kein einziger Anhang gefunden wird (PDF-Anhänge sind meiner Erfahrung nach ungewöhnlich).

2.9 Recoll und OCR

Dies ist neu in Recoll 1.26.5. Ältere Versionen hatten eine eingeschränktere, nicht zwischenspeichernde Möglichkeit, ein externes OCR-Programm im PDF-Handler auszuführen. Die neue Funktion hat die folgenden Merkmale:

- Die OCR-Ausgabe wird zwischengespeichert und in separaten Dateien abgelegt. Die Zwischenspeicherung basiert letztlich auf einem Hash-Wert des ursprünglichen Dateiinhalts, so dass sie gegen Dateiumbenennungen immun ist. Eine erste pfadbasierte Schicht gewährleistet einen schnellen Betrieb für unveränderte (nicht verschobene) Dateien, und der Datenhash (der immer noch um Größenordnungen schneller ist als OCR) wird nur dann neu berechnet, wenn die Datei verschoben wurde. OCR wird nur durchgeführt, wenn die Datei zuvor nicht verarbeitet wurde oder sich geändert hat.
- Die Unterstützung für ein bestimmtes Programm ist in einem einfachen Python-Modul implementiert. Es sollte einfach sein, Unterstützung für jede OCR-Engine hinzuzufügen, die über die Befehlszeile ausgeführt werden kann.
- Zunächst gibt es Module für Tesseract (Linux und Windows) und ABBYY FineReader (Linux, getestet mit Version 11). ABBYY FineReader ist ein kommerzielles, quelloffenes Programm, das aber manchmal besser funktioniert als Tesseract.
- Die OCR wird derzeit nur vom PDF-Handler aufgerufen, aber es sollte kein Problem sein, sie für andere Bildtypen zu verwenden.

Um diese Funktion zu aktivieren, müssen Sie eine der unterstützten OCR-Anwendungen (Tesseract oder ABBYY) installieren, OCR im PDF-Handler aktivieren und Recoll mitteilen, wo sich der entsprechende Befehl befindet. Die letzten Teile werden durch das Setzen von Konfigurationsvariablen erledigt. Siehe den entsprechenden Abschnitt. Alle Parameter können in

Unterverzeichnissen über den üblichen Hauptkonfigurationsmechanismus (Pfadabschnitte) lokalisiert werden.

2.10 Periodische Indizierung

2.10.1 Ausführen des Indexers

Das Programm **recollindex** führt Indexaktualisierungen durch. Sie können es entweder über die Kommandozeile oder über das Menü Datei im GUI-Programm **recoll** starten. Wenn Sie es von der GUI aus starten, wird die Indizierung mit derselben Konfiguration durchgeführt, mit der **recoll** gestartet wurde. Wenn es von der Kommandozeile aus gestartet wird, verwendet **recollindex** die Variable `RECOLL_CONFDIR` oder akzeptiert die Option `-c` *`confdir`*, um ein anderes Konfigurationsverzeichnis als das Standardverzeichnis anzugeben.

Wenn das Programm **recoll** beim Start keinen Index vorfindet, beginnt es automatisch mit der Indizierung (es sei denn, der Vorgang wird abgebrochen).

Das GUI-Menü Datei enthält Einträge zum Starten oder Stoppen der aktuellen Indizierung. Wenn die Indizierung gerade nicht läuft, haben Sie die Wahl zwischen Index aktualisieren und Index neu aufbauen. Bei der ersten Option werden nur geänderte Dateien verarbeitet, bei der zweiten wird der Index vor dem Start gelöscht, so dass alle Dateien verarbeitet werden.

Unter Linux und Windows kann die grafische Benutzeroberfläche zur Verwaltung des Indizierungsvorgangs verwendet werden. Das Anhalten des Indexers kann über die
rufen Sie den Menüeintrag GUI File → Stop Indexing **auf**.

Unter Linux kann der Indexierungsprozess von **recollindex** durch Senden eines Interrupt- (Ctrl-C, SIGINT) oder Terminate-Signals (SIGTERM) unterbrochen werden.

Wenn **recollindex** gestoppt wird, kann einige Zeit vergehen, bevor es beendet wird, da es den Index ordnungsgemäß leeren und schließen muss.

Nach einer Unterbrechung ist der Index etwas inkonsistent, da einige Operationen, die normalerweise am Ende des Indizierungsdurchlaufs durchgeführt werden, übersprungen wurden (z. B. sind die Stemming- und Rechtschreibdatenbanken nicht vorhanden oder veraltet). Sie müssen die Indizierung einfach zu einem späteren Zeitpunkt neu starten, um die Konsistenz wiederherzustellen. Die Indizierung wird am Unterbrechungspunkt neu gestartet (der gesamte Dateibaum wird durchlaufen, aber Dateien, die bis zur Unterbrechung indiziert wurden und für die der Index noch aktuell ist, müssen nicht neu indiziert werden).

2.10.2 recollindex-Befehlszeile

recollindex hat viele Optionen, die in seiner Handbuchseite aufgelistet sind. Hier werden nur einige davon beschrieben.

Mit der Option `-z` wird der Index beim Start zurückgesetzt. Dies ist fast dasselbe wie das Zerstören der Indexdateien (mit dem Unterschied, dass die Xapian-Formatversion nicht verändert wird).

Mit der Option `-Z` wird die Aktualisierung aller Dokumente erzwungen, ohne dass der Index vorher zurückgesetzt wird. Dies hat nicht den Aspekt des "sauberen Starts" von
`-z`, aber der Vorteil ist, dass der Index für Abfragen verfügbar bleibt, während er neu aufgebaut wird, was ein erheblicher Vorteil sein kann, wenn er sehr groß ist (manche Installationen brauchen Tage für einen vollständigen Indexneuaufbau).

Die Option `-k` erzwingt die Wiederholung der Indizierung von Dateien, die zuvor nicht indiziert werden konnten, z. B. weil ein Hilfsprogramm fehlt.

Von besonderem Interesse sind vielleicht auch die Optionen `-i` und `-f`. `-i` erlaubt die Indizierung einer expliziten Liste von Dateien (die als Kommandozeilenparameter angegeben oder auf `stdin` gelesen werden). Die Option `-f` weist **recollindex** an, Dateiauswahlparameter aus der Konfiguration zu ignorieren. Zusammen ermöglichen diese Optionen die Erstellung eines benutzerdefinierten Dateiauswahlprozesses für einen Bereich des Dateisystems, indem das oberste Verzeichnis zur skippedPaths-Liste hinzugefügt und eine geeignete Dateiauswahlmethode verwendet wird, um die Dateiliste zu erstellen, die an **recollindex** `-if` `übergeben wird`. Triviales Beispiel:

```
find . -name indexable.txt -print | recollindex -if
```

recollindex `-i` steigt nicht in die als Parameter angegebenen Unterverzeichnisse hinab, sondern fügt sie lediglich als Indexeinträge hinzu. Es ist Aufgabe der externen Dateiauswahlmethode, die vollständige Dateiliste zu erstellen.

2.10.3 Linux: cron zur Automatisierung der Indizierung verwenden

Die gängigste Art, die Indizierung einzurichten, besteht darin, sie jede Nacht von einem Cron-Task ausführen zu lassen. Zum Beispiel die folgende `crontab`
Die Eingabe würde jeden Tag um 3:30 Uhr erfolgen (vorausgesetzt, **recollindex** befindet sich in Ihrem PATH):

```
303  * * * recollindex > /some/tmp/dir/recolltrace 2>&1
```

Oder die Verwendung von **Anacron**:

```
115  su mylogin -c "recollindex recollindex > /tmp/rcltraceme 2>&1"
```

Die GUI von Recoll verfügt über Dialoge zur Verwaltung von crontab-Einträgen für **recollindex**. Sie können sie über das Menü Einstellungen Indizierungszeitplan erreichen. Sie funktionieren nur mit dem guten alten **Cron** und bieten keinen Zugriff auf alle Funktionen der Cron-Planung. Über das Tool erstellte Einträge werden mit einer RCLCRON_RCLINDEX=-Markierung versehen, damit das Tool weiß, welche Einträge dazu gehören. Als Nebeneffekt wird dadurch eine Umgebungsvariable für den Prozess gesetzt, die aber eigentlich nicht verwendet wird, sondern nur eine Markierung ist.

Der übliche Befehl zum Bearbeiten Ihrer `crontab` ist **crontab** `-e` (damit wird normalerweise der vi-Editor zum Bearbeiten der Datei gestartet). Möglicherweise stehen Ihnen auf Ihrem System komplexere Werkzeuge zur Verfügung.

Bitte beachten Sie, dass es Unterschiede zwischen Ihrer gewohnten interaktiven Kommandozeilenumgebung und derjenigen, die von crontab-Befehlen gesehen wird, geben kann. Insbesondere die PATH-Variable kann von Bedeutung sein. Bitte informieren Sie sich in den crontab-Handbuchseiten über mögliche Probleme.

2.11 Unix-ähnliche Systeme: Indizierung in Echtzeit

Die Überwachung/Indizierung in Echtzeit erfolgt durch Starten des Befehls **recollindex** `-m`. Mit dieser Option löst sich **recollindex** vom Terminal und wird zu einem Daemon, der permanent Dateiänderungen überwacht und den Index aktualisiert.

In dieser Situation stellt das Menü Datei der grafischen Benutzeroberfläche **von recoll** zwei Operationen zur Verfügung: Anhalten und Inkrementellen Durchlauf auslösen.

Inkrementellen Durchlauf auslösen hat den gleichen Effekt wie ein Neustart des Indexierers und führt zu einem vollständigen Durchlauf des indizierten Bereichs, wobei die geänderten Dateien verarbeitet werden, und schaltet dann auf Überwachung um. Dies ist nur bedingt nützlich, vielleicht in Fällen, in denen der Indexer so konfiguriert ist, dass er Aktualisierungen verzögert oder einen sofortigen Neuaufbau der Stemming- und Phonetikdaten erzwingt, die nur in Intervallen vom Echtzeit-Indexer verarbeitet werden.

Zwar ist es praktisch, dass die Daten in Echtzeit indiziert werden, doch kann eine wiederholte Indizierung das System erheblich belasten, wenn sich Dateien wie E-Mail-Ordner ändern. Auch die Überwachung großer Dateibäume allein beansprucht die Systemressourcen erheblich. Wenn Ihr System knapp an Ressourcen ist, sollten Sie diese Funktion wahrscheinlich nicht aktivieren. Eine periodische Indizierung ist in den meisten Fällen ausreichend.

Ab Recoll 1.24 können Sie die Konfigurationsvariable monitordirs setzen, um festzulegen, dass nur eine Teilmenge Ihrer indizierten Dateien für die sofortige Indizierung überwacht wird. In dieser Situation kann ein inkrementeller Durchlauf des gesamten Baums entweder durch einen Neustart des Indexers oder durch die Ausführung von **recollindex** ausgelöst werden, was den laufenden Prozess benachrichtigt. Die recoll-GUI hat auch einen Menüeintrag für diesen Vorgang.

2.11.1 Automatischer Daemon-Start mit systemd

Die Installation enthält zwei Beispieldateien (in `share/recoll/examples`) zum Starten des Indexierungs-Daemons mit systemd.

`recollindex.service` wird verwendet, um **recollindex** als Benutzerdienst zu starten. Der Indexer wird gestartet, wenn sich der Benutzer anmeldet und läuft, solange eine Sitzung für ihn geöffnet ist.

`recollindex@.service` ist ein Vorlagedienst, der zum Starten des Indexers beim Booten verwendet wird und als ein bestimmter Benutzer läuft. Er kann nützlich sein, wenn die Textsuche als gemeinsam genutzter Dienst ausgeführt wird (z. B. wenn Benutzer über das WEB UI darauf zugreifen).

Wenn sie so konfiguriert sind, sollten die Unit-Dateien in den systemd-Standardpfaden Ihres Systems installiert sein (normalerweise `/usr/lib/systemd/system/` und `/usr/lib/systemd/user/`). Falls nicht, müssen Sie die Dateien dorthin kopieren, bevor Sie den Dienst starten.

Wenn die Unit-Dateien am richtigen Ort installiert sind, kann die User Unit mit den folgenden Befehlen gestartet werden:

```
systemctl --user daemon-reload
systemctl --user enable --now recollindex.service
```

Die System-Unit-Datei kann für einen bestimmten Benutzer aktiviert werden, indem man sie als root ausführt:

```
systemctl daemon-reload
systemctl enable --now recollindex@username.service
```

(Anstelle des `Benutzernamens` sollte natürlich ein gültiger Benutzername verwendet werden).

2.11.2 Automatischer Start des Daemons aus der Desktop-Sitzung

Unter KDE, Gnome und einigen anderen Desktop-Umgebungen kann der Daemon automatisch gestartet werden, wenn Sie sich anmelden, indem Sie eine Desktop-Datei im Verzeichnis `~/.config/autostart` erstellen. Dies kann von der Recoll-GUI für Sie erledigt werden. Benutzen Sie die Preferences->Indizierung Zeitplan.

Bei älteren X11-Konfigurationen wird der Daemon normalerweise als Teil des Skripts der Benutzersitzung gestartet.

Das Skript `rclmon.sh` kann verwendet werden, um den Daemon einfach zu starten und zu stoppen. Es befindet sich im Verzeichnis `examples` (typischerweise `/usr/local/[share/]recoll/examples`).

Eine gute alte xdm-basierte Sitzung könnte zum Beispiel ein `.xsession`-Skript mit den folgenden Zeilen am Ende haben:

```
recollconf=$HOME/.recoll-home
recolldata /usr/local/share/recoll
RECOLL_CONFDIR=$recollconf $recolldata/examples/rclmon.sh start
fvwm
```

Der Indexierungs-Daemon wird gestartet, dann der Window-Manager, auf den die Sitzung wartet.
Standardmäßig überwacht der Indizierungs-Daemon den Zustand der X11-Sitzung und beendet sich, wenn er fertig ist; es ist nicht notwendig, ihn explizit zu beenden. (Die Überwachung des X11-Servers kann mit der Option `-x` für **recollindex** deaktiviert werden).

Wenn Sie den Daemon vollständig aus einer X11-Sitzung heraus verwenden, müssen Sie die Option `-x` hinzufügen, um die Überwachung von X11-Sitzungen zu deaktivieren (sonst wird der Daemon nicht gestartet).

2.11.3 Verschiedene Details

Standardmäßig werden die Meldungen des Indizierungsdämons in dieselbe Datei wie die der interaktiven Befehle (`logfilename`) gesendet. Sie können dies ändern, indem Sie die Konfigurationsparameter `daemlogfilename` und `daemloglevel` setzen. Außerdem wird die Protokolldatei nur abgeschnitten, wenn der Daemon startet. Wenn der Daemon permanent läuft, kann die Protokolldatei je nach Protokollstufe ziemlich groß werden.

Erhöhung der Ressourcen für inotify Auf Linux-Systemen kann die Überwachung eines großen Baums eine Erhöhung der für inotify verfügbaren Ressourcen erfordern, die normalerweise in `/etc/sysctl.conf` definiert sind.

```
###
inotify
###    /proc/sys/fs/inotify/max_queued_events   - 16384
Katz   /proc/sys/fs/inotify/max_user_instances    - 128
e   #  /proc/sys/fs/inotify/max user watches    - 16384
Katz
# We#hseln zu:
#katz
#fs#inotify.max_queued_events=32768
fs.inotify.max_user_instances=256
fs.inotify.max_user_watches=32768
```

Insbesondere müssen Sie Ihren Baum beschneiden oder den Wert für `max_user_watches` anpassen, wenn die Indizierung mit einer Meldung über errno `ENOSPC` (28) von `inotify_add_watch` beendet wird.

Verlangsamung der Re-Indizierungsrate für sich schnell ändernde Dateien Bei der Verwendung des Echtzeit-Monitors kann es vorkommen, dass einige Dateien indiziert werden müssen, sich aber so oft ändern, dass sie eine übermäßige Belastung für das System darstellen. Recoll bietet eine Konfigurationsoption, um die Mindestzeit festzulegen, vor der eine Datei, die durch ein Wildcard-Muster spezifiziert ist, nicht neu indiziert werden kann. Siehe den Parameter `mondelaypatterns` im Abschnitt Konfiguration.

Kapitel 3

Suche auf

3.1 Einführung

Antworten auf spezifische Anfragen zu erhalten, ist natürlich der Sinn von Recoll. Die mehrfach vorhandenen Schnittstellen verstehen immer einfache Abfragen, die aus einem oder mehreren Wörtern bestehen, und liefern in den meisten Fällen entsprechende Ergebnisse.

Um das Beste aus Recoll herauszuholen, lohnt es sich jedoch zu verstehen, wie es Ihre Eingaben verarbeitet. Es gibt fünf verschiedene Modi:

- Im Modus `Alle Begriffe` sucht Recoll nach Dokumenten, die alle von Ihnen eingegebenen Begriffe enthalten.
- Der Abfragesprachmodus verhält sich wie `Alle Begriffe`, wenn keine speziellen Eingaben gemacht werden, aber er kann auch viel mehr. Dies ist der beste Modus, um das Beste aus Recoll herauszuholen. Er ist von allen möglichen Schnittstellen aus nutzbar (GUI, Kommandozeile, WEB UI, ...) und wird hier beschrieben.
- Im Modus "`Beliebiger Begriff`" sucht Recoll nach Dokumenten, die alle von Ihnen eingegebenen Begriffe enthalten, wobei diejenigen bevorzugt werden, die mehr Begriffe enthalten.
- Im Dateinamen-Modus findet Recoll nur Dateinamen, nicht den Inhalt. Die Verwendung einer kleinen Teilmenge des Indexes ermöglicht Dinge wie linksseitige Platzhalter ohne Leistungsprobleme und kann manchmal nützlich sein.
- Der `erweiterte` Suchmodus der grafischen Benutzeroberfläche ist eigentlich nicht leistungsfähiger als die Abfragesprache, aber er hilft Ihnen bei der Erstellung komplexer Abfragen, ohne dass Sie sich die Sprache merken müssen, und vermeidet jegliche Interpretationszweideutigkeit, da er den Parser für Benutzereingaben umgeht.

Diese fünf Eingabemodi werden von den verschiedenen Benutzeroberflächen unterstützt, die in den folgenden Abschnitten beschrieben werden.

3.2 Suche mit der grafischen Benutzeroberfläche Qt

Das Programm **recoll** bietet die Hauptbenutzeroberfläche für die Suche. Es basiert auf der Qt-Bibliothek.

recoll hat zwei Suchoberflächen:

- Die einfache Suche (die Standardeinstellung auf dem Hauptbildschirm) hat ein einziges Eingabefeld, in das Sie mehrere Wörter eingeben können.
- Die erweiterte Suche (auf die Sie über das Menü "Extras" oder das Symbol in der Symbolleiste zugreifen können) verfügt über mehrere Eingabefelder, mit denen Sie eine logische Bedingung erstellen können, mit zusätzlicher Filterung nach Dateityp, Speicherort im Dateisystem, Änderungsdatum und Größe.

In den meisten Fällen können Sie die Begriffe so eingeben, wie Sie sie denken, auch wenn sie eingebettete Satzzeichen oder andere nicht-textuelle Zeichen enthalten (z. B. kann Recoll Dinge wie E-Mail-Adressen verarbeiten).

Vor allem bei ostasiatischen Sprachen (Chinesisch, Japanisch, Koreanisch) sollten Sie den Text anders eingeben, als er gedruckt wird. Wörter, die aus einzelnen oder mehreren Zeichen bestehen, sollten in diesem Fall durch Leerzeichen getrennt

eingegeben werden (normalerweise werden sie ohne Leerzeichen gedruckt).

Einige Suchvorgänge können recht komplex sein, und Sie möchten sie vielleicht später wiederverwenden, vielleicht mit einigen Anpassungen. Recoll kann Abfragen speichern und wiederherstellen. Siehe Speichern und Wiederherstellen von Suchanfragen.

3.2.1 Einfache Suche

1. Starten Sie das Programm **recoll**.
2. Wählen Sie eventuell einen Suchmodus: Beliebiger Begriff, Alle Begriffe, Dateiname oder Abfragesprache.
3. Geben Sie den/die Suchbegriff(e) in das Textfeld am oberen Rand des Fensters ein.
4. Klicken Sie auf die Schaltfläche Suchen oder drücken Sie die Eingabetaste, um die Suche zu starten.

Der anfängliche Standard-Suchmodus ist Abfragesprache. Ohne spezielle Anweisungen wird nach Dokumenten gesucht, die alle Suchbegriffe enthalten (die mit mehr Begriffen erhalten bessere Ergebnisse), genau wie im Modus Alle Begriffe. Jeder Begriff sucht nach Dokumenten, in denen mindestens einer der Begriffe vorkommt. Dateiname sucht ausschließlich nach Dateinamen, nicht nach Inhalten

In allen Suchmodi können Begriffe mit Platzhaltern (`*`, `?` , `[]`) erweitert werden. Weitere Einzelheiten finden Sie im Abschnitt über Wildcards.

In allen Modi außer "Dateiname" können Sie nach exakten Phrasen (benachbarte Wörter in einer bestimmten Reihenfolge) suchen, indem Sie die Eingabe in doppelte Anführungszeichen setzen. Beispiel: `"virtuelle Realität"`.

Die Funktionen der Abfragesprache werden in einem separaten Abschnitt beschrieben.

Wenn Sie einen Index ohne Groß- und Kleinschreibung verwenden (die Standardeinstellung), hat die Groß- und Kleinschreibung keinen Einfluss auf die Suche, außer dass Sie die Stammerweiterung für jeden Begriff deaktivieren können, indem Sie ihn groß schreiben. D.h.: eine Suche nach `"floor"` sucht normalerweise auch nach `"flooring"`, `"floored"` usw., aber eine Suche nach `"Floor"` sucht nur nach `"floor"`, egal in welcher Groß- und Kleinschreibung. Stemming kann auch global in den Einstellungen deaktiviert werden. Bei der Verwendung eines Rohindexes sind die Regeln etwas komplizierter.

Recoll merkt sich die letzten Suchvorgänge, die Sie durchgeführt haben. Sie können direkt auf den Suchverlauf zugreifen, indem Sie auf die Uhr-Taste rechts neben dem Sucheintrag klicken, wenn dieser leer ist. Andernfalls wird die Historie zur Vervollständigung des Eintrags verwendet (siehe unten). Es werden nur die Suchtexte gespeichert, nicht der Modus (alle/beliebig/Dateiname).

Während der Texteingabe im Suchbereich zeigt **recoll** mögliche Vervollständigungen an, gefiltert aus dem Verlauf und den Index-Suchbegriffen. Dies kann mit einer Option in den GUI-Einstellungen deaktiviert werden.

Ein Doppelklick auf ein Wort in der Ergebnisliste oder in einem Vorschaufenster fügt es in das Eingabefeld für die einfache Suche ein.

Sie können jeden beliebigen Text ausschneiden und in das Suchfeld Alle Begriffe oder Beliebiger Begriff einfügen, auch Satzzeichen und Zeilenumbrüche - mit Ausnahme von Platzhalterzeichen (einzelne `?` Zeichen sind in Ordnung). Recoll wird den Text verarbeiten und eine sinnvolle Suche erstellen. Dies unterscheidet diesen Modus am meisten von dem der Abfragesprache, bei dem Sie auf die Syntax achten müssen.

Für komplexere Suchen können Sie das Dialogfeld WerkzeugeErweiterte Suche verwenden.

Der Suchmodus Dateinamen sucht speziell nach Dateinamen. Der Sinn einer separaten Dateinamensuche besteht darin, dass die Platzhalterexpansion auf einer kleinen Teilmenge des Index effizienter durchgeführt werden kann (was Platzhalter auf der linken Seite von Begriffen ohne übermäßige Kosten ermöglicht). Wichtig zu wissen:

- Leerzeichen im Eintrag sollten mit Leerzeichen im Dateinamen übereinstimmen und werden nicht besonders behandelt.
- Bei der Suche werden Groß- und Kleinschreibung sowie Akzente nicht berücksichtigt, unabhängig von der Art des Indexes.
- Ein Eintrag ohne Platzhalter und ohne Großbuchstaben wird vorangestellt und mit '*' angehängt (z. B.: *`etc`* -> *`*etc*`*, aber *`Etc`* -> *`etc`*).
- Wenn Sie einen großen Index haben (viele Dateien), können zu viele generische Fragmente zu ineffizienten Suchen führen.

3.2.2 Die Ergebnisliste

Nach dem Start einer Suche wird sofort eine Liste der Ergebnisse im Hauptfenster angezeigt.

Standardmäßig wird die Liste der Dokumente in der Reihenfolge ihrer Relevanz angezeigt (d. h. wie gut das System die Übereinstimmung des Dokuments mit der Suchanfrage einschätzt). Sie können das Ergebnis nach auf- oder absteigendem

Datum sortieren, indem Sie die vertikalen Pfeile in der Symbolleiste verwenden.

Wenn Sie auf den Link `Vorschau` für einen Eintrag klicken, wird ein internes Vorschaufenster für das Dokument geöffnet. Weitere Klicks auf die `Vorschau` für dieselbe Suche öffnen Registerkarten im bestehenden Vorschaufenster. Mit **Umschalt+Klick** können Sie die Erstellung einer weiteren Vorschau erzwingen

was nützlich sein kann, um die Dokumente nebeneinander zu sehen. (Sie können auch die aufeinanderfolgenden Ergebnisse in einem einzigen Vorschaufenster durchsuchen, indem Sie **Umschalt+Pfeil-nach-oben/unten** in das Fenster eingeben).

Wenn Sie auf den Link `Öffnen` klicken, wird ein externer Viewer für das Dokument gestartet. Standardmäßig lässt Recoll den Desktop die passende Anwendung für die meisten Dokumenttypen wählen. Siehe weiter für die Anpassung der Anwendungen.

Sie können auf den Link `Abfragedetails` oben auf der Ergebnisseite klicken, um die tatsächlich durchgeführte Abfrage nach der Stammerweiterung und anderen Verarbeitungen zu sehen.

Ein Doppelklick auf ein beliebiges Wort in der Ergebnisliste oder einem Vorschaufenster fügt es in den einfachen Suchtext ein.

Die Ergebnisliste ist in Seiten unterteilt (deren Größe Sie in den Einstellungen ändern können). Verwenden Sie die Pfeilschaltflächen in der Symbolleiste oder die Links am unteren Rand der Seite, um die Ergebnisse zu durchsuchen.

Die Links `Vorschau` und Bearbeiten `öffnen` sind möglicherweise nicht für alle Einträge vorhanden, was bedeutet, dass Recoll keine konfigurierte Möglichkeit hat, eine Vorschau eines bestimmten Dateityps anzuzeigen (der nur über den Namen indiziert wurde), oder keinen konfigurierten externen Editor für den Dateityp hat. Dies kann manchmal einfach durch die Anpassung der Konfigurationsdateien mimemap und mimeview korrigiert werden (letztere kann über den Benutzereinstellungen-Dialog geändert werden).

Das Format der Einträge in der Ergebnisliste ist vollständig konfigurierbar, indem Sie den Einstellungsdialog zum Bearbeiten eines HTML-Fragments verwenden.

3.2.2.1 Anpassung der Anwendungen

Standardmäßig überlässt Recoll dem Desktop die Wahl, welche Anwendung zum Öffnen eines bestimmten Dokuments verwendet werden soll, mit Ausnahmen.

Die Details dieses Verhaltens können mit der Konfiguration Preferences GUI User interface→Choose editor applications oder durch Bearbeiten der Konfigurationsdatei von mimeview angepasst werden.

Wenn das Kontrollkästchen Desktop-Voreinstellungen verwenden oben im Dialogfeld aktiviert ist, wird in der Regel die Desktop-Vorgabe verwendet, aber es gibt eine kleine Liste von Ausnahmen für MIME-Typen, bei denen die Recoll-Auswahl die Desktop-Vorgabe überschreiben sollte. Dies sind Anwendungen, die gut in Recoll integriert sind, z. B. unter Linux evince für die Anzeige von PDF- und Postscript-Dateien, da es das Öffnen des Dokuments auf einer bestimmten Seite und die Übergabe eines Suchstrings als Argument unterstützt. Über den Dialog können Sie Dokumenttypen zu den Ausnahmen hinzufügen oder entfernen.

Wenn Sie die Auswahl der Anwendungen vollständig anpassen möchten, können Sie die Option Desktop-Voreinstellungen verwenden deaktivieren. In diesem Fall werden die von Recoll vordefinierten Anwendungen verwendet, die für jeden Dokumententyp geändert werden können. Dies ist wahrscheinlich in den meisten Fällen nicht der bequemste Ansatz.

In allen Fällen akzeptiert das Dialogfeld für die Anwendungsauswahl im oberen Abschnitt eine Mehrfachauswahl von MIME-Typen und lässt Sie im unteren Abschnitt festlegen, wie diese verarbeitet werden sollen. In den meisten Fällen werden Sie `%f` als Platzhalter verwenden, der in der Befehlszeile der Anwendung durch den Dateinamen ersetzt wird.

Sie können die Auswahl der Anwendungen auch ändern, indem Sie die Konfigurationsdatei von mimeview bearbeiten, wenn Sie dies für bequemer halten.

Unter Unix-ähnlichen Systemen verfügt jeder Eintrag in der Ergebnisliste auch über ein Rechtsklick-Menü mit einem Eintrag Öffnen mit. Damit können Sie von Fall zu Fall eine Anwendung aus der Liste derjenigen auswählen, die auf dem Desktop für den MIME-Typ des Dokuments registriert sind.

3.2.2.2 Keine Ergebnisse: die Rechtschreibvorschläge

Wenn eine Suche zu keinem Ergebnis führt und das Aspell-Wörterbuch konfiguriert ist, versucht Recoll, die Suchbegriffe auf Rechtschreibfehler zu prüfen und schlägt Listen mit Ersatzbegriffen vor. Wenn Sie auf einen der Vorschläge klicken, wird das Wort ersetzt und die Suche neu gestartet. Sie können eine der Modifikatortasten (Strg, Shift, etc.) gedrückt halten, während Sie klicken, wenn Sie lieber auf dem Vorschlagsbildschirm bleiben möchten, weil mehrere Begriffe ersetzt werden müssen.

3.2.2.3 Das Rechtsklickmenü der Ergebnisliste

Neben den Links zur Vorschau und zum Bearbeiten können Sie ein Popup-Menü anzeigen, indem Sie mit der rechten Maustaste auf einen Absatz in der Ergebnisliste klicken. Dieses Menü enthält die folgenden Einträge:

- Vorschau

- Öffnen Sie
- Öffnen mit
- Skript ausführen
- Dateiname kopieren
- Url kopieren
- In Datei speichern
- Ähnliches finden
- Vorschau des übergeordneten Dokuments
- Übergeordnetes Dokument öffnen
- Fenster Schnipsel öffnen

Die Einträge Vorschau und Öffnen haben die gleiche Funktion wie die entsprechenden Links.

Öffnen mit (Unix-ähnliche Systeme) lässt Sie das Dokument mit einer der Anwendungen öffnen, die den MIME-Typ des Dokuments verarbeiten können (die Informationen stammen aus den .desktop-Dateien in `/usr/share/applications`).

Run Script (Unix-ähnliche Systeme) ermöglicht das Starten eines beliebigen Befehls in der Ergebnisdatei. Es erscheint nur für Ergebnisse, die Top-Level-Dateien sind. Für eine detailliertere Beschreibung siehe weiter.

Mit den Befehlen Dateiname kopieren und Url kopieren werden die entsprechenden Daten in die Zwischenablage kopiert, damit sie später eingefügt werden können.

In Datei speichern ermöglicht das Speichern des Inhalts eines Ergebnisdokuments in einer ausgewählten Datei. Dieser Eintrag erscheint nur, wenn das Dokument nicht zu einer bestehenden Datei gehört, sondern ein Unterdokument innerhalb einer solchen Datei ist (z.B. ein E-Mail-Anhang). Dies ist besonders nützlich, um Anhänge zu extrahieren, denen kein Editor zugeordnet ist.

Die Einträge Öffnen/Vorschau übergeordnetes Dokument ermöglichen die Arbeit mit dem übergeordneten Dokument (z.B. die E-Mail-Nachricht, aus der ein Anhang stammt). Recoll ist manchmal nicht ganz genau, was es in diesem Bereich tun kann und was nicht. Zum Beispiel erscheint der Eintrag Übergeordnetes Dokument auch für eine E-Mail, die Teil einer mbox-Ordnerdatei ist, aber Sie können die mbox nicht wirklich anzeigen (es wird ein Fehlerdialog angezeigt, wenn Sie es versuchen).

Wenn es sich bei dem Dokument um eine Datei der obersten Ebene handelt, startet Open Parent den Standard-Dateimanager für das umschließende Dateisystemverzeichnis.

Der Eintrag Ähnliches suchen wählt eine Reihe von relevanten Begriffen aus dem aktuellen Dokument aus und gibt sie in das einfache Suchfeld ein. Sie können dann eine einfache Suche starten, mit einer guten Chance, Dokumente zu finden, die mit dem aktuellen Ergebnis verwandt sind. Ich kann mich an kein einziges Beispiel erinnern, in dem diese Funktion für mich tatsächlich nützlich war...

Der Eintrag Snippets-Fenster öffnen erscheint nur bei Dokumenten, die Seitenumbrüche unterstützen (typischerweise PDF, Postscript, DVI). Das Snippets-Fenster listet Auszüge aus dem Dokument auf, die um das Vorkommen von Suchbegriffen herum aufgenommen wurden, zusammen mit der entsprechenden Seitenzahl, als Links, die verwendet werden können, um den nativen Viewer auf der entsprechenden Seite zu starten. Wenn der Viewer dies unterstützt, wird auch seine Suchfunktion mit einem der Suchbegriffe eingeleitet.

3.2.3 Die Ergebnistabelle

Alternativ zur Ergebnisliste können die Ergebnisse auch in einer tabellenartigen Darstellung angezeigt werden. Sie können zu dieser Darstellung wechseln, indem Sie auf das tabellenartige Symbol in der Symbolleiste klicken (dies ist eine Umschaltfunktion, klicken Sie erneut, um die Liste wiederherzustellen).

Wenn Sie auf die Spaltenüberschriften klicken, können Sie nach den Werten in der Spalte sortieren. Sie können erneut klicken, um die Reihenfolge umzukehren, und das Rechtsklickmenü der Kopfzeile verwenden, um die Sortierung auf die Standard-Relevanzreihenfolge zurückzusetzen (Sie können dazu auch die Pfeile für die Sortierung nach Datum verwenden).

Sowohl in der Liste als auch in der Tabelle werden die gleichen Ergebnisse angezeigt. Die in der Tabelle eingestellte Sortierreihenfolge ist auch dann noch aktiv, wenn Sie zurück in den Listenmodus wechseln. Sie können zweimal auf einen Datumssortierpfeil klicken, um ihn von dort aus zurückzusetzen.

Das Rechtsklick-Menü der Kopfzeile ermöglicht das Hinzufügen oder Löschen von Spalten. Die Größe der Spalten kann verändert werden, und ihre Reihenfolge kann geändert werden (durch Ziehen). Alle Änderungen werden gespeichert, wenn Sie **recoll** verlassen.

Wenn Sie den Mauszeiger über eine Tabellenzeile bewegen, wird der Detailbereich am unteren Rand des Fensters mit den entsprechenden Werten aktualisiert. Sie können auf die Zeile klicken, um die Anzeige zu fixieren. Der untere Bereich entspricht einem Absatz der Ergebnisliste mit Links zum Starten einer Vorschau oder einer systemeigenen Anwendung und einem entsprechenden Rechtsklickmenü. Wenn Sie **Esc** (die Escape-Taste) drücken, wird die Anzeige wieder eingefroren.

3.2.4 Unix-ähnliche Systeme: Ausführung beliebiger Befehle auf Ergebnisdateien

Abgesehen von den Operationen Öffnen und Öffnen mit, die es ermöglichen, eine Anwendung auf einem Ergebnisdokument (oder einer temporären Kopie) auf der Grundlage seines MIME-Typs zu starten, ist es auch möglich, beliebige Befehle auf Ergebnissen auszuführen, die Dateien der obersten Ebene sind, indem man den Eintrag Skript ausführen im Popup-Menü der Ergebnisse verwendet.

Die Befehle, die im Untermenü Skript ausführen erscheinen, müssen durch `.desktop`-Dateien im Unterverzeichnis `scripts` des aktuellen Konfigurationsverzeichnisses definiert werden.

Nachfolgend ein Beispiel für eine `.desktop`-Datei, die z. B. `~/.recoll/scripts/myscript. desktop` heißen könnte (der genaue Dateiname innerhalb des Verzeichnisses ist irrelevant):

```
[Desktop-Eintrag]
Typ=Anwendung
Name=MyExecScript
Exec=/home/me/bin/tryscript
%F MimeType=*/*
```

Das Attribut `Name` definiert die Bezeichnung, die im Menü Skript ausführen angezeigt wird. Das Exec-Attribut definiert das auszuführende Programm, das natürlich kein Skript sein muss. Das MimeType-Attribut wird nicht verwendet, muss aber vorhanden sein.

Die auf diese Weise definierten Befehle können auch über Links innerhalb des Ergebnisabsatzes verwendet werden.

So könnte es beispielsweise sinnvoll sein, ein Skript zu schreiben, das das Dokument in den Papierkorb verschiebt und aus dem Recoll-Index löscht.

3.2.5 Unix-ähnliche Systeme: Anzeige von Miniaturbildern

Das Standardformat für die Einträge der Ergebnisliste und den Detailbereich der Ergebnistabelle zeigt ein Symbol für jedes Ergebnisdokument an. Das Symbol ist entweder ein allgemeines Symbol, das anhand des MIME-Typs bestimmt wird, oder ein Miniaturbild des Dokuments. Miniaturansichten werden nur angezeigt, wenn sie am Standard-Speicherort von freedesktop gefunden werden, wo sie normalerweise von einem Dateimanager erstellt wurden.

Recoll hat keine Möglichkeit, Miniaturansichten zu erstellen. Ein relativ einfacher Trick besteht darin, den Eintrag Übergeordnetes Dokument/Ordner öffnen im Popup-Menü der Ergebnisliste zu verwenden. Dies sollte ein Dateimanager-Fenster für das enthaltende Verzeichnis öffnen, das wiederum die Thumbnails erstellen sollte (abhängig von Ihren Einstellungen). Wenn Sie die Suche erneut starten, sollten die Miniaturansichten angezeigt werden.

Es gibt auch einige Hinweise zur Erstellung von Miniaturbildern in der Recoll-FAQ.

3.2.6 Das Vorschaufenster

Das Vorschaufenster öffnet sich, wenn Sie zum ersten Mal auf einen Vorschau-Link innerhalb der Ergebnisliste klicken.

Nachfolgende Vorschaubestellungen für eine bestimmte Suche öffnen neue Registerkarten im bestehenden Fenster (es sei denn, Sie halten beim Klicken die **Umschalttaste gedrückt**, dann wird ein neues Fenster für die nebeneinander liegende Ansicht geöffnet).

Wenn Sie eine weitere Suche starten und eine Vorschau anfordern, wird ein neues Vorschaufenster erstellt. Das alte Fenster bleibt geöffnet, bis Sie es schließen.

Sie können eine Vorschauregisterkarte schließen, indem Sie **Strg-W** (**Strg** + **W**) im Fenster eingeben. Wenn Sie die letzte Registerkarte schließen oder die Schaltfläche "Fenstermanager" oben im Rahmen verwenden, wird das Fenster ebenfalls geschlossen.

Sie können aufeinanderfolgende oder vorherige Dokumente aus der Ergebnisliste innerhalb eines Vorschauregisters anzeigen, indem Sie **Umschalt+Abwärts** oder **Umschalt+Auf** eingeben
(**Ab** und **Auf** sind die Pfeiltasten).

Ein Rechtsklick-Menü im Textbereich ermöglicht das Umschalten zwischen der Anzeige des Haupttextes und des Inhalts von Feldern, die mit dem Dokument verknüpft sind (z. B. Autor, Abstrakt usw.). Dies ist besonders dann nützlich, wenn der übereinstimmende Begriff nicht im Haupttext, sondern in einem der Felder vorkommt. Bei Bildern können Sie zwischen drei Anzeigen wählen: dem Bild selbst, den von **exiftool** extrahierten Bild-Metadaten und den Feldern, d. h. den im Index gespeicherten Metadaten.

Sie können den aktuellen Inhalt des Vorschaufensters ausdrucken, indem Sie **Strg-P** (**Strg** + **P**) im Fenstertext eingeben.

3.2.6.1 Suche innerhalb der Vorschau

Das Vorschaufenster verfügt über eine interne Suchfunktion, die größtenteils über das Panel am unteren Rand des Fensters gesteuert wird und in zwei Modi funktioniert: als klassische inkrementelle Suche des Editors, bei der wir nach dem im Eingabebereich eingegebenen Text suchen, oder als Möglichkeit, die Übereinstimmungen zwischen dem Dokument und der Recoll-Abfrage, die es gefunden hat, zu durchsuchen.

Inkrementelle Textsuche Die Vorschauregisterkarten verfügen über eine interne inkrementelle Suchfunktion. Sie starten die Suche entweder durch Eingabe eines / (Schrägstrich) oder **CTL-F** im Textbereich oder durch Klicken in das Textfeld Suchen nach: und Eingabe des Suchbegriffs. Mit den Schaltflächen Weiter und Zurück können Sie dann das nächste/vorherige Vorkommen suchen. Sie können auch **F3** in das Textfeld eingeben, um zum nächsten Vorkommen zu gelangen.

Wenn Sie einen Suchbegriff eingegeben haben und mit Strg-nach-oben/Strg-nach-unten durch die Ergebnisse blättern, wird die Suche für jedes nachfolgende Dokument gestartet. Wenn die Zeichenfolge gefunden wird, wird der Cursor auf das erste Vorkommen der Suchzeichenfolge gesetzt.

Durchlaufen der Übereinstimmungslisten Wenn der Eingabebereich leer ist, wenn Sie auf die Schaltflächen Weiter oder Zurück klicken, wird der Editor durchlaufen, um die nächste Übereinstimmung mit einem beliebigen Suchbegriff anzuzeigen (der nächste markierte Bereich). Wenn Sie eine Suchgruppe aus der Dropdown-Liste auswählen und auf Weiter oder Zurück klicken, wird die Trefferliste für diese Gruppe durchlaufen. Dies ist nicht dasselbe wie eine Textsuche, da die Trefferliste auch nicht exakte Übereinstimmungen enthält (z. B. aufgrund von Stemming oder Wildcards). Die Suche kehrt in den Textmodus zurück, sobald Sie den Eingabebereich bearbeiten.

3.2.7 Das Fenster Abfrage-Fragmente

Wenn Sie den Menüeintrag Extras Abfrage-Fragmente wählen, öffnet sich ein Fenster mit Options- und Kontrollkästchen, mit denen Sie die aktuelle Abfrage mit benutzerdefinierten Fragmenten der Abfragesprache filtern können. Dies kann nützlich sein, wenn Sie häufig wiederverwendbare Selektoren haben, die nicht von den Standard-Kategorie-Selektoren abgedeckt werden, z. B. das Filtern nach alternativen Verzeichnissen oder die Suche nach nur einer Kategorie von Dateien. In der Praxis werden die Abfragefragmente mit der aktuellen Abfrage als UND-Klausel verknüpft.

Der Inhalt des Fensters ist vollständig anpassbar und wird durch den Inhalt einer XML-Textdatei mit dem Namen `fragment-buttons.xml` definiert, die im aktuellen Indexkonfigurationsverzeichnis gesucht wird. Die mit Recoll mitgelieferte Beispieldatei enthält eine Reihe von Beispielfiltern. Diese wird automatisch in das Konfigurationsverzeichnis kopiert, wenn die Datei dort nicht vorhanden ist (z.B. `~/.recoll/fragment-buttons.xml` unter Linux und Mac OS, `$HOME/AppData/Local/Recoll` für Windows). Durch die Bearbeitung dieser Kopie können Sie das Tool für Ihre Bedürfnisse konfigurieren.

Hinweis

Die Datei `fragment-buttons.xml` hieß bis zur Version von Recoll `fragbuts.xml`. Dies1.31.0. wurde als zu nahe an einer Beleidigung für englische Muttersprachler angesehen, so dass die Datei umbenannt wurde. Eine vorhandene `fragbuts.xml` wird weiterhin verwendet, wenn `fragment-buttons.xml` nicht vorhanden ist. Es wird keine automatische Umbenennung vorgenommen.

Es folgt ein Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<fragbuttons version="1.0">

  <radiobuttons>
    <!-- Eigentlich nützlich: Einschluss der Ergebnisse der Web-Warteschlange umschalten -->
    <Fragbutton>
      <label>Web-Ergebnisse einbeziehen</label>
      <frag></frag>
    </fragbutton>

    <Fragbutton>
      <label>Web-Ergebnisse ausschließen</label>
      <frag>-rclbes:BGL</frag>
    </fragbutton>

    <Fragbutton>
```

```
      <label>Nur Web-Ergebnisse</label>
      <frag>rclbes:BGL</frag>
    </fragbutton>

  </radiobuttons>

  <Knöpfe>

    <Fragbutton>
      <label>Beispiel: Jahr 2010</label>
      <frag>date:2010-01-01/2010-12-31</frag>
    </fragbutton>

    <Fragbutton>
      <label>Beispiel: C++-Dateien</label>
      <frag>ext:cpp OR ext:cxx</frag>
    </fragbutton>

    <Fragbutton>
      <label>Beispiel: Mein großes Verzeichnis</label>
      <frag>dir:/mein/großes/Verzeichnis</frag>
    </fragbutton>

  </buttons>

</fragbuttons>
```

Jeder Abschnitt mit `Radiobuttons` oder `Schaltflächen` definiert eine Reihe von Checkbuttons oder Radiobuttons innerhalb des Fensters. Eine beliebige Anzahl von
`Tasten` können ausgewählt werden, aber die `Radiobuttons` in einer Zeile sind exklusiv.

Jeder fragbutton-Abschnitt definiert die Beschriftung einer Schaltfläche und das Abfragesprachenfragment, das (als AND-Filter) vor der Durchführung der Abfrage hinzugefügt wird, wenn die Schaltfläche aktiv ist.

Das Einzige, was Sie über XML wissen müssen, um diese Datei zu bearbeiten, ist, dass jedes öffnende Tag wie *<label>* durch ein schließendes Tag nach dem Wert ergänzt werden muss: *</label>*.

Normalerweise werden Sie die Datei mit einem normalen Texteditor bearbeiten, wie z. B. **vi** oder **notepad**. Ein Doppelklick auf die Datei in einem Dateimanager funktioniert möglicherweise nicht, da die Datei dann in der Regel in einem Webbrowser geöffnet wird, in dem Sie den Inhalt nicht ändern können.

3.2.8 Komplexe/erweiterte Suche

Das Dialogfeld für die erweiterte Suche hilft Ihnen bei der Erstellung komplexerer Abfragen, ohne dass Sie sich die Konstrukte der Suchsprache merken müssen. Er kann über das Menü "Extras" oder über die Hauptsymbolleiste geöffnet werden.

Recoll speichert einen Suchverlauf. Siehe Erweiterter

Suchverlauf. Der Dialog hat zwei Registerkarten:

1. Auf der ersten Registerkarte können Sie die zu suchenden Begriffe angeben und mehrere Klauseln spezifizieren, die zur Erstellung der Suche kombiniert werden.
2. Auf der zweiten Registerkarte können Sie die Ergebnisse nach Dateigröße, Änderungsdatum, MIME-Typ oder Speicherort filtern.

Klicken Sie auf die Schaltfläche Suche starten im Dialogfenster für die erweiterte Suche, oder geben Sie die **Eingabetaste** in ein beliebiges Textfeld ein, um die Suche zu starten. Die Schaltfläche im Hauptfenster führt immer eine einfache Suche durch.

Klicken Sie auf den Link `Abfragedetails anzeigen` am oberen Rand der Ergebnisseite, um die Abfrageerweiterung zu sehen.

3.2.8.1 Erweiterte Suche: die Registerkarte "Finden".

In diesem Teil des Dialogfelds können Sie eine Abfrage konstruieren, indem Sie mehrere Klauseln unterschiedlichen Typs kombinieren. Jedes Eingabefeld ist für die folgenden Modi konfigurierbar:

- Alle Begriffe.
- Jeder Begriff.
- Keiner der Begriffe.
- Phrase (genaue Begriffe in der Reihenfolge innerhalb eines einstellbaren Fensters).
- Nähe (Begriffe in beliebiger Reihenfolge innerhalb eines einstellbaren Fensters).
- Suche nach Dateinamen.

Zusätzliche Eingabefelder können durch Klicken auf die Schaltfläche Klausel hinzufügen erstellt werden.

Bei der Suche werden die nicht leeren Klauseln entweder mit einer UND- oder einer ODER-Verknüpfung kombiniert, je nach der links getroffenen Auswahl (Alle Klauseln oder Jede Klausel).

Bei allen Eingabearten außer "Phrase" und "Near" ist eine Mischung aus einzelnen Wörtern und in Anführungszeichen eingeschlossenen Phrasen zulässig. Stemming und Wildcard-Expansion werden wie bei der einfachen Suche durchgeführt.

Phrasen und Proximity-Suche Diese beiden Klauseln funktionieren auf ähnliche Weise, mit dem Unterschied, dass die Proximity-Suche den Wörtern keine Reihenfolge vorschreibt. In beiden Fällen kann eine einstellbare Anzahl von nicht übereinstimmenden Wörtern zwischen den gesuchten Wörtern akzeptiert werden (verwenden Sie den Zähler auf der linken Seite, um diese Anzahl einzustellen). Bei Phrasen ist die Standardanzahl Null (exakte Übereinstimmung). Für die Annäherung ist sie zehn (was bedeutet, dass zwei Suchbegriffe übereinstimmen würden, wenn sie innerhalb eines Fensters von zwölf Wörtern gefunden werden). Beispiele: Eine Phrasensuche nach `quick fox` mit einem Slack von 0findet `quick fox`, aber nicht `quick brown fox`. Mit einer Lücke von 1wird letzteres gefunden, aber nicht `quick fox`. Eine Umkreissuche nach `quick fox` mit der Standardphrase passt zu letzterem, und auch `ein Fuchs ist ein schlaues und schnelles Tier`.

3.2.8.2 Erweiterte Suche: die Registerkarte "Filter".

Dieser Teil des Dialogs hat mehrere Abschnitte, die es ermöglichen, die Ergebnisse einer Suche nach einer Reihe von Kriterien zu filtern

- Der erste Abschnitt ermöglicht die Filterung nach dem Datum der letzten Änderung. Sie können sowohl ein Mindest- als auch ein Höchstdatum angeben. Die Anfangswerte werden entsprechend den ältesten und neuesten Dokumenten im Index festgelegt.
- Im nächsten Abschnitt können die Ergebnisse nach der Dateigröße gefiltert werden. Es gibt zwei Einträge für die minimale und maximale Größe. Geben Sie Dezimalzahlen ein. Sie können Suffix-Multiplikatoren verwenden: `k/K`, `m/M`, `g/G`, `t/T` für 1E3, 1E6, 1E9 bzw. 1E12.
- Der nächste Abschnitt ermöglicht die Filterung der Ergebnisse nach MIME-Typen oder MIME-Kategorien (z. B. Medien/Text/Nachricht/etc.). Sie können die Typen zwischen zwei Feldern verschieben, um festzulegen, welche von der Suche ein- oder ausgeschlossen werden sollen.

  Der Zustand der Dateitypauswahl kann als Standard gespeichert werden (der Dateitypfilter wird beim Programmstart nicht aktiviert, aber die Listen befinden sich im wiederhergestellten Zustand).
- Der untere Abschnitt ermöglicht die Einschränkung der Suchergebnisse auf einen Teilbaum des indizierten Bereichs. Sie können das Kontrollkästchen Invertieren verwenden, um stattdessen nach Dateien zu suchen, die sich nicht im Teilbaum befinden. Wenn Sie die Verzeichnisfilterung häufig und für große Teilmengen des Dateisystems verwenden, sollten Sie stattdessen mehrere Indizes einrichten, da die Leistung dann möglicherweise besser ist.

  Sie können relative/teilweise Pfade für die Filterung verwenden. D.h. die Eingabe von `dirA/dirB` würde entweder `/dir1/dirA/dirB/meineDatei1` oder `/dir2/dirA/dirB/andere/meineDatei2` entsprechen.

3.2.8.3 Erweiterte Suchhistorie

Das Tool für die erweiterte Suche speichert die letzten 100 durchgeführten Suchen. Sie können die gespeicherten Suchvorgänge mit den Pfeiltasten nach oben und unten durchgehen, während der Tastaturfokus auf dem Dialogfeld für die erweiterte Suche liegt.

Der komplexe Suchverlauf kann zusammen mit dem einfachen Suchverlauf gelöscht werden, indem Sie den Menüeintrag Datei - Suchverlauf löschen wählen.

3.2.9 Der Begriff Entdeckerwerkzeug

Recoll verwaltet automatisch die Erweiterung von Suchbegriffen auf ihre Ableitungen (z. B. Plural/Singular, Verbbeugungen). Es gibt aber auch Fälle, in denen der genaue Suchbegriff nicht bekannt ist. Zum Beispiel können Sie sich nicht an die genaue Schreibweise erinnern oder kennen nur den Anfang des Namens.

Die Suche schlägt nur dann Ersatzbegriffe mit abweichender Schreibweise vor, wenn kein passendes Dokument gefunden wurde. In einigen Fällen sind sowohl richtige Schreibweisen als auch falsche Schreibweisen im Index vorhanden, und es kann interessant sein, explizit danach zu suchen.

Mit dem Termexplorer (der über das Symbol in der Symbolleiste oder über den Eintrag Termexplorer im Menü Extras aufgerufen wird) können Sie die vollständige Liste der Indexbegriffe durchsuchen. Es hat drei Betriebsarten:

Wildcard In dieser Betriebsart können Sie einen Suchstring mit Shell-ähnlichen Wildcards (*, ?, []) eingeben. d.h.: *xapi** würde alle Indexbegriffe anzeigen, die mit *xapi* beginnen. (Mehr über Wildcards hier).

Regulärer Ausdruck Dieser Modus akzeptiert einen regulären Ausdruck als Eingabe. Beispiel: *wort[0-9]+*. Der Ausdruck wird implizit am Anfang verankert. D.h.: *press* passt auf *pression*, aber nicht auf *expression*. Sie können *.*press* verwenden*, um letzteren zu finden, aber bedenken Sie, dass dies eine vollständige Suche nach Begriffen im Index auslöst, die recht lang sein kann.

Stammexpansion In diesem Modus wird die übliche Stammexpansion durchgeführt, die normalerweise als Teil der Benutzereingabeverarbeitung erfolgt. Als solcher ist er wahrscheinlich hauptsächlich zur Demonstration des Prozesses nützlich.

Rechtschreibung/Phonetisch In diesem Modus geben Sie den Begriff so ein, wie er Ihrer Meinung nach geschrieben wird, und Recoll wird sein Bestes tun, um Indexbegriffe zu finden, die wie Ihre Eingabe klingen. Dieser Modus verwendet die Aspell-Rechtschreibanwendung, die auf Ihrem System installiert sein muss, damit alles funktioniert (wenn Ihre Dokumente Nicht-Ascii-Zeichen enthalten, benötigt Recoll eine neuere Aspell-Version als für die 0.60UTF-8-Unterstützung). Die Sprache, die verwendet wird, um das Wörterbuch aus den Indexbegriffen zu erstellen (was am Ende eines Indexierungsdurchgangs geschieht), ist diejenige, die von Ihrer NLS-Umgebung definiert wird. Wenn die Sprachen verwechselt werden, passieren wahrscheinlich merkwürdige Dinge.

Index-Statistiken anzeigen Dies gibt eine lange Liste von langweiligen Zahlen über den Index aus

Dateien auflisten, die nicht indiziert werden konnten Hier werden die Dateien angezeigt, die Fehler verursacht haben, in der Regel weil **recollindex** ihr Format nicht in Text umwandeln konnte.

Beachten Sie, dass in Fällen, in denen Recoll den Anfang der zu suchenden Zeichenkette nicht kennt (z. B. bei einem Wildcard-Ausdruck wie **coll*), die Expansion recht lange dauern kann, da die gesamte Liste der Indexbegriffe verarbeitet werden muss. Die Expansion ist derzeit auf Ergebnisse10000 für Wildcards und reguläre Ausdrücke beschränkt. Es ist möglich, diese Begrenzung in der Konfigurationsdatei zu ändern.

Ein Doppelklick auf einen Begriff in der Ergebnisliste fügt ihn in das Eingabefeld der einfachen Suche ein. Sie können auch zwischen der Ergebnisliste und einem beliebigen Eingabefeld ausschneiden und einfügen (das Zeilenende wird dabei berücksichtigt).

3.2.10 Mehrere Indizes

Siehe den Abschnitt, der die Verwendung mehrerer Indizes beschreibt, für allgemeine Informationen. Hier werden nur die Aspekte beschrieben, die die Benutzeroberfläche von **recoll** betreffen.

Ein Recoll-Programm ist immer mit einem bestimmten Index verbunden, der aktualisiert wird, wenn er über das Menü "Datei" angefordert wird, aber es kann eine beliebige Anzahl von Recoll-Indizes für die Suche verwenden. Die externen Indizes können über die Registerkarte "Externe Indizes" im Einstellungsdialog ausgewählt werden.

Die Auswahl der Indizes erfolgt in zwei Phasen. Zunächst muss eine Menge aller verwendbaren Indizes definiert werden, und dann die Teilmenge der Indizes, die für die Suche verwendet werden soll. Diese Parameter werden über alle Programmausführungen hinweg beibehalten (sie werden für jede Recoll-Konfiguration separat gespeichert). Die Menge aller Indizes ist in der Regel recht stabil, während die aktiven Indizes in der Regel recht häufig angepasst werden können.

Der Hauptindex (definiert durch `RECOLL_CONFDIR`) ist immer aktiv. Wenn dies unerwünscht ist, können Sie Ihre Basiskonfiguration so einrichten, dass ein leeres Verzeichnis indiziert wird.

Wenn Sie einen neuen Index zum Set hinzufügen, können Sie entweder ein Recoll-Konfigurationsverzeichnis oder direkt ein Xapian-Indexverzeichnis auswählen. Im ersten Fall wird das Xapian-Indexverzeichnis aus der ausgewählten Konfiguration

übernommen.

Da die Zusammenstellung aller Indizes über die Benutzeroberfläche etwas mühsam sein kann, können Sie die Umgebungsvariable `RECOLL_EXTRA_DBS` verwenden, um einen Anfangssatz bereitzustellen. Diese kann typischerweise von einem Systemadministrator eingerichtet werden, damit nicht jeder Benutzer dies tun muss. Die Variable sollte eine durch Doppelpunkte getrennte Liste von Indexverzeichnissen definieren, d.h.:

```
export RECOLL_EXTRA_DBS=/einige/platz/xapiandb:/einige/andere/db
```

Eine weitere Umgebungsvariable, `RECOLL_ACTIVE_EXTRA_DBS`, ermöglicht das Hinzufügen zur aktiven Liste der Indizes. Diese Variable wurde von einem Recoll-Benutzer vorgeschlagen und implementiert. Sie ist vor allem dann nützlich, wenn Sie Skripte verwenden, um externe Volumes mit Recoll-Indizes zu mounten. Durch die Verwendung von `RECOLL_EXTRA_DBS` und `RECOLL_ACTIVE_EXTRA_DBS` können Sie den Index für den gemounteten Datenträger beim Start von **Recoll** hinzufügen und aktivieren. Nicht erreichbare Indizes werden beim Starten automatisch deaktiviert.

3.2.11 Geschichte dokumentieren

Dokumente, die Sie tatsächlich ansehen (mit der internen Vorschau oder einem externen Tool), werden in die Dokumentenhistorie eingetragen, die gespeichert wird.

Sie können die Historienliste über den Menüeintrag Tools/Doc History anzeigen.

Sie können den Dokumentenverlauf löschen, indem Sie den Eintrag Dokumentenverlauf löschen im Menü Datei wählen.

3.2.12 Sortieren von Suchergebnissen und Ausblenden von Duplikaten

Die Dokumente in einer Ergebnisliste werden normalerweise in der Reihenfolge ihrer Relevanz sortiert. Es ist möglich, eine andere Sortierreihenfolge festzulegen, indem Sie entweder die vertikalen Pfeile in der GUI-Toolbox verwenden, um nach Datum zu sortieren, oder zur Anzeige der Ergebnistabelle wechseln und auf eine beliebige Überschrift klicken. Die in der Ergebnistabelle gewählte Sortierreihenfolge bleibt aktiv, wenn Sie zurück zur Ergebnisliste wechseln, bis Sie auf einen der vertikalen Pfeile klicken, bis beide nicht mehr markiert sind (Sie sind wieder bei der Sortierung nach Relevanz).

Die Sortierparameter werden zwischen Programmaufrufen gespeichert, aber die Ergebnissortierung ist normalerweise immer inaktiv, wenn das Programm startet. Es ist möglich, den Aktivierungszustand der Sortierung zwischen Programmaufrufen beizubehalten, indem die Option Aktivierungszustand der Sortierung merken in den Voreinstellungen aktiviert wird.

Es ist auch möglich, doppelte Einträge in der Ergebnisliste auszublenden (Dokumente mit genau demselben Inhalt wie das angezeigte Dokument). Die Identitätsprüfung basiert auf einem MD5-Hash des Dokument-Containers, nicht nur des Textinhalts (so dass z.B. ein Textdokument, dem ein Bild hinzugefügt wurde, kein Duplikat des reinen Textes ist). Das Ausblenden von Duplikaten wird durch einen Eintrag im Konfigurationsdialog der GUI gesteuert und ist standardmäßig ausgeschaltet.

Wenn ein Ergebnisdokument nicht angezeigte Duplikate enthält, wird ein Link `"Dups"` zusammen mit dem Eintrag in der Ergebnisliste angezeigt. Wenn Sie auf diesen Link klicken, werden die Pfade (URLs + ipaths) für die doppelten Einträge angezeigt.

3.2.13 Tastaturkurzbefehle

Eine Reihe gängiger Aktionen innerhalb der grafischen Oberfläche können über Tastenkombinationen ausgelöst werden. Seit Recoll 1.29 können viele der Tastenkombinationen über einen Bildschirm in den GUI-Einstellungen angepasst werden. Die meisten Tastenkombinationen sind spezifisch für einen bestimmten Kontext (z. B. innerhalb eines Vorschaufensters, innerhalb der Ergebnistabelle).

Die meisten Tastenkombinationen können mit dem GUI-Tastenkürzel-Editor auf einen bevorzugten ~~Wer~~t geändert werden: Einstellungen GUI-Konfiguration Tastenkombinationen. Um ein Tastaturkürzel zu ändern, klicken Sie einfach auf die entsprechende Zelle in der Spalte Tastaturkürzel und geben Sie die gewünschte Sequenz ein.

3.2.14 Tipps zur Suche

3.2.14.1 Begriffe und Sucherweiterung

Begriffsvervollständigung Während der Eingabe in die einfache Suche erscheint ein Popup-Menü, das Vervollständigungen für die aktuelle Zeichenfolge anzeigt. Werte, denen ein Uhrensymbol vorangestellt ist, stammen aus dem Verlauf, solche, denen ein Lupensymbol vorangestellt ist, stammen aus den Indexbegriffen. Diese Funktion kann in den Einstellungen deaktiviert werden.

Neue Begriffe aus dem Ergebnis- oder Vorschautext übernehmen Ein Doppelklick auf ein Wort in der Ergebnisliste oder in einem Vorschaufenster kopiert es in das Eingabefeld der einfachen Suche.

Wildcards Wildcards können innerhalb von Suchbegriffen in allen Formen der Suche verwendet werden. Mehr über Wildcards.

Beschreibung	Standardwert
Kontext: fast überall	
Programm beenden	Strg+Q
Kontext: Erweiterte Suche	
Den nächsten Eintrag aus dem Suchverlauf laden	Nach oben
Den vorherigen Eintrag aus dem Suchverlauf laden	Daunen
Kontext: Hauptfenster	
Suche löschen. Dadurch wird der Tastaturcursor auf das Feld einfache Sucheingabe und Löschen des aktuellen Textes	Strg+S
Bewegen Sie den Tastaturcursor in den Sucheingabebereich ohne Löschen des aktuellen Textes	Strg+L
Bewegen Sie den Tastaturcursor in den Sucheingabebereich ohne Löschen des aktuellen Textes	Strg+Umschalt+S
Umschalten zwischen der Anzeige der aktuellen Ergebnisse als Tabelle oder als Liste	Strg+T
Kontext: Hauptfenster, wenn die Ergebnisse in einer Tabelle angezeigt werden	
Bewegen Sie den Tastaturcursor auf die aktuell ausgewählte Zeile in die Tabelle, oder zur ersten, wenn keine ausgewählt ist	Strg+R
Sprung zur Zeile 0-9 oder a-z in der Tabelle	Strg+[0-9] oder Strg+Shift+[a-z]
Abbrechen der aktuellen Auswahl	Esc
Kontext: Vorschaufenster	
Schließen Sie das Vorschaufenster	Esc
Schließen Sie die aktuelle Registerkarte	Strg+W
Öffnen eines Druckdialogs für den Inhalt der aktuellen Registerkarte	Strg+P
Laden des nächsten Ergebnisses aus der Liste in die aktuelle Registerkarte	Umschalt+Abwärts
Laden des vorherigen Ergebnisses aus der Liste in die aktuelle Registerkarte	Umschalt+Hoch
Kontext: Ergebnistabelle	
Kopieren Sie den im markierten Dokument enthaltenen Text in den Ordner Zwischenablage	Strg+G
Kopieren Sie den im markierten Dokument enthaltenen Text in den Ordner Zwischenablage, dann beenden Sie recoll	Strg+Alt+Umschalt+G
Das aktuelle Dokument öffnen	Strg+O
Das aktuelle Dokument öffnen und Recoll beenden	Strg+Alst+Umschalt+O
Eine vollständige Vorschau für das aktuelle Dokument anzeigen	Strg+D
Anzeige der Spaltennamen ein- und ausschalten	Strg+H
Zeigt eine Snippets-Liste (Stichwort im Kontext) für die aktuelle Dokument	Strg+E
Umschalten der Anzeige der Zeilenbuchstaben/-zahlen	Strg+V
Kontext: Schnipsel-Fenster	
Schließen Sie das Snippet-Fenster	Esc
In der Liste der Schnipsel suchen (Methode #1)	Strg+F
In der Liste der Schnipsel suchen (Methode #2)	/
Das nächste Vorkommen des Suchbegriffs finden	F3
Das vorherige Vorkommen des Suchbegriffs finden	Umschalt+F3

Tabelle 3.1: Tastaturkürzel

Automatische Suffixe Wörter wie `odt` oder `ods` können automatisch in Klauseln der Abfragesprache `ext:xxx` umgewandelt werden. Dies kann in den Sucheinstellungen in der grafischen Benutzeroberfläche aktiviert werden.

Deaktivieren der Stammerweiterung Die Eingabe eines großgeschriebenen Wortes in ein beliebiges Suchfeld verhindert die Stammerweiterung (keine Suche nach `Garten`, wenn Sie `Garten` statt `Garten` eingeben). Dies ist der einzige Fall, in dem die Groß- und Kleinschreibung bei einer Recoll-Suche einen Unterschied machen sollte. Sie können die Stammexpansion auch deaktivieren oder die Sprache der Stammexpansion in den Einstellungen ändern.

Suche nach verwandten Dokumenten Durch Auswahl des Eintrags Ähnliche Dokumente suchen im Rechtsklickmenü des Ergebnislistenabschnitts wird eine Reihe "interessanter" Begriffe aus dem aktuellen Ergebnis ausgewählt und in das Eingabefeld für die einfache Suche eingefügt. Sie können dann möglicherweise die Liste bearbeiten und eine Suche starten, um Dokumente zu finden, die mit dem aktuellen Ergebnis verwandt sein könnten.

Dateinamen Dateinamen werden bei der Indizierung als Begriffe hinzugefügt, und Sie können sie als normale Begriffe in normalen Suchfeldern angeben (Recoll indizierte früher alle Verzeichnisse im Dateipfad als Begriffe. Dies wurde aufgegeben, da es sich als nicht wirklich nützlich erwiesen hat). Alternativ können Sie die spezifische Suche nach Dateinamen verwenden, die *nur* nach Dateinamen sucht und möglicherweise schneller ist als die generische Suche, insbesondere wenn Sie Platzhalter verwenden.

3.2.14.2 Arbeiten mit Phrasen und Nähe

Phrasensuche Eine Phrase kann gesucht werden, indem eine Reihe von Begriffen in Anführungszeichen eingeschlossen wird. Beispiel: `"Benutzerhandbuch"` sucht nur nach Vorkommen von `"Benutzer"` unmittelbar gefolgt von `"Handbuch"`. Sie können das Feld "Phrase" des erweiterten Suchdialogs für den gleichen Effekt verwenden. Phrasen können zusammen mit einfachen Begriffen in alle Eingabefelder der einfachen oder erweiterten Suche eingegeben werden, mit Ausnahme von "Phrase".

Proximity-Suche Eine Proximity-Suche unterscheidet sich von einer Phrasensuche dadurch, dass sie keine Reihenfolge der Begriffe vorgibt. Proximity-Suchen können durch Angabe des Typs "Proximity" in der erweiterten Suche oder durch Anhängen eines 'p' an eine Phrasensuche eingegeben werden. Beispiel: "Benutzerhandbuch "p würde auch auf "Benutzerhandbuch" passen. Siehe auch den Abschnitt über Modifikatoren in der Dokumentation der Abfragesprache.

AutoPhrases Diese Option kann im Einstellungsdialog eingestellt werden. Wenn sie aktiviert ist, wird bei der Suche nach `beliebigen Begriffen` automatisch eine Phrase gebildet und zu einfachen Suchen hinzugefügt. Dadurch werden die Ergebnisse nicht grundlegend verändert, aber die Relevanz der Ergebnisse, in denen die Suchbegriffe als Phrase erscheinen, wird erhöht. D.h. eine Suche nach `"virtuelle Realität"` findet immer noch alle Dokumente, in denen entweder `"virtuell"` oder `"Realität"` oder beides vorkommt, aber diejenigen, die `"virtuelle Realität"` enthalten, sollten früher in der Liste erscheinen.

Die Phrasensuche kann eine Abfrage verlangsamen, wenn die meisten Begriffe in der Phrase häufig vorkommen. Wenn die Option `"Automatische Phrase"` aktiviert ist, werden sehr häufige Begriffe aus der automatisch erstellten Phrase entfernt. Der Schwellenwert für die Entfernung kann in den Sucheinstellungen angepasst werden.

Phrasen und Abkürzungen Gepunktete Abkürzungen wie `I.B.M.` werden ebenfalls automatisch als ein Wort ohne Punkte indiziert: `IBM`. Die Suche nach dem Wort innerhalb eines Satzes (z. B. `"die Firma IBM"`) führt nur dann zu einer Übereinstimmung mit der punktierten Abkürzung, wenn Sie den Phrasendruck erhöhen (über das erweiterte Suchfeld oder den Sprachmodifikator `o`). Wörtliche Vorkommen des Wortes werden normal gefunden.

3.2.14.3 Andere

Felder verwenden Sie können die Abfragesprache und Feldspezifikationen verwenden, um nur bestimmte Teile von Dokumenten zu durchsuchen. Dies kann vor allem bei E-Mails hilfreich sein, z. B. um nur E-Mails eines bestimmten Absenders zu suchen: `Suchtipps von:helpfulgui`

Anpassen der Spalten der Ergebnistabelle Wenn Sie die Ergebnisse im Tabellenmodus anzeigen, können Sie mit einem Rechtsklick auf die Tabellenköpfe ein Popup-Menü aktivieren, mit dem Sie die angezeigten Spalten anpassen können. Sie können die Spaltenüberschriften ziehen, um ihre Reihenfolge zu ändern. Sie können sie anklicken, um nach dem in der Spalte angezeigten Feld zu sortieren. Sie können die Ergebnisliste auch im CSV-Format speichern.

Ändern der GUI-Geometrie Es ist möglich, die GUI im Breitformat zu konfigurieren, indem Sie die Symbolleisten auf eine der Seiten ziehen (ihre Position wird zwischen den Sitzungen gespeichert) und die Kategoriefilter in ein Menü verschieben (kann in den Einstellungen → GUI-Konfiguration → Benutzeroberfläche eingestellt werden).

Abfrageerläuterung Eine genaue Beschreibung dessen, wonach die Abfrage gesucht hat, einschließlich der Stammerweiterung und der verwendeten booleschen Operatoren, erhalten Sie, wenn Sie auf die Kopfzeile der Ergebnisliste

klicken.

Historie der erweiterten Suche Sie können alle zuletzt durchgeführten komplexen100 Suchvorgänge anzeigen, indem Sie die Pfeiltasten nach oben und unten verwenden, während das Bedienfeld für die erweiterte Suche aktiv ist.

Erzwungenes Öffnen eines Vorschaufensters Sie können mit **Umschalt+Klick** auf einen Vorschaulink in der Ergebnisliste die Erstellung eines Vorschaufensters erzwingen, anstatt eine neue Registerkarte im bestehenden Fenster zu öffnen.

3.2.15 Speichern und Wiederherstellen von Abfragen (1.21 und später)

Sowohl die einfachen als auch die erweiterten Abfragedialoge speichern den aktuellen Verlauf, aber die Anzahl ist begrenzt: alte Abfragen werden irgendwann vergessen. Außerdem kann es schwierig sein, wichtige Abfragen unter anderen zu finden. Aus diesem Grund können beide Arten von Abfragen auch explizit in Dateien gespeichert werden, und zwar über die Menüs der Benutzeroberfläche: Datei → Letzte Abfrage speichern / Letzte Abfrage laden

Der Standardspeicherort für gespeicherte Abfragen ist ein Unterverzeichnis des aktuellen Konfigurationsverzeichnisses, aber gespeicherte Abfragen sind normale Dateien und können überall geschrieben oder verschoben werden.

Einige der gespeicherten Abfrageparameter sind Teil der Voreinstellungen (z.B. `Autophrase` oder die aktiven externen Indizes) und können sich beim Laden der Abfrage von dem Zeitpunkt unterscheiden, an dem sie gespeichert wurde. In diesem Fall warnt Recoll vor den Unterschieden, ändert aber nicht die Benutzereinstellungen.

3.2.16 Anpassen der Suchoberfläche

Sie können einige Aspekte der Suchoberfläche anpassen, indem Sie den Eintrag GUI-Konfiguration im Menü Voreinstellungen verwenden.

Das Dialogfeld enthält mehrere Registerkarten, die sich mit der Schnittstelle selbst, den für die Suche und die Rückgabe von Ergebnissen verwendeten Parametern und den zu durchsuchenden Indizes befassen.

Parameter der Benutzeroberfläche:

- Hervorhebungsfarbe für Abfragebegriffe: Begriffe aus der Benutzerabfrage werden in den Beispielen der Ergebnisliste und im Vorschaufenster hervorgehoben. Die Farbe kann hier gewählt werden. Jeder Qt-Farbstring sollte funktionieren (z.B. `rot`, `#ff0000`). Die Voreinstellung ist `blau`.
- Stilvorlage: Der Name einer Qt-Stylesheet-Textdatei, die beim Starten auf die gesamte Recoll-Anwendung angewendet wird. Der Standardwert ist leer, aber es gibt ein Skelett-Stylesheet (`recoll.qss`) im Verzeichnis `/usr/share/recoll/examples`. Mit einer Stilvorlage können Sie die meisten grafischen Parameter von **recoll** ändern: Farben, Schriftarten usw. In der Beispieldatei finden Sie ein paar einfache Beispiele.

  Sie sollten sich darüber im Klaren sein, dass Parameter (z.B. die Hintergrundfarbe), die in der Recoll-GUI-Vorlage gesetzt werden, die globalen Systemeinstellungen überschreiben, was zu merkwürdigen Nebeneffekten führen kann: Wenn Sie z.B. den Vordergrund auf eine helle Farbe und den Hintergrund auf eine dunkle Farbe in den Desktop-Einstellungen setzen, aber nur der Hintergrund in der Recoll-Vorlage gesetzt ist, und auch dieser ist hell, dann wird der Text in der Recoll-GUI hell auf hell erscheinen.
- Maximale Textgröße, die für die Vorschau hervorgehoben wird Das Hervorheben von Suchbegriffen innerhalb des Textes vor dem Einfügen in das Vorschaufenster ist mit ziemlich viel Rechenarbeit verbunden und kann bei einer bestimmten Textgröße deaktiviert werden, um das Laden zu beschleunigen.
- HTML gegenüber reinem Text für die Vorschau bevorzugen: Wenn diese Option aktiviert ist, zeigt Recoll HTML als solches im Vorschaufenster an. Wenn dies zu Problemen mit der Qt-HTML-Anzeige führt, können Sie diese Option deaktivieren, um stattdessen die reine Textversion anzuzeigen.
- Links in der Vorschau aktivieren: Wenn diese Option aktiviert ist, wandelt Recoll HTTP-Links, die im reinen Text gefunden werden, in richtige HTML-Anker um, und wenn Sie auf einen Link in einem Vorschaufenster klicken, wird der Standardbrowser auf dem Linkziel gestartet.
- Klartext in HTML-Zeilenstil: Bei der Anzeige von Klartext im Vorschaufenster versucht Recoll, einige der ursprünglichen Zeilenumbrüche und Einrückungen beizubehalten. Es kann entweder PRE-HTML-Tags verwenden, wodurch die Einrückung gut erhalten bleibt, aber bei langen Zeilen ein horizontaler Bildlauf erzwungen wird, oder BR-Tags verwenden, um an den ursprünglichen Zeilenumbrüchen umzubrechen, wodurch der Editor andere Zeilenumbrüche entsprechend der Fensterbreite einfügen kann, aber ein Teil der ursprünglichen Einrückung verloren geht. Die dritte Möglichkeit wurde in den letzten Versionen eingeführt und ist jetzt wahrscheinlich die beste: die Verwendung von PRE-Tags mit Zeilenumbruch.
- Editor-Anwendung wählen: Dies öffnet ein Dialogfeld, in dem Sie die Anwendung auswählen können, die zum Öffnen der einzelnen MIME-Typen verwendet werden soll. Standardmäßig wird das Dienstprogramm **xdg-open** verwendet, aber Sie können dieses Dialogfeld verwenden, um Ausnahmen für MIME-Typen festzulegen, die dennoch gemäß den Recoll-Einstellungen geöffnet werden sollen. Dies ist nützlich für die Übergabe von Parametern wie Seitenzahlen oder Suchstrings an Anwendungen, die diese unterstützen (z. B. evince). Dies kann nicht mit **xdg-open** gemacht werden, das nur die

Übergabe eines Parameters unterstützt.

- Qt-Autovervollständigung in der Sucheingabe deaktivieren: Damit wird das Vervollständigungs-Popup deaktiviert. Es erscheint nur, wenn Sie nur Leerzeichen in den Suchbereich eingeben oder wenn Sie auf die Uhr-Schaltfläche rechts neben dem Bereich klicken, und zeigt den vollständigen Verlauf an.
- Art der Dokumentenfilterauswahl: Hier können Sie wählen, ob die Dokumentenkategorien als Liste, als Schaltflächen oder als Menü angezeigt werden sollen.

- Start mit einfachem Suchmodus: Damit können Sie den Wert des einfachen Suchtyps beim Programmstart wählen. Entweder ein fester Wert (z. B. `Query Language`) oder der Wert, der beim letzten Beenden des Programms verwendet wurde.
- Mit geöffnetem Dialog für erweiterte Suche starten: Wenn Sie diesen Dialog häufig verwenden, können Sie ihn beim Start von recoll öffnen lassen.
- Remember sort activation state wenn gesetzt, merkt sich Recoll den Status des Sortierwerkzeugs zwischen den Aufrufen. Normalerweise startet es mit deaktivierter Sortierung.

Parameter der Ergebnisliste:

- Anzahl der Ergebnisse auf einer Ergebnisseite
- Schriftart der Ergebnisliste: In der Ergebnisliste werden ziemlich viele Informationen angezeigt, und Sie möchten vielleicht die Schriftart und/oder Schriftgröße anpassen. Der Rest der von Recoll verwendeten Schriftarten wird durch Ihre allgemeine Qt-Konfiguration bestimmt (versuchen Sie den Befehl **qtconfig**).
- Absatzformat-String der Ergebnisliste bearbeiten: Ermöglicht es Ihnen, die Darstellung jedes Eintrags der Ergebnisliste zu ändern. Siehe den Abschnitt Anpassung der Ergebnisliste.
- HTML-Kopfzeile der Ergebnisseite bearbeiten: ermöglicht es Ihnen, einen Text zu definieren, der am Ende der HTML-Kopfzeile der Ergebnisseite eingefügt wird. Weitere Einzelheiten finden Sie im Abschnitt Anpassung der Ergebnisliste.
- Datumsformat: ermöglicht die Angabe des Formats für die Anzeige von Datumsangaben in der Ergebnisliste. Dies sollte als strftime()-String angegeben werden (man strftime).
- Trennzeichen für abstrakte Auszüge: Für synthetische Auszüge, die aus Indexdaten erstellt werden und in der Regel aus mehreren Auszügen aus verschiedenen Teilen des Dokuments bestehen, wird hier das Trennzeichen für die Auszüge festgelegt, das standardmäßig ein Ellipsis ist.

Suchparameter:

- Doppelte Ergebnisse ausblenden: entscheidet, ob Einträge in der Ergebnisliste für identische Dokumente, die an verschiedenen Orten gefunden wurden, angezeigt werden.
- Stemming-Sprache: Das Stemming hängt natürlich von der Sprache des Dokuments ab. In dieser Listbox können Sie zwischen den Stemming-Datenbanken wählen, die während der Indizierung erstellt wurden (dies wird in der Hauptkonfigurationsdatei festgelegt) oder später mit **recollindex** hinzugefügt wurden **-s** (siehe das Handbuch zu recollindex). Dynamisch hinzugefügte Stemming-Sprachen werden beim nächsten Indizierungsdurchgang gelöscht, sofern sie nicht ebenfalls in der Konfigurationsdatei hinzugefügt wurden.
- Automatisches Hinzufügen von Phrasen zu einfachen Suchen: Bei der Suche nach `beliebigen Begriffen wird` automatisch eine Phrase gebildet und zu einfachen Suchen hinzugefügt. Dies erhöht die Relevanz der Ergebnisse, wenn die Suchbegriffe als Phrase erscheinen (aufeinanderfolgend und in der richtigen Reihenfolge).
- Prozentualer Schwellenwert für die Häufigkeit von Autophrasen: Sehr häufige Begriffe sollten aus Leistungsgründen nicht in die automatische Phrasensuche einbezogen werden. Der Parameter definiert den Schwellenwert in Prozent (Prozentsatz der Dokumente, in denen der Begriff vorkommt).
- Ersetzen von Zusammenfassungen aus Dokumenten: Hier wird entschieden, ob eine Zusammenfassung anstelle einer expliziten Zusammenfassung aus dem Dokument selbst synthetisiert und angezeigt werden soll.
- Dynamische Erstellung von Zusammenfassungen: Hier wird festgelegt, ob Recoll bei der Anzeige der Ergebnisliste versucht, Dokumentzusammenfassungen (Listen von *Snippets*) zu erstellen. Die Zusammenfassungen werden aus dem Kontext der Dokumentinformationen rund um die Suchbegriffe erstellt.
- Synthetische abstrakte Größe: nach Geschmack anpassen...
- Synthetische abstrakte Kontextwörter: wie viele Wörter um jedes Vorkommen eines Begriffs angezeigt werden sollen.
- Magische Dateinamenssuffixe für Abfragesprachen: eine Liste von Wörtern, die automatisch in `ext:xxx`-Dateinamenssuffix-Klauseln umgewandelt werden, wenn eine Abfrage in einer Abfragesprache gestartet wird (z.B.: `doc xls xlsx...`). Dies erspart Leuten, die bei Abfragen häufig Dateitypen verwenden, einiges an Tipparbeit.

Externe Indizes:

In diesem Bereich können Sie nach weiteren Indizes suchen, die Sie durchsuchen möchten. Externe Indizes werden durch ihr Datenbankverzeichnis bezeichnet (z. B. `/home/someothergui/.recoll/xapiandb`, `/usr/local/recollglobal/xapiandb`).

Nach der Eingabe erscheinen die Indizes in der Liste Externe Indizes, und Sie können jederzeit auswählen, welche Indizes Sie verwenden möchten, indem Sie die entsprechenden Einträge an- oder abwählen.

Ihre Hauptdatenbank (diejenige, die von der aktuellen Konfiguration indiziert wird), ist immer implizit aktiv. Wenn dies nicht erwünscht ist, können Sie Ihre Konfiguration so einrichten, dass sie z.B. ein leeres Verzeichnis indiziert. Ein alternativer Indexer muss möglicherweise auch eine Möglichkeit implementieren, den Index von veralteten Daten zu bereinigen,

3.2.16.1 Das Format der Ergebnisliste

Recoll verwendet normalerweise einen voll funktionsfähigen HTML-Prozessor, um die Ergebnisliste und das Snippets-Fenster anzuzeigen. Je nach Version kann dieser entweder auf Qt WebKit oder Qt WebEngine basieren. Es ist dann möglich, die Ergebnisliste mit voller Unterstützung für CSS und Javascript vollständig anzupassen.

Es ist auch möglich, Recoll so zu bauen, dass es ein einfacheres Qt QTextBrowser Widget zur Anzeige des HTML verwendet, was notwendig sein kann, wenn die oben genannten nicht auf das System portiert sind, oder um die Größe der Anwendung und die Abhängigkeiten zu reduzieren. In diesem Fall sind den Möglichkeiten Grenzen gesetzt, aber es ist immer noch möglich, zu entscheiden, welche Daten jedes Ergebnis enthält und wie sie angezeigt werden sollen.

Die Darstellung der Ergebnisliste kann durch die Anpassung von zwei Elementen angepasst werden:

- Das Absatzformat
- HTML-Code innerhalb des Kopfbereichs. Für Versionen und1.21 spätere Versionen wird dies auch für das Snippets-Fenster verwendet.

Das Absatzformat und das Kopffragment können auf der Registerkarte Ergebnisliste der GUI-Konfiguration bearbeitet werden.

Das Header-Fragment wird sowohl für die Ergebnisliste als auch für das Snippets-Fenster verwendet. Die Snippets-Liste ist eine Tabelle und hat ein Klassenattribut `snippets`. Jeder Absatz in der Ergebnisliste ist eine Tabelle mit der Klasse `respar`, aber das kann durch Bearbeiten des Absatzformats geändert werden.

Auf der Seite zur Anpassung der Ergebnisliste auf der Recoll-Website finden Sie einige Beispiele.

3.2.16.1.1 Das Absatzformat

Dies ist eine beliebige HTML-Zeichenkette, in der die folgenden printf-ähnlichen %-Ersetzungen durchgeführt werden:

- **%A** Zusammenfassung
- **%D** Datum
- **%I** Name des Symbolbildes. Dieser wird normalerweise anhand des MIME-Typs ermittelt. Die Zuordnungen werden in der Konfigurationsdatei mimeconf definiert. Wenn ein Thumbnail für die Datei am Standardspeicherort von Freedesktop gefunden wird, wird dieses stattdessen angezeigt.
- **%K** Schlüsselwörter (falls vorhanden)
- **%L** Vorgefertigte Vorschau-, Bearbeitungs- und möglicherweise Snippet-Links
- **%M** MIME-Typ
- **%N** Ergebnis Nummer innerhalb der Ergebnisseite
- **%P** Übergeordneter Ordner Url. Im Falle eines eingebetteten Dokuments ist dies der übergeordnete Ordner für die Containerdatei der obersten Ebene.
- **%R** Relevanz in Prozent
- **%S** Größeninformationen

- **%T** Titel oder Dateiname, falls nicht festgelegt.

- **%t** Titel oder leer.
- **%(Dateiname)** Dateiname.
- **%U** Url

Das Format der Links Vorschau, Bearbeiten und Schnipsel ist `<a href="P%N">` , `<a href="E%N">` und `<a href="A%N">`, wobei *docnum* (%N) wird zur Dokumentennummer auf der Ergebnisseite).

Ein als `"F%N"` definiertes Verknüpfungsziel öffnet das Dokument, das der Erweiterung des übergeordneten Ordners `%P` entspricht, und erzeugt in der Regel ein Dateimanager-Fenster in dem Ordner, in dem sich die Containerdatei befindet. Z.B.:

```
<a href="F%N">%P</a>
```

Ein als `R%N|scriptname` definiertes Verknüpfungsziel führt das entsprechende Skript in der Ergebnisdatei aus (wenn das Dokument eingebettet ist, wird das Skript auf der obersten übergeordneten Ebene gestartet). Siehe den Abschnitt über die Definition von Skripten.

Zusätzlich zu den oben genannten vordefinierten Werten werden alle Zeichenfolgen wie `%(Feldname)` durch den Wert des Feldes namens `Feldname` für dieses Dokument ersetzt. Nur auf gespeicherte Felder kann auf diese Weise zugegriffen werden, der Wert indizierter, aber nicht gespeicherter Felder ist zu diesem Zeitpunkt des Suchprozesses nicht bekannt (siehe Feldkonfiguration). Derzeit werden außer den oben genannten Werten nur sehr wenige Felder standardmäßig gespeichert (nur `Autor` und `Dateiname`), so dass diese Funktion nur dann sinnvoll ist, wenn sie lokal konfiguriert werden kann. Ein Beispiel hierfür wäre das Empfängerfeld, das von den Nachrichteneingabe-Handlern erzeugt wird.

Der Standardwert für die Absatzformatzeichenfolge ist:

```
"<table
class=\"respar\">\n"
"<tr>\n"
"<td><a href='%U'><img src='%I'
width='64'></a></td>\n"
"<td>%L %M/%S %D    <i
%U %i<br><br>\n"
"%A %K</td>\n"
"</tr></table>\n"
```

Sie können z. B. Folgendes versuchen, um eine webähnliche Erfahrung zu machen:

```
<u><b><a href="P%N">%T</a></b></u><br>
%A<font color="#008000">%U</font> %S<br>%L
```

Beachten Sie, dass der Link P%N im obigen Absatz den Titel zu einem Vorschaulink macht. Oder das saubere Aussehen:

```
<img src="%I" align="left">%L <font
color="#808080">%R</font>   <b>%T</b><br>%S 
<font color="#808080"><i>%U</i></font>
<table bgcolor="#e0e0e0">
<tr><td><div>%A %K</div></td></tr>
</tabelle>%K
```

Diese und einige andere Beispiele sind auf der Website zu finden, mit Bildern, die zeigen, wie sie aussehen. Es ist auch möglich, den Wert des Snippet-Trennzeichens innerhalb des Abschnitts "Zusammenfassung" zu definieren.

3.3 Suche mit dem KDE KIO-Slave

3.3.1 Was ist das?

Der Recoll KIO-Slave ermöglicht die Durchführung einer Recoll-Suche durch Eingabe einer entsprechenden URL in einem KDE-Dialog oder mit einer HTML-basierten Schnittstelle, die in **Konqueror** angezeigt wird.

Die HTML-basierte Schnittstelle ähnelt der Qt-basierten Schnittstelle, ist aber im Moment etwas weniger leistungsfähig. Ihr Vorteil ist, dass Sie Ihre Suche durchführen können, während Sie vollständig im KDE-Rahmen bleiben: Ziehen und Ablegen aus der Ergebnisliste funktioniert normal und Sie haben Ihre normale Auswahl an Anwendungen zum Öffnen von Dateien.

Die alternative Schnittstelle verwendet eine Verzeichnisansicht der Suchergebnisse. Aufgrund von Einschränkungen in der aktuellen KIO-Slave-Schnittstelle ist sie derzeit (für mich) nicht offensichtlich nützlich.

Die Schnittstelle wird in einer Hilfedatei detaillierter beschrieben, die Sie durch Eingabe von `recoll:/` im **Konqueror** aufrufen können URL-Zeile (dies funktioniert nur, wenn der recoll KIO-Slave zuvor installiert wurde).

Die Anweisungen zur Erstellung dieses Moduls befinden sich im Quellbaum. Siehe: `kde/kio/recoll/00README.txt`. Einige Linux-Distributionen paketieren das kio-recoll-Modul, also prüfen Sie, bevor Sie in den Erstellungsprozess einsteigen, ob es vielleicht schon da ist und mit einem Klick installiert werden kann.

3.3.2 Durchsuchbare Dokumente

Als Beispielanwendung könnte der Recoll KIO-Slave es ermöglichen, eine Reihe von HTML-Dokumenten (z.B. ein Handbuch) so aufzubereiten, dass sie zu einer eigenen Suchoberfläche innerhalb von **Konqueror** werden.

Dies kann entweder durch das explizite Einfügen von `<a href="recoll://...">`Links um einige Dokumentbereiche herum geschehen oder automatisch durch das Hinzufügen eines sehr kleinen Javascript-Programms zu den Dokumenten, wie im folgenden Beispiel, das eine Suche durch Doppelklick auf einen beliebigen Begriff auslöst:

```
<script language="JavaScript">
        function recollsearch()
        {
        var t = document.getSelection();
        window.location.href = 'recoll://search/query?qtp=a&p=0&q='
        + encodeURIComponent(t);
        }
        </script>
        ...
        <body ondblclick="recollsearch()">
```

3.4 Suche in der Befehlszeile

Es gibt mehrere Möglichkeiten, Suchergebnisse als Textstrom zu erhalten, ohne eine grafische Schnittstelle:

- Durch Übergabe der Option `-t` an das Programm **recoll** oder durch Aufruf als **recollq** (über einen Link).
- Mit Hilfe des Programms **recollq**.
- Durch das Schreiben eines benutzerdefinierten Python-Programms unter Verwendung der Recoll Python API.

Die ersten beiden Methoden funktionieren auf die gleiche Weise und akzeptieren/benötigen die gleichen Argumente (außer dem zusätzlichen `-t` für **recoll**). Die auszuführende Abfrage wird als Befehlszeilenargumente angegeben.

recollq wird nicht immer standardmäßig gebaut. Sie können das `Makefile` im Abfrageverzeichnis verwenden, um es zu erstellen. Dies ist ein sehr einfaches Programm, und wenn Sie ein wenig C++ programmieren können, können Sie das Ausgabeformat an Ihre Bedürfnisse anpassen. Abgesehen davon, dass es

leicht angepasst werden kann, ist **recollq** nur auf Systemen wirklich nützlich, auf denen die Qt-Bibliotheken nicht verfügbar sind, ansonsten ist es überflüssig mit
`recoll -t`.

recollq hat eine Manpage. Der Usage String folgt:

```
recollq: Verwendung:
 -P: Zeigt die Datumsspanne für alle im Index vorhandenen
 Dokumente an [-o|-a|-f] [-q] <Abfragezeichenfolge>
 Führt eine Rückabfrage durch und zeigt die Ergebniszeilen an.
  Standard: interpretiert das/die Argument(e) als xesam-
  Abfragezeichenfolge Abfrageelemente:
   * Implizites AND, Ausschluss, Feld spec: t1 -t2 Titel:t3
   * OR hat Vorrang: t1 OR t2 t3 OR t4 bedeutet (t1 OR t2) AND (t3 OR t4)
   * Phrase: "t1 t2" (erfordert zusätzliche Anführungszeichen in der Befehlszeile)
 -o Emulation der einfachen GUI-Suche im Modus ANY TERM
 -a Emulation der einfachen GUI-Suche im Modus ALL TERMS
 -f Emulation der einfachen GUI-Suche im Dateinamenmodus
 -q wird einfach ignoriert (Kompatibilität mit der GUI-Befehlszeile von
recoll) Allgemeine Optionen:
 -c <Konfigurationsverzeichnis> : Angabe des Konfigurationsverzeichnisses mit Vorrang vor
 $RECOLL_CONFDIR
 -und auch den Inhalt der Datei ausgeben
 -n [first-]<cnt> definieren das Ergebnis-Slice. Der Standardwert für
    [first] ist Ohne0. diese Option ist der Standardwert für die
    maximale Anzahl 2000.
    Verwenden Sie n=0 für keine Begrenzung
 -b : einfach. Nur Urls ausgeben, keine Mime-Typen oder Titel
 -Q : keine Ergebniszeilen, nur die verarbeitete Abfrage und die Anzahl der Ergebnisse
 -m : das gesamte meta[]-Array des Dokuments für jedes Ergebnis ausgeben
 -A : Ausgabe der Dokumentzusammenfassungen
 -S fld : Sortieren nach Feld <fld>
   -D : absteigend sortieren
 -s stemlang : die zu verwendende Stemming-Sprache festlegen (muss im
    Index vorhanden sein...) -s "" verwenden, um die Stem-Expansion zu
    deaktivieren
 -T <Synonyme-Datei>: den Parameter (Thesaurus) für die Worterweiterung verwenden
 -i <dbdir> : zusätzlicher Index, es können mehrere angegeben werden
 -e Url-Kodierung (%xx) für Urls verwenden
 -F <Feldnamenliste> : gibt genau diese Felder für jedes Ergebnis aus.
    Die Feldwerte werden in base64 kodiert, in einer Zeile ausgegeben
    und durch ein Leerzeichen getrennt. Dies ist das empfohlene Format
    für die Verwendung durch andere Programme. Verwenden Sie eine
    normale Abfrage mit der Option -m, um
    sehen Sie die Feldnamen. Verwenden Sie -F '', um alle Felder auszugeben,
    aber Sie wollen in diesem Fall wahrscheinlich auch die Option -N
  -N : mit -F werden die Feldnamen (Klartext) vor den Feldwerten ausgegeben
```

Beispielhafte Ausführung:

```
recollq 'ilur -nautique mime:text/html'
Abfrage wiederholen: ((((ilur:(wqf=11) OR ilurs) AND_NOT (nautique:(wqf=11) OR
nautiques OR nautiquement)) FILTER Ttext/html))
4 Ergebnisse
text/html       [file:///Users/uncrypted-dockes/projets/bateaux/ilur/comptes.html]         [
    comptes.html] 18593    Bytes                                                          ←'
text/html       [file:///Users/uncrypted-dockes/projets/nautique/webnautique/articles/ilur1 ←'
    /index.html] [Konstrukt...

                text/html[file:///Benutzer/unverschlüsselt-
dockes/projets/pagepers/index.html][psxtcl/←'
    writemime/recoll]...
text/html       [file:///Users/uncrypted-dockes/projets/bateaux/ilur/factEtCie/recu-chasse- ←'
    maree....
```

3.5 Die Abfragesprache

Die Recoll-Abfragesprache basiert auf der inzwischen nicht mehr existierenden Xesam-Suchsprache. Sie ermöglicht die Definition allgemeiner boolescher Suchen im Haupttext oder in bestimmten Feldern und verfügt über viele zusätzliche Funktionen, die im Großen und Ganzen denen der *komplexen* Suchschnittstelle in der GUI entsprechen.

Der Abfragesprachprozessor wird im GUI-Eintrag für die einfache Suche aktiviert, wenn der Suchmodus-Selektor auf `Abfragesprache` eingestellt ist. Er kann auch über die Befehlszeilensuche, den KIO-Slave oder das WEB UI verwendet werden.

Wenn die Ergebnisse einer Abfragesprachen-Suche Sie verwirren und Sie Zweifel haben, wonach tatsächlich gesucht wurde, können Sie die GUI verwenden
Link `Abfrage anzeigen` am Anfang der Ergebnisliste, um die genaue Abfrage zu überprüfen, die schließlich von Xapian ausgeführt wurde.

3.5.1 Allgemeine Syntax

Im Folgenden wird ein Beispielantrag erläutert:

```
autor: "john doe" Beatles OR Lennon Live OR Unplugged -Kartoffeln
```

Dies würde nach allen Dokumenten suchen, in denen *`John Doe`* als Phrase im Autorenfeld auftaucht (was genau das ist, hängt vom Dokumententyp ab, z. B. von der `Von:` Kopfzeile bei einer E-Mail-Nachricht) und die entweder *`Beatles`* oder *`Lennon`* und entweder *`Live`* oder *`Unplugged`*, aber keine *`Kartoffeln`* (in irgendeinem Teil des Dokuments) enthalten.

Ein Element besteht aus einer optionalen Feldangabe und einem Wert, die durch einen Doppelpunkt getrennt sind (das Feldtrennzeichen ist der letzte Doppelpunkt im Element). Beispiele:

- *`Eugenie`*
- *`autor:balzac`*
- *`dc:title:grandet`*
- *`dc:title: "eugenie grandet"`*

Der Doppelpunkt, falls vorhanden, bedeutet "enthält". Xesam definiert weitere Relationen, die derzeit größtenteils nicht unterstützt werden (außer in speziellen Fällen, die weiter unten beschrieben werden).

Alle Elemente im Sucheintrag werden normalerweise mit einem impliziten UND verknüpft. Es ist möglich, Elemente stattdessen mit ODER zu verknüpfen, wie in *`Beatles`* `ODER` *`Lennon`*. Das `ODER` muss wörtlich (in Großbuchstaben) eingegeben werden und hat Vorrang vor den UND-Verknüpfungen: *`wort1 wort2`* `ODER` *`wort3`* bedeutet *`wort1`* UND (*`wort2`* `ODER` *`wort3`*) nicht (*`wort1`* UND *`wort2`*) `ODER` *`wort3`*.

Sie können Klammern verwenden, um Elemente zu gruppieren (ab Version 1.21), was die Dinge manchmal klarer macht und möglicherweise Kombinationen ermöglicht, die sonst schwierig gewesen wären.

Ein Element, dem ein `-` vorangestellt ist, gibt einen Begriff an, der *nicht* erscheinen soll.

Wie üblich definieren Wörter in Anführungszeichen eine Phrase (die Reihenfolge der Wörter ist von Bedeutung), so dass *`title: "Vorurteil Stolz"`* nicht dasselbe ist wie *`title:Vorurteil title:Stolz`*, und es ist unwahrscheinlich, dass man ein Ergebnis findet.

Wörter innerhalb von Phrasen und großgeschriebene Wörter werden nicht stamm-expandiert. Platzhalter können an beliebiger Stelle innerhalb eines Begriffs verwendet werden. Die Angabe eines Platzhalters auf der linken Seite eines Begriffs kann zu einer sehr langsamen Suche führen (oder sogar zu einer falschen, wenn die Expansion aufgrund einer zu großen Größe abgeschnitten wird). Siehe auch Mehr über Wildcards.

Um Ihnen einige Tipparbeit zu ersparen, interpretieren Recoll-Versionen ab 1.20 einen Feldwert, der als kommagetrennte Liste von Begriffen angegeben wird, als UND-Liste und eine durch Schrägstrich getrennte Liste als ODER-Liste. Leerzeichen sind nicht erlaubt. Also

```
autor:john,lennon
```

sucht nach Dokumenten mit `john` und `lennon` im Autorenfeld (in beliebiger Reihenfolge), und

```
autor:john/ringo
```

würde nach "john" oder "ringo" suchen. Dieses Verhalten wird nur durch ein Feldpräfix ausgelöst: Ohne dieses Präfix führen komma- oder schrägstrichgetrennte Eingaben zu einer Phrasensuche. Sie können jedoch auch einen Textfeldnamen verwenden, um den Haupttext auf diese Weise zu durchsuchen, als Alternative zu einem expliziten ODER, z. B. würde text:napoleon/bonaparte eine Suche nach *napoleon* oder *bonaparte* im Haupttextkörper erzeugen.

Modifikatoren können auf einen Wert in doppelten Anführungszeichen gesetzt werden, um z. B. eine Umkreissuche (ungeordnet) anzugeben. Siehe den Abschnitt Modifikatoren. Zwischen dem letzten Anführungszeichen und dem Modifikatorwert darf kein Leerzeichen stehen, z. B. *"zwei eins "po10*

Recoll verwaltet derzeit die folgenden Standardfelder:

- title, subject oder caption sind Synonyme, die angeben, nach welchen Daten im Titel oder Betreff des Dokuments gesucht werden soll.
- Autor oder von für die Suche nach den Urhebern der Dokumente.
- Empfänger oder zur Suche nach den Dokumentenempfängern.
- Schlüsselwort für die Suche nach den im Dokument angegebenen Schlüsselwörtern (nur wenige Dokumente haben tatsächlich welche).
- filename für den Dateinamen des Dokuments. Sie können den kürzeren Alias fn verwenden. Dieser Wert wird nicht für alle Dokumente gesetzt: Interne Dokumente, die in einem zusammengesetzten Dokument enthalten sind (z. B. ein EPUB-Abschnitt), erben den Container-Dateinamen nicht mehr, dieser wurde durch ein explizites Feld ersetzt (siehe unten). Unterdokumente können immer noch einen Dateinamen haben, wenn dieser durch das Dokumentformat impliziert wird, z. B. der Dateiname für einen E-Mail-Anhang.
- containerfilename, auch bekannt als cfn. Dieses Feld wird für alle Dokumente gesetzt, sowohl für Dokumente der obersten Ebene als auch für enthaltene Unterdokumente, und ist immer der Name der Dateisystemdatei, die die Daten enthält. Die Begriffe aus diesem Feld können nur durch eine explizite Feldspezifikation abgeglichen werden (im Gegensatz zu Begriffen aus filename, die auch als allgemeiner Dokumentinhalt indiziert werden). Dadurch wird vermieden, dass bei der Suche nach dem Namen der Containerdatei Treffer für alle Unterdokumente erzielt werden.
- ext gibt die Erweiterung des Dateinamens an (Beispiel: ext:html).
- rclmd5 die MD5-Prüfsumme für das Dokument. Sie wird verwendet, um die Duplikate eines Suchergebnisses anzuzeigen (bei Abfragen mit der Option, doppelte Ergebnisse auszublenden). Nebenbei bemerkt, könnte man damit auch die Duplikate einer beliebigen Datei finden, indem man ihre MD5-Prüfsumme berechnet und eine Abfrage nur mit dem rclmd5-Wert ausführt.

Sie können Aliase für Feldnamen definieren, um Ihre bevorzugte Bezeichnung zu verwenden oder um sich die Eingabe zu ersparen (z.B. die vordefinierten fn und
cfn-Aliase für filename und containerfilename definiert). Siehe den Abschnitt über die Datei fields.

Die Dokumenteneingabe-Handler haben die Möglichkeit, andere Felder mit beliebigen Namen zu erstellen, und Aliase können in der Konfiguration definiert werden, so dass die genauen Feldsuchmöglichkeiten für Sie anders sein können, wenn jemand die Anpassung vorgenommen hat.

3.5.2 Spezielle feldähnliche Bezeichner

Die Feldsyntax unterstützt auch einige feldähnliche, aber spezielle Kriterien, bei denen die Werte anders interpretiert werden. Die reguläre Verarbeitung gilt nicht (z. B. funktionieren die Schrägstrich- oder Komma-getrennten Listen nicht). Es folgt eine Liste.

- dir, um die Ergebnisse nach dem Dateispeicherort zu filtern. Beispiel: dir:/home/me/somedir schränkt die Suche auf Ergebnisse ein, die sich im Verzeichnis */home/me/somedir* befinden (einschließlich Unterverzeichnisse).

  Die Tilde-Erweiterung wird wie üblich durchgeführt. Wildcards werden expandiert, aber beachten Sie bitte eine wichtige Einschränkung von Wildcards in Pfadfiltern.

  Sie können auch relative Pfade verwenden. Zum Beispiel würde dir:share/doc entweder /usr/share/doc oder /usr/local/ share/doc entsprechen.

  -dir findet Ergebnisse, *die sich nicht* an dem angegebenen Ort befinden.

  Es können mehrere dir-Klauseln angegeben werden, sowohl positive als auch negative. Zum Beispiel ist das Folgende sinnvoll:

```
dir:recoll dir:src -dir:utils -dir:common
```

Dies würde Ergebnisse auswählen, die sowohl `recoll` als auch `src` im Pfad haben (in beliebiger Reihenfolge), und die weder `utils` noch
`gemeinsam`.

Sie können auch OR-Konjunktionen mit `dir:`-Klauseln verwenden.

Ein besonderer Aspekt von dir-Klauseln besteht darin, dass die Werte im Index nicht in UTF-8 transkodiert und niemals klein- oder kleingeschrieben werden, sondern binär gespeichert werden. Das bedeutet, dass Sie die Werte in der exakten Groß- oder Kleinschreibung eingeben müssen und dass die Suche nach Namen mit diakritischen Zeichen aufgrund von Zeichensatzkonvertierungsproblemen manchmal unmöglich ist. Nicht-ASCII-UNIX-Dateipfade sind eine nicht enden wollende Quelle von Problemen und werden am besten vermieden.

Sie müssen den Pfadwert in Anführungszeichen setzen, wenn er Leerzeichen enthält.

Die Shortcut-Syntax zur Definition von OR- oder AND-Listen innerhalb von Feldern mit Kommas oder Schrägstrichen ist nicht verfügbar.

- `Größe`, um die Ergebnisse nach der Dateigröße zu filtern. Beispiel: `Größe<10000`. Sie können `<` , `>` oder `=` als Operatoren verwenden. Sie können einen Bereich wie den folgenden angeben: `Größe>100 Größe<1000`. Die üblichen `k/K`, `m/M`, `g/G`, `t/T` können als (dezimale) Multiplikatoren verwendet werden. Beispiel: `Größe>1k` für die Suche nach Dateien, die größer als Bytes1000 sind.

- `date` zum Suchen oder Filtern nach Datumsangaben. Die Syntax für das Argument basiert auf dem ISO8601-Standard für Datumsangaben und Zeitintervalle. Es werden nur Datumsangaben unterstützt, keine Zeitangaben. Die allgemeine Syntax besteht aus Elementen, die durch ein /-Zeichen 2getrennt sind. Jedes Element kann ein Datum oder ein Zeitabschnitt sein. Zeiträume werden als `PnYnMnD` angegeben. Die *`n`* Zahlen sind die jeweilige Anzahl der Jahre, Monate oder Tage, von denen eines fehlen kann. Datumsangaben werden als *`JJJJ`* -MM -TT angegeben. Die Teile für Tage und Monate können fehlen. Wenn die `/` vorhanden ist, aber ein Element fehlt, wird das fehlende Element als das niedrigste oder höchste Datum im Index interpretiert. Beispiele:

  – `2001-03-01/2002-05-01` die grundlegende Syntax für ein Intervall von Daten.

  – `2001-03-01/P1Y2M` das gleiche mit einem Punkt angegeben.

  – `2001/` vom Beginn des bis2001 zum letzten Datum im Index.

  – `2001` das ganze Jahr über 2001

  – `P2D/` bedeutet vor Tagen2 bis heute, wenn es keine Dokumente mit Daten in der Zukunft gibt.

  – `/2003` alle Dokumente von oder2003 älter.

  Zeiträume können auch mit Kleinbuchstaben angegeben werden (z. B.: p2y).

- `mime` oder `format` zur Angabe des MIME-Typs. Diese Klauseln werden abweichend von der normalen booleschen Logik der Suche verarbeitet: mehrere Werte werden ODER-verknüpft (anstelle des normalen UND). Sie können auszuschließende Typen mit dem üblichen `-` angeben und Wildcards verwenden. Beispiel: *`mime:text/* -mime:text/plain`*. Die Angabe eines expliziten Booleschen Operators vor einer mime-Spezifikation wird nicht unterstützt und führt zu seltsamen Ergebnissen.

- `type` oder `rclcat` zur Angabe der Kategorie (wie in text/media/presentation/etc.). Die Einteilung der MIME-Typen in Kategorien wird in der Recoll-Konfiguration (`mimeconf`) festgelegt und kann geändert oder erweitert werden. Die Standard-Kategorienamen sind diejenigen, die eine Filterung der Ergebnisse im Hauptbildschirm der grafischen Benutzeroberfläche ermöglichen. Kategorien werden wie MIME-Typen mit ODER verknüpft und können mit `-` negiert werden.

- `issub`, um festzulegen, dass nur eigenständige (`issub:0`) oder nur eingebettete (`issub:1`) Dokumente als Ergebnisse zurückgegeben werden sollen.

Hinweis

Die Kriterien `mime`, `rclcat`, `size`, `issub` und `date` wirken sich immer auf die gesamte Abfrage aus (sie werden als letzter Filter angewendet), auch wenn sie mit anderen Begriffen innerhalb einer Klammer gesetzt werden.

Hinweis

`mime` (oder das entsprechende `rclcat`) ist das *einzige* Feld mit einer OR-Vorgabe. Sie müssen `OR` zum Beispiel mit ext-Begriffen verwenden.

3.5.3 Bereichsklauseln

Recoll und 1.24spätere Versionen unterstützen Bereichsklauseln für Felder, die dafür konfiguriert wurden. Kein Standardfeld verwendet sie derzeit, so dass dieser Absatz nur interessant ist, wenn Sie die Feldkonfiguration geändert haben und möglicherweise einen benutzerdefinierten Input-Handler verwenden.

Eine Bereichsklausel sieht wie eine der folgenden aus:

```
myfield:small..big
myfield:small..
myfield:..big
```

Die Art der Klausel wird durch die beiden Punkte `..` angezeigt. Sie bewirkt, dass die Ergebnisse gefiltert werden, bei denen der Wert von *myfield* in dem möglicherweise nicht abgeschlossenen Intervall liegt.

Im Abschnitt über die Feldkonfigurationsdatei finden Sie die Einzelheiten zur Konfiguration eines Feldes für die Bereichssuche (diese sind im Abschnitt [Werte] aufgeführt).

3.5.4 Modifikatoren

Einige Zeichen werden als Suchmodifikatoren erkannt, wenn sie unmittelbar nach dem schließenden Anführungszeichen einer Phrase stehen, wie in
`"irgendein Begriff"-Modifizierungszeichen`. Die eigentliche "Phrase" kann natürlich ein einzelner Begriff sein. Unterstützte Modifikatoren:

- `l` kann verwendet werden, um das Stemming auszuschalten (macht vor allem bei `p` Sinn, da das Stemming bei Phrasen standardmäßig ausgeschaltet ist).
- `s` kann verwendet werden, um die Synonymexpansion zu deaktivieren, wenn eine Synonymdatei vorhanden ist (nur für Recoll und1.22 später).
- `o` kann verwendet werden, um einen "Slack" für die Phrasen- und Proximity-Suche anzugeben: die Anzahl der zusätzlichen Begriffe, die zwischen den angegebenen Begriffen gefunden werden können. Wenn `o` von einer ganzen Zahl gefolgt wird, ist dies die Lücke, andernfalls ist der Standardwert 10.
- `p` kann verwendet werden, um die Standardsuche nach Phrasen in eine Umkreissuche (ungeordnet) umzuwandeln. Beispiel: `"bestelle irgendetwas in "p`
- `C` schaltet die Groß- und Kleinschreibung ein (sofern der Index sie unterstützt).
- `D` schaltet die Empfindlichkeit für diakritische Zeichen ein (sofern der Index dies unterstützt).
- Ein Gewicht kann für ein Abfrageelement angegeben werden, indem ein Dezimalwert am Anfang der Modifikatoren angegeben wird. Beispiel: `"Wichtig "2,5`

3.6 Wildcards und verankerte Suchen

Einige Sonderzeichen werden von Recoll in Suchstrings interpretiert, um die Suche zu erweitern oder zu spezialisieren. Wildcards erweitern einen Stammbegriff auf kontrollierte Weise. Ankerzeichen können eine Suche so einschränken, dass sie nur dann erfolgreich ist, wenn die Übereinstimmung am oder in der Nähe des Anfangs des Dokuments oder eines seiner Felder gefunden wird.

3.6.1 Wildcards

Alle Wörter, die in die Recoll-Suchfelder eingegeben werden, werden vor der endgültigen Ausführung der Anfrage auf die Verwendung von Platzhaltern geprüft. Die Platzhalterzeichen sind:

- `*`, der mit einem oder0 mehreren Zeichen übereinstimmt.
- `?` der auf ein einzelnes Zeichen passt.
- `[]`, die es ermöglichen, Zeichensätze zu definieren (z. B.: `[abc]` entspricht einem einzelnen Zeichen, das 'a' oder 'b' oder 'c' sein

kann, `[0-9]` entspricht einer beliebigen Zahl).

Bei der Verwendung von Wildcards sollten Sie einige Dinge beachten.

- Die Verwendung eines Platzhalters am Anfang eines Wortes kann zu einer langsamen Suche führen, da Recoll die gesamte Liste der in- dex Begriffe durchsuchen muss, um die Übereinstimmungen zu finden. Bei Feldsuchen ist dies jedoch weniger ein Problem, und Abfragen wie `author:*@domain.com` können manchmal sehr nützlich sein.
- Nur18 bei der Version Recoll wird bei der Arbeit mit einem Rohindex (unter Beibehaltung von Groß- und Kleinschreibung) der wörtliche Teil eines Platzhalterausdrucks genau auf Groß- und Kleinschreibung abgestimmt. Dies gilt nicht mehr für Versionen und19 spätere Versionen.
- Die Verwendung eines `*` am Ende eines Wortes kann zu mehr Übereinstimmungen führen, als Sie denken, und zu seltsamen Suchergebnissen. Mit dem Begriffsexplorer können Sie prüfen, welche Vervollständigungen für einen bestimmten Begriff existieren. Sie können auch genau sehen, welche Suche durchgeführt wurde, indem Sie auf den Link am Anfang der Ergebnisliste klicken. Bei Begriffen aus der natürlichen Sprache liefert die Stammexpansion in der Regel bessere Ergebnisse als die Endung `*` (die Stammexpansion wird ausgeschaltet, wenn ein Platzhalterzeichen im Begriff vorkommt).

3.6.1.1 Wildcards und Pfadfilterung

Aufgrund der Art und Weise, wie Recoll Wildcards in dir-Pfad-Filterklauseln verarbeitet, haben sie einen multiplikativen Effekt auf die Abfragegröße. Eine Klausel, die Wildcards in mehreren Pfadelementen enthält, wie z.B. `dir:/home/me/*/*/docdir`, wird mit ziemlicher Sicherheit fehlschlagen, wenn Ihr indizierter Baum eine realistische Größe hat.

Je nach Fall können Sie das Problem umgehen, indem Sie die Pfade enger spezifizieren, mit einem konstanten Präfix, oder indem Sie separate2 `dir:`-Klauseln anstelle von mehreren Platzhaltern verwenden, wie in `dir:/home/me dir:docdir`. Die letztgenannte Abfrage ist nicht äquivalent zu der ursprünglichen, da sie keine Anzahl von Verzeichnisebenen angibt, aber das ist das Beste, was wir tun können (und es kann in einigen Fällen tatsächlich nützlicher sein).

3.6.2 Verankerte Suche

Mit zwei Zeichen wird festgelegt, dass ein Suchtreffer am Anfang oder am Ende des Textes erfolgen soll. `^` am Anfang eines Begriffs oder einer Phrase bewirkt, dass die Suche am Anfang stattfindet, `$` am Ende erzwingt, dass sie am Ende stattfindet.

Da diese Funktion als Phrasensuche implementiert ist, ist es möglich, eine maximale Entfernung anzugeben, bei der der Treffer auftreten soll, entweder über die Steuerelemente des erweiterten Suchfensters oder über die Abfragesprache, z. B. wie in:

```
"^someterm "o10
```

was dazu führen würde, dass `someterm` innerhalb von Begriffen10 am Anfang des Textes gefunden wird. Dies kann mit einer Feldsuche wie in `somefield: "^someterm "o10` oder `somefield:someterm$` kombiniert werden.

Diese Funktion kann auch bei einer eigentlichen Phrasensuche verwendet werden, aber in diesem Fall gilt der Abstand für die gesamte Phrase und den Anker, so dass z. B. `bla bla mein unerwarteter Begriff` am Anfang des Textes eine Übereinstimmung mit `"^mein Begriff "o5` wäre.

Die verankerte Suche kann sehr nützlich sein für die Suche in einigermaßen strukturierten Dokumenten wie wissenschaftlichen Artikeln, wenn keine expliziten Metadaten zur Verfügung gestellt wurden (was sehr häufig der Fall ist), z. B. für die Suche nach Übereinstimmungen in der Zusammenfassung oder der Liste der Autoren (die am Anfang des Dokuments stehen).

3.7 Synonyme verwenden (1.22)

Begriffssynonyme und Textsuche: Im Allgemeinen gibt es zwei Hauptmöglichkeiten, Begriffssynonyme für die Textsuche zu verwenden:

- Bei der Indexerstellung können sie verwendet werden, um die indizierten Begriffe zu ändern, indem entweder ihre Anzahl erhöht oder verringert wird, indem die ursprünglichen Begriffe auf alle Synonyme erweitert werden oder indem alle synonymen Begriffe auf einen kanonischen Begriff reduziert werden.
- Bei der Abfrage können sie verwendet werden, um Texte abzugleichen, die Begriffe enthalten, die Synonyme der vom Benutzer angegebenen Begriffe sind, indem entweder die Abfrage auf alle Synonyme erweitert oder die Benutzereingabe auf kanonische Begriffe reduziert wird (letzteres funktioniert nur, wenn die entsprechende Verarbeitung bei der Erstellung des Indexes durchgeführt wurde).

Recoll verwendet Synonyme nur zur Abfragezeit. Ein Suchbegriff, der Teil einer Synonymgruppe ist, wird optional zu einem OR-Abfrage für alle Begriffe in der Gruppe.

Synonymgruppen werden in gewöhnlichen Textdateien definiert. Jede Zeile in der Datei

definiert eine Gruppe. Beispiel:

```
hallo "Guten Morgen"
# Anderenfalls bin ich mir bei "au revoir" nicht
sicher. Ist das Englisch? bye goodbye "see you" \
"Auf Wiedersehen"
```

Wie üblich sind Zeilen, die mit einem # beginnen, Kommentare, Leerzeilen werden ignoriert, und Zeilen können mit einem Backslash abgeschlossen werden.

Mehrwortsynonyme werden unterstützt, aber beachten Sie, dass diese Phrasenabfragen erzeugen, die die Leistung beeinträchtigen können und die Stemming-Erweiterung für die Phrasenbegriffe deaktivieren.

Der Inhalt der Synonymdatei muss in Großbuchstaben geschrieben werden (nicht nur in Kleinbuchstaben), da dies an der Stelle in der Abfrageverarbeitung, an der sie verwendet wird, erwartet wird. Es gibt ein paar Fälle, in denen dies einen Unterschied macht, zum Beispiel sollte das deutsche scharfe s als `ss` ausgedrückt werden, das griechische Endsigma als sigma. Als Referenz bietet Python3 eine einfache Möglichkeit zur Groß- und Kleinschreibung von Wörtern (str.casefold()).

Die Synonymdatei kann auf der Registerkarte Suchparameter des Menüeintrags Voreinstellungen der GUI-Konfiguration oder als Option für die Befehlszeilensuche angegeben werden.

Sobald die Datei definiert ist, kann die Verwendung von Synonymen direkt über das Menü Einstellungen aktiviert oder deaktiviert werden.

Die Synonyme werden nach Übereinstimmungen mit Benutzerbegriffen durchsucht, nachdem letztere stammexpandiert wurden, aber der Inhalt der Synonymdatei selbst wird nicht der Stammexpansion unterzogen. Das bedeutet, dass eine Übereinstimmung nicht gefunden wird, wenn die in der Synonymdatei enthaltene Form nirgendwo im Dokumentensatz vorkommt (dasselbe gilt für Akzente bei Verwendung eines Rohindex).

Die Funktion Synonyme wird Ihnen wahrscheinlich nicht helfen, Ihre Briefe an Mr. Smith zu finden. Sie ist am besten für bereichsspezifische Suchen geeignet. Zum Beispiel wurde sie ursprünglich von einem Benutzer vorgeschlagen, der in historischen Dokumenten suchte: Die Synonymdatei würde Spitznamen und Aliasnamen für jede der interessierenden Personen enthalten.

3.8 Pfadübersetzungen

In einigen Fällen stimmen die im Index gespeicherten Dokumentpfade nicht mit den tatsächlichen Pfaden überein, so dass die Vorschau und der Zugriff auf die Dokumente fehlschlagen. Dies kann in einer Reihe von Fällen auftreten:

- Bei der Verwendung mehrerer Indizes kommt es relativ häufig vor, dass sich einige davon auf einem entfernten Datenträger befinden, der z. B. über NFS gemountet ist. In diesem Fall sind die Pfade, die für den Zugriff auf die Dokumente auf dem lokalen Rechner verwendet werden, nicht notwendigerweise die gleichen wie die, die beim Indizieren auf dem entfernten Rechner verwendet werden. Beispielsweise kann `/home/me` während der Indizierung als topdirs-Element verwendet worden sein, aber das Verzeichnis könnte auf dem lokalen Rechner als `/net/server/home/me` gemountet sein.
- Dieser Fall kann auch bei Wechseldatenträgern auftreten. Es ist durchaus möglich, einen Index so zu konfigurieren, dass er mit den Dokumenten auf dem Wechseldatenträger zusammenlebt, aber es kann vorkommen, dass der Datenträger nicht an der gleichen Stelle eingehängt ist, so dass die Dokumentpfade aus dem Index ungültig sind.
- Als letztes Beispiel könnte man sich vorstellen, dass ein großes Verzeichnis verschoben wurde, es aber derzeit nicht möglich ist, den Indexer zu starten.

Recoll verfügt über eine Funktion zum Umschreiben der Zugriffspfade beim Extrahieren der Daten aus dem Index. Die Übersetzungen können für den Hauptindex und für jeden zusätzlichen Abfrageindex definiert werden.

Die Möglichkeit der Pfadübersetzung ist immer dann nützlich, wenn die vom Indexierer gesehenen Dokumentpfade nicht mit den Pfaden übereinstimmen, die bei der Abfrage verwendet werden sollen.

Im obigen NFS-Beispiel könnte Recoll angewiesen werden, jede `file:///home/me`-URL vom Index in die `Datei` umzuschreiben: `///net/server/home/me`, um Zugriffe vom Client aus zu ermöglichen.

Die Übersetzungen werden in der ptrans-Konfigurationsdatei definiert, die von Hand oder über den Konfigurationsdialog der GUI für externe Indizes bearbeitet werden kann: Einstellungen Dialog Externer Index, dann klicken Sie auf die Schaltfläche Pfadübersetzungen rechts unterhalb der Indexliste.

Hinweis
Aufgrund eines aktuellen Fehlers muss die GUI neu gestartet werden, nachdem die ptrans-Werte geändert wurden (auch wenn sie von der GUI aus geändert wurden).

3.9 Groß-/Kleinschreibung und diakritische Zeichen berücksichtigen

Bei Recoll-Versionen und1.18 neueren Versionen sowie *bei der Arbeit mit einem Rohindex* (nicht die Standardeinstellung) kann die Suche auf Groß- und Kleinschreibung reagieren. Wie dies geschieht, wird durch Konfigurationsvariablen und die eingegebenen Suchdaten gesteuert.

Die allgemeine Standardeinstellung ist, dass bei der Eingabe von Suchbegriffen ohne Großbuchstaben oder Akzente Groß- und Kleinschreibung nicht beachtet werden. Die Eingabe von `"Lebenslauf"` entspricht allen Suchbegriffen wie `"Lebenslauf"`, `"Lebenslauf"`, `"Resümee"`, `"Lebenslauf"` usw.

Zwei Konfigurationsvariablen können das Einschalten der Empfindlichkeit automatisieren (sie waren zwar dokumentiert, haben aber bis Recoll 1.22 nichts bewirkt):

autodiacsens Wenn diese Variable gesetzt ist, wird die Suchempfindlichkeit für diakritische Zeichen eingeschaltet, sobald ein akzentuiertes Zeichen in einem Suchbegriff vorkommt. Wenn die Variable auf true gesetzt ist, startet `resume` eine diakritik-unempfindliche Suche, aber `résumé wird` genau abgeglichen. Der Standardwert ist *false*.

autocasesens Wenn diese Variable gesetzt ist, wird die Groß- und Kleinschreibung bei der Suche berücksichtigt, sobald ein Großbuchstabe in einem Suchbegriff vorkommt, *mit Ausnahme des ersten Zeichens*. Wenn die Variable auf true gesetzt ist, wird mit `us` oder `Us` eine Suche ohne Berücksichtigung von Diakritika gestartet, aber `US` wird genau abgeglichen. Der Standardwert ist *true* (im Gegensatz zu `autodiacsens`).

Wie in der Vergangenheit schaltet die Großschreibung des ersten Buchstabens eines Wortes die Wortstamm-Erweiterung aus und hat keinen Einfluss auf die Groß-/Kleinschreibung.

Sie können die Unterscheidung zwischen Groß- und Kleinschreibung auch explizit aktivieren, indem Sie Modifikatoren in der Abfragesprache verwenden. `Mit C` wird die Groß- und Kleinschreibung beachtet, `mit D` die diakritischen Zeichen. Beispiele:

```
"uns "C
```

sucht genau nach dem Begriff `us` (`Us` ist keine Übereinstimmung).

```
"Lebenslauf "D
```

sucht genau nach dem Begriff `"Lebenslauf"` (`"résumé"` ist kein Treffer).

Wenn entweder die Groß- und Kleinschreibung oder die diakritischen Zeichen aktiviert sind, ist die Stammerweiterung ausgeschaltet. Beides zu haben, ist nicht sehr sinnvoll.

3.10 Desktop-Integration

Die Unabhängigkeit vom Desktop-Typ hat ihre Nachteile: Die Desktop-Integration von Recoll ist minimal. Es sind jedoch einige Tools verfügbar:

- Benutzer neuerer, von Ubuntu abgeleiteter Distributionen oder anderer Gnome-Desktop-Systeme (z.B. Fedora) können den Recoll GSSP (Gnome Shell Search Provider) installieren.
- Der KDE-KIO-Slave wurde in einem früheren Abschnitt beschrieben. Er kann Suchergebnisse innerhalb von **Dolphin** liefern.
- Wenn Sie eine ältere Version von Ubuntu Linux verwenden, finden Sie vielleicht das Ubuntu Unity Lens Modul nützlich.
- Es gibt auch ein unabhängig entwickeltes Krunner-

Plugin. Hier folgen ein paar andere Dinge, die helfen

können.

3.10.1 Hotkeys für die Aufzeichnung

Es ist überraschend praktisch, die Benutzeroberfläche von Recoll mit einem einzigen Tastendruck ein- und ausblenden zu können. Recoll wird mit einem kleinen Python-Skript ausgeliefert, das auf der libwnck-Fenstermanager-Schnittstellenbibliothek basiert und Ihnen genau dies ermöglicht. Die detaillierte Anleitung finden Sie auf dieser Wiki-Seite.

3.10.2 Das KDE-Kicker-Recoll-Applet

Das ist jetzt wahrscheinlich überholt. Wie auch immer:

Der Recoll-Quellbaum enthält den Quellcode für das recoll_applet, eine kleine Anwendung, die vom find_applet abgeleitet ist. Diese kann verwendet werden, um dem KDE-Panel einen kleinen Recoll-Launcher hinzuzufügen.

Das Applet wird nicht automatisch mit den Hauptprogrammen von Recoll gebaut und ist auch nicht in der Hauptquelldistribution enthalten (weil die KDE-Build-Boilerplate es relativ groß macht). Sie können den Quellcode von der recoll.org Download-Seite herunterladen. Verwenden Sie die allmächtige **`configure;make;make`** install-Beschwörung zum Bauen und Installieren.

Sie können dann das Applet zum Panel hinzufügen, indem Sie mit der rechten Maustaste auf das Panel klicken und den Eintrag Applet hinzufügen wählen.

Das recoll_applet hat ein kleines Textfenster, in das Sie eine Recoll-Abfrage (in Form einer Abfragesprache) eingeben können, und ein Symbol, mit dem Sie die Suche auf bestimmte Dateitypen beschränken können. Es ist recht primitiv und startet jedes Mal eine neue recoll-GUI-Instanz (auch wenn sie bereits läuft). Vielleicht finden Sie es trotzdem nützlich.

Kapitel 4

Programmierschnittstelle

Recoll verfügt über eine Anwendungsprogrammierschnittstelle, die sowohl für die Indizierung als auch für die Suche verwendet werden kann und derzeit über die Sprache Python zugänglich ist.

Eine andere, weniger radikale Möglichkeit, die Anwendung zu erweitern, besteht darin, Input-Handler für neue Dokumenttypen zu schreiben. Die Verarbeitung von Metadatenattributen für Dokumente (`Felder`) ist in hohem Maße konfigurierbar.

4.1 Schreiben eines Dokumenteneingabe-Handlers

Terminologie
Die kleinen Programme oder Codestücke, die die Verarbeitung der verschiedenen Dokumenttypen für Recoll übernehmen, wurden früher `Filter` genannt, was sich auch im Namen des Verzeichnisses, in dem sie sich befinden, und in vielen Konfigurationsvariablen widerspiegelt. Sie wurden so benannt, weil eine ihrer Hauptfunktionen darin besteht, die Formatierungsanweisungen herauszufiltern und den Textinhalt beizubehalten. Diese Module können jedoch auch andere Funktionen haben, so dass in der Dokumentation nach und nach der Begriff `input handler verwendet wird. filter` wird jedoch immer noch an vielen Stellen verwendet.

Recoll-Eingabehandler arbeiten zusammen, um die Vielzahl von Eingabedokumentformaten, einfache wie Opendocument, Acrobat oder zusammengesetzte wie Zip oder E-Mail, in das endgültige Recoll-Indizierungs-Eingabeformat zu übersetzen, das reiner Text ist (in vielen Fällen hat die Verarbeitungspipeline einen zwischengeschalteten HTML-Schritt, der für eine bessere Vorschaudarstellung verwendet werden kann). Die meisten Input-Handler sind ausführbare Programme oder Skripte. Einige wenige Handler sind in C++ kodiert und befinden sich in **recollindex**. Diese letztere Art wird hier nicht beschrieben.

Es gibt zwei Arten von externen ausführbaren Eingabehandlern:

- Einfache exec-Handler werden einmal ausgeführt und beendet. Es kann sich dabei um reine Programme wie **antiword** oder um Skripte handeln, die andere Programme verwenden. Sie sind sehr einfach zu schreiben, da sie nur das konvertierte Dokument auf der Standardausgabe ausgeben müssen. Die Ausgabe kann als einfacher Text oder HTML erfolgen. HTML wird in der Regel bevorzugt, da es Metadatenfelder speichern kann und es erlaubt, einige der Formatierungen für die GUI-Vorschau beizubehalten. Diese Handler haben jedoch Einschränkungen:
  - Sie können nur ein Dokument pro Datei verarbeiten.
  - Der MIME-Typ der Ausgabe muss bekannt und festgelegt sein.
  - Die Zeichenkodierung, sofern relevant, muss bekannt und festgelegt sein (oder möglicherweise nur vom Standort abhängen).
- Mehrere execm-Handler können mehrere Dateien verarbeiten (was die Startzeit des Prozesses erspart, die sehr hoch sein kann), oder mehrere Dokumente pro Datei (z. B. für Archive oder Publikationen mit mehreren Kapiteln). Sie kommunizieren mit dem Indexer über ein einfaches Protokoll, sind aber dennoch etwas komplizierter als die ältere Art. Die meisten der neuen Handler sind in Python geschrieben (Ausnahme: **rclimg**, das in Perl geschrieben ist, da `exiftool` keine echte Python-Entsprechung hat). Die Python-Handler verwenden allgemeine Module, um das Boilerplate zu eliminieren, was sie in günstigen Fällen sehr einfach machen kann. Die von diesen Handlern ausgegebenen Unterdokumente können direkt indizierbar sein (Text oder HTML), oder sie können andere einfache oder zusammengesetzte Dokumente sein, die von einem anderen

Handler verarbeitet werden müssen.

In beiden Fällen befassen sich die Handler mit normalen Dateisystemdateien und können entweder ein einzelnes Dokument oder eine lineare Liste von Dokumenten in jeder Datei verarbeiten. Recoll ist für die Durchführung von Aktualitätsprüfungen, die Behandlung komplexerer Einbettungen und andere übergeordnete Probleme zuständig.

Ein einfacher Handler, der ein Dokument im text/plain-Format zurückgibt, kann keine Metadaten an den Indexer übertragen. Allgemeine Metadaten, wie Dokumentgröße oder Änderungsdatum, werden vom Indexer erfasst und gespeichert.

Handler, die das Format `text/html` erzeugen, können eine beliebige Menge von Metadaten in HTML-Meta-Tags zurückgeben. Diese werden gemäß den Richtlinien in der Konfigurationsdatei für Felder verarbeitet.

Die Handler, die mehrere Dokumente pro Datei verarbeiten können, geben ein einzelnes Datenelement zurück, um jedes Dokument in der Datei zu identifizieren. Diese Daten, die als `ipath` bezeichnet werden, werden von Recoll zurückgesendet, um das Dokument bei der Abfrage zu extrahieren, eine Vorschau anzuzeigen oder eine temporäre Datei zu erstellen, die von einem Viewer geöffnet werden kann. Diese Handler können auch Metadaten entweder als HTML-Meta-Tags oder als benannte Daten über das Kommunikationsprotokoll zurückgeben.

Der folgende Abschnitt beschreibt die einfachen Handler, und der nächste Abschnitt enthält einige Erläuterungen zu den execm-Handlern. Es ist denkbar, dass Sie einen einfachen Handler schreiben können, der nur die Elemente aus dem Handbuch enthält. Dies gilt nicht für die anderen Handler, für die Sie sich den Code ansehen müssen.

4.1.1 Einfache Eingabehandler

Die einfachen Handler von Recoll sind in der Regel Shell-Skripte, aber das ist keineswegs notwendig. Das Extrahieren des Textes aus dem nativen Format ist der schwierige Teil. Das Ausgeben des von Recoll erwarteten Formats ist trivial. Glücklicherweise verfügen die meisten Dokumentformate über Übersetzer oder Textextraktoren, die vom Handler aus aufgerufen werden können. In einigen Fällen ist die Ausgabe des übersetzenden Programms völlig angemessen, und es wird kein dazwischenliegendes Shell-Skript benötigt.

Input-Handler werden mit einem einzigen Argument aufgerufen, nämlich dem Namen der Quelldatei. Sie sollten das Ergebnis auf stdout ausgeben.

Wenn Sie einen Handler schreiben, sollten Sie entscheiden, ob er reinen Text oder HTML ausgeben soll. Normaler Text ist einfacher, aber Sie können keine Metadaten hinzufügen oder die Zeichenkodierung der Ausgabe ändern (dies wird in einer Konfigurationsdatei festgelegt). Außerdem können einige Formatierungen bei der HTML-Vorschau leichter beibehalten werden. Der entscheidende Faktor sind also die Metadaten: Recoll bietet eine Möglichkeit, Metadaten aus dem HTML-Header zu extrahieren und sie für die Feldsuche zu verwenden.

Die Umgebungsvariable `RECOLL_FILTER_FORPREVIEW` (Werte `yes`, `no`) teilt dem Handler mit, ob der Vorgang zur Indizierung oder zur Vorschau dient. Einige Handler verwenden dies, um ein leicht abweichendes Format auszugeben, z. B. um uninteressante wiederholte Schlüsselwörter (z. B.: `Betreff:` für E-Mail) bei der Indizierung zu entfernen. Dies ist nicht unbedingt erforderlich.

Sie sollten sich einen der einfachen Handler ansehen, z.B. **rclps**, um einen Anfang zu machen. Vergessen Sie nicht, Ihren Handler vor dem Testen ausführbar zu machen!

4.1.2 "Mehrere" Bearbeiter

Wenn Sie programmieren können und einen execm-Handler schreiben wollen, sollte es nicht allzu schwierig sein, einen der vorhandenen Handler sinnvoll zu nutzen.

Die vorhandenen Handler unterscheiden sich durch die Menge an Hilfscode, den sie verwenden:

- `rclimg` ist in Perl geschrieben und wickelt das execm-Protokoll ganz allein ab (was zeigt, wie trivial es ist).
- Alle Python-Handler nutzen zumindest das Modul `rclexecm.py`, das die Kommunikation abwickelt. Schauen Sie sich z.B. `rclzip` an, ein Handler, der `rclexecm.py` direkt verwendet.
- Die meisten Python-Handler, die einzelne Dokumentdateien durch die Ausführung eines anderen Befehls verarbeiten, werden durch Verwendung des Moduls `rclexec1.py` weiter abstrahiert. Siehe z.B. `rclrtf.py` für einen einfachen Befehl oder `rcldoc.py` für einen etwas komplizierteren Befehl (der möglicherweise mehrere Befehle ausführt).
- Handler, die Text aus einem XML-Dokument mit Hilfe eines XSLT-Stylesheets extrahieren, werden jetzt innerhalb von **recollindex** ausgeführt, wobei nur das Stylesheet im Verzeichnis `filters/` gespeichert wird. Diese können ein einziges Stylesheet (z.B. `abiword.xsl`) oder zwei Sheets für die Daten und Metadaten (z.B. `opendoc-body.xsl` und `opendoc-`

`meta.xsl`) verwenden. Die mimeconf-Konfigurationsdatei legt fest, wie die Sheets verwendet werden, sehen Sie sich das an. Vor dem C++-Import benutzten die xsl-basierten Handler ein gemeinsames Modul `rclgenxslt.py`, das immer noch existiert, aber im Moment nicht benutzt wird. Der Handler für OpenXML-Präsentationen ist immer noch die Python-Version, weil das Format nicht zu dem passt, was der C++-Code tut. Es wäre eine gute Basis für ein weiteres ähnliches Thema.

Es gibt ein Beispiel für einen trivialen Handler, der auf `rclexecm.py` basiert, mit vielen Kommentaren, die von Recoll nicht verwendet werden. Er würde eine Textdatei als ein Dokument pro Zeile indizieren. Suchen Sie nach `rcltxtlines.py` im Verzeichnis `src/filters` im Git-Repository von Recoll (das Beispiel ist derzeit nicht in der verteilten Version enthalten).

Sie können auch einen Blick auf das etwas komplexere **rclzip** werfen, das Zip-Dateipfade als Bezeichner (`ipath`) verwendet.

execm-Handler müssen manchmal eine Entscheidung über die Art der ipath-Elemente treffen, die sie bei der Kommunikation mit dem Indexer verwenden. Hier sind ein paar Richtlinien:

- Verwenden Sie ASCII oder UTF-8 (wenn der Bezeichner eine ganze Zahl ist, drucken Sie ihn z. B. wie printf %d es tun würde).
- Wenn möglich, sollten die Daten einen Sinn ergeben, wenn sie in eine Protokolldatei ausgegeben werden, um die Fehlersuche zu erleichtern.
- Recoll verwendet einen Doppelpunkt (`:`) als Trennzeichen, um einen komplexen Pfad intern zu speichern (für eine tiefere Einbettung). Doppelpunkte innerhalb der ipath-Elemente, die von einem Handler ausgegeben werden, werden escaped, wären aber eine schlechte Wahl als Handler-spezifisches Trennzeichen (hauptsächlich wieder aus Gründen der Fehlersuche).

In jedem Fall sollte es für den Handler einfach sein, das Zieldokument zu extrahieren, wenn der Dateiname und die ipath-Element.

execm-Handler erzeugen auch ein Dokument mit einem Null-ipath-Element. Je nach Art des Dokuments kann dieses Dokument mit Daten verknüpft sein (z. B. der Textkörper einer E-Mail-Nachricht) oder auch nicht (typisch für eine Archivdatei). Wenn es leer ist, ist dieses Dokument trotzdem für einige Operationen nützlich, da es das übergeordnete Dokument der eigentlichen Datendokumente ist.

4.1.3 Informieren von Recoll über den Handler

Es gibt zwei Elemente, die eine Datei mit dem Handler verbinden, der sie verarbeiten soll: die Zuordnung einer Datei zu einem MIME-Typ und die Zuordnung eines MIME-Typs zu einem Handler.

Die Zuordnung von Dateien zu MIME-Typen erfolgt meist auf der Grundlage von Namenssuffixen. Die Typen werden in der mimemap-Datei definiert. Beispiel:

```
.doc = Anwendung/msword
```

Wenn keine Suffix-Zuordnung für den Dateinamen gefunden wird, versucht Recoll, einen Systembefehl auszuführen (typischerweise **file -i** oder **xdg-mime**), um einen MIME-Typ zu bestimmen.

Das zweite Element ist die Zuordnung von MIME-Typen zu Handlern in der Datei mimeconf. Ein Beispiel ist wahrscheinlich besser als eine lange Erklärung:

```
[index]
        application/msword = exec antiword -t -i -m1 UTF-8;\
        mimetype = text/plain; charset=utf-8

        Anwendung/ogg = exec rclogg

        text/rtf = exec unrtf --nopict --html; charset=iso-8859-1;

        mimetype=text/html application/x-chm = execm rclchm
```

Das Fragment legt fest, dass:

- application/msword-Dateien werden verarbeitet, indem das Programm **antiword** ausgeführt wird, das `text/plain` kodiert ausgibt in `utf-8`.

- application/ogg-Dateien werden vom Skript **rclogg** verarbeitet, mit dem Standard-Ausgabetyp (`text/html`, mit der im Header angegebenen Kodierung, oder standardmäßig `utf-8`).
- `text/rtf` wird von **unrtf** verarbeitet, das `text/html` ausgibt. Die Kodierung `iso-8859-1` wird angegeben, weil sie nicht der utf-8-Standard, und nicht von **unrtf** im HTML-Header-Bereich ausgegeben.
- `application/x-chm` wird von einem persistenten Handler verarbeitet. Dies wird durch das Schlüsselwort `execm` bestimmt.

4.1.4 Eingabe-Handler-Ausgabe

Sowohl der einfache als auch der persistente Input-Handler können jeden MIME-Typ an Recoll zurückgeben, das die Daten entsprechend der MIME-Konfiguration weiterverarbeitet.

Die meisten Eingabefilter liefern entweder text/plain- oder text/html-Daten. Es gibt Ausnahmen, z. B. Filter, die Archivdateien (`zip`, `tar` usw.) verarbeiten, geben die Dokumente normalerweise so zurück, wie sie gefunden wurden, ohne sie weiter zu verarbeiten.

Es gibt nichts über die text/plain-Ausgabe zu sagen, außer dass ihre Zeichenkodierung mit der in der mimeconf-Datei angegebenen übereinstimmen sollte.

Bei Filtern, die HTML erzeugen, könnte die Ausgabe sehr minimal sein, wie im folgenden Beispiel:

```
<html>
<kopf>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
</kopf>
<body>
Einfache Textinhalte
</body>
</html>
```

Sie sollten darauf achten, dass einige Zeichen innerhalb des Textes durch Umwandlung in geeignete Einheiten entfallen. Zumindest sollte "&" in "&" und "<" in "<" umgewandelt werden. Dies wird von externen Hilfsprogrammen, die HTML ausgeben, nicht immer richtig gemacht, und natürlich auch nicht von denen, die reinen Text ausgeben.

Bei der Kapselung von einfachem Text in einem HTML-Textkörper kann die Anzeige einer Vorschau verbessert werden, indem der Text in <pre>-Tags eingeschlossen wird.

Der Zeichensatz muss in der Kopfzeile angegeben werden. Er muss nicht UTF-8 sein (Recoll kümmert sich um die Übersetzung), aber er muss genau sein, um gute Ergebnisse zu erzielen.

Recoll verarbeitet Meta-Tags in der Kopfzeile als mögliche Kandidaten für Dokumentfelder. Dokumentenfelder können vom Indexer auf unterschiedliche Weise verarbeitet werden, um sie in den Suchergebnissen zu suchen oder anzuzeigen. Dies wird in einem der folgenden Abschnitte beschrieben.

Standardmäßig verarbeitet der Indexer die Standard-Header-Felder, wenn sie vorhanden sind: `title`, `meta/description` und `meta/keyword` werden sowohl indexiert als auch für die Anzeige bei Abfragen gespeichert.

Ein vordefiniertes, nicht standardisiertes Meta-Tag wird von Recoll ebenfalls ohne weitere Konfiguration verarbeitet: Wenn ein Datums-Tag vorhanden ist und das richtige Format hat, wird es als Dokumentdatum (für die Anzeige und Sortierung) verwendet, anstelle des Änderungsdatums der Datei. Das Datumsformat sollte wie folgt sein:

Beispiel:

```
<meta name="Datum" content="JJJJ-mm-tt
HH:MM:SS"> oder
<meta name="Datum" content="JJJJ-mm-ddTHH:MM:SS">
```

```
<meta name="date" content="2013-02-24 17:50:00">
```

Input-Handler haben auch die Möglichkeit, Feldnamen zu "erfinden". Dies sollte auch als Meta-Tags ausgegeben werden:

```
        <meta name="somefield" content="Einige textuelle Daten" />
```

Sie können HTML-Markup in den Inhalt von benutzerdefinierten Feldern einbetten, um die Anzeige in Ergebnislisten zu verbessern. Fügen Sie in diesem Fall ein (wildes, nicht standardisiertes) Markup-Attribut hinzu, um Recoll mitzuteilen, dass der Wert HTML ist und für die Anzeige nicht escaped werden soll.

```
        <meta name="somefield" markup="html" content="Einige <i>textuelle</i> Daten" />
```

Wie oben geschrieben, wird die Verarbeitung der Felder in einem weiteren Abschnitt beschrieben.

Persistente Filter können eine andere, wahrscheinlich einfachere Methode zur Erzeugung von Metadaten verwenden, indem sie die Hilfsmethode `setfield()` aufrufen. Dies vermeidet die Notwendigkeit, HTML zu erzeugen, und jedes Problem mit HTML-Zitaten. Siehe zum Beispiel `rclaudio` in Recoll und1.23 später für ein Beispiel eines Handlers, der `text/plain` ausgibt und `setfield()` zur Erzeugung von Metadaten verwendet.

4.1.5 Seitenzahlen

Der Indexer interpretiert ^L-Zeichen in der Ausgabe des Handlers als Hinweis auf Seitenumbrüche und zeichnet sie auf. Bei der Abfrage kann so ein Viewer auf der richtigen Seite für einen Treffer oder ein Snippet gestartet werden. Derzeit erzeugen nur die PDF-, Postscript- und DVI-Handler Seitenumbrüche.

4.2 Verarbeitung von Felddaten

`Felder` sind benannte Informationen in oder über Dokumente, wie `Titel`, `Autor`, `Zusammenfassung`.

Die Feldwerte für Dokumente können während der Indizierung auf verschiedene Weise erscheinen: entweder werden sie von Input-Handlern als Meta-Felder im HTML-Header-Bereich ausgegeben, oder sie werden aus den erweiterten Attributen der Datei extrahiert, oder sie werden als Attribute des Doc-Objekts hinzugefügt, wenn die API verwendet wird, oder sie werden wiederum intern von Recoll synthetisiert.

Die Recoll-Abfragesprache ermöglicht die Suche nach Text in einem bestimmten Feld.

Recoll definiert eine Reihe von Standardfeldern. Zusätzliche Felder können von Handlern ausgegeben und in der Konfigurationsdatei `fields` beschrieben werden.

Felder können sein:

- `indiziert`, d.h. ihre Begriffe werden separat in invertierten Listen (mit einem bestimmten Präfix) gespeichert, und eine feldspezifische Suche ist möglich.
- `gespeichert`, d. h. ihr Wert wird im Indexdatensatz des Dokuments gespeichert und kann mit den Suchergebnissen zurückgegeben und angezeigt werden.

Ein Feld kann entweder indiziert oder gespeichert werden. Diese und andere Aspekte der Handhabung von Feldern werden innerhalb der `Felder` definiert
Konfigurationsdatei.

Einige Felder unterstützen auch Bereichsabfragen, was bedeutet, dass die Ergebnisse für ein Intervall von Werten ausgewählt werden können. Weitere Einzelheiten finden Sie im Abschnitt "Konfiguration".

Die Abfolge der Ereignisse bei der Feldbearbeitung ist wie folgt

- Während der Indizierung durchsucht **recollindex** alle Meta-Felder in HTML-Dokumenten (die meisten Dokumenttypen werden irgendwann in HTML umgewandelt). Es vergleicht den Namen jedes Elements mit der Konfiguration, die festlegt, was mit den Feldern geschehen soll (die Felddatei)
- Wenn der Name des Meta-Elements mit dem eines Feldes übereinstimmt, das indiziert werden soll, wird der Inhalt verarbeitet und die Begriffe werden mit dem in der Felddatei definierten Präfix in den Index aufgenommen.

- Wenn der Name des Meta-Elements mit dem eines Feldes übereinstimmt, das gespeichert werden soll, wird der Inhalt des Elements mit dem Dokumentdatensatz gespeichert, aus dem er zur Abfragezeit extrahiert und angezeigt werden kann.
- Wenn eine Feldsuche durchgeführt wird, wird zum Zeitpunkt der Abfrage das Indexpräfix berechnet, und der Abgleich wird nur mit Begriffen im Index durchgeführt, denen ein entsprechendes Präfix zugeordnet ist.
- Zur Abfragezeit kann das Feld innerhalb der Ergebnisliste angezeigt werden, indem die entsprechende Direktive in der Definition des Absatzformats der Ergebnisliste verwendet wird. Alle Felder werden auf dem Feldbildschirm des Vorschaufensters angezeigt (das Sie über das Rechtsklickmenü erreichen können). Dies ist unabhängig davon, ob die Suche, die zu den Ergebnissen geführt hat, das Feld verwendet hat oder nicht.

Weitere Informationen finden Sie im Abschnitt über die Felddatei oder in den Kommentaren innerhalb der Datei.

Sie können sich auch das Beispiel im FAQ-Bereich ansehen, in dem beschrieben wird, wie man pdf-Dokumenten ein Feld für *die Seitenzahl* hinzufügen kann, das in Ergebnislisten angezeigt wird.

4.3 Python-API

4.3.1 Einführung

Die Python-Programmierschnittstelle von Recoll kann sowohl für die Suche als auch für die Erstellung/Aktualisierung eines Index verwendet werden. Bindungen existieren für Python2 und Python3 (Jan 2021: Python2-Unterstützung wird bald eingestellt).

Die Suchschnittstelle wird in einer Reihe aktiver Projekte verwendet: dem Recoll Gnome Shell Search Provider , der Recoll Web UI und dem upmpdcli UPnP Media Server, zusätzlich zu vielen kleinen Skripten.

Der Abschnitt zur Indexaktualisierung der API kann verwendet werden, um Recoll-Indizes für bestimmte Konfigurationen zu erstellen und zu aktualisieren (unabhängig von den durch **recollindex** erstellten Indizes). Die resultierenden Datenbanken können allein oder in Verbindung mit regulären Datenbanken über die grafische Benutzeroberfläche oder eine der Abfrage-Schnittstellen abgefragt werden.

Die Such-API ist nach der Versionsspezifikation2.0 der Python-Datenbank-API modelliert (frühe Versionen verwendeten die

Versionsspezifikation1.0). Das Paket `recoll` enthält zwei Module:

- Das Modul `recoll` enthält Funktionen und Klassen, die zur Abfrage (oder Aktualisierung) des Index verwendet werden.
- Das Modul `rclextract` enthält Funktionen und Klassen, die zur Abfragezeit verwendet werden, um auf Dokumentdaten zuzugreifen. Das Modul `recoll` muss importiert werden, bevor `rclextract`

Es ist gut möglich, dass Ihr System-Repository Pakete für die Recoll-Python-API enthält, manchmal in einem vom Hauptpaket getrennten Paket (vielleicht mit einem Namen wie python-recoll). Andernfalls lesen Sie das Kapitel Bauen aus dem Quellcode.

Als Einführung führt das folgende kleine Beispiel eine Abfrage durch und listet den Titel und die URL für jedes der Ergebnisse auf. Das Quellverzeichnis `python/ samples` enthält mehrere Beispiele für die Python-Programmierung mit Recoll, in denen die Erweiterung und insbesondere ihre Datenextraktionsfunktionen ausführlicher behandelt werden.

```
#!/usr/bin/python3
von recoll importieren recoll

db =
recoll.connect()
abfrage =
db.abfrage()
nres = query.execute("some
query") results =
query.fetchmany(20)
für doc in results:
    print("%s %s" % (doc.url, doc.title))
```

Sie können auch einen Blick auf den Quellcode für die Recoll WebUI, den lokalen Medienserver upmpdcli oder den Gnome Shell Search Provider werfen.

4.3.2 Elemente der Schnittstelle

Einige Elemente der Schnittstelle sind spezifisch und bedürfen einer Erklärung.

ipath Dieser Datenwert (als Feld im Doc-Objekt festgelegt) wird zusammen mit der URL gespeichert, aber nicht von Recoll indiziert. Sein Inhalt wird von der Indexschicht nicht interpretiert, und seine Verwendung bleibt der Anwendung überlassen. Der Recoll-Dateisystem-Indexer verwendet beispielsweise `ipath`, um den Teil des Dokumentenzugriffspfads zu speichern, der in (möglicherweise verschachtelten) Containerdokumenten enthalten ist. `ipath` ist in diesem Fall ein Vektor von Zugriffselementen (z.z. B. könnte der erste Teil ein Pfad innerhalb einer zip-Datei zu einem Archivmitglied sein, das zufällig eine mbox-Datei ist, das zweite Element wäre die fortlaufende Nummer der Nachricht innerhalb der mbox usw.). `url` und `ipath` werden in jedem Suchergebnis zurückgegeben und definieren den Zugriff auf das Originaldokument. `ipath` ist leer für Dokumente/Dateien der obersten Ebene (z. B. ein PDF-Dokument, das eine Dateisystemdatei ist). Die Recoll-GUI kennt die Struktur der ipath-Werte, die vom Dateisystem-Indexer verwendet werden, und nutzt sie für Funktionen wie das Öffnen des übergeordneten Dokuments eines bestimmten Dokuments.

udi Ein `udi` (unique document identifier) identifiziert ein Dokument. Aufgrund von Beschränkungen innerhalb der Indexmaschine ist er in seiner Länge begrenzt (auf Bytes200), weshalb ein regulärer URI nicht verwendet werden kann. Die Struktur und der Inhalt des `udi` wird von der Anwendung definiert und ist für die Indexmaschine undurchsichtig. Der interne Dateisystem-Indexer verwendet beispielsweise den vollständigen Dokumentpfad (Dateipfad + interner Pfad), der auf die Länge gekürzt wird, wobei der unterdrückte Teil durch einen Hash-Wert ersetzt wird. Das udi ist nicht explizit in der Abfrageschnittstelle enthalten (es wird "unter der Haube" vom rclextract-Modul verwendet), aber es ist ein explizites Element der Aktualisierungsschnittstelle.

parent_udi Wenn dieses Attribut bei der Aufnahme eines Dokuments in den Index gesetzt wird, bezeichnet es sein physisches Containerdokument. In einer mehrstufigen Hierarchie muss dies nicht das unmittelbare Elterndokument sein. `parent_udi` ist optional, aber seine Verwendung durch einen Indexer kann die Indexpflege vereinfachen, da Recoll automatisch alle durch `parent_udi == udi` definierten Kinder löscht, wenn das durch udi bezeichnete Dokument zerstört wird. Wenn ein Zip-Archiv Einträge enthält, die selbst Container sind, wie z.B. mbox-Dateien, würden alle Unterdokumente innerhalb der Zip-Datei (mbox, Nachrichten, Nachrichtenanhänge usw.) denselben `parent_udi` haben, der mit dem `udi` für die Zip-Datei übereinstimmt, und alle würden zerstört werden, wenn die Zip-Datei (identifiziert durch ihren `udi`) aus dem Index entfernt wird. Der Standard-Dateisystem-Indexer verwendet `parent_udi`.

Gespeicherte und indizierte Felder Die Felddatei in der Recoll-Konfiguration definiert, welche Dokumentfelder entweder `indiziert` (durchsuchbar), `gespeichert` (mit Suchergebnissen abrufbar) oder beides sind. Abgesehen von einigen standardmäßigen/internen Feldern sind nur die `gespeicherten` Felder über die Python-Suchschnittstelle abrufbar.

4.3.3 Protokollmeldungen für Python-Skripte

Zwei spezifische Konfigurationsvariablen: `pyloglevel` und `pylogfilename` ermöglichen es, die allgemeinen Werte für Python-Programme zu überschreiben. Setzen Sie `pyloglevel` auf, um die Standard-Startmeldungen zu 2unterdrücken (die auf Stufe 3 ausgegeben werden).

4.3.4 Python-Suchschnittstelle

4.3.4.1 Das Modul recoll

4.3.4.1.1 connect(confdir=None, extra_dbs=None, writable = False)

Die Funktion `connect()` stellt eine Verbindung zu einem oder mehreren Recoll-Index(en) her und gibt ein Db-Objekt zurück.

Dieser Aufruf initialisiert das Modul recoll und sollte immer vor allen anderen Aufrufen oder Objekterzeugungen durchgeführt werden.

- `confdir` kann ein Konfigurationsverzeichnis angeben. Es gelten die üblichen Standardwerte.
- `extra_dbs` ist eine Liste von zusätzlichen Indizes (Xapian-Verzeichnisse).
- `writable` entscheidet, ob neue Daten über diese Verbindung indiziert werden können.

4.3.4.1.2 Die Klasse Db

Ein Db-Objekt wird durch einen `connect()`-Aufruf erstellt und enthält eine Verbindung zu einem Recoll-Index.

Db.close() Schließt die Verbindung. Danach können Sie nichts mehr mit dem Db-Objekt tun.

Db.query(), Db.cursor() Diese Aliasnamen geben ein leeres Abfrageobjekt für diesen Index zurück.

Db.setAbstractParams(maxchars, contextwords) Legt die Parameter fest, die zur Erstellung von Snippets (Sätze von Schlüsselwörtern in Kontexttextfragmenten) verwendet werden. `maxchars` definiert die maximale Gesamtgröße des Abstracts. `contextwords` definiert, wie viele Begriffe um das Schlüsselwort herum angezeigt werden.

Db.termMatch(match_type, expr, field='', maxlen=-1, casesens=False, diacsens=False, lang='english') Erweitert einen Ausdruck anhand der Index-Term-Liste. Führt die Grundfunktion des GUI-Term-Explorers aus. `match_type` kann entweder `Wildcard`, `regexp` oder `stem` sein. Gibt eine Liste von Begriffen zurück, die anhand des Eingabeausdrucks expandiert wurden.

4.3.4.1.3 Die Klasse Query

Ein Query-Objekt (entspricht einem Cursor in der Python-DB-API) wird durch einen `Db.query()`-Aufruf erstellt. Es wird verwendet, um Indexsuchen durchzuführen.

Query.sortby(Feldname, ascending=True) Sortiert die Ergebnisse nach *`Feldname`* in auf- oder absteigender Reihenfolge. Muss vor dem Ausführen der Suche aufgerufen werden.

Query.execute(query_string, stemming=1, stemlang="english", fetchtext=False, collapseduplicates=False) Startet eine Suche nach *`query_string`*, einem Suchsprachstring von Recoll. Wenn der Index die Dokumenttexte speichert und `fetchtext` True ist, wird der extrahierte Dokumenttext in `doc.text` gespeichert.

Query.executesd(SearchData, fetchtext=False, collapseduplicates=False) Startet eine Suche nach der durch das Search-Data-Objekt definierten Anfrage. Wenn der Index die Dokumenttexte speichert und fetchtext=True ist, wird der extrahierte Dokumenttext in `doc.text` gespeichert.

Query.fetchmany(size=query.arraysize) Holt die nächsten Doc-Objekte aus den aktuellen Suchergebnissen und gibt sie als Array mit der erforderlichen Größe zurück, die standardmäßig dem Wert des Datenelements `arraysize` entspricht.

Query.fetchone() Holt das nächste Doc-Objekt aus den aktuellen Suchergebnissen. Erzeugt eine StopIteration-Ausnahme, wenn es keine Ergebnisse mehr gibt.

Query.close() Schließt die Abfrage. Das Objekt ist nach dem Aufruf unbrauchbar.

Query.scroll(value, mode='relative') Passt die Position in der aktuellen Ergebnismenge an. `mode` kann `relativ` oder `absolut` sein.

Query.getgroups() Ruft die erweiterten Abfragebegriffe als Liste von Paaren ab. Sinnvoll nur nach executexx In jedem Paar ist der erste Eintrag eine Liste von Benutzerbegriffen (der Größe nach einer für einfache Begriffe oder mehr für Gruppen- und Phrasenklauseln), der zweite eine Liste von Abfragebegriffen, wie sie von den Benutzerbegriffen abgeleitet und in der Xapian-Abfrage verwendet werden.

Query.getxquery() Gibt die Beschreibung der Xapian-Abfrage als Unicode-String zurück. Sinnvoll nur nach executexx.

Query.highlight(text, ishtml = methods0, = object) Fügt <span "class=rclmatch">, -Tags um die übereinstimmenden Bereiche im Eingabetext ein und gibt den geänderten Text zurück. `ishtml` kann gesetzt werden, um anzugeben, dass der Eingabetext HTML ist und dass HTML-Sonderzeichen nicht escaped werden sollen. `methods`, falls gesetzt, sollte ein Objekt mit den Methoden startMatch(i) und endMatch() sein, das für jede Übereinstimmung aufgerufen wird und ein begin- und end-Tag zurückgeben sollte

Query.makedocabstract(doc, methods = object)) Erstellt ein Snippets-Abstract für `doc` (ein Doc-Objekt), indem der Text um die übereinstimmenden Begriffe herum ausgewählt wird. Wenn methods gesetzt ist, wird auch eine Hervorhebung durchgeführt. Siehe die Methode highlight.

Query.getsnippets(doc, maxoccs = -1, ctxwords = -1, sortbypage=False, methods = object) Gibt eine Liste von Auszügen aus dem Ergebnisdokument zurück, indem der Text um die passenden Begriffe herum ausgewählt wird. Jeder Eintrag in der Ergebnisliste ist ein Tripel: Seitennummer, Begriff, Text. Standardmäßig erscheinen die relevantesten Snippets zuerst in der Liste. Setzen Sie `sortbypage`, um stattdessen nach Seitenzahl zu sortieren. Wenn `methods` gesetzt ist, werden die

Fragmente hervorgehoben (siehe die highlight-Methode). Wenn `maxoccs` gesetzt ist, wird damit die maximale Länge der Ergebnisliste festgelegt. `ctxwords` ermöglicht die Anpassung der Größe des individuellen Snippet-Kontextes.

Query. iter () und Query.next(), damit Dinge wie `for doc in query:` funktionieren.

Query.arraysize Standardanzahl der von fetchmany verarbeiteten Datensätze (r/w).

Query.rowcount Anzahl der Datensätze, die bei der letzten Ausführung zurückgegeben wurden.

Query.rownumber Nächster Index, der aus den Ergebnissen geholt werden soll. Wird normalerweise nach jedem fetchone()-Aufruf erhöht, kann aber vor dem Aufruf gesetzt/zurückgesetzt werden, um die Suche zu beeinflussen (entspricht der Verwendung von `scroll()`). Beginnt bei 0.

4.3.4.1.4 Die Klasse Doc

Ein Doc-Objekt enthält Indexdaten für ein bestimmtes Dokument. Die Daten werden bei der Suche aus dem Index extrahiert oder bei der Aktualisierung durch das Indexierungsprogramm gesetzt. Das Doc-Objekt hat viele Attribute, die von seinem Benutzer gelesen oder gesetzt werden können. Es entspricht größtenteils dem Rcl::Doc C++ Objekt. Einige der Attribute sind vordefiniert, aber insbesondere beim Indizieren können andere gesetzt werden, deren Namen von der Indizierungskonfiguration als Feldnamen verarbeitet werden. Eingaben können als Unicode oder Strings angegeben werden. Ausgaben sind Unicode-Objekte. Alle Datumsangaben werden als Unix-Zeitstempel angegeben und als Zeichenketten ausgegeben. Eine vollständige Beschreibung der vordefinierten Attribute finden Sie in der C++-Datei `rcldb/rcldoc.cpp`. Hier folgt eine kurze Liste.

- `url` die URL des Dokuments, siehe aber auch `getbinurl()`
- `ipath` der `Dokumentpfad` für eingebettete Dokumente.
- `fbytes, dbytes` die Datei- und Textgröße des Dokuments.
- `fmtime, dmtime` die Zeiten der Dokumentendatei und des Dokuments.
- `xdocid` die Xapian-Dokumenten-ID des Dokuments. Dies ist nützlich, wenn Sie über eine direkte Xapian-Operation auf das Dokument zugreifen möchten.
- `mtype` der MIME-Typ des Dokuments.
- Standardmäßig gespeicherte Felder: `Autor`, `Dateiname`, `Schlüsselwörter`, `Empfänger`

Zum Zeitpunkt der Abfrage sind nur die Felder im Doc-Objekt von Bedeutung, die entweder standardmäßig oder in der Feldkonfigurationsdatei als `gespeichert` definiert sind. Der verarbeitete Text des Dokuments kann vorhanden sein oder nicht, abhängig davon, ob der Index den Text überhaupt speichert, und wenn ja, von der Ausführungsoption der Abfrage `fetchtext`. Siehe auch das Modul `rclextract` für den Zugriff auf Dokumentinhalte.

get(key), [] operator Ruft das benannte Dokumentattribut ab. Sie können auch `getattr(doc, key)` oder `doc.key`

verwenden. **doc.key = value** Setzt das benannte Dokumentattribut. Sie können auch `setattr(doc, key,`

`value)` verwenden. **getbinurl()** Ruft die URL im Byte-Array-Format ab (keine Transkodierung), zur Verwendung

als Parameter für einen Systemaufruf. **setbinurl(url)** Setzt die URL im Byte-Array-Format (keine Transkodierung).

items() Gibt ein Wörterbuch mit den Schlüsseln/Werten der Doc-Objekte zurück

keys() Liste der Doc-Objekt-Schlüssel (Attributnamen).

4.3.4.1.5 Die SearchData-Klasse

Ein SearchData-Objekt ermöglicht die Erstellung einer Abfrage durch die Kombination von Klauseln, die mit `Query.executesd()` ausgeführt werden können. Es kann als Ersatz für den Abfragesprachenansatz verwendet werden. Die Schnittstelle wird sich ein wenig ändern, daher vorerst keine detaillierte Dokumentation...

addclause(type='und'|'oder'|'excl'|'phrase'|'near'|'sub', qstring=string, slack=0, field='', stemming=1, subSearch=SearchData)

4.3.4.2 Das Modul rclextract

Vor Recoll konnten Indexabfragen1.25, keine Dokumentinhalte liefern, da diese nie gespeichert wurden. Recoll und 1.25spätere Versionen speichern in der Regel den Dokumententext, der bei der Ausführung einer Abfrage optional abgerufen werden kann (siehe `query.execute()` oben - das Ergebnis ist immer reiner Text).

Das Modul `rclextract` ermöglicht den Zugriff auf das Originaldokument und den Textinhalt des Dokuments (falls nicht im Index gespeichert, oder den Zugriff auf eine HTML-Version des Textes). Der Zugriff auf das Originaldokument ist besonders nützlich, wenn es eingebettet ist (z. B. ein E-Mail-Anhang).

Sie müssen das Modul `recoll` vor dem Modul `rclextract` importieren.

4.3.4.2.1 Die Klasse Extractor

Extractor(doc) Ein Extractor-Objekt wird aus einem Doc-Objekt erstellt, das von einer Abfrage ausgegeben wird.

Extractor.textextract(ipath) Extrahiert das durch *`ipath`* definierte Dokument und gibt ein Doc-Objekt zurück. Im Feld `doc.text` wird der Text des Dokuments je nach `doc.mimetype` entweder in text/plain oder text/html umgewandelt. Eine typische Anwendung wäre wie folgt:

```
from recoll import recoll, rclextract
qdoc = query.fetchone()
extractor = recoll.Extractor(qdoc)
doc =
extractor.textextract(qdoc.ipath) #
doc.text verwenden, z.B. für die
```

Die Übergabe von `qdoc.ipath` an `textextract()` ist redundant, spiegelt aber die Tatsache wider, dass das Extractor-Objekt tatsächlich die Möglichkeit hat, auf die anderen Einträge in einem zusammengesetzten Dokument zuzugreifen.

Extractor.idoctofile(ipath, targetmtype, outfile='') Extrahiert das Dokument in eine Ausgabedatei, die explizit angegeben werden kann oder als temporäre Datei erstellt wird, die vom Aufrufer gelöscht werden kann. Typische Anwendung:

```
from recoll import recoll, rclextract
qdoc = query.fetchone()
extractor = recoll.Extractor(qdoc)
Dateiname = extractor.idoctofile(qdoc.ipath, qdoc.mimetype)
```

In allen Fällen ist die Ausgabe eine Kopie, auch wenn das angeforderte Dokument eine normale Systemdatei ist, was in einigen Fällen verschwenderisch sein kann. Wenn Sie dies vermeiden wollen, können Sie für ein einfaches Dateidokument wie folgt testen:

```
not doc.ipath und (not "rclbes" in doc.keys() oder doc["rclbes"] == "FS")
```

4.3.4.3 Beispiel für die Verwendung der Such-API

Das folgende Beispiel würde den Index mit einer Zeichenkette in der Benutzersprache abfragen. Weitere Beispiele finden Sie im Verzeichnis `python/samples` in den Recoll-Quellen. Das Unterverzeichnis `recollgui` hat eine sehr embryonale GUI, die die Funktionen zum Hervorheben und zur Datenextraktion demonstriert.

```
#!/usr/bin/python3

from recoll import

recoll db =

recoll.connect()
```

```
db.setAbstractParams(maxchars=80, contextwords=4)

Abfrage = db.query()
nres = query.execute("some user
question") print("Ergebnisanzahl: %d" %
nres)
 wenn nres >
          5:
    nres = 5
for i in range(nres):
    doc = query.fetchone()
    print("Ergebnis #%d" %
    (query.rownumber)) for k in ("title",
    "size"):
        print("%s : %s" % (k, getattr(doc, k)))
    print("%s\n" % db.makeDocAbstract(doc,
    query))
```

4.3.5 Erstellung externer Python-Indexer

Die Aktualisierungs-API kann verwendet werden, um einen Index aus Daten zu erstellen, auf die der reguläre Recoll-Indexer nicht zugreifen kann, oder die so strukturiert sind, dass sie den Recoll-Input-Handlern Schwierigkeiten bereiten.

Ein Indexer, der mit dieser API erstellt wird, hat die gleiche Aufgabe wie der Indexer des Recoll-Dateisystems: Er sucht nach geänderten Dokumenten, extrahiert deren Text, ruft die API für die Indizierung auf und kümmert sich um die Bereinigung des Indexes von Daten aus Dokumenten, die nicht mehr im Dokumentenspeicher vorhanden sind.

Die Daten für einen solchen externen Indexer sollten in einem Index gespeichert werden, der von dem des internen Indexers des Recoll-Dateisystems getrennt ist. Der Grund dafür ist, dass der Bereinigungsvorgang des Hauptindexierers (Entfernen gelöschter Dokumente) auch alle Dokumente des externen Indexierers entfernen würde, da sie während des Dateisystemlaufs nicht gesehen wurden. Die Dokumente des Hauptindizierers würden wahrscheinlich auch ein Problem für die Bereinigungsoperation des externen Indizierers darstellen.

Es gibt zwar Möglichkeiten, die Zusammenarbeit mehrerer fremder Indexer an einem einzigen Index zu ermöglichen, doch ist es einfacher, getrennte Indexer zu verwenden und bei Bedarf die Fähigkeiten der Abfrage-Schnittstelle für den Zugriff auf mehrere Indizes zu nutzen.

Die Aktualisierungsschnittstelle besteht aus zwei Teilen:

- Methoden innerhalb des recoll-Moduls ermöglichen das Einfügen von Daten in den Index, um sie über die normale Abfrageoberfläche zugänglich zu machen.
- Eine Schnittstelle, die auf der Ausführung von Skripten basiert, wird definiert, um entweder der grafischen Benutzeroberfläche oder dem rclextract-Modul den Zugriff auf Originaldokumentdaten zur Vorschau oder Bearbeitung zu ermöglichen.

4.3.5.1 Python-Update-Schnittstelle

Die Aktualisierungsmethoden sind Teil des oben beschriebenen recoll-Moduls. Die Methode connect() wird mit einem `writable=true` verwendet
Parameter, um ein beschreibbares Db-Objekt zu erhalten. Die folgenden Methoden des Db-Objekts sind dann verfügbar.

addOrUpdate(udi, doc, parent_udi=None) Hinzufügen oder Aktualisieren von Indexdaten für ein bestimmtes Dokument Der String `udi` muss eine eindeutige ID für das Dokument definieren. Sie ist ein undurchsichtiges Schnittstellenelement und wird innerhalb von Recoll nicht interpretiert. `doc` ist ein Doc-Objekt, das aus den zu indizierenden Daten erstellt wurde (der Haupttext sollte in `doc.text` stehen). Wenn `parent_udi` gesetzt ist, ist dies ein eindeutiger Bezeichner für den übergeordneten Container (z.B. für den Dateisystem-Indexer wäre dies derjenige, der eine tatsächliche Datei ist).

delete(udi) Löscht den Index von allen Daten für udi und allen Dokumenten (falls vorhanden), die ein `übergeordnetes Dokument_udi` haben.

needUpdate(udi, sig) Testet, ob der Index für das durch udi identifizierte Dokument aktualisiert werden muss. Wenn dieser Aufruf verwendet werden soll, sollte das Feld `doc.sig` beim Aufruf von `addOrUpdate()` einen Signaturwert enthalten. Der Aufruf `needUpdate()` vergleicht dann seinen Parameterwert mit dem gespeicherten `sig` für `udi`. `sig` ist ein undurchsichtiger Wert, der als String verglichen wird.

Der Dateisystem-Indexer verwendet eine Verkettung der dezimalen Zeichenkettenwerte für Dateigröße und Aktualisierungszeit, es könnte aber auch ein Hash des Inhalts verwendet werden.

Als Nebeneffekt, wenn der Rückgabewert false ist (der Index ist aktuell), setzt der Aufruf das Existenzflag für das Dokument (und jedes Unterdokument, das durch sein `parent_udi` definiert ist), so dass ein späterer `purge()`-Aufruf diese beibehält).

Die Verwendung von `needUpdate()` und `purge()` ist optional, und der Indexer kann eine andere Methode verwenden, um die Notwendigkeit einer Neuindizierung zu prüfen oder veraltete Einträge zu löschen.

purge() Löscht alle Dokumente, die während des soeben abgeschlossenen Indizierungsvorgangs (seit open-for-write) nicht berührt wurden. Dies sind die Dokumente, für die der needUpdate()-Aufruf nicht durchgeführt wurde, was bedeutet, dass sie im primären Speichersystem nicht mehr existieren.

4.3.5.2 Abfrage des Datenzugriffs für externe Indexierer (1.23)

Recoll verfügt über interne Methoden zum Zugriff auf Dokumentdaten für seinen internen (Dateisystem-) Indexer. Ein externer Indexer muss Methoden für den Datenzugriff bereitstellen, wenn er eine Integration mit der GUI (z.B. Vorschau-Funktion) oder Unterstützung für das rclextract-Modul benötigt.

Die Indexdaten und die Zugriffsmethode sind durch das Feld `rclbes` (recoll backend storage) `Doc` verbunden. Sie sollten dies auf einen kurzen String-Wert setzen, der Ihren Indexer identifiziert (z. B. verwendet der Dateisystem-Indexer entweder "FS" oder einen leeren Wert, der Web-History-Indexer verwendet "BGL").

Die Verknüpfung erfolgt in einer Backend-Konfigurationsdatei (die im Konfigurationsverzeichnis gespeichert ist). Diese definiert die auszuführenden Befehle für den Zugriff auf die Daten des angegebenen Indexers. Beispiel für das mbox-Indizierungsbeispiel aus dem Recoll-Quellcode (der `rclbes="MBOX"` setzt):

```
[MBOX]
fetch = /prad/zui/recoll/src/python/samples/rclmbox.py fetch
makesig = /prad/zui/recoll/src/python/samples/rclmbox.py
makesig
```

`fetch` und `makesig` definieren zwei auszuführende Befehle, um den Dokumententext abzurufen bzw. die Dokumentensignatur zu berechnen (die Beispielimplementierung verwendet dasselbe Skript mit unterschiedlichen ersten Parametern, um beide Operationen durchzuführen).

Die Skripte werden mit drei zusätzlichen Argumenten aufgerufen: udi, `url`, `ipath`, die mit dem Dokument gespeichert wurden, als es indiziert wurde, und können eines oder alle verwenden, um die angeforderte Operation durchzuführen. Der Aufrufer erwartet die Ergebnisdaten auf `stdout`.

4.3.5.3 Beispiele für externe Indexer

Der Quellbaum von Recoll enthält zwei Beispiele für externe Indexer im Verzeichnis `src/python/samples`. Das interessantere Beispiel ist `rclmbox.py`, das ein Verzeichnis mit mbox-Ordnern indiziert. Er nutzt die meisten Funktionen der Aktualisierungsschnittstelle und verfügt über eine Schnittstelle für den Datenzugriff.

Weitere Informationen finden Sie in den Kommentaren innerhalb der Datei.

4.3.6 Kompatibilität des Pakets mit der Vorgängerversion

Die folgenden Codefragmente können verwendet werden, um sicherzustellen, dass der Code sowohl mit der alten als auch mit der neuen API ausgeführt werden kann (solange er natürlich nicht die neuen Fähigkeiten der neuen API nutzt).

Anpassung an die neue Paketstruktur:

```
versuchen:
from recoll import recoll
Funktionsaufruf()
außer:
importiere
```

Anpassung an die Änderung der Art des `nächsten` Abfrageelements. Derselbe Test kann verwendet werden, um zu entscheiden, ob die Methode (neu) oder setzen Sie den `nächsten` Wert (alt).

```
rownum = query.next if type(query.next) == int else query.rownumber
```

Kapitel 5

Installation und Konfiguration

5.1 Installieren einer Binärkopie

Die Binärkopien von Recoll werden immer als reguläre Pakete für Ihr System verteilt. Sie können entweder über den normalen Softwareverteilungsrahmen des Systems bezogen werden (z.B. Debian/Ubuntu apt, FreeBSD ports, etc.), oder von einer Art "Backports"-Repository, das neuere Versionen als die Standardversionen bereitstellt, oder in einigen Fällen auf der Recoll-Website gefunden werden. Die aktuellsten Informationen über Recoll-Pakete finden Sie normalerweise auf der Download-Seite der Recoll-Website

Die Windows-Version von Recoll wird in einer in sich geschlossenen Setup-Datei geliefert, es muss nichts weiter installiert werden.

Auf Unix-ähnlichen Systemen installieren die Paketverwaltungswerkzeuge automatisch harte Abhängigkeiten für Pakete, die aus einem geeigneten Paket-Repository stammen. Bei heruntergeladenen Paketen müssen Sie diese von Hand nachinstallieren (z. B. wenn **dpkg** sich über fehlende Abhängigkeiten beschwert).

In jedem Fall müssen Sie die unterstützenden Anwendungen für die Dateitypen, die Sie indizieren möchten, überprüfen oder installieren, die nicht von Recoll verarbeitet werden (Text, HTML, E-Mail-Dateien und einige andere).

Vielleicht sollten Sie auch einen Blick in den Konfigurationsbereich werfen (für einen schnellen Test mit Standardparametern ist dies jedoch nicht unbedingt erforderlich). Die meisten Parameter können bequemer über die GUI-Schnittstelle eingestellt werden.

5.2 Unterstützende Pakete

Hinweis
Die Windows-Installation von Recoll ist in sich geschlossen. Windows-Benutzer können diesen Abschnitt überspringen.

Recoll verwendet externe Anwendungen, um einige Dateitypen zu indizieren. Sie müssen diese für die Dateitypen, die Sie indiziert haben möchten, installieren (dies sind optionale Abhängigkeiten für die Laufzeit. Keine wird für die Erstellung oder Ausführung von Recoll benötigt, außer für die Indizierung ihres spezifischen Dateityps).

Nach einem Indizierungsdurchgang können die fehlenden Befehle über das Menü **recoll** File angezeigt werden. Die Liste wird in der `fehlenden` Textdatei im Konfigurationsverzeichnis gespeichert.

Die Vergangenheit hat gezeigt, dass ich nicht in der Lage war, eine aktuelle Anwendungsliste in diesem Handbuch zu führen. Bitte schauen Sie unter http://www.recoll.org/- pages/features.html nach einer vollständigen Liste, zusammen mit Links zu den Homepages oder den Seiten mit den besten Quellen/Patches und diversen Tipps. Was folgt, ist nur ein sehr kurzer Auszug aus den stabilen Grundfunktionen.

- PDF-Dateien benötigen **pdftotext**, das Teil von Poppler ist (normalerweise im Paket `poppler-utils` enthalten). Vermeiden Sie das Original von Xpdf.
- MS Word-Dokumente benötigen **antiword**. Es ist auch nützlich, **wvWare** installiert zu haben, da es als Ausweichlösung für einige Dateien verwendet werden kann, die von **antiword** nicht verarbeitet werden können.

- RTF-Dateien benötigen **unrtf**, das in seinen älteren Versionen große Probleme mit nicht-westlichen Zeichensätzen hat. Viele Linux-Distributionen enthalten veraltete unrtf-Versionen. Prüfen Sie http://www.recoll.org/pages/features.html für Details.
- Bilder: Recoll verwendet das Perl-Paket Exiftool, um Tag-Informationen zu extrahieren. Die meisten Bilddateiformate werden unterstützt.
- Bis Recoll 1.24 benötigen viele XML-basierte Formate den Befehl **xsltproc**, der normalerweise mit libxslt geliefert wird. Diese sind: abiword, fb2 ebooks, kword, openoffice, opendocument svg. Recoll und1.25 verarbeiten sie später intern (mit libxslt).

5.3 Bauen von der Quelle aus

5.3.1 Voraussetzungen

Die folgenden Voraussetzungen sind in allgemeinen Begriffen beschrieben und nicht als spezifische Paketnamen (die von der genauen Plattform abhängen). Die Abhängigkeiten sollten auf den meisten gängigen Unix-Derivaten als Pakete verfügbar sein, und es sollte recht selten vorkommen, dass Sie eines davon nachbauen müssen.

Wenn Sie die grafische Benutzeroberfläche nicht benötigen, können Sie alle GUI-Abhängigkeiten vermeiden, indem Sie

deren Erstellung deaktivieren. (Siehe dazu den Abschnitt configure). Die Einkaufsliste:

- Wenn Sie von Git-Code ausgehen, benötigen Sie die Trias **autoconf**, **automake** und **libtool**. Für die Erstellung von tar-Distributionen werden sie nicht benötigt.
- C++-Compiler. Aktuelle Versionen erfordern C++11-Kompatibilität (1.23 und später).
- bison-Befehl (für Recoll und1.21 später).
- Für die Erstellung der Dokumentation: der Befehl **xsltproc** und die Docbook XML- und Stylesheet-Dateien. Sie können diese Abhängigkeit vermeiden, indem Sie die Erstellung der Dokumentation mit der Konfigurationsoption `--disable-userdoc` deaktivieren.
- Entwicklungsdateien für den Xapian-Kern.

Wichtig
Wenn Sie Xapian für eine ältere CPU (vor Pentium oder 4Athlon 64) bauen, müssen Sie das Flag `--disable-sse` zum configure-Befehl hinzufügen. Andernfalls werden alle Xapian-Anwendungen mit einem `illegalen` Befehlsfehler abstürzen.

- Entwicklungsdateien für Qt .5 und seine eigenen Abhängigkeiten (X11 usw.)
- Entwicklungsdateien für libxslt
- Entwicklungsdateien für zlib.
- Entwicklungsdateien für Python (oder verwenden Sie `--disable-python-module`).
- Entwicklungsdateien für libchm
- Möglicherweise benötigen Sie auch libiconv. Auf Linux-Systemen ist die iconv-Schnittstelle Teil von libc und Sie sollten nichts Besonderes tun müssen.

Auf der Download-Seite von Recoll finden Sie aktuelle Versionsinformationen.

5.3.2 Gebäude

Recoll wurde auf Linux, FreeBSD, Mac OS X und Solaris gebaut, die meisten Versionen danach sollten2005 in Ordnung sein, vielleicht auch einige ältere (Solaris war früher8 in Ordnung). Wenn Sie auf einem anderen System bauen, und Dinge ändern

müssen, würde ich Patches sehr begrüßen.

5.3.2.1 Konfigurieren Sie die Optionen:

`--without-aspell` schaltet den Code für die phonetische Zuordnung von Suchbegriffen aus.

`--with-fam` oder `--with-inotify` aktiviert den Code für die Indizierung in Echtzeit. Die Unterstützung von Inotify ist auf Linux-Systemen standardmäßig aktiviert.

`--with-qzeitgeist` aktiviert das Senden von Zeitgeist-Ereignissen über die besuchten Suchergebnisse und benötigt das Paket qzeitgeist.

`--disable-qtgui` schaltet die grafische Qt-Oberfläche ab, die es ermöglicht, den Indexer und das Kommandozeilen-Suchprogramm zu erstellen, wenn keine Qt-Umgebung vorhanden ist.

`--disable-webkit` implementiert die Ergebnisliste mit einem Qt QTextBrowser anstelle eines WebKit-Widgets, wenn Sie nicht auf letzteres angewiesen sind oder sein können.

`--enable-webengine` aktiviert die Verwendung von Qt Webengine (nur sinnvoll, wenn die Qt GUI aktiviert ist), anstelle von Qt Webkit.

`--enable-guidebug` wird das GUI-Programm von recoll mit Debug-Symbolen erstellen. Dies macht es sehr groß (~50MB), weshalb es standardmäßig entfernt wird.

`--disable-idxthreads` ist ab Version verfügbar, um Multithreading innerhalb des Indizierungsprozesses zu 1.19unterdrücken. Sie können auch die Laufzeitkonfiguration verwenden, um **recollindex** auf die Verwendung eines einzigen Threads zu beschränken, aber die Option zur Kompilierzeit kann einige weitere ungenutzte Sperren deaktivieren. Dies gilt nur für die Verwendung von Multithreading für die zentrale Indexverarbeitung (Dateneingabe). Im Überwachungsmodus von Recoll werden immer mindestens zwei Threads zur Ausführung verwendet.

`--disable-python-module` verhindert die Erstellung des Python-Moduls.

`--disable-python-chm` verhindert die Erstellung der Python-Schnittstelle libchm, die zum Indizieren von CHM-Dateien verwendet wird.

`--enable-camelcase` aktiviert die Aufteilung von camelCase-Wörtern. Dies ist nicht standardmäßig aktiviert, da es den unglücklichen Nebeneffekt hat, dass einige Phrasensuchen ziemlich verwirrend sind: z.B. würde "`MySQL manual`" mit "`MySQL manual`" und "`my sql manual`" übereinstimmen, aber nicht mit "`mysql manual`" (nur innerhalb von Phrasensuchen).

`--with-file-command` Geben Sie die zu verwendende Version des Befehls "file" an (z.B.: --with-file-command=/usr/local/bin/file). Kann nützlich sein, um die gnu-Version auf Systemen zu aktivieren, auf denen die native Version schlecht ist.

`--disable-x11mon` Deaktiviert die X11-Verbindungsüberwachung in recollindex. Zusammen mit --disable-qtgui ermöglicht dies die Erstellung von recoll ohne Qt und X11.

`--disable-userdoc` verhindert die Erstellung des Benutzerhandbuchs. Dadurch wird vermieden, dass die Docbook XML/XSL-Dateien und die TeX-Toolchain für die Übersetzung des Handbuchs in PDF installiert werden müssen.

`--enable-recollq` Aktiviert die Erstellung des Kommandozeilen-Abfragetools **recollq** (recoll -t ohne Qt). Dies wird standardmäßig gemacht, wenn --disable-qtgui gesetzt ist, aber diese Option ermöglicht es, dies zu erzwingen.

`--disable-pic` (nur1.21 für Recoll-Versionen bis zu) kompiliert Recoll mit positionsabhängigem Code. Dies ist inkompatibel mit der Erstellung der KIO oder der Python- oder PHP-Erweiterungen, kann aber zu geringfügig schnellerem Code führen.

`--without-systemd` Deaktiviert die automatische Installation von systemd-Unit-Dateien. Normalerweise werden Unit-Dateien installiert, wenn der Installationspfad erkannt werden kann.

`--with-system-unit-dir=DIR` Geben Sie einen Installationspfad für die systemd-Systemeinheit-Vorlagendatei an.

`--with-user-unit-dir=DIR` Geben Sie einen Installationspfad für die systemd-

Benutzereinheitendatei an. Natürlich gelten die üblichen autoconf-Konfigurationsoptionen,

wie `--prefix`.

5.3.2.2 Normales Verfahren, für eine aus einer tar-Distribution extrahierte Quelle)

```
cd recoll-xxx
./configure [Optionen]
machen.
(praktiziert übliche härtefallabweisende Beschwörungen)
```

5.3.2.3 Bauen aus Git-Code

Wenn Sie aus den vom Git-Repository geklonten Quellen bauen, müssen Sie auch autoconf, automake und libtool installieren und Sie müssen `sh autogen.sh` im obersten Quellverzeichnis ausführen, bevor Sie `configure` starten.

5.3.3 Installation von

Verwenden Sie **`make install`** in der Wurzel des Quellbaums. Dies kopiert die Befehle nach *`prefix/bin`* und die Beispielkonfigurationsdateien, Skripte und andere gemeinsame Daten nach *`prefix/share/recoll`*.

5.3.4 Python-API-Paket

Die Python-Schnittstelle befindet sich im Quellbaum unter dem Verzeichnis `python/recoll`.

Ab Recoll kann das1.19, Modul für Python3 kompiliert werden.

Der normale Recoll-Bauvorgang (siehe oben) installiert das API-Paket für die Standard-Systemversion (Python) zusammen mit dem Hauptcode. Das Paket für andere Python-Versionen (z.B. python3, wenn die Systemvorgabe python2 ist) muss explizit gebaut und installiert werden.

Das Verzeichnis `python/recoll/` enthält die übliche `setup.py`. Nachdem Sie den Hauptcode von Recoll konfiguriert und erstellt haben, können Sie das Skript verwenden, um das Python-Modul zu erstellen und zu installieren:

```
cd recoll-
xxx/python/recoll pythonX
setup.py build
sudo pythonX setup.py installieren
```

5.3.5 Bauen auf Solaris

Wir haben die Erstellung der GUI auf Solaris nicht für aktuelle Versionen getestet. Sie benötigen mindestens Qt. Es 4.4.gibt einige Hinweise auf einer alten Website-Seite, die vielleicht noch gültig sind.

Jemand hat den Indexer1.19 und das Python-Modul getestet, sie funktionieren, mit ein paar kleinen Pannen. Stellen Sie sicher, dass Sie GNU **make** verwenden
und **installieren**.

5.4 Überblick über die Konfiguration

Die meisten Parameter, die für die recoll-GUI spezifisch sind, werden über das Preferences-Menü gesetzt und an der Standard-Qt-Stelle (`$HOME/.config/Recoll.org/recoll.conf`) gespeichert. Sie möchten diese wahrscheinlich nicht von Hand bearbeiten.

Die Indizierungsoptionen von Recoll werden in Textkonfigurationsdateien festgelegt, die sich in einem Konfigurationsverzeichnis befinden. Es kann mehrere solcher Verzeichnisse geben, von denen jedes die Parameter für einen Index definiert.

Die Konfigurationsdateien können von Hand oder über den Indexkonfigurationsdialog (Menü Einstellungen) bearbeitet werden. Das GUI-Tool wird versuchen, Ihre Formatierungen und Kommentare so weit wie möglich zu berücksichtigen, so dass es durchaus möglich ist, beide Ansätze für dieselbe Konfiguration zu verwenden.

Die genaueste Dokumentation zu den Konfigurationsparametern findet sich in den Kommentaren innerhalb der Standarddateien, und wir werden hier nur einen allgemeinen Überblick geben.

Für jeden Index gibt es mindestens zwei Sätze von Konfigurationsdateien. Systemweite Konfigurationsdateien befinden sich in einem Verzeichnis mit dem Namen `/usr/share/recoll/examples` und definieren Standardwerte, die für alle Indizes gelten. Für jeden Index gibt es einen parallelen Satz von Dateien, in denen die individuellen Parameter definiert sind.

Der Standardspeicherort der angepassten Konfiguration ist das Verzeichnis `.recoll` in Ihrem Zuhause. Die meisten Benutzer werden nur dieses Verzeichnis verwenden.

Dieser Speicherort kann geändert werden, oder andere können mit der Umgebungsvariablen `RECOLL_CONFDIR` oder dem Optionsparameter `-c` zu **recoll** und **recollindex** hinzugefügt werden.

Zusätzlich (ab Recoll Version 1.19.7) ist es möglich, zwei zusätzliche Konfigurationsverzeichnisse anzugeben, die vor und nach dem Benutzerkonfigurationsverzeichnis gestapelt werden. Diese werden durch die Umgebungsvariablen `RECOLL_CONFTOP` und `RECOLL_CONFMID` definiert. Werte aus Konfigurationsdateien im obersten Verzeichnis haben Vorrang vor den Benutzerwerten, Werte aus Konfigurationsdateien im mittleren Verzeichnis haben Vorrang vor den Systemwerten und werden von den Benutzerwerten überschrieben. Diese beiden Variablen können für Anwendungen nützlich sein, die die Funktionalität von Recoll erweitern und Konfigurationsdaten hinzufügen müssen, ohne die Dateien des Benutzers zu stören. Bitte beachten Sie, dass die beiden, derzeit einzelnen, Werte in Zukunft wahrscheinlich als durch Doppelpunkte getrennte Listen interpretiert werden: verwenden Sie keine Doppelpunkte innerhalb der Verzeichnispfade.

Wenn das Verzeichnis `.recoll` beim Start von **recoll** oder **recollindex noch nicht** existiert, wird es mit einem Satz leerer Konfigurationsdateien angelegt. **recoll** gibt Ihnen die Möglichkeit, die Konfigurationsdatei zu bearbeiten, bevor die Indizierung beginnt. **recollindex** fährt sofort fort. Um Fehler zu vermeiden, erfolgt die automatische Erstellung des Verzeichnisses nur für den Standardspeicherort, nicht aber, wenn `-c` oder `RECOLL_CONFDIR` verwendet wurde (in letzteren Fällen müssen Sie das Verzeichnis selbst erstellen).

Alle Konfigurationsdateien haben dasselbe Format. Ein kurzer Auszug aus der Hauptkonfigurationsdatei könnte zum Beispiel wie folgt aussehen:

```
        # Durch Leerzeichen getrennte Liste der zu indizierenden
        Dateien und Verzeichnisse. topdirs = ~/docs
        /usr/share/doc
        [~/somedirectory-with-utf8-txt-files]
        defaultcharset = utf-8
```

Es gibt drei Arten von Linien:

- Kommentar (beginnt mit #) oder leer.
- Parameterzuflussung (*name* = *Wert*).
- Definition des Abschnitts ([*somedirname*]).

Lange Zeilen können umbrochen werden, indem jeder unvollständige Teil mit einem Backslash (\) abgeschlossen wird.

Je nach Art der Konfigurationsdatei trennen Abschnittsdefinitionen entweder Gruppen von Parametern oder ermöglichen die Neudefinition einiger Parameter für einen Verzeichnisteilbaum. Sie bleiben so lange in Kraft, bis eine andere Abschnittsdefinition oder das Ende der Datei erreicht wird. Einige der Parameter, die für die Indizierung verwendet werden, werden hierarchisch von der aktuellen Verzeichnisposition aufwärts nachgeschlagen. Nicht alle Parameter können sinnvoll umdefiniert werden; dies wird für jeden einzelnen im nächsten Abschnitt angegeben.

Wichtig
Globale Parameter *dürfen nicht* in einem Verzeichnisunterabschnitt definiert werden, da sie sonst vom Recoll-Code nicht gefunden werden, der sie auf der obersten Ebene sucht (z. B. `skippedPaths`).

Wenn das Tilde-Zeichen (~) am Anfang eines Dateipfades gefunden wird, wird es zum Namen des Heimatverzeichnisses des Benutzers erweitert, wie es eine Shell tun würde.

Einige Parameter sind Listen von Zeichenketten. Leerzeichen werden zur Trennung verwendet. Listenelemente mit eingebetteten Leerzeichen können mit doppelten Anführungszeichen versehen werden. Doppelte Anführungszeichen innerhalb dieser Elemente können durch einen Backslash ersetzt werden.

Kein Wert in einer Konfigurationsdatei darf ein Zeilenumbruchszeichen enthalten. Lange Zeilen können fortgesetzt werden, indem der physische Zeilenumbruch durch einen Backslash ersetzt wird, auch innerhalb von Zeichenketten in Anführungszeichen.

```
        astringlist = "eine
        Zeichenkette mit Leerzeichen"
        thesame = "eine Zeichenkette mit Leerzeichen"
```

Parameter, die nicht Teil von Stringlisten sind, können nicht in Anführungszeichen gesetzt werden, und führende und nachfolgende Leerzeichen werden entfernt, bevor der Wert verwendet wird.

Kodierungsprobleme Die meisten Konfigurationsparameter sind reine ASCII-Werte. Zwei bestimmte Gruppen von Werten können Kodierungsprobleme verursachen:

- Dateipfad-Parameter können Nicht-Ascii-Zeichen enthalten und sollten genau die gleichen Byte-Werte verwenden, wie sie im Dateisystemverzeichnis zu finden sind. Normalerweise bedeutet dies, dass die Konfigurationsdatei die standardmäßige Gebietsschema-Kodierung des Systems verwenden sollte.
- Der Parameter `unac_except_trans` sollte in UTF-8 kodiert sein. Wenn Ihr Systemgebietsschema nicht UTF-8 ist und Sie auch Nicht-Ascii-Dateipfade angeben müssen, stellt dies ein Problem dar, da gängige Texteditoren nicht mit mehreren Kodierungen in einer einzigen Datei umgehen können. In diesem relativ unwahrscheinlichen Fall können Sie die Konfigurationsdatei als zwei getrennte Textdateien mit den entsprechenden Kodierungen bearbeiten und sie miteinander verknüpfen, um die vollständige Konfiguration zu erstellen.

5.4.1 Umgebungsvariablen

`RECOLL_CONFDIR` Legt das Hauptkonfigurationsverzeichnis fest.

`RECOLL_TMPDIR, TMPDIR` Speicherorte für temporäre Dateien, in dieser Reihenfolge der Priorität. Wenn keine dieser Angaben gemacht wird, wird standardmäßig `/tmp` verwendet. Während der Indizierung können große temporäre Dateien erstellt werden, vor allem für die Dekomprimierung, aber auch für die Verarbeitung, z. B. von E-Mail-Anhängen.

`RECOLL_CONFTOP, RECOLL_CONFMID` Erlaubt das Hinzufügen von Konfigurationsverzeichnissen mit Prioritäten unterhalb und oberhalb des Benutzerverzeichnisses (siehe oben den Abschnitt Konfigurationsübersicht für weitere Einzelheiten).

`RECOLL_EXTRA_DBS, RECOLL_ACTIVE_EXTRA_DBS` Hilfe zum Einrichten externer Indizes. Siehe diesen Abschnitt für Erklärungen.

`RECOLL_DATADIR` Legt den Ersatz für den Standardspeicherort der Recoll-Datendateien fest, der sich normalerweise z. B. in `/usr/share/ recoll` befindet.)

`RECOLL_FILTERSDIR` Definiert den Ersatz für den Standardspeicherort von Recoll-Filtern, der normalerweise z.B. in `/usr/share/ recoll/filters` liegt.)

`ASPELL_PROG` aspell-Programm, das für die Erstellung des Rechtschreibwörterbuchs verwendet werden soll. Das Ergebnis muss mit der `libaspell` kompatibel sein, die Recoll verwendet.

5.4.2 Die Hauptkonfigurationsdatei von Recoll, recoll.conf

5.4.2.1 Parameter, die beeinflussen, welche Dokumente indiziert werden

`topdirs` Durch Leerzeichen getrennte Liste von Dateien oder Verzeichnissen, die rekursiv indiziert werden sollen. Standardwert ist ~ (indiziert $HOME). Sie können symbolische Links in der Liste verwenden, diese werden unabhängig vom Wert der Variable followLinks verfolgt.

`monitordirs` Durch Leerzeichen getrennte Liste von Dateien oder Verzeichnissen, die auf Aktualisierungen überwacht werden sollen. Wenn der Echtzeit-Indexer läuft, kann damit nur eine Teilmenge des gesamten indizierten Bereichs überwacht werden. Die Elemente müssen in dem durch die 'topdirs'-Mitglieder definierten Baum enthalten sein.

`skippedNames` Dateien und Verzeichnisse, die ignoriert werden sollen. Durch Leerzeichen getrennte Liste von Platzhaltermustern (einfache Muster, keine Pfade, dürfen keine '/'-Zeichen enthalten), die gegen Datei- und Verzeichnisnamen getestet werden.

Werfen Sie einen Blick auf die Standardkonfiguration für den Anfangswert, einige Einträge passen vielleicht nicht zu Ihrer Situation. Der einfachste Weg, dies zu sehen, ist über die GUI-Indexkonfiguration das Feld "Lokale Parameter".

Die Liste in der Standardkonfiguration schließt versteckte Verzeichnisse (Namen, die mit einem Punkt beginnen) nicht aus, was bedeutet, dass sie möglicherweise eine ganze Reihe von Dingen indiziert, die Sie nicht haben wollen. Andererseits

speichern E-Mail-Benutzer-Agenten wie Thunderbird Nachrichten in der Regel in versteckten Verzeichnissen, und Sie möchten wahrscheinlich, dass diese indiziert werden. Eine mögliche Lösung ist, ".*" in "skippedNames" zu haben, und Dinge wie "~/.thunderbird" "~/.evolution" zu "topdirs" hinzuzufügen.

Nicht einmal die Dateinamen werden für Muster in dieser Liste indiziert, siehe die Variable "noContentSuffixes" für einen alternativen Ansatz, der die Dateinamen indiziert. Kann für jeden Teilbaum neu definiert werden.

skippedNames- Liste der Namensendungen, die aus der Standardliste skippedNames entfernt werden sollen.

skippedNames+ Liste der Namensendungen, die der Standardliste skippedNames hinzugefügt werden sollen.

onlyNames Reguläre Filtermuster für Dateinamen Wenn dies gesetzt ist, werden nur die Dateinamen, die nicht in skippedNames enthalten sind und einem der Muster entsprechen, für die Indizierung berücksichtigt. Kann pro Teilbaum neu definiert werden. Gilt nicht für Verzeichnisse.

noContentSuffixes Liste der Namensendungen (nicht notwendigerweise punktgetrennte Suffixe), für die keine MIME-Typ-Identifikation versucht wird und die Inhalte nicht dekomprimiert oder indiziert werden. Nur die Namen werden indiziert. Dies ergänzt die inzwischen überholte recoll_noindex-Liste aus der mimemap-Datei, die in einer zukünftigen Version verschwinden wird (der Wechsel von mimemap zu recoll.conf erlaubt die Bearbeitung der Liste über die GUI). Dies unterscheidet sich von skippedNames, da es sich hier nur um Übereinstimmungen mit Namensendungen handelt (nicht um Wildcard-Muster), und der Dateiname selbst wird normal indiziert. Dies kann für Unterverzeichnisse umdefiniert werden.

noContentSuffixes- Liste der Namensendungen, die aus der Standardliste noContentSuffixes entfernt werden sollen.

noContentSuffixes+ Liste der Namensendungen, die der Standardliste noContentSuffixes hinzugefügt werden sollen.

skippedPaths Absolute Pfade, die nicht betreten werden sollen. Durch Leerzeichen getrennte Liste von Platzhalterausdrücken für absolute Dateisystempfade (für Dateien oder Verzeichnisse). Die Variable muss auf der obersten Ebene der Konfigurationsdatei definiert werden, nicht in einem Unterabschnitt.

Jeder Wert in der Liste muss textlich mit den Werten in topdirs übereinstimmen, es werden keine Versuche unternommen, symbolische Links aufzulösen. In der Praxis, wenn, wie es häufig der Fall ist, /home ein Link zu /usr/home ist, wird Ihr Standard-Topdirs einen einzelnen Eintrag "~" haben, der zu "/home/yourlogin" übersetzt wird. In diesem Fall sollte jeder skippedPaths-Eintrag mit '/home/yourlogin' beginnen, *nicht* mit '/usr/home/yourlogin'.

Die Index- und Konfigurationsverzeichnisse werden automatisch in die Liste aufgenommen.

Die Ausdrücke werden mit "fnmatch(3)" abgeglichen, wobei das Flag FNM_PATHNAME standardmäßig gesetzt ist. Das bedeutet, dass '/'-Zeichen explizit abgeglichen werden müssen. Sie können 'skippedPathsFnmPathname' setzen, um die Verwendung von FNM_PATHNAME zu 0deaktivieren (was bedeutet, dass '/*/dir3' auf '/dir1/dir2/dir3' passt).

Der Standardwert enthält den üblichen Einhängepunkt für Wechseldatenträger, um Sie daran zu erinnern, dass es in den meisten Fällen keine gute Idee ist, Recoll mit diesen arbeiten zu lassen. Wenn Sie explizit '/media/xxx' zur 'topdirs'-Variable hinzufügen, wird dies außer Kraft gesetzt.

skippedPathsFnmPathname Wird gesetzt, um die Verwendung von FNM_PATHNAME für übersprungene Pfade außer 0Kraft zu setzen.

nowalkfn Dateiname, der dazu führt, dass das übergeordnete Verzeichnis übersprungen wird. Jedes Verzeichnis, das eine Datei mit diesem Namen enthält, wird übersprungen, als ob es Teil der Liste skippedPaths wäre. Beispiel: .recoll-noindex

daemSkippedPaths skippedPaths ist ein Äquivalent für die Indizierung in Echtzeit. Dies ermöglicht es, Teile des Baums zu haben, die zunächst indiziert, aber nicht überwacht werden. Wenn daemSkippedPaths nicht gesetzt ist, verwendet der Daemon skippedPaths.

zipUseSkippedNames Verwendung von skippedNames in Zip-Archiven. Wird direkt vom rclzip-Handler abgerufen. Überspringt die Muster de-
durch skippedNames in Zip-Archiven verfeinert. Kann für Unterverzeichnisse neu definiert werden. Siehe https://www.lesbonscomptes.com/recoll/faqs

zipSkippedNames Leerzeichengetrennte Liste von Platzhalterausdrücken für Namen, die in Zip-Archiven ignoriert werden sollen. Diese Liste wird direkt vom zip-Handler verwendet. Wenn zipUseSkippedNames nicht gesetzt ist, definiert zipSkippedNames die Muster, die in Archiven übersprungen werden sollen. Wenn zipUseSkippedNames gesetzt ist, werden die beiden Listen verkettet und verwendet. Kann für Unterverzeichnisse neu definiert werden.
Siehe https://www.lesbonscomptes.com/recoll/faqsandhowtos/FilteringOutZipArchiveMembers.html

followLinks Symbolische Links bei der Indizierung verfolgen. Standardmäßig werden symbolische Links ignoriert, um eine Mehrfachindizierung von verknüpften Dateien zu vermeiden. Wenn diese Option auf true gesetzt ist, wird keine Anstrengung unternommen, Duplikation zu vermeiden. Diese Option kann für jedes der 'topdirs'-Mitglieder mit Hilfe von Abschnitten individuell eingestellt werden. Sie kann nicht unterhalb der 'topdirs'-Ebene geändert werden. Links in der 'topdirs'-Liste selbst werden immer befolgt.

indexedmimetypes Eingeschränkte Liste der indizierten Mime-Typen. Normalerweise nicht gesetzt (in diesem Fall werden alle unterstützten Typen indiziert). Wenn sie gesetzt ist, wird nur der Inhalt der Typen aus der Liste indiziert. Die Namen werden ohnehin indiziert, wenn indexall- filenames gesetzt ist (Voreinstellung). Die Namen der MIME-Typen sollten der mimemap-Datei entnommen werden (die Werte können sich in einigen Fällen von der Ausgabe von xdg-mime oder file -i unterscheiden). Kann für Teilbäume umdefiniert werden.

excludedmimetypes Liste der ausgeschlossenen MIME-Typen. Ermöglicht den Ausschluss einiger Typen von der Indizierung. Die Namen der MIME-Typen sollten der mimemap-Datei entnommen werden (die Werte können sich in einigen Fällen von der xdg-mime- oder file -i-Ausgabe unterscheiden). Kann für Unterbäume neu definiert werden.

nomd5types Für diese Typen wird kein md5 berechnet. md5-Prüfsummen werden nur zur Deduplizierung von Ergebnissen verwendet und können bei Multimedia- oder anderen großen Dateien sehr teuer sein. Mit dieser Liste können Sie die md5-Berechnung für ausgewählte Typen deaktivieren. Sie ist global (keine Neudefinition für Unterbäume). Im Moment hat sie nur Auswirkungen auf externe Handler (exec und execm). Die Dateitypen können entweder durch Auflistung der MIME-Typen (z.B. audio/mpeg) oder der Handlernamen (z.B. rclaudio) angegeben werden.

compressedfilemaxkbs Größenbegrenzung für komprimierte Dateien. Diese müssen zur Identifizierung in einem temporären Verzeichnis dekomprimiert werden, was in einigen Fällen sehr aufwendig sein kann. Begrenzen Sie die Verschwendung. Negativ bedeutet keine Begrenzung. führt0 dazu, dass keine komprimierte Datei verarbeitet wird. Voreinstellung MB50.

textfilemaxmbs Größenbegrenzung für Textdateien. Hauptsächlich zum Überspringen von Monster-Logs. Voreinstellung MB20.

indexallfilenames Indiziert die Dateinamen von unverarbeiteten Dateien Indiziert die Namen von Dateien, deren Inhalt wir nicht indizieren, weil ein MIME-Typ ausgeschlossen oder nicht unterstützt wird.

usesystemfilecommand Einen Systembefehl zur Erkennung des MIME-Typs einer Datei als letzten Schritt zur Identifizierung des Dateityps verwenden Dies ist im Allgemeinen nützlich, führt aber normalerweise zur Indizierung vieler falscher "Text"-Dateien. Siehe 'systemfilecommand' für den verwendeten Befehl.

systemfilecommand Befehl, der verwendet wird, um MIME-Typen zu erraten, wenn die internen Methoden fehlschlagen Dies sollte ein "file -i" ähnlicher Befehl sein. Der Dateipfad wird als letzter Parameter in die Befehlszeile eingefügt. "xdg-mime" funktioniert besser als der traditionelle "file"-Befehl und ist nun die konfigurierte Vorgabe (mit einem hart kodierten Fallback zu "file")

processwebqueue Entscheiden Sie, ob die Web-Warteschlange verarbeitet werden soll. Die Warteschlange ist ein Verzeichnis, in dem die Webbrowser-Plugins von Recoll die Kopien der besuchten Seiten erstellen.

textfilepagekbs Seitengröße für Textdateien. Wenn dies eingestellt ist, werden Text-/Plain-Dateien in Dokumente von ungefähr dieser Größe unterteilt. Verringert die Speichernutzung zur Indexierungszeit und hilft beim Laden der Daten im Vorschaufenster zur Abfragezeit. Besonders nützlich bei sehr großen Dateien, wie z. B. Anwendungs- oder Systemprotokollen. Siehe auch textfilemaxmbs und compressedfilemaxkbs.

membermaxkbs Größenbegrenzung für Archivmitglieder. Dies wird an die Filter in der Umgebung als RECOLL_FILTER_MAXMEMBER übergeben

5.4.2.2 Parameter, die beeinflussen, wie wir Begriffe erzeugen und den Index organisieren

indexStripChars Entscheidet, ob Groß- und Kleinschreibung sowie diakritische Zeichen im Index gespeichert werden. Wenn dies der Fall ist, können Suchvorgänge unter Berücksichtigung von Groß- und Kleinschreibung durchgeführt werden, aber der Index wird größer, und es kann zu geringfügigen Unregelmäßigkeiten kommen. Die Standardeinstellung ist ein Index ohne Umlaute. Wenn mehrere Indizes für eine Suche verwendet werden, muss dieser Parameter für alle identisch definiert werden. Eine Änderung des Wertes führt zu einem Index-Reset.

indexStoreDocText Entscheiden Sie, ob der Textinhalt der Dokumente im Index gespeichert werden soll. Die Speicherung des Textes ermöglicht es, zur Abfragezeit Ausschnitte daraus zu extrahieren, anstatt sie aus den Indexpositionsdaten zu erstellen.

Neuere Xapian-Indexformate haben unsere Verwendung der Positionsliste in einigen Fällen unannehmbar langsam gemacht. Das letzte Xapian-Indexformat mit guter Leistung für die alte Methode ist Chert, das standardmäßig für noch 1.2,unterstützt wird, aber nicht standardmäßig in
1.4 und wird in 1.6.

Der gespeicherte Dokumententext wird von seinem ursprünglichen Format in UTF-8-Klartext übersetzt, aber nicht von Großbuchstaben, diakritischen Zeichen oder Interpunktionszeichen befreit. Die Speicherung erhöht die Indexgröße in der Regel um 10-20 %, ermöglicht aber auch schönere Schnipsel, so dass es sich lohnen kann, dies zu aktivieren, auch wenn es nicht unbedingt für die Leistung erforderlich ist, wenn Sie sich den Platz leisten können.

Die Variable wirkt sich nur beim Erstellen eines Index aus, d.h. das Verzeichnis xapiandb darf noch nicht existieren. Ihre genaue Wirkung hängt von der Xapian-Version ab.

Für Xapian wird das Format 0,Chert verwendet, wenn1.4, die Variable auf Chert gesetzt ist, und der Text wird nicht gespeichert. Wenn die Variable Glass1, ist, wird verwendet, und der Text wird gespeichert.

Für Xapian 1.2 und für Versionen ab 1.5 und neuer ist das Indexformat immer die Standardeinstellung, aber die Variable

steuert, ob der Text gespeichert wird oder nicht, und die Methode der abstrakten Erzeugung. Mit Xapian 1.5 und neuer und der Variable auf 0 gesetzt, kann die abstrakte Generierung sehr langsam sein, aber diese Einstellung kann immer noch nützlich sein, um Platz zu sparen, wenn Sie die abstrakte Generierung überhaupt nicht verwenden.

nonumbers Legt fest, ob Begriffe für Zahlen generiert werden sollen. Zum Beispiel würden "123", "1.5e6", 192.168.1.4, nicht indiziert werden, wenn nonumbers gesetzt ist ("value123" wäre es trotzdem). Zahlen sind oft sehr interessant für die Suche, und dies sollte wahrscheinlich nicht eingestellt werden, außer in besonderen Situationen, z.B. bei wissenschaftlichen Dokumenten mit einer großen Anzahl von Zahlen darin, wo die Einstellung "nonumbers" die Indexgröße reduziert. Dies kann nur für einen ganzen Index gesetzt werden, nicht für einen Teilbaum.

dehyphenate Legt fest, ob "coworker" auch dann indiziert wird, wenn die Eingabe "co-worker" lautet. Dies ist neu in Version 1.22 und standardmäßig aktiviert. Wenn Sie die Variable auf off setzen, können Sie das vorherige Verhalten wiederherstellen.

backslashasletter Verarbeitet den Backslash als normalen Buchstaben. Dies mag für Leute sinnvoll sein, die TeX-Befehle als solche indizieren wollen, ist aber nicht von allgemeinem Nutzen.

underscoreasletter Verarbeitet den Unterstrich als normalen Buchstaben. Dies ist in so vielen Fällen sinnvoll, dass man sich fragt, ob dies nicht der Standard sein sollte.

maxtermlength Maximale Wortlänge. Wörter, die länger sind als dieser Wert, werden verworfen. Der Standardwert ist 40 und war früher fest kodiert, kann aber jetzt angepasst werden. Sie müssen den Index zurücksetzen, wenn Sie den Wert ändern.

nocjk Legt fest, ob bestimmte ostasiatische (chinesisch-koreanisch-japanische) Zeichen/Worttrennungen ausgeschaltet sind. Dies spart ein wenig CPU-Leistung, wenn Sie keine CJK-Dokumente haben. Wenn Ihre Dokumentenbasis einen solchen Text enthält, Sie aber nicht daran interessiert sind, ihn zu durchsuchen, kann die Einstellung nocjk eine erhebliche Zeit- und Platzersparnis bedeuten.

cjkngramlen Hier können Sie die Größe der n-Gramme einstellen, die für die Indizierung von CJK-Text verwendet werden. Der Standardwert von 2 ist wahrscheinlich in den meisten Fällen angemessen. Ein Wert von 3 würde eine höhere Präzision und Effizienz bei längeren Wörtern ermöglichen, aber der Index wird ungefähr doppelt so groß sein.

indexstemminglanguages Sprachen, für die Stemming-Expansionsdaten erstellt werden sollen. Stemmer-Namen können durch Ausführen von 'recollindex -l' gefunden werden, oder dies kann auch aus einer Liste in der GUI gesetzt werden. Die Werte sind vollständige Sprachnamen, z.B. englisch, französisch...

defaultcharset Standardzeichensatz. Dieser wird für Dateien verwendet, die keine Zeichensatzdefinition enthalten (z. B.: text/plain). Werte, die innerhalb von Dateien gefunden werden, z. B. ein "charset"-Tag in HTML-Dokumenten, überschreiben ihn. Wenn dies nicht gesetzt ist, ist der Standardzeichensatz derjenige, der durch die NLS-Umgebung ($LC_ALL, $LC_CTYPE, $LANG) definiert ist, oder letztendlich iso-8859-1 (eigentlich cp-1252). Wenn Sie aus irgendeinem Grund eine allgemeine Voreinstellung wünschen, die nicht mit Ihrem LANG übereinstimmt und nicht 8859-1 ist, verwenden Sie diese Variable. Diese kann für jedes Unterverzeichnis neu definiert werden.

unac_except_trans Eine Liste von in UTF-8 kodierten Zeichen, die bei der Konvertierung von Text in unbetonte Kleinschreibung besonders behandelt werden sollten. Zum Beispiel hat der Buchstabe a mit Diaeresis im Schwedischen die volle Alphabetbürgerschaft und sollte nicht in ein a umgewandelt werden. Jedes Element in der durch Leerzeichen getrennten Liste hat das Sonderzeichen als erstes Element und die Übersetzung folgt. Die Behandlung sowohl der Klein- als auch der Großbuchstabenversion eines Zeichens sollte angegeben werden, da die Aufnahme in die Liste sowohl die Standardverarbeitung von Akzenten als auch von Großbuchstaben ausschaltet. Der Wert ist global und beeinflusst sowohl die Indizierung als auch die Abfrage. Wir konvertieren auch einige verwirrende Unicode-Zeichen (Anführungszeichen, Bindestrich) in ihre ASCII-Entsprechung, um "unsichtbare" Suchfehler zu vermeiden.

Beispiele: Schwedisch: unac_except_trans = ää Ää öö Öö üü Üü ßss œoe Œoe æae Æae ffff fifi flfl åå Åå " ' " - . Deutsch: unac_except_trans = ää Ää öö Öö üü Üü ßss œoe Œoe æae Æae ffff fifi flfl "' ' " - . Französisch: Sie wollen wahrscheinlich oe und ae zerlegen und niemand würde ein deutsches ß unac_except_trans = ßss œoe Œoe æae Æae ffff fifi flfl "' ' " schreiben. - . Es folgt die Vorgabe für alle, bis jemand protestiert. Diese Zerlegungen werden von unac nicht durchgeführt, aber es ist unwahrscheinlich, dass jemand die zusammengesetzten Formen bei einer Suche eingeben würde. unac_except_trans = ßss œoe Œoe æae Æae ffff fifi flfl "' ' " -

maildefcharset Setzt den Standardzeichensatz für E-Mail-Nachrichten außer Kraft, die keinen angeben. Dies ist hauptsächlich für readpst (libpst)-Dumps nützlich, die utf-8 sind, dies aber nicht angeben.

localfields Setzt Felder in allen Dateien (normalerweise in einem bestimmten fs-Bereich). Die Syntax ist die übliche: name = value ; attr1 = val1 ; [...] value ist leer, also braucht es ein Semikolon am Anfang. Dies ist z. B. nützlich, um das Feld rclaptg für die Anwendungsauswahl in mimeview zu setzen.

testmodifusemtime Verwenden Sie mtime anstelle von ctime, um zu prüfen, ob eine Datei geändert wurde. Die Zeit wird zusätzlich zur Größe verwendet, die immer verwendet wird. Diese Einstellung kann die Neuindizierung auf Systemen reduzieren, auf denen erweiterte Attribute (von einer anderen Anwendung) verwendet, aber nicht indiziert werden, da die Änderung der erweiterten Attribute nur die ctime beeinflusst. Hinweise: - Dies kann die Erkennung von Änderungen in einigen Fällen marginaler Dateiumbenennungen verhindern (das Ziel müsste dieselbe Größe und mtime haben). - Sie sollten in diesem Fall wahrscheinlich auch noxattrfields auf 1 setzen, es sei denn, Sie ziehen es vor, eine xattr-Indizierung durchzuführen, z. B. wenn das lokale Dateiaktualisierungsmuster dies sinnvoll erscheinen lässt (da im

Allgemeinen die Gefahr besteht, dass reine Aktualisierungen erweiterter Attribute ohne Dateiveränderung unentdeckt bleiben). Führen Sie einen vollständigen Index-Reset durch, nachdem Sie dies geändert haben.

noxattrfields Deaktiviert die Umwandlung von erweiterten Attributen in Metadatenfelder. Dies muss wahrscheinlich gesetzt werden, wenn testmodifusem- time gesetzt ist.

metadatacmds Definieren Sie Befehle zur Erfassung externer Metadaten, z. B. tmsu-Tags. Es kann mehrere Einträge geben, die durch Semikolons getrennt sind und jeweils festlegen, in welches Feld die Daten eingetragen werden und welcher Befehl verwendet werden soll. Vergessen Sie das erste Semikolon nicht. Alle Feldnamen müssen unterschiedlich sein. Wenn nötig, können Sie in der "field"-Datei Aliasnamen verwenden. Als nicht allzu hübscher Hack, der der Bequemlichkeit halber eingeräumt wurde, wird jeder Feldname, der mit "rclmulti" beginnt, als Hinweis darauf gewertet, dass der Befehl mehrere Feldwerte in einem Textblob zurückgibt, der als Recoll-Konfigurationsdatei formatiert ist (Zeilen "fieldname = fieldvalue"). Der Name rclmultixx wird ignoriert, und die Feldnamen und -werte werden aus den Daten geparst. Beispiel: metadatacmds = ; tags = tmsu tags %f; rclmulti1 = cmdOutputsConf %f

5.4.2.3 Parameter, die beeinflussen, wo und wie wir Dinge speichern

cachedir Oberstes Verzeichnis für Recoll-Daten. Die Verzeichnisse für Recoll-Daten befinden sich normalerweise relativ zum Konfigurationsverzeichnis (z.B.
~/.recoll/xapiandb, ~/.recoll/mboxcache). Wenn 'cachedir' gesetzt ist, werden die Verzeichnisse stattdessen unter dem angegebenen Wert gespeichert (z.B. wenn cachedir ~/.cache/recoll ist, wäre das Standard-Dbdir ~/.cache/recoll/xapiandb). Dies betrifft dbdir, webcachedir, mboxcachedir, aspellDicDir, die immer noch individuell angegeben werden können, um cachedir zu überschreiben. Beachten Sie, dass bei mehreren Konfigurationen, von denen jede ein anderes cachedir haben muss, keine automatische Berechnung eines Unterpfades unter cachedir stattfindet.

maxfsoccuppc Maximale Belegung des Dateisystems, bei der wir die Indizierung beenden. Der Wert ist ein Prozentsatz, der dem entspricht, was die df-Ausgabespalte "Capacity" anzeigt. Der Standardwert bedeutet 0,keine Überprüfung.

dbdir Verzeichnis der Xapian-Datenbank. Dieses wird bei der ersten Indizierung erstellt. Wenn der Wert kein absoluter Pfad ist, wird er als relativ zu cachedir interpretiert, falls gesetzt, oder zum Konfigurationsverzeichnis (Argument -c oder $RECOLL_CONFDIR). Wenn nichts angegeben wird, ist die Vorgabe ~/.recoll/xapiandb/

idxstatusfile Name der Scratch-Datei, in der der Indexer-Prozess seinen Status aktualisiert. Standard: idxstatus.txt im Konfigurationsverzeichnis.

mboxcachedir Verzeichnis, in dem die Cachedateien für mbox-Nachrichtenoffsets gespeichert werden. Normalerweise ist dies 'mboxcache' unter cachedir, wenn es gesetzt ist, oder sonst unter dem Konfigurationsverzeichnis, aber es kann nützlich sein, ein Verzeichnis zwischen verschiedenen Konfigurationen zu teilen.

mboxcacheminmbs Mindestgröße der mbox-Datei, ab der wir die Offsets zwischenspeichern. Es macht wirklich keinen Sinn, Offsets für kleine Dateien zu cachen. Der Standardwert ist MB5.

mboxmaxmsgmbs Maximale Größe der mbox-Mitgliedsnachricht in Megabyte. Größe, bei deren Überschreitung wir davon ausgehen, dass das mbox-Format fehlerhaft ist oder wir es falsch interpretiert haben, woraufhin wir die Verarbeitung der Datei einfach abbrechen.

webcachedir Verzeichnis, in dem wir die archivierten Webseiten speichern. Dies wird nur vom Code für die Indizierung des Webverlaufs verwendet Standard: cachedir/webcache, wenn cachedir gesetzt ist, sonst $RECOLL_CONFDIR/webcache

webcachemaxmbs Maximale Größe des Web-Archivs in MB. Dies wird nur vom Code für die Indizierung des Webverlaufs verwendet. Voreinstellung: 40 MB. Durch Verkleinern der Größe wird die Datei nicht physisch abgeschnitten.

webqueuedir Der Pfad zur Web-Indizierungswarteschlange. Im alten Plugin war dieser Wert als ~/.recollweb/ToIndex fest codiert, so dass es keine Notwendigkeit oder Möglichkeit gab, ihn zu ändern, aber das WebExtensions-Plugin lädt die Dateien jetzt in das Downloads-Verzeichnis des Benutzers herunter und ein Skript verschiebt sie nach webqueuedir. Das Skript liest diesen Wert aus der Konfiguration, so dass es nun möglich ist, ihn zu ändern.

webdownloadsdir Der Pfad zum Verzeichnis der Browser-Downloads. Hier muss die neue Browser-Erweiterung die Dateien erstellen. Sie werden dann von einem Skript nach webqueuedir verschoben.

webcachekeepinterval Seitenwiederholungsintervall Standardmäßig wird nur eine Instanz einer URL im Cache gehalten. Dies kann geändert werden, indem man einen Wert festlegt, der bestimmt, in welchen Abständen mehrere Instanzen aufbewahrt werden ('Tag', 'Woche', 'Monat', 'Jahr'). Beachten Sie, dass das Erhöhen des Intervalls die vorhandenen Einträge nicht löscht.

aspellDicDir Speicherort des Aspell-Wörterbuchs. Das aspell-Wörterbuch (aspdict.(lang).rws) wird normalerweise in dem Verzeichnis gespeichert, das durch cachedir angegeben wird, wenn es gesetzt ist, oder unter dem Konfigurationsverzeichnis.

filtersdir Verzeichnis für ausführbare Eingabe-Handler. Wenn RECOLL_FILTERSDIR in der Umgebung gesetzt ist, wird es stattdessen verwendet. Der Standardwert ist $prefix/share/recoll/filters. Kann für Unterverzeichnisse umdefiniert werden.

iconsdir Verzeichnis für die Symbole. Der einzige Grund, dies zu ändern, wäre, wenn Sie die in der Ergebnisliste angezeigten Icons ändern wollen. Der Standardwert ist $prefix/share/recoll/images

5.4.2.4 Parameter, die die Indizierungsleistung und Ressourcennutzung beeinflussen

idxflushmb Schwellenwert (Megabyte neuer Daten), bei dem wir den Index aus dem Speicher auf die Festplatte übertragen. Diese Einstellung ermöglicht eine gewisse Kontrolle über die Speichernutzung durch den Indexierungsprozess. Ein Wert von 0 bedeutet, dass kein explizites Flushing durchgeführt wird, wodurch Xapian sein eigenes Ding macht, d.h. jedes $XAPIAN_FLUSH_THRESHOLD Dokument, das erstellt, geändert oder gelöscht wird, flusht: Da der Speicherverbrauch von der durchschnittlichen Dokumentengröße abhängt, nicht nur von der Anzahl der Dokumente, ist der Xapian-Ansatz nicht sehr nützlich, und Sie sollten Recoll die Flushes verwalten lassen. Der vom Programm kompilierte Wert ist 0. Der konfigurierte Standardwert (aus dieser Datei) ist jetzt 50 MB und sollte in vielen Fällen in Ordnung sein. Sie können den Wert auch auf 10 setzen, um Speicher zu sparen, aber wenn Sie maximale Geschwindigkeit erreichen wollen, sollten Sie mit Werten zwischen 20 und 200 experimentieren. Meiner Erfahrung nach sind Werte darüber hinaus immer kontraproduktiv. Wenn Sie etwas anderes feststellen, schreiben Sie mir bitte eine Nachricht.

filtermaxseconds Maximale Ausführungszeit des externen Filters in Sekunden. Voreinstellung (120020mn). Für 0keine Begrenzung auf gesetzt. Dies dient hauptsächlich zur Vermeidung von Endlosschleifen in Postscript-Dateien (loop.ps)

filtermaxmbytes Maximaler virtueller Speicherplatz für Filterprozesse (setrlimit(RLIMIT_AS)), in Megabyte. Beachten Sie, dass dies alle gemappten Bibliotheken einschließt (es gibt keine zuverlässige Linux-Möglichkeit, nur den Datenbereich zu begrenzen), daher müssen wir hier etwas großzügig sein. Alles, was über 2000 liegt, wird auf 32-Bit-Maschinen ignoriert. Der vorherige Standardwert von 2000 würde verhindern, dass java pdftk funktioniert, wenn es von Python rclpdf.py ausgeführt wird.

thrQSizes Konfiguration der Eingabewarteschlangen der Stufe. Es gibt drei interne Warteschlangen in den Stufen der Indizierungspipeline (Extraktion von Dateidaten, Termerzeugung, Indexaktualisierung). Dieser Parameter definiert die Warteschlangentiefen für jede Stufe (drei ganzzahlige Werte). Wird für eine bestimmte Phase der Wert -1 angegeben, wird keine Warteschlange verwendet, und der Thread fährt mit der nächsten Phase fort. In der Praxis hat sich gezeigt, dass tiefe Warteschlangen die Leistung nicht erhöhen. Standard: Ein Wert von für0 die erste Warteschlange weist Recoll an, die automatische Konfiguration auf der Grundlage der ermittelten Anzahl von CPUs durchzuführen (die beiden anderen Werte sind in diesem Fall nicht erforderlich). Verwenden Sie thrQSizes =
-1 -1 -1, um Multithreading vollständig zu deaktivieren.

thrTCounts Anzahl der für jede Indizierungsphase verwendeten Threads. Die drei Phasen sind: Extraktion der Dateidaten, Erzeugung der Terme, Aktualisierung des Index). Die Verwendung der Zählungen wird auch durch einige spezielle Werte in thrQSizes gesteuert: Wenn die erste Warteschlangentiefe 0 ist, werden alle Zählungen ignoriert (automatisch konfiguriert); wenn ein Wert von -1 für eine Warteschlangentiefe verwendet wird, wird die entsprechende Thread-Zahl ignoriert. Es ist nicht sinnvoll, für die letzte Stufe einen anderen Wert als 1 zu verwenden, da die Aktualisierung des Xapian-Index notwendigerweise single-threaded (und durch einen Mutex geschützt) erfolgt.

5.4.2.5 Verschiedene Parameter

loglevel Ausführlichkeit der Protokolldatei 1-6. Bei einem Wert von 2 werden nur Fehler und Warnungen ausgegeben. 3 druckt Informationen wie Dokumentaktualisierungen, ist 4ziemlich ausführlich und sehr 6ausführlich.

logfilename Ziel der Protokolldatei. Verwenden Sie 'stderr' (Standard), um in die Konsole zu schreiben.

idxloglevel Überschreibt den Loglevel für den Indexer.

idxlogfilename Überschreiben des Logfilenamens für den Indexer.

helperlogfilename Zieldatei für die Standard-Fehlerausgabe externer Hilfsprogramme. Die Fehlerausgabe des externen Programms wird standardmäßig in Ruhe gelassen, z.B. wird sie auf das Terminal ausgegeben, wenn das Programm recoll[index] von der Kommandozeile aus ausgeführt wird. Verwenden Sie
/dev/null oder eine Datei in einem nicht existierenden Verzeichnis, um die Ausgabe vollständig zu unterdrücken.

daemloglevel Überschreibt den Loglevel für den Indexer im Echtzeitmodus. Standardmäßig werden die idx...-Werte verwendet, falls gesetzt, ansonsten die log...-Werte.

daemlogfilename Überschreibt den Logfilename für den Indexer im Echtzeitmodus. Standardmäßig werden die idx...-Werte verwendet, falls gesetzt, ansonsten die log...-Werte.

pyloglevel Überschreibt den Loglevel für das Python-Modul.

pylogfilename Überschreiben Sie den Logfilenamen für das Python-Modul.

orgidxconfdir Ursprünglicher Speicherort des Konfigurationsverzeichnisses. Dies wird ausschließlich für bewegliche Datensätze verwendet. Die Angabe des Konfigurationsverzeichnisses innerhalb des Verzeichnisbaums ermöglicht eine automatische Pfadübersetzung zur Abfragezeit, wenn der Datensatz verschoben wurde (z. B. weil er an einem anderen Ort gemountet wurde).

curidxconfdir Aktueller Speicherort des Konfigurationsverzeichnisses. Ergänzung zu orgidxconfdir für bewegliche Datensätze. Dies sollte verwendet werden, wenn das Konfigurationsverzeichnis aus dem Dataset an einen anderen Ort kopiert wurde, entweder weil der Dataset schreibgeschützt ist und eine s/w-Kopie gewünscht wird oder aus Leistungsgründen. Dabei wird der ursprünglich verschobene Speicherort vor dem Kopieren aufgezeichnet, um Pfadübersetzungsberechnungen zu ermöglichen. Wenn beispielsweise ein ursprünglich als "/home/me/mydata/config" indizierter Datensatz nach "/media/me/mydata" gemountet wurde und die grafische Benutzeroberfläche mit einer kopierten Konfiguration läuft, wäre orgidxconfdir "/home/me/mydata/config" und curidxconfdir (wie in der kopierten Konfiguration festgelegt) "/media/me/mydata/config".

idxrundir Indizierungsprozess aktuelles Verzeichnis. Die Eingabehandler hinterlassen manchmal temporäre Dateien im aktuellen Verzeichnis, daher ist es sinnvoll, recollindex chdir in ein temporäres Verzeichnis schreiben zu lassen. Wenn der Wert leer ist, wird das aktuelle Verzeichnis nicht geändert. Wenn der Wert (literal) tmp ist, wird das temporäre Verzeichnis verwendet, das in der Umgebung festgelegt ist (RECOLL_TMPDIR else TMPDIR else /tmp). Wenn der Wert ein absoluter Pfad zu einem Verzeichnis ist, wird dorthin gewechselt.

checkneedretryindexscript Skript, das heuristisch prüft, ob die Indizierung von Dateien, die zuvor fehlgeschlagen ist, wiederholt werden muss. Das Standardskript überprüft die Änderungsdaten von /usr/bin und /usr/local/bin. Ein relativer Pfad wird in den Filterverzeichnissen und dann im Pfad nachgeschlagen. Andernfalls verwenden Sie einen absoluten Pfad.

recollhelperpath Zusätzliche Orte für die Suche nach ausführbaren Hilfsprogrammen. Dies wird z.B. unter Windows vom Python-Code und unter Mac OS von der gebündelten recoll.app verwendet (weil ich keinen zuverlässigen Weg gefunden habe, launchd anzuweisen, den PATH zu setzen). Das folgende Beispiel ist für Windows. Verwenden Sie ':' als Eintragstrennzeichen für Mac und Ux-ähnliche Systeme, ';' ist nur für Windows.

idxabsmlen Länge der Zusammenfassungen, die wir während der Indizierung speichern. Recoll speichert eine Zusammenfassung für jede indizierte Datei. Der Text kann aus einem eigentlichen 'abstract'-Abschnitt im Dokument stammen oder einfach der Anfang des Dokuments sein. Er wird im Index gespeichert, damit er in den Ergebnislisten angezeigt werden kann, ohne die Originaldatei zu dekodieren. Der Parameter idxabsmlen bestimmt die Größe der gespeicherten Zusammenfassung. Der Standardwert ist 250 Bytes. Die Suchschnittstelle bietet Ihnen die Möglichkeit, diesen gespeicherten Text oder eine synthetische Zusammenfassung anzuzeigen, die durch Extraktion von Text um die Suchbegriffe herum erstellt wird. Wenn Sie immer die synthetische Zusammenfassung bevorzugen, können Sie diesen Wert verringern und so ein wenig Platz sparen.

idxmetastoredlen Kürzungslänge der gespeicherten Metadatenfelder. Dies wirkt sich nicht auf die Indizierung aus (das gesamte Feld wird ohnehin verarbeitet), sondern nur auf die Datenmenge, die im Index gespeichert wird, um Felder in Ergebnislisten oder Vorschauen anzeigen zu können. Der Standardwert ist Bytes150, was bei benutzerdefinierten Feldern zu niedrig sein kann.

idxtexttruncatelen Kürzungslänge für alle Dokumenttexte. Indiziert nur den Anfang der Dokumente. Dies wird nicht empfohlen, es sei denn, Sie sind sicher, dass sich die interessanten Schlüsselwörter am Anfang befinden und Sie haben große Probleme mit dem Speicherplatz.

idxsynonyms Name der Synonymdatei für die Indexierungszeit. Diese wird für die Indizierung von Mehrwortsynonymen als Einzelbegriffe verwendet, was wiederum nur sinnvoll ist, wenn Sie mit solchen Begriffen eine Proximity-Suche durchführen wollen.

aspellLanguage Sprachdefinitionen, die bei der Erstellung des aspell-Wörterbuchs verwendet werden sollen. Der Wert muss mit einer Reihe von aspell-Sprachdefinitionsdateien übereinstimmen. Sie können "aspell dicts" eintippen, um eine Liste zu sehen. Wenn dieser Wert nicht gesetzt ist, wird standardmäßig die NLS-Umgebung verwendet, um den Wert zu erraten. Die Werte sind die 2-Buchstaben-Sprachcodes (z.B. 'en', 'fr'...)

aspellAddCreateParam Zusätzliche Option und Parameter für den Befehl zur Erstellung eines aspell-Wörterbuchs. Einige aspell-Pakete benötigen möglicherweise eine zusätzliche Option (z.B. unter Debian Jessie: --local-data-dir=/usr/lib/aspell). Siehe Debian-Fehler 772415.

aspellKeepStderr Setzen Sie dies, um einen Blick auf Fehler bei der Erstellung von aspell-Wörterbüchern zu werfen. Es gibt immer viele, daher ist dies hauptsächlich zur Fehlersuche gedacht.

noaspell Deaktiviert die Verwendung von aspell. Die Generierung des aspell-Wörterbuchs nimmt Zeit in Anspruch, und einige Kombinationen aus aspell-Version, Sprache und lokalen Begriffen führen zu einem Absturz von aspell, so dass es manchmal sinnvoll ist, die Funktion einfach zu deaktivieren.

monauxinterval Aktualisierungsintervall der Hilfsdatenbanken. Der Echtzeit-Indexer aktualisiert die Hilfsdatenbanken

(stemdb, aspell) nur periodisch, da es zu kostspielig wäre, dies bei jeder Dokumentänderung zu tun. Der Standardzeitraum ist eine Stunde.

monixinterval Mindestintervall (Sekunden) zwischen den Abläufen in der Indizierungswarteschlange. Der Echtzeit-Indexer verarbeitet nicht jedes Ereignis, wenn es eintrifft, sondern lässt die Warteschlange anwachsen, um den Overhead zu verringern und mehrere Ereignisse, die dieselbe Datei betreffen, zusammenzufassen. Voreinstellung S30.

mondelaypatterns Timing-Parameter für die Echtzeit-Indizierung. Definitionen für Dateien, die eine längere Verzögerung erhalten, bevor eine Neuindizierung erlaubt ist. Dies ist für sich schnell ändernde Dateien, die nur ab und zu neu indiziert werden sollen. Eine Liste von WildcardPat- tern:Sekunden-Paaren. Die Muster werden mit fnmatch(muster, pfad, 0) abgeglichen. Sie können Einträge, die Leerzeichen enthalten, in Anführungszeichen setzen (setzen Sie den gesamten Eintrag in Anführungszeichen, nicht das Muster). Der Standardwert ist leer. Beispiel: mondelaypatterns = *.log:20 "*mit Leerzeichen.*:30"

idxniceprio "schöne" Prozesspriorität für die Indizierungsprozesse. Standard: 19 (niedrigste) Erscheint mit 1.26.5. Frühere Versionen wurden behoben bei 19.

monioniceclass ionice-Klasse für den Indizierungsprozess. Trotz des irreführenden Namens und auf Plattformen, auf denen dies unterstützt wird, betrifft dies alle Indizierungsprozesse, nicht nur die Echtzeit-/Überwachungsprozesse. Der Standardwert ist (3niedrigste "Idle"-Priorität verwenden).

monioniceclassdata ionice Klassenstufenparameter, wenn die Klasse ihn unterstützt. Der Standardwert ist leer, da die Standardklasse "Idle" keine Ebenen hat.

5.4.2.6 Parameter zur Abfragezeit (keine Auswirkungen auf den Index)

autodiacsens automatische Auslösung der Empfindlichkeit für diakritische Zeichen (nur Rohindex). Wenn der Index nicht entfernt wird, entscheiden Sie, ob die Empfindlichkeit für diakritische Zeichen automatisch ausgelöst werden soll, wenn der Suchbegriff akzentuierte Zeichen enthält (nicht in unac_except_trans). Andernfalls müssen Sie die Abfragesprache und den Modifikator "D" verwenden, um die Empfindlichkeit für diakritische Zeichen festzulegen. Die Voreinstellung ist nein.

autocasesens automatische Unterscheidung von Groß- und Kleinschreibung (nur Rohindex). WENN der Index nicht gestrippt ist (siehe indexStripChars), entscheiden Sie, ob die Groß- und Kleinschreibung automatisch berücksichtigt werden soll, wenn der Suchbegriff Großbuchstaben nur an der ersten Position enthält. Andernfalls müssen Sie die Abfragesprache und den Modifikator "C" verwenden, um die Groß- und Kleinschreibung zu berücksichtigen. Die Voreinstellung ist ja.

maxTermExpand Maximale Anzahl der Abfrageerweiterungen für einen einzelnen Begriff (z. B.: bei Verwendung von Wildcards). Dies betrifft nur Abfragen, nicht die Indizierung. Früher gab es hier keine Begrenzung (außer für Dateinamen, wo die Grenze mit 1000 zu niedrig war), aber das ist bei einem großen Index unangemessen. Standard 10000.

maxXapianClauses Maximale Anzahl von Klauseln, die wir einer einzelnen Xapian-Abfrage hinzufügen. Dies betrifft nur Abfragen, nicht die Indizierung. In einigen Fällen kann das Ergebnis der Termexpansion multiplikativ sein, und wir wollen vermeiden, dass der gesamte Speicher verbraucht wird. Voreinstellung 50000.

snippetMaxPosWalk Maximale Anzahl von Positionen, die beim Auffüllen eines Snippets für die Ergebnisliste durchlaufen werden. Der Standardwert von kann1,000,000 für sehr große Dokumente unzureichend sein, was zu Snippets mit möglicherweise bedeutungsverändernden fehlenden Wörtern führen würde.

5.4.2.7 Parameter für das PDF-Eingabeskript

pdfocr Versucht OCR von PDF-Dateien ohne Textinhalt. Dies kann in Unterverzeichnissen definiert werden. Die Voreinstellung ist aus, da OCR sehr langsam ist.

pdfattach Aktiviert die Extraktion von PDF-Anhängen durch die Ausführung von pdftk (falls verfügbar). Dies ist normalerweise deaktiviert, da es die PDF-Indizierung etwas verlangsamt, selbst wenn kein einziger Anhang gefunden wird.

pdfextrameta Extrahiert Text aus ausgewählten XMP-Metadaten-Tags. Dies ist eine durch Leerzeichen getrennte Liste von qualifizierten XMP-Tag-Namen. Jedes Element kann auch eine Übersetzung in einen Recoll-Feldnamen enthalten, getrennt durch ein '|'-Zeichen. Fehlt das zweite Element, wird der Tag-Name als Recoll-Feldname verwendet. Sie müssen der Datei "fields" auch Spezifikationen hinzufügen, um die Verarbeitung der extrahierten Daten zu steuern.

pdfextrametafix Name des Skripts zur Bearbeitung von XMP-Feldern festlegen. Hier wird der Name eines Skripts definiert, das zur Bearbeitung von XMP-Feldwerten geladen wird. Das Skript sollte eine 'MetaFixer'-Klasse mit einer metafix()-Methode definieren, die mit dem qualifizierten Tag-Namen und dem Wert jedes ausgewählten Feldes zur Bearbeitung oder Löschung aufgerufen wird. Für jedes Dokument wird eine neue Instanz erstellt, so dass das Objekt seinen Zustand beibehalten kann, um z. B. doppelte Werte zu entfernen.

5.4.2.8 Parameter für die OCR-Verarbeitung

ocrprogs Zu testende OCR-Module. Das oberste OCR-Skript versucht, die entsprechenden Module der Reihe nach zu laden, und verwendet das erste, das meldet, dass es in der Lage ist, OCR an der Eingabedatei durchzuführen. Module für Tesseract (tesseract) und ABBYY FineReader (abbyy) sind in der Standarddistribution enthalten. Aus Gründen der Kompatibilität mit der Vorgängerversion ist der Standardwert "tesseract", wenn dies nicht definiert ist. Verwenden Sie bei Bedarf einen expliziten leeren Wert. Ein Wert von "abbyy tesseract" wird alles versuchen.

ocrcachedir Ort für die Zwischenspeicherung von OCR-Daten. Wenn diese Angabe leer oder nicht definiert ist, werden die zwischengespeicherten OCR-Daten standardmäßig unter
$RECOLL_CONFDIR/ocrcache.

tesseractlang Sprache, die für tesseract OCR vorausgesetzt wird. Wichtig für die Verbesserung der OCR-Genauigkeit. Dies kann auch über den Inhalt einer Datei im aktuell bearbeiteten Verzeichnis festgelegt werden. Siehe das Skript rclocrtesseract.py. Beispielwerte: eng, fra... Siehe die Tesseract-Dokumentation.

tesseractcmd Pfad für den Tesseract-Befehl. Nicht in Anführungszeichen setzen. Dies ist vor allem unter Windows nützlich, oder um einen nicht standardmäßigen Tesseract-Befehl anzugeben. Z.B. unter Windows: tesseractcmd = C:/ProgramFiles(x86)/Tesseract-OCR/tesseract.exe

abbyylang Sprache, die für abbyy OCR angenommen wird. Wichtig für die Verbesserung der OCR-Genauigkeit. Dies kann auch über den Inhalt einer Datei im aktuell bearbeiteten Verzeichnis festgelegt werden. Siehe das Skript rclocrabbyy.py. Typische Werte: Englisch, Französisch... Siehe die ABBYY-Dokumentation.

abbyycmd Pfad für den abbyy-Befehl Das ABBY-Verzeichnis ist in der Regel nicht im Pfad enthalten, daher sollten Sie dies angeben.

5.4.2.9 Parameter für bestimmte Standorte festgelegt

mhmboxquirks Enable thunderbird/mozilla-seamonkey mbox format quirks Legen Sie dies für das Verzeichnis fest, in dem die E-Mail-mbox-Dateien gespeichert werden.

5.4.3 Die Datei der Felder

Diese Datei enthält Informationen über die Handhabung dynamischer Felder in Recoll. Einige sehr einfache Felder haben ein fest verdrahtetes Verhalten, und in der Regel sollten Sie die Originaldaten in der Felddatei nicht ändern. Sie können jedoch benutzerdefinierte Felder erstellen, die zu Ihren Daten passen, und sie genauso behandeln, als wären es native Felder.

Die Felddatei hat mehrere Abschnitte, die jeweils einen Aspekt der Feldverarbeitung definieren. Oft müssen Sie mehrere Abschnitte ändern, um das gewünschte Verhalten zu erreichen.

Wir werden hier nur eine kurze Beschreibung geben, genauere Informationen finden Sie in den Kommentaren in der Standarddatei.

Feldnamen sollten in Kleinbuchstaben (ASCII) geschrieben sein.

[Präfixe] Ein Feld wird indiziert (durchsuchbar), indem ein Präfix in diesem Abschnitt definiert wird. Es gibt eine ausführlichere Erklärung, welche Präfixe bei einer Standard-Recoll-Installation verwendet werden. Kurz gesagt: Erweiterungspräfixe sollten in Großbuchstaben geschrieben sein, mit XY beginnen und kurz sein. Z.B. XYMFLD.

[Werte] Die in diesem Abschnitt aufgeführten Felder werden als `Xapian-Werte` im Index gespeichert. Dadurch stehen sie für Bereichsabfragen zur Verfügung und ermöglichen es, die Ergebnisse nach dem Feldwert zu filtern. Diese Funktion unterstützt derzeit String- und Integer-Daten. Weitere Einzelheiten finden Sie in den Kommentaren in der Datei

[Gespeichert] Ein Feld wird gespeichert (innerhalb der Ergebnisse anzeigbar), wenn sein Name in diesem Abschnitt aufgeführt wird (normalerweise mit einem leeren Wert).

[aliases] Dieser Abschnitt definiert Listen von Synonymen für die kanonischen Namen, die in den Abschnitten `[prefixes]` und `[stored]` verwendet werden

[queryaliases] In diesem Abschnitt werden auch Aliase für die kanonischen Feldnamen definiert, mit dem Unterschied, dass die Ersetzung nur zum Zeitpunkt der Abfrage verwendet wird, wodurch die Möglichkeit vermieden wird, dass der Wert zufällige Metadaten aus Dokumenten aufnimmt.

handler-spezifische Abschnitte Einige Eingabe-Handler benötigen möglicherweise eine spezielle Konfiguration für die Behandlung von Feldern. Nur der E-Mail-Nachrichten-Handler hat derzeit einen solchen Abschnitt (namens `[mail]`). Er erlaubt die Indizierung beliebiger E-Mail-Kopfzeilen, zusätzlich zu den standardmäßig indizierten. Weitere solche Abschnitte können in der Zukunft erscheinen.

Hier folgt ein kleines Beispiel für eine Datei mit persönlichen `Feldern`. Diese würde eine bestimmte E-Mail-Kopfzeile extrahieren und sie als durchsuchbares Feld verwenden, wobei die Daten in den Ergebnislisten angezeigt werden können. (Nebenbei

bemerkt: Da der E-Mail-Handler die Werte nicht dekodiert, können nur einfache ASCII-Kopfzeilen indiziert werden, und bei Kopfzeilen, die mehrmals vorkommen, wird nur das erste Vorkommen verwendet).

```
[Präfixe]
        # Index mailmytag-Inhalte (mit dem angegebenen Präfix)
        mailmytag = XMTAG

        [gespeichert]
        # mailmytag im Dokumentendatensatz speichern (damit es - als
        %(mailmytag) - in Ergebnislisten # angezeigt werden kann).
        mailmytag =

        [queryaliases]
        filename = fn
        Container-Dateiname = cfn

        [mail]
        # Extrahieren Sie den X-My-Tag-Mail-Header und verwenden Sie ihn
        intern mit dem # mailmytag-Feldnamen
        x-mein-tag = mailmytag
```

5.4.3.1 Erweiterte Attribute in der Felddatei

Recoll-Versionen und1.19 spätere Versionen verarbeiten benutzererweiterte Dateiattribute

standardmäßig als Dokumentfelder. Attribute werden als gleichnamige Felder verarbeitet, nachdem

das Benutzerpräfix unter Linux entfernt wurde.

Der Abschnitt `[xattrtofields]` der Datei `fields` ermöglicht die Angabe von Übersetzungen von erweiterten Attributnamen in Recoll-Feldnamen. Eine leere Übersetzung deaktiviert die Verwendung der entsprechenden Attributdaten.

5.4.4 Die Mimemap-Datei

`mimemap` gibt die Dateinamenerweiterung zu MIME-Typ-Zuordnungen an.

Bei Dateinamen ohne oder mit unbekannter Erweiterung wird ein Systembefehl (**file** `-i` oder **xdg-mime**) ausgeführt, um den MIME-Typ zu bestimmen (dies kann ausgeschaltet oder der Befehl in der Hauptkonfigurationsdatei geändert werden).

Alle Erweiterungswerte in `mimemap` müssen in Kleinbuchstaben eingegeben werden. Dateinamenserweiterungen werden beim Vergleich während der Indizierung kleingeschrieben, was bedeutet, dass ein mimemap-Eintrag in Großbuchstaben nie gefunden wird.

Die Zuordnungen können für jeden Teilbaum einzeln festgelegt werden, was in manchen Fällen nützlich sein kann. Beispiel: okular-Notizen haben eine `.xml`
Erweiterung, sondern sollten speziell behandelt werden, was möglich ist, da sie sich normalerweise alle an einem Ort befinden. Beispiel:

```
[~/.kde/share/apps/okular/docdata]
.xml = Anwendung/x-okular-notes
```

Die Mimemap-Variable `recoll_noindex` wurde in `recoll.conf` verschoben und in `noContentSuffixes` umbenannt, unter Beibehaltung der gleichen Funktion, wie in der Recoll-Version Für1.21. ältere Recoll-Versionen, siehe die Dokumentation für `noContentSuffixe`, aber verwenden Sie `recoll_noindex` in `mimemap`.

5.4.5 Die Datei mimeconf

Der Hauptzweck der Datei `mimeconf` besteht darin, festzulegen, wie die verschiedenen MIME-Typen für die Indizierung behandelt werden. Dies geschieht im Abschnitt `[index]`, der nicht ohne weiteres geändert werden sollte. Siehe die Kommentare in der Datei.

Die Datei enthält auch andere Definitionen, die die Abfragesprache und die grafische Benutzeroberfläche betreffen und die im Nachhinein an anderer Stelle hätten gespeichert werden sollen.

Im Abschnitt `[icons]` können Sie die Icons ändern, die von der recoll-GUI in den Ergebnislisten angezeigt werden (die Werte sind die Basisnamen der png-Bilder im iconsdir-Verzeichnis (das wiederum in `recoll.conf` definiert ist).

Der Abschnitt `[categories]` definiert die Gruppierung von MIME-Typen in `Kategorien`, wie sie beim Hinzufügen einer rclcat-Klausel zu einer Query-Language-Abfrage verwendet werden. rclcat-Klauseln werden auch von den Standard-Guifilter-Schaltflächen in der GUI verwendet (siehe unten).

Die Filterkontrollen erscheinen oben in der Recoll-GUI, entweder als Kontrollkästchen direkt über der Ergebnisliste oder als Dropbox im Werkzeugbereich.

Standardmäßig sind sie wie folgt beschriftet: `Medien`, `Nachricht`, `Sonstige`, `Präsentation`, `Tabellenkalkulation` und `Text`, und jedes ist einer Dokumentenkategorie zugeordnet. Dies wird im Abschnitt `[guifilters]` festgelegt, wo jedes Steuerelement durch eine Variable definiert wird, die ein Fragment der Abfragesprache benennt.

Ein einfaches Beispiel wird die Sache hoffentlich verdeutlichen.

```
[guifilters]
Big Books = dir:"~/My Books"
Größe>10K My Docs = dir:"~/My Documents"
Kleine Bücher = dir:"~/Kleine Bücher"
Größe>10K System Docs =
```

Die obige Definition würde vier Filter-Kontrollkästchen mit der Bezeichnung `Big Books`, `My Docs` usw. erzeugen.

Der Text nach dem Gleichheitszeichen muss ein gültiges Abfragesprachfragment sein und wird bei Aktivierung der Schaltfläche mit dem Rest der Abfrage mit einer UND-Verknüpfung kombiniert.

Jeder Namenstext vor einem Doppelpunkt wird in der Anzeige gelöscht, aber für die Sortierung verwendet. Auf diese Weise können Sie die Kontrollkästchen in beliebiger Reihenfolge anzeigen lassen. Das folgende Beispiel würde genau das Gleiche wie oben tun, aber die Reihenfolge der Kontrollkästchen in umgekehrt Reihenfolge.

```
[guifilters]
d:Big Books = dir:"~/My Books" Größe>10K
c:My Docs = dir:"~/My Documents"
b:Kleine Bücher = dir:"~/Kleine Bücher"
a:Größe>10K System Docs =
```

Wie Sie vielleicht schon erraten haben, sieht der Standardabschnitt `[guifilters]` wie folgt aus:

```
[guifilters]
text = rclcat:text
Tabellenkalkulation = rclcat:Tabellenkalkulation
Präsentation = rclcat:Präsentation
Medien = rclcat:Medien
Nachricht = rclcat:message andere =
```

5.4.6 Die mimeview-Datei

`mimeview` legt fest, welche Programme gestartet werden, wenn Sie in einer Ergebnisliste auf einen Öffnen-Link klicken. D.h.: HTML wird normalerweise mit Firefox angezeigt, aber Sie bevorzugen vielleicht Konqueror, Ihr openoffice.org-Programm könnte **oofice** statt **openoffice** heißen usw.

Änderungen an dieser Datei können durch direktes Editieren oder über den Einstellungsdialog von **recoll** GUI vorgenommen werden.

Wenn in den Einstellungen der Recoll-GUI die Option Use desktop preferences to choose document editor aktiviert ist, werden alle mimeview-Einträge ignoriert, mit Ausnahme des Eintrags mit der Bezeichnung `application/x-all` (der standardmäßig auf **xdg-open** eingestellt ist).

In diesem Fall definiert die Top-Level-Variable `xallexcepts` eine Liste von MIME-Typ-Ausnahmen, die entsprechend den lokalen Einträgen verarbeitet werden, anstatt an den Desktop übergeben zu werden. Dadurch können spezifische Recoll-Optionen wie eine Seitenzahl oder ein Suchstring an Anwendungen weitergegeben werden, die diese unterstützen, wie z. B. der evince viewer.

Was die anderen Konfigurationsdateien anbelangt, so wird normalerweise eine `mimeview` in Ihrem eigenen Konfigurationsverzeichnis angelegt, die nur die nicht standardmäßigen Einträge enthält, welche die Einträge in der zentralen Konfigurationsdatei überschreiben.

Alle Viewer-Definitionseinträge müssen unter einem `[view]`-Abschnitt platziert werden.

Die Schlüssel in der Datei sind normalerweise MIME-Typen. Sie können ein Anwendungs-Tag hinzufügen, um die Auswahl für einen Bereich des Dateisystems zu spezialisieren (unter Verwendung einer localfields-Spezifikation in `mimeconf`). Die Syntax für den Schlüssel lautet *`mimetype|tag`*

Der Eintrag `nouncompforviewmts` (auf der obersten Ebene, außerhalb des Abschnitts `[view]`) enthält eine Liste von MIME-Typen, die vor dem Start des Viewers nicht dekomprimiert werden sollten (falls sie komprimiert gefunden werden, z. B. *`mydoc.doc.gz`*).

Auf der rechten Seite jeder Zuweisung steht ein Befehl, der zum Öffnen der Datei ausgeführt werden soll. Die folgenden Ersetzungen werden durchgeführt:

- **%D** Datum des Dokuments
- **%f** Dateiname. Dies kann der Name einer temporären Datei sein, wenn es notwendig war, eine solche zu erstellen (z. B. zum Extrahieren eines Unterdokuments aus einem Container).
- **%i** Interner Pfad, für Unterdokumente von Containern. Das Format hängt vom Containertyp ab. Wenn diese Angabe in der Befehlszeile erscheint, erstellt Recoll keine temporäre Datei, um das Unterdokument zu extrahieren, sondern erwartet, dass die aufgerufene Anwendung (möglicherweise ein Skript) dies übernehmen kann.
- **%M** MIME-Typ
- **%p** Seitenindex. Nur für eine Teilmenge von Dokumenttypen von Bedeutung, derzeit nur PDF-, Postscript- und DVI-Dateien. Kann verwendet werden, um den Editor auf der richtigen Seite für einen Treffer oder ein Snippet zu starten.
- **%s** Suchbegriff. Der Wert wird nur für Dokumente mit indizierten Seitenzahlen gesetzt (z. B. PDF). Der Wert wird einer der übereinstimmenden Suchbegriffe sein. Dies würde es ermöglichen, den Wert im "Find"-Eintrag in Evince voreinzustellen, um den Begriff leicht hervorzuheben.
- **%u** Url.

Zusätzlich zu den oben genannten vordefinierten Werten werden alle Zeichenfolgen wie `%(Feldname)` durch den Wert des Feldes mit dem Namen
`Feldname` für das Dokument. Dies könnte in Kombination mit der Feldanpassung verwendet werden, um das Öffnen des Dokuments zu erleichtern.

5.4.7 Die Datei `ptrans`

`ptrans` gibt Pfadübersetzungen zur Abfragezeit an. Diese können in mehreren Fällen nützlich sein.

Die Datei enthält einen Abschnitt für jeden Index, der übersetzt werden muss, entweder den Hauptindex oder zusätzliche Abfrageindizes. Die Abschnitte sind mit den Namen der Xapian-Indexverzeichnisse benannt. Am Ende der Pfade sollte kein Schrägstrich stehen (alle Vergleiche sind textuell). Ein Beispiel sollte die Dinge ausreichend verdeutlichen

```
[/home/me/.recoll/xapiandb]
/dieses/Verzeichnis/verschoben = /nach/diesem/Ort

[/pfad/zu/zusätzlich/xapiandb]
/server/volume1/docdir = /net/server/volume1/docdir
/server/volume2/docdir = /net/server/volume2/docdir
```

5.4.8 Beispiele für Konfigurationsanpassungen

5.4.8.1 Hinzufügen eines externen Viewers für einen nicht indizierten Typ

Stellen Sie sich vor, Sie haben eine Art von Datei, die keinen indizierbaren Inhalt hat, für die Sie aber gerne einen funktionalen Öffnen-Link in der Ergebnisliste haben möchten (wenn sie über den Dateinamen gefunden wird). Die Dateinamen enden auf *`.blob`* und können von der Anwendung *`blobviewer`* angezeigt werden.

Sie benötigen zwei Einträge in den Konfigurationsdateien, damit dies funktioniert:

- Fügen Sie in `$RECOLL_CONFDIR/mimemap` (typischerweise `~/.recoll/mimemap`) die folgende Zeile ein:

```
.blob = Anwendung/x-blobapp
```

Beachten Sie, dass der MIME-Typ hier erfunden ist und Sie ihn genauso gut *`Diesel/Öl`* nennen könnten.

- Fügen Sie in `$RECOLL_CONFDIR/mimeview` unter dem Abschnitt `[view]` hinzu:

```
anwendung/x-blobapp = blobviewer %f
```

Wir gehen davon aus, dass *`blobviewer`* hier einen Dateinamen-Parameter haben möchte, Sie würden `%u` verwenden, wenn es URLs besser findet.

Wenn Sie nur die von Recoll verwendete Anwendung ändern möchten, um einen MIME-Typ anzuzeigen, den es bereits kennt, müssen Sie nur `mimeview` bearbeiten. Die Einträge, die Sie in Ihrer persönlichen Datei hinzufügen, überschreiben die Einträge in der zentralen Konfiguration, die Sie nicht zu ändern brauchen. `mimeview` kann auch über die Benutzeroberfläche geändert werden.

5.4.8.2 Hinzufügen von Indizierungsunterstützung für einen neuen Dateityp

Stellen wir uns nun vor, dass die oben genannten `.blob`-Dateien tatsächlich indizierbaren Text enthalten und dass Sie wissen, wie man ihn mit einem Kommandozeilenprogramm extrahiert. Es ist einfach, Recoll dazu zu bringen, die Dateien zu indizieren. Sie müssen die obige Änderung vornehmen und außerdem Daten zur mimeconf-Datei hinzufügen (normalerweise in `~/.recoll/mimeconf`):

- Fügen Sie unter dem Abschnitt `[index]` die folgende Zeile ein (mehr über das Indexierungsskript *`rclblob`* später):

```
anwendung/x-blobapp = exec rclblob
```

Oder wenn es sich bei den Dateien hauptsächlich um Text handelt und Sie sie nicht für die Indizierung verarbeiten müssen:

```
application/x-blobapp = intern text/plain
```

- Unter dem Abschnitt `[icons]` sollten Sie ein Symbol auswählen, das für die Dateien in den Ergebnislisten angezeigt werden soll. Icons sind normalerweise 64x64 Pixel große PNG-Dateien, die sich in `/usr/share/recoll/images` befinden.
- Unter dem Abschnitt `[Kategorien]` sollten Sie den MIME-Typ dort hinzufügen, wo es Sinn macht (Sie können auch eine Kategorie erstellen). Kategorien können für die Filterung in der erweiterten Suche verwendet werden.

Der rclblob-Handler sollte ein ausführbares Programm oder Skript sein, das in `/usr/share/recoll/filters` existiert. Er erhält einen Dateinamen als Argument und sollte den Text- oder HTML-Inhalt auf der Standardausgabe ausgeben.

Der Abschnitt über die Filterprogrammierung beschreibt detaillierter, wie man einen Input-Handler schreibt.