

Handbuch zum Skript kdesrc-build

Michael Pyne

Carlos Woelz

Deutsche Übersetzung: Thomas Reitelbach



Handbuch zum Skript kdesrc-build

Inhaltsverzeichnis

1	Einführung	8
1.1	Eine kurze Einführung zu kdesrc-build	8
1.1.1	Was ist kdesrc-build?	8
1.1.2	Arbeitsweise von kdesrc-build „kurz zusammengefasst“	8
1.2	Übersicht der Dokumentation	9
2	Erste Schritte	10
2.1	Vorbereiten des Systems zum Kompilieren von KDE	10
2.1.1	Einrichten eines neuen Benutzerkontos	10
2.1.2	Das System für das Kompilieren der KDE-Quelltexte vorbereiten	10
2.1.3	Einrichtung von kdesrc-build	12
2.1.3.1	Installation von kdesrc-build	12
2.1.3.2	Einstellungen der Einrichtungsdatei	12
2.1.3.2.1	Manuelle Einstellung der Einrichtungsdatei	12
2.2	Festlegen der Einstellungen	13
2.3	Das Skript kdesrc-build verwenden	14
2.3.1	Projekt-Metadaten laden	14
2.3.2	Vorschau auf das, was beim Ausführen von kdesrc-build passiert	15
2.3.3	Fehler beim Erstellen beheben	15
2.4	Erstellen eines Moduls	16
2.5	Einrichten der KDE-Umgebung zur Ausführung des Plasma-Desktop	17
2.5.1	Automatische Installation eines Anmeldungstreibers	18
2.5.1.1	Unterstützung von xsession für Distributionen hinzufügen	18
2.5.1.2	Unterstützung von xsession für manuell hinzufügen	18
2.5.2	Manuelle Einrichtung der Umgebung	19
2.6	Auswahl und Organisation der Module	19
2.6.1	Organisation der KDE-Software	19
2.6.2	Auswahl der Module zum Erstellen	20
2.6.3	Modulgruppen	20
2.6.3.1	Das grundlegende Konzept der Modulgruppen	20
2.6.3.2	Besondere Unterstützung für KDE-Modulgruppen	21
2.6.4	Die offizielle KDE-Moduldatenbank	22
2.6.5	KDE-Projektmodule ausfiltern	24
2.7	Zusammenfassung	24

3	Besondere Funktionen des Skripts	25
3.1	Übersicht der Funktionen	25
3.2	Protokolle von kdesrc-build	27
3.2.1	Übersicht der Protokollfunktionen	27
3.2.1.1	Ordnerlayout für Protokolle	27
4	kdesrc-build einrichten	28
4.1	Übersicht über die Einrichtung von kdesrc-build	28
4.1.1	Layout der Einrichtungsdatei	28
4.1.1.1	Globale Einstellungen	28
4.1.1.2	Einstellungen für Module	28
4.1.1.3	Verarbeitung von Options-Werten	29
4.1.1.4	„options“-Module	29
4.1.2	Andere Einrichtungsdateien einschließen	30
4.1.3	Häufig verwendete Einstellungsoptionen	30
4.2	Tabelle der verfügbaren Einstellungsoptionen	31
5	Befehlszeilen-Parameter und Umgebungsvariablen	69
5.1	Anwendung auf der Befehlszeile	69
5.1.1	Häufig verwendete Befehlszeilenargumente	69
5.1.2	Festlegung der Module zum Erstellen	70
5.2	Unterstützte Umgebungsvariablen	70
5.3	Unterstützte Befehlszeilenparameter	70
6	kdesrc-build verwenden	78
6.1	Einleitung	78
6.2	Grundfunktionen von kdesrc-build	78
6.2.1	Unterstützung für qt	78
6.2.2	Standard-Parameter von kdesrc-build	79
6.2.3	Ändern der Priorität von kdesrc-build beim Erstellen	79
6.2.4	Installation als Systemverwalter	80
6.2.5	Fortschrittsanzeige beim Erstellen eines Moduls anzeigen	80
6.3	Weitergehende Funktionen	81
6.3.1	Teile eines Moduls erstellen	81
6.3.1.1	Teile eines Moduls herunterladen	81
6.3.1.2	Ordner beim Erstellen auslassen	81
6.3.2	Zweige (Branches) und Tags mit kdesrc-build verwenden	82
6.3.2.1	Was sind eigentlich Zweige (Branches) und Tags?	82
6.3.2.2	Zweige (Branches) und Tags verwenden	82
6.3.2.3	Erweiterte Optionen für Zweige (branch)	83
6.3.3	Arbeitsweise von kdesrc-build zum erfolgreichen Erstellen	83

Handbuch zum Skript kdesrc-build

6.3.3.1	Automatisch das Erstellen neu ausführen	83
6.3.3.2	Module manuell neu erstellen	84
6.3.4	Einstellungen für Umgebungsvariablen ändern	84
6.3.5	Das Erstellen fortsetzen	84
6.3.5.1	Ein abgebrochenes oder fehlgeschlagenes Erstellen fortsetzen . . .	84
6.3.5.2	Module vom Erstellen ausschließen	85
6.3.6	Optionen auf der Befehlszeile ändern	85
6.3.6.1	Globale Optionen ändern	85
6.3.6.2	Änderung der Optionen für Module	85
6.4	Funktionen für KDE-Entwickler	86
6.4.1	Überprüfung des SSH-Agenten	86
6.5	Weitere Funktionen von kdesrc-build	86
6.5.1	Ausgabe des Skripts kdesrc-build einstellen	86
6.5.2	Farbige Ausgaben	86
6.5.3	Ordner nach dem Erstellen löschen	87
7	CMake, das System zum Erstellen von KDE	89
7.1	Einführung zu CMake	89
8	Danksagungen und Lizenz	90
A	KDE-Module und Organisation des Quelltexts	91
A.1	Die „Module“	91
A.1.1	Einzelne Module	91
A.1.2	Gruppen zusammengehöriger Module	91
A.1.3	„Zweiggruppen“-Module	92
B	Überholte Prozedur zur Einrichtung von Profilen	94
B.1	Einrichtung eines Profils für die KDE-Anmeldung	94
B.1.1	Einstellungen des Anmeldeprofils ändern	94
B.1.2	Starten von KDE	95

Tabellenverzeichnis

4.1	Tabelle der Optionen	68
-----	--------------------------------	----

Zusammenfassung

Das Skript kdesrc-build kompiliert und installiert KDE direkt aus den Quelltexten der KDE-Projektarchive.

Kapitel 1

Einführung

1.1 Eine kurze Einführung zu kdesrc-build

1.1.1 Was ist kdesrc-build?

kdesrc-build ist ein Skript zur Installation und Aktualisierung von [KDE-Software](#) aus den [Git](#)- und [Subversion](#)-Quellcode-Repositories. Es ist insbesondere für das Testen und die Entwicklung von KDE-Software gedacht, sowohl für Benutzer, die Fehlerkorrekturen testen als auch für Entwickler, die an neuen Funktionen arbeiten.

Das Skript kdesrc-build kann so eingestellt werden, dass ein einzelnes Modul, eine vollständige Plasma-Arbeitsfläche mit KDE-Anwendungen oder eine beliebige Auswahl erstellt wird.

Informationen zum Start finden Sie unter [Kapitel 2](#) oder im Folgenden mit für weiteren Details darüber, wie kdesrc-build funktioniert und was in dieser Dokumentation behandelt wird.

1.1.2 Arbeitsweise von kdesrc-build „kurz zusammengefasst“

kdesrc-build arbeitet mit Werkzeugen, die ein Benutzer auf der Befehlszeile zur Verfügung hat, und benutzt dieselbe Schnittstelle. Bei der Ausführung von kdesrc-build werden folgende Arbeitsschritte durchgeführt:

1. kdesrc-build liest die auf der [Befehlszeile](#) angegebenen Optionen und die [Einrichtungsdatei](#) ein. Damit wird festgelegt, welche Module erstellt werden, wo sie installiert werden usw.
2. kdesrc-build aktualisiert den Quelltext für jedes [Modul](#), bis alle Module auf den neuesten Stand gebracht sind. Fehler bei der Aktualisierung führen nicht zum Abbruch des Skripts – am Ende dieser Phase werden Sie benachrichtigt, bei welchen Modulen die Aktualisierung nicht erfolgreich durchgeführt werden konnte.
3. Module, die erfolgreich aktualisiert wurden, werden gebaut, die Tests werden ausgeführt und dann werden sie installiert. Um den gesamten Zeitaufwand zu reduzieren, beginnt kdesrc-build standardmäßig mit der Erstellung, sobald das erste Modul die Aktualisierung abgeschlossen hat. und lässt die übrigen Aktualisierungen im Hintergrund weiterlaufen.

TIP

Eine *umfassende* Übersicht, wie KDE-Module aufgebaut sind, finden Sie einschließlich informativer Diagramme im Online Artikel [über die KDE-Anwendung Krita](#). Der Arbeitsablauf in diesem Artikel wird von kdesrc-build für alle KDE Module automatisiert durchgeführt.

1.2 Übersicht der Dokumentation

Dieses Handbuch beschreibt die folgenden Aspekte der Arbeitsweise von kdesrc-build:

- Ein [Überblick](#) über die erforderlichen Schritte für einen erfolgreichen Start.
- Wichtige [Leistungsmerkmale](#).
- Die Syntax und Optionen der [Einrichtungsdatei](#).
- Die [Optionen auf der Befehlszeile](#).

Außerdem wird erläutert, welche zusätzlichen Programme benutzt werden müssen, um alle nicht automatisch von kdesrc-build ausgeführten Aktionen auszuführen.

Kapitel 2

Erste Schritte

In diesem Kapitel wird erklärt, wie Sie mit Hilfe von kdesrc-build die entsprechenden Module aus dem KDE-Quelltextarchiv herunterladen und kompilieren. Des Weiteren werden einige Grundlagen zur Struktur der Quelltexte im KDE-Archiv und die nötigen Schritte vor dem Ausführen des Skripts erklärt.

Alle in diesem Kapitel angesprochene Punkte werden detaillierter im Artikel [Selbst kompilieren im Wiki der KDE-Gemeinschaft](#) erklärt. Wenn Sie KDE zum ersten Mal kompilieren, sollten Sie dieses Dokument lesen und bei Fragen als Referenz heranziehen. Darin finden Sie auch Informationen zu Paketierungs-Hilfsprogrammen, Kompilier-Voraussetzungen, häufigen Stolperfallen und allgemeinen Strategien zur Verwendung Ihrer neuen KDE-Installation.

2.1 Vorbereiten des Systems zum Kompilieren von KDE

2.1.1 Einrichten eines neuen Benutzerkontos

Es wird empfohlen, für das Erstellen, die Installation und das Ausführen von KDE einen eigenen Benutzer anzulegen. Das erfordert weniger Berechtigungen und verhindert Konflikte mit den von Ihrer Distribution installierten Paketen. Wenn Sie bereits KDE-Pakete installiert haben, sollten Sie einen anderen Benutzer nur für das Kompilieren und Ausführen des neuen KDE verwenden.

TIP

Lassen Sie Ihr KDE aus der Distribution unverändert, dann haben Sie immer noch ein funktionsfähiges Arbeitssystem, wenn ein selbsterstelltes KDE aus welchen Gründen auch immer nicht stabil läuft.

Sie können auch einstellen, dass die Installation in ein systemweites Verzeichnis erfolgt (z. B. `/usr/src/local`). In dieser Dokumentation wird diese Installationsart nicht behandelt.

2.1.2 Das System für das Kompilieren der KDE-Quelltexte vorbereiten

Bevor Sie kdesrc-build (oder eine andere Kompiliermethode) verwenden, müssen Sie die Entwickler-Werkzeuge und -Bibliotheken für KDE installieren. Ein fast vollständige Liste finden Sie auf der Seite [KDE Community Wiki Build Requirements](#).

Hier ist eine Liste einiger erforderlicher Entwicklungswerkzeuge:

- Sie benötigen CMake, diese Software wird von KDE verwendet, um die Einstellungen für den Quellcode und die Generierung der spezifischen Erstellungsbefehle für Ihr System zu erzeugen. Die benötigte Version hängt davon ab, welche Versionen von KDE Software Sie erstellen möchten, aber bei modernen Distributionen sollte das CMake in Ihrer Distribution ausreichend sein.
- Sie müssen auch die Dienstprogramme zum Herunterladen der KDE-Quelltexte installieren. d. h. mindestens folgende Programme installiert haben:
 - Das [Quelltextverwaltungssystem Git](#), das für alle KDE- [Quelltexte](#) benutzt wird.
 - Die Quelltext-Versionsverwaltung [Bazaar](#) wird für ein einziges Modul (libdbusmenu-qt) gebraucht, das für die KDE-Bibliotheken benötigt wird. Die meisten Benutzer können diese Bibliothek von Ihrer Distribution installieren. Trotzdem unterstützt kdesrc-build das Erstellen dieser Bibliothek aus dem Quelltext. Aber um libdbusmenu-qt zu erstellen, muss Bazaar installiert sein.
- Die Skriptsprache Perl wird für kdesrc-build, einige KDE-Repositories und Qt™ benötigt, falls Sie Qt™ selbst kompilieren möchten.

Die Version von Perl in Ihrer Distribution sollte geeignet sein, es muss mindestens Perl 5.14 sein, aber Sie benötigen auch einige zusätzliche Module (kdesrc-build gibt eine Warnung aus, wenn diese nicht vorhanden sind):

 - IO::Socket::SSL
 - JSON::PP oder JSON::XS
 - YAML::PP, YAML::XS oder YAML::Syck
- Sie benötigen eine vollständige C++-Entwicklungsumgebung (Compiler, Standard-Bibliothek, Laufzeitumgebung) und alle erforderlichen Entwicklungspakete. Die minimal benötigten Versionen variieren je nach KDE-Modul: die KDE Frameworks 5 unterstützt auch ältere Compiler, während KDE Plasma 5 und KDE Anwendungen tendenziell neuere Compiler erfordern.

Die Compiler GCC 4.8 oder Clang 4 sind das empfohlene Minimum. Viele Distributionen unterstützen die einfache Installation dieser Werkzeuge mit einem „build-essentials“-Paket, einer Option zur Installation von „build dependencies“ mit Qt™, oder ähnliche Funktionen. Das KDE Community Wiki hat eine Seite [, die die empfohlenen Pakete für die wichtigsten Distributionen](#) aufführt.
- Sie benötigen ein Build-Tool, das die Kompilierungsschritte tatsächlich durchführt, wie sie von CMake generiert wurden. GNU Make wird empfohlen und sollte über Ihre Paketverwaltung verfügbar sein. CMake unterstützt auch andere Optionen, wie z.B. das Ninja Build-Tool, das von kdesrc-build verwendet werden kann, indem die Option [custom-build-command](#) in der Einrichtungsdatei angegeben wird.
- Schließlich benötigen Sie die passenden Qt™-Bibliotheken einschließlich der Entwicklungspakete für die Version von KDE-Software, die Sie bauen möchten. kdesrc-build unterstützt offiziell nicht die Erstellung von Qt™ 5 (der aktuellen Hauptversion). Es wird daher empfohlen, die Entwicklungspakete Ihrer Distribution zu verwenden oder die KDE-Community-Wiki-Seite auf [Qt 5 selbst bauen](#) zu lesen.

ANMERKUNG

Die meisten Betriebssystem-Distributionen enthalten Methoden, die erforderlichen Entwicklungswerkzeuge einfach zu installieren. Sehen Sie im Community-Wiki [Erforderliche Entwicklungspakete](#) nach, ob hier bereits Anleitungen vorliegen.

WICHTIG

Einige der benötigten Pakete werden von den Distributoren in Bibliotheken, Programmdateien oder Werkzeuge und Entwickler-Pakete unterteilt. In diesen Fällen benötigen Sie wenigstens die Programmdateien *und* die Entwicklungs-Pakete.

2.1.3 Einrichtung von kdesrc-build

2.1.3.1 Installation von kdesrc-build

Die KDE-Entwickler nehmen häufige Änderungen an kdesrc-build vor, um es auf dem Stand für die Fortschritten in der KDE-Entwicklung halten, einschließlich Verbesserungen der empfohlenen kdesrc-build-Einrichtung, hinzugefügter Module, Verbesserung von CMake usw..

Daher wird empfohlen kdesrc-build direkt aus dem Quelltextarchiv herunterzuladen und regelmäßig zu aktualisieren.

Sie können kdesrc-build direkt aus dem Quelltextarchiv herunterladen, indem Sie folgenden Befehl in einer Konsole eingeben:

```
$ git clone https://invent.kde.org/sdk/kdesrc-build.git ~/kdesrc-build
```

Ersetzen Sie ~/kdesrc-build durch den Ordner, in den Sie das Skript installieren möchten.

Später können Sie kdesrc-build mit folgendem Befehl aktualisieren:

```
$ cd ~/kdesrc-build
$ git pull
```

TIP

Es ist empfehlenswert, das Installationsverzeichnis von kdesrc-build zu Ihrer Umgebungsvariable `PATH` hinzuzufügen, damit Sie kdesrc-build ausführen können, ohne den Pfad jedes Mal vollständig angeben zu müssen.

2.1.3.2 Einstellungen der Einrichtungsdatei

kdesrc-build verwendet eine [Einrichtungsdatei](#) im Ordner `~/.kdesrc-buildrc`, um zu steuern welche Module erstellt werden, wo sie installiert werden usw.

Sie können das in kdesrc-build enthaltene Programm `kdesrc-build-setup` benutzen, um eine einfache Einrichtungsdatei für kdesrc-build zu erzeugen. Bearbeiten Sie dann die Datei `~/.kdesrc-buildrc` und passen sie an Ihre Wünsche an.

Das Programm `kdesrc-build-setup` wird in einem Terminal ausgeführt und verwendet keine grafische Bedienungsfläche, genauso wie `kdesrc-build`. Daher können Sie es auch auf einem System ohne Grafik benutzen.

2.1.3.2.1 Manuelle Einstellung der Einrichtungsdatei

Sie können Ihre Einrichtungsdatei auch manuell einstellen, indem Sie die mitgelieferte Beispielkonfigurationsdatei `kdesrc-buildrc-kf5-sample` nach `~/.kdesrc-buildrc` kopieren und dann die Datei bearbeiten. Kapitel 4 ist dafür eine nützliche Referenz, insbesondere deren [Tabelle der Optionen](#).

kdesrc-build enthält viele empfohlene Einrichtungsdateien zur Unterstützung von KDE-Frameworks 5, Plasma 5, und anderen KDE-Anwendungen. Das Programm `kdesrc-build-setup` benutzt diese Dateien, um die Einrichtungsdatei zu erstellen. Hier finden Sie Informationen darüber, wie Sie Ihre Abschnitt 4.1.2 eigenen Einrichtungsdateien aus einer angepassten Datei `~/.kdesrc-buildrc` erstellen.

Weitere Informationen zur Syntax der [Einrichtungsdatei](#) können Sie in Abschnitt 2.2 und Kapitel 4 nachlesen.

2.2 Festlegen der Einstellungen

Damit kdesrc-build funktioniert, sollten Sie die Datei `.kdesrc-buildrc` in Ihrem persönlichen Ordner haben, in der die allgemeinen Einstellungen festgelegt sind.

ANMERKUNG

Es ist möglich, mehrere Einrichtungsdateien für kdesrc-build zu verwenden, das wird im Kapitel Kapitel 4 beschrieben. In diesem Handbuch wird davon ausgegangen, dass die Einstellungen in der Datei `~/kdesrc-buildrc` gespeichert werden.

Benutzen Sie die Datei `kdesrc-buildrc-kf5-sample` als Vorlage und stellen Sie die allgemeinen Optionen und die Module ein, die Sie installieren möchten.

Die Standardeinstellungen sind normalerweise ausreichend, um KDE zu erstellen. Folgende Einstellungen bieten sich zu Anpassung an:

- **kdedir**: Der Zielordner, in den die KDE-Software installiert wird. Als Voreinstellung wird `~/kde` für eine Installation für einen einzelnen Benutzer verwendet.
- **branch-group** kann verwendet werden, um den passenden Entwicklungszweig für die KDE-Module als Ganzes zu wählen. Es gibt viele unterstützte Build-Einstellungen, aber Sie werden wahrscheinlich `kf5-qt5` benutzen, damit kdesrc-build den neuesten Quelltext auf der Basis von Qt™ 5 und KDE Frameworks 5 herunterlädt.

TIP

kdesrc-build verwendet eine Standard-Branch-Gruppe, wenn Sie keine auswählen, aber diese Vorgabe wird sich im Laufe der Zeit ändern, daher ist eine Auswahl besser, damit sich die Branch-Gruppe sich nicht unerwartet ändert.

- **source-dir**: Der Ordner, den kdesrc-build zum Herunterladen des Quellcodes, zum Ausführen des Build-Prozesses und zum Speichern von Protokollen verwendet. Der Standardwert ist `~/kdesrc`.
- **cmake-options**: Optionen, die an CMake übergeben werden, wenn jedes Modul gebaut wird. Typischerweise wird dies verwendet, um zwischen „debug“- oder „release“-Builds zu wechseln, optionale Funktionen zu aktivieren oder zu deaktivieren oder Informationen über den Speicherort der benötigten Bibliotheken an den dem Build-Prozess zu übergeben.
- **make-options**: Optionen, die bei der Ausführung von make verwendet werden, um jedes Modul zu bauen, sobald CMake das Build-System eingerichtet hat.

Die am häufigsten verwendete Option ist `-jN`, wobei `N` durch die maximale Anzahl der Kompilervorgänge, die Sie zulassen möchten ersetzt werden sollte. Eine höhere Zahl bis zu der Anzahl der logischen CPUs in Ihrem System führt zu schnelleren Builds, benötigt aber mehr Systemressourcen.

TIP

kdesrc-build setzt die Option `num-Kerne` auf die erkannte Anzahl der verfügbaren Prozessorkerne. Sie können diesen Wert in Ihrer eigenen Einrichtungsdatei verwenden, damit Sie ihn nicht manuell einstellen zu müssen.

Example 2.1 Einrichtung von Make, um alle verfügbaren CPUs mit Ausnahmen zu benutzen

```
global
# Diese Umgebungsvariable wird automatisch von make benutzt, ↔
  einschließlich
# der make-Befehle, die nicht direkt von by kdesrc-build ausgeführt ↔
  werden wie das Einrichten für Qt
set -env MAKEFLAGS -j${num-cores}
  &#8230;
end global

&#8230;

module-set big-module-set
  repository kde-projects
  use-modules calligra
  make-options -j2 # Reduzierte Anzahl der Erstellungsvorgänge nur für ↔
    diese Module
end module-set
```

ANMERKUNG

Einige sehr große Git-Repositorys können Ihr System überfordern, wenn Sie versuchen, mit zu vielen Erstellungsvorgängen gleichzeitig zu kompilieren, insbesondere Repositorys wie die Qt™-WebKit- und Qt™-WebEngine. Um die Bedienbarkeit des System aufrechtzuerhalten, müssen Sie möglicherweise die Anzahl der Erstellungsvorgänge für bestimmte Module reduzieren. [Beispiel 2.1](#) Ein Beispiel, wie dies ausgeführt werden sollte.

Sie möchten wahrscheinlich verschiedene Module zum Erstellen auswählen, dies wird in [Abschnitt 2.6.2](#) beschrieben.

2.3 Das Skript kdesrc-build verwenden

Nachdem Sie die Einrichtungsdaten festgelegt haben, können Sie nun das Skript zu starten. Auch wenn Sie noch einige Anpassungen vornehmen oder andere Dinge lesen möchten, sollten Sie zumindest die KDE-Projekt-Metadaten laden.

2.3.1 Projekt-Metadaten laden

Melden Sie sich in einem Terminal-Fenster als der Benutzer an, den Sie zum Kompilieren und Installieren der KDE-Software verwenden möchten, und führen Sie das Skript aus:

```
% kdesrc-build --metadata-only
```

Dieser Befehl richtet den Quelltextordner ein und stellt eine Verbindung zu den Git-Repositorys von KDE her, um die Datenbank der KDE-git-Repositorys und die Datenbank mit den Metadaten der Abhängigkeiten herunterzuladen, ohne weitere Änderungen vorzunehmen. Es ist sinnvoll, dies getrennt auszuführen, da diese Metadaten für andere kdesrc-build-Befehle benötigt werden.

2.3.2 Vorschau auf das, was beim Ausführen von kdesrc-build passiert

Mit den installierten Projekt-Metadaten ist es möglich, eine Vorschau darauf zu erhalten, was kdesrc-build beim Start ausführen wird. Dies kann mit der `--pretend` Befehlszeilenoption erfolgen.

```
% ./kdesrc-build --pretend
```

Sie sollten eine Meldung sehen, dass einige Pakete erfolgreich gebaut wurden, obwohl eigentlich nichts gebaut wurde. Wenn es keine signifikanten Probleme gab, können Sie mit der eigentlichen Ausführung des Skripts fortfahren.

```
% kdesrc-build --stop-on-failure
```

Dieser Befehl lädt den entsprechenden Quellcode herunter und baut und installiert jedes Modul der Reihe nach, hält aber an, wenn ein Modul aufgrund der Option `--stop-on-failure` nicht gebaut werden kann. Danach sollten Sie eine Ausgabe ähnlich der in Beispiel 2.2 sehen:

Example 2.2 Beispielausgabe der Ausführung von kdesrc-build

```
% kdesrc-build
Updating kde-build-metadata (to branch master)
Updating sysadmin-repo-metadata (to branch master)

Building libdbusmenu-qt (1/200)
  No changes to libdbusmenu-qt source, proceeding to build.
  Compiling... succeeded (after 0 seconds)
  Installing.. succeeded (after 0 seconds)

Building taglib (2/200)
  Updating taglib (to branch master)
  Source update complete for taglib: 68 files affected.
  Compiling... succeeded (after 0 seconds)
  Installing.. succeeded (after 0 seconds)

Building extra-cmake-modules from <module-set at line 32> (3/200)
  Updating extra-cmake-modules (to branch master)
  Source update complete for extra-cmake-modules: 2 files affected.
  Compiling... succeeded (after 0 seconds)
  Installing.. succeeded (after 0 seconds)

  ...

Building kdevelop from kdev (200/200)
  Updating kdevelop (to branch master)
  Source update complete for kdevelop: 29 files affected.
  Compiling... succeeded (after 1 minute, and 34 seconds)
  Installing.. succeeded (after 2 seconds)

<<<  PACKAGES SUCCESSFULLY BUILT  >>>
Built 200 modules

Your logs are saved in /home/kde-src/kdesrc/log/2018-01-20-07
```

2.3.3 Fehler beim Erstellen beheben

Wenn Sie viele Module herunterladen und erstellen wollen, kann es vorkommen, dass bei der ersten Kompilierung von KDE-Software Fehler auftreten. Das ist aber kein Problem.

kdesrc-build protokolliert die Ausgabe jedes ausgeführten Befehls. In der Standardeinstellung werden die Protokolldateien in den Ordner `~/kdesrc/log` geschrieben. Um die Ursache für den Fehler beim Erstellen eines Moduls im letzten kdesrc-build-Befehl zu finden, lesen Sie die Datei `~/kdesrc/log/latest/ module-name /error.log`.

TIP

Um die Ursache des Fehlers beim Erstellen eines Moduls zu finden, suchen Sie vom Ende der Datei rückwärts, ohne zwischen Groß-/Kleinschreibung zu unterscheiden, nach dem Wort `error`. Blättern Sie bei einem Treffer nach oben, um auch eventuell kurz davor aufgetretenen Fehlermeldungen zu sehen. Die erste Fehlermeldung in dieser Gruppe zeigt normalerweise die Ursache an.

In dieser Datei finden Sie die Fehlermeldungen, mit denen das Kompilieren des Moduls fehlgeschlagen ist. Wurde der Fehler durch fehlende Softwarepakete verursacht, installieren Sie diese mit den zugehörigen Entwicklungspaketen (`-dev`). Dann versuchen Sie das Modul mit der Option `--reconfigure` neu zu kompilieren, so dass kdesrc-build das Modul wieder auf fehlende Pakete überprüft.

Wenn Kompilierfehler wie zum Beispiel ein Syntaxfehler, „falscher Prototyp“, „unbekannter Typ“ oder ähnlich aufgetreten sind, liegt die Ursache dafür vielleicht in den KDE-Quelltexten. Warten Sie ein paar Tage, sollte der Fehler dann immer noch auftreten, schicken Sie eine E-Mail an die Mailingliste kde-devel@kde.org und berichten den Fehler.

Weitere Hinweise über mögliche Fehler und deren Beseitigung wie auch allgemeine Tipps und Strategien zum Erstellen von KDE-Software finden Sie auf der Seite [Build from Source](#).

Andererseits, wenn alles ohne Probleme erstellt wurde, sollten Sie eine neue KDE-Installation auf Ihrem Rechner haben. Jetzt müssen Sie es nur noch ausführen, wie im nächsten Abschnitt [Abschnitt 2.5](#) beschrieben wird.

ANMERKUNG

Weitere Informationen über die Protokolle von kdesrc-build finden Sie im Kapitel [Abschnitt 3.2](#).

2.4 Erstellen eines Moduls

Anstatt immer alle Module zu bauen, möchten Sie vielleicht nur ein einzelnes Modul oder eine andere kleine Teilmenge bauen. Anstatt Ihre Einrichtungsdatei zu bearbeiten, können Sie einfach die Namen der zu erstellenden Module oder Modulgruppen auf der Befehlszeile übergeben.

Example 2.3 Beispielausgabe vom Erstellen eines einzelnen Moduls

```
% kdesrc-build --include-dependencies dolphin
Updating kde-build-metadata (to branch master)
Updating sysadmin-repo-metadata (to branch master)

Building extra-cmake-modules from frameworks-set (1/79)
  Updating extra-cmake-modules (to branch master)
  No changes to extra-cmake-modules source, proceeding to build.
  Running cmake...
  Compiling... succeeded (after 0 seconds)
  Installing.. succeeded (after 0 seconds)

Building phonon from phonon (2/79)
  Updating phonon (to branch master)
  No changes to phonon source, proceeding to build.
  Compiling... succeeded (after 0 seconds)
  Installing.. succeeded (after 0 seconds)

Building attica from frameworks-set (3/79)
  Updating attica (to branch master)
  No changes to attica source, proceeding to build.
  Compiling... succeeded (after 0 seconds)
  Installing.. succeeded (after 0 seconds)

  ...

Building dolphin from base-apps (79/79)
  Updating dolphin (to branch master)
  No changes to dolphin source, proceeding to build.
  Compiling... succeeded (after 0 seconds)
  Installing.. succeeded (after 0 seconds)

<<<  PACKAGES SUCCESSFULLY BUILT  >>>
Built 79 modules

Your logs are saved in /home/kde-src/kdesrc/log/2018-01-20-07
```

In diesem Fall wurde zwar nur die Anwendung *dolphin* angegeben, aber wegen der Option `--include-dependencies` werden auch Abhängigkeiten für *dolphin* eingeschlossen.

ANMERKUNG

Die Auflösung der Abhängigkeit funktionierte in diesem Fall nur, weil *dolphin* in einem Modulsatz auf der Basis von `kde-projects`, in diesem Beispiel `base-apps` angegeben wurde. Siehe Abschnitt [2.6.3.2](#).

2.5 Einrichten der KDE-Umgebung zur Ausführung des Plasma-Desktop

Im Folgenden wird davon ausgegangen, dass Sie einen eigenen Benutzer zum Kompilieren und Installieren von Plasma verwendet haben. Des weiteren wird davon ausgegangen, dass Sie bereits eine PlasmaVersion installiert haben. Wenn dies der Fall ist, muss das neue Plasma Vorrang vor der alten Version erhalten. Sie müssen Sie die Umgebungsvariablen in Ihren Anmeldungs-skripten ändern, damit die neu installierte Arbeitsumgebung benutzt wird.

2.5.1 Automatische Installation eines Anmeldungstreibers

Seit Version 1.16 versucht kdesrc-build einen passenden Treiber für die Anmeldung installieren. Damit können Sie sich an dem mit kdesrc-build erstellten KDE-Arbeitsumgebung anmelden. Dieses Verhalten kann mit der Option `install-session-driver` in der Einrichtungsdatei eingestellt werden.

ANMERKUNG

Eine Sitzung wird nicht eingerichtet, wenn kdesrc-build im Simulationsmodus läuft.

Mit diesem Treiber wird ein benutzerdefinierter Sitzungstyp für „xsession“ eingerichtet. Dieser Sitzungstyp sollte als Standard mit der Anmeldung über kdm funktionieren. In kdm wird die neue Sitzung als benutzerdefinierte Sitzung angezeigt. Bei anderen Anmeldeverwaltungen wie LightDM und gdm müssen möglicherweise zusätzliche Dateien installiert werden, damit Anmeldung mit xsession aktiviert wird.

2.5.1.1 Unterstützung von xsession für Distributionen hinzufügen

Für die Standard-Anmeldeverwaltung einiger Distributionen müssen zusätzliche Pakete installiert werden, um die Anmeldung über xsession zu ermöglichen.

- Für die Linux[®]-Distribution [Fedora](#) muss das Paket `xorg-x11-xinit-session` installiert werden, damit die benutzerdefinierte Anmeldung für xsession unterstützt wird.
- [Debian](#) und Linux[®]-Distribution auf der Basis von Debian unterstützen benutzerdefinierte xsession-Anmeldungen. Dafür muss aber die Option `allow-user-xsession` in der Datei `/etc/X11/Xsession.options` eingestellt sein. Weitere Informationen dazu finden Sie im [Debian-Handbuch](#) im Abschnitt über die [Anpassung der X-Sitzung](#).
- Für andere Distributionen lesen Sie Abschnitt [2.5.1.2](#).

2.5.1.2 Unterstützung von xsession für manuell hinzufügen

Finden Sie keine Anleitungen für Ihre Distribution in Abschnitt [2.5.1.1](#), können Sie manuell einen „eigenen Eintrag für die „xsession“-Anmeldung“ in der Liste von Sitzungstypen für Ihre Distribution mit dem folgenden Vorgehen eintragen:

ANMERKUNG

Um diese Prozedur auszuführen, brauchen Sie wahrscheinlich Systemverwalterrechte.

1. Erstellen Sie die Datei `/usr/share/xsessions/kdesrc-build.desktop`.
2. Stellen Sie sicher, dass die gerade neu erstellte Datei folgenden Text enthält:

```
Type=XSession
Exec=$HOME/.xsession
Name=KDE Plasma Desktop (unstable; kdesrc-build)
```

- ❶ Der Eintrag `$HOME` muss durch den absoluten Pfad zur Ihrem persönlichen Ordner ersetzt werden. zum Beispiel `/home/ user`. Die Spezifikation für Desktop-Einträge erlaubt keine benutzergenerierten Dateien.
3. Wenn die Anmeldeverwaltung neu gestartet wurde, zeigt sie den neuen Sitzungstyp „KDE Plasma Desktop (unstable; kdesrc-build)“ in der Liste der vorhandenen Sitzungen an. Mit diesem Eintrag wird die von kdesrc-build installierte `.xsession`-Datei bei der Anmeldung gestartet.

ANMERKUNG

Am einfachsten ist es, den Rechner und damit auch die Anmeldeverwaltung neu zu starten, wenn die Anmeldeverwaltung den Ordner `/usr/share/xsessions` nicht auf Aktualisierungen überwacht.

2.5.2 Manuelle Einrichtung der Umgebung

Diese Dokumentation enthält Anleitungen, welche Umgebungsvariablen gesetzt werden müssen, um die neu erstellte Arbeitsumgebung zu starten. Diese Anleitungen finden Sie jetzt im Anhang Abschnitt [B.1](#).

Beabsichtigen Sie, Ihre eigene Anmeldeverwaltung einzurichten, finden Sie Informationen dazu in diesem Anhang oder lesen Sie die Datei `sample-kde-env-master.sh`, die im Quelltextpaket von `kdesrc-build` enthalten ist.

2.6 Auswahl und Organisation der Module

2.6.1 Organisation der KDE-Software

Die KDE-Software ist in verschiedene Komponenten aufgeteilt, die meisten können mit `kdesrc-build` erstellt werden. Das Verständnis dieser Organisation hilft dabei, die gewünschten Module zum Erstellen richtig auszuwählen.

1. Auf der untersten Ebene befindet sich die Qt™-Bibliothek, eine sehr leistungsfähige „Bibliothek“ für mehrere Plattformen. KDE basiert auf Qt™, ebenso wie einige Bibliotheken außerhalb von KDE, die aber für KDE erforderlich sind. Mit `kdesrc-build` können Sie Qt™ erstellen oder die bereits auf Ihrem System installierte Version verwenden, wenn Sie aktuell genug ist.
2. Auf Qt™ bauen weitere Bibliotheken auf, die für die Funktionen der KDE-Software erforderlich sind. Einige dieser Bibliotheken sind wegen des breiten Anwendungsbereichs nicht Bestandteil von KDE, aber trotzdem zwingend für die KDE-Plattform nötig. Diese Bibliotheken sind im Modul `kdesupport` zusammengefasst, gehören aber nicht zu den „Frameworks“-Bibliotheken.
3. Zu diesen wichtigen Bibliotheken gehören die [KDE-Frameworks](#), manchmal abgekürzt als KF5, die wesentliche Bibliotheken für den KDE Plasma-Desktop, KDE-Anwendungen und andere Software von Drittanbietern.
4. Auf diesen Frameworks-Bibliotheken als Basis kommt dann Folgendes:
 - Programme „anderer Anbieter“. Diese Programme benutzen die KDE-Frameworks oder benötigen KDE-Plasma zur Ausführung, werden aber nicht vom KDE-Projekt entwickelt und haben auch keine Verbindung dazu.
 - Plasma, eine vollständige „Arbeitsflächen“-Umgebung. Das sehen Benutzer normalerweise, wenn Sie sich für KDE anmelden.
 - Die KDE Anwendungen. Dies ist eine Sammlung von nützlicher Software, die in der Plattform und der Plasma -Arbeitsfläche in einzelnen Modulgruppen enthalten ist, darunter Dienstprogramme wie Dolphin, Spiele wie KSudoku und Produktivitätssoftware, die von KDE wie z. B. Kontakt.
 - Schließlich gibt es noch Softwaresammlung, die auch in Modulen zusammengefasst sind, deren Entwicklung durch die KDE-Infrastruktur (wie Übersetzung, Versionsverwaltung, Fehlerverfolgung usw.) unterstützt wird. Diese Programme werden aber unabhängig von den Anwendungen und Plasma veröffentlicht und sind nicht Bestandteil der Softwaresammlung. Diese Module zusammen werden „Extragear“ genannt.

2.6.2 Auswahl der Module zum Erstellen

Die Auswahl der zu erstellenden Module wird durch die [Einrichtungsdatei](#) gesteuert. Darin folgt nach dem Abschnitt `global` eine Liste der zu erstellenden Module, eingefasst durch die Zeilen „`module`“ ... „`end module`“. Ein Beispiel für einen Moduleintrag finden Sie in [Beispiel 2.4](#).

Example 2.4 Beispieleintrag für ein Modul in der Einrichtungsdatei

```
module kdesrc-build-git
  # Optionen für dieses Modul werden hier eingetragen, zum Beispiel:
  repository kde:kdesrc-build
  make-options -j4 # Startet vier Compilerläufe gleichzeitig
end module
```

ANMERKUNG

n der Praxis wird dieses Module meist nicht direkt verwendet. Stattdessen werden die meisten Module über Modulgruppen angegeben, wie unten beschrieben.

Wenn Sie nur `module`-Einträge verwenden, baut `kdesrc-build` diese in der von Ihnen angegebenen Reihenfolge und versucht nicht, andere Repositories herunterzuladen, außer denen, die Sie direkt angeben.

2.6.3 Modulgruppen

Der KDE-Quellcode ist in eine große Anzahl von relativ kleinen Git-Repositories aufgeteilt. Um die Verwaltung der großen Anzahl von Repositories zu erleichtern, die an einer sinnvollen KDE-basierten Installation dazugehören, unterstützt `kdesrc-build` die Gruppierung mehrerer Module und behandelt sie als „Modulgruppen“.

2.6.3.1 Das grundlegende Konzept der Modulgruppen

Durch Verwendung von Modulgruppen können Sie schnell viele Git-Module deklarieren, die heruntergeladen und erstellt werden sollen, ohne jedes Modul einzeln eintragen zu müssen. Mit der Option `repository` wird automatisch ermittelt, wo jedes einzelne Modul heruntergeladen werden kann und jede weitere Option der Modulgruppe wird auch für jedes auf diese Weise erzeugte Modul kopiert und angewendet.

Example 2.5 Modulgruppen verwenden

```
global
  git-repository-base kde-git kde:
end global

module qt
  # Optionen entfernt, um das Beispiel kurz zu halten
end module

module-set kde-support-libs
  repository kde-git
  use-modules automoc attica akonadi
end module-set

# Andere Module soweit erforderlich ..
module kdesupport
end module
```

In [Beispiel 2.5](#) wird ein Beispiel einer einfachen Modulgruppe gezeigt. Wenn `kdesrc-build` eine Modulgruppe bearbeitet, wird für jedes in der Option `use-modules` angegebene Modul eine einzelne Moduldefinition aus dem Wert der Option `repository` der Modulgruppe und dem angegebenen Namen des Moduls erstellt.

Zusätzlich können noch weitere Optionen für eine Modulgruppe gesetzt werden. Diese Optionen werden dann automatisch in jede einzelne Moduldefinition kopiert. Mit Modulgruppen ist es einfach, viele Git-Module zu deklarieren, die alle auf der gleichen Archiv-URL basieren. Zusätzlich ist es auch möglich, für Modulgruppen Aliasnamen wie im Beispiel zu benutzen. Mit einem Alias können Sie eine gesamte Modulgruppe auf der Befehlszeile angeben.

2.6.3.2 Besondere Unterstützung für KDE-Modulgruppen

Die bisher beschriebene Unterstützung für Modulgruppen gilt allgemein für alle Git-basierten Module. Für die KDE-Repositories enthält `kdesrc-build` zusätzliche Funktionen, um die Arbeit für Benutzer und Entwickler zu erleichtern. Diese Unterstützung wird durch die Angabe von `kde-projects` für die Option `repository` für diese Modulgruppen aktiviert.

`kdesrc-build` erstellt normalerweise nur die Module in der Reihenfolge, wie Sie in Ihrer Einrichtungsdatei angegeben sind. Aber mit einer Modulgruppe `kde-projects` kann `kdesrc-build` die Auflösung von Abhängigkeiten von KDE-spezifischen Modulen durchführen und darüber hinaus automatisch Module in den Build einbeziehen, auch wenn diese nur indirekt angegeben sind.

Example 2.6 „kde-projects“-Modulgruppen verwenden

```
# Nur juk aus kde/kdemultimedia/juk)
module-set juk-set
    repository kde-projects
    use-modules juk
end module-set

# Alle Module in kde/multimedia/*, einschließlich juk,
# aber keine anderen Abhängigkeiten
module-set multimedia-set
    repository kde-projects
    use-modules kde/multimedia
end module-set

# Alle Module in kde/multimedia/* und alle kde-projects-
# Abhängigkeiten für kde/kdemultimedia
module-set multimedia-deps-set
    repository kde-projects
    use-modules kde/multimedia
    include-dependencies true
end module-set

# Alle Module, die aus diesen drei Modulgruppen generiert wurden, erhalten ←
    automatisch
# die richtige Reihenfolge der Abhängigkeiten, unabhängig von den ←
    Einstellungen für include-dependencies
```

TIP

Diese Modulgruppen `kde-projects` sind die wichtigste Methode, die Module zum Erstellen zu deklarieren.

Alle Modulgruppen verwenden die Optionen `repository` und `use-modules`. Die Modulgruppe `kde-projects` hat einen vordefinierten Wert für die Option `repository`, aber andere Arten von Modulgruppen verwenden ebenfalls die Option `git-repository-base`.

2.6.4 Die offizielle KDE-Moduldatenbank

Die Git-Repositories von KDE ermöglichen die Gruppierung verwandter Git-Module z. B. `kde-graphics`. Git erkennt diese Gruppierungen nicht, aber `kdesrc-build` kann diese Gruppen durch **Modulgruppen** mit einer Option `repository` vom Wert „`kde-projects`“ korrekt verarbeiten.

`kdesrc-build` erkennt, dass das `kde-projects`-Repositories besonders bearbeitet werden müssen und passt den Erstellungsprozess dafür an. Unter anderem werden folgende Schritte ausgeführt:

- Herunterladen der neuesten Moduldatenbank vom [KDE-Git-Archiv](#).
- Suche nach einem Modul mit dem in der Option `use-modules` angegebenen Namen in dieser Datenbank.
- Für jedes gefundene Modul sucht `kdesrc-build` das entsprechende Repository in der Datenbank, basierend auf der Einstellung `Branch-group`. Wenn ein Repository existiert und der Branch aktiv ist, verwendet ihn `kdesrc-build` automatisch, um den Quellcode herunterzuladen oder zu aktualisieren.

ANMERKUNG

In der aktuellen Datenbank bestehen einige Modulgruppen nicht nur aus einer Sammlung von Modulen, sondern deklarieren *auch* ein eigenes Git-Archiv. In diesem Fall wird aktuell des Git-Archiv der Gruppe anstelle der darin enthaltenen Archive verarbeitet.

Das folgende Beispiel zeigt, wie die KDE-Moduldatenbank verwendet werden kann, um die Phonon-Multimediabibliothek zu installieren.

```
module-set media-support
  # Als Option muss kde-projects angegeben werden, damit die ↔
  Moduldatenbank verwendet wird.
  repository kde-projects

  # Diese Option bestimmt, nach welchem Modul in der Datenbank gesucht ↔
  wird.
  use-modules phonon/phonon phonon-gstreamer phonon-vlc
end module-set
```

TIP

Hier wird `phonon/phonon` mit der aktuellen Projektdatenbank verwendet, um zwischen der Gruppe von Projekten mit dem Namen „phonon“ und einzelnen Projekten mit diesem Namen zu unterscheiden. In der aktuellen Version von `kdesrc-build` wird die erste Variante benutzt, aber dann werden viel mehr als nötig erstellt.

Das folgende Beispiel ist wahrscheinlich realistischer und zeigt eine Funktion, die nur für die KDE-Moduldatenbank verfügbar ist: Erstellen aller KDE-Grafikprogramme mit einer Deklaration.

```
module-set kdegraphics
  # Als option muss kde-projects für die Moduldatenbank verwendet werden.
  repository kde-projects

  # Diese Option legt fest, welche Module in der Datenbank gesucht werden ↔
  sollen.
  use-modules kdegraphics/libs kdegraphics/*
end module-set
```

Hier werden zwei wichtige Fähigkeiten gezeigt:

1. Mit `kdesrc-build` können Sie Module angeben, die von einem angegebenen Modul abstammen, ohne das übergeordnete Modul selbst zu erstellen, indem Sie die Syntax **module-name /*** verwenden. In diesem Beispiel ist das sogar erforderlich, das das Modul `kdegraphics` als inaktiv markiert ist, so dass es nicht unbeabsichtigt mit allen darin enthaltenen Modulen erstellt wird. Mit der Angabe von untergeordneten Modulen, kann `kdesrc-build` dieses deaktivierte Modul übergehen.
2. `kdesrc-build` fügt außerdem auch mehrfach angegebene Module nur einmal in die Erstellungsliste ein. Dadurch kann `kdegraphics/libs` manuell als erstes Modul zum Erstellen vor `kdegraphics` angegeben werden, ohne dass `kdegraphics/libs` zweimal erstellt wird. Dies war früher für die korrekte Behandlung von Abhängigkeiten erforderlich und bleibt heute eine Ausweichmöglichkeit für den Fall, dass in der KDE-Projektdatenbank Metadaten zu Abhängigkeiten fehlen.

2.6.5 KDE-Projektmodule ausfiltern

Vielleicht möchten Sie alle Programme in einer KDE-Modulgruppe mit *Ausnahme* eines bestimmten Programms erstellen.

In der Gruppe `kdeutils` gibt es zum Beispiel ein Programm mit dem Namen `kremotecontrol`. Hat Ihr Rechner nicht die passende Hardware, um Signale von Fernsteuerungen zu empfangen, werden Sie möglicherweise darauf verzichten wollen, jedesmal `kremotecontrol` herunterzuladen, zu erstellen und zu installieren, wenn Sie `kdeutils` aktualisieren.

Dies erreichen Sie durch die Option `ignore-modules`. Auf der Befehlszeile bewirkt die Option `--ignore-modules` dasselbe, ist aber praktischer, um ein Modul nur einmal herauszufiltern.

Example 2.7 Beispiel für das Ignorieren eines KDE-Projektmoduls aus einer Gruppe

```
module-set utils
  repository kde-projects

  # Diese Option bestimmt, nach welchem Modul in der Datenbank gesucht ←
  wird.
  use-modules kdeutils

  # Diese Option legt fest, welche der mit use-modules oben gewählten ←
  Module ausgelassen wird.
  ignore-modules kremotecontrol
end module-set

module-set graphics
  repository kde-projects

  # Diese Option bestimmt, nach welchem Modul in der Datenbank gesucht wird.
  use-modules extragear/graphics

  # Diese Option legt fest, welche der mit use-modules oben gewählten Module ←
  ausgelassen wird.
  # In diesem Fall werden *sowohl* extragear/graphics/kipi-plugins und
  # extragear/graphics/kipi-plugins/kipi-plugins-docs ignoriert
  ignore-modules extragear/graphics/kipi-plugins
end module-set
```

2.7 Zusammenfassung

Dies sind die wichtigsten Funktionen und Konzepte, die für den Einstieg in `kdesrc-build`

Weitere Informationen finden Sie in dieser Dokumentation. Insbesondere die [Liste der Befehlszeilenoptionen](#) und die [Tabelle der Optionen für Einrichtungsdateien](#) sind nützliche Referenzen.

Die KDE-Gemeinschaft unterhält außerdem [eine Online-Wiki-Referenz, wie wie man den Quellcode baut](#), darin finden Sie auch Hinweise und weitere Anleitungen zu `kdesrc-build`.

Kapitel 3

Besondere Funktionen des Skripts

3.1 Übersicht der Funktionen

kdesrc-build beinhaltet die folgenden Funktionen und Besonderheiten:

- Sie können alle Vorgänge „simulieren“ anstatt sie wirklich ausführen zu lassen. Benutzen Sie dazu die Option `--pretend` oder `-p` und das Skript wird eine detaillierte Beschreibung aller Befehle ausgeben, die es ausführen würde, ohne tatsächlich etwas zu unternehmen. Wenn Sie jedoch noch nie kdesrc-build ausgeführt haben, sollten Sie zuerst kdesrc-build mit der Option `kdesrc-build --metadata-only` ausführen, damit `--pretend` funktioniert.

TIP

Für noch ausführlichere Ausgaben von kdesrc-build verwenden Sie die Option `--debug`.

- kdesrc-build kann mit Unterstützung der KDE-FTP-Server einige Subversion-Module beschleunigt herunterladen. Wenn für diese Module bereits komprimierte Pakete auf der Webseite vorhanden sind, werden diese geladen und ausgepackt.

TIP

Allgemein sind keine besonderen Vorbereitungen nötig, um das erste Herunterladen eines Git-Moduls durchzuführen, da in jedem Fall das gesamte Gut-Archiv heruntergeladen werden muss. Daher ist für den Server eindeutig klar, was gesendet werden muss.

Dies ist für Sie schneller und verringert die Last der kde.org-Server für anonymes Subversion.

- Um den Ablauf weiter zu beschleunigen, wird das Kompilieren eines Moduls sofort gestartet, wenn es vollständig heruntergeladen wurde. Dies funktioniert seit Version 1.6.
- Ausgezeichnete Unterstützung für das Erstellen der Qt™-Bibliothek, wenn eine KDE-Software, die Sie erstellen möchten, eine aktuelle Version von Qt™ benötigt, die in Ihrer Distribution nicht vorhanden ist.
- kdesrc-build erfordert keine grafische Bedienungsfläche (GUI), Sie können KDE-Software ohne eine zusätzliche grafische Bedienungsfläche erstellen.
- Standardeinstellungen für alle Module wie Optionen zum Kompilieren und Konfigurieren. Diese Optionen können gleichfalls für einzelne Module angegeben werden.

kdesrc-build verwendet [Standard-Parameter](#), das erspart die Eingabe und verhindert Tippfehler. Hinweis: Dies gilt nicht, wenn benutzerdefinierte Entwicklungs-Werkzeuge z. B. mit [cmake-toolchain](#) verwendet werden.

- kdesrc-build kann einen bestimmten [Zweig \(branch\) oder Tag](#) oder auch eine vorgegebene [Revision](#) eines Moduls herunterladen.
- kdesrc-build kann automatisch für einen Quelltextordner das Archiv zum Herunterladen wechseln, zum Beispiel von [Zweig \(branch\)](#) auf [trunk](#). Das geschieht automatisch, wenn Optionen geändert werden, die auch die URL des Archivs ändern. Es muss dazu aber die Befehlszeilenoption [--src-only](#) verwendet werden, damit kdesrc-build den Wechsel durchführt.
- kdesrc-build kann ein [Modul nur teilweise](#) herunterladen, wenn Sie zum Beispiel nur ein Programm aus einem großen Modul benötigen.
- Für Entwickler, die `svn+ssh://` verwenden, [überprüft](#) kdesrc-build, ob der Dienst `ssh-agent` läuft, damit es nicht zu wiederholten Passwortabfragen durch SSH kommt.
- Der Ordner zum Erstellen kann [gelöscht](#) werden, nachdem ein Modul installiert wurde, um Speicherplatz auf der Festplatte auf Kosten der Zeit zum erneuten Kompilieren einzusparen.
- Die von kdesrc-build benutzten Ordner können sogar für jedes einzelne Modul angepasst werden.
- Sudo oder ein anderer benutzerdefinierter Befehl kann zur [Installation von Modulen](#) benutzt werden, sodass kdesrc-build nicht mit den Rechten des Systemverwalters ausgeführt werden muss.
- kdesrc-build wird als Voreinstellung mit [reduzierter Priorität](#) ausgeführt, sodass am Rechner auch dann noch gearbeitet werden kann, wenn gerade kdesrc-build ausgeführt wird.
- Unterstützt KDE's [Tags und Zweige \(branches\)](#)
- Das [erneute Erstellen](#) kann mit einem angegebenen Modul wieder aufgenommen werden. Es können bei der Ausführung auch [einige Module ausgelassen](#) werden.
- kdesrc-build zeigt den [Fortschritt](#) beim Kompilieren mit CMake an und schreibt die Dauer des Kompilierens in das Protokoll.
- Die voreingestellten Optionen sind gut geeignet für die Installation von KDE aus dem anonymen Quelltext-Archiv für einen einzelnen Benutzer.
- Benutzung der Tilde als Platzhalter in den Optionen. Beispielsweise ist folgender Befehl möglich:

```
qtdir ~/kdesrc/build/qt
```

- Erstellt automatisch ein System mit getrennten Ordnern für den Quelltext und das Kompilieren, damit bleibt der Quelltextordner unverändert.
- Es können sowohl globale Einstellungen für alle Module als auch spezifische Einstellungen für bestimmte Module vorgenommen werden.
- Erzwingt ein vollständig neues Erstellen durch Ausführung von kdesrc-build mit der Option `--refresh-build`.
- Es können verschiedene Umgebungsvariablen benutzt werden, die den Kompilervorgang beeinflussen. Dazu gehören unter anderem `KDEDIR`, `QTDIR`, `DO_NOT_COMPILE` und `CXXFLAGS`.
- Befehls-Protokolle. Alle Protokolle werden durchnummeriert und mit einem Datums-Stempel versehen, sodass Sie pro Skript-Durchlauf ein Protokoll erhalten. Des weiteren wird mit jedem Skript-Durchlauf ein symbolischer Link namens `latest` erzeugt, der auf das jeweils letzte Protokoll verweist.
- Sie können auch nur einzelne Teile von KDE aus einem Subversion-Modul herunterladen lassen. Beispielsweise könnten Sie nur `taglib` aus `kdesupport` herunterladen.

3.2 Protokolle von kdesrc-build

3.2.1 Übersicht der Protokollfunktionen

Beim Protokollieren schreibt kdesrc-build die Ausgabe jedes Befehls bei der Ausführung in eine Datei. So können bei einem Fehler die Protokolle nach der möglichen Ursache durchsucht werden.

3.2.1.1 Ordnerlayout für Protokolle

Protokolle werden immer in einem eigenen Ordner gespeichert. Dieser Ordner wird mit der Option `log-dir` angegeben, die Voreinstellung dafür ist `/${source-dir} /log`. Dabei wird `/${source-dir}` durch den Wert der Option `source-dir` ersetzt. Im weiteren Text wird für diesen Wert `/${log-dir}` benutzt.

Der Ordner `/${log-dir}` hat weitere Unterordner, einen für jeden Ausführungszeitpunkt von kdesrc-build. Jeder Ordner wird mit dem Datum und einer fortlaufenden Zahl benannt. Wird kdesrc-build zum Beispiel am 26. Mai 2004 zum zweiten mal ausgeführt, lautet der Name des Ordners `2004-05-26-02`.

Für einen leichteren Zugriff erstellt kdesrc-build eine Verknüpfung zu den Protokollen der letzten Ausführung mit dem Namen `latest`. Daher sind die aktuellen Protokolle immer unter `/${log-dir} /latest` zu finden.

Der Protokollordner für jede Ausführung von kdesrc-build enthält Unterordner, einen für jedes Modul, das erstellt wurde. Zusätzlich wird die Datei `build-status` angelegt, in der aufgeführt ist, welche Module erstellt bzw. nicht erstellt werden konnten.

ANMERKUNG

Wenn ein Modul weitere Untermodule enthält, wie zum Beispiel `extragear/multimedia`, `playground/utils` oder `KDE/kdelibs`, dann wird dieses Ordnerlayout auch für die Protokollordner verwendet. Das Protokoll für die letzte Ausführung von kdesrc-build für das Modul `KDE/kdelibs` finden Sie in `/${log-dir} /latest/KDE/kdelibs`, nicht in `/${log-dir} /latest/kdelibs`.

Im Protokollordner eines Moduls werden Dateien für jede Operation angelegt, die kdesrc-build ausführt. Aktualisiert kdesrc-build ein Modul, werden Dateien wie `svn-co.log` - wenn ein Modul das erste Mal heruntergeladen wird - oder `svn-up.log` - wenn das Modul aktualisiert wird - erstellt. Wird der Befehl **configure** ausgeführt, so wird das in der Datei `configure.log` protokolliert.

Ist ein Fehler aufgetreten, finden Sie eine Erklärung der Ursache in einer dieser Dateien. kdesrc-build erstellt eine Verknüpfung von der Datei mit dem Fehlerprotokoll wie zum Beispiel `build-1.log` zu der Datei `error.log`.

Um zu sehen, warum ein Modul bei der letzten Ausführung von kdesrc-build nicht erstellt werden konnte, sollten Sie daher als erstes die Datei `/${log-dir} /latest/ module-name /error.log` begutachten.

TIP

Wenn die Datei `error.log` leer ist, insbesondere nach einer Installation, dann ist wahrscheinlich kein Fehler aufgetreten. Einige Programme im KDE-Erstellungssystem melden manchmal fälschlicherweise einen Fehler.

Außerdem können manche Befehle die Ausgabeumleitung von kdesrc-build unter bestimmten Umständen umgehen, normalerweise beim ersten Herunterladen aus dem Subversion-Archiv. Die Fehlerausgabe wird in diesem Fall nicht in die Protokolldatei geschrieben, sondern erscheint auf der Konsole oder im Terminal, in dem kdesrc-build ausgeführt wird.

Kapitel 4

kdesrc-build einrichten

4.1 Übersicht über die Einrichtung von kdesrc-build

Für die korrekte Funktion benötigt das Skript eine Datei namens `.kdesrc-buildrc` in Ihrem persönlichen Ordner. Diese beschreibt unter anderem die KDE-Module, die Sie herunterladen und installieren möchten, und alle Optionen und Einstellungs-Parameter, die für diese Module verwendet werden sollen.

4.1.1 Layout der Einrichtungsdatei

4.1.1.1 Globale Einstellungen

In der Einrichtungsdatei werden zuerst die globalen Einstellungen wie folgt festgelegt:

```
global
option-name option-wert
[...]
end global
```

4.1.1.2 Einstellungen für Module

Danach folgen eine oder mehrere Modul-Abschnitte, die in einer der beiden folgenden Formen angegeben werden:

- ```
module modul-name
option-name option-wert
[...]
end module
```

- ```
module-set module-set-name
  repository kde-projects or git://host.org/path/to/repo.git
  use-modules module-names

# Andere Optionen können auch gesetzt werden
option-name option-value
[...]
end module-set
```

WICHTIG

Beachten Sie, dass die zweite Form „module-set“ *nur mit Git-basierten Modulen* funktioniert.

Für Subversion-Module muss `modul-name` ein Modulname aus dem KDE-Subversion-Archiv sein (zum Beispiel `kdeartwork` oder `kde-wallpapers`). Dies können Sie umgehen, wenn Sie die [Subversion-URL direkt eingeben](#).

Für Git-Module ist im Prinzip jeder beliebige Name zulässig, solange er in der Einrichtungsdatei eindeutig ist und nur einmal vorkommt. Beachten Sie, dass die Namen der Quelltext- und Erstellungssystem auf dem Modulnamen basieren, solange Sie nicht die Option `dest-dir` benutzen.

Bei *Git-Modulgruppen* jedoch muss die Option `module-names` den tatsächlichen Git-Modulen in der ausgewählten Option `repository` entsprechen. Weitere Informationen finden Sie in den Abschnitten [git-repository-base](#) oder [use-modules](#).

4.1.1.3 Verarbeitung von Options-Werten

Generell wird der gesamte Inhalt einer Zeile nach `option-name` als `option-value` verwendet.

`kdesrc-build` ersetzt eine Folge von `#{Name-der-Option}` durch den Wert dieser Option aus der globalen Einrichtung. Damit können Sie den Wert vorhandener Optionen referenzieren, einschließlich Optionen, die bereits durch `kdesrc-build` festgelegt sind.

Ein Beispiel dazu finden Sie unter [Beispiel 2.1](#).

4.1.1.4 „options“-Module

Es gibt einen abschließenden Eintrag `options-Gruppen` für die Einrichtungsdatei, der immer dann eingesetzt werden kann, wenn auch `module` oder `module-set` verwendet werden kann.

```
options module-name
option-name option-value
[...]
end options
```

Für eine `options`-Gruppe können wie in einer Modul-Deklaration Optionen angegeben werden. Die `options`-Gruppe wird für ein vorhandenes Modul definiert. Alle auf diese Weise gesetzten Optionen *überschreiben* die Optionen, die für das zugehörige Modul definiert wurden.

WICHTIG

Der zugehörige Modulname *muss* dem in der Deklaration von `options` angegebenen Namen entsprechen. Achten Sie daher besonders auf die Schreibweise.

Dies ermöglicht die Deklaration einer gesamten Modulgruppe `module-set`, in der die gleichen Optionen für alle Module benutzt werden, aber mit der `options`-Gruppe Änderungen für einzelne Module eingestellt werden.

`options`-Gruppen können auch für benannte Modulgruppen gelten. Dies ermöglicht es erfahrenen Benutzern, eine gemeinsame Einrichtungsdatei mit Deklarationen für `module-set` als Basis zu verwenden und dann Änderungen an den Optionen vornehmen, die von diesen Modulgruppen in Einrichtungsdateien verwendet werden, die den `include`-Befehl verwenden, um auf die Basiskonfiguration zu verweisen.

Example 4.1 Beispiel für die Anwendung von „options“

In diesem Beispiel sollen alle Module aus der Softwaregruppe KDE.Multimedia erstellt werden. Für das Programm KMix soll jedoch eine unterschiedliche Version verwendet werden, um vielleicht eine Fehlerkorrektur zu testen. Dies funktioniert wie folgt:

```
module-set kde-multimedia-set
  repository kde-projects
  use-modules kde/kdemultimedia
  branch master
end module-set

# kmix ist in der Gruppe kde/kdemultimedia enthalten, auch wenn kmix bisher ←
  noch
# in dieser Datei genannt wurde, kann kdesrc-build die Änderung erkennen.
options kmix
  branch KDE/4.12
end options
```

Wird nun kdesrc-build ausgeführt, werden alle KDE-Multimediaprogramme aus dem Zweig „master“ des Quelltextarchivs erstellt, nur KMix vom älteren Zweig „KDE/4.12“. Durch die Benutzung von options müssen Sie nicht alle *anderen* KDE-Multimediaprogramme einzeln auflisten, um für sie den richtigen Zweig einzustellen.

ANMERKUNG

Beachten Sie, dass diese Option nur in kdesrc-build ab Version 1.16 oder in der Entwicklerversion von kdesrc-build nach dem 12.01.2014 vorhanden ist.

4.1.2 Andere Einrichtungsdateien einschließen

In einer Einrichtungsdatei können Sie eine Referenz auf andere Dateien mit dem Schlüsselwort `include` und dem Namen einer Datei benutzen. Das führt zum gleichen Ergebnis, als wenn die eingeschlossene Datei an dieser Stelle eingefügt wäre.

Sie können zum Beispiel folgende Einstellung benutzen:

```
global
  include ~/common-kdesrc-build-options

  # Spezielle Optionen hier einfügen.
end global
```

ANMERKUNG

Wenn Sie nicht den vollständigen Pfad zur einzuschließenden Datei angegeben haben, dann wird die Datei vom Ordner mit der Quelldatei aus gesucht. Die Suche wird auch rekursiv ausgeführt.

4.1.3 Häufig verwendete Einstellungsoptionen

Im Folgenden sehen Sie eine Liste von häufig benutzten Optionen. Klicken Sie auf eine Option, um mehr darüber zu erfahren. Eine vollständige Liste aller Optionen finden Sie unter Abschnitt [4.2](#).

- **cmake-options**: Legt fest, welche Parameter zur Konfiguration eines bestimmten Moduls mit CMake übergeben werden sollen.
- **branch**: Zum Herunterladen aus einem bestimmten Zweig (branch) anstelle von /trunk für Subversion oder `master` für Git.
- **configure-flags**: Legen fest, welche Parameter zur Konfiguration von Qt™ benutzt werden soll.
- **kdedir**: Legt den Zielordner für die KDE-Installation fest.
- **make-options**: Übergibt bestimmte Optionen an das Programm Make, wie zum Beispiel die Anzahl der benutzten CPUs.
- **qtdir**: Legt den Pfad zu Qt™ fest.
- **source-dir**: Ändert den Pfad für heruntergeladene Quellen.

4.2 Tabelle der verfügbaren Einstellungsoptionen

Es folgt eine Tabelle der verschiedenen Optionen mit folgenden Informationen:

- Der Name der Option
- Eine Beschreibung des Verhaltens von kdesrc-build beim Erstellen eines Moduls, wenn diese Option sowohl im globalen Abschnitt als auch im Abschnitt für das Modul in der [Einrichtungsdatei](#) gesetzt ist..
- Zusätzliche Hinweise zum Zweck und zur Benutzung der Option.

Name der Option	Modul -> Globales Verhalten	Bemerkungen
apidox		Mit dieser Option wurde die API-Dokumentation der KDE-Module erstellt. Diese Option wurde in kdesrc-build 1.6.3 entfernt. Die API-Dokumentation finden Sie im Internet auf kde.org . Zusätzlich kann die API-Dokumentation für KDE mit dem Skript kdedoxigen.sh aus dem Modul <code>kde-dev-scripts</code> lokal erstellt werden. Weitere Informationen dazu finden Sie auf der KDE TechBase .
apply-qt-patches		Diese Option wurde in kdesrc-build Version 1.10 entfernt. Um den gleichen Effekt zu erreichen, lesen Sie bitte die Informationen zu Abschnitt 6.2.1 und der Option repository .

<p>async</p>	<p>Kann nicht überschrieben werden</p>	<p>Diese Option aktiviert den asynchronen Arbeitsmodus, in dem Quelltexte gleichzeitig heruntergeladen und die Module kompiliert werden, anstatt erst nach dem Laden des Quelltextes mit dem Kompilieren zu beginnen. Diese Option ist als Voreinstellung eingeschaltet, um sie abzuschalten, wird diese Option auf den Wert false gesetzt. Diese Option gibt es seit Version 1.6.</p>
<p>binpath</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option können Sie die Umgebungsvariable PATH für den Kompilervorgang festlegen. Diese Option lässt sich nicht mit einer Modul-Option überschreiben. Die Voreinstellung ist die beim Start des Skripts gesetzte Umgebungsvariable \$PATH. Diese Umgebungsvariable sollte die Pfade zu den notwendigen Entwicklungs-Werkzeugen beinhalten. Die Pfade \$KDEDIR /bin und \$QTDIR /bin werden automatisch hinzugefügt. Sie können auch die Tilde (~) für Pfadangaben verwenden.</p>

branch	Moduleinstellung überschreibt die globale Einstellung	<p>Mit dieser Option legen Sie fest, dass die Quellen aus einem bestimmten KDE-Zweig anstelle von <i>master</i> (bei Git-Modulen) oder <i>trunk</i> (bei Subversion), in denen die KDE-Entwicklung stattfindet, heruntergeladen werden sollen.</p> <p>Um beispielsweise KDE 4.6 herunterzuladen, stellen Sie die Option auf <code>4.6</code> ein.</p> <p>Wenn das Herunterladen eines Zweigs mit dieser Option fehlschlägt, müssen Sie möglicherweise die korrekte Adresse zum Herunterladen manuell eingeben. Verwenden Sie dazu die Optionen module-base-path oder override-url.</p> <div data-bbox="997 1045 1338 1281" style="border: 1px solid black; padding: 5px;"><p>ANMERKUNG</p><p>Für die meisten KDE-Module sollten Sie vermutlich stattdessen die Option branch-group benutzen und diesen Option nur ausnahmsweise verwenden.</p></div>
--------	---	---

branch-group	Moduleinstellung überschreibt die globale Einstellung	<p>Stellen Sie diese Option auf eine allgemeine Gruppe ein, aus der Sie Module auswählen möchten. Für unterstützte Git-Modularten bestimmt kdesrc-build automatisch den aktuell zu verwendenden Zweig auf der Grundlage der Regeln, die von den KDE-Entwicklern festgelegt wurden. Diese Regeln finden Sie im Quelltextarchiv <code>kde-build-metadata</code> in Ihrem Quelltextordner. Wenn der Zweig bestimmt ist, wird dieser benutzt, als wenn Sie ihn selbst mit der Option <code>branch</code> festgelegt hätten. Dies ist besonders hilfreich, wenn Sie nur die Aktualisierungen an einigen normalen Entwicklungen verfolgen möchten, ohne sich dabei um alle Änderungen an Zweignamen kümmern zu müssen. Zum 11.08.2013 gibt es folgende aktuelle Zweiggruppen:</p> <ul style="list-style-type: none">• <code>stable-qt4</code>, für die Verfolgung von Fehlerkorrekturen an den KDE-Bibliotheken und -Programmen auf der Basis von Qt™ 4.• <code>latest-qt4</code>, für die Verfolgung von Entwicklung und neuer Funktionen an den KDE-Bibliotheken und -Programmen auf der Basis von Qt™ 4.• <code>kf5-qt5</code>, für die Verfolgung der „bleeding-edge“ Entwicklung des zukünftigen Qt™ 5-basierten KDE Frameworks 5, Plasma Workspace 2, usw.
--------------	---	--

<p>build-dir</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Verwenden Sie diese Option zum Ändern des Ordners, in dem die Quellen kompiliert werden sollen. Sie können diese Option auf drei Arten verwenden:</p> <ol style="list-style-type: none">1. Relativ zum KDE-Subversion-Quellordner (siehe die Option source-dir). Dies ist die Voreinstellung und dieser Modus wird automatisch verwendet, wenn Sie einen Ordner-Namen eingeben, der nicht mit einer Tilde (~) oder einem Schrägstrich (/) beginnt. Die Voreinstellung ist build.2. Absoluter Pfad. Wenn Sie einen Pfad eingeben, der mit einem Schrägstrich (/) beginnt, so wird dieser Pfad wie angegeben verwendet. Beispiel: <code>/tmp/kde-obj-dir/ .</code>3. Relativ zum persönlichen Ordner. Wenn Sie einen Pfad angeben, der mit einer Tilde (~) beginnt, so wird der Pfad als relativ zu Ihrem persönlichen Ordner angesehen. Dies funktioniert analog zu der Art und Weise, wie die Shell eine Tilde interpretiert. Beispiel: <code>~/builddir</code> würde den Kompilier-Ordner auf <code>/home/benutzer/builddir</code> einstellen. <p>Diese Option kann für jedes Modul separat eingestellt werden.</p>
------------------	--	---

<p>build-when-unchanged</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option stellen Sie ein, ob kdesrc-build ein immer Modul erstellt, auch wenn der Quelltext nicht geändert wurde. Mit der Option <code>build-when-unchanged</code> gleich true erstellt kdesrc-build ein Modul immer neu, auch wenn der Quelltext dieses Moduls nicht geändert wurde. Dies ist die Voreinstellung, damit lassen sich Module meistens erfolgreich erstellen. Mit der Option <code>build-when-unchanged</code> gleich false wird ein Modul nur dann neu erstellt, wenn entweder der Quelltext geändert wurde oder das erneute Erstellen wahrscheinlich erforderlich ist. Mit dieser Einstellung sparen Sie Zeit, besonders wenn Sie kdesrc-build täglich oder noch häufiger ausführen.</p> <div data-bbox="997 1249 1336 1589" style="border: 1px solid black; padding: 5px;"><p>WICHTIG Diese Funktion dient nur zur Optimierung. Wie viele Optimierungen kann es Probleme mit einer korrekten Installation geben. Änderungen in qt oder kdelibs erfordern zum Beispiel das erneute Erstellen anderer Module, auch wenn deren Quelltext unverändert ist.</p></div>
-----------------------------	--	--

<p>checkout-only</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit Hilfe dieser Option können Sie Subversion-Quelltexte nacheinander herunterladen. Die Einstellung für diese Option sollte eine durch Leerzeichen getrennte Liste der herunterzuladenden Ordner sein. Zwar überschreibt diese Option die globale Version, es ist jedoch dennoch nicht sinnvoll, die Option global einzustellen. Beachten Sie, dass diese Einstellung bei Modulen aus Git-Archiven wirkungslos ist. Der Grund dafür liegt in der Arbeitsweise des Git-Quelltext-Verwaltungssystems. Ein Beispiel finden Sie unter Abschnitt 6.3.1.1.</p>
<p>cmake-generator</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Verwenden Sie diese Option, um anzugeben, welches Generierungs-Programm mit CMake verwendet werden soll. Derzeit werden sowohl Ninja und Unix Makefiles als auch darauf basierende Generierungs-Programme wie Eclipse CDT4 - Ninja unterstützt. Ungültige nicht unterstützte Werte werden ignoriert und nicht beachtet. Ist diese Option nicht gesetzt, wird Unix Makefiles standardmäßig verwendet. Wenn ein gültiges Generierungs-Programm auch über cmake-options angegeben wird, überschreibt das den Wert für cmake-generator.</p>

cmake-toolchain	Moduleinstellung überschreibt die globale Einstellung	<p>Mit dieser Option wird eine Datei mit Entwicklungs-Werkzeugen für CMake angegeben. Wenn gültige Entwicklungs-Werkzeuge eingerichtet sind, wird kdesrc-build <i>Umgebungsvariablen nicht mehr automatisch setzen</i>. Sie können mit set-env, binpath und libpath die Umgebung manuell einrichten, wenn Ihre Entwicklungs-Werkzeuge nicht sofort mit kdesrc-build funktionieren. Weitere Informationen finden Sie im Abschnitt Übersicht der Standard-Parameter von kdesrc-build.</p> <p>Wenn ein gültiges Entwicklungs-Werkzeug auch über cmake-options angegeben wird, überschreibt das den Wert für <code>cmake-toolchain</code>.</p>
-----------------	---	--

<p>cmake-options</p>	<p>Wird den globalen Optionen für das Standard-Erstellungssystem angefügt, überschreibt die globale Einstellung für andere Erstellungssysteme.</p>	<p>Mit Hilfe dieser Option können Sie die Parameter festlegen, die CMake für das jeweilige Modul übergeben werden. Wenn Sie diese Option global einstellen, wird sie für alle Module angewandt. Wenn Optionen für einzelne Module angegeben sind, werden sie nach den globalen Optionen eingefügt. Dadurch können CMake-Optionen für alle Module im globalen Abschnitt eingegeben werden.</p> <p>Diese Option wird nicht für qt angewendet, da CMake dafür nicht benutzt wird. Verwenden Sie stattdessen die Option configure-flags. Wenn ein gültiges Generierungs-Programm unter den aufgelisteten Optionen angegeben ist, überschreibt das den Wert von cmake-generator. Ungültige, nicht unterstützte Generierungs-Programme werden ignoriert und nicht an CMake übergeben. Wenn ein gültiges Entwicklungs-Werkzeuge unter den aufgelisteten Optionen angegeben ist, überschreibt das den Wert von cmake-toolchain. Ungültige Entwicklungs-Werkzeuge werden ignoriert und nicht an CMake übergeben. Da diese Optionen direkt ohne weitere Bearbeitung an CMake übergeben werden, müssen Sie in der gleichen Form wie für CMake eingetragen werden. Zum Beispiel:</p> <pre>cmake-options -DCMAKE_BUILD_TYPE=RelWithDebInfo</pre> <p>Solange alle anderen Optionen richtig gesetzt sind, wird diese Option nicht benötigt und kann leer bleiben. kdesrc-build setzt alle <i>erforderlichen</i> Parameter</p>	
----------------------	--	--	--

colorful-output	Kann nicht überschrieben werden	Wenn Sie diese Option auf false setzen, wird die farbige Ausgabe von kdesrc-build unterbunden. Die Voreinstellung ist <i>true</i> . Beachten Sie, dass kdesrc-build die Farbcodes nur auf ein Terminal ausgibt (wie beispielsweise xterm, Konsole oder die normale Linux [®] -Konsole).
compile-commands-export	Moduleinstellung überschreibt die globale Einstellung	Ermöglicht die Erzeugung einer Datei <code>compile_commands.json</code> über CMake im Erstellungsordner. Diese Option ist standardmäßig auf <i>true</i> , setzen Sie sie auf false , um dieses Verhalten zu deaktivieren.
compile-commands-linking	Moduleinstellung überschreibt die globale Einstellung	Ermöglicht das Erstellen von symbolischen Verknüpfungen von <code>compile_commands.json</code> , die über CMake innerhalb des Erstellungsordners erzeugt werden, auf den passende Quellcodeordner. Diese Option ist standardmäßig auf <i>false</i> gesetzt, setzen Sie sie auf true , um die automatische Erstellung von symbolischen Verknüpfungen zu aktivieren.
configure-flags	Wird den globalen Optionen für das Standard-Erstellungssystem angefügt, überschreibt die globale Einstellung für andere Erstellungssysteme.	Mit Hilfe dieser Option können Sie die Parameter festlegen, die dem Skript <code>./configure</code> für das jeweilige Modul übergeben werden. Wenn Sie diese Option global einstellen, wird sie für alle Module angewandt. <i>Diese Option funktioniert nur für qt.</i> Die Einstellungen für KDE-Module werden mit cmake-options vorgenommen.

<p>custom-build-command</p>	<p>Moduleinstellung überschreibt die globale Einstellung (Option für das Erstellen)</p>	<p>Diese Option kann verwendet werden, um zum Beispiel einen anderen Befehl als make für die Ausführung des Erstellungsprozesses zu verwenden. Normalerweise funktioniert kdesrc-build auch ohne die Option korrekt, aber es kann für alternative Erstellungssysteme nützlich sein. Der Wert dieser Option wird üblich in der Befehlszeile ausgeführt und durch die Option make-options wie sonst auch verändert.</p>
<p>cxxflags</p>	<p>Wird den globalen Optionen für das Standard-Erstellungssystem angefügt, überschreibt die globale Einstellung für andere Erstellungssysteme.</p>	<p>Mit dieser Option können Sie die Parameter für das Erstellen der Module festlegen. Diese Option wird hier angegeben und nicht einfach an configure-flags oder an cmake-options mit angehängt, weil sie auch die Umgebungsvariable <code>CXXFLAGS</code> während des Kompilierens setzt. Beachten Sie, dass in KDE 4 für jedes Modul, das CMake verwendet, die Option <code>CMAKE_BUILD_TYPE</code> auf den Wert none eingestellt werden muss. Dazu dient die Option cmake-options.</p>
<p>dest-dir</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option wird der Ordnername eines Moduls auf der Festplatte eingestellt. Das Modul <code>extragear/network</code> zum Beispiel kann mit dieser Option in <code>extragear-network</code> umbenannt werden. Das Trennzeichen für Ordner sollte nicht im Namen benutzt werden, dadurch gibt es Probleme mit den Optionen build-dir und source-dir.</p>

disable-agent-check	Kann nicht überschrieben werden	Wenn Sie SSH zum Herunterladen der Subversion-Quelltexte verwenden (zum Beispiel mit dem Protokoll svn+ssh), wird kdesrc-build überprüfen, dass Ihr SSH-Agent auch einige SSH-Identitäten verwalten kann. Auf diese Weise soll SSH daran gehindert werden, Sie andauernd zur Passwordeingabe aufzufordern. Sie können diese Prüfung unterbinden, indem Sie die Option <code>disable-agent-check</code> auf true einstellen.
do-not-compile	Moduleinstellung überschreibt die globale Einstellung	Verwenden Sie diese Option, um bestimmte Ordner aus einem Modul nicht zu erstellen. Die Namen dieser Ordner werden durch Leerzeichen getrennt. Beachten Sie, dass die Quellen zu den übersprungenen Programmen dennoch heruntergeladen werden. Um auch dies zu unterbinden, können Sie die Option <code>checkout-only</code> verwenden. Geben Sie mit dieser Option nur die Ordner an, die Sie herunterladen möchten. Um zum Beispiel JuK und KsCD aus dem Modul <code>kdemultimedia</code> nicht zu erstellen, fügen Sie „do-not-compile juk kscd“ zu den Einstellungen für <code>kdemultimedia</code> hinzu. Ein Beispiel finden Sie unter Abschnitt 6.3.1.2.
email-address	Kann nicht überschrieben werden	Diese Option wurde in kdesrc-build 1.14 entfernt.
email-on-compile-error	Kann nicht überschrieben werden	Diese Option wurde in kdesrc-build 1.14 entfernt.
inst-apps		Diese Option wurde in Version 1.10 entfernt.

<p>git-desired-protocol</p>	<p>Kann nicht überschrieben werden</p>	<p>Diese Option betrifft nur Module aus einem KDE-Projektarchiv. Mit dieser Option wird eingestellt, welches Netzwerkprotokoll bevorzugt zum Einspielen von Quelltext für diese Module verwendet werden soll. Normalerweise wird das sehr effiziente <code>git</code>-Protokoll benutzt, aber dies kann in manchen Netzwerken wie zum Beispiel Firmen-Intranets oder öffentliche Wi-Fi-Netze gesperrt sein. Als Alternative steht das viel besser unterstützte <code>https</code>-Protokoll für Internet-Webseiten zur Verfügung. Wenn sie eines dieser eingeschränkten Netzwerke benutzen, können Sie diese Option auf <code>http</code> setzen, damit das <code>https</code>-Protokoll bevorzugt wird.</p> <div data-bbox="997 1224 1338 1430" style="border: 1px solid black; padding: 5px;"><p>TIP Sie müssen möglicherweise auch die Option <code>http-proxy</code> setzen, wenn ein HTTP-Proxy für den Netzwerkverkehr erforderlich ist.</p></div> <p>In allen anderen Fällen sollten Sie diese Option nicht benutzen, da das Standardprotokoll effektiver ist. Diese Option wurde in <code>kdesrc-build 1.16</code> hinzugefügt. Vor <code>20.06</code> wurde diese Option verwendet, um die Fetch-URL anstelle der Push-URL zu einzustellen. Ab <code>20.06</code> wird <code>https</code> immer beim Aktualisieren von KDE-Projekten verwendet.</p>
-----------------------------	--	---

git-repository-base

Kann nicht überschrieben werden

Diese seit Version 1.12.1 vorhandene Option kann dazu verwendet werden, um eine Abkürzung für eine bestimmte Basis-URL eines Git-Archivs zu definieren, die dann in der Deklaration von **Modulgruppen** benutzt werden kann. Damit lassen sich dann schnell viele Git-Module zum Erstellen deklarieren. Sie müssen zwei durch Leerzeichen getrennte Werte angeben: Der Name der der Basis-URL zugewiesen wird und die tatsächliche Basis-URL selbst, wie zum Beispiel:

```
global
  # andere Optionen

  # Der übliche Pfad für den anonymen Zugriff
  git-repository-base kde-git kde:
end global

# Modul-Deklarationen

module-set
  # Nun können Sie den vorher definierten Alias
  # in einer Modulgruppe.
  repository kde-git

  use-modules module1.git module2.git
end module-set
```

Mit der Option `use-modules` einer Modulgruppe verhält sich `kdesrc-build` intern so, als ob folgende Optionen angegeben wären:

```
module module1
  repository kde:module1.git
end module

module module2
  repository kde:module2.git
end module
```

Das oben benutzte Präfix `kde:` für Git-Archive ist eine Kurzform, die von `kdesrc-build` automatisch eingerichtet wird. Weiter Informationen dazu finden Sie auf der TechBase-Seite

<p>git-user</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Diese Option ist für KDE-Entwickler vorgesehen. Ein hier eingetragener Wert wird automatisch für die Einstellung zur Identität für das Git-Softwaresystem für <i>neu heruntergeladene</i> Git-Module einschließlich der Mehrheit der KDE-Module. Insbesondere für den Benutzernamen und die E-Mail-Adresse eines jeden neuen Git-Archiv werden die Werte dieser Option verwendet. Der Wert muss im Format User Name <email@example.com> angegeben werden. Ein Entwickler mit dem Namen „Foo Barbaz“ mit der E-Mail-Adresse „foo@abc.xyz“ würde zum Beispiel folgendes verwenden:</p> <pre>git-user Foo Barbaz <foo@abc.xyz></pre> <p>Diese Option wurde in kdesrc-build 15.09 hinzugefügt.</p>
-----------------	--	---

<p>http-proxy</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option wird eine URL eines Proxy-Servers angegeben, die für alle HTTP-Netzwerkverbindungen verwendet werden soll, um zum Beispiel Archive für eine neues Modul oder die KDE-Projektdatenbank herunterzuladen. Zusätzlich versucht kdesrc-build sicherzustellen, dass alle verwendeten Hilfsprogramme auch diesen Proxy-Server benutzen. Dazu wird die Umgebungsvariable <code>http_proxy</code> auf den vorgegebenen Server eingestellt, <i>wenn die Variable bisher noch keinen Wert hatte</i>. Diese Option wurde in kdesrc-build 1.16 hinzugefügt.</p>
<p>ignore-kde-structure</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Diese Option ist veraltet und wird in einer der nächsten Versionen wieder entfernt. Bitte benutzen Sie stattdessen die Option <code>directory-layout</code>, die mehr Auswahl ermöglicht.</p>

directory-layout

Moduleinstellung
überschreibt die globale
Einstellung

Diese Option wird verwendet, um die Ordnerstruktur für kdesrc-build zum Erstellen von Quell- und Build-Ordern anzugeben. Aktuell gibt es drei mögliche Werte: **metadata**, **invent** und **flat**

Das Layout **flat** gruppiert alle Module direkt unter den Quell- und Build-Ordern der obersten Ebene. Statt zum Beispiel `source/extragear/network/telepathy/ktp-text-ui` im Layout **metadata** wird dieses Modul im Layout **flat** unter `source/ktp-text-ui` angelegt.

Mit dem Layout **invent** wird eine Ordnerhierarchie die die relativen Pfade der Repositories auf invent.kde.org angelegt.

Zum Beispiel statt `source/kde/applications/kate` im Layout **metadata** wird dieses Modul im Layout **invent** unter

`source/utilities/kate` angelegt. Dieses Layout

wirkt sich nur auf KDE-Projekte aus. Es ist eine gute Wahl für den Anfang mit kdesrc-build. Schließlich entspricht das Layout **metadata** dem früheren

Standardverhalten. Dieses Layout organisiert die KDE-Projekte entsprechend den Projektpfaden, die in den Projekt-Metadaten für diese Module angegeben sind. Dies ist eine gute Wahl, wenn Sie ein Ordner-Layout wünschen, das mit bestimmten KDE-Prozessen zusammenarbeitet.

Beachten Sie, dass dieser Pfad deshalb nicht immer stabil ist. Als Ergebnis kann es vorkommen, dass kdesrc-build eine alte Kopie

<p>ignore-modules</p>	<p>Kann nicht überschrieben werden</p>	<p>In dieser Option genannte Module, die von kdesrc-build wegen einer Option use-modules bearbeitet würden, werden statt dessen vollständig ausgelassen. Benutzen Sie diese Option, um das gesamte KDE-Projekt mit Ausnahme von bestimmten Modulen zu erstellen. Als Wert der Option muss nicht zwingend der Name des Moduls direkt eingegeben werden. Jedes Modul, dessen vollständige fortlaufend Teile des KDE-Projektmodulpfads auf eine dieser Optionswerte zutrifft, wird ignoriert. Daher können Sie mit dieser Methode mehrere Module ignorieren. Mit einer Option <i>libs</i> werden zum Beispiel sowohl <code>kde/kdegraphics/libs</code> als auch <code>playground/libs</code> ausgeschlossen, aber nicht <code>kde/kdelibs</code>, da mit dem vollständige Teil „kdelibs“ verglichen wird.</p> <div data-bbox="997 1377 1339 1493" style="border: 1px solid black; padding: 5px;"> <p>TIP Weitere Informationen finden Sie unter Beispiel 2.7.</p> </div> <p>Diese Option wurde in kdesrc-build 1.16 hinzugefügt.</p>
-----------------------	--	---

<p>include-dependencies</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Ist diese Option auf true gesetzt, werden auch alle bekannten Abhängigkeiten dieses Moduls erstellt, ohne dass Sie diese Abhängigkeiten - nicht einmal indirekt - angeben müssen.</p> <div data-bbox="997 590 1341 863" style="border: 1px solid black; padding: 5px;"> <p>ANMERKUNG Diese Option funktioniert nur mit kde-project-basierten Modulen. Dazu müssen die von den KDE-Entwicklern betreuten Metadaten für die ausgewählte branch-group richtig eingestellt sein.</p> </div> <p>Diese Option ist standardmäßig aktiviert, um die Erstellung von Anwendungen zu unterstützen, die die neuere Versionen von Qt™ oder Plasma als auf den normalen Betriebssystemen benötigen,</p>
<p>install-after-build</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Diese Option legt fest, dass ein Paket nach erfolgreichem Kompilieren installiert werden soll. Dies ist in der Voreinstellung eingeschaltet. Wenn Sie dieses Verhalten nicht wünschen, stellen Sie die Option in der Einrichtungsdatei auf false. Sie können analog auch den Befehlszeilen-Parameter <code>--no-install</code> verwenden.</p>

<p>install-environment-driver</p>	<p>Kann nicht überschrieben werden</p>	<p>Standardmäßig wird mit kdesrc-build ein Shell-Skript installiert, das in den Profil-Einstellungs-Skripten eines Benutzers geladen werden kann, um die benötigten Umgebungsvariablen für die Ausführung der von kdesrc-build erstellten Plasma-Arbeitsfläche zu setzen. Dieser Treiber verändert die folgenden Dateien:</p> <ul style="list-style-type: none"> • \$XDG_CONFIG_HOME/kde-env-master.sh (normalerweise unter ~/.config/kde-env-master.sh). • \$XDG_CONFIG_HOME/kde-env-user.sh (normalerweise unter ~/.config/kde-env-user.sh). <p>Die Datei kde-env-user.sh ist optional. Sie ist für Benutzeranpassungen vorgesehen. siehe den Abschnitt Problembeseitigung und Fehlersuche der KDE UserBase mit Beispielen für benutzerdefinierte Einstellungen. Diese Einstellungen können vom Benutzer auch an anderer Stelle in den vorhandenen Skripten zur Einrichtung des Profils festgelegt werden. Sie können diese Funktion deaktivieren, indem Sie diese Option auf <i>false</i> setzen und die Option install-session-driver ebenfalls deaktivieren. Diese Option wurde in kdesrc-build 17.08 hinzugefügt.</p>
-----------------------------------	--	---

TIP
kdesrc-build überschreibt Ihre eigenen vorhandenen Dateien...

<p>install-session-driver</p>	<p>Kann nicht überschrieben werden</p>	<p>Ist diese Einstellung aktiviert, installiert kdesrc-build einen Treiber für die grafische Anmeldung, mit dem Sie sich für die durch kdesrc-build erstellte KDE-Arbeitsumgebung anmelden können. Dieser Treiber verändert die folgenden Dateien:</p> <ul style="list-style-type: none"> • <code>~/.xsession</code> • <code>\$XDG_CONFIG_HOME/kde-env-master.sh</code> (normalerweise unter <code>~/.config/kde-env-master.sh</code>). • <code>\$XDG_CONFIG_HOME/kde-env-user.sh</code> (normalerweise unter <code>~/.config/kde-env-user.sh</code>). <p>Wenn Sie eine eigenen Anmeldeprozedur benutzen, können Sie diese Funktion abschalten, indem Sie die Option auf den Wert <code>false</code> setzen. Ist diese Option aktiviert, dann wird auch die Funktion install-environment-driver verwendet. Diese Option wurde in kdesrc-build 1.16 hinzugefügt.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>TIP kdesrc-build überschreibt Ihre eigenen vorhandenen Dateien nicht, es sei denn, Sie geben die Befehlszeilenoption <code>--delete-my-settings</code> an.</p> </div>
-------------------------------	--	---

kdedir	Moduleinstellung überschreibt die globale Einstellung	Mit dieser Option wird der Zielordner festgelegt, in den KDE nach dem Kompilieren installiert wird. Die Voreinstellung ist <code>~/kde</code> . Wenn Sie hier einen Ordner einstellen, für den Systemverwalter-Rechte benötigt werden, sollten Sie auch die Anmerkungen zur Option make-install-prefix lesen.
kde-languages	Kann nicht überschrieben werden	Mit dieser Option kann ausgewählt werden, welche Sprachpakete heruntergeladen und kompiliert werden. So kann KDE mit Übersetzungen in Ihrer Sprache ausgeführt werden. Geben Sie alle Sprachen, die installiert werden sollen, in einer durch Leerzeichen getrennten Liste an. Für jede gewünschte Sprache ist der Sprachencode dafür einzutragen, den Sie auf der Seite http://l10n.kde.org/teams-list.php . Es muss nur eine Sprache ausgewählt werden. Als Voreinstellung wird kein Sprachenpaket heruntergeladen. KDE wird in amerikanischem Englisch angezeigt. Um zum Beispiel die französische Sprache zu installieren, wird für diese Option kde-languages fr eingetragen. Nach der Installation muss diese Sprache dann in den Systemeinstellungen als erste Sprache ausgewählt werden.

libpath	Moduleinstellung überschreibt die globale Einstellung	Mit dieser Option wird die Umgebungsvariable <code>LD_LIBRARY_PATH</code> für den Kompilervorgang gesetzt. Sie können die Option für jedes Modul separat einstellen. In der Voreinstellung ist sie nicht gesetzt, aber die Pfade <code>\$KDEDIR /lib</code> und <code>\$QTDIR/lib</code> werden automatisch hinzugefügt. Sie können für Pfadangaben auch die Tilde (<code>~</code>) benutzen.
log-dir	Moduleinstellung überschreibt die globale Einstellung	Diese Option wird zum Ändern des Ordners für Protokoll-Dateien des Skripts verwendet.
make-install-prefix	Moduleinstellung überschreibt die globale Einstellung	Geben Sie für diese Variable eine Liste mit Befehlen ein. Mehrere Befehle werden durch Leerzeichen getrennt. Die hier eingegebenen Befehle werden vor jedem Aufruf von make install ausgeführt. Dies ist hilfreich, wenn Sie für die Installation einiger Pakete Systemverwalter-Rechte benötigen und daher zum Beispiel Sudo aufrufen müssen. Bitte seien Sie umsichtig im Umgang mit Systemverwalter-Rechten.
make-options	Moduleinstellung überschreibt die globale Einstellung (Option für das Erstellen)	Mit Hilfe dieser Variablen können Sie Befehlszeilen-Parameter an make übergeben. Das kann beispielsweise bei Verwendung von <code>distcc</code> oder auf Systemen mit mehr als einem Prozessor nützlich sein. Nicht alle unterstützten Erstellungs-Systeme verwenden make . Für Systeme, die ninja für die Erstellung wie z.B. <code>Meson</code> verwenden, finden Sie im Abschnitt ninja-Optionen Informationen zur Einstellung.

<p>manual-build</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Stellen Sie diese Option auf true, um zu verhindern, dass ein Modul kompiliert wird. Es wird dennoch weiterhin aktualisiert, wenn Quellen aus dem Subversion-Archiv heruntergeladen werden. Diese Option entspricht dem Befehlszeilen-Parameter <code>--no-build</code>.</p>
<p>manual-update</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Stellen Sie diese Option auf true, um zu verhindern, dass kdesrc-build ein Modul aktualisiert (und auch nicht kompiliert und installiert). Wenn Sie diese Option so verwenden, entspricht das im Grunde dem Auskommentieren - es wird völlig übergangen.</p>
<p>module-base-path</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option können Sie den in kdesrc-build voreingestellten Pfad zu einem Modul übergehen. Das kann zum Beispiel verwendet werden, um für ein Modul einen besonderen Zweig oder eine als Tag vorhandene Version herunterzuladen. Bei der Suche nach dem richtigen Pfad kann Ihnen das KDE-WebSVN behilflich sein.</p> <p>Beachten Sie: kdesrc-build setzt den Pfad nach dem folgenden Schema zusammen: <code>\$svn-server /home/kde/ \$module-base-path</code>.</p> <p>Die Voreinstellung ist entweder <code>trunk/ \$module</code> oder <code>trunk/KDE/ \$module</code>, abhängig vom Namen des Moduls.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>TIP Verwenden Sie stattdessen die Optionen <code>branch</code> oder <code>tag</code>, sofern möglich.</p> </div>

<p>niceness</p>	<p>Kann nicht überschrieben werden</p>	<p>Diese Option können Sie auf einen Wert zwischen 20 und 0 einstellen. Je höher die Zahl, desto niedriger ist die Laufzeit-Priorität für kdesrc-build, d. h. bei einer höheren Zahl ist das Programm „netter“. Die Voreinstellung ist 10.</p>
<p>ninja-options</p>	<p>Moduleinstellung überschreibt die globale Einstellung (Option für das Erstellen)</p>	<p>Setzen Sie diese Variable, um Befehlszeilenoptionen an den Erstellungsbefehl ninja zu übergeben. Dies kann nützlich sein, um die Ausgabe „verbose“ zu aktivieren oder manuell die Anzahl der parallelen Erstellungsvorgänge zu reduzieren, die ninja verwendet.</p> <div style="border: 1px solid black; padding: 5px;"> <p>ANMERKUNG Diese Einstellung steuert nur ninja, wenn sie von kdesrc-build verwendet wird. Das Qt™Modul „webengine“ verwendet ninja indirekt, unterstützt dies aber offiziell nur, wenn es von make gebaut wird. In dieser Situation können Sie <code>NINJAFLAGS</code> angeben, um make die entsprechenden Parameter zu übergeben, wenn es später ninja mit <code>make-options</code> aufruft.</p> <pre>options qtwebengine # Beschränkt make und ninja auf maximal 6 # wenn mehr CPU-Leistung verfügbar ist, um make-options -j6 NINJAFLAGS=-j6 end options</pre> </div>
<p>no-svn</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Steht diese Option auf <code>true</code>, aktualisiert kdesrc-build nicht automatisch den Quelltext für dieses Modul, erstellt aber trotzdem das Modul, sofern dies nicht durch eine andere Option verhindert wird.</p>

Handbuch zum Skript kdesrc-build

no-rebuild-on-fail		Diese Option wurde in Version 1.10 entfernt, da dieses Verhalten wegen Fehlerbehebungen im zugrunde liegenden Erstellungssystem nicht mehr erforderlich ist.
num-cores	Kann nicht überschrieben werden	Diese Option wird durch kdesrc-build mit kdesrc-build-setup oder kdesrc-build --initial-setup auf die Anzahl der CPUs eingestellt, die mit dem externen Programm nproc) ermittelt wird. Kann kdesrc-build die Anzahl der CPUs nicht erkennen, wird ein Wert von 4 verwendet. Unter Beispiel 2.1 finden Sie ein Beispiel für die Verwendung dieser Option. Diese Option wurde in Version 20.07 hinzugefügt.

<p>num-cores-low-mem</p>	<p>Kann nicht überschrieben werden</p>	<p>Diese Option wird durch kdesrc-build mit kdesrc-build-setup oder kdesrc-build --initial-setup definiert, um die Anzahl der CPUs für Module wie qtwebengine mit hohem Speicherbedarf zu begrenzen, damit beim Erstellen der vorhandene Speicher nicht vollständig benutzt wird. Als typische Berechnung wird ein CPU-Kern für jeweils 2 Gigabyte (GiB) des gesamten Speichers benutzt. Es wird mindestens 1 Kern und nicht mehr als die mit der Option num-cores angegebenen Kerne verwendet. Obwohl diese Option für Qt™-Modulen gedacht ist, können Sie sie für beliebige Module auf dieselbe Weise wie num-cores verwenden. Wenn kdesrc-build den verfügbaren Arbeitsspeicher nicht erkennen kann, wird dieser Wert auf 2 gesetzt. Diese Option wurde in Version 20.07 hinzugefügt.</p>
--------------------------	--	---

<p>override-build-system</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Dies ist eine fortgeschrittene Option, sie wurde in kdesrc-build 1.16 hinzugefügt. Normalerweise erkennt kdesrc-build das für ein heruntergeladenes Modul zu verwendende Erstellungssystem. Dies wird durch Suche nach besonderen Dateien im Quelltextordner des Moduls erreicht. Einige Module können mehr als eine Datei enthalten, die Erstellungssysteme definieren. Dann funktioniert die automatische Erkennung nicht mehr korrekt. In diesem Fall kann das zu verwendende Erstellungssystem manuell vorgegeben werden. Folgende Typen von Erstellungssystemen können zurzeit eingestellt werden:</p> <p>KDE</p> <p>Wird für die Erstellung von KDE-Modulen verwendet. Tatsächlich kann damit jedes Modul erstellt werden, das CMake benutzt. Aber darauf kann man sich nicht verlassen.</p> <p>Qt</p> <p>Wird für die Qt™-Bibliothek selbst verwendet.</p> <p>qmake</p> <p>Wird zur Erstellung von Qt™-Modulen verwendet, die .pro-Dateien im Stil von qmake enthalten.</p> <p>generic</p>
------------------------------	--	--

Handbuch zum Skript kdesrc-build

override-url	Moduleinstellung überschreibt die globale Einstellung	Wenn Sie diese Option einstellen, wird kdesrc-build den eingestellten Wert <i>unverändert</i> als Adresse an Subversion übergeben. Sie können diese Option grundsätzlich verwenden, um eine bestimmte Programmversion herunterzuladen. Dann kann kdesrc-build allerdings nichts mit der Option branch anfangen.
persistent-data-file	Kann nicht überschrieben werden	Benutzen Sie diese Option, um den Speicherort der ständigen Daten von kdesrc-build zu ändern. In der Voreinstellung werden diese Datei in der Datei <code>.kdesrc-build-data</code> im gleichen Ordner wie die verwendete Einrichtungsdatei gespeichert. Benutzen Sie mehrere unterschiedliche Konfigurationen im selben Ordner, dann können Sie mit dieser Option verhindern, dass die verschiedenen Konfigurationen zu Konflikten bei den ständigen Daten führen. Diese Option wurde in kdesrc-build 1.15 hinzugefügt.

<p>prefix</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option wird festgelegt, in welchen Ordner die erstellten Module installiert werden. Als Voreinstellung wird <code>kdedir</code> verwendet. Mit dieser Option können Sie Module in einen anderen Ordner als die Bibliotheken der KDE-Plattform installieren. Damit benutzen Sie <code>kdesrc-build</code> nur zum Erstellen einzelner Programme, Sie können <code>MODULE</code> or <code>\$MODULE</code> im Pfad verwenden, um sie zum Namen des Moduls zu erweitern.</p>
<p>--purge-old-logs</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Diese Einstellung bestimmt, ob alte Protokollordner automatisch gelöscht werden. Der Standardwert ist true.</p>
<p>qmake-options</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Alle hier angegebenen Optionen werden bei Modulen, die das Erstellungssystem <code>qmake</code> verwenden, an den Befehl qmake weitergegeben, Sie können zum Beispiel die Option PREFIX=/pfad/zu/qt für <code>qmake</code> benutzen, um den Installationsordner zu überschreiben. Diese Option wurde in <code>kdesrc-build</code> 1.16 hinzugefügt.</p>
<p>qtdir</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option geben Sie einen Wert für die Umgebungsvariable <code>QTDIR</code> an. Wenn Sie diese Option nicht angeben, nimmt <code>kdesrc-build</code> an, dass Qt™ vom Betriebssystem bereitgestellt wird.</p>

<p>remove-after-install</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Wenn Sie wenig Festplattenspeicher zur Verfügung haben, können Sie mit Hilfe dieser Option alle Kompilier-Ordner oder zusätzlich auch den Quelltext-Ordner nach der erfolgreichen Installation löschen lassen. Die möglichen Einstellungen für diese Option sind:</p> <ul style="list-style-type: none"> • none - Nichts wird gelöscht (dies ist die Voreinstellung). • builddir - Kompilier-Ordner werden gelöscht, aber keine Quelltext-Ordner. • all - Sowohl Kompilier- als auch Quelltext-Ordner werden gelöscht. <p>Beachten Sie: Die Verwendung dieser Option kann negative Auswirkungen auf Ihre Internet-Bandbreite und die Gesamtdauer einer KDE-Installation haben, weil kdesrc-build nicht mehr inkrementell kompilieren kann und die Pakete unter Umständen immer wieder neu heruntergeladen muss. Dies ist insbesondere der Fall, wenn Sie die Option auf <i>all</i> einstellen.</p>
<p>repository</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Diese Option wurde mit der Version 1.10 eingeführt. Damit wird das Git-Archiv angegeben, aus dem der Quelltext heruntergeladen wird. Für Qt™ und qt ist diese Option nötig, ebenso die verschiedenen KDE-Module, deren Quelltext sich mittlerweile in Git-Archiven befindet.</p>

<p>revision</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Wird diese Option auf einen Wert größer 0 gesetzt, aktualisiert kdesrc-build den Quelltext des Moduls auf genau diese Revision, auch wenn zusätzlich Optionen wie branch verwendet werden. Hat das Modul bereits diese Revision, wird es nicht weiter aktualisiert, bis diese Option geändert oder aus der Einrichtungsdatei entfernt worden ist.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>ANMERKUNG Diese Option funktionierte nicht für Git-basierte Module einschließlich der Module aus dem KDE-Projektarchiv bis kdesrc-build-Version 1.16.</p> </div>
<p>run-tests</p>	<p>Moduleinstellung überschreibt die globale Einstellung (Option für das Erstellen)</p>	<p>Ist diese Option auf true gesetzt, dann werden für das Modul auch die Tests erstellt und beim Erstellen mit ausgeführt. kdesrc-build zeigt dann einen einfachen Bericht über das Testergebnis an. Die hilft Entwicklern und anderen zu überprüfen, ob das System richtig eingerichtet ist.</p>

set-env	Moduleinstellung überschreibt die globale Einstellung	<p>Diese Option erwartet eine Liste mit durch Leerzeichen getrennten Werten. Der erste Wert ist die zu setzende Umgebungsvariable, die restlichen Werte sind der Inhalt der Variable. Wenn Sie beispielsweise die Umgebungsvariable <code>RONALD</code> auf <code>McDonald</code> setzen möchten, würden Sie diese Einstellung verwenden:</p> <pre>set-env RONALD McDonald</pre> <p>Das besondere an dieser Option ist, dass Sie sie mehrfach verwenden können, ohne dass die jeweils vorige set-env-Option im gleichen Abschnitt der Einrichtungsdatei überschrieben wird. Auf diese Weise können Sie pro Modul und auch global mehrere Umgebungsvariablen setzen.</p>
source-dir	Moduleinstellung überschreibt die globale Einstellung	<p>Diese Option legt den Ordner auf Ihrer Festplatte fest, in dem die KDE-Subversion-Quellen gespeichert werden sollen. Wenn Sie die Option nicht setzen, lautet die Voreinstellung <code>~/kdesrc</code>. Sie können für diese Option die Tilde (<code>~</code>) als Abkürzung für Ihren persönlichen Ordner benutzen.</p>
ssh-identity-file	Kann nicht überschrieben werden	<p>Mit dieser Option wird festgelegt, welche private SSH-Schlüsseldatei beim Befehl <code>ssh-add</code> verwendet wird, wenn <code>kdesrc-build</code> Quelltext aus einem Archiv herunterlädt, für das eine Authentifizierung erforderlich ist. Siehe auch Abschnitt 6.4.1. Diese Option wurde in Version 1.14.2 hinzugefügt.</p>

<p>stop-on-failure</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Setzen Sie diese Option auf true, wenn kdesrc-build beendet werden soll, sobald beim Kompilieren oder Installieren ein Fehler auftritt. Diese Einstellung ist in der Voreinstellung ausgeschaltet.</p>
<p>svn-server</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Mit dieser Option können Sie den zu verwendenden Subversion-Server festlegen. Die Voreinstellung ist der Server für anonymes Subversion <code>svn://anonsvn.kde.org/</code> .</p> <div data-bbox="997 873 1343 1087" style="border: 1px solid black; padding: 5px;"> <p>ANMERKUNG KDE-Entwickler sollten hier das Subversion-Archiv benutzen, das Ihnen mit dem Entwickler-Zugang mitgeteilt wurde, nicht das anonyme Archiv.</p> </div>
<p>tag</p>	<p>Moduleinstellung überschreibt die globale Einstellung</p>	<p>Benutzen Sie diese Option, um eine bestimmte Version eines Moduls herunterzuladen. <i>Hinweis:</i> Im Regelfall ist es sinnvoller, diese Option <i>nicht</i> zu benutzen. Fertige KDE-Versionen können üblicherweise als Tar-Archiv von einem der Softwarespiegelserver heruntergeladen werden.</p> <div data-bbox="997 1524 1343 1677" style="border: 1px solid black; padding: 5px;"> <p>ANMERKUNG Diese Option wird für Git-basierte Module seit kdesrc-build 1.16 unterstützt..</p> </div>

<p>use-clean-install</p>	<p>Moduleinstellung überschreibt die globale Einstellung (Option für das Erstellen)</p>	<p>Setzen Sie diese Option auf true, damit kdesrc-build make uninstall direkt vor dem Befehl make install ausführt. Mit dieser Option wird sichergestellt, dass es keine alten Bibliotheksdateien, CMake-Metadaten usw. mehr gibt, die in über längere Zeit aktualisierten KDE-Installationen manchmal zu Problemen führen. Dies funktioniert jedoch nur für Erstellungssysteme, die den Befehl make uninstall unterstützen. Diese Option wurde in kdesrc-build 1.12 hinzugefügt, war aber bis Version 1.16 nicht dokumentiert.</p>
<p>use-cmake</p>		<p>Diese Option wurde in kdesrc-build Version 1.4 entfernt, da alle KDE 4 Module CMake erfordern. Die Verwendung von CMake für andere Module ist nicht möglich.</p>
<p>use-idle-io-priority</p>	<p>Kann nicht überschrieben werden</p>	<p>Mit dieser Option, die es in kdesrc-build seit der Version 1.12 gibt, wird die Auslastung der Ein-/Ausgabe zum Beispiel zur Festplatte verringert. Damit kann das Verhalten des Systems auf Benutzeraktionen spürbar verbessert werden, die Laufzeit von kdesrc-build wird nur unwesentlich verlängert. In der Standardeinstellung ist diese Option deaktiviert. Setzen Sie diese Option auf true, um die Ein-/Ausgabe zur Festplatte weniger auszunutzen.</p>

use-modules

Kann nur in [Modulgruppen](#) verwendet werden

Mit dieser Option, die in kdesrc-build-Version 1.12.1 eingeführt wurde, können mehrere verschiedenen Module mit einem Eintrag in der [Einrichtungsdatei](#) angegeben werden.

Diese Option *muss* innerhalb einer Option `module-set` verwendet werden. Jeder an diese Option übergebene Name wird intern in ein Modul umgewandelt. Dabei wird als Option `repository` das Archiv der Modulgruppe in Kombination mit dem Namen benutzt, um das tatsächliche Archiv festzulegen, von dem die Quelltexte heruntergeladen werden. Alle weiteren Optionen für eine Modulgruppe werden unverändert auch für die generierten Module benutzt.

Die Reihenfolge der Module in dieser Option ist wichtig, da kdesrc-build die Module auch in dieser Reihenfolge aktualisiert, erstellt und installiert. Alle in einer Modulgruppe definierten Module werden von kdesrc-build verarbeitet, ehe das nächste Modul in der Modulgruppe bearbeitet wird.

Wenn Sie die Einstellungen für ein generiertes Modul ändern möchten, deklarieren Sie das Modul nochmal nach der Definition über die Modulgruppe. Auch wenn Sie auf diese Weise die Optionen für ein Modul verändern, wird das Modul trotzdem aktualisiert und in der Reihenfolgen erstellt, die durch die Modulgruppe vorgegeben wird. Die Reihenfolge der Erstellung kann durch diese Option also nicht geändert werden.

Handbuch zum Skript kdesrc-build

use-qt-build-dir-hack	Moduleinstellung überschreibt die globale Einstellung	Diese Option wurde entfernt, sie ist wegen des verbesserten Erstellungssystem für Qt™ nicht mehr erforderlich.
use-stable-kde	Kann nicht überschrieben werden	Diese Option ist veraltet und wird in einer der nächsten Versionen wieder entfernt. Bitte benutzen Sie stattdessen die Option branch-group , die mehr Auswahl ermöglicht.

Tabelle 4.1: Tabelle der Optionen

Kapitel 5

Befehlszeilen-Parameter und Umgebungsvariablen

5.1 Anwendung auf der Befehlszeile

kdesrc-build wurde für folgende Art der Ausführung entwickelt:

```
kdesrc-build [--options...] [zu erstellende module...]
```

Werden auf der Befehlszeile keine Module zum Erstellen angegeben, dann erstellt kdesrc-build alle in der Einrichtungsdatei definierten Module in der Reihenfolge, in der sie in der Datei aufgeführt sind. Dieses Verhalten kann aber durch verschiedene Optionen in der Einrichtungsdatei geändert werden.

5.1.1 Häufig verwendete Befehlszeilenargumente

Eine vollständige Liste der Optionen für die Befehlszeile finden Sie in Abschnitt 5.3. Die am häufigsten benutzten Optionen sind:

--pretend (oder -p)

Mit dieser Option zeigt kdesrc-build alle auszuführenden Aktionen an, ohne Sie tatsächlich durchzuführen. Damit können Sie überprüfen, ob die gewünschten Module tatsächlich erstellt werden.

--refresh-build

Mit dieser Option wird das angegebene Modul von Grund auf neu erstellt. Jeder vorhandene Erstellungsordner dieses Moduls wird gelöscht und das Modul neu erstellt. Diese Option hilft, wenn es Fehler beim Erstellen eines Moduls gibt und sie ist manchmal bei Änderungen der Qt™- oder KDE-Bibliotheken erforderlich.

--no-src

Diese Option überspringt die Aktualisierung des Quelltextes. Sie können Sie benutzen, wenn Sie erst kürzlich den Quelltext manuell oder durch die Ausführung von kdesrc-build aktualisiert haben, aber trotzdem einige Module neu erstellen möchten.

--no-build

Diese Option ähnelt der oben genannten Option `--no-src`, aber diesmal wird der Erstellungsprozess übersprungen.

5.1.2 Festlegung der Module zum Erstellen

Im allgemeinen ist es nur erforderlich, die Namen der zu erstellenden Module wie bereits in der Einrichtungsdatei anzugeben. Sie können auch Module aus einer Modulgruppe verwenden, entweder mit der Option `use-modules` oder mit dem Namen der Modulgruppe selbst, wenn Sie dafür einen Aliasnamen definiert haben.

Im besonderen Fall von Modulgruppen aus der [KDE-Projektdatei](#) kann `kdesrc-build` die Namen der Modulkomponenten erweitern, um das tatsächlich gewünschte Modul zu ermitteln. Der Eintrag `kdesrc-build` zum Beispiel ist in der KDE-Projektdatei in `extragear/utils/kdesrc-build`. Daher können Sie folgendes angeben, um `kdesrc-build` zu erstellen:

```
% kdesrc-build +extragear/utils/kdesrc-build
% kdesrc-build +utils/kdesrc-build
% kdesrc-build +kdesrc-build
```

ANMERKUNG

Bei den Befehlen im vorherigen Beispiel ist dem Modulnamen ein Symbol `+` vorangestellt. Damit wird der Modulname als Modul aus der KDE-Projektdatei interpretiert, sogar wenn das Modul nicht in Ihrer Einrichtungsdatei definiert ist.

Vorsicht bei der Angabe sehr allgemeiner Projekte wie z. B. `extragear/utils`. Das kann dazu führen, dass eine große Zahl an Modulen erstellt wird. Sie sollten die Option `--pretend` benutzen, bevor Sie eine neue Modulgruppe erstellen, um zu prüfen, ob tatsächlich nur die gewünschten Module enthalten sind.

5.2 Unterstützte Umgebungsvariablen

`kdesrc-build` verwendet keine Umgebungsvariablen. Falls Sie welche für den Kompilier- oder Installationsvorgang benötigen, lesen Sie bitte die Hinweise für die Option `set-env`.

5.3 Unterstützte Befehlszeilenparameter

Das Skript kennt die folgenden Befehlszeilenparameter:

`--async`

Aktualisiert den Quelltext und kompiliert gleichzeitig im [asynchronen Modus](#). Dies ist die Voreinstellung, daher muss diese Option nur angegeben werden, wenn sie in der Einrichtungsdatei abgeschaltet wurde.

`--help`

Zeigt eine kurze Hilfe für dieses Skript an.

`--version` (oder `-v`)

Zeigt die Programmversion an.

`--show-info`

Zeigt Informationen über `kdesrc-build` und das Betriebssystem an, die in Fehlerberichten oder bei Hilfeanfragen in Foren oder Mailinglisten nützlich sein können.

Verfügbar seit Version 18.11.

--initial-setup

kdesrc-build führt die einmalige Ersteinrichtung durch, um das System für den Betrieb von kdesrc-build und für die Ausführung der neu installierten KDE-Software vorzubereiten.

Dies schließt folgendes ein:

- Installation bekannter Abhängigkeiten unter unterstützten Linux[®]-Distributionen
- Hinzufügen erforderlicher Umgebungsvariablen zu `~/ .bashrc`
- Erstellen einer [Einrichtungsdatei](#)

Verfügbar seit Version 18.11.

--author

Zeigt Kontaktinformationen zum Autor des Skripts an.

--color

Aktiviert die farbige Darstellung der Ausgabe. Dies ist die Voreinstellung für interaktive Terminals.

--nice=wert

Dieser Wert bestimmt die Priorität der CPU des Rechners, die von kdesrc-build angefordert wird und sollte einen Wert zwischen 0 und 20 haben. 0 entspricht der höchsten Priorität und am wenigsten „nett“, 20 entspricht der niedrigsten Priorität. Als Standard benutzt kdesrc-build den Wert 10.

--no-async

Deaktiviert den [asynchronen Modus](#). So werden erst alle Module aktualisiert, ehe mit dem Erstellen begonnen wird. Insgesamt dauert damit die Ausführung länger, wenn aber IPC-Fehler bei der Ausführung von kdesrc-build auftreten, verwenden Sie diese Option und berichten den [Fehler](#).

--no-color

Deaktiviert die farbige Darstellung der Ausgaben.

--pretend (oder -p)

kdesrc-build simuliert das Aktualisieren der Quelltexte und das Erstellen, aber ohne diese Aktionen tatsächlich auszuführen. Stattdessen werden alle simulierten Aktionen, d. h. die auszuführenden Befehle und Aktionen protokolliert.

ANMERKUNG

Einfache Befehle, die nur Informationen wie zum Beispiel Informationen über Dateien einlesen, können immer noch ausgeführt werden, damit das Protokoll auch zutreffend wird. Die Ausführung von Befehlen, die nur Daten lesen, ist zum Beispiel nötig, um zu prüfen, ob der Quelltext heruntergeladen oder aktualisiert werden muss.

WICHTIG

Für diese Option müssen einige Metadaten vorhanden sein, die normalerweise automatisch heruntergeladen werden. Im Modus `--pretend` ist das Herunterladen jedoch abgeschaltet. Wenn Sie kdesrc-build noch nie ausgeführt haben und daher diese Metadaten noch nicht heruntergeladen wurden, sollten Sie **kdesrc-build --metadata-only**, damit diese Metadaten zuerst heruntergeladen werden.

--quiet (oder -q)

Reduziert die Ausgaben auf der Konsole auf grundlegende Informationen.

--really-quiet

Nur Warnungen und Fehler ausgeben.

--verbose

Es werden ausführliche Ausgaben erzeugt, Sie können den genauen Ablauf verfolgen.

--src-only (oder --svn-only)

Aktualisiert nur den Quelltext. Die Option `--svn-only` wird nur noch unterstützt, damit auch ältere Skripte noch funktionieren.

--build-only

Führt nur den Kompiliervorgang durch.

--install-only

Führt nur den Installationsvorgang durch.

--metadata-only

Lädt nur die Metadaten herunter. `kdesrc-build` führt dies normalerweise automatisch durch. Sie können dies manuell ausführen, damit die Befehlszeilenoption `--pretend` funktioniert.

--rebuild-failures

Verwenden Sie diese Option, um nur die Module zu erstellen, die bei der letzten Ausführung von `kdesrc-build` wegen Fehlern nicht erstellt wurden. Dies ist nützlich, wenn eine erhebliche Anzahl von Fehlern zusammen mit fehlerfreier Erstellung von Modulen aufgetreten ist. Nachdem Sie alle Ursachen für die Fehler bei der Erstellung beseitigt haben, können Sie mit dieser Option nur die Module neu erstellen, bei denen vorher Fehler aufgetreten sind,

ANMERKUNG

Beachten Sie, dass die Liste der „vorherigen Fehler bei der Erstellung von Modulen“ jedes mal zurückgesetzt wird, wenn die Ausführung von `kdesrc-build` mit Fehlern für einige Module beendet wird. Durch eine vollständig erfolgreiche Erstellung wird diese Liste nicht zurückgesetzt, daher können Sie ein oder zwei Module neu erstellen und diese Option funktioniert auch dann noch.

Diese Option wurde in `kdesrc-build` 15.09 hinzugefügt.

--include-dependencies, --no-include-dependencies

Mit dieser Option veranlasst werden automatisch andere KDE- und QtTM-Module einbezogen, falls dies für die Module erforderlich ist, die Sie auf der Befehlszeile oder in Ihrer [Einrichtungsdatei](#) zum Erstellen angegeben haben.

Die Module werden wie im KDE-Quellcode-Verwaltungssystem hinzugefügt. Siehe Abschnitt [2.6.4](#).

Die entsprechende Option der Einrichtungsdatei [include-dependencies](#).

Sie können auch `--no-include-dependencies` verwenden, das schaltet die automatische Einbeziehung zusätzlicher Abhängigkeitsmodule aus.

--ignore-modules

Alle auf diese Option folgenden Module auf der Befehlszeile werden nicht aktualisiert oder erstellt. Diese Option ist dann sinnvoll, wenn fast alle Module in Ihrer [Einrichtungsdatei](#) erstellt werden sollen und nur einige wenige ausgelassen werden.

--no-src (oder --no-svn)

Überspringt das Aktualisieren der Quelltexte vom Subversion-Server. Der Parameter `--no-svn` wird nur noch zur Kompatibilität mit älteren Versionen des Skripts unterstützt.

--no-build

Den Kompiliervorgang überspringen.

--no-metadata

Lädt die zusätzlichen Metadaten für die KDE-Git-Module nicht automatisch. Die Quelltexte der Module werden immer noch aktualisiert, wenn Sie nicht zusätzlich noch die Option `--no-src` verwenden.

Die ist nützlich, wenn Sie häufiger `kdesrc-build` neu ausführen, da die Metadaten sich nicht sehr häufig ändern. Beachten Sie jedoch, dass die Metadaten für viele andere Funktionen erforderlich sind. Sie sollten jedoch in Betracht ziehen, `kdesrc-build` einmal mit der Option `--metadata-only` auszuführen und dann diese Option für darauf folgende Ausführungen von `kdesrc-build` zu benutzen.

--no-install

Pakete nicht automatisch installieren, nachdem sie kompiliert wurden.

--no-build-when-unchanged, --force-build

Mit dieser Option wird das Überspringen von Modulen, eine Optimierung, die mit `build-when-unchanged` festgelegt wird, ausdrücklich verhindert. Diese Option ist dann sinnvoll, wenn das `kdesrc-build` Module erstellen soll, weil Sie Änderungen vorgenommen haben, die `kdesrc-build` nicht erkennt.

`--force-build` hat genau die gleiche Funktion, diese Option lässt sich vielleicht leichter merken.

--debug

Aktiviert den Modus zur Fehlersuche. Dies bedeutet, dass alle Ausgaben auf der Standardausgabe ausgegeben und zusätzlich im Protokoll abgelegt werden. Des weiteren sind die meisten Funktionen wesentlich „gesprächiger“.

--query=mode

Mit diesem Befehl fragt `kdesrc-build` einen Parameter der Module in der Erstellungs-Liste ab, der entweder auf der Befehlszeile angegeben oder aus der Einrichtungsdatei eingelesen wurde, und gibt das Ergebnis mit einem Modul je Zeile auf dem Bildschirm aus.

Diese Option muss mit einem der folgenden „Abfragemodi“ benutzt werden:

- `source-dir`, damit gibt `kdesrc-build` den vollständigen Pfad zum Ordner aus, in dem der Quellcode des Moduls gespeichert ist.
- `build-dir`, damit gibt `kdesrc-build` den vollständigen Pfad zum Ordner aus, in dem der Erstellungsprozess des Moduls erfolgt.
- `install-dir`, damit gibt `kdesrc-build` den vollständigen Pfad zum Ordner aus, in dem das Modul installiert wird.
- `project-path`, damit gibt `kdesrc-build` den Ort des Moduls innerhalb der Hierarchie der KDE-Quellcode-Repositories aus. Weitere Informationen über diese Hierarchie finden Sie unter Abschnitt 2.6.4.
- Mit `branch` gibt `kdesrc-build` den aufgelösten Git-Branch aus, der für jedes Modul verwendet wird, basierend auf den Einstellungen `tag-`, `branch-` und `branch-group`.
- Andernfalls können Namen von Optionen verwendet werden, die für Module in der `Einrichtungsdatei` gültig sind, deren aufgelöster Wert wird für jedes Modul aufgelistet.

Wenn ein einzelnes Modul auf der Befehlszeile übergeben wird, dann ist die Ausgabe einfach der Wert des abgefragten Parameters. Wenn mehrere (oder keine) Module in der Befehlszeile übergeben werden, wird jeder Zeile der Name des Moduls vorangestellt. In beiden Fällen wird `kdesrc-build` nicht mehr ausgeführt, sobald jeder Wert ausgegeben wurde.

Diese Option wurde in `kdesrc-build` 16.05 hinzugefügt.

Mit dem Befehl „`kdesrc-build --query branch kactivities kdepim`“ wird zum Beispiel diese Ausgabe erzeugt:

```
kactivities: master
kdepim: master
```

`--no-rebuild-on-fail`

Es wird nicht versucht, ein Modul nach einem Fehler erneut zu kompilieren. Es werden grundsätzlich keine Module neu kompiliert, bei denen es bereits einen Fehler beim Kompilieren gegeben hat.

`--refresh-build`

Aktualisiert das Erzeugungssystem und beginnt den Kompilervorgang von vorne.

`--reconfigure`

Führt die Befehle **cmake** (für KDE-Module) oder **configure** (für Qt™) nochmal aus, ohne den Erstellungsordner zu löschen. Diese Option ist normalerweise nicht erforderlich, da kdesrc-build erkennt, wenn wichtige Optionen geändert wurden, und dann das Erstellen erneut startet. Diese Option ist in `--refresh-build` automatisch enthalten.

`--resume-from`

Mit dieser Option wird das Erstellen mit dem angegebenen Modul fortgesetzt, das als nächste Option auf der Befehlszeile angegebenen ist. Sie sollten keine weiteren Modul-Namen auf der Befehlszeile übergeben.

ANMERKUNG

Mit dieser Option wurde früher auch die Option `--no-src` angewendet, dies ist aber seit kdesrc-build Version 1.13 geändert. Wenn Sie bei der Wiederaufnahme die Aktualisierung des Quelltextes vermeiden möchten, müssen Sie zusätzlich `--no-src` zu den anderen Optionen angeben.

Siehe auch `--resume-after` und Abschnitt 6.3.5.1. Diese Befehlszeilenoption sollten Sie benutzen, wenn Sie den Fehler beim Erstellen beseitigt haben und mit kdesrc-build das Erstellen fortsetzen möchten.

`--resume-after`

Mit dieser Option wird das Erstellen mit dem als nächste Option auf der Befehlszeile angegebenen Modul fortgesetzt. Sie sollten keine weiteren Modul-Namen auf der Befehlszeile übergeben.

ANMERKUNG

Mit dieser Option wurde früher auch die Option `--no-src` angewendet, dies ist aber seit kdesrc-build Version 1.13 geändert. Wenn Sie bei der Wiederaufnahme die Aktualisierung des Quelltextes vermeiden möchten, müssen Sie zusätzlich `--no-src` zu den anderen Optionen angeben.

Siehe auch `--resume-after` und Abschnitt 6.3.5.1. Diese Befehlszeilenoption sollten Sie benutzen, wenn Sie den Fehler beim Erstellen beseitigt und auch das betreffende Modul manuell erstellt und installiert haben und mit kdesrc-build das Erstellen mit dem nächsten Modul fortsetzen möchten.

`--resume`

Diese Option kann zur Ausführung von kdesrc-build nach einem Erstellungsfehler verwendet werden.

Mit dieser Option wird das Erstellen mit dem fehlerhaften Modul fortgesetzt und dabei die Liste mit den noch zu verarbeitenden Modulen vom letzten Erstellen verwendet. Die Aktualisierung der Quelltexte und Metadaten ist abgeschaltet. Diese Option kann benutzt werden, wenn ein einfacher Fehler oder eine fehlende Abhängigkeit zum Erstellungsfehler führte. Nach der Korrektur des Fehlers können Sie einfach mit dem Erstellen weitermachen, ohne die Optionen `--resume-from` und `--stop-before` anwenden zu müssen.

Die ist sogar noch praktischer als mit der Option `--stop-on-failure`, besonders wenn Sie zum ersten Mal Ihre Entwicklungsumgebung einrichten.

Diese Option wurde in kdesrc-build 1.16 hinzugefügt.

--stop-before

Diese Befehlszeilenoption wird benutzt, um den normalen Erstellungsprozess genau *vor* einem Modul anzuhalten.

Wenn zum Beispiel die normale Erstellungsliste die Einträge ModulA, ModulB, ModulC enthält, erstellt kdesrc-build mit der Option `--stop-before=ModulB` nur das ModulA.

Diese Befehlszeilenoption wurde in kdesrc-build 1.16 hinzugefügt.

--stop-after

Diese Befehlszeilenoption wird benutzt, um den normalen Erstellungsprozess genau *nach* einem Modul anzuhalten.

Wenn zum Beispiel die normale Erstellungsliste die Einträge ModulA, ModulB, ModulC enthält, erstellt kdesrc-build mit der Option `--stop-after=ModulB` die Module ModulA und ModulB.

Diese Befehlszeilenoption wurde in kdesrc-build 1.16 hinzugefügt.

--stop-on-failure

Mit dieser Option wird das Erstellen abgebrochen, sobald ein Fehler auftritt. Dies ist nützlich, wenn Sie zum ersten Mal eine Entwicklungsumgebung einrichten, Ohne diese Option werden die restlichen Module durch kdesrc-build weiter erstellt, damit keine Zeit verschwendet wird, wenn es nur einen Fehler bei einem Modul gibt.

Diese Option wurde in kdesrc-build Version 1.16 hinzugefügt, lesen Sie bitte auch die Informationen zur Option [stop-on-failure](#).

--rc-file

Der auf diese Option folgende Parameter ist die Einrichtungsdatei, aus der die Optionen eingelesen werden sollen. Die Voreinstellung für diesen Parameter ist `kdesrc-buildrc`. Diese Datei wird im aktuellen Ordner gesucht. Ist diese Datei nicht vorhanden wird die Datei `~/kdesrc-buildrc` gelesen. Siehe auch Kapitel 4.

--print-modules

Führt alle Aktionen einschließlich Neuordnung der Abhängigkeiten der auf der Befehlszeile oder in der Einrichtungsdatei angegebenen Module und gibt die Module, die verarbeitet würden, einzeln je Zeile aus. Dann wird das Skript ohne weitere Aktionen beendet.

Die Metadaten `kde-project` werden zuerst heruntergeladen, siehe allerdings auch [--pretend](#) oder [--no-src](#).

Die Ausgabe ist nicht vollständig kompatibel mit der Verwendung des Skript, da andere Ausgabenachrichten generiert werden können, bis die Modulliste angezeigt wird.

Dies ist hauptsächlich sinnvoll, um schnell festzustellen, wie kdesrc-build die Abhängigkeiten eines Moduls versteht, das heißt die Option ist nur für Module in `kde-projects` zu verwenden. Diese Option ist kompatibel mit [--resume-from](#), [--resume-after](#), [--stop-before](#), [--stop-after](#).

--list-build

Listet die Module auf, die gebaut und in welcher Reihenfolge sie gebaut werden würden. Falls zutreffend, wird in der Ausgabeliste auch erwähnt, welcher Commit/Branch/Tag gewählt würde.

Diese Option ist ähnlich wie [--print-modules](#). Genauere Informationen darüber, wie Module miteinander in Beziehung stehen, finden Sie unter [--dependency-tree](#).

--dependency-tree

Gibt Abhängigkeitsinformationen zu den Modulen in einem Baumformat (rekursiv) aus. Die aufgelisteten Informationen enthalten auch die Abhängigkeit von einem bestimmte Commit/Branch/Tag und ob die Abhängigkeit gebaut wird oder nicht. gebaut wird. Hinweis: Die generierte Ausgabe kann für Anwendungen mit vielen Abhängigkeiten ziemlich groß werden.

--run

Diese Option interpretiert den nächsten Parameter auf der Befehlszeile als Programm, das auszuführen ist. `kdesrc-build` beendet erst das Einlesen der Einrichtungsdatei, aktualisiert die Umgebungsvariablen und führt dann das Programm aus.

Damit lässt sich keine neue Shell mit den Umgebungsvariablen von `kdesrc-build` starten, da interaktive Shells normalerweise mindestens einen Teil der Umgebungsvariablen wie `PATH` und `KDEDIRS` beim Startvorgang zurücksetzen.

TIP

Möchten Sie die von `kdesrc-build` benutzten Umgebungsvariablen sehen, geben Sie den Befehl **`printenv`** ein.

```
$ kdesrc-build --run printenv
KDE_SESSION_VERSION=4
SDL_AUDIODRIVER=alsa
LANGUAGE=
XCURSOR_THEME=Oxygen_Blue
LESS=-R -M --shift 5
QMAIL_CONTROLDIR=/var/qmail/control
... etc.
```

--prefix=</pfad/zu/kde>

Ermöglicht Ihnen, den Ordner zu wählen, in den KDE installiert wird. Diese Option schließt automatisch `--reconfigure` ein, aber die Option `--refresh-build` kann immer noch erforderlich sein.

--revision

Mit dieser Option wird `kdesrc-build` eine vorgegebene Revisionsnummer für jedes Modul heruntergeladen. Damit werden alle Einstellungen der Optionen `branch`, `tag` oder `revision` für diese Module überschrieben.

Diese Option sollten Sie nicht mehr verwenden, Sie wird nur noch unterstützt, damit ältere Skripte noch funktionieren

--build-system-only

Mit dieser Option bricht `kdesrc-build` das Erstellen eines Moduls direkt vor der Ausführung des Befehls `make` ab. Diese option wird nur zur Kompatibilität mit älteren Versionen des Skripts unterstützt. Beim aktuellen Erstellungssystem für KDE ist diese Option nicht mehr nötig.

--install

Wenn dies der einzige Befehlszeilen-Parameter ist, wird versucht, alle Module zu installieren, die in der Datei `log/latest/build-status` aufgelistet sind. Wenn Sie nach `--install` weitere Parameter angeben, werden diese als Module interpretiert, die installiert werden sollen, auch wenn diese Module beim letzten Mal nicht erstellt werden konnten.

--no-snapshots

Mit dieser Option wird beim ersten Mal der Quelltext immer normal heruntergeladen anstatt ein komprimiertes Archiv zu benutzen, das nur für Git-Module aus dem `kde-projects`-Archiv zur Verfügung gestellt wird. Diese Option sollte nur verwendet werden, falls Fehler bei komprimierten Archiven auftreten, um die KDE-Quelltextarchive nicht zu überlasten.

ANMERKUNG

Komprimierte Archive sind *tatsächlich* Subversion-Archive. Diese Option ist normalerweise nicht erforderlich, sie sollte nur bei Problemen mit dem Herunterladen benutzt werden.

`--delete-my-patches`

Diese Option wird von kdesrc-build dazu verwendet, Quelltextordner mit Benutzerdaten zu löschen, damit ein Modul wieder heruntergeladen werden kann. Dies ist normalerweise nur für KDE-Entwickler sinnvoll, die möglicherweise lokale Änderungen haben, die beim Herunterladen gelöscht würden.

Diese Option wird zurzeit nur für das Herunterladen von Modulen benutzt, die von Subversion nach Git konvertiert wurden. Normalerweise sollten Sie diese Option nicht verwenden, da kdesrc-build nachfragt, ob dieser Vorgang wiederholt werden soll, falls nötig.

`--delete-my-settings`

Mit dieser Option überschreibt kdesrc-build vorhandene Dateien, die Daten von Benutzern enthalten können.

Diese Option wird zurzeit nur für die Einrichtung von „xsession“ für die Anmeldung benutzt. Normalerweise sollten Sie diese Option nicht verwenden, da kdesrc-build nachfragt, ob dieser Vorgang wiederholt werden soll, falls nötig.

`--<option-name>=`

Dies können Sie verwenden, um eine Option in Ihrer [Einrichtungsdatei](#) für einzelne Module zu überschreiben. Beispiel: `--log-dir= /pfad/zu/ordner` würde die Option `log-dir` überschreiben.

ANMERKUNG

Diese Funktion kann nur für Namen von Optionen benutzt werden, die bereits vorher durch kdesrc-build erkannt wurden und die noch nicht auf der Befehlszeile unterstützt werden. Die Option `async` in der Einrichtungsdatei zum Beispiel hat spezifische Befehlszeilenoption `--async` und `--no-async`, die von kdesrc-build bevorzugt verwendet werden.

`--set-module-option-value=<module-name>, <option-name>, <option-value>`

Mit dieser Option können Einstellungen aus der [Einrichtungsdatei](#) für ein einzelnes Modul überschrieben werden.

Alle weiteren Parameter werden als Modul-Namen interpretiert, die aktualisiert oder kompiliert werden sollen. Verwechseln Sie Kompilieren nicht mit Installieren.

Kapitel 6

kdesrc-build verwenden

6.1 Einleitung

Nachdem Sie das Kapitel 2 gelesen haben, können Sie kdesrc-build im Terminal mit dieser Eingabe starten:

```
% kdesrc-build
```

kdesrc-build lädt dann die Quelltexte für KDE herunter und konfiguriert, kompiliert und installiert das KDE-System.

In den folgenden Kapiteln erfahren Sie, wie kdesrc-build diese Aufgabe ausführt und welche weiteren Hilfen geboten werden.

6.2 Grundfunktionen von kdesrc-build

6.2.1 Unterstützung für qt

Als Hilfe für die Benutzer unterstützt kdesrc-build das Erstellen der Qt™-Bibliothek, die von den KDE-Programmen benutzt wird. Dabei handelt es sich um ein besonderes Modul mit dem Namen qt.

ANMERKUNG

Qt™ wird in einen eigenen Archiv auf <http://code.qt.io/cgit/qt/> unabhängig von der KDE-Software entwickelt.

Um Qt™ zu erstellen, müssen Sie den Pfad `qtdir` für den Ordner einstellen, in den Sie Qt™ installieren möchten. Dies wird im Abschnitt [Abschnitt 2.2](#) beschrieben.

Stellen Sie sicher dass das Modul qt in Ihrer Datei `.kdesrc-buildrc` enthalten ist und vor allen anderen Modulen steht. Wenn Sie die mitgelieferte Beispieldatei benutzen, müssen Sie nur das Kommentarzeichen vor dem Eintrag für das Modul qt entfernen.

Nun sollten Sie überprüfen, ob die Optionen `repository` und `branch` richtig eingestellt sind.

1. Mit der ersten Option wird Qt™ von einem Spiegelserver erstellt, der im KDE-Quelltextarchiv vorgehalten wird, aber gegenüber dem Original unverändert ist. Dies wird dringend empfohlen, da es manchmal Probleme beim Klonen des vollständigen Qt™-Moduls vom offiziellen Archiv gibt.

Setzen Sie die Einstellung `repository` für das Modul Qt™ auf `kde:qt`, um dies zu benutzen.

2. Um anderenfalls die Standardversion von Qt™ zu erstellen, geben Sie für das Archiv die Option `repository` `git://gitorious.org/qt/qt.git` an. Beachten Sie, dass es möglicherweise Probleme beim ersten Klonen von Qt™ aus diesem Archiv gibt.

In beiden Fällen sollten Sie die Option „branch“ auf `master` setzen, es sei denn, Sie möchten eine anderen Zweig (branch) erstellen.

6.2.2 Standard-Parameter von kdesrc-build

Dieser Abschnitt gilt nicht für Module, für die Sie eigene Entwicklungs-Werkzeuge eingestellt haben, z.B. mit `cmake-toolchain`.

kdesrc-build fügt einige Standardpfade zur der Entwicklungsumgebung hinzu:

- Der Pfad zu den KDE- und Qt™-Bibliotheken wird automatisch zur Variablen `LD_LIBRARY_PATH` hinzugefügt. Es ist nicht erforderlich, `libpath` zu bearbeiten und den Pfad dort einzutragen.
- Der Pfad zu den KDE- und Qt™-Hilfsprogrammen für die Entwicklung wird automatisch zur Variablen `PATH` hinzugefügt. Es ist nicht erforderlich, `binpath` zu bearbeiten und den Pfad dort einzutragen.
- Der Pfad zum Programm `pkg-config` von KDE wird automatisch zur Variablen `PKG_CONFIG_PATH` hinzugefügt. Es ist nicht erforderlich, `set-env` zu bearbeiten und den Pfad dort einzutragen.
- Die Einstellung für `kdedir` wird automatisch beim Erstellen zur Umgebungsvariablen `KDEDIR` hinzugefügt. Die Variable `KDEDIRS` wird nicht verändert.
- Die Einstellung für `qtdir` wird automatisch beim Erstellen zur Umgebungsvariablen `QTDIR` hinzugefügt.

6.2.3 Ändern der Priorität von kdesrc-build beim Erstellen

Programme können auf Betriebssystemen wie Linux® und BSD mit unterschiedlichen Prioritäten ausgeführt werden. Damit werden den verschiedenen Programmen je nach Wichtigkeit vom System unterschiedliche Anteile an der Prozessorlaufzeit zugewiesen.

kdesrc-build wird normalerweise mit einer niedrigen Priorität ausgeführt, sodass die Ausführung der anderen laufenden Programme auf dem System nicht behindert wird. Mit dieser Technik verwendet kdesrc-build CPU-Zeit, wenn sie verfügbar ist.

kdesrc-build wird aber noch mit einer genügend hohen Priorität gestartet, dass es vorrangig vor normalen Wartungsprogrammen oder Programmen wie `Seti@Home` ausgeführt wird.

Um kdesrc-build immer mit niedrigerer oder höherer Priorität auszuführen, muss die Einstellung für `niceness` in der `Einrichtungsdatei` geändert werden. Dabei bedeutet ein höherer Wert für `niceness` eine niedrigere Priorität und umgekehrt. Der Wert für `niceness` kann zwischen 0 - höchste Priorität - und 20 - niedrigste Priorität liegen.

Mit der `Befehlszeilenoption` `--nice` kann kdesrc-build vorübergehend mit geänderter Priorität ausgeführt werden. Für diesen Befehl werden die gleichen Werte wie für `niceness` verwendet.

ANMERKUNG

Programme können vom Systemverwalter sogar mit einem negativen Nice-Wert und damit sehr hoher Priorität ausgeführt werden. kdesrc-build mit einer `niceness` von 0 laufen zu lassen, spart nicht merklich Zeit, kann aber dazu führen, dass andere Programme auf dem Rechner nur noch träge reagieren.

Um kdesrc-build mit einem Nice-Wert von 15 (kleiner als normal) auszuführen. geben Sie folgendes ein:

```
% kdesrc-build --nice=15
```

Alternativ ändern Sie den Wert dauerhaft in der [Einrichtungsdatei](#):

```
niceness 15
```

TIP

Die Option [niceness](#) beeinflusst nur die Auslastung des Prozessors Ihres Rechners. Ein weiterer wichtiger Einfluß der Leistungsfähigkeit eines Rechners ist die Auslastung der Dateneingabe oder -Ausgabe (I/O) durch ein Programm. Um festzulegen, wieviel dieser Ein-/Ausgabe (I/O) ein Programm benutzen darf, kann in modernen Linux[®]-Systemen das Werkzeug [ionice](#) benutzt werden. [kdesrc-build](#) unterstützt [ionice](#) mit der Option [use-idle-io-priority](#) seit der [kdesrc-build](#)-Version 1.12. [ionice](#) kann aber nur ein- oder ausgeschaltet werden.

6.2.4 Installation als Systemverwalter

[kdesrc-build](#) kann auch mit der Berechtigung des Systemverwalters ausgeführt werden, zum Beispiel für eine nicht empfohlene Installation in den Systemordnern. Es gibt aber einige Module wie besonders [kdebase](#), die Programme installieren, die mit erweiterten Berechtigungen ausgeführt werden müssen. Dazu müssen diese Programme aber auch mit erweiterten Berechtigungen installiert werden.

[kdesrc-build](#) könnte einfach mit der Berechtigung des Systemverwalters ausgeführt werden, aber davon wird abgeraten, da das Programm für diesen Anwendungsfall nicht getestet wurde. Wenn immer möglich, sollte ein Programm mit begrenzten Berechtigungen ausgeführt werden.

Dazu gibt es in [kdesrc-build](#) die Option [make-install-prefix](#). Mit dieser Option wird ein Befehl angegeben, der die Installation mit der Berechtigung eines anderen Benutzers ausführt. Es wird empfohlen, für diese Option den Befehl [Sudo](#) zu benutzen, damit kann die Installation als Systemverwalter ausgeführt werden.

Um zum Beispiel alle Module mit [Sudo](#) zu installieren, geben Sie folgendes ein:

```
global
  make-install-prefix sudo
  # Weitere Optionen
end global
```

Um [make-install-prefix](#) nur für ein einzelnes Modul zu benutzen, geben Sie folgendes ein:

```
module svn-module-name
  make-install-prefix sudo
end module
```

6.2.5 Fortschrittsanzeige beim Erstellen eines Moduls anzeigen

Diese Funktion wird immer automatisch aktiviert. Sie zeigt den geschätzten Fortschritt für das Erstellen eines Moduls an. Damit ist erkennbar, wie lange das Erstellen eines Moduls noch dauert.

6.3 Weitergehende Funktionen

6.3.1 Teile eines Moduls erstellen

Es ist möglich, nur Teile eines einzelnen KDE-Moduls zu erstellen, zum Beispiel nur ein Programm aus einem Modul. Das ist mit `kdesrc-build` einfach durchzuführen, es gibt dafür mehrere Möglichkeiten.

6.3.1.1 Teile eines Moduls herunterladen

Mit dieser Methode wird Zeit beim Herunterladen und Speicherplatz auf der Festplatte eingespart. `kdesrc-build` lädt nur die angegebenen Teile eines Moduls herunter. Dazu wird die Option `checkout-only` mit einer Liste der herunterzuladenden Ordner für ein Modul verwendet.

TIP

Wenn Sie nicht genau wissen, welche Programme ein Modul enthält, sollten Sie zuerst die Ordner im Modul mit [WebSVN](#) ansehen.

Um nur `KUser` und `KSystemLog` aus dem Modul `kdeadmin` herunterzuladen, benutzen Sie `checkout-only` wie im folgenden Beispiel:

```
module kdeadmin
  checkout-only kuser ksystemlog
end module
```

WICHTIG

Die Ordner werden in der gleichen Reihenfolge wie in der Liste kompiliert. Wenn für das Kompilieren eines Ordners vorher andere Teile des Moduls kompiliert werden müssen, sind alle erforderlichen Ordner zusammen in die Zeile `checkout-only` einzutragen, und zwar in der richtigen Reihenfolge entsprechend den Abhängigkeiten.

Manchmal benötigt ein Programm andere Ordner, aber es ist nicht direkt erkennbar welche erforderlich sind. In diesem Fall ist es am einfachsten, einen Ordner nach dem anderen hinzuzufügen bis das Erstellen gelingt. Diese Option funktioniert nur, wenn das Erstellungssystem des Moduls dies unterstützt. Daher ist die Option nur für Module sinnvoll, die aus einer Sammlung von einzelnen Programmen bestehen.

Ein letzter Hinweis zu dieser Option: Wenn der Wert der Option geändert wird, sollte mit `kdesrc-build --refresh-build module` sichergestellt werden, dass das Modul korrekt neu konfiguriert wird. `kdesrc-build` entfernt keine vorhandenen Dateien, wenn einige Ordner aus der Option `checkout-only` gelöscht oder diese Option für ein bereits heruntergeladenes Modul eingefügt wird.

6.3.1.2 Ordner beim Erstellen auslassen

Anstatt die Menge der herunterzuladenden Quelltexte zu beschränken, kann alles heruntergeladen werden und einige wenige Ordner werden beim Kompilieren ausgelassen. Diese Vorgehensweise kann zum Beispiel benutzt werden, wenn bei einem Ordner immer Fehler auftreten, dieser Ordner aber für die übrigen Programme im Modul nicht erforderlich ist.

Dazu wird die Option `do-not-compile` verwendet, Sie funktioniert ähnlich wie die gerade beschriebene Option `checkout-only` und enthält ebenfalls eine Liste der Ordner, die nicht kompiliert werden sollen.

WICHTIG

Wie [checkout-only](#) erfordert auch diese Option, dass das Erstellungssystem nach den Änderungen für das Modul erneut ausgeführt wird. Dies erreichen Sie mit dem Befehl **kdesrc-build --reconfigure module**.

Um den Ordner `python` beim Erstellen des Moduls `kdebindings` auszulassen, geben Sie folgendes ein:

```
module kdebindings
  do-not-compile python
end module
```

ANMERKUNG

Diese Option hängt von einigen Standardkonventionen ab, die in den meisten KDE-Modulen benutzt werden. Daher kann diese Option für einige Programme nicht funktionieren.

6.3.2 Zweige (Branches) und Tags mit kdesrc-build verwenden

6.3.2.1 Was sind eigentlich Zweige (Branches) und Tags?

Subversion verwaltet den Verlauf der Änderungen am KDE-Quelltext. Im KDE-System werden mit dieser Funktion Zweige für die Entwicklung und Tags des Archivs für eine Veröffentlichung erstellt.

Die KMail-Entwickler können zum Beispiel an neuen Funktionen in einem eigenen Zweig arbeiten, ohne Fehler in die von den meisten Entwicklern benutzten Hauptversion einzubringen. Im getrennten Zweig kann die Software unabhängig von der ebenfalls weiterentwickelten Hauptversion /trunk verbessert werden.

Ein Tag ist ein Abbild des Quelltextarchivs zu einem bestimmten Zeitpunkt. Die KDE-Systemverwalter verwenden Tags, um eine neue Version des Quelltextes für eine Veröffentlichung zu kennzeichnen. Die Softwareentwicklung kann dann in der Hauptversion /trunk weitergehen.

In Subversion gibt es keinen Unterschied zwischen Zweigen, Tags oder Trunk im Quelltext. Diese Bezeichnungen sind nur eine Konvention der Entwickler. Damit ist es schwierig, Zweige und Tags in `kdesrc-build` richtig zu verarbeiten.

6.3.2.2 Zweige (Branches) und Tags verwenden

Zweige und Tags werden mit einigen Optionen angegeben, von einer generellen Angabe für eine Version bis zu bestimmten URLs zum Herunterladen für erfahrene Benutzer.

Am einfachsten ist es, die Optionen `branch` und `tag` zu verwenden. Dazu wird die Option mit dem Namen des gewünschten Zweigs oder Tags für ein Modul angegeben. `kdesrc-build` ermittelt dann den passenden Pfad im KDE-Archiv. Für die meisten KDE-Module funktioniert das ohne Probleme.

Mit diesem Eintrag wird das Modul `kdelibs` aus KDE 4.6 - 4.6-Zweig genannt - heruntergeladen:

```
module kdelibs
  branch 4.6
  # weitere Optionen...
end module
```

Mit diesem Eintrag wird das Modul `kdemultimedia` aus KDE 4.6.1 heruntergeladen:

```
module kdemultimedia
  tag 4.6.1
  # weitere Optionen...
end module
```

TIP

Für die Option `branch` kann ein globaler Wert eingestellt werden. Denken Sie aber daran, für alle Module einen zutreffenden Zweig (`branch`) einzustellen, die diesen globalen Wert nicht verwenden sollen.

6.3.2.3 Erweiterte Optionen für Zweige (`branch`)

kdesrc-build bietet zwei Optionen für Situationen, in denen mit `branch` und `tag` der falsche Pfad benutzt wird: `module-base-path` und `override-url`.

- Mit `module-base-path` kann kdesrc-build den fehlenden Teil für den Pfad eines Moduls einfügen. Im KDE-Archiv haben alle Pfade die Form `svnRoot/module-base-path/ module-name`. Normalerweise ist kdesrc-build dazu in der Lage, den passenden mittleren Teil des Pfades selbst zu bestimmen. Funktioniert dies nicht, ist `module-base-path` wie im folgenden Beispiel zu verwenden:

```
module kdesupport
  # kdesupport supports various tags to easily organize the required
  # software for a given KDE Platform release.
  module-base-path tags/kdesupport-for-4.5
end module
```

Mit dieser Option lädt kdesrc-build in diesem Beispiel kdesupport von `svn://anonsvn.kde.org/home/kde/ tags/kdesupport-for-4.5` herunter.

TIP

In früheren Versionen von kdesrc-build wurde die Option `module-base-path` anders verwendet. Stellen Sie Probleme mit einer alten Definition für diese Option fest, überprüfen Sie ob der tatsächliche Pfad in der von kdesrc-build erwarteten Form vorliegt, indem Sie die Option `--pretend` benutzen.

- Die Option `override-url` erfordert die genaue Angabe des Pfades zum Modul im Archiv. Damit wird es kdesrc-build ermöglicht, Module zu laden, die mit den anderen Optionen nicht eingegeben werden können. Zurzeit sollte die Option `module-base-path` für alle Subversion-Archive funktionieren.

WICHTIG

kdesrc-build ändert oder bearbeitet den angegebenen Wert für die Option `override-url` nicht. Wenn die Einstellung `svn-server` geändert wird, müssen auch die Pfade in `override-url` entsprechend angepasst werden.

6.3.3 Arbeitsweise von kdesrc-build zum erfolgreichen Erstellen

6.3.3.1 Automatisch das Erstellen neu ausführen

kdesrc-build enthielt in früheren Versionen Funktionen, um ein Module nach einem Fehler automatisch wieder neu zu erstellen. Dieses Verhalten war häufig wegen Fehlern im alten Erstellungssystem erfolgreich. Mit dem Wechsel zu CMake sind diese Fehler beseitigt, daher versucht

kdesrc-build nur noch einmal, ein Modul zu erstellen. Trotzdem gibt es noch Situationen, in denen kdesrc-build automatisch aus diesen Gründen ein Modul erneut erstellt:

- Wenn Sie für ein Modul [configure-flags](#) oder [cmake-options](#) geändert haben, erkennt kdesrc-build dies und führt automatisch erneut „configure“ oder „cmake“ für dieses Modul aus.
- Wenn der Erstellungsordner nicht vorhanden ist - auch wenn er nicht von kdesrc-build entfernt wurde - erstellt das Skript ihn automatisch neu. Damit ist es möglich, einen vollständigen [--refresh-build](#) nur für ein bestimmtes Modul und nicht für alle anderen Module auszuführen.

6.3.3.2 Module manuell neu erstellen

Wenn die Optionen oder der Quelltext eines Moduls in einer Weise geändert wurden, die von kdesrc-build nicht erkannt wird, muss ein Modul manuell neu erstellt werden.

Dazu wird der Befehl **kdesrc-build --refresh-build module** ausgeführt.

Wenn dies erst bei der nächsten normalen Ausführung von kdesrc-build automatisch erfolgen soll, kann eine Datei `.refresh-me` in den Erstellungsordner eines Moduls geschrieben werden. So wird kdesrc-build bei der nächsten Ausführung das Modul neu erstellen und nicht wie normalerweise nur inkrementell.

TIP

In der Voreinstellung ist der Erstellungsordner `~/kdesrc/build/ module /`. Haben Sie die Einstellung für die Option [build-dir](#) geändert, verwenden Sie diese Einstellung anstelle von `~/kdesrc/build`.

Neu erstellen mit `.refresh-me` für das Modul `kdelibs`:

```
% touch ~/kdesrc/build/kdelibs/.refresh-me
% kdesrc-build
```

6.3.4 Einstellungen für Umgebungsvariablen ändern

Normalerweise benutzt kdesrc-build die Umgebungsvariablen, die beim Start des Skripts gesetzt sind, um den Quelltext zu aktualisieren und zu kompilieren. Das funktioniert ohne Probleme, wenn Sie kdesrc-build von der Befehlszeile ausführen.

Manchmal möchten Sie die Umgebungsvariablen ändern, für die keine Optionen in kdesrc-build vorhanden sind. Wenn Sie zum Beispiel kdesrc-build als Cron-Job regelmäßig ausführen möchten, können Sie mit der Option [set-env](#) alle dafür nötigen Umgebungsvariablen setzen.

Anders als alle anderen Optionen kann diese Option mehrfach verwendet werden. Sie erfordert zwei durch Leerzeichen getrennte Einträge. Der erste Eintrag ist der Name der Umgebungsvariablen und der zweite der einzustellende Wert.

Setzt **DISTRO=BSD** für alle Module:

```
global
  set-env DISTRO BSD
end global
```

6.3.5 Das Erstellen fortsetzen

6.3.5.1 Ein abgebrochenes oder fehlgeschlagenes Erstellen fortsetzen

kdesrc-build kann das Erstellen mit einem bestimmten Modul fortsetzen. Diese Funktion kann mit den Optionen [--resume-from](#) und [--resume-after](#) benutzt werden, wenn bei einigen Modulen Fehler aufgetreten sind, oder das Kompilieren abgebrochen wurde.

ANMERKUNG

Frühere Versionen von kdesrc-build haben die Aktualisierung des Quelltextes beim erneuten Erstellen übersprungen. Dieses Verhalten ist nicht mehr die Voreinstellung, aber Sie können immer noch die Befehlszeilenoption `--no-src` verwenden, um das Aktualisieren des Quelltextes zu überspringen.

Das Erstellen mit kdebase fortsetzen:

```
% kdesrc-build --resume-from=kdebase
```

Das Erstellen mit dem Modul nach kdebase fortsetzen (für den Fall, dass Sie den Fehler in kdebase manuell beseitigt und dieses Modul selbst installiert haben):

```
% kdesrc-build --resume-after=kdebase
```

Wenn das letzte Erstellen mit kdesrc-build mit einem Fehler beendet wurde, können Sie auch die Befehlszeilenoption `--resume` verwenden, mit der das Erstellen beim letzten fehlgeschlagenen Modul fortgesetzt wird. Das Aktualisieren der Quelltexte und Metadaten wird gleichfalls übersprungen. Möchten Sie dies jedoch ausführen, sollten Sie stattdessen die Option `--resume-from` verwenden.

6.3.5.2 Module vom Erstellen ausschließen

Ähnlich wie das Erstellen eines Moduls `fortgesetzt` wird, kann stattdessen das Aktualisieren und Erstellen für alles ausgeführt werden, wobei einige Module ausgelassen werden.

Verwenden Sie dazu die Option `--ignore-modules`. Damit werden alle mit dieser Option angegebenen Module nicht aktualisiert oder erstellt.

Mit dieser Eingabe werden extragear/multimedia und kdereview beim Erstellen ausgelassen:

```
% kdesrc-build --ignore-modules extragear/multimedia kdereview
```

6.3.6 Optionen auf der Befehlszeile ändern

6.3.6.1 Globale Optionen ändern

Die Einstellungen aus der `Einrichtungsdatei` können auf der Befehlszeile geändert werden. Damit werden die globalen Einstellungen nur für die Ausführung dieses Befehls überschrieben.

In kdesrc-build können interne Optionen mit Namen wie `option-name` durch Eingabe eines Arguments auf der Befehlszeile in der Form `--option-name=value` geändert werden. kdesrc-build sucht den angegebenen Namen in der internen Liste der Optionen. Wird der Name nicht gefunden, so wird eine Warnung ausgegeben, anderenfalls wird der angegebene Wert der Option übernommen und der Wert aus der Einrichtungsdatei überschrieben.

Einstellung der Option `source-dir` auf `/dev/null` zum Testen:

```
% kdesrc-build --pretend --source-dir=/dev/null
```

6.3.6.2 Änderung der Optionen für Module

Sie können Optionen nur für bestimmte Module ändern. Die Syntax ist ähnlich: `--module,option-name=value`.

Diese Änderung überschreibt jede Einstellung für das Modul aus der `Einrichtungsdatei` und wird nur bei Eingabe auf der Befehlszeile benutzt.

Mit dieser Eingabe wird ein anderer Ordner zum Erstellen des Moduls kdedu verwendet:

```
% kdesrc-build --kdedu,build-dir=temp-build
```

6.4 Funktionen für KDE-Entwickler

6.4.1 Überprüfung des SSH-Agenten

kdesrc-build kann überprüfen, ob KDE-Entwickler mit SSH-Zugriff auf das KDE-Quelltextarchiv nicht aus Versehen den Dienst SSH-Agent deaktiviert haben. In diesem Fall würde kdesrc-build auf die Eingabe des SSH-Passwort des Entwicklers warten und die Ausführung anhalten. Daher wird als Standard vor dem Aktualisieren der Quelltexte überprüft, ob der Dienst SSH-Agent läuft.

ANMERKUNG

Dies ist nur für KDE-Entwickler mit SSH-Zugriff auf das Archiv erforderlich, über anonymes SVN kann der Quelltext ohne Passwort heruntergeladen werden.

Diese Prüfung kann für die Fälle abgeschaltet werden, in denen kdesrc-build irrtümlicherweise diesen Dienst falsch erkennt. Um die Überprüfung des SSH-Agenten abzuschalten, setzen Sie die Option `disable-agent-check` auf **true**.

Überprüfung des SSH-Agenten abschalten:

```
global
  disable-agent-check true
end global
```

6.5 Weitere Funktionen von kdesrc-build

6.5.1 Ausgabe des Skripts kdesrc-build einstellen

kdesrc-build hat mehrere Optionen, um den Umfang der vom Skript generierten Ausgaben einzustellen. Fehler werden in jedem Fall angezeigt.

- Mit der Option `--quiet` (Kurzform `-q`) werden nur wichtige Meldungen, Warnungen und Fehler angezeigt. Wenn verfügbar, wird auch der Fortschritt beim Erstellen ausgegeben.
- Mit der Option `--really-quiet` (keine Kurzform) werden nur wichtige Warnungen oder Fehler angezeigt.
- Mit der Option `--verbose` (Kurzform `-v`) wird eine sehr detaillierte Ausgabe eingestellt.
- Die Option `--debug` ist nur für die Fehlersuche erforderlich, damit verhält sich kdesrc-build als wenn die Option `--verbose` aktiviert ist. Zusätzlich werden alle Meldungen auf der Konsole und viele Informationen über die ausgeführten Funktionen ausgegeben.

6.5.2 Farbige Ausgaben

Bei der Ausführung von kdesrc-build in der Konsole oder einem anderen Terminal wird die Ausgabe normalerweise als farbiger Text dargestellt.

Dies kann mit der Option `--no-color` auf der Befehlszeile ausgeschaltet werden oder indem in der [Einrichtungsdatei](#) die Option `colorful-output` gleich **false** gesetzt wird.

Farbige Ausgabe in der Einrichtungsdatei abschalten:

```
global
  colorful-output false
end global
```

6.5.3 Ordner nach dem Erstellen löschen

Sie haben wenig Festplattenspeicherplatz, möchten aber trotzdem die Entwicklerversion von KDE herunterladen? kdesrc-build kann dabei helfen, Speicher auf der Festplatte beim Erstellen von KDE aus dem Subversion-Archiv einzusparen.

ANMERKUNG

Beachten Sie, dass das Erstellen von KDE viel Speicherplatz benötigt. Dafür gibt es mehrere Ursachen:

1. Der aktuelle Quelltext beansprucht viel Speicherplatz, alle Standardmodule zusammen etwa 1,6 Gigabyte. Diesen Wert können Sie dadurch verringern, dass nur die tatsächlich erforderlichen Module heruntergeladen werden. kdesrc-build löscht keinen Quelltext von der Festplatte, auch wenn ein Modul aus der [Einrichtungsdatei](#) entfernt wird. Unbenutzter Quelltext muss manuell gelöscht werden. Es sollten aber keine Quelltexte gelöscht werden, wenn kdesrc-build noch läuft.

Wenn Sie Qt™ bereits über Ihre Distribution installiert haben, können Sie vielleicht auf das Modul qt verzichten und damit etwa 200 MB Quelltext einsparen.

Ein Hinweis zur Arbeitsweise von Subversion: Es gibt immer zwei Dateien auf der Festplatte für jede aus dem Archiv heruntergeladene Datei. Mit kdesrc-build ist es zurzeit nicht möglich, den von nicht verwendetem Quelltext belegten Speicherplatz frei zu machen.

2. kdesrc-build erstellt einen eigenen Ordner zum Erstellen der Programme aus den Quelltexten. Manchmal muss kdesrc-build einen Quelltextordner kopieren, um einen Erstellungsordner vorzutauschen. In diesem Fall werden symbolische Verknüpfungen verwendet. Der Erstellungsordner ist typischerweise erheblich größer als der Quelltextordner. Für kdatabase werden im Erstellungsordner etwa 1050 Megabyte benötigt, für den Quelltext jedoch nur 550 Megabyte.

Der Ordner zum Kompilieren der Software wird aber nicht mehr benötigt, wenn das Modul erfolgreich erstellt und installiert worden ist. kdesrc-build kann automatisch den Erstellungsordner löschen, nachdem das Modul installiert wurde, ein Beispiel dazu wird weiter unten gezeigt. Damit ist es aber mit kdesrc-build nicht mehr möglich, nur die Änderungen in einem Modul zu kompilieren und damit Zeit einzusparen.

3. Außerdem ist noch Speicherplatz auf der Festplatte für den Installationsordner von KDE erforderlich. Dieser Ordner verbraucht weniger Platz als der Erstellungsordner, genaue Angaben sind aber nicht möglich.

Wie kann der erforderliche Speicherplatz für KDE verringert werden? Eine Möglichkeit ist die Verwendung der richtigen Compiler-Schalter - Optimierung des Speicherbedarfs anstelle der Ausführungsgeschwindigkeit. Die andere Möglichkeit ist das Entfernen der Debugging-Informationen aus dem kompilierten KDE, das spart viel Speicherplatz.

WARNUNG

Bedenken Sie vor dem Entfernen der Debugging-Informationen: Entwicklersoftware kann viel häufiger abstürzen als eine stabile Version. Ohne Debugging-Informationen ist es kaum möglich, Fehlerberichte zu schreiben, die alle zur Behebung benötigten Informationen enthalten. Dazu müssen Sie unter Umständen ein Modul mit Debugging-Informationen neu kompilieren, um den Entwicklern ausreichende Informationen zur Behebung des Fehlers zu liefern. Bedenken Sie dies, wenn Sie Debugging-Informationen entfernen wollen.

Löschen des Erstellungsordners nach der Installation eines Moduls. Der Quelltextordner bleibt erhalten und das Debuggen ist eingeschaltet:

Handbuch zum Skript kdesrc-build

```
global
  configure-flags      --enable-debug
  remove-after-install builddir      # Build-Ordner nach der Installation ↔
  löschen
end global
```

Löschen des Erstellungsordners nach der Installation, ohne Debugging-Informationen, mit Optimierung für die Dateigröße:

```
global
  cxxflags             -Os             # Optimimierung für Dateigröße
  configure-flags      --disable-debug
  remove-after-install builddir      # Build-Ordner nach der Installation ↔
  löschen
end global
```


Kapitel 7

CMake, das System zum Erstellen von KDE

7.1 Einführung zu CMake

Im März 2006 wurde CMake als neues System zum Erstellen von KDE 4 ausgewählt. Damit wurde das autotools-basierte System abgelöst, das in KDE von Anfang an verwendet wurde.

Eine Einführung zu CMake finden Sie im [KDE-Community-Wiki](#). Anstatt `make -f Makefile .cvs, configure` und Make wird nur noch CMake und dann Make eingegeben.

Einige Funktionen von kdesrc-build waren tatsächlich Funktionen des Erstellungssystems, wie [configure-flags](#) und [do-not-compile](#). Alle diese Funktionen sind auch weiterhin verfügbar, wenn sie durch CMake unterstützt werden. Als Ersatz für die Option „configure-flags“ gibt es in CMake [cmake-options](#) und die Option [do-not-compile](#) wird auch seit kdesrc-build Version 1.6.3 unterstützt.

Kapitel 8

Danksagungen und Lizenz

Übersetzung: Thomas Reitelbach tr@erdfunkstelle.de

Diese Dokumentation ist unter den Bedingungen der [GNU Free Documentation License](#) veröffentlicht.

Anhang A

KDE-Module und Organisation des Quelltexts

A.1 Die „Module“

KDE-Software ist in „Module“ unterschiedlicher Größe zusammengefasst. Ursprünglich war das nur eine lockere Gruppe weniger großer Module, aber mit der Einführung [Git-basierter Quelltextarchive](#) wurden diese großen Module in viele kleinere Module aufgeteilt.

kdesrc-build benutzt gleichfalls dies Modulkonzept. Im Wesentlichen ist ein „module“ eine Zusammenfassung von Quelltexten, die heruntergeladen, erstellt, getestet und installiert werden kann.

A.1.1 Einzelne Module

Es ist einfach möglich, mit kdesrc-build nur ein einzelnes Modul zu erstellen. Die folgenden Befehle sind ein Beispiel dafür, wie ein einzelnes Modul aus einem Subversion-Versionsverwaltungssystem in der [Einrichtungsdatei](#) einzutragen ist.

```
module kdefoo
  cmake-options -DCMAKE_BUILD_TYPE=Debug
end module
```

TIP

Da die Option [repository](#) nicht verwendet wird, handelt es sich um ein Modul aus einem Subversion-Versionsverwaltungssystem. Die Option `cmake-options` ist nur als Beispiel aufgeführt, aber nicht erforderlich.

A.1.2 Gruppen zusammengehöriger Module

Zurzeit sind die meisten KDE-Quelltextmodule in Git-Versionsverwaltungssystemen. Sie sind normalerweise in Super-Modulen zusammengefasst.

Daher unterstützt kdesrc-build ebenfalls Super-Module mit [Modulgruppen](#). Ein Beispiel:

```
module-set base-modules
  repository kde-projects
  use-modules kde-runtime kde-workspace kde-baseapps
end module-set
```

TIP

Der Name der Modulgruppe (hier *base-modules*) darf auch leer sein. Die Einstellung `repository` gibt an, von wo der Quelltext heruntergeladen werden soll, aber es kann auch eine `git://`-URL benutzt werden.

Die besondere Eigenschaft der Option „`repository kde-projects`“ liegt darin, dass `kdesrc-build` automatisch alle Git-Module einschließt, die in diesem Modul in der KDE-Projektdatenbank enthalten sind.

A.1.3 „Zweiggruppen“-Module

Bei der Weiterentwicklung des Konzepts der [Modul-Gruppen](#) stellte sich heraus, dass es für KDE-Entwickler immer schwieriger wird, die Namen von Git-Zweigen in einer Vielzahl von Quelltextarchiven abzugleichen, besonders beim Entwicklungssprung für das neue KDE Frameworks für Qt™ 5.

Dazu wurde das Konzept von „Zweiggruppen“ entwickelt. Es ermöglicht Benutzern und Entwicklern, nur eine von wenigen Gruppen zu wählen. Das Skript kann dann automatisch den passenden Git-Zweig ermitteln.

`kdesrc-build` unterstützt diese Funktion seit Version 1.16-pre2 mit der Option `branch-group`.

Example A.1 Beispiel für die Anwendung von „branch-group“

„branch-group“ kann in der Einrichtungsdatei wie folgt benutzt werden:

```
global
  # Wählt KDE Frameworks 5 und andere Qt5-basierte Programme
  branch-group kf5-qt5

  # Weitere globale Optionen hier ...
end global

module-set
  # branch-group funktioniert nur für KDE-Projekte
  repository kde-projects

  # branch-group wird von der global definierten Einstellung benutzt,
  # aber kann hier angegeben werden.

  use-modules kdelibs kde-workspace
end module-set

# Der Zweig für kdelibs ist "frameworks"
# Der Zweig für kde-workspace ist "master" (Stand August 2013)
```

In diesem Fall ergibt dieselbe Option `branch-group` unterschiedliche Zweignamen für jedes Git-Modul.

Diese Funktion erfordert einige Daten, die von KDE-Entwicklern in einem Git-Archiv mit dem Namen `kde-build-metadata` verwaltet werden. Dieses Modul wird automatisch durch `kdesrc-build` mit heruntergeladen, wie Sie in der Ausgabe des Skripts lesen können.

TIP

Bei KDE-Modulen, für die kein Zweignamen für die von Ihnen ausgewählte Zweiggruppe festgelegt ist, haben als Voreinstellung einen passenden Zweignamen, so als wenn Sie die Option `branch-group` nicht angegeben hätten.

Anhang B

Überholte Prozedur zur Einrichtung von Profilen

B.1 Einrichtung eines Profils für die KDE-Anmeldung

Diese Informationen erläutern, wie ein Profil eingerichtet werden kann, damit Sie sich für die neu erstellte Plasma-Arbeitsumgebung von KDE anmelden können. Normalerweise wird dies automatisch von kdesrc-build ausgeführt, siehe xref linkend="session-driver"/>. Dieser Anhang ist hilfreich, wenn Sie die Unterstützung von kdesrc-build für die Einrichtung von Anmeldeprofilen nicht verwenden können. Da diese Informationen nicht immer aktuell sein können, lesen Sie bitte auch die Datei `kde-env-master.sh`, die im Quelltextpaket von kdesrc-build enthalten ist.

B.1.1 Einstellungen des Anmeldeprofils ändern

WICHTIG

Die Datei `.bash_profile` enthält die Einstellungen zur Anmeldung für die von vielen Linux[®]-Distributionen verwendete bash-Shell. Benutzen Sie eine andere Shell, müssen alle in diesem Abschnitt genannten Beispiele dafür angepasst werden.

Öffnen Sie die Datei `.bash_profile` in Ihrem persönlichen Ordner oder erzeugen Sie eine neue, falls sie noch nicht existiert. Am Ende der Datei fügen Sie folgendes ein:

```
QTDIR=(pfad zu qt_dir) # wie z. B. ~/kdesrc/build/qt als Standard.
KDEDIR=(pfad zu kdedir) # wie z. B. ~/kde als Standard.
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH
MANPATH=$QTDIR/doc/man:$MANPATH

# Überprüfen, ob LD_LIBRARY_PATH gesetzt ist.
if [ -z $LD_LIBRARY_PATH ]; then
    LD_LIBRARY_PATH=$KDEDIR/lib:$QTDIR/lib
else
    LD_LIBRARY_PATH=$KDEDIR/lib:$QTDIR/lib:$LD_LIBRARY_PATH
fi

export QTDIR KDEDIRS PATH MANPATH LD_LIBRARY_PATH
```

Handbuch zum Skript kdesrc-build

Wenn Sie das Module qt nicht selbst bauen, sondern das in Ihrem System installierte Qt™ benutzen, dann fügen Sie dies ein:

```
KDEDIR=(path to kdedir) # wie z. B. ~/kde als Standard.
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH

# Überprüfen, ob LD_LIBRARY_PATH gesetzt ist.
if [ -z $LD_LIBRARY_PATH ]; then
    LD_LIBRARY_PATH=$KDEDIR/lib
else
    LD_LIBRARY_PATH=$KDEDIR/lib:$LD_LIBRARY_PATH
fi

export KDEDIRS PATH LD_LIBRARY_PATH
```

Wenn Sie keinen eigenen Benutzer verwenden, ändern Sie die Umgebungsvariable \$KDEHOME in .bash_profile für die neue Umgebung.

```
export KDEHOME="${HOME}/.kde-svn"

# Create it if needed
[ ! -e ~/.kde-svn ] && mkdir ~/.kde-svn
```

ANMERKUNG

Falls das K-Menü später leer oder völlig überfüllt mit Programmen Ihrer Distribution sein sollte, müssen Sie unter Umständen die XDG-Umgebungsvariable in .bash_profile anpassen:

```
XDG_CONFIG_DIRS="/etc/xdg"
XDG_DATA_DIRS="${KDEDIR}/share:/usr/share"
export XDG_CONFIG_DIRS XDG_DATA_DIRS
```

B.1.2 Starten von KDE

Nachdem jetzt die Einstellungen der Umgebungsvariablen für das richtige KDE angepasst wurden, muss auch der richtige Befehl **startkde** benutzt werden.

Öffnen Sie die Datei .xinitrc in Ihrem persönlichen Ordner. Erzeugen Sie sie, falls nötig. Fügen Sie die folgende Zeile hinzu:

```
exec ${KDEDIR}/bin/startkde
```

WICHTIG

Für einige Distributionen ist es erforderlich, dies auch in die Datei .xsession in Ihrem persönlichen Ordner einzutragen. Dies gilt besonders, wenn grafische Anmeldeverwaltungen wie kdm,gdm oder xdm verwendet werden.

Starten Sie nun Ihr neues KDE. In BSD und Linux® mit Unterstützung für virtuelle Terminals können Sie die Tastenkombination **Strg+Alt+F1 ... Strg+Alt+F12** benutzen, um zwischen den Terminals zu wechseln. Auf diese Weise können Sie mehrere Arbeitsumgebungen gleichzeitig ausführen. Die ersten sechs Terminals sind Text-Terminals, die folgenden sechs sind grafische Terminals.

Handbuch zum Skript kdesrc-build

Falls beim Systemstart die grafische Anmeldeverwaltung erscheint, können Sie Ihr neues KDE dennoch verwenden, auch wenn es nicht in der Liste für die grafische Anmeldung aufgeführt ist. In den meisten Anmeldeverwaltungen wie zum Beispiel kdm kann eine „benutzerdefinierte Sitzung“ bei der Anmeldung gestartet werden. Mit dieser Option werden die Einstellungen für die Sitzung aus der Datei `.xsession` im persönlichen Ordner gelesen. Wenn Sie diese Datei bereits wie oben beschrieben geändert haben, sollte mit dieser Option Ihre neue KDE-Installation starten.

Falls dies nicht funktioniert, gibt es noch eine weitere Möglichkeit zum Starten: Drücken Sie **Strg+Alt+F2** um zu einem Text-Terminal zu gelangen. Melden Sie sich als der separate KDE-Benutzer an und geben Sie folgendes ein:

```
startx -- :1
```

TIP

Sie können das aus den Quelltexten installierte KDE und das KDE aus Ihrer Distribution gleichzeitig ausführen. Melden Sie sich als normaler Benutzer an und starten Sie das KDE-System Ihrer Distribution. Drücken Sie dann **Strg+Alt+F2** (oder **F1**, **F3**, usw.) und öffnen damit eine Textkonsole. Melden Sie sich als KDE-Subversion-Benutzer an und geben Sie ein:

```
startx -- :1
```

Zurück zum KDE-System des normalen Benutzers kommen Sie mit den Kurzwahltasten für diese Arbeitsumgebung. Das ist normalerweise **Strg+Alt+F7**, manchmal auch **F6** oder **F8**. Zum mit `kdesrc-build` kompilierten KDE wechseln Sie mit der gleichen Tastenkombination und verwenden dabei nur die nächste F-Taste. Benutzen Sie zum Beispiel **Strg+Alt+F7** zum Wechsel in das normale KDE, dann drücken Sie **Strg+Alt+F8**, um wieder zurück zum `kdesrc-build`-KDE zu kommen.