

Handboek scripts voor kdesrc-build

Michael Pyne

Carlos Woelz

Vertaler/Nalezer: Freek de Kruijf

Vertaler: Ronald Stroethoff



Handboek scripts voor kdesrc-build

Inhoudsopgave

1	Inleiding	8
1.1	Een beknopte introductie in kdesrc-build	8
1.1.1	Wat is kdesrc-build?	8
1.1.2	De werking van kdesrc-build' in een notendop'	8
1.2	Documentatieoverzicht	9
2	Grondbeginselen	10
2.1	Het systeem voorbereiden voor het bouwen van KDE	10
2.1.1	Een nieuw gebruikersaccount instellen	10
2.1.2	Ga na of uw systeem klaar is om vanuit de KDE software te bouwen	10
2.1.3	kdesrc-build instellen	12
2.1.3.1	kdesrc-build installeren	12
2.1.3.2	Het configuratiebestand voorbereiden	12
2.1.3.2.1	Handmatig instellen van het configuratiebestand	12
2.2	Instellen van de configuratiegegevens	13
2.3	Het kdesrc-build-script gebruiken	14
2.3.1	Laden van metagegevens van het project	14
2.3.2	Bekijken wat er gaat gebeuren als u kdesrc-build opstart	15
2.3.3	Het oplossen van een mislukte bouw	16
2.4	Specifieke modules bouwen	16
2.5	De omgeving instellen om uw KDEPlasma Desktop te starten	17
2.5.1	Automatisch een login driver installeren	18
2.5.1.1	Xsession ondersteuning voor distributies toevoegen	18
2.5.1.2	Handmatig ondersteuning voor xsession toevoegen	18
2.5.2	De omgeving handmatig instellen	19
2.6	Organisatie en selectie van modules	19
2.6.1	Organisatie van KDE Software	19
2.6.2	Selectie van te bouwen modules	20
2.6.3	Module Sets	20
2.6.3.1	Het basisidee module set	20
2.6.3.2	Speciale ondersteuning voor KDE module sets	21
2.6.4	De officiële database voor KDE-modules	22
2.6.5	KDE project modules uitfilteren	23
2.7	Samenvatting Grondbeginselen	24

3	Mogelijkheden van het script	25
3.1	Overzicht van de mogelijkheden	25
3.2	Log van het bouwproces van kdesrc-build	27
3.2.1	Overzicht logging	27
3.2.1.1	Indeling logging-map	27
4	De kdesrc-build configureren	28
4.1	Overzicht van de kdesrc-build configuratie	28
4.1.1	Indeling van het configuratiebestand	28
4.1.1.1	Globale configuratie	28
4.1.1.2	Module configuratie	28
4.1.1.3	Verwerken van optiewaarden	29
4.1.1.4	'options' modules	29
4.1.2	Naar andere configuratiebestanden verwijzen	30
4.1.3	De belangrijkste configuratieopties	30
4.2	Tabel met beschikbare configuratieopties	31
5	Commandoregel-opties en omgevingsvariabelen (Environment Variables)	69
5.1	Gebruik van de commandoregel	69
5.1.1	Veel gebruikte commandoregel-opties	69
5.1.2	Specificatie van te bouwen modules	70
5.2	Ondersteunde omgevingsvariabelen	70
5.3	Ondersteunde commandoregelparameters	70
6	kdesrc-build gebruiken	78
6.1	Inleiding	78
6.2	Standaard functionaliteiten van kdesrc-build	78
6.2.1	qt-ondersteuning	78
6.2.2	Standaard flags toegevoegd door kdesrc-build	79
6.2.3	Het wijzigen van de bouwprioriteit van kdesrc-build	79
6.2.4	Installatie als de Root	80
6.2.5	De voortgang van het bouwen van een module tonen	81
6.3	Uitgebreide functionaliteiten	81
6.3.1	Een module gedeeltelijk bouwen	81
6.3.1.1	Gedeeltes van een module uitchecken	81
6.3.1.2	Mappen uit een bouw verwijderen	82
6.3.2	Ondersteuning voor branching en tagging bij for kdesrc-build	82
6.3.2.1	Wat zijn branches en tags?	82
6.3.2.2	Het gebruik van branches en tags	82
6.3.2.3	Geavanceerde opties wat betref branches	83

Handboek scripts voor kdesrc-build

6.3.3	Hoe kdesrc-build probeert zich te verzekeren van een succesvolle bouw . . .	84
6.3.3.1	Automatisch herbouwen	84
6.3.3.2	Handmatig een module herbouwen	84
6.3.4	De instellingen van de omgevingsvariabelen wijzigen	84
6.3.5	Hervatten van bouwprocessen	85
6.3.5.1	Hervatten na een mislukte of afgebroken bouwproces	85
6.3.5.2	Modules tijdens de bouw negeren	85
6.3.6	Vanaf de commandoregel opties wijzigen	85
6.3.6.1	Algemene (global) opties wijzigen	85
6.3.6.2	Module opties wijzigen	86
6.4	Functionaliteiten voor KDE-ontwikkelaars	86
6.4.1	SSH Agent controles	86
6.5	Andere functionaliteiten van kdesrc-build	87
6.5.1	Het aantal meldingen van kdesrc-build wijzigen	87
6.5.2	Gekleurde meldingen	87
6.5.3	Na een bouw de onnodige mappen verwijderen	87
7	CMake, het bouwsysteem voor KDE	89
7.1	Inleiding tot CMake	89
8	Dankbetuigingen en licentie	90
A	De organisatie van de modules en broncode van KDE	91
A.1	De 'Module'	91
A.1.1	Individuele modules	91
A.1.2	Groepen van gerelateerde modules	91
A.1.3	Module 'branch groups'	92
B	Achterhaalde procedures voor het instellen van profielen	93
B.1	Een KDE aanmeldprofiel instellen	93
B.1.1	Uw instellingen voor het opstartprofiel wijzigen	93
B.1.2	KDE opstarten	94

Lijst van tabellen

4.1	Tabel met opties	68
-----	----------------------------	----

Samenvatting

kdesrc-build is een script dat KDE software bouwt en installeert direct vanaf de broncode in de broncoderepositories van het KDE project.

Hoofdstuk 1

Inleiding

1.1 Een beknopte introductie in kdesrc-build

1.1.1 Wat is kdesrc-build?

kdesrc-build is een script om de KDE community te helpen met het installeren van KDE software van broncoderepositories in [Git](#) en [Subversion](#), en om daarna de software voortdurend bij te werken. Het is met name bedoeld om diegene te helpen die de KDE-software ontwikkelen en testen, inclusief de gebruikers die bugfixes testen en de ontwikkelaars die aan nieuwe functionaliteiten werken.

U kunt het script kdesrc-build instellen om een enkele individuele module, het complete KDE bureaublad met alle KDE-programma's, of iets daartussen in te onderhouden.

Om te kunnen beginnen, leest u hoofdstuk 2, of u gaat verder met lezen over de details van hoe kdesrc-build werkt en van een overzicht van documentatie.

1.1.2 De werking van kdesrc-build 'in een notendop'

kdesrc-build maakt gebruik van hulpmiddelen die op de opdrachtregel beschikbaar zijn voor de gebruiker, met dezelfde voor de gebruiker beschikbare interface. Wanneer kdesrc-build wordt uitgevoerd zal de volgende volgorde worden gebruikt:

1. kdesrc-build leest de [opdrachtregel](#) en het [configuratiebestand](#) om te bepalen wat te bouwen, welke opties voor compileren te gebruiken, waar te installeren, etc.
2. kdesrc-build voert het bijwerken van broncode voor elke [module](#) uit. Het bijwerken gaat door totdat alle modules zijn bijgewerkt. Modules waarvan het bijwerken mislukt stoppen het bouwen niet – u krijgt aan het eind een melding welke modules niet zijn bijgewerkt.
3. Modules die succesvol zijn bijgewerkt, worden gebouwd, hun test suite uitgevoerd, en tenslotte geïnstalleerd. Om de totale benodigde tijd te verminderen, zal kdesrc-build standaard beginnen met de code te bouwen zodra de eerste module is bijgewerkt, en zal verdergaan op de achtergrond met het bijwerken van de overige updates.

TIP

Een *zeer informatief* overzicht van hoe KDE modules worden gebouwd, inclusief informatieve diagrammen, kunt u vinden op [een online artikel dat het Krita-programma van KDE beschrijft](#). Deze workflow is wat kdesrc-build automatiseert voor alle KDE modules.

1.2 Documentatieoverzicht

Deze handleiding is een overzicht om de volgende aspecten van de werking van kdesrc-build te beschrijven:

- Een [overzicht](#) van de vereiste stappen om te beginnen.
- Met name [mogelijkheden](#).
- De syntaxis en de opties van het [configuratiebestand](#).
- De [opties van de commandoregel](#).

Ook zijn de stappen die u zou moeten nemen met andere hulpmiddelen gedocumenteerd, (bijv. stappen die niet automatisch uitgevoerd worden door kdesrc-build).

Hoofdstuk 2

Grondbeginselen

In dit hoofdstuk beschrijven we hoe u kdesrc-build kan gebruiken om modules uit de repositories van KDE uitgecheckt kunnen worden en ze te bouwen. Ook bieden we een basis uitleg van de structuur van de broncode van KDE en de stappen die u moet nemen voordat u het script gebruikt.

Alle in dit hoofdstuk gepresenteerde onderwerpen worden nog meer in detail behandeld in het artikel [Bouwen vanuit de broncode](#) op de [wiki van de KDE gemeenschap](#). Als u KDE voor de eerste keer compileert, is het een goed idee om het te lezen of het te raadplegen als bron voor referenties. U zult er gedetailleerde informatie vinden over hulpmiddelen voor het maken van pakketten en de vereisten, veel voorkomende valkuilen bij compilatie en strategieën en informatie over het gebruik van uw nieuwe KDE installatie vinden.

2.1 Het systeem voorbereiden voor het bouwen van KDE

2.1.1 Een nieuw gebruikersaccount instellen

Het is aanbevolen dat u een aparte gebruikersaccount gebruikt om uw KDE software in te bouwen, te installeren en uit te voeren, omdat er minder toegangsrechten zijn vereist en om conflicten met de pakketten in uw distributie te vermijden. Als u al KDE-pakketten heeft geïnstalleerd, dan is de beste keuze om een aparte (alleen voor dat doel) gebruiker aan te maken om de nieuwe KDE te bouwen en te gebruiken.

TIP

Door de KDE van het systeem onaangeraakt laten, heeft u de mogelijkheid om in nood op terug te vallen in het geval een gecompileerd KDE onbruikbaar is.

Maar als u het wenst dan kunt u ook het zo instellen dat u in een systeembrede map installeert (bijv. `/usr/src/local`). Dit document beschrijft dit type installatie niet, omdat we aannemen dat u in dat geval weet wat u doet.

2.1.2 Ga na of uw systeem klaar is om vanuit de KDE software te bouwen

Alvorens het script kdesrc-build gebruikt kan worden (of elke andere bouwstrategie) is het nodig dat u de ontwikkelomgeving en de bibliotheken nodig voor KDE installeert. De bijna complete lijst met vereiste hulpmiddelen is te vinden op de pagina [KDE wiki van de gemeenschap Vereisten voor bouwen](#).

Hier volgt een lijst met enkele dingen die u nodig hebt:

Handboek scripts voor kdesrc-build

- U heeft CMake nodig, deze software is wat KDE gebruikt voor de bouwconfiguratie van de broncode en de generatie van specifieke bouwcommando's voor uw systeem. De vereiste versie is afhankelijk van welke versies van KDE software u wilt bouwen (zie TechBase voor de specificaties), maar de met moderne distributies meegeleverde CMake zal goed genoeg zijn.
- U moet ook de client-software installeren die nodig is om de KDE broncode uit te checken. Dit betekent dat u minstens het volgende nodig hebt:
 - De [Git source control manager](#) die nodig is voor alle [broncode](#) van KDE.
 - Hoewel het niet is vereist, wordt de beheerder van broncode [Bazaar](#) gebruikt voor een enkele module (libdbusmenu-qt) die nodig is voor de KDE bibliotheken. de meeste gebruikers kunnen deze bibliotheek installeren via hun pakketten uit de distributie maar kdesrc-build ondersteunt ook het bouwen als u dat wilt. Maar om libdbusmenu-qt te kunnen bouwen moet Bazaar zijn geïnstalleerd.
- De Perl script-taal is vereist voor kdesrc-build, sommige KDE repositories, en Qt™ (als u dat vanuit broncode wilt bouwen).

De Perl die met uw distributie meekomt zou geschikt moeten zijn (het moet tenminste Perl 5.14 moeten zijn), maar u heeft ook enkele extra modules nodig(kdesrc-build geeft een waarschuwing als ze er niet zijn):

 - IO::Socket::SSL
 - JSON::PP of JSON::XS
 - YAML::PP, YAML::XS, of YAML::Syck
- U heeft een volledige C++ ontwikkelomgeving (compiler, standaard library, runtime, en de nodige ontwikkelpakketten). De vereiste minimum versies variëren en hangen af van de KDE module: de KDE Frameworks 5 collectie ondersteunt de oudste compilers, terwijl KDE Plasma 5 en KDE-programma's waarschijnlijk meer recente compilers nodig hebben.

We bevelen tenminste de compilers GCC 4.8 of Clang 4 aan. Bij veel distributies kunt u deze hulpmiddelen makkelijk installeren via een 'build-essentials' pakket, een hulpprogramma om "build dependencies" voor Qt™ te installeren, of vergelijkbare hulpprogramma's. De KDE Community Wiki heeft een pagina [met daarop voor de grote distributies de aanbevolen pakketten](#).
- U heeft een bouwprogramma nodig dat daadwerkelijk de compilatie-stappen uitvoert (gegenereerd door CMake). Wij bevelen GNU Make aan dat via uw pakketbeheerder beschikbaar zou moeten zijn. CMake ondersteunt andere mogelijkheden, zoals het bouwprogramma Ninja, wat u door kdesrc-build kan laten gebruiken via het de optie [custom-build-command](#) in het configuratiebestand.
- Tenslotte heeft u de juiste Qt™ libraries (inclusief de development pakketten) voor de versie van de KDE software nodig die u gaat bouwen. kdesrc-build ondersteunt niet officieel het bouwen met Qt™ 5 (de huidige major versie), het wordt daarom aangeraden om de development pakketten van uw distributie te gebruiken of om de KDE Community wiki page over [self-building Qt 5](#) te bestuderen.

OPMERKING

De meeste distributies van besturingssystemen hebben een methode om makkelijk de vereiste ontwikkel-programma's te installeren. Lees de Community Wiki pagina [Required devel packages](#) om uit te vinden of deze instructies al beschikbaar zijn.

BELANGRIJK

Sommige van deze pakketten zijn verdeelt in libraries (of programma's of hulpprogramma's), en ontwikkeling-pakketten U heeft in ieder geval het programma of library *en* het ontwikkeling-pakket daarvan nodig.

2.1.3 kdesrc-build instellen

2.1.3.1 kdesrc-build installeren

De KDE-ontwikkelaars wijzigingen regelmatig kdesrc-build om het in de pas te laten lopen met de vorderingen van de ontwikkelingen in KDE, voegen verbeteringen toe aan de aanbevolen kdesrc-build-configuratie, voegen modules toe, verbeteren CMake flags, etc.

Daarom bevelen we aan om de kdesrc-build direct op te halen uit zijn broncoderepository en het daarna regelmatig bij te werken.

U kunt kdesrc-build ophalen uit zijn broncoderepository, door de volgende commando's uit te voeren:

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

Vervang `~/kdesrc-build` door de map waarin u het wilt installeren.

U kunt later kdesrc-build bijwerken door uit te voeren:

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

TIP

Wij raden aan om de kdesrc-build installatie-map aan uw `PATH` omgevingsvariabele toe te voegen, zodat u kdesrc-build kunt opstarten zonder dat u iedere keer het volledige pad daarvan hoeft op te geven.

2.1.3.2 Het configuratiebestand voorbereiden

kdesrc-build gebruikt een [configuratiebestand](#) (genaamd `~/kdesrc-buildrc`) voor het bepalen welke modules worden gebouwd, waar ze worden geïnstalleerd, enz.

U kunt een met kdesrc-build meegeleverd programma gebruiken, genaamd `kdesrc-build-setup` om een eenvoudige configuratie voor kdesrc-build te maken. U kunt dan vervolgens het `~/kdesrc-buildrc` bewerken om de gewenste wijzigingen aan te brengen.

U start `kdesrc-build-setup` zelf op van uit een terminal (in plaats van een grafische interface gebruik te maken), net zoals `kdesrc-build`, zodat u het zelfs kunt gebruiken als u nog geen grafische interface heeft.

2.1.3.2.1 Handmatig instellen van het configuratiebestand

U kunt ook uw configuratiebestand handmatig instellen, door het meegeleverde voorbeeld configuratiebestand `kdesrc-buildrc-kf5-sample` naar `~/kdesrc-buildrc` te kopiëren en vervolgens dat bestand te bewerken. hoofdstuk 4 heeft een handige uitleg hierover, met name de [tabel met configuratieopties](#).

kdesrc-build heeft vele voorbeeld configuratiebestanden voor KDE Frameworks 5, Plasma 5, en andere KDE-programma's. De `kdesrc-build-setup` maakt referenties naar deze bestanden in het configuratiebestand dat het genereert, maar u kunt ze ook zelf gebruiken. lees Section 4.1.2 voor informatie over hoe u andere configuratiebestanden van u eigen `~/kdesrc-buildrc` kunt gebruiken.

U kunt meer informatie over de syntaxis van het [configuratiebestand](#) vinden in Section 2.2 en in hoofdstuk 4.

2.2 Instellen van de configuratiegegevens

Om kdesrc-build te kunnen gebruiken, moet u in uw thuismap een bestand hebben genaamd `.kdesrc-buildrc` hebben, waar in u de algemene instellingen vastlegt en bepaalt welke modules u wilt downloaden en bouwen.

OPMERKING

Het is mogelijk om bij kdesrc-build verschillende configuratiebestanden te gebruiken, dit wordt beschreven in hoofdstuk 4. als u verschillende configuraties nodig heeft, lees dan die sectie. Hier nemen we aan dat de configuratie is te vinden in `~/kdesrc-buildrc`.

De makkelijkste manier om verder te gaan is door het bestand `kdesrc-buildrc-kf5-sample` als voorbeeld te gebruiken, de algemene instellingen naar wens te wijzigen, en ook de lijst met modules die u wilt bouwen aan te passen.

De standaardinstellingen zouden geschikt moeten zijn om een KDE-bouw uit te voeren. Enkele instellingen die u wellicht wilt wijzigen zijn:

- `kdedir`, wat de doelmap wijzigt waarin de KDE-software wordt geïnstalleerd. Standaard is dit `~/kde`, wat voor een single-user installatie is.
- `branch-group`, wat u kunt gebruiken om de toepasselijke ontwikkel-branch van de KDE modules als geheel te selecteren. U kunt uit veel bouw-configuraties kiezen maar u zal waarschijnlijk `kf5-qt5` willen kiezen zodat kdesrc-build de laatste code gebaseerd op Qt™ 5 en KDE Frameworks 5 zal downloaden.

TIP

Als u er geen kiest dan zal kdesrc-build de standaard branch groep gebruiken, maar welke de standaard is verandert in de loop van de tijd, het is daarom beter om een te kiezen zodat de branch-groep niet onverwacht wijzigt.

- `source-dir`, wat de map wijzigt wat kdesrc-build gebruikt om de broncode naar toe te downloaden, het bouwproces uit te voeren en de logbestanden op te slaan. Standaard is dit `~/kdesrc`.
- `cmake-options`, waar u de opties instelt die voor het bouwen van elke module aan het commando CMake worden doorgegeven. Normaal gebruikt u dit om te schakelen tussen 'debug' en 'release' bouw, om optionele functionaliteiten IN of UIT te schakelen, of om informatie over de locatie van de vereiste libraries aan het bouwproces door te geven.
- `make-options`, waar u de opties instelt die voor het bouwen van elke module (nadat CMake het bouwsysteem heeft ingesteld) bij het gebruik van het commando make wordt gebruikt.

De meest ingestelde optie is `-jN`, waar u `N` vervangt door het maximum aantal jobs dat u wilt toestaan. Een groter getal (tot aan het aantal logische CPUs dat uw systeem beschikbaar heeft) geeft sneller bouwen, maar vereist wel meer systeem-resources.

TIP

kdesrc-build stelt de optie `num-cores` in op het gedetecteerde aantal beschikbare verwerkingskernen. U kunt deze waarde in uw eigen configuratiebestand gebruiken om te vermijden het handmatig te moeten instellen.

Example 2.1 Make configureren om alle beschikbare CPU's te gebruiken, met uitzonderingen

```
global
  # Deze omgevingsvariabele wordt automatisch door make gebruikt, ↔
  inclusief
  # make-commando's die niet direct worden uitgevoerd door kdesrc-build, ↔
  zoals configuratie van 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 # Reduced number of build jobs for just these modules
end module-set
```

OPMERKING

Sommige zeer grote Git repositories kunnen uw systeem verdrinken als u probeert om teveel bouwprocessen tegelijk uit te voeren, met name repositories zoals van Qt™ WebKit en de Qt™ WebEngine. Om het systeem interactief te houden, kan het nodig zijn dat u voor bepaalde modules het aantal bouwprocessen beperkt.

Voorbeeld 2.1 Geeft een voorbeeld hoe u dit kunt doen.

U wilt waarschijnlijk andere modules selecteren om te gaan bouwen, dit is beschreven in Section 2.6.2.

2.3 Het kdesrc-build-script gebruiken

Nu de configuratie-data is ingesteld, bent u klaar om het script op te starten. Zelfs als u nog iets wilt instellen, of u wilt nog iets wilt bestuderen, is het een goed idee om tenminste de KDE project metadata te downloaden.

2.3.1 Laden van metagegevens van het project

Vanuit een terminalvenster, logt u in als de gebruiker waarmee u KDE software wilt compileren en start u het script op:

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

Dit commando zal de map voor de broncode instellen en verbinding zoeken met de KDE Git repositories om de database van de KDE git repositories, en de database van de dependency metadata te downloaden, zonder verdere wijzigingen uit te voeren. Het is verstandig om dit separaat uit te voeren omdat deze metadata ook handig is voor andere commando's van kdesrc-build.

2.3.2 Bekijken wat er gaat gebeuren als u kdesrc-build opstart

Met de metadata van het project geïnstalleerd, is het mogelijk om te bekijken van wat kdesrc-build zal gaan doen als het wordt opgestart. Dit kunt u doen met de commandoregelparameter `--pretend`.

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

U zou een melding moeten zien dat sommige pakketten succesvol zijn gebouwd (ofschoon er niets daadwerkelijk is gebouwd). Als er geen belangrijke problemen zijn te zien, dan kunt u verder gaan door het script daadwerkelijk te starten.

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

Dit commando zal de juiste broncode downloaden, en elke module in de opgedragen volgorde bouwen en installeren, maar het zal stoppen als het niet lukt om een module te bouwen (vanwege de optie `--stop-on-failure`). Naderhand zou u een uitvoer moeten zien vergelijkbaar met dat in Voorbeeld 2.2:

Example 2.2 Voorbeeld uitvoer van een uitgevoerde 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 Het oplossen van een mislukte bouw

Afhankelijk van hoeveel modules u download, is het mogelijk dat kdesrc-build het de eerste keer dat u KDE software compileert niet lukt. Wanhoop niet!

kdesrc-build logt de uitvoer van elk uitgevoerd commando. Standaard worden de logbestanden bewaard in `~/kdesrc/log`. Om uit te vinden wat de oorzaak van een fout bij een module in de laatste kdesrc-build commando is, is het meestal voldoende om `~/kdesrc/log/latest/ module -name /error.log` te bestuderen.

TIP

Misschien is de eenvoudigste manier om uit te vinden welke fout de oorzaak was dat een module niet lukte om te bouwen, het naar achteren hoofdletter-ongevoelig zoeken, aan het eind van het bestand startend naar het woord `error`. Als u die heeft gevonden, scrollt u omhoog om er zeker van te zijn dat er geen andere foutmeldingen in de buurt zijn. De eerste foutmelding in een groep is meestal het onderliggende probleem.

In dat bestand kunt u de fout vinden waardoor het bouwen van die module mislukte. Als het bestand (onderaan) zegt dat er bij u enkele pakketten ontbreken, probeer dan het pakket (inclusief toepasselijke `-dev` pakketten) voordat u opnieuw probeert dat pakket te bouwen. Zorg ervoor dat wanneer u kdesrc-build opnieuw opstart, de optie `--reconfigure` toevoegt om daarmee kdesrc-build te forceren om bij de module opnieuw te controleren voor de ontbrekende pakketten.

Of, als de fout een fout in het bouwproces lijkt te zijn (zoals een `syntax error`, `'incorrect prototype'`, `'unknown type'`, of iets dergelijks) dan is het waarschijnlijk een fout in de KDE-broncode, wat hopelijk in een paar dagen opgelost zal zijn. Als het niet in die tijd opgelost is, aarzel dan niet om een email te sturen naar de kde-devel@kde.org mailing list (het kan nodig zijn om eerst in te schrijven) om melding te maken van de build failure.

U kunt meer vaak voorkomende voorbeelden van dingen die fout kunnen gaan en de oplossingen daarvoor, met daarnaast algemene tips en strategieën voor het bouwen van KDE software vinden in de [Bouwen vanuit de broncode](#).

Aan de andere kant, aangenomen dat alles goed is gegaan, dan zou een nieuwe KDE op uw computer geïnstalleerd moeten zijn, en is het nu een kwestie van eenvoudig opstarten, zoals hierna beschreven in [Section 2.5](#).

OPMERKING

Meer informatie over de mogelijkheden van kdesrc-build voor het loggen, leest u in [Section 3.2](#).

2.4 Specifieke modules bouwen

In plaats van alle modules elke keer te bouwen, wilt u wellicht alleen een enkele module bouwen, of een andere kleine verzameling. In plaats van uw configuratiebestand aan te passen, kunt u op de commandoregels gewoon de namen van de te bouwen modules of module sets opgeven.

Example 2.3 Voorbeeld uitvoer van de bouw van een specifieke module door kdesrc-build gebouwd

```
% 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
```

Ofschoon in dit geval alleen het programma *dolphin* was opgegeven, zorgde de flag `--include-dependencies` ervoor dat kdesrc-build ook de voor *dolphin* opgegeven dependencies (door de optie [include-dependencies](#) op yes te zetten).

OPMERKING
De oplossing voor de dependency werkte in dit geval alleen omdat *dolphin* toevallig in een kde-projects-gebaseerde module set is gespecificeerd (in dit geval namelijk *base-apps*). Zie [Section 2.6.3.2](#).

2.5 De omgeving instellen om uw KDEPlasma Desktop te starten

Aangenomen dat u een speciale gebruiker heeft gebruikt om KDE Plasma te bouwen, en u al een geïnstalleerde Plasma versie heeft, dan kan het starten van uw nieuwe Plasma een beetje tricky

zijn, omdat de nieuwe Plasma voorrang moet krijgen over de oude. U moet de omgevingsvariabele van uw login scripts wijzigen om er voor te zorgen dat de nieuw gebouwde desktop wordt gebruikt.

2.5.1 Automatisch een login driver installeren

Vanaf versie 1.16, zal kdesrc-build proberen om een geschikte login driver te installeren, waarmee u kunt inloggen naar uw kdesrc-build-gebouwde KDE desktop vanaf uw inlog manager. Dit kunt u uitschakelen door de optie `install-session-driver` in het configuratiebestand te gebruiken.

OPMERKING

Als kdesrc-build in de pretend mode zal de sessie niet ingesteld worden.

Deze driver werkt door een aangepaste 'xsession' sessie-type op te zetten. Dit sessie-type zou standaard moeten werken met kdm inlog manager (waar het verschijnt als een 'Aangepast' sessie), maar voor andere inlog managers (zoals LightDM en gdm) kan het nodig zijn dat u extra bestanden moet installeren om xsession mogelijk te maken.

2.5.1.1 Xsession ondersteuning voor distributies toevoegen

De standaard login managers van sommige distributies kunnen extra pakketten nodig hebben om xsession logins te kunnen ondersteunen.

- De [Fedora Linux](#)[®] distributie verlangt dat het `xorg-x11-xinit-session` pakket geïnstalleerd is voor aangepaste xsession login support.
- [Debian](#) en van Debian-afgeleide Linux[®] distributies zouden aangepaste xsession logins moeten ondersteunen, maar vereisen wel dat de `allow-user-xsession` optie is ingesteld in `/etc/X11/Xsession.options`. Zie ook de [Debian documentatie over het aanpassen van de X sessie](#).
- Ga voor andere distributies naar [Section 2.5.1.2](#).

2.5.1.2 Handmatig ondersteuning voor xsession toevoegen

Als er voor uw distributie geen distributie-specifieke aanwijzingen aanwezig zijn in [Section 2.5.1.1](#), dan kunt u als volgt handmatig een 'Custom xsession login' entry aan de lijst van sessie-types voor uw distributie toevoegen:

OPMERKING

Deze procedure heeft waarschijnlijk administratieve privileges nodig om uitgevoerd te kunnen worden.

1. Creëer het bestand `/usr/share/xsessions/kdesrc-build.desktop`.
2. Zorg ervoor dat in het zojuist aangemaakte bestand de volgende tekst voorkomt:

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

- De entry `$HOME` moet u vervangen door het volledige pad naar uw persoonlijke map (voorbeeld, `/home/ user`). De desktop entry specificatie is niet toegestaan in algemene (user-generic) bestanden.

3. Als de login manager opnieuw is gestart, dan zou het een nieuw sessie type moeten tonen, 'KDE Plasma Desktop (unstable; kdesrc-build)' in zijn lijst met sessies, die zou moeten proberen om de `.xsession`-bestand geïnstalleerd door kdesrc-build op te starten als u die selecteert bij het inloggen.

OPMERKING

Het is wellicht het makkelijkste de computer te herstarten om de login manager te herstarten, als de login manager de updates van de `/usr/share/xsessions` map niet opmerkt.

2.5.2 De omgeving handmatig instellen

Deze documentatie had instructies over welke omgevingsvariabele u moest instellen om de nieuw gebouwde desktop te laden. Deze instructies zijn verplaatst naar een appendix (Section B.1).

Als u de intentie heeft om uw eigen login support in te stellen dan kunt die appendix consulteren of het bestand `sample-kde-env-master.sh` bekijken dat met de kdesrc-build-broncode is meegeleverd.

2.6 Organisatie en selectie van modules

2.6.1 Organisatie van KDE Software

KDE software is opgesplitst in verschillende onderdelen, veel daarvan kan door kdesrc-build worden gebouwd. Begrip van de indeling hiervan helpt u bij het correct selecteren van de software modules die u wilt bouwen.

1. Op het laagste niveau komt de Qt™ library, wat een erg krachtige, cross-platform 'toolkit' library is. KDE is gebaseerd op Qt™, en sommige van de niet-KDE libraries die vereist zijn door KDE zijn ook gebaseerd op Qt™. kdesrc-build kan Qt™ bouwen, of diegene gebruiken die al op uw systeem is geïnstalleerd als deze recent genoeg is.
2. Boven op die van Qt™ zijn er meer vereiste libraries die nodig zijn voor KDE software om te kunnen werken. Sommige van deze libraries worden niet als onderdeel van KDE zelf beschouwt vanwege hun algemene inzetbaarheid, maar zijn toch essentieel voor het KDE Platform. Deze libraries zijn verzameld in een `kdesupport` module maar worden toch niet als een onderdeel van 'Frameworks' beschouwt.
3. Boven op deze essentiële libraries komen de [KDE Frameworks](#), soms afgekort als KF5, wat essentiële libraries zijn voor de KDE Plasma desktop, KDE programma's, en software van anderen.
4. Boven op het Platform, komen verschillende anderen dingen:
 - 'Third-party' programma's. Dit zijn programma's die het KDE Frameworks wel gebruiken maar niet door het KDE project zijn geprogrammeerd of daarmee geassocieerd zijn.
 - Plasma, wat een volledige 'workspace' desktop omgeving is. Dit is wat gebruikers normaal gesproken zien als ze 'inloggen bij KDE'.
 - Het programma-pakket van KDE. Dit is een verzameling van handige software dat met het Platform en de Plasma Desktop wordt meegeleverd, onderverdeelt in aparte modules, waaronder ook hulpprogramma's zoals Dolphin, spelletjes zoals KSudoku, en productie software gereleased door KDE zoals Kontakt.
 - Tenslotte, is er een verzameling software (ook in modules verzameld) waarvan de ontwikkeling door KDE resources wordt ondersteunt (zoals vertalingen, source control, bug tracking, etc.) maar niet door KDE is gereleased als onderdeel van Plasma of de verzameling programma's. Deze modules staan bekend als 'Extragear'.

2.6.2 Selectie van te bouwen modules

De selectie van de mogelijke modules voor het bouwen, regelt u in [het configuratiebestand](#). Na de sectie `global` is een lijst van te bouwen modules, Omsloten door `module ...` end module regels. Een voorbeeld van een module is te zien in [Voorbeeld 2.4](#).

Example 2.4 Voorbeeld van een module-item in het configuratiebestand

```
module kdesrc-build-git
  # Opties voor deze module komen hier, voorbeeld:
  repository kde:kdesrc-build
  make-options -j4 # Voer 4 compilaties tegelijk uit
end module
```

OPMERKING

In de praktijk wordt deze module constructie meestal niet direct gebruikt. In plaats daarvan zijn de meeste modules zoals hierboven beschreven gespecificeerd via module-sets.

Als u alleen `module` entries gebruikt, dan zal `kdesrc-build` ze in de opgegeven volgorde bouwen, en zal niet proberen om andere dan de door u opgegeven repositories te downloaden.

2.6.3 Module Sets

De broncode van KDE is onderverdeelt in een groot aantal relatief kleine op Git gebaseerde repositories. Om het grote aantal repositories dat nodig is bij een op KDE-gebaseerde installatie makkelijk te beheren, ondersteunt `kdesrc-build` het groeperen van meerdere modules en behandelt een dergelijke groep als een 'module set'.

2.6.3.1 Het basisidee module set

Door een module set te gebruiken, kunt u op een snelle manier declareren welke Git modules u wilt downloaden en bouwen, net alsof u voor elk een aparte module declaratie heeft getyped. De optie `repository` is speciaal bedoelt om in te stellen waar elke module vandaan moet worden gedownload, en elke andere optie die in de module set voorkomt op dezelfde manier naar elke andere module wordt gekopieerd.

Example 2.5 Modulen sets gebruiken

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

module qt
  # Opties verwijderd voor beknoptheid
end module

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

# Andere modules indien nodig...
module kdesupport
end module
```

In Voorbeeld 2.5 ziet u een kleine module set. Als kdesrc-build deze module set tegenkomt, dan handelt het alsof, bij elke module opgegeven in `use-modules`, dat een individuele module is gedeclareerd met zijn `repository` overeenkomstig met de module-set's `repository` onmiddellijk gevolgd door de opgegeven modulenaam.

Daarnaast kunt u in een module set andere opties opgegeven, die naar elke nieuwe module wordt gekopieerd dat op deze manier wordt gecreëerd. Door module-set te gebruiken, is het mogelijk om op een snelle manier veel Git modules te declareren die alle op dezelfde repository URL zijn gebaseerd. Daarnaast is het mogelijk om een bepaalde module-sets een naam te geven (zoals in het voorbeeld te zien is), zodat u vanaf de commandoregel makkelijk naar een hele groep van modules kunt refereren.

2.6.3.2 Speciale ondersteuning voor KDE module sets

De tot nu toe beschreven ondersteuning voor module set is voor alle op Git gebaseerde modules. Voor de Git repositories van KDE, heeft kdesrc-build extra functionaliteiten om het makkelijker voor de gebruikers en de ontwikkelaars te maken. U schakelt deze functionaliteit in door `kde-projects` te specificeren voor de `repository` in de module set.

kdesrc-build bouwt normaal gesproken alleen de modules die u heeft opgegeven in uw configuratiebestand, in de opgegeven volgorde. Maar bij een `kde-projects` module set, kan kdesrc-build de dependency van de KDE-specifieke modules oplossen, en kan automatisch modules aan de bouw toevoegen zelfs als ze indirect zijn gespecificeerd.

Example 2.6 kde-projects module sets gebruiken

```
# Voegt alleen een module voor juk toe (de kde/kdemultimedia/juk repo)
module-set juk-set
  repository kde-projects
  use-modules juk
end module-set

# Voegt alle modules toe die in kde/multimedia/* aanwezig zijn, inclusief ↔
  juk,
# maar geen andere dependencies
module-set multimedia-set
  repository kde-projects
  use-modules kde/multimedia
end module-set

# Voegt alle modules toe die in kde/multimedia/* aanwezig zijn, en all kde- ↔
  projects
# dependencies die buiten kde/kdemultimedia zijn
module-set multimedia-deps-set
  repository kde-projects
  use-modules kde/multimedia
  include-dependencies true
end module-set

# Alle modules die in deze drie module sets zijn gecreëerd worden ↔
  automatisch in
# juiste dependency volgorde gezet, onafhankelijk van de instellingen voor ↔
  de include-dependencies
```

TIP

Deze module set constructie `kde-projects` is de belangrijkste methode om te declareren welke modules u wilt bouwen.

Alle module sets gebruiken de opties `repository` en `use-modules`. `kde-projects` module sets hebben de voorgedefinieerde waarde `repository`, maar andere types module sets kunnen ook de optie `git-repository-base` gebruiken.

2.6.4 De officiële database voor KDE-modulen

Bij de Git repositories van KDE is mogelijk om bij elkaar horende Git modules te groeperen in verzamelingen van bij elkaar horen modules (b.v. `kdegraphics`). Git herkent deze groeperingen niet, maar `kdesrc-build` kan deze groepen wel begrijpen, door `module sets` met de optie `repository` ingesteld op `'kde-projects'`.

`kdesrc-build` zal herkennen dat de `kde-projects` repository een speciale behandeling vereist, en zal het bouwproces aanpassen. Naast andere dingen, zal `kdesrc-build`:

- De laatste modulendatabase vanaf het [Git-archief van KDE](#) downloaden.
- In die database een module proberen te vinden met de naam opgegeven in de `use-modules` instelling van de moduleset.
- Voor elke gevonden module, zal `kdesrc-build` de juiste repository in de database opzoeken, gebaseerd op de instelling `branch-group`. Als een repository bestaat en geldig is voor de `branch group`, dan zal `kdesrc-build` deze automatisch gebruiken om de broncode te downloaden of up te daten.

OPMERKING

In de huidige database, hebben sommige module groepen niet alleen een verzameling modules, maar ze declareren *ook* hun eigen Git repository. In zulke situaties zal `kdesrc-build` op dit moment de voorkeur geven aan de Git repository van de groep in plaats van de repositories van de onderdelen.

Het volgende voorbeeld toont hoe u de KDE module database moet gebruiken om de Phonon multimedia library te installeren.

```
module-set media-support
# Deze optie moet kde-projects zijn om de module database te gebruiken.
repository kde-projects

# Deze optie geeft aan naar welke modules gezocht moet worden in de ↔
database.
use-modules phonon/phonon phonon-gstreamer phonon-vlc
end module-set
```

TIP

`phonon/phonon` is gebruikt omdat (met de huidige project database) `kdesrc-build` anders moet kiezen tussen de groep projecten genaamd `'phonon'` en het individuele project genaamd `'phonon'`. Op dit moment zal `kdesrc-build` de eerste selecteren, zodat er veel meer backends dan nodig worden gebouwd.

Het volgende voorbeeld is misschien realistischer, en toont een functionaliteit die alleen bij de KDE module database beschikbaar is: het met alleen maar een enkele declaratie bouwen van alle KDE grafische programma's.

Handboek scripts voor kdesrc-build

```
module-set kdegraphics
# Deze optie moet kde-projects zijn om de module database te gebruiken.
repository kde-projects

# Deze optie geeft aan naar welke modules gezocht moet worden in de ↔
database.
use-modules kdegraphics/libs kdegraphics/*
end module-set
```

Er worden hier twee belangrijke mogelijkheden gedemonstreerd:

1. kdesrc-build staat het u toe dat u modules opgeeft die afstammen van een bepaalde module, zonder dat de ouder module gebouwd moet worden, door gebruik van de syntax **module-name /***. In dit geval is dat eigenlijk vereist omdat de basismodule, kdegraphics, als inactief is gemarkeerd zodat het niet per ongeluk samen met zijn kinderen gebouwd kan worden. Door de afstammende modules op te geven, kan kdesrc-build de uitgeschakelde module overslaan.
2. kdesrc-build zal een bepaalde module niet meer dan een keer aan de bouwlijst toevoegen. Hierdoor kunnen wij handmatig kdegraphics/libs bovenaan de lijst toevoegen, voor de rest van kdegraphics, zonder dat geprobeerd wordt om kdegraphics/libs twee keer te bouwen. Dit was vroeger nodig voor het afhandelen van de juiste dependencies, en is nu een reserve voor het geval dat de dependency metadata in de KDE project database ontbreekt.

2.6.5 KDE project modules uitfilteren

U zou kunnen besluiten dat u alle programma's in een KDE module groep wilt bouwen *behalve* een bepaald programma.

Bijvoorbeeld, in de groep kdeutils is ook een programma genaamd kremotecontrol. Als uw computer niet de juiste hardware heeft om de door een afstandsbediening verzonden signalen te ontvangen, dan zou u kunnen besluiten dat u kremotecontrol niet iedere keer wilt downloaden, compileren en installeren als u kdeutils update.

U kunt dit bereiken door de configuratieoptie **modules negeren** te gebruiken. Op de opdrachtregel doet de optie **--ignore-modules** hetzelfde, maar het is gemakkelijker om een module eenmalig uit te filteren.

Example 2.7 Voorbeeld van het negeren van een kde-project module in een groep

```
module-set utils
  repository kde-projects

  # Hier stelt u in welke modules in de database worden opgezocht.
  use-modules kdeutils

  # Hier stelt u in welke modules worden "uitgefilterd" van de modules ←
    gekozen via use-modules, hierboven.
  ignore-modules kremotecontrol
end module-set

module-set graphics
  repository kde-projects

  # Hier stelt u in welke modules in de database worden opgezocht.
  use-modules extragear/graphics

  # Hier stelt u in welke modules worden "uitgefilterd" van de modules ←
    gekozen via use-modules, hierboven.
  # In dit geval, zijn *zowel* extragear/graphics/kipi-plugins en
  # extragear/graphics/kipi-plugins/kipi-plugins-docs genegeerd
  ignore-modules extragear/graphics/kipi-plugins
end module-set
```

2.7 Samenvatting Grondbeginselen

Dit zijn de belangrijkste mogelijkheden en ideeën die u nodig heeft om te starten met kdesrc-build

Voor meer informatie, kunt u blijven lezen in deze documentatie. Met name zijn de [lijst met opties voor de commandoregel](#) en de [tabel met opties voor in het configuratiebestand](#) belangrijk.

De KDE Community onderhoud ook [een Wiki reference op het internet over hoe u de broncode kunt bouwen](#), met beschrijvingen van kdesrc-build en tips en andere richtlijnen over hoe u dit hulpprogramma kunt gebruiken.

Hoofdstuk 3

Mogelijkheden van het script

3.1 Overzicht van de mogelijkheden

kdesrc-build heeft de volgende mogelijkheden:

- U kunt 'net doen als of' u de operaties uitvoert. Als u `--pretend` of `-p` op de commandoregel meegeeft, dan zal het script een uitvoerige omschrijving van de commando's geven die het gaat uitvoeren, zonder ze daadwerkelijk uit te voeren. Maar als u nog nooit kdesrc-build heeft uitgevoerd, dan wilt u waarschijnlijk eerst het commando `kdesrc-build --metadata-only` uitvoeren voordat u `--pretend` uitvoert.

TIP

Als u een nog uitvoeriger omschrijving wilt hebben van wat kdesrc-build aan het doen is, dan kunt u het gebruik van de optie `--debug` uitproberen.

- kdesrc-build kan (met assistentie van de KDE FTP server) de checkouts van sommige Subversion modules versnellen. Als de module die u wilt uitchecken al is ingepakt in de website, dan zal kdesrc-build de snapshot downloaden en het voor gebruik op uw computer gereedmaken.

TIP

In het algemeen zijn er geen speciale voorbereidingen nodig om de eerste checkout van een Git module uit te kunnen voeren, omdat de hele Git repository toch moet worden gedownload, het is dus voor de server makkelijk vast te stellen wat het moet versturen.

Dit is voor u sneller, en helpt om de belasting van de kde.org anonymous Subversion servers te verminderen.

- Een andere manier van sneller werken krijgt u door het bouwproces van een module onmiddellijk te starten nadat de broncode van die module is gedownload. (beschikbaar vanaf versie 1.6)
- Uitgebreide ondersteuning voor het bouwen van de Qt™ library (in het geval dat de KDE software die u probeert te bouwen, afhankelijk is van een recente Qt™ die beschikbaar is bij uw distributie).
- kdesrc-build vereist niet dat een GUI aanwezig is om te kunnen werken. Daardoor kunt u KDE software bouwen zonder dat u een alternatieve grafische omgeving nodig heeft.

- Ondersteunt het instellen van standaard opties voor alle modules (zoals de compilatie-instellingen of de configuratie-instellingen). Dergelijke opties kunt u normaal gesproken ook voor specifieke modules instellen.

Daarnaast zal kdesrc-build indien van toepassing [standaard flags toevoegen](#) om het u makkelijker te maken en mogelijke typefouten te voorkomen als u ze zelf invoert. Opmerking: dit is niet van toepassing wanneer een (aangepaste) toolchain is geconfigureerd via bijv.: [cmake-toolchain](#)

- kdesrc-build kan een specifieke [branch of tag](#) van een module uitchecken. U kunt daarmee ervoor zorgen dat een specifieke [revisie](#) van een module is uitgecheckt.
- kdesrc-build kan automatisch van broncoderepository omschakelen om van een andere repository, branch, of tag uit te checken. Dit gebeurt automatisch als u een optie wijzigt die invloed heeft op wat de repository URL zou moeten zijn, maar u moet de optie `--src-only` gebruiken op het kdesrc-build te laten weten dat het acceptabel is om de switch uit te voeren.
- kdesrc-build kan [alleen gedeeltes van een module uitchecken](#), in die situaties dat u alleen een programma uit een grote module nodig heeft.
- Voor ontwikkelaars: kdesrc-build zal [u er aan herinneren](#) als u `svn+ssh://` gebruikt terwijl ssh-agent niet is opgestart, omdat dit er toe zal leiden tot herhaaldelijke verzoeken om het wachtwoord van SSH.
- Kan [de build-map verwijderen](#) van een module na een installatie om ruimte te besparen, ten koste van toekomstige compilatie tijd.
- De locaties van door kdesrc-build gebruikte mappen zijn instelbaar (zelfs per module).
- Kan Sudo gebruiken, of een andere user-specified commando voor het [installeren van modules](#) zo dat kdesrc-build niet als de super user hoeft te worden opgestart.
- kdesrc-build start standaard [met lagere prioriteit](#) zodat u uw computer nog steeds kan gebruiken terwijl kdesrc-build aan het werk is.
- Kan gebruik maken van KDE's [tags en branches](#).
- Het is mogelijk om [een bouw te vervolgen](#) van een bepaalde module. U kunt zelfs tijdelijk [enkele modules negeren](#) voor een bepaalde bouw.
- kdesrc-build zal de [voortgang van uw bouwprocestonen](#) als het CMake gebruikt, en zal altijd de tijd opnemen van het bouwproces zodat u later weet hoelang het heeft geduurd.
- Komt met een handige set van standaardinstellingen geschikt om een basis KDE installatie voor een enkele gebruiker te bouwen vanuit anonieme broncoderepositories.
- Tilde-expansie voor uw configuratie-opties. U kunt bijvoorbeeld opgeven:

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

- Stelt automatisch een bouwstrategie op, waarbij de broncode-map niet dezelfde is als de bouwmap, om de broncode-map maagdelijk te houden.
- U kunt globale opties opgeven die geldig is voor elke module die u uitcheckt, maar u kunt ook opties opgeven die voor individuele modules gelden.
- Forceer volledige rebuilds, als u kdesrc-build opstart met de optie `--refresh-build`.
- U kunt verschillende omgevingsvariabelen voor de bouw opgeven, waaronder `KDEDIR`, `QTDIR`, `DO_NOT_COMPILE`, en `CXXFLAGS`.
- Commando logging. Logs zijn gedateerd en genummerd zodat u altijd een log heeft van een script run. Daarnaast is er een speciale symlink genaamd `latest` die altijd naar de meest recente log entry in de log-map wijst.
- U kunt ook alleen een gedeelte van een KDE Subversion-module uitchecken. U kunt bijvoorbeeld alleen de taglib vankdesupport uitchecken.

3.2 Log van het bouwproces van kdesrc-build

3.2.1 Overzicht logging

Logging is een functionaliteit van kdesrc-build waarbij de uitvoer van elke commando die kdesrc-build uitvoert wordt opgeslagen voor later onderzoek, indien noodzakelijk. Dit is gedaan omdat vaak nodig is om de uitvoer van deze programma's te hebben als het bouwen is mislukt, op de eerste plaats omdat er zo veel verschillende redenen kunnen zijn waarom een bouwproces kan mislukken.

3.2.1.1 Indeling logging-map

De logs worden altijd opgeslagen in de log-map. De locatie van de log-map stelt u in met de optie `log-dir`, die standaard wijst naar `/${source-dir} /log` (waar `/${source-dir}` de waarde is van de optie `source-dir`). Hierna in deze sectie, wordt naar deze waarde gerefereerd als `/${log-dir}`.

In `/${log-dir}`, is een set van mappen, een voor elke keer dat kdesrc-build is opgestart. Elke map heeft als naam de datum, en een startnummer. Als voorbeeld, de tweede keer dat kdesrc-build is opgestart op 26 mei, 2004, creëert het een map genaamd `2004-05-26-02`, waar de `2004-05-26` voor de datum staat, en de `-02` is het startnummer.

Om het u gemakkelijk te maken, zal kdesrc-build ook een link naar de logs van uw laatste run creëren, genaamd `latest`. Zodat de logs van de meest recente kdesrc-build run altijd te vinden zijn onder `/${log-dir} /latest`.

Nu zal in elke map van een kdesrc-build run zelf een set mappen aanwezig zijn, een voor elke KDE module dat kdesrc-build probeert te bouwen. Ook zal er een bestand genaamd `build-status` in de map aanwezig zijn, waarmee u kunt bepalen bij welke modules het gelukt is en bij welke het mislukt is om ze te bouwen.

OPMERKING

Als een module zelf weer een submodule (zoals `extragear/multimedia`, `playground/utils`, of `KDE/kdelibs`) heeft, dan zal er een vergelijkbare layout in de log map aanwezig zijn. Bijvoorbeeld, de logs voor `KDE/kdelibs` zullen na de laatste kdesrc-build run te vinden zijn in `/${log-dir} /latest/KDE/kdelibs`, en niet in `/${log-dir} /latest/kdelibs`.

In elke module log-map, vindt u een set bestanden voor elke operatie die kdesrc-build uitvoert. Als kdesrc-build een module update, dan zal bestanden zien zoals `svn-co.log` (voor een module checkout) of `svn-up.log` (bij het updaten van een module dat al uitgecheckt is). Als het commando `configure` is opgeroepen, dan kunt u een `configure.log` in die map verwachten.

Als er een fout is opgetreden, dan zou u in een van de bestanden moeten kunnen zien waardoor de fout is opgetreden. Om u te helpen bij het bepalen in welk bestand de foutmelding is te vinden, zal kdesrc-build een link creëren naar het bestand met de foutmelding (zoals `build-1.1.log` genaamd `error.log`).

Het uiteindelijke doel van dit alles is om te zien waarom een module na uw laatste kdesrc-build heeft gefaald om te bouwen, het bestand waar u als eerste naar moet kijken is `/${log-dir} /latest/ module-name /error.log`.

TIP

Als het bestand `error.log` leeg is (met name na een installatie), dan was er misschien geen fout. Sommige van de hulpprogramma's die het KDE bouwsysteem gebruikt, kan soms bij vergissing een fout rapporteren terwijl er geen een was.

Het kan ook gebeuren dat sommige commando's de kdesrc-build's uitvoer redirection ontwijken en het logbestand in bepaalde omstandigheden bypassen (meestal bij het uitvoeren van de eerste Subversion checkout), en is in dat geval de foutmelding niet in het logbestand maar in plaats daarvan in de Konsole of de terminal waarin u kdesrc-build opstartte.

Hoofdstuk 4

De kdesrc-build configureren

4.1 Overzicht van de kdesrc-build configuratie

Om het script te gebruiken, moet u een bestand in uw thuismap hebben genaamd `.kdesrc-buildrc`, welke beschrijft welke modules u wilt downloaden en bouwen, en welke opties of configuratiegegevens u voor deze modules wilt gebruiken.

4.1.1 Indeling van het configuratiebestand

4.1.1.1 Globale configuratie

Het configuratiebestand start met de globale optie, die als volgt is gespecificeerd:

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

4.1.1.2 Module configuratie

Het wordt dan gevolgd door een of meer module secties, gespecificeerd in een van de volgende twee vormen:

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

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

# Andere opties zijn ook instelbaar als
option-name option-value
[...]
end module-set
```

BELANGRIJK

Merk op dat de tweede vorm, module sets, *alleen werkt bij Git-gebaseerde modules*.

Bij Subversion modules, moet *module-name* een module zijn uit de KDE Subversion repository (bijvoorbeeld, kdeartwork of kde-wallpapers), ofschoon het mogelijk is om dit te omzeilen als u [handmatig de Subversion URL specificeert](#).

Bij Git modules, kunt u de module naam naar wens kiezen, zolang het geen duplicaat is van een andere module naam in de configuratie. Vergeet niet dat de layout van de broncode-map en de bouwmap gebaseerd zijn op de modulenaam als u de optie [dest-dir](#) niet gebruikt.

Echter, bij Git *module sets* moeten de *module-names* overeenkomen met de daadwerkelijke git modules in de gekozen repository. Zie [git-repository-base](#) of [use-modules](#) voor meer informatie.

4.1.1.3 Verwerken van optiewaarden

In het algemeen wordt de gehele inhoud van de regel na de *optienaam* gebruikt als de *optiewaarde*.

Eén wijziging die kdesrc-build uitvoert is dat een reeks `#{naam-van-optie}` vervangen wordt door de waarde van die optie uit de globale configuratie. Dit biedt u om naar de waarde van bestaande opties te refereren, inclusief opties die al ingesteld zijn door kdesrc-build.

Om een voorbeeld van dit gebruik te zien, zie [Voorbeeld 2.1](#).

4.1.1.4 'options' modules

Er is tenslotte een laatste type entry in een configuratiebestand, *options groups*, welke u kunt gebruiken waar een *module* of *module-set* kan worden gebruikt.

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

In een *options* group kunt u opties ingesteld hebben net zoals bij een module declaratie, en hoort bij een bestaande module. De hierin ingestelde opties *overschrijven* de in de bijbehorende module ingestelde opties.

BELANGRIJK

De bijbehorende module naam *moet* overeenkomen met de opgegeven naam in de *options* declaratie. Let op het mistypen van de naam.

Dit is handig bij het declareren van een complete *module-set* van modules, die allemaal dezelfde opties gebruiken, om daarna *options groups* te gebruiken voor individuele wijzigingen.

options groups kunt u ook toepassen op benoemde *module sets*. Hierdoor kunnen expert gebruikers een algemene configuratiebestand (waarin *module-set* declaraties voorkomen) als startpunt gebruiken, en vervolgens wijzigingen op de opties toepassen die door die *module-sets* in configuratiebestanden worden gebruikt die het commando *include* gebruiken om naar het basisconfiguratie te refereren.

Example 4.1 Voorbeeld van het gebruik van options

In dit voorbeeld kiezen we ervoor om alle modules uit de KDE multimedia software groepen te bouwen. Maar we willen verschillende versies van het programma KMix (misschien voor het testen van een bug fix) gebruiken. Dat gaat dan als volgt:

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

# kmix is een onderdeel van kde/kdemultimedia groep, ook als we
# kmix nooit eerder in dit bestand hebben genoemd, zal kdesrc-build de ←
  wijziging ontdekken.
options kmix
  branch KDE/4.12
end options
```

Wanneer u nu kdesrc-build wilt uitvoeren, dan zullen alle KDE multimedia programma's worden gebouwd vanuit de 'master' branch van de broncoderepository, maar KMix zal worden gebouwd van de oudere 'KDE/4.12' branch. Door options te gebruiken, hoeft u niet individueel alle *andere* KDE multimedia programma's op te noemen om ze de juiste branch optie te geven.

OPMERKING

Merk op dat deze mogelijkheid pas beschikbaar is in kdesrc-build vanaf versie 1.16, of als u de development versie van kdesrc-build gebruikt van na 2014-01-12.

4.1.2 Naar andere configuratiebestanden verwijzen

In een configuratiebestand, mag u verwijzen naar andere bestanden door het keyword `include` met een bestand te gebruiken, zodat het zal reageren alsof het bestand waarnaar is verwezen, vanaf dat punt in het configuratiebestand is geplakt.

Bijvoorbeeld, u kunt zoiets als het volgende hebben gedaan:

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

  # Voeg specifieke opties hier toe.

end global
```

OPMERKING

Als u niet het volledige pad opgeeft naar het bestand waarvan u een verwijzing wilt opnemen, dan zal naar het bestand worden gezocht vanuit de map waarin het bestand met de broncode staat. Dit werkt ook recursief.

4.1.3 De belangrijkste configuratieopties

Hier volgt een lijst met de belangrijkste opties. klik op de optie om meer erover te lezen. Voor de volledige lijst met opties, zie Section 4.2.

- [cmake-options](#) voor het definiëren welke flags te gebruiken bij het configureren van een module bij het gebruik van CMake.
- [branch](#), voor het uitchecken van een branch in plaats van /trunk (bij Subversion) of master (bij Git).
- [configure-flags](#) voor het definiëren welke flags te gebruiken bij het configureren van Qt™.
- [kdedir](#), voor het instellen naar welke map KDE wordt geïnstalleerd.
- [make-options](#), voor het instellen welke opties naar het Make programma (zoals hoeveel te gebruiken CPU's) worden doorgegeven.
- [qtdir](#), voor het instellen van het pad naar Qt™.
- [source-dir](#), voor het wijzigen van waar de broncode naar toe moet worden gedownload.

4.2 Tabel met beschikbare configuratieopties

Hieronder is een tabel met de verschillende opties, met de volgende informatie:

- Naam van optie
- Een beschrijving van hoe kdesrc-build reageert als de optie is ingesteld in zowel de global section, en de module section van het [configuratiebestand](#) bij het bouwen van een module.
- Speciale opmerkingen over het doel en het gebruik van de optie.

Naam optie	Module -> Globaal gedrag	Opmerkingen
apidox		Deze optie werd gebruikt voor het bouwen van de KDE module API documentatie. Het is verwijderd in kdesrc-build 1.6.3 omdat het niet werd gebruikt. Online API documentatie is beschikbaar in kde.org . Daarnaast is mogelijk om KDE API documentatie te bouwen met de hulp van het kdedoxygen.sh script dat is meegeleverd met de kde-dev-scripts module. Zie KDE TechBase voor meer details.
apply-qt-patches		Deze optie is verwijderd in kdesrc-build 1.10. Om hetzelfde effect te krijgen, zie Section 6.2.1 en de optie repository .

Handboek scripts voor kdesrc-build

async	Is niet overschrijfbaar	<p>Deze optie maakt een asynchrone operatiemodus mogelijk, waar het parallel uitvoeren van de update van de broncode en het bouwproces mogelijk is, in plaats dat er gewacht op dat alle broncode geupdated is voordat het bouwproces start. De standaardinstelling van deze optie is dat de asynchrone modus ingeschakeld is. Om dit uit te schakelen, stelt u deze optie in op false</p> <p>Deze optie is vanaf de 1.6 release beschikbaar.</p>
binpath	Instelling bij Module overschrijft de instelling bij global	<p>Gebruik deze optie om de omgevingsvariabele PATH in te stellen voor tijdens het bouwen. U kunt deze instelling niet met een module-optie overschrijven. De standaardinstelling is de \$PATH dat is ingesteld als het script start. Deze omgevingsvariabele zou ook de door dubbele punt (:) gescheiden paden van uw development toolchain moeten hebben. De paden \$KDEDIR /bin en \$QTDIR /bin worden automatisch toegevoegd. U mag in elk pad dat u toevoegt de tilde (~) gebruiken.</p>

branch	Instelling bij Module overschrijft de instelling bij global	<p>Stel deze optie in om een checkout te doen van een branch van KDE in plaats van de standaard <i>master</i> (bij Git modules) of <i>trunk</i> (bij Subversion), waar de KDE ontwikkeling gebeurt. Bijvoorbeeld, om de KDE 4.6 branch uit te checken, zou u deze optie in willen stellen op <code>4.6</code>.</p> <p>Als het kdesrc-build niet lukt om met deze optie een branch goed te downloaden, dan moet u wellicht handmatig de URL opgeven waarvan te downloaden met behulp van de opties <code>module-base-path</code> of <code>override-url</code>.</p> <div data-bbox="997 982 1338 1224" style="border: 1px solid black; padding: 5px;"><p>OPMERKING Bij de meeste KDE modules wilt u waarschijnlijk in plaats daarvan de optie <code>branch-group</code> gebruiken en deze optie gebruiken voor de uitzonderingen.</p></div>
--------	---	--

branch-group	Instelling bij Module overschrijft de instelling bij global	<p>Gebruik deze optie om een algemene groep in te stellen waarvan u de modules wilt selecteren.</p> <p>Voor de ondersteunde Git module types, zal kdesrc-build bepalen welke branch automatisch daadwerkelijk gebruikt zal worden, gebaseerd op door de KDE ontwikkelaars bepaalde regels (deze regels zijn te bekijken in de <code>kde-build-metadata</code> broncoderepository in uw broncode-map). Nadat een branch is gekozen, zal die branch worden gebruikt alsof u het zelf had opgegeven met de optie branch.</p> <p>Dit is handig als u alleen maar probeert om bij te blijven met sommige gewone ontwikkelingen zonder dat u aandacht hoeft te geven aan alle naamswijzigingen van branches.</p> <p>De huidige branch groups (op of 2013-08-11) zijn:</p> <ul style="list-style-type: none">• <code>stable-qt4</code>, voor het volgen van bugfixes van de op Qt™ 4-gebaseerde KDE bibliotheken en programma's.• <code>latest-qt4</code>, voor de huidige ontwikkelingen en nieuwe mogelijkheden van de Qt™ 4-gebaseerde KDE bibliotheken en programma's.• <code>kf5-qt5</code>, voor het volgen van 'bleeding-edge' ontwikkelingen van de komende Qt™ 5-gebaseerde KDE Frameworks 5, Plasma Workspace 2, etc. <p>Merk op dat als u <i>daadwerkelijk zelf</i> een branch kiest, dat het deze instelling overschrijft. Datzelfde geldt ook voor andere branch-selectie</p>
--------------	---	---

<p>build-dir</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Gebruik deze optie om de map te wijzigen waarin het compilatieresultaat geplaatst wordt. Er zijn drie verschillende manieren om het te gebruiken:</p> <ol style="list-style-type: none"> 1. Relatief ten opzichte van de KDE Subversion map met broncode (zie de optie source-dir). Deze is de standaard, en wordt geselecteerd als u een naam van een map invoert die niet een tilde (~) of een slash (/) begint. De standaard waarde is <code>build</code>. 2. Absoluut pad. Als u een pad specificeert dat begint met een /, dan wordt dat pad in deze vorm gebruikt. Bijvoorbeeld, <code>/tmp/kde-obj-dir/</code> . 3. Relatief ten opzichte van uw thuismap. Als u een pad specificeert dat begint met een ~, dan is het opgegeven pad relatief ten opzichte van uw thuismap, overeenkomend met de tilde-expansie in het terminalvenster. Bijvoorbeeld, <code>~/builddir</code> zal de bouwmap instellen naar <code>/home/user-name/builddir</code>. <p>Verrassend misschien, maar deze optie is per module instelbaar.</p>
------------------	--	--

<p>build-when-unchanged</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Gebruik deze om in te stellen of kdesrc-build altijd probeert om een module te bouwen die geen broncode update heeft gehad.</p> <p>Door <code>build-when-unchanged</code> naar true in te stellen, zal kdesrc-build altijd het bouwproces voor een module proberen, zelfs als er voor de module geen updates van de broncode zijn. Dit is de standaardinstelling omdat dit een grotere kans geeft op een correcte bouw.</p> <p>Door <code>build-when-unchanged</code> naar false in te stellen, zal kdesrc-build alleen het bouwproces voor een module proberen als er voor de module updates van de broncode zijn, of in andere situaties waar het waarschijnlijk is dat opnieuw bouwen daadwerkelijk nodig is. Dit kan tijd besparen, met name als u kdesrc-build dagelijks, of nog vaker uitvoert.</p> <div data-bbox="997 1352 1341 1745" style="border: 1px solid black; padding: 5px;"><p>BELANGRIJK Deze mogelijkheid is er alleen als optimalisatie. Zoals zoveel andere optimalisaties, is er een afweging wat betreft de correctheid van uw installatie. Bijvoorbeeld, bij wijzigingen van de qt of kdelibs modules kan een herbouw van andere modules nodig zijn, zelfs als de broncode daarvan helemaal niet is veranderd.</p></div>
-----------------------------	--	--

<p>checkout-only</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Gebruik deze optie om de broncode in Subversion stuk voor stuk uit te checken. De waarden voor deze optie moet een door spaties gescheiden lijst zijn van de mappen die u wilt uitchecken. Al hoe wel deze optie de global optie overschrijft, is het gebruiken van deze instelling als global option niet zinvol. Merk op dat deze instelling niet werkt bij Git modules vanwege de werking van het Git systeem. Zie Section 6.3.1.1 voor een voorbeeld.</p>
<p>cmake-generator</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Deze optie gebruiken om te specificeren welke generator samen met CMake te gebruiken. Op dit moment worden beiden Ninja en Unix Makefiles evenals extra generators op hen gebaseerd zoals Eclipse CDT4 - Ninja ondersteund. Ongeldige (niet-ondersteunde) waarden worden genegeerd en behandeld alsof ze niet zijn ingesteld. Indien niet ingesteld zal Unix Makefiles als standaard worden gebruikt. Merk op dat als een geldige generator ook is gespecificeerd via cmake-options het de waarde voor <code>cmake-generator</code> zal overschrijven.</p>

cmake-toolchain	Instelling bij Module overschrijft de instelling bij global	<p>Deze optie gebruiken om een toolchain-bestand te specificeren om te gebruiken met CMake. Wanneer een geldig toolchain-bestand is geconfigureerd zal <i>kdesrc-build</i> <i>niet langer automatisch omgevingsvariabelen instellen</i>. U kunt set-env, binpath en libpath gebruiken om de omgeving handmatig bij te werken als uw toolchain-bestand niet zomaar werkt met <i>kdesrc-build</i>. Kijk in het overzicht van standaard vlaggen toegevoegd door kdesrc-build voor meer informatie.</p> <p>Merk op dat als een geldige toolchain ook is gespecificeerd via cmake-options het de waarde voor <code>cmake-toolchain</code> zal overschrijven.</p>
-----------------	---	---

<p>cmake-options</p>	<p>Voegt toe aan de global options voor het standaard bouwproces, overschrijft global voor andere bouwprocessen.</p>	<p>Gebruik deze optie om op te geven welke flags aan CMake worden doorgegeven bij het creëren van het bouwsysteem voor de module. Als u dit gebruikt als een global optie, dan wordt het toegepast bij alle modules dat dit script bouwt. Als u dit als een module optie gebruikt, dan wordt het aan het eind van de global opties toegevoegd. Hierdoor kunt u algemene CMake opties in het global sectie specificeren. Deze optie is niet van toepassing op qt (wat CMake niet gebruikt). Gebruik in plaats daarvan configure-flags. Als een geldige generator is gespecificeerd onder de gegeven opties het de waarde van cmake-generator zal overschrijven. Ongeldige (niet-ondersteunde) generators worden genegeerd en zullen niet worden doorgegeven aan CMake. Als een geldige toolchain-bestand is gespecificeerd onder de gegeven opties het de waarde van cmake-toolchain zal overschrijven. Ongeldige (niet-ondersteunde) toolchains worden genegeerd en zullen niet worden doorgegeven aan CMake. Omdat deze opties direct worden doorgegeven naar CMake commandoregel, moet u ze opgeven alsof u intikt bij CMake. bijvoorbeeld:</p> <pre>cmake-options -DCMAKE_BUILD_TYPE=RelWithDebInf</pre> <p>Omdat dit altijd een probleem is, neemt kdesrc-build de zorg op zich om er voor te zorgen</p>	
----------------------	--	--	--

Handboek scripts voor kdesrc-build

colorful-output	Is niet overschrijfbaar	Stel deze optie in op false om de gekleurde uitvoer van kdesrc-build uit te schakelen. De standaardinstelling van deze optie is <i>true</i> . Merk op dat kdesrc-build de kleurcodes alleen naar het terminalvenster (zoals xterm, Konsole, of de normale Linux [®] console) uitvoert.
compile-commands-export	Instelling bij Module overschrijft de instelling bij global	Schakelt de generatie van een <code>compile_commands.json</code> in via CMake in de bouwmap. Deze optie is standaard <i>true</i> , zet het op false om dit gedrag uit te schakelen.
compile-commands-linking	Instelling bij Module overschrijft de instelling bij global	Schakelt het aanmaken van een symbolische koppeling in van <code>compile_commands.json</code> in gegenereerd via CMake in de bouwmap naar de broncodemap. Deze optie is standaard <i>false</i> , zet het op true om het maken van de symbolische koppeling in te schakelen.
configure-flags	Voegt toe aan de global options voor het standaard bouwproces, overschrijft global voor andere bouwprocessen.	Gebruik deze optie om op te geven welke flags naar <code>./configure</code> moeten worden doorgegeven bij het creëren van het bouwsysteem van de module. Als u dit gebruikt als een global-optie, dan is het van toepassing op alle modules dat dit script bouwt. <i>Deze optie is alleen van toepassing op qt.</i> Voor het wijzigen van configuratiegegevens voor KDE modules, zie cmake-options .

<p>custom-build-command</p>	<p>Instelling bij Module overschrijft de instelling bij global (build system optie)</p>	<p>U kunt deze optie zo instellen dat het een ander commando uitvoert (ander dan make, bijvoorbeeld) in om het bouwproces uit te voeren. kdesrc-build zou in het algemeen de juiste actie uit moeten voeren, daarom zou het niet nodig moeten zijn om deze optie in te stellen. Maar het kan handig zijn om alternatieve bouwsystemen te gebruiken. De inhoud van deze optie wordt als commando op de commandoregel gebruikt, normaal gesproken gemodificeerd door de optie make-options.</p>
<p>cxxflags</p>	<p>Voegt toe aan de global options voor het standaard bouwproces, overschrijft global voor andere bouwprocessen.</p>	<p>Gebruik deze optie om te specificeren welke flags gebruikt worden voor het bouwen van de module. Deze optie wordt hier gespecificeerd en niet bij configure-flags of cmake-options omdat deze optie de omgevingsvariabele <code>CXXFLAGS</code> instelt tijdens het bouwproces. Noot: merk op dat voor KDE 4 en elke andere modules die gebruik maken van CMake, het nodig is om de optie <code>CMAKE_BUILD_TYPE</code> in te stellen op none bij het configureren van de module. Dit kunt u doen door de optie cmake-options te gebruiken.</p>

Handboek scripts voor kdesrc-build

dest-dir	Instelling bij Module overschrijft de instelling bij global	Gebruik deze optie om de naam van een module gebruikt op de harde schijf te wijzigen. Bijvoorbeeld, als uw module was extragear/network, dan zou u het met deze optie kunnen hernoemen naar extragear-network. Merk wel op dat ofschoon dit wel de naam op de harde schijf wijzigt, het geen goed idee is om mappen of map-scheidingen in de naam op te nemen omdat dit interfereert met de build-dir of source-dir opties.
disable-agent-check	Is niet overschrijfbaar	Als u normaal gesproken SSH gebruikt om de broncode uit Subversion te downloaden (als of u het svn+ssh protocol gebruikt), dan zal kdesrc-build er voor zorgen dat als u ssh-agent gebruikt, het daadwerkelijk enkele SSH-identiteiten gebruikt. Dit is om te voorkomen dat SSH voor elke module weer om uw wachtwoord vraagt. U kunt deze controle uitschakelen door <code>disable-agent-check</code> in te stellen op true .

Handboek scripts voor kdesrc-build

do-not-compile	Instelling bij Module overschrijft de instelling bij global	<p>Gebruik deze optie om een specifieke set van mappen op te geven die niet in een module gebouwd moeten worden (in plaats van allemaal). De niet te bouwen modules moeten door spaties gescheiden zijn.</p> <p>Merk op dat de broncode voor de programma's toch gewoon worden gedownload. U kunt de optie checkout-only gebruiken voor het selecteren van de mappen die u wilt uitchecken.</p> <p>Bijvoorbeeld, om JuK en KsCD in de kdemultimedia module uit te sluiten van compilatie, voegt u <code>~do-not-compile juk kscd</code> aan uw kdemultimedia instellingen toe.</p> <p>Zie Section 6.3.1.2 voor een voorbeeld.</p>
email-address	Is niet overschrijfbaar	Deze optie is verwijderd in kdesrc-build 1.14.
email-on-compile-error	Is niet overschrijfbaar	Deze optie is verwijderd in kdesrc-build 1.14.
inst-apps		Deze optie is verwijderd in versie 1.10

<p>git-desired-protocol</p>	<p>Is niet overschrijfbaar</p>	<p>Deze optie is alleen van toepassing op modules van een KDE project repository. Wat deze optie eigenlijk doet is het instellen van welk netwerk-protocol de voorkeur heeft bij het opsturen van de broncode voor deze modules. Normaal gesproken wordt het zeer efficiënte <code>git</code> protocol gebruikt, maar bij sommige netwerken (b.v. intranetten van bedrijven, publieke Wi-Fi) kan deze geblokkeerd worden. Een alternatief protocol welke veel beter wordt ondersteund is het <code>https</code> protocol gebruikt voor Internet web sites. Als u een van deze beperkende netwerken moet gebruiken dan kunt u deze optie op <code>http</code> instellen om aan <code>https</code> communicatie de voorkeur te geven.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>TIP U heeft wellicht ook de optie http-proxy nodig omdat ook een HTTP proxy nodig is voor netwerkverkeer.</p> </div> <p>In alle andere situaties zou u deze optie niet moeten instellen omdat het standaardprotocol het meeste efficiënt is. Deze optie is toegevoegd in kdesrc-build 1.16. Vóór 20.06 werd deze optie gebruikt om ophaal-URL te configureren in plaats van de opstuur-URL. Vanaf 20.06 wordt <code>https</code> altijd gebruikt bij het bijwerken van KDE-projecten.</p>
-----------------------------	--------------------------------	--

		<p>Deze optie, toegevoegd in versie 1.12.1, gebruikt u voor het creëren van een verkorte naam om naar een specifieke Git-repository gebaseerde URL te refereren in latere module set declaraties, wat handig is bij het snel declareren van veel te bouwen Git modules. U moet twee dingen (gescheiden door een spatie) specificeren: De naam waaraan de basis URL wordt toegekend en de actuele basis URL zelf. Bijvoorbeeld:</p> <pre>global # andere opties # Dit is het algemene pad naar alle anonieme git-repository-base kde-git kde: end global # Module declaraties module-set # Nu kunt u de alias gebruiken die u eerder # in een module-set. repository kde-git use-modules module1.git module2.git end module-set</pre>	
git-repository-base	Is niet overschrijfbaar	<p>De module-instelling <code>use-modules</code> optie creëerde twee modules intern, waardoor <code>kdesrc-build</code> zich gedraagt alsof het heeft ingelezen:</p> <pre>module module1 repository kde:module1.git end module module module2 repository kde:module2.git end module</pre> <p>Het hierboven gebruikte Git repository voorvoegsel <code>kde:</code> is een afkorting die automatisch door <code>kdesrc-build</code> wordt opgezet. Zie het artikel URL Renaming in de TechBase voor meer informatie. Merk</p>	

Handboek scripts voor kdesrc-build

<code>git-user</code>	Instelling bij Module overschrijft de instelling bij global	<p>Deze optie is bedoeld voor KDE-ontwikkelaars. Indien ingesteld, zal het gebruikt worden om automatisch identiteitsinformatie voor de besturingssoftware voor broncode Git in te stellen voor <i>nieuw gedownload</i> Git-modules (inclusief de grote meerderheid van KDE-modules). Specifiek, de velden voor de naam en email van de gebruiker voor elk nieuw Git repository moet u invullen bij deze optie. De waarde moet gespecificeerd worden in de vorm Naam gebruiker <email@voorbeeld.com>. Een ontwikkelaar genaamd 'Foo Barbaz' met het e-mailadres 'foo@abc.xyz' zou bijvoorbeeld gebruiken:</p> <pre>git-user Foo Barbaz <foo@abc.xyz></pre> <p>Deze optie was geïntroduceerd in kdesrc-build 15.09.</p>
-----------------------	---	--

Handboek scripts voor kdesrc-build

http-proxy	Instelling bij Module overschrijft de instelling bij global	<p>Deze optie, indien ingesteld, gebruikt de gespecificeerde URL als een proxy server voor elk HTTP netwerk-communicatie (bijvoorbeeld, voor het downloaden van snapshots voor nieuwe modules, of de KDE project database).</p> <p>Daarnaast zal kdesrc-build proberen om de hulpprogramma's waarvan het afhankelijk is, ook die proxy server te laten gebruiken, indien mogelijk, door de omgevingsvariabele <code>http_proxy</code> naar de aangegeven server te laten wijzen, <i>als die omgevingsvariabele niet al is ingesteld</i>.</p> <p>Deze optie was geïntroduceerd in kdesrc-build 1.16.</p>
ignore-kde-structure	Instelling bij Module overschrijft de instelling bij global	<p>Deze optie is verouderd en zal (opnieuw) verwijderd worden in een volgende release.</p> <p>Gebruik in plaats daarvan de <code>indeling-mappen</code>, wat meer mogelijkheden voor selectie biedt.</p>

indeling-mappen	Instelling bij Module overschrijft de instelling bij global	<p>Deze optie wordt gebruikt om de indeling die kdesrc-build zou moeten gebruiken bij het aanmaken van de broncode- en bouwmappen.</p> <p>Op dit moment zijn er drie mogelijke waarden: metadata, invent en flat</p> <p>De indeling flat zal alle modules direct onder het topniveau broncode en bouwmappen groeperen. Bijvoorbeeld <code>source/extragear/network/telepathy/ktp-text-ui</code> zou in de indeling metadata zijn en <code>source/ktp-text-ui</code> in plaats daarvan in de indeling flat.</p> <p>De indeling invent maakt een hiërarchie in mappen aan de de de relatieve paden van opslagruimten spiegelt op invent.kde.org. Bijvoorbeeld <code>source/kde/applications/kate</code> in de indeling metadata zou zijn <code>source/utilities/kate</code> in plaats daarvan in de indeling invent. Deze indeling heeft alleen effect op KDE projecten. Het is een goede keuze voor mensen die beginnen met kdesrc-build.</p> <p>Tenslotte, de indeling metadata is hetzelfde als het oude standaard gedrag. Deze indeling organiseert KDE projecten volgens de projectpaden gespecificeerd in de projectmetadata voor deze modules. Dit is een goede keuze als u een indeling van mappen wilt die bepaalde KDE-processen volgt, maar merk op dat dit pad daarom niet altijd stabiel is. Als resultaat kan kdesrc-build een oude kopie van de opslagruimte verlaten en een nieuwe klonen voor een project vanwege</p>
-----------------	---	--

<p>ignore-modules</p>	<p>Is niet overschrijfbaar</p>	<p>De in deze optie genoemde modules, welke door kdesrc-build worden geselecteerd wegens de optie <code>use-modules</code>, worden echter compleet overgeslagen. Gebruik deze optie als u een complete <code>kde-projects</code> project wilt bouwen <i>behalve</i> enkel specifieke modules. De instellingswaarde hoeft niet noodzakelijkerwijs direct de modulenaam te hebben. Alle modules waarvan het <code>KDE projects module pad</code> overeenkomt met een onderdeel van de optiewaarden zal overgesloten worden, zodat u op die manier meerdere modules kan overslaan. Bijvoorbeeld, een instellingswaarde met <code>libs</code> zal als resultaat hebben dat zowel <code>kde/kdegraphics/libs</code> en <code>playground/libs</code> worden overgeslagen worden (maar niet <code>kde/kdelibs</code> omdat het complete 'kdelibs' wordt vergeleken).</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>TIP Zie ook Voorbeeld 2.7.</p> </div> <p>Deze optie was geïntroduceerd in kdesrc-build 1.16.</p>
-----------------------	--------------------------------	---

<p>include-dependencies</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Deze optie zal, indien ingesteld op true aan kdesrc-build vragen om ook bekende afhankelijkheden van deze module tijdens het bouwen mee te nemen, zonder dat er van u verlangd wordt om deze afhankelijkheden te noemen (zelfs indirect).</p> <div data-bbox="997 653 1341 926" style="border: 1px solid black; padding: 5px;"> <p>OPMERKING Deze optie werkt alleen voor op kde-project-gebaseerde modules, en vereist dat de metadata die door KDE ontwikkelaars is ingevuld accuraat is voor de door u geselecteerde branch-group.</p> </div> <p>Deze optie is standaard ingeschakeld om het bouwen van toepassingen die recentere versies van Qt™ of Plasma nodig hebben dan ondersteund op gebruikelijke besturingssystemen.</p>
<p>install-after-build</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Met deze optie stelt u in of het pakket geïnstalleerd moet worden als de bouw succesvol is. Deze optie is standaard ingeschakeld. Als u dit wilt uitschakelen, dan moet u deze optie instellen op false in het configuratiebestand. U kunt ook op de commandoregel de parameter <code>--no-install</code> gebruiken.</p>

<p>install-environment-driver</p>	<p>Is niet overschrijfbaar</p>	<p>Standaard zal kdesrc-build een shell script installeren dat u kunt gebruiken in setup scripts voor een gebruikers-profiel om makkelijk de nodige omgevingsvariabelen in te stellen om de door kdesrc-build gebouwde Plasma desktop te gebruiken. Dit stuurprogramma zal de volgende bestanden veranderen:</p> <ul style="list-style-type: none"> • <code>\$XDG_CONFIG_HOME/kde-env-master.sh</code> (normaal gesproken gevonden als <code>~/.config/kde-env-master.sh</code>). • <code>\$XDG_CONFIG_HOME/kde-env-user.sh</code> (normaal gesproken gevonden als <code>~/.config/kde-env-user.sh</code>). <p>Het bestand <code>kde-env-user.sh</code> is optioneel. Het is bedoeld voor aanpassingen door de gebruiker (zie de sectie Troubleshooting and Debugging van de KDE UserBase voor voorbeelden van aanpasbare instellingen), maar deze instellingen zijn ook door de gebruiker elders in te stellen in zijn bestaande profiel setup scripts. U kunt deze mogelijkheid uitschakelen door deze optie in te stellen op <code>false</code>, en er voor zorg te dragen dat de optie install-session-driver ook is uitgeschakeld. Deze optie was geïntroduceerd in kdesrc-build 17.08.</p>
-----------------------------------	--------------------------------	--

TIP
kdesrc-build zal niet uw bestaande bestanden (indien aanwezig) overschrijven tenzij u ook de parameter `--delete-`

<p>install-session-driver</p>	<p>Is niet overschrijfbaar</p>	<p>Indien ingeschakeld, zal kdesrc-build proberen om een driver voor de grafische login manager te installeren zodat u kunt inloggen naar de door uw kdesrc-build-gebouwde KDE-desktop. Dit stuurprogramma zal de volgende bestanden veranderen:</p> <ul style="list-style-type: none"> • <code>~/.xsession</code> • <code>\$XDG_CONFIG_HOME/kde-env-master.sh</code> (normaal gesproken gevonden als <code>~/.config/kde-env-master.sh</code>). • <code>\$XDG_CONFIG_HOME/kde-env-user.sh</code> (normaal gesproken gevonden als <code>~/.config/kde-env-user.sh</code>). <p>Als u uw eigen login driver wilt blijven gebruiken, dan kunt u deze instelling uitschakelen door deze optie in te stellen op <i>false</i>. Indien ingeschakeld, zal deze mogelijkheid ook het onderdeel install-environment-driver inschakelen. Deze optie was geïntroduceerd in kdesrc-build 1.16.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>TIP kdesrc-build zal niet uw bestaande bestanden (indien aanwezig) overschrijven tenzij u ook de parameter <code>--delete-my-settings</code> op de commandoregel meegeeft.</p> </div>
-------------------------------	--------------------------------	---

<p>kdedir</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Met deze optie stelt u de map in waar KDE zal worden geïnstalleerd na het bouwproces. De standaardinstelling is <code>~/kde</code>. Als u dit wijzigt naar een map die root-rechten nodig heeft, dan wilt u wellicht meer lezen over de optie make-install-prefix.</p>
<p>kde-languages</p>	<p>Is niet overschrijfbaar</p>	<p>Met deze optie kunt u ervoor kiezen om localization pakketten samen KDE te downloaden en te installeren. U wilt dit waarschijnlijk doen als u niet in de Verenigde Staten leeft en KDE in uw eigen taal vertaald wilt gebruiken. Om deze optie te gebruiken, vult u hier een lijst in met te installeren talen gescheiden door spaties. Bij elke taal hoort een bijbehorende taalcode, die u kunt opzoeken op deze pagina: http://l10n.kde.org/teams-list.php. Het is geen probleem als u maar een taal selecteert. Standaard wordt er geen gedownload, wat inhoud dat KDE alles in Amerikaans Engels toont. Bijvoorbeeld, als u er voor kiest om de Franse taal te installeren, dan zou u deze optie kunnen instellen op zo iets als: kde-languages fr. U moet echter nog steeds Systeeminstellingen gebruiken om de Franse taal te selecteren.</p>

libpath	Instelling bij Module overschrijft de instelling bij global	Gebruik deze optie om de omgevingsvariabele <code>LD_LIBRARY_PATH</code> voor gebruik tijdens het bouwen in te stellen. U kunt in een module option deze instelling niet overschrijven. De standaardinstelling is blank, maar de paden <code>\$KDEDIR /lib</code> en <code>\$QTDIR /lib</code> worden automatisch toegevoegd. U mag in deze optie bij elk pad dat u toevoegt de tilde (~) gebruiken.
log-dir	Instelling bij Module overschrijft de instelling bij global	Gebruik deze optie om de map te wijzigen waarin de door het script gegenereerde logbestanden worden opgeslagen.
make-install-prefix	Instelling bij Module overschrijft de instelling bij global	Vul deze variabele met een door spaties gescheiden lijst, wat zal worden geïnterpreteerd als een commando en de daarbij horende opties die de voor het installeren van modules gebruikte commando make install voorafgaan. Dit is handig bijvoorbeeld bij het met Sudo installeren van pakketten, maar wees wel voorzichtig bij het gebruik van root privileges.
make-options	Instelling bij Module overschrijft de instelling bij global (build system optie)	Vul deze variabele met commandoregelparameters die u wilt doorgeven aan het commando make . Dit is handig bij programma's zoals <code>distcc</code> of bij systemen met meer dan een processor core. Merk op dat niet alle ondersteunde bouwsystemen make gebruiken. Voor bouwsystemen die ninja voor bouwen gebruiken (zoals het <code>Meson</code> bouwsysteem), zie de instelling <code>ninja-opties</code> .

Handboek scripts voor kdesrc-build

<p>manual-build</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Zet de instellingswaarde op true om te voorkomen dat het bouwproces gaat proberen om deze module te bouwen. Het blijft echter wel up-to-date bij het updating met Subversion. Deze optie is het exacte equivalent van de commandoregelparameter <code>--no-build</code>.</p>
<p>manual-update</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Zet de instellingswaarde op true om te voorkomen dat het bouwproces gaat proberen om deze module up te daten (en daarop volgend, bouwen of installeren). Als u dit voor een module instelt, dan heeft u het in essentie uitgecomment.</p>
<p>module-base-path</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Stel deze optie in om het standaard pad van kdesrc-build naar de betreffende module te overschrijven. Dit kunt u bijvoorbeeld gebruiken, om specifieke branches of getagde versies van libraries te downloaden. De KDE Source Viewer is onmisbaar bij het zoeken naar het juiste pad.</p> <p>Vergeet niet dat kdesrc-build het uiteindelijke pad samenstelt uit de volgende onderdelen: <code>\$svn-server /home/kde/ \$module-base-path.</code> De standaardwaarde is naar keuze <code>trunk/ \$module</code> of <code>trunk/KDE/ \$module</code>, afhankelijk van de module naam.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>TIP Gebruik in plaats daarvan waar mogelijk de opties branch of tag.</p> </div>

Handboek scripts voor kdesrc-build

<p>niceness</p>	<p>Is niet overschrijfbaar</p>	<p>Stel deze optie in op een getal tussen de 20 en de 0. Hoe hoger het getal is, hoe lager de prioriteit is die kdesrc-build voor zich zelf zal instellen, d.w.z. hoe hoger het getal, hoe "aardiger" het programma is. de standaardinstelling is 10.</p>
<p>ninja-options</p>	<p>Instelling bij Module overschrijft de instelling bij global (build system optie)</p>	<p>Stel deze variabele in om opdrachtregelopties door te geven aan het bouwcommando ninja. Dit kan nuttig zijn om 'verbose' uitvoer in te schakelen of om handmatig het aantal parallelle bouwjobs te verminderen die ninja zou gebruiken.</p> <div style="border: 1px solid black; padding: 5px;"> <p>OPMERKING Merk op dat deze instelling alleen ninja bestuurt bij gebruik door kdesrc-build. De module Qt™-'webengine' gebruikt ninja indirect, maar ondersteunt alleen officieel het bouwen door make. In deze situatie kunt u NINJAFLAGS instellen als een manier om make de toepasselijke vlaggen door te geven wanneer het later ninja aanroept, door make-options te gebruiken.</p> <pre> opties qtwebengine # Beperk make en ninja om niet meer dan 6 # zelfs wanneer meer CPU's beschikbaar zijn make-options -j6 NINJAFLAGS=-j6 einde opties </pre> </div>
<p>no-svn</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Als u deze optie op true heeft gezet dan zal kdesrc-build de broncode voor de module niet automatisch updaten. Maar het zal wel proberen om de module zoals altijd te bouwen.</p>

Handboek scripts voor kdesrc-build

no-rebuild-on-fail		Deze optie is verwijderd in versie 1.10, omdat dit gedrag niet meer behulpzaam is vanwege fixes in het onderliggende bouwsysteem.
aantal-kernen	Is niet overschrijfbaar	Deze optie wordt gedefinieerd door kdesrc-build (bij gebruik van het hulpmiddel kdesrc-build-setup van kdesrc-build --initial-setup), ingesteld om het aantal beschikbare CPU's te zijn (zoals aangegeven door de externe toepassing nproc). Als kdesrc-build het aantal CPU's niet kan detecteren, wordt deze waarde ingesteld op 4. Zie Voorbeeld 2.1 voor een voorbeeld van het gebruik van deze optie. Deze optie is toegevoegd in versie 20.07.

aantal-kernen-laag-geheugen	Is niet overschrijfbaar	<p>Deze optie is gedefinieerd door kdesrc-build (bij gebruik van het hulpmiddel kdesrc-build-setup of kdesrc-build --initial-setup), ingesteld om het aantal CPU's te zijn dat is geacht wordt veilig te zijn voor zwaargewicht of andere zeer intensieve modules, zoals qtwebengine, om te vermijden uit geheugen te lopen tijdens het bouwen. De typische berekening is één CPU kern voor elke 2 gigabytes (GiB) totaal geheugen. Minstens 1 kern zal gespecificeerd worden en niet meer dan aantal-kernen kernen zal gespecificeerd worden. Hoewel deze optie bedoeld is om Qt™-modules te ondersteunen kunt u het gebruiken voor elke module op dezelfde manier waarop aantal-kernen wordt gebruikt. Als kdesrc-build het beschikbare geheugen niet kan detecteren dan zal deze waarde ingesteld worden op 2. Deze optie is toegevoegd in versie 20.07.</p>
-----------------------------	-------------------------	--

<p>override-build-system</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Dit is een geavanceerde optie, toegevoegd in kdesrc-build 1.16. Normaal gesproken zal kdesrc-build na een download detecteren welk bouwsysteem het moet gebruiken voor een module. Dit wordt gedaan door te controleren op de aanwezigheid van bepaalde bestanden in de map met de broncode van de module. Sommige modules kunnen meerdere vereiste sets bestanden hebben, wat de auto-detectie in verwarring kan brengen. In een dergelijk geval kan u handmatig de correcte bouw type specificeren. De op dit moment ondersteunde bouwtypes die u kunt instellen zijn:</p> <p>KDE</p> <p>Gebruikt om KDE modules te bouwen. In werkelijkheid kunt u het gebruiken om bijna elke module te bouwen dat gebruik maakt van CMake maar het is het verstandigst om niet ervan afhankelijk te zijn.</p> <p>Qt</p> <p>Gebruikt om de Qt™ bibliotheek zelf te bouwen.</p> <p>qmake</p> <p>Gebruikt om Qt™ modules te bouwen die gebruik maken van qmake-style .pro-bestanden.</p> <p>generic</p> <p>Gebruikt om modules te bouwen die gebruik</p>
------------------------------	--	---

Handboek scripts voor kdesrc-build

<p>override-url</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Als u deze optie instelt, dan zal kdesrc-build de waarde daarvan gebruiken als de URL die doorgegeven wordt aan <i>Subversioncomplete</i> <i>ongewijzigd</i>. In het algemeen zal u dit willen gebruiken als u een specifieke release wilt downloaden en kdesrc-build niet kan vaststellen wat u bedoelt als u branch gebruikt.</p>
<p>persistent-data-file</p>	<p>Is niet overschrijfbaar</p>	<p>Gebruik deze optie om de locatie te wijzigen waar kdesrc-build zijn persistente data opslaat. De standaard is dat deze data wordt opgeslagen in een bestand genaamd <code>.kdesrc-build-data</code> geplaatst in de dezelfde map als waar het gebruikte configuratiebestand is. Als u in dezelfde map meerdere configuraties beschikbaar heeft, dan wilt u deze optie handmatig instellen zodat de verschillende configuraties niet eindigen met conflicterende persistente data. Deze optie is toegevoegd in kdesrc-build 1.15.</p>
<p>prefix</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Met deze optie stelt u in waar de module wordt geïnstalleerd (normaal wordt de instelling <code>kdedir</code> gebruikt). Door deze optie te gebruiken kunt u een module in een andere map installeren dan waar de KDE Platform libraries zijn geïnstalleerd, in het geval dat u kdesrc-build gebruikt om alleen programma's te bouwen. U kunt <code> \${MODULE} </code> of <code> \$MODULE </code> in het pad gebruiken om ze te expanderen tot de naam van de module.</p>

Handboek scripts voor kdesrc-build

<p>purge-old-logs</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Hiermee stelt u in of oude log mappen automatisch worden verwijderd. De standaardwaarde is <i>true</i>.</p>
<p>qmake-options</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Alle hier gespecificeerde opties worden doorgegeven aan het commando qmake, bij modules die het qmake bouwsysteem gebruiken. Bijvoorbeeld, u kunt de optie PREFIX=/path/to/qt gebruiken om bij qmake te overschrijven waar het de module installeert. Deze optie is toegevoegd in kdesrc-build 1.16.</p>
<p>qtdir</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Gebruik deze optie om de omgevingsvariabele <code>QTDIR</code> in te stellen voor bij het bouwen. Als u deze optie niet instelt, dan zal kdesrc-build aannemen dat Qt™ wordt geleverd door het besturingssysteem.</p>

<p>remove-after-install</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Als u weinig harde schijfruimte heeft, dan wilt u wellicht deze optie gebruiken om automatisch de bouwmap te verwijderen (of zowel de broncode-map als de bouwmap bij eenmalige installaties) nadat de module succesvol is geïnstalleerd. Mogelijke waarden voor deze optie zijn:</p> <ul style="list-style-type: none"> • none - Verwijder niets (Dit is de standaard). • builddir - verwijder de bouwmap, maar niet de broncode. • all - verwijder zowel de broncode en de bouwmap. <p>Merk op dat het gebruik van deze optie een ongunstige impact kan hebben op zowel het gebruik van uw bandbreedte (als u <i>all</i> gebruikt) en op de tijd die nodig is voor het compileren van KDE software, omdat kdesrc-build geen incrementele bouw kan uitvoeren.</p>
<p>repository</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Deze optie was geïntroduceerd bij versie 1.10, en kunt u gebruiken om de Git repository te specificeren waar u de broncode voor de module kunt downloaden. Qt™ (en dus ook qt) heeft de informatie uit deze optie nodig, net zoals verschillende KDE modules die in het proces van conversie naar het gebruik van Git zitten.</p>

<p>revision</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Als deze optie een andere waarde heeft dan 0 (nul), dan zal kdesrc-build de update van de broncode forceren met exact de opgegeven revisie, zelfs als opties zoals branch zijn ingesteld. als de module al de opgegeven revisie heeft dan zal het verder niet worden geupdate tenzij deze optie is gewijzigd of verwijderd uit de configuratie.</p> <div data-bbox="997 779 1338 957" style="border: 1px solid black; padding: 5px;"> <p>OPMERKING Deze optie werkte niet bij git-gebaseerde modules (inclusief kde-projects modules) tot kdesrc-build versie 1.16.</p> </div>
<p>run-tests</p>	<p>Instelling bij Module overschrijft de instelling bij global (build system optie)</p>	<p>indien ingesteld op true, dan zal de module worden gebouwd met ondersteuning voor gebruik van zijn test suite, en zal de test suite worden uitgevoerd als onderdeel van het bouwproces. kdesrc-build zal een eenvoudig rapport met de testresultaten tonen. Dit is handig voor ontwikkelaars en diegene die zeker willen zijn dat hun systeem correct is opgezet.</p>

set-env	Instelling bij Module overschrijft de instelling bij global	<p>Deze optie accepteert een door spaties gescheiden lijst met waarden, waar de eerste waarde de in te stellen omgevingsvariabele is, en de overige waarden is waarnaar toe u de variabele in wilt stellen. Bijvoorbeeld, om de variabele <code>RONALD</code> naar <code>McDonald</code> in te stellen, moet u in het betreffende sectie het volgende commando zetten:</p> <pre>set-env RONALD McDonald</pre> <p>Deze optie is speciaal omdat u het kan herhalen zonder dat u eerder in dezelfde sectie van het configuratiebestand ingestelde set-env instellingen overschrijft. Op deze manier kunt u meerdere omgevingsvariabelen per module (of globaal) instellen.</p>
source-dir	Instelling bij Module overschrijft de instelling bij global	<p>Deze gebruikt u om de map op uw computer in te stellen waar de KDE-broncode uit Subversion wordt opgeslagen. Als u deze waarde niet instelt, dan wordt de standaardinstelling <code>~/kdesrc</code> gebruikt. U mag bij deze optie de tilde (<code>~</code>) die de persoonlijke map voorstelt gebruiken.</p>
ssh-identity-file	Is niet overschrijfbaar	<p>Stel bij deze optie in welke private SSH key bestand wordt doorgegeven aan het commando <code>ssh-add</code> als kdesrc-build broncode gaat downloaden van repositories die authenticatie vereisen. zie ook: Section 6.4.1. Deze optie is toegevoegd in versie 1.14.2.</p>

Handboek scripts voor kdesrc-build

<p>stop-on-failure</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Stel de waarde van deze optie in op true om te zorgen dat de uitvoering van het script stopt als er een fout optreed tijdens het bouwproces of het installatieproces. Deze optie staat standaard op uit.</p>
<p>svn-server</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Deze optie gebruikt u om de server in te stellen die u gebruikt om van Subversion uit te checken. De standaardinstelling is de anonieme Subversion repository, <code>svn://anonsvn.kde.org/</code></p> <div data-bbox="997 842 1343 1087" style="border: 1px solid black; padding: 5px;"> <p>OPMERKING Als u programmeert voor KDE, gebruik dan de Subversion repository die aan u is toegewezen toen u uw developer account ontving, in plaats van de anonieme repository.</p> </div>
<p>tag</p>	<p>Instelling bij Module overschrijft de instelling bij global</p>	<p>Gebruik deze optie om een specifieke release van een module te downloaden. <i>Opmerking:</i> De kans is groot dat u deze optie <i>niet wilt</i> gebruiken. KDE releases zijn in tarball vorm vanaf de KDE download site beschikbaar.</p> <div data-bbox="997 1430 1343 1608" style="border: 1px solid black; padding: 5px;"> <p>OPMERKING Deze optie is alleen ondersteund voor op git gebaseerde modulen sinds kdesrc-build 1.16.</p> </div>

Handboek scripts voor kdesrc-build

use-clean-install	Instelling bij Module overschrijft de instelling bij global (build system optie)	<p>Stel de waarde van deze optie in op true om te zorgen dat kdesrc-build het commando make uninstall uitvoert direct voordat het make install uitvoert. Dit kan handig zijn om te zorgen dat er geen vergeten oude library-bestanden, CMake metadata, enz. meer zijn die problemen kunnen veroorzaken in lang-levende KDE installaties. Maar dit werkt alleen bij bouwsystemen waar ook make uninstall is. Deze optie is toegevoegd in kdesrc-build 1.12, maar bleef ongedocumenteerd tot kdesrc-build 1.16.</p>
use-cmake		<p>Deze optie is verwijderd in kdesrc-build 1.4 omdat alle KDE 4 modulen CMake vereisen en het gebruik van CMake niet is toegestaan bij alle andere modules.</p>
use-idle-io-priority	Is niet overschrijfbaar	<p>Deze optie, toegevoegd in kdesrc-build 1.12, zal een lagere prioriteit geven aan schijf en andere I/O gebruik, wat de responsiviteit van de rest van het systeem significant kan verbeteren ten koste van een beetje tragere uitvoering van kdesrc-build. Standaard is dit uitgeschakeld, voor een lagere prioriteit van de harde schijf stelt u dit in op true.</p>

use-modules

Is alleen te gebruiken in [module-set](#)

Met deze optie, toegevoegd in kdesrc-build 1.12.1, kunt u makkelijk meerdere te bouwen modules op een punt in [het configuratiebestand](#) specificeren.

Deze optie *moet* u gebruiken in een `module-set`. Elke via deze optie opgegeven identifier, is intern geconverteerd naar een kdesrc-build module, wordt met de `repository` optie van de `module-set`'s repository gecombineerd met de identifier-naam om de uiteindelijke repository te krijgen waarvan gedownload wordt. Alle andere opties opgegeven in de `module-set` worden ongewijzigd naar de gegenereerde modules doorgegeven.

De volgorde waarin die modules in deze optie zijn gedefinieerd is belangrijk, omdat dat ook de volgorde is waarmee kdesrc-build de gegenereerde modules zal behandelen bij het updaten, bouwen en installeren. Alle modules gedefinieerd in de `opgegevenmodule-set` zullen worden verwerkt voordat kdesrc-build naar de volgende module gaat na deze `module-set`.

Als u de opties voor een gegenereerde module moet wijzigen, declareer dan de module gewoon opnieuw nadat het is gedefinieerd in de `module-set`, en stel uw opties in zoals gewenst. Al hoe wel u op deze manier de opties voor de module wijzigt, zal de module gewoon in de `module-set` in de ingestelde volgorde worden geupdate en gebouwd (d.w.z. u kunt de bouwvolgorde niet op deze manier wijzigen).

Handboek scripts voor kdesrc-build

use-qt-build-dir-hack	Instelling bij Module overschrijft de instelling bij global	Deze optie is verwijderd vanwege verbeteringen in het Qt™ bouwsysteem.
use-stable-kde	Is niet overschrijfbaar	Deze optie is verouderd en zal (opnieuw) verwijderd worden in een volgende release. Gebruik in plaats daarvan de <code>branch-group</code> , wat meer mogelijkheden voor selectie biedt.

Tabel 4.1: Tabel met opties

Hoofdstuk 5

Commandoregel-opties en omgevingsvariabelen (Environment Variables)

5.1 Gebruik van de commandoregel

kdesrc-build is ontworpen om als volgt te worden gebruikt:

```
kdesrc-build [--opties...] [te bouwen modules...]
```

Als u op de commandoregel geen modules specificeert, dan zal kdesrc-build alle in de het configuratiebestand gedefinieerde modules bouwen, in de in dat bestand opgegeven volgorde (maar dit kan via verschillende bestand-opties gewijzigd worden).

5.1.1 Veel gebruikte commandoregel-opties

De complete lijst met opties voor op de commandoregel vindt u in Section 5.3. De meest gebruikte opties zijn:

--pretend (of -p)

Deze optie zorgt er voor dat kdesrc-build aangeeft welke acties het zou uitvoeren, zonder dat het ze daadwerkelijk uitvoert. Dit kan handig zijn om vast te stellen dat de modules die u denkt te gaan bouwen, ook daadwerkelijk zullen worden gebouwd.

--refresh-build

Deze optie forceert dat kdesrc-build de opgegeven modules van een absoluut vers startpunt bouwt. Elke bestaande bouwmap voor die module wordt verwijderd en opnieuw gebouwt. Deze optie is handig als er fouten optreden bij het bouwen van een module, en is soms nodig als Qt™ of KDE libraries wijzigen.

--no-src

Deze optie slaat het bijwerk-proces over. U kan dit gebruiken als u zeer recent de broncode heeft bijgewerkt (misschien deed u dit handmatig of gebruikte u recent kdesrc-build) maar u wilt toch sommige modules herbouwen.

--no-build

Deze optie is vergelijkbaar met `--no-src` hierboven, maar deze keer wordt het bouwproces overgeslagen.

5.1.2 Specificatie van te bouwen modules

In het algemeen is het specificeren van de te bouwen modules eenvoudig het opgeven van hun modulenaam zoals u het heeft gedefinieerd in het configuratiebestand. U kunt ook modules opgegeven die onderdeel zijn van een module set, zowel benoemd in [use-modules](#), als de naam van de hele module set zelf, als u deze een naam heeft gegeven.

In het specifieke geval van module sets gebaseerd op de [KDE project database](#), zal kdesrc-build de module-naam van van componenten uitbreiden om te bepalen welke module u precies wilt hebben. Bijvoorbeeld, kdesrc-build's KDE project entry bepaalt de locatie van het onderdeel in `extragear/utils/kdesrc-build`. U kunt elk van de volgende manieren van specificeren gebruiken om kdesrc-build te laten bouwen:

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

OPMERKING

De in het voorgaande voorbeeld gebruikte commando's gaan de module-naam vooraf, gescheiden door een `+`. Dit forceert dat de module-naam wordt geïnterpreteerd als een module uit de KDE project database, zelfs als die module niet in uw configuratiebestand is gedefinieerd.

Wees voorzichtig met het specificeren van zeer algemene onderdelen (bijv. `extragear/utils` zelf), omdat dit er toe kan leiden dat een groot aantal modules wordt gebouwd. U zou de optie `--pretend` kunnen gebruiken voordat u een nieuwe module set gaat bouwen om er zeker van te zijn dat het alleen de gewenste modules gaat bouwen.

5.2 Ondersteunde omgevingsvariabelen

kdesrc-build gebruikt geen omgevingsvariabelen. Als het nodig is dat u omgevingsvariabelen instelt voor het bouwproces of het installatieproces, lees dan eerst over de optie [set-env](#).

5.3 Ondersteunde commandoregelparameters

Het script accepteert de volgende commandoregelopties:

--async

Schakelt de [asynchronous modus](#) in, zodat de update van de broncode en het bouwproces van modules tegelijk kan gebeuren. Dit is de standaardinstelling, u hoeft deze optie alleen op te geven als u het in het configuratiebestand heeft uitgeschakeld.

--help

Toont voor dit script alleen een eenvoudige hulp.

--version (of -v)

Toont de programma-versie.

--show-info

Toont informatie over kdesrc-build en het besturingssysteem wat nuttig kan zijn in bugrapporten of bij vragen om hulp in forums of op e-maillijsten.

Beschikbare sinds versie 18.11.

--initial-setup

Laat kdesrc-build de eenmalige initiële instelling doen nodig om het systeem voor te bereiden om kdesrc-build te laten werken en om de nieuw geïnstalleerde KDE software uit te voeren.

Dit omvat:

- Bekende afhankelijkheden worden geïnstalleerd (op ondersteunde Linux[®] distributies)
- Vereiste omgevingsvariabelen worden aan `~/ .bashrc` toegevoegd
- Het [configuratiebestand](#) instellen.

Beschikbare sinds versie 18.11.

--author

Toont contactinformatie van de auteur.

--color

Kleurrijke uitvoer inschakelen (dit is de standaard voor interactieve terminals).

--nice=waarde

Deze waarde wijzigt de prioriteit voor de computer CPU zoals verlangt door kdesrc-build, en zou in de range van 0-20 moeten liggen. 0 is de hoogste prioriteit (omdat dat het minste 'nice' is), 20 is de laagste prioriteit. De standaardinstelling van kdesrc-build is 10.

--no-async

Schakelt de [asynchronous modus](#) van de update uit. In plaats daarvan zal de update in zijn geheel worden uitgevoerd voordat de bouw start. Deze optie vertraagt het gehele proces, maar als u last heeft van IPC fouten tijdens het gebruik van kdesrc-build, probeer dan het gebruik van deze optie, en dien een [bug rapport](#) in.

--no-color

Schakelt "gekleurde uitvoer" uit.

--pretend (of -p)

kdesrc-build zal door het update en bouwproces gaan, maar in plaats van enige actie voor het updaten en bouwen daadwerkelijk uit te voeren, zal het de uitvoer tonen van wat het script zou hebben gedaan (b.v. welke commando's uit te voeren, welke algemene stappen zijn genomen, enz.).

OPMERKING

Eenvoudige alleen-lezen commando's (zoals het lezen van bestandsinformatie) worden nog steeds uitgevoerd om de uitvoer meer relevant te maken (zoals het correct simuleren of broncode uitgecheckt of upgedate kan worden).

BELANGRIJK

Voor deze optie is het nodig dat sommige benodigde metadata beschikbaar is, wat normaal gesproken automatisch gedownload is, maar downloads zijn in de pretend modus uitgeschakeld. Als u nog nooit kdesrc-build eerder heeft gebruikt (en daarom deze metadata niet heeft), dan moet u eerst **kdesrc-build --metadata-only** opstarten om eerst de benodigde metadata te downloaden.

--quiet (of -q)

Wees niet zo luidruchtig met de uitvoer. Met dit ingeschakeld zal alleen het minimum in de uitvoer te zien zijn.

--really-quiet

Alleen waarschuwingen/fouten worden getoond.

--verbose

Beschrijft uitgebreid wat er gebeurt, en wat kdesrc-build aan het doen is.

--src-only (of --svn-only)

Voert alleen de update van de broncode uit. (De `--svn-only` parameter is alleen ondersteund vanwege compatibiliteit met oudere versies van het script).

--build-only

Voert alleen het bouwproces uit.

--install-only

Voert alleen het installatieproces uit.

--metadata-only

Voert alleen het metadata downloadproces uit. kdesrc-build doet dit normaal automatisch, maar u wilt dit wellicht handmatig doen zodat de commandoregelparameter `--pretend` kan werken.

--rebuild-failures

Gebruik deze optie om alleen die modules te bouwen die eerder mislukt zijn bij uitvoeren van kdesrc-build. Dit is handig als een belangrijk aantal mislukkingen optraden gemend met succesvol bouwen. Na het repareren van de oorzaken van de mislukkingen kunt u dan gemakkelijk alleen de modulen bouwen die eerder mislukten.

OPMERKING

Merk op dat de lijst met 'eerder mislukte modules' wordt gereset elke keer dat het uitvoeren van kdesrc-build eindigt met enige mislukkingen. Het wordt niet gereset door een volledig succesvol bouwen, u kunt dus met succes een paar modules bouwen en dan zal deze vlag nog steeds werken.

Deze optie is toegevoegd in kdesrc-build 15.09.

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

Deze optie veroorzaakt dat kdesrc-build automatisch andere KDE en Qt™ modulen invoegt in het bouwen, indien vereist voor de modulen die u hebt gevaagd om te bouwen op de opdrachtregel of in uw [configuratiebestand](#).

De modulen die zijn toegevoegd zijn zoals opgenomen in de broncode van het KDE beheersysteem. Zie Section 2.6.4.

Het corresponderende configuratiebestands optie is [include-dependencies](#).

U kunt ook `--no-include-dependencies` gebruiken, die automatisch invoegen van extra afhankelijkheidsmodulen uitsluitent.

--ignore-modules

Neem niet de modulen mee die in de rest van de commandoregel worden doorgegeven in het bijwerken/bouwproces (dit is handig als u de meeste modulen in uw [configuratiebestand](#) wilt bouwen en alleen een paar overslaan).

--no-src (of --no-svn)

Slaat het contact maken met de Subversion server over. (De `--no-svn` parameter is alleen ondersteund vanwege compatibiliteit met oudere versies van het script).

--no-build

Het bouwproces overslaan.

--no-metadata

Download niet automatisch de voor KDE git modules benodigde extra metadata. De updates van de broncode voor de modules zelf zal nog gewoon gebeuren tenzij u ook `--no-src` opgeeft.

Dit kan handig zijn als u frequent kdesrc-build herhaalt omdat de metadata niet vaak wijzigt. Maar merk op dat het voor veel andere functionaliteiten nodig is dat de metadata beschikbaar is. U kunt overwegen om eenmalig kdesrc-build met de optie `--metadata-only` te gebruiken en daarna deze optie voor de daaropvolgende runs te gebruiken.

--no-install

Installeert niet automatisch pakketten nadat ze zijn gebouwd.

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

Deze optie schakelt expliciet het overslaan van het bouwproces uit (een optimalisatie gecontroleerd door de optie `build-when-unchanged`). Dit is handig om te zorgen dat kdesrc-build het bouwproces uitvoert als u iets heeft gewijzigd dat kdesrc-build niet kan controleren.

`--force-build` heeft dezelfde functie, en is misschien makkelijker te onthouden.

--debug

Schakelt de debug modus van het script in. Op dit moment houd dat in dat alle uitvoer naast dat het zoals normaal naar de log-map gaat ook naar de gewone uitvoer gaat. Daarnaast geven veel functies in de debug modus meer commentaar over wat ze aan het doen zijn.

--query=modus

Dit commando veroorzaakt dat kdesrc-build een parameter opvraagt van de modules in de bouwlijst (ofwel doorgegeven op de commandoregel of gelezen uit het configuratiebestand), waarna het resultaat op het scherm wordt getoond (één module per regel).

Deze optie moet geleverd worden met een 'query-modus', die één van de volgende zou moeten zijn:

- `source-dir`, die zorgt dat kdesrc-build het volledige pad opgeeft waar de broncode van de module is opgeslagen.
- `build-dir`, die zorgt dat kdesrc-build het volledige pad opgeeft waar het bouwproces van de broncode van de module gebeurt.
- `install-dir`, die zorgt dat kdesrc-build het volledige pad opgeeft waar de module zal worden geïnstalleerd.
- `project-path`, die zorgt dat kdesrc-build de locatie van de module in de hierarchy van KDE broncoderepositories opgeeft. Zie Section 2.6.4 voor meer informatie over de hiërarchie.
- `branch`, die zorgt dat kdesrc-build de opgeloste git-branch opgeeft die voor elke module zal worden gebruikt, gebaseerd op de effectieve instellingen van `tag`, `branch` en `branch-group`.
- Anders kunnen optie-namen gebruikt worden die geldig zijn voor modules in het `configuratiebestand`, de opgeloste waarde daarvan zal voor elke module worden getoond.

Als een enkele module wordt doorgegeven op de opdrachtregel, dan is de uitvoer gewoon de waarde van de parameter die wordt afgevraagd. Als meerdere (of geen) modules worden doorgegeven op de opdrachtregel, dan wordt elke regel voorafgegaan door de naam van de module. In alle gevallen zal kdesrc-build stoppen met de uitvoering wanneer elke waarde is uitgevoerd.

Deze optie is toegevoegd in kdesrc-build 16.05.

Bijvoorbeeld, het commando `'kdesrc-build --query branch kactivities kdepim'` zou kunnen eindigen met meldingen zoals:

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

`--no-rebuild-on-fail`

Probeert niet meer modules te herbouwen nadat het de eerste keer is mislukt om ze te bouwen. `kdesrc-build` zal dit nooit proberen bij een module waarbij het al de eerste keer is geprobeerd om te bouwen.

`--refresh-build`

Maakt het bouwsysteem en make opnieuw vanaf het begin aan.

`--reconfigure`

Start **cmake** (voor KDE modules) of **configure** (voor Qt™) opnieuw, zonder de bouwmap op te ruimen. U zou normaal gesproken dit niet hoeven te specificeren, omdat `kdesrc-build` het zal detecteren als u relevante optiewaarden heeft gewijzigd en automatisch de build setup opnieuw zal uitvoeren. Deze optie is wordt uitgevoerd als u `--refresh-build` gebruikt.

`--resume-from`

Deze optie gebruikt u om de bouw te hervatten vanaf de opgegeven module, wat de volgende optie op de commandoregel zou zijn. Specificeer geen andere modulenames op de commandoregel.

OPMERKING

Deze optie voegde vroeger `--no-src` toe, maar doet dit niet meer (vanaf `kdesrc-build` 1.13). Als u het updaten van broncode wilt vermijden bij het verdergaan, voeg dan naast de andere opties gewoon `--no-src` toe.

Zie ook: `--resume-after` en Section 6.3.5.1. U geeft waarschijnlijk de voorkeur aan deze commandoregelparameter als u het probleem met het bouwen heeft opgelost en wilt dat `kdesrc-build` verder gaat met het bouwen.

`--resume-after`

Deze optie gebruikt u om de bouw te hervatten na de opgegeven module, wat de volgende optie op de commandoregel zou zijn. Specificeer geen andere modulenames op de commandoregel.

OPMERKING

Deze optie voegde vroeger `--no-src` toe, maar doet dit niet meer (vanaf `kdesrc-build` 1.13). Als u het updaten van broncode wilt vermijden bij het verdergaan, voeg dan naast de andere opties gewoon `--no-src` toe.

Zie ook: `--resume-after` en Section 6.3.5.1. U geeft waarschijnlijk de voorkeur aan deze commandoregeloptie als u het probleem met het bouwen heeft opgelost en de module zelf heeft gebouwd en geïnstalleerd, en wilt dat `kdesrc-build` verder gaat met het bouwen van de volgende module.

`--resume`

Deze optie kan gebruikt worden om `kdesrc-build` uit te voeren nadat het bouwen is mislukt.

Het hervat het bouwen vanaf de module die is mislukt, met gebruik van de lijst van modules die eerder wachten om te worden gebouwd en schakelt eveneens het bijwerken van broncode en metagegevens uit. Dit is te gebruiken wanneer een eenvoudige vergissing of ontbrekende afhankelijkheid het bouwen laat mislukken. Nadat u de fout hebt gerepareerd kunt u snel terug gaan naar het bouwen van de modules die u eerder aan het bouwen was, zonder te rommelen met `--resume-from` en `--stop-before`.

Dit is zelfs nog handiger als u het combineert met commandoregelparameter `--stop-on-failure`, met name als u initieel uw ontwikkelomgeving opzet.

Deze optie is toegevoegd in `kdesrc-build` 1.16.

--stop-before

Deze commandoregeloptie gebruikt u om het bouwproces te stoppen net *voor* een module normaal gesproken zou worden gebouwd.

Bijvoorbeeld, als het normale bouwproces is moduleA, moduleB, moduleC, dan zal `--stop-before=moduleB` veroorzaken dat kdesrc-build alleen moduleA zal bouwen.

Deze optie voor de opdrachtregel is toegevoegd in kdesrc-build 1.16.

--stop-after

Deze commandoregeloptie gebruikt u om het bouwproces te stoppen net *na* een module normaal gesproken zou worden gebouwd.

Bijvoorbeeld, als het normale bouwproces is moduleA, moduleB, moduleC, dan zal `--stop-after=moduleB` veroorzaken dat kdesrc-build zowel moduleA en moduleB zal bouwen.

Deze optie voor de opdrachtregel is toegevoegd in kdesrc-build 1.16.

--stop-on-failure

Deze optie zorgt er voor dat het bouwen wordt afgebroken zodra er iets mislukt. Nuttig wanneer u initieel uw ontwikkelomgeving aan het instellen bent. Zonder deze vlag zal kdesrc-build proberen door te gaan met de rest van de modules in het bouwen om tijdverspilling te voorkomen in het geval er een probleem is met een enkele module.

Deze optie is toegevoegd in kdesrc-build 1.16. Zie ook de optie [stop-on-failure](#).

--rc-file

Dit interpreteert de volgende commandoregelparameter als het bestand waarvan de configuratiegegevens moeten worden ingelezen. De standaardwaarde voor deze parameter is `kdesrc-buildrc` (deze wordt in de huidige map gezocht) indien aanwezig, of anders `~/kdesrc-buildrc`. Zie ook hoofdstuk 4.

--print-modules

Neemt alle acties tot en met opnieuw ordenen van moduleafhankelijkheid gespecificeerd op de opdrachtregel (of configuratiebestand), toont één per regel de modules die verwerkt zouden worden en eindigt dan zonder verdere actie.

De metagegevens van `kde-project` wordt eerst gedownload (maar, zie `--pretend` of `--no-src`).

De meldingen zijn niet volledig compatibel met gebruik door scripts omdat andere meldingen gegenereerd kunnen worden totdat de lijst met modules wordt getoond.

Dit is vooral handig om snel te bepalen wat kdesrc-build begrijpt wat de afhankelijkheden van een module zijn, wat inhoud dat het alleen bruikbaar is bij modules van `kde-projects`. Deze optie is ook te combineren met `--resume-from`, `--resume-after`, `--stop-before`, `--stop-after`.

--list-build

Maakt een lijst met de modules die gebouwd zouden worden, in de volgorde waarin ze gebouwd worden. Indien van toepassing, bevat de uitvoerlijst ook welke commit/branch/tag geselecteerd zou worden voor checkout.

Deze optie is gelijk aan `--print-modules`. Voor meer gedetailleerde informatie over hoe modules in relatie staan tot elkaar, zie ook: `--dependency-tree`.

--dependency-tree

Drukt informatie over afhankelijkheden af over de modulen die gebouwd zouden worden in boomstructuurformaat (recursief). Getoonde informatie omvat ook waarvan specifieke commit/branch/tag afhankelijk is en of wel of niet de afhankelijkheid zou worden gebouwd. Let op: de gegenereerde uitvoer kan behoorlijk groot worden voor toepassingen met veel afhankelijkheden.

Handboek scripts voor kdesrc-build

--run

Deze optie interpreteert het volgende item op de commandoregel als een uit te voeren programma, en kdesrc-build zal dan het inlezen van het configuratiebestand beëindigen, de omgevingsvariabelen zoals normaal bijwerken, en het opgegeven programma opstarten.

Dit zal in de meeste gevallen echter geen shell met de omgevingsvariabelen van kdesrc-build opstarten, omdat interactieve shells tijdens de start standaard tenminste een gedeelte van de omgevingsvariabelen (zoals `PATH` en `KDEDIRS`) resetten.

TIP

Als u de de omgevingsvariabelen wilt zien die door kdesrc-build wordt gebruikt, dan kunt het commando **printenv** gebruiken:

```
$ 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=</path/to/kde>

Hiermee kunt u de map wijzigen waar in KDE vanaf de commandoregel zal worden geïnstalleerd. Deze optie houdt `--reconfigure` in, maar het gebruik van `--refresh-build` kan nodig zijn.

--revision

Deze optie zorgt ervoor dat kdesrc-build voor elke Subversion module een specifieke genummerde revisie uitcheckt, daarmee alle voor deze modules ingestelde [branch](#), [tag](#), of [revision](#) optie's overschrijven.

Deze optie is waarschijnlijk geen goed idee, en is alleen ondersteund vanwege compatibiliteit met oudere versies van het script.

--build-system-only

Deze optie zorgt er voor dat kdesrc-build het bouwen van een module afbreekt net voordat het commando **make** zou worden opgestart. Dit wordt alleen ondersteund vanwege de compatibiliteit met oudere versies, Deze functionaliteit is niet behulpzaam bij het huidige KDE build systeem.

--install

Als dit de enige commandoregelparameter is, dan probeert het alle modules te installeren die voorkomen in `log/latest/build-status`. Als u commandoregelparameters specificeert na `--install`, dan wordt aangenomen dat dit allemaal te installeren modules zijn (zelfs als ze bij de laatste run niet succesvol gebouwd zijn).

--no-snapshots

Deze optie zorgt er voor dat kdesrc-build altijd een normale initiële checkout van een module doet in plaats van een quick-start snapshot (alleen beschikbaar voor Git modules uit de `kde-projects` repository). Merk op dat u deze optie alleen moet gebruiken als het gebruiken van snapshots mislukt, omdat de quick-start snapshot minder belastend zijn voor de KDE broncoderepositories.

OPMERKING

Module snapshots *zijn* echte checkouts. Normaal gesproken heeft u deze optie niet nodig, het is alleen een hulpmiddel voor troubleshooting.

`--delete-my-patches`

Deze optie zorgt er voor dat kdesrc-build de mappen met broncode verwijdert waarin gebruikersdata kan voorkomen, zodat de module opnieuw kan worden gedownload. Dit is normaal gesproken alleen zinnig voor KDE-ontwikkelaars (die wellicht lokaal wijzigingen hebben die verwijderd zouden kunnen worden).

Dit wordt op dit moment alleen gebruikt om modules uit te checken die zijn geconverteerd van Subversion naar Git. Normaal gesproken heeft u deze optie niet nodig, kdesrc-build zal vragen om opnieuw te worden gestart als dat nodig is.

`--delete-my-settings`

Deze optie gebruikt u om kdesrc-build bestaande bestanden waarin gebruikersdata kan voorkomen te laten overschrijven.

Dit wordt op dit moment alleen gebruikt voor de xsession setup van de login manager. Normaal gesproken heeft u deze optie niet nodig, kdesrc-build zal vragen om opnieuw te worden gestart als dat nodig is.

`--<optie-naam>=`

U kunt dit gebruiken om een optie in uw [configuratiebestand](#) voor elke module te overschrijven. Bijvoorbeeld, om de optie `log-dir` te overschrijven, zou u kunnen doen: `--log-dir= /path/to/dir`.

OPMERKING

U kunt deze functionaliteit alleen gebruiken bij optie-namen die al bekend zijn in kdesrc-build, en dat ze niet al worden gebruikt bij relevante commandoregelopties. Bijvoorbeeld de optie `async` van het configuratiebestand heeft de specifieke commandoregelopties `--async` en `--no-async` geprefereerd door kdesrc-build.

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

U kunt dit gebruiken om een optie in uw [configuratiebestand](#) voor specifieke module te overschrijven.

Van alle andere commandoregelopties wordt aangenomen dat het bij te werken en te bouwen modules zijn. Haal het bouwen en het installeren niet door elkaar.

Hoofdstuk 6

kdesrc-build gebruiken

6.1 Inleiding

Normaal gesproken is het gebruik van kdesrc-build nadat u door hoofdstuk 2 bent gegaan niet moeilijker dan het invoeren op de commandoregel van:

```
% kdesrc-build
```

kdesrc-build zal de broncode voor KDE downloaden, vervolgens proberen dat te configureren en te bouwen, en dat daarna te installeren.

Lees verder om te ontdekken hoe kdesrc-build dit doet, en wat u verder met dit hulpmiddel kan doen.

6.2 Standaard functionaliteiten van kdesrc-build

6.2.1 qt-ondersteuning

kdesrc-build ondersteunt het bouwen van de Qt™ toolkit die door KDE software wordt gebruikt als handreiking aan de gebruikers. Deze ondersteuning wordt afgehandeld door een speciale module genaamd qt.

OPMERKING

Qt™ wordt ontwikkeld in een separate repository van KDE de software is te vinden op <http://code.qt.io/-cgit/qt/>.

Om Qt™ te kunnen bouwen, moet u er op letten dat `qtdir` verwijst naar de map waar u Qt™ wilt installeren zoals is beschreven in Section 2.2.

Let er op dat de qt module aanwezig is in uw `.kdesrc-buildrc`, voor enig andere module in het bestand. Als u het voorbeeld configuratiebestand gebruikt, dan hoeft u enkel de al aanwezige qt module entry te uncommenten.

Nu moet u controleren dat de opties `repository` en `branch` juist zijn ingesteld:

1. De eerste optie is om Qt™ te bouwen met behulp van een mirror in de KDE broncode-repositories (zonder verdere wijzigingen, het is gewoon een clon van de officiële broncode).

Dit wordt sterk aanbevolen vanwege problemen die u kunt ondervinden met het clonen van de complete Qt™ module uit de officiële repository.

U kunt de optie `repository` voor de qt module instellen op `kde:qt` om deze mogelijkheid te gebruiken.

2. In het andere geval, kunt u om de standaard Qt™ te bouwen, uw optie `repository` instellen op `git://gitorious.org/qt/qt.git`. Merk op dat u problemen kunt ondervinden met de initiële clon van Qt™ uit deze repository.

In beide gevallen, moet u de `branch`-optie instellen op `master` (tenzij u een andere branch wilt gebruiken).

6.2.2 Standaard flags toegevoegd door kdesrc-build

Opmerking: deze sectie is niet van toepassing op modules waarvoor u een aangepaste toolchain hebt geconfigureerd, met bijv. [cmake-toolchain](#).

Om het u makkelijker te maken en tijd te besparen, voegt kdesrc-build enkele standaard paden toe aan uw omgeving:

- De paden naar de KDE en de Qt™ libraries worden automatisch aan de variabele `LD_LIBRARY_PATH` toegevoegd. Dit betekent dat u `libpath` niet hoeft te bewerken om ze toe te voegen.
- De paden naar de ontwikkel-hulpprogramma's van KDE en de Qt™ worden automatisch aan de variabele `PATH` toegevoegd. Dit betekent dat u `binpath` niet hoeft te bewerken om ze toe te voegen.
- Het pad naar de door KDE geleverde pkg-config worden automatisch aan de variabele `PKG_CONFIG_PATH` toegevoegd. Dit betekent dat u `set-env` niet hoeft te gebruiken om ze toe te voegen.
- De instellingen voor `kdedir` wordt automatisch doorgegeven aan de omgevingsvariabele `KDEDIR` tijdens het bouwen. (`KDEDIRS` wordt niet permanent gewijzigd).
- De instellingen voor `qtdir` wordt automatisch doorgegeven aan de omgevingsvariabele `QTDIR` tijdens het bouwen.

6.2.3 Het wijzigen van de bouwprioriteit van kdesrc-build

Programma's kunnen verschillende prioriteiten hebben in besturingssystemen zoals Linux® en BSD. Hierdoor kan het systeem computertijd toewijzen aan de verschillende programma's overeenkomstig met hoe belangrijk ze zijn.

kdesrc-build zal zichzelf normaal gesproken een lage prioriteit geven zodat de rest van de programma's op uw systeem onaangetast zijn en normaal zullen werken. Door deze techniek te gebruiken, zal kdesrc-build alleen extra CPU-capaciteit gebruiken als dat beschikbaar is.

kdesrc-build houdt een hogere prioriteit dan batch-processen en CPU-donatie programma's zoals [Seti@Home](#).

Om kdesrc-build zodanig te wijzigen dat het permanent een hogere (of lagere) prioriteit heeft, moet u de `niceness`-instelling in het [configuratiebestand](#) wijzigen. De instelling `niceness` regelt hoe 'nice' kdesrc-build is naar andere programma's. Met andere woorden, door het hebben van een hogere `niceness`, geeft kdesrc-build een lagere prioriteit. Dus door kdesrc-build een hogere prioriteit te geven, reduceert u de `niceness` (en vice versa). De `niceness` kan gaan van 0 (helemaal niet nice, hoogste prioriteit) tot 20 (super nice, laagste prioriteit).

U kunt ook de prioriteit van kdesrc-build tijdelijk wijzigen me de hulp van `--nice` [commandoptie](#). De gebruikte optiewaarde moet precies hetzelfde zijn als voor `niceness`.

OPMERKING

Root kan sommige programma's gebruiken met een negatieve waarde voor nice, met voor dergelijke programma's de bijbehorende nog hogere prioriteit. Een negatieve (of zelfs 0) [niceness](#) voor kdesrc-build instellen is geen goed idee, omdat het niet helpt om de CPU-tijden significant korter te maken, maar het zorgt er wel voor dat uw computer erg traag lijkt in het geval dat u uw computer wilt gebruiken.

Om kdesrc-build met een prioriteit van 15 (een lagere prioriteit dan normaal) te laten werken:

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

Maar u kunt ook het [configuratiebestand](#) bewerken om de wijziging permanent te maken:

```
niceness 15
```

TIP

De [niceness](#)-optie heeft alleen effect op het gebruik van de processor(s) van de computer. Een ander belangrijke factor die invloed heeft op de performance van de computer is hoeveel data input of output (I/O) een programma gebruikt. Om te regelen hoeveel I/O een programma kan gebruiken, hebben moderne Linux[®] besturingssystemen een vergelijkbaar hulpmiddel genaamd ionice. kdesrc-build ondersteunt ionice, (maar alleen om het compleet IN- of UIT- te schakelen) door de optie [use-idle-io-priority](#) te gebruiken, vanaf kdesrc-build versie 1.12.

6.2.4 Installatie als de Root

U wilt wellicht kdesrc-build gebruiken voor de installatie met root-privileges. Maar dit is wellicht voor de niet aanbevolen systeembrede installatie. Maar dit is ook handig voor gebruik bij een aanbevolen single user KDE bouw. Dat is omdat sommige modules (met name kdatabase) programma's installeert die tijdens gebruik kort hogere permissies nodig hebben. Zij kunnen niet op deze permissie-niveau's werken tenzij ze met de verhoogde permissies zijn geïnstalleerd.

U zou kdesrc-build gewoon direct als root kunnen opstarten, maar dit wordt niet aanbevolen, omdat het programma niet voor dat gebruik is uitgetest. Al hoe wel het zou veilig moeten zijn om het programma op die manier te gebruiken, is het beter om het gebruik als root zoveel mogelijk te vermijden.

Om dit probleem op te lossen, heeft kdesrc-build de optie [make-install-prefix](#). U kunt deze optie gebruiken om een te gebruiken commando te specificeren dat de installatie als een andere gebruiker uitvoert. De aanbevolen manier om dit commando te gebruiken is met de hulp van het programma Sudo, wat het installatieproces zal starten als root.

Bijvoorbeeld, om met gebruik van Sudo alle modules te installeren, kan u het volgende doen:

```
global
  make-install-prefix sudo
  # Other options
end global
```

Om [make-install-prefix](#) te gebruiken voor een enkele module, moet u het volgende doen:

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


6.2.5 De voortgang van het bouwen van een module tonen

Deze functionaliteit is altijd beschikbaar, en is indien mogelijk automatisch ingeschakeld. Wat dit doet is het tonen van de geschatte voortgang van het bouwproces tijdens het bouwen van een module; op die manier kunt u weten hoelang het nog duurt om een module te bouwen.

6.3 Uitgebreide functionaliteiten

6.3.1 Een module gedeeltelijk bouwen

Het is mogelijk om alleen gedeeltes van een enkele KDE module te bouwen. Bijvoorbeeld, u wilt alleen een programma uit een module compileren. `kdesrc-build` heeft mogelijkheden om het u makkelijk te maken. Er zijn meerdere manieren om dit te doen.

6.3.1.1 Gedeeltes van een module uitchecken

Dit is misschien de beste manier om dit te doen. Als het werkt, dan bespaart het u download-tijd en hardeschijfruimte. Wat er gebeurt is dat `kdesrc-build` alleen de gedeeltes van een module download die u specificeert. Dit doet u met de `checkout-only`-optie voor een module, waarin u een lijst met te downloaden mappen specificeert.

TIP

Als u nog niet weet wat uit een module te downloaden, dan kan het een goed idee zijn om eerst door de Subversion-indeling van een module te bladeren, met behulp van [WebSVN](#).

Om alleen `KUser` en `KSystemLog` uit `kdeadmin` te pakken, zou u `checkout-only` op de volgende manier kunnen gebruiken:

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

BELANGRIJK

De mappen worden gebouwd in de volgorde zoals ze genoemd zijn in de optie. Als een van de mappen iets nodig heeft uit de module om te compileren, dan moet u er zorg voor dragen dat beide in de `checkout-only`-regel voorkomen, en dat de benodigde dependency eerder genoemd wordt dan de map die het nodig heeft.

Ook heeft een programma soms andere mappen nodig en is het lastig om uit te vinden welke dat zijn, zodat het nodig kan zijn om uit te proberen door het constant toevoegen van mappen aan de optie. Deze optie is afhankelijk van de ondersteuning van het bouwsysteem van de module, en is daarom alleen handig bij modules die een verzameling zijn van individuele programma's.

Een laatste opmerking over deze optie: Als u de waarde van deze optie wijzigt, dan moet u `kdesrc-build --refresh-build module` gebruiken om er voor te zorgen dat deze module goed wordt hergeconfigureerd. Bovendien, `kdesrc-build` zal nooit bestaande bestanden verwijderen als u een aantal mappen uit uw optie `checkout-only` verwijdert, of de optie toevoegt aan een module dat al is uitgecheckt.

6.3.1.2 Mappen uit een bouw verwijderen

In plaats van wat wordt gedownload beperken, is het ook mogelijk om alles te downloaden en dat het bouwsysteem een paar mappen niet meeneemt bij het bouwen. Dit kan handig zijn als een map altijd breekt en onnodig is voor de rest van de module.

Dit regelt u met de optie `do-not-compile`. Het werkt vergelijkbaar met de al eerder beschreven optie `checkout-only`, wat inhoudt dat het gewoon een lijst met mappen is die niet gecompileerd moeten worden.

BELANGRIJK

Net zoals bij `checkout-only`, vereist deze optie dat u in ieder geval na een wijziging het bouwsysteem herconfigureert. Dit doet u door het commando `kdesrc-build --reconfigure module` te gebruiken.

Om de map `python` uit het kdebindings bouwproces te verwijderen:

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

OPMERKING

Deze functie is afhankelijk van enkele afspraken gebruikt in de meeste KDE modules. Het hoeft daarom niet bij alle programma's te werken.

6.3.2 Ondersteuning voor branching en tagging bij for kdesrc-build

6.3.2.1 Wat zijn branches en tags?

Subversion ondersteunt het beheer van de geschiedenis van de KDE-broncode. KDE gebruikt deze ondersteuning om voor ontwikkeling branches (afsplitsingen) te creëren, en de repository zo nu en dan te taggen (markeren) met een nieuwe versie release.

Bijvoorbeeld, de KMail-ontwikkelaars kunnen aan het werk zijn voor een nieuwe functie in een andere branch om te vermijden dat ze de versie breken die bij de meeste ontwikkelaars in gebruik is. In deze branch kunnen ontwikkelingen aan de gang zijn, terwijl in de hoofd-branch (genaamd /trunk) ook ontwikkelingen aan de gang kunnen zijn.

Een tag, aan de andere kant, is een snapshot van de broncoderepository op een tijdstip. Dit wordt door het administratieteam van KDE gebruikt om een versie van de code te markeren als geschikt voor release en de ontwikkelaars toch te laten werken aan de code.

In Subversion is er geen verschil tussen branches, tags, of trunk wat betreft de code. Het is alleen maar een afspraak tussen de ontwikkelaars. Dit maakt het moeilijk om goed branches en tags te hanteren door `kdesrc-build`. Maar, er zijn toch enkele dingen die gedaan kunnen worden.

6.3.2.2 Het gebruik van branches en tags

Het gebruik van branches en tags regelt u met een set van opties, die variëren van een gewoon verzoek voor een versie, tot een specifieke URL om van te downloaden voor gevorderde gebruikers.

De makkelijkste methode is om de `branch` en `tag`-opties te gebruiken. U gebruikt gewoon de optie samen met de naam van de gewenste branch of tag van een module, en `kdesrc-build` zal proberen om de juiste locatie in de KDE repository te bepalen waarvan het moet downloaden. Bij de meeste KDE modules werkt dit vrij goed.

Om `kdlibs` van KDE 4.6 (die gewoon bekend staat als de 4.6 branch) te downloaden:

```
module kdelibs
  branch 4.6
  # other options...
end module
```

Of, om kdemultimedia te downloaden zoals het was released met KDE 4.6.1:

```
module kdemultimedia
  tag 4.6.1
  # other options...
end module
```

TIP

U kunt een algemene (global) branch waarde instellen. Maar als u dat doet, vergeet dan niet om een andere branch voor de modules die niet de global branch gebruiken te specificeren!

6.3.2.3 Geavanceerde opties wat betref branches

kdesrc-build heeft twee opties voor situaties waar [branch](#) en [tag](#) het juiste pad incorrect bepaalt: [module-base-path](#) en [override-url](#).

- [module-base-path](#) gebruikt u om kdesrc-build te helpen met het invullen van het ontbrekende gedeelte van een module's pad. In de repository van KDE, zijn alle paden in de vorm `svnRoot/module-base-path/module-name`. Normaal gesproken kan kdesrc-build zelf het van toepassing zijnde middelste gedeelte bepalen. Als dat niet lukt, dan kunt u [module-base-path](#), op de volgende manier gebruiken:

```
module kdesupport
  # kdesupport ondersteunt verschillende tags om makkelijk het vereiste
  # software voor een bepaalde KDE Platform release te organiseren.
  module-base-path tags/kdesupport-for-4.5
end module
```

Dit zorgt ervoor dat kdesrc-build gaat downloaden kdesupport van (in dit voorbeeld), `svn://a.nonsvn.kde.org/home/kde/tags/kdesupport-for-4.5`.

TIP

In vorige versies van kdesrc-build werd de [module-base-path](#) anders behandeld. Als u problemen tegenkomt bij het gebruik van een oude [module-base-path](#) definitie dan zou misschien moeten controleren of het daadwerkelijke pad hetzelfde is als wat kdesrc-build verwacht bij het gebruik van de optie `--pretend`.

- Bij de optie [override-urls](#) het vereist dat u het exacte pad specificeert waarvan u download. Maar, hiermee kunt u wel van paden downloaden waarmee u bij voorgaande versies van kdesrc-build geen hoop had om van te kunnen downloaden. Op dit moment zou de optie [module-base-path](#) genoeg moeten zijn om van elke Subversion bron URL te kunnen downloaden.

BELANGRIJK

kdesrc-build zal de waarde die u specificeerde voor [override-url](#) absoluut niet wijzigen of corrigeren, dus als u uw [svn-server](#)-instelling wijzigt, dan zal u dit ook moeten bijwerken.

6.3.3 Hoe kdesrc-build probeert zich te verzekeren van een succesvolle bouw

6.3.3.1 Automatisch herbouwen

kdesrc-build had vroeger de gewoonte om onderdelen toe te voegen bij automatische pogingen om een module te herbouwen na een mislukking (omdat soms dan deze poging tot herbouwen wel werkte, vanwege bugs in het bouwsysteem op dat moment). Dankzij de omschakeling naar CMake heeft het bouwsysteem geen last meer van deze bugs, zodat kdesrc-build niet langer meer dan een keer zal proberen een module te herbouwen. Er zijn echter situaties waar kdesrc-build toch zal proberen actie te ondernemen:

- Als u [configure-flags](#) of [cmake-options](#) voor een module wijzigt, dan zal kdesrc-build dit detecteren en automatisch configure of cmake voor dat module herstarten.
- Als het bouwsysteem niet bestaat (ook als kdesrc-build het niet heeft verwijderd) dan zal kdesrc-build het automatisch recreëren. Dit is handig om een volledige [--refresh-build](#) uit te kunnen voeren voor een specifiek module zonder dat het ook voor andere modules uitgevoerd te hoeven worden.

6.3.3.2 Handmatig een module herbouwen

Als u een wijziging aan de optiewaarden van een module aanbrengt, of de broncode van de module wijzigt zodanig dat kdesrc-build het niet herkent, dan kan het nodig zijn dat u handmatig de module moet herbouwen.

U kunt dit gewoon doen door **kdesrc-build --refresh-build module** op te starten.

Als u graag heeft dat kdesrc-build automatisch de module tijdens de volgende normale bouw update automatisch wordt herbouwd, dan kunt een speciaal bestand creëren. Elke module heeft een bouwmap. Als u u een bestand genaamd `.refresh-me` in de bouwmap van een module creëert, dan zal kdesrc-build de volgende keer dat het bouwproces gebeurt de module herbouwen, zelfs als het normaal gesproken de snellere incrementele bouw zou uitvoeren.

TIP

Standaard is de bouwmap `~/kdesrc/build/ module /`. Als u de waarde van de optie [build-dir](#) wijzigt, dan wordt dat gebruikt in plaats van `~/kdesrc/build`.

Herbouwen door `.refresh-me` voor de module `kdelibs` te gebruiken:

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

6.3.4 De instellingen van de omgevingsvariabelen wijzigen

Normaal gesproken gebruikt kdesrc-build om updates en het bouwproces uit te voeren de bij de start aanwezige omgeving voor het gebruik van programma's. Dit is handig voor het gebruik van kdesrc-build vanaf de commandoregel.

Maar als u de instellingen voor de omgevingsvariabelen wilt wijzigen waarvoor kdesrc-build niet direct een optie voor heeft. (Bijvoorbeeld, om de vereiste omgevingsvariabelen op te zetten om kdesrc-build door een timer zoals Cron te laten starten) Dan kunt u dit doen met de optie [set-env](#).

Afwijkend van de meeste andere opties, kunt u deze meer dan een keer instellen, en het accepteert twee waarden, gescheiden door een spatie. De eerste is de naam van de in te stellen omgevingsvariabele, en de rest van de regel is de waarde.

Set **DISTRO=BSD** for all modules:

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

6.3.5 Hervatten van bouwprocessen

6.3.5.1 Hervatten na een mislukte of afgebroken bouwproces

U kunt opdracht geven aan kdesrc-build om de bouw vanaf een andere module te starten dan wat het normaal zou doen. Dit kan handig zijn als een groep modules mislukte, of als u een de bouw in het midden af heeft gebroken. U kunt dit regelen met de hulp van de optie `--resume-from` en de optie `--resume-after`.

OPMERKING

Oudere versies van kdesrc-build zouden de update van de broncode overslaan bij het hervatten van een bouw. Standaard gebeurt dit niet langer, maar u kunt altijd de commandoregelparameter `--no-src` gebruiken om de update van de broncode over te slaan.

De start van de bouw hervatten vanaf kdebase:

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

De start van de bouw hervatten na kdebase (in het geval dat u handmatig het probleem herstelde en de module zelf installeerde):

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

Als de laatste kdesrc-build-bouw eindigde met een bouw-mislukking, dan kunt u ook de commandoregelparameter `--resume` gebruiken, wat hervat met de laatste bouw, startend met de module die mislukte. De update van de broncode en metadata worden ook overgeslagen (maar als u dit nodig heeft, dan is het in het algemeen beter om `--resume-from` te gebruiken).

6.3.5.2 Modules tijdens de bouw negeren

Vergelijkbaar met de manier waarop u [de bouw hervat vanaf een module](#), in plaats daarvan kunt u ervoor kiezen om alles zoals normaal te updaten en te bouwen, en een groep modules te negeren.

Dit kunt u doen door de optie `--ignore-modules` te gebruiken. Met deze optie kunt u opdracht geven aan kdesrc-build om bij de update en de bouw op de commandoregel opgegeven modules te negeren.

Tijdens een complete run extragear/multimedia en kdereview negeren:

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

6.3.6 Vanaf de commandoregel opties wijzigen

6.3.6.1 Algemene (global) opties wijzigen

U kunt direct vanaf de commandoregel de optiewaarden ingelezen van het [configuratiebestand](#) wijzigen. Dit overschrijft de instellingen uit het configuratiebestand, maar dat is alleen maar tijdelijk. Het heeft alleen maar effect zolang het opdrachtregel aanwezig blijft.

Bij `kdesrc-build` kunt u opties zoals *option-name* wijzigen door op de commandoregel een argument op te geven in de vorm **--optie-naam=waarde**. `kdesrc-build` zal herkennen dat het niet weet wat de optie inhoud, en de naam opzoeken in de lijst met optienamen. Als de naam onbekend is, dan geeft het een waarschuwing, anders zal het tijdelijk de instellingen in het configuratiebestand overschrijven met de door u gegeven waarde.

De optie `source-dir` instellen op `/dev/null` voor een test:

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

6.3.6.2 Module opties wijzigen

Het is ook mogelijk om opties alleen voor een specifieke module te wijzigen. De syntax is vergelijkbaar: `--module,optienaam=waarde`.

Deze wijziging overschrijft alle overeenkomstige instellingen in het [configuratiebestand](#) voor de gevonden module, en past alleen die waarden toe die met de optie op de commandoregel zijn opgegeven.

Een ander bouwmap gebruiken voor de `kdeedu` module:

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

6.4 Functionaliteiten voor KDE-ontwikkelaars

6.4.1 SSH Agent controles

`kdesrc-build` kan er op letten dat KDE-ontwikkelaars die SSH gebruiken voor toegang tot de KDE-broncoderepository niet per ongeluk het SSH Agent-hulpprogramma ingeschakeld laten. Dit kan veroorzaken dat `kdesrc-build` onafgebroken blijft wachten totdat de ontwikkelaar zijn SSH-wachtwoord invoert, daarom zal `kdesrc-build` standaard controleren of de Agent is opgestart voordat het de broncode update.

OPMERKING

Dit is alleen gedaan voor KDE-ontwikkelaars die SSH gebruiken. Omdat geen wachtwoord nodig is voor de standaard anonieme checkout. Subversion zal de wachtwoorden voor het tweede mogelijke protocol voor KDE-ontwikkelaars behandelen, `https`.

U wilt wellicht de SSH Agent controle uitschakelen, in situaties waar `kdesrc-build` is de aanwezigheid van een agent niet goed detecteert. Om de agent controle uit te schakelen, stelt u de optie `disable-agent-check` in op **true**.

De SSH agent controle uitschakelen:

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

6.5 Andere functionaliteiten van kdesrc-build

6.5.1 Het aantal meldingen van kdesrc-build wijzigen

kdesrc-build heeft verschillende opties om de hoeveelheid meldingen die het script genereert te regelen. Fouten geven in elk geval altijd een melding.

- De optie `--quiet` (de afkorting is `-q`) zorgt ervoor dat kdesrc-build veelal stil is. Alleen belangrijke meldingen, waarschuwingen, of fouten krijgt u te zien. Indien beschikbaar, is informatie over de voortgang van de bouw nog steeds zichtbaar.
- De optie `--really-quiet` (geen afkorting) zorgt ervoor dat kdesrc-build u alleen belangrijke waarschuwingen en fouten tijdens de uitvoering te zien geeft.
- De optie `--verbose` (de afkorting is `-v`) zorgt ervoor dat kdesrc-build zeer gedetailleerde meldingen geeft.
- De optie `--debug` is alleen voor debuggen, zorgt ervoor dat kdesrc-build hetzelfde reageert alsof `--verbose` is ingeschakeld, zodat commando's ook uitvoer naar het terminalvenster geven, en geeft voor veel functies debug-informatie.

6.5.2 Gekleurde meldingen

Indien opgestart vanuit een Konsole of en andere terminal, zal kdesrc-build normaal gesproken gekleurde tekst tonen.

U kunt dit uitschakelen door op de commandoregel `--no-color` te gebruiken, of door in het [configuratiebestand](#) de optie `colorful-output` op **false** te zetten.

Gekleurde meldingen uitschakelen in het configuratiebestand:

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

6.5.3 Na een bouw de onnodige mappen verwijderen

Heeft u weinig ruimte op de hardeschip maar u wilt wel een bleeding-edge KDE checkout hebben? kdesrc-build kan helpen bij het reduceren van hardeschijfruimte bij het bouwen van KDE vanuit Subversion.

OPMERKING

Wees u ervan bewust dat het bouwen van KDE veel ruimte nodig heeft. Er zijn verschillende veel ruimtevrage onderdelen bij het gebruik van kdesrc-build:

1. De daadwerkelijke checkout van de broncode kan een aanzienlijke hoeveelheid ruimte vragen. De standaard modules hebben ongeveer aan 1.6 gigabytes hardeschijfruimte nodig. U kunt dit reduceren door er zeker van te zijn dat u alleen de modules bouwt die u echt wil hebben. kdesrc-build zal geen broncode van de harde schijf verwijderen, zelfs als u het item uit het [configuratiebestand](#) verwijdert, zorg er daarom voor dat u de onnodige broncode checkouts in de map met broncode verwijdert. Merk op dat de broncode-bestanden van het internet worden gedownload, u *moet niet* ze verwijderen als u ze daadwerkelijk gebruikt, tenminste totdat u klaar bent met het gebruik van kdesrc-build.

Daarnaast, als u een Qt™ van uw distributie heeft geïnstalleerd (en de kans is groot dat u dat heeft gedaan), dan hoeft u waarschijnlijk de qt module niet te installeren. Dat bespaart ongeveer 200 megabytes hardeschijfruimte.

Nog een ding over de manier waarop Subversion werkt: er zijn op de harde schijf eigenlijk twee bestanden voor elk uit het repository uitgecheckt bestand. kdesrc-build heeft op dit punt geen code om te proberen de grootte van de broncode te minimaliseren als de broncode niet in gebruik is.

2. kdesrc-build zal voor het bouwen een separate bouwmap creëren waarin de broncode wordt gebouwd. Soms zal kdesrc-build een map met broncode kopiëren om een fake bouwmap te creëren. Als dit voorkomt, worden ruimtebesparende symlinks gebruikt, zodat dit wat hardeschijfruimte betreft geen probleem vormt. Standaard is voor een module de bouwmap veel groter dan de map met broncode. Bijvoorbeeld, de bouwmap voor kdbase is ongeveer 1050 megabytes, terwijl de broncode van kdbase's alleen maar ongeveer 550 megabytes is.

Gelukkig is de bouwmap niet meer nodig nadat een module succesvol is gebouwd en geïnstalleerd. kdesrc-build kan automatisch de bouwmap na de installatie van een module verwijderen, zie de voorbereiden hieronder voor meer informatie. Merk op dat deze stap het onmogelijk maakt voor kdesrc-build om de tijdbesparende incrementele bouw uit te voeren.

3. En tenslotte is er voor de eigenlijke installatie van KDE hardeschijfruimte nodig, wat niet in de bouwmap wordt geïnstalleerd. Standaard neemt dit minder ruimte in beslag dan de bouwmap. Maar het is moeilijker om hierover cijfers te geven.

Hoe vermindert u de nodige ruimte voor KDE? Een manier is om de juiste compiler flags te gebruiken, voor een optimalisatie voor opslagruimte in plaats van voor snelheid. Een andere manier, wat een groot effect kan hebben, is het verwijderen van debug-informatie van uw KDE-bouw.

WAARSCHUWING

U moet zeker weten wat u doet voor dat u besluit om de debug-informatie te verwijderen. Het gebruiken van bleeding-edge software houdt in dat u software gebruikt die een grotere kans heeft om te crashen dan software uit een stabiele release. Als u software gebruikt zonder debug-informatie, dan kan het erg moeilijk zijn om een goed bug rapport te creëren zodat uw bug wordt opgelost, en zal u waarschijnlijk de debug-informatie opnieuw moeten inschakelen voor het betreffende programma en opnieuw moeten bouwen om de ontwikkelaar te helpen met het fixen van de crash. Verwijder daarom de debug-informatie op uw eigen risico!

De bouwmap verwijderen na de installatie van een module. De map met broncode wordt echter bewaard, en debugging is mogelijk:

```
global
  configure-flags      --enable-debug
  remove-after-install builddir      # Remove build directory after ←
  install
end global
```

De bouwmap verwijderen na de installatie van een module. zonder debug-informatie, met ruimte-optimalisatie.

```
global
  cxxflags             -Os             # Optimize for size
  configure-flags      --disable-debug
  remove-after-install builddir      # Remove build directory after ←
  install
end global
```


Hoofdstuk 7

CMake, het bouwsysteem voor KDE

7.1 Inleiding tot CMake

In Maart 2006, versloeg het CMake programma verschillende concurrenten en werd geselecteerd als bouwsysteem voor KDE 4, daarmee het op autotools-gebaseerde systeem vervangend dat KDE vanaf het begin gebruikte.

Een introductie-pagina over CMake is beschikbaar op de [KDE Community Wiki](#). Waar het eigenlijk op neer komt is dat we in plaats van `make -f Makefile.cvs`, en dan `configure`, en tenslotte Make gebruiken, we eenvoudig CMake en Make gebruiken.

kdesrc-build heeft ondersteuning voor CMake. Enkele functionaliteiten van kdesrc-build zijn eigenlijk functionaliteiten van het onderliggende bouwsysteem, inclusief [configure-flags](#) en [do-not-compile](#). Als er equivalente functionaliteiten beschikbaar zijn, dan worden ze beschikbaar gemaakt. Bijvoorbeeld, het equivalent voor de optie `configure-flags` is `cmake-options`, en de optie `do-not-compile` wordt ook ondersteunt voor CMake vanaf kdesrc-build versie 1.6.3.

Hoofdstuk 8

Dankbetuigingen en licentie

Op- of aanmerkingen over de vertalingen van de toepassing en haar documentatie kunt u melden op <http://www.kde.nl/bugs>.

Dit document is vertaald in het Nederlands door Freek de Kruijf freekdekruijf@kde.nl.

Deze documentatie valt onder de bepalingen van de [GNU vrije-documentatie-licentie](#).

Bijlage A

De organisatie van de modules en broncode van KDE

A.1 De 'Module'

KDE groepeert de software in 'modules' van verschillende grootte. Oorspronkelijk was dit losjes gegroepeerd in een paar grote modules, maar met de introductie van de op [Git](#)-gebaseerde [broncoderepositories](#), zijn deze grote modules verder opgesplitst in vele kleinere modules.

Ook kdesrc-build gebruikt dit concept van modules. In essentie, is een 'module' een groepering van code die kan worden gedownload, gebouwd, getest, en geïnstalleerd.

A.1.1 Individuele modules

Het is makkelijk om het zo in te stellen dat kdesrc-build een enkele module bouwt. De volgende lijst is een voorbeeld van hoe een declaratie voor een op Subversion-gebaseerde module er uit zou zien in het [configuratiebestand](#).

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

TIP

Dit is een op Subversion-gebaseerde module omdat het geen [repository](#)-optie gebruikt. Ook is de `cmake-options`-optie alleen als voorbeeld gegeven, het is niet vereist.

A.1.2 Groepen van gerelateerde modulen

Op dit moment zijn de meeste KDE broncodemodules gebaseerd op Git en worden gewoonlijk gecombineerd in groepen van modules.

kdesrc-build ondersteunt daarom ook groepen modules, met [module sets](#). Een voorbeeld:

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

TIP

U kunt de module set naam (*base-modules* in dit geval) leeg laten als u dat wilt. Deze instelling `repository` geeft aan waar kdesrc-build de broncode kan downloaden, maar u kunt ook een git:// URL gebruiken.

Een speciale functionaliteit van de ‘`repository kde-projects`’ is dat kdesrc-build automatisch alle Git modules die in de door u opgegeven modules (in the KDE Project database) zijn gegroepeerd, zal downloaden.

A.1.3 Module ‘branch groups’

Het concept van een [groep van modules](#) verder uitbouwend, merkten de KDE-ontwikkelaars uiteindelijk dat het synchroniseren van namen van de Git branches over een groot aantal repositories steeds moeilijker werd, met name tijdens de ontwikkeling van de nieuwe KDE Frameworks voor Qt™ 5.

Daarom is het concept ‘branch groups’ ontwikkeld, om gebruikers en ontwikkelaars in staat te stellen een of alleen een paar groepen te selecteren en het script toestaan om automatisch de juiste Git-branch te selecteren.

kdesrc-build ondersteunt deze functie vanaf versie 1.16-pre2, via de optie [branch-group](#).

Example A.1 Voorbeeld van gebruikt van een branch-group

`branch-group` kunt u in het configuratiebestand als volgt gebruiken:

```
global
  # Selecteert KDE Frameworks 5 aen andere Qt5-gebaseerde apps
  branch-group kf5-qt5

  # Andere global opties hier ...
end global

module-set
  # branch-groep werkt alleen voor kde-projects
  repository kde-projects

  # branch-groep is doorgegeven van de set globally, maar kan ook
  # hier gespecificeerd.

  use-modules kdelibs kde-workspace
end module-set

# kdelibs's branch zal zijn "frameworks"
# kde-workspace's branch zal zijn "master" (vanaf Augustus 2013)
```

In dit geval geeft dezelfde `branch-group` voor elke Git-module verschillende branch-namen.

Voor deze functionaliteit moeten de KDE-ontwikkelaars in een Git repository genaamd `kde-build-metadata` wat data bijhouden, maar deze module zal automatisch door kdesrc-build (u kunt het zien verschijnen in de uitvoer van het script) worden meegeleverd.

TIP

KDE modules die nog geen naam hebben voor de branch group die u heeft gekozen, zullen standaard een toepasselijke naam voor de branch krijgen, als u nog geen `branch-group` heeft opgegeven.

Bijlage B

Achterhaalde procedures voor het instellen van profielen

B.1 Een KDE aanmeldprofiel instellen

Deze instructies beschrijven hoe u een profiel kan opstellen waarmee u zich kan aanmelden bij uw nieuw gebouwde KDE Plasma desktop. kdesrc-build zal normaal gesproken dit automatisch proberen te doen (zie Section 2.5.1). Deze appendix- sectie kan handig zijn voor personen die de functionaliteit van kdesrc-build voor het instellen van een inlog-profiel niet kunnen gebruiken. Maar deze instructies hoeven niet altijd up-to-date te zijn, het kan ook handig zijn om het bestand `kde-env-master.sh` te bestuderen wat met de kdesrc-build-broncode is meegeleverd.

B.1.1 Uw instellingen voor het opstartprofiel wijzigen

BELANGRIJK

Het `.bash_profile` is een bestand met de inloginstellingen voor de populaire bash shell in gebruik bij veel Linux[®]-distributies. Als u een andere shell gebruikt, dan moet u waarschijnlijk de in deze sectie gegeven voorbeelden aanpassen voor de door u gebruikte shell.

Open of creëer met uw favoriete editor in uw thuismap het bestand `.bash_profile`, en voeg aan het eind van het bestand toe: als u de qt module bouwt (standaard doet u dat), voeg in plaats daarvan dan toe:

```
QTDIR=(pad naar qtdir) # zoals standaard ~/kdesrc/build/qt .
KDEDIR=(pad naar kdedir) # zoals standaard ~/kde .
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH
MANPATH=$QTDIR/doc/man:$MANPATH

# Reageer op de juiste manier als LD_LIBRARY_PATH nog niet is ingesteld.
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
```

Handboek scripts voor kdesrc-build

, en als u niet qt bouwt (en in plaats daarvan uw systeem Qt™ gebruikt), voeg dan dit toe:

```
KDEDIR=(pad naar kdedir) # Zoals standaard ~/kde.
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH

# Reageer op de juiste manier als LD_LIBRARY_PATH nog niet is ingesteld.
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
```

Als u geen speciale gebruiker gebruikt, stel dan een andere \$KDEHOME in uw .bash_profile voor uw nieuwe omgeving in:

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

# Maak het indien nodig
[ ! -e ~/.kde-svn ] && mkdir ~/.kde-svn
```

OPMERKING

Als later uw K Menu leeg is of juist te vol is met toepassingen van uw distributie, dan moet u wellicht in uw .bash_profile de omgevingsvariabele XDG instellen :

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

B.1.2 KDE opstarten

Nu u de omgevingsvariabelen zodanig heeft ingesteld dat u de juiste KDE gebruikt, is het belangrijk dat u ook het juiste **startkde** script gebruikt.

Open in de thuishmap het tekstbestand .xinitrc of maak indien noodzakelijk deze aan. Voeg de regel toe:

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

BELANGRIJK

Bij sommige distributies, kan het nodig zijn om dezelfde stappen ook bij het bestand .xsession, ook in uw thuishmap, uit te voeren. Dit geldt met name als u grafische login managers zoals kdm, gdm, of xdm gebruikt.

Start nu uw gloednieuwe KDE op: in BSD en Linux® systemen met virtuele terminal support, u gebruikt sneltoetsen zoals **Ctrl+Alt+F1 ... Ctrl+Alt+F12** om naar de Virtuele Console 1 tot en met 12 om te schakelen. Hierdoor kunt u op hetzelfde moment meer dan een desktop-omgeving gebruiken. De eerste zes zijn tekst-terminals en de volgende zes zijn grafische vensters.

Als u uw computer wilt starten, dan krijgt u in plaats daarvan een grafische display manager te zien, u kunt de nieuwe KDE-omgeving gebruiken, zelfs als het niet als optie erbij staat. De meeste

Handboek scripts voor kdesrc-build

display managers, ook kdm, hebben bij het inloggen als optie om een 'Custom Session' te gebruiken. Bij deze optie, worden van het `.xsession`-bestand in uw thuismap uw sessie-instellingen geladen. Als u al zoals hierboven beschreven dit bestand heeft aangepast, dan zou deze optie u naar uw nieuwe KDE-installatie moeten leiden.

Als dat niet werkt, dan is er een andere manier die u kunt proberen die normaal gesproken zou moeten werken: Druk op **Ctrl+Alt+F2**, en u zal een tekst-terminalvenster te zien krijgen. Log in als de speciale gebruiker en type:

```
startx -- :1
```

TIP

U kunt tegelijk de nieuwe KDE en de oude KDE gebruiken! Log in als normale gebruiker, start de stabiele KDE desktop. Druk op **Ctrl+Alt+F2** (of **F1**, **F3**, enz..), en u krijgt een tekst-terminalvenster te zien. Log in als de speciale KDE Subversion-gebruiker en type:

```
startx -- :1
```

U kunt terug gaan naar de KDE-desktop van uw normale gebruiker door de sneltoets van een al opgestarte desktop in te drukken. Normaal gesproken is dit **Ctrl+Alt+F7**, maar misschien moet u in plaats daarvan **F6** of **F8** gebruiken. Om terug te gaan naar uw door kdesrc-build-gecompileerde KDE, moet u dezelfde stappen uitvoeren, maar dan met de volgende functietoets. Als u bijvoorbeeld **Ctrl+Alt+F7** moet invoeren om naar uw normale KDE om te schakelen, dan moet u **Ctrl+Alt+F8** invoeren om naar uw door kdesrc-build gebouwde KDE om te schakelen.