

Manuale dello script kdesrc-build

Michael Pyne

Carlos Woelz

Traduzione della documentazione in italiano: Federico Zenith



Manuale dello script kdesrc-build

Indice

1	Introduzione	8
1.1	Una breve introduzione a kdesrc-build	8
1.1.1	Cos'è kdesrc-build?	8
1.1.2	Uso di kdesrc-build in breve	8
1.2	Panoramica della documentazione	9
2	Per cominciare	10
2.1	Preparare il sistema per la generazione di KDE	10
2.1.1	Configurare un nuovo account utente	10
2.1.2	Assicurati che il tuo sistema sia pronto a generare il software di KDE	10
2.1.3	Configurare kdesrc-build	12
2.1.3.1	Installare kdesrc-build	12
2.1.3.2	Preparare il file di configurazione	12
2.1.3.2.1	Impostazione manuale del file di configurazione	12
2.2	Impostare i dati della configurazione	12
2.3	Usare lo script kdesrc-build	14
2.3.1	Caricare i metadati del progetto	14
2.3.2	Eeguire un'anteprima di ciò che avverrà quando avvierai kdesrc-build	14
2.3.3	Risoluzione di errori di generazione	15
2.4	Generazione di moduli specifici	16
2.5	Impostare l'ambiente per far funzionare il desktop Plasma di KDE	17
2.5.1	Installare automaticamente un driver d'accesso	18
2.5.1.1	Aggiungere il supporto per xsession in varie distribuzioni	18
2.5.1.2	Aggiungere il supporto per xsession manualmente	18
2.5.2	Configurare l'ambiente manualmente	19
2.6	Organizzazione e selezione dei moduli	19
2.6.1	Organizzazione dei programmi di KDE	19
2.6.2	Selezionare i moduli da generare	19
2.6.3	Insiemi di moduli	20
2.6.3.1	Il concetto di insieme di moduli di base	20
2.6.3.2	Supporto speciale per gli insiemi di moduli KDE	21
2.6.4	La banca dati dei moduli ufficiale di KDE	22
2.6.5	Scartare dei moduli del progetto KDE	23
2.7	Conclusioni del capitolo Per cominciare	24

3	Funzionalità dello script	25
3.1	Panoramica delle funzionalità	25
3.2	Registro della generazione di kdesrc-build	26
3.2.1	Panoramica del registro	26
3.2.1.1	Schema della cartella di registro	27
4	Configurare kdesrc-build	28
4.1	Panoramica della configurazione di kdesrc-build	28
4.1.1	Disposizione del file di configurazione	28
4.1.1.1	Configurazione globale	28
4.1.1.2	Configurazione dei moduli	28
4.1.1.3	Elaborazione dei valori delle opzioni	29
4.1.1.4	Gruppi 'options'	29
4.1.2	Includere altri file di configurazione	30
4.1.3	Opzioni di configurazione comunemente usate	30
4.2	Tabella delle opzioni di configurazione disponibili	31
5	Opzioni da riga di comando e variabili d'ambiente	57
5.1	Uso dalla riga di comando	57
5.1.1	Opzioni dalla riga di comando di uso comune	57
5.1.2	Specificare i moduli da generare	58
5.2	Variabili d'ambiente supportate	58
5.3	Parametri da riga di comando supportati	58
6	Usare kdesrc-build	66
6.1	Prefazione	66
6.2	Funzionalità di base di kdesrc-build	66
6.2.1	Supporto per qt	66
6.2.2	Flag standard aggiunti da kdesrc-build	67
6.2.3	Cambiare le priorità di generazione di kdesrc-build	67
6.2.4	Installazione come amministratore	68
6.2.5	Mostrare il progresso della generazione di un modulo	68
6.3	Funzionalità avanzate	69
6.3.1	Generare parzialmente un modulo	69
6.3.1.1	Rimuovere delle cartelle da una generazione	69
6.3.2	Supporto per rami e tag in kdesrc-build	69
6.3.2.1	Cosa sono i rami e i tag?	69
6.3.2.2	Come usare rami e tag	69
6.3.3	Fermare prima la generazione	70
6.3.3.1	La generazione normalmente prosegue anche se si verificano problemi	70

Manuale dello script kdesrc-build

6.3.3.2	Non fermare prima con --no-stop-on-failure	70
6.3.3.3	Fermare kdesrc-build con eleganza quando stop-on-failure è «false»	71
6.3.4	Come kdesrc-build cerca di far riuscire la generazione	71
6.3.4.1	Rigenerazioni automatiche	71
6.3.4.2	Rigenerare un modulo manualmente	71
6.3.5	Cambiare le impostazioni delle variabili d'ambiente	72
6.3.6	Riprendere le generazioni	72
6.3.6.1	Riprendere una generazione non riuscita o annullata	72
6.3.6.2	Ignorare dei moduli in una generazione	73
6.3.7	Cambiare le opzioni dalla riga di comando	73
6.3.7.1	Cambiare le opzioni globali	73
6.3.7.2	Cambiare le opzioni dei moduli	73
6.4	Funzionalità per gli sviluppatori di KDE	73
6.4.1	Controlli dell'agente SSH	73
6.5	Altre funzionalità di kdesrc-build	74
6.5.1	Cambiare la quantità di output da kdesrc-build	74
6.5.2	Output colorato	74
6.5.3	Rimuovere cartelle non necessarie dopo la generazione	74
7	CMake, il sistema di generazione di KDE	76
7.1	Introduzione a CMake	76
8	Riconoscimenti e licenza	77
A	Moduli di KDE e organizzazione del codice sorgente	78
A.1	Il 'modulo'	78
A.1.1	Moduli singoli	78
A.1.2	Gruppi di moduli collegati	78
A.1.3	'Gruppi di rami' dei moduli	79
B	Procedure sostituite di configurazione dei profili	80
B.1	Configurare in profilo d'accesso di KDE	80
B.1.1	Cambiare le impostazioni del profilo di avvio	80
B.1.2	Avviare KDE	81

Elenco delle tabelle

4.1	Opzioni solo per gli ambiti globali	36
4.2	Opzioni di tutti gli ambiti (modulo, insieme di moduli e globali)	53
4.3	Opzioni per gli ambiti globali e l'insieme di moduli	56

Sommario

kdesrc-build è uno script che genera e installa il software di KDE direttamente dai depositi del codice sorgente di KDE.

Capitolo 1

Introduzione

1.1 Una breve introduzione a kdesrc-build

1.1.1 Cos'è kdesrc-build?

kdesrc-build è uno script che aiuta la comunità di KDE a installare il software [KDE](#) dai depositi dei sorgenti [Git](#) e a continuare ad aggiornarlo dopo l'installazione. È rivolto in particolare a coloro i quali hanno bisogno di supporto nel test e nello sviluppo del software KDE, inclusi gli utenti che verificano le correzioni di errori e gli sviluppatori che lavorano su nuove funzionalità.

Lo script kdesrc-build può essere usato per mantenere un solo modulo individuale, un desktop Plasma completo con un set di applicazioni KDE, o anche una via di mezzo.

Per iniziare, consulta [capitolo 2](#) o continua a leggere qui i dettagli su come kdesrc-build funziona e gli argomenti trattati in questa documentazione.

1.1.2 Uso di kdesrc-build in breve

kdesrc-build funziona usando gli strumenti disponibili sulla riga di comando e le stesse interfacce disponibili agli utenti. All'esecuzione di kdesrc-build, si segue questa sequenza:

1. kdesrc-build legge nella [riga di comando](#) e nel [file di configurazione](#), per determinare cosa generare, quali opzioni di compilazione usare, dove installare, eccetera.
2. kdesrc-build aggiorna il codice sorgente di ogni [modulo](#). I moduli che non vengono aggiornati correttamente di solito non interrompono la generazione, e ti verrà notificato alla fine quali moduli non sono stati aggiornati.
3. I moduli che sono aggiornati correttamente sono generati, verificati e poi installati. Per ridurre il tempo trascorso totale, per impostazione predefinita kdesrc-build avvia la generazione del codice appena il primo modulo ha completato l'aggiornamento, e permette ai restanti aggiornamenti di continuare in background.

SUGGERIMENTO

Un'*ottima* panoramica su come vengono generati i moduli KDE, che comprende anche grafici informativi, si trova in [un articolo che discute dell'applicazione Krita di KDE](#) (in inglese). Questo flusso di lavoro è appunto quello che kdesrc-build automatizza per tutti i moduli KDE.

1.2 Panoramica della documentazione

Questa guida è una panoramica per descrivere i seguenti aspetti dell'operazione di kdesrc-build:

- Una [panoramica](#) dei passi necessari per cominciare;
- [Funzionalità](#) importanti;
- La sintassi e le opzioni del [file di configurazione](#);
- Le [opzioni dalla riga di comando](#).

Sono documentati anche i passi da eseguire con altri strumenti, cioè i passi non eseguiti automaticamente da kdesrc-build.

Capitolo 2

Per cominciare

In questo capitolo, mostriamo come usare `kdesrc-build` per ritirare i moduli dal deposito di KDE e generarli. Diamo anche una rapida spiegazione della struttura del codice sorgente di KDE e di cosa devi fare prima di eseguire lo script.

Tutti gli argomenti qui presenti sono trattati in maggiore dettaglio nell'articolo [Build from Source](#), nel [wiki della Comunità di KDE](#). Se stai compilando KDE per la prima volta, è una buona idea leggerlo, o almeno consultarlo come riferimento. Troverai informazioni dettagliate sugli strumenti di gestione dei pacchetti e su requisiti, errori di compilazione comuni, strategie e informazioni su come far funzionare la nuova installazione di KDE.

2.1 Preparare il sistema per la generazione di KDE

2.1.1 Configurare un nuovo account utente

Si raccomanda di usare un diverso account per generare, installare ed eseguire il software di KDE, perché servono meno permessi e per evitare di interferire con i pacchetti della tua distribuzione. Se hai già installato i pacchetti di KDE, la cosa migliore sarebbe creare un altro utente (apposito) per generare ed eseguire il nuovo KDE.

SUGGERIMENTO

Lasciare intatto il tuo sistema KDE permette anche di avere un ripiego d'emergenza nel caso in cui un errore di codice renda inutilizzabile la tua ultima generazione del software.

Puoi anche configurare un'installazione in una cartella globale (per es. `/usr/src/local`), se vuoi. Questo documento non tratta questo tipo di installazione, dato che presupponiamo che tu sappia quello che stai facendo.

2.1.2 Assicurati che il tuo sistema sia pronto a generare il software di KDE

Prima di usare lo script `kdesrc-build` (o qualsiasi altra strategia di generazione) devi installare gli strumenti di sviluppo e le librerie necessarie a KDE. L'elenco quasi completo degli strumenti necessari può essere recuperato dalla pagina [Build Requirements del wiki della Comunità di KDE](#).

Ecco un elenco di alcune delle cose che ti serviranno:

Manuale dello script kdesrc-build

- Ti servirà CMake, il software che KDE usa per gestire la configurazione della generazione del codice sorgente e la generazione dei comandi di compilazione specifici per il tuo sistema. La versione richiesta varia a seconda delle versioni del software KDE che stai generando (per le specifiche consulta la TechBase), ma con le moderne distribuzioni il CMake incluso nella tua dovrebbe andare bene.
- Devi anche installare i client di controllo sorgenti necessari per eseguire il checkout del codice sorgente di KDE. Ciò vuol dire che ti serve almeno quanto segue:
 - Il gestore di controllo dei sorgenti [Git](#), utilizzato per tutto il [codice sorgente](#) di KDE
 - Anche se non è un requisito, il gestore di controllo dei sorgenti [Bazaar](#) viene usato per un solo modulo (libdbusmenu-qt) richiesto per le librerie di KDE. La maggior parte degli utenti può installare questa libreria con i loro sistemi di distribuzione dei pacchetti, ma kdesrc-build ne supporta la generazione, se lo vuoi. Per generare libdbusmenu-qt, però, dovrai installare Bazaar.
- Per kdesrc-build è richiesto il linguaggio di script Perl, alcuni depositi KDE e Qt™ (se compili da sorgente).

Il Perl incluso nella tua distribuzione dovrebbe andare bene (è necessario almeno Perl 5.14), ma avrai bisogno anche di alcuni moduli aggiuntivi (kdesrc-build ti avviserà se non sono presenti):

 - IO::Socket::SSL
 - JSON::PP o JSON::XS
 - YAML::PP, YAML::XS, o YAML::Syck
- Avrai bisogno di un ambiente di sviluppo C++ completo (compilatore, libreria standard, runtime e tutti i pacchetti di sviluppo richiesti). Le versioni richieste minime variano a seconda del modulo KDE: la raccolta KDE Frameworks 5 supporta i compilatori più datati, mentre KDE Plasma 5 e KDE Applications richiedono in genere compilatori più recenti.

I compilatori minimi raccomandati sono GCC 4.8 o Clang 4. Molte distribuzioni supportano un'installazione facile di questi strumenti tramite il pacchetto 'build-essentials', un'opzione per installare «dipendenze di compilazione» con Qt™, o funzionalità simili. Il wiki KDE Community ha una pagina [che riassume i pacchetti raccomandati per le distribuzioni principali](#) (in inglese).
- Avrai bisogno di uno strumento di compilazione che esegue realmente i passaggi della compilazione (come generati da CMake). È raccomandato GNU Make e deve essere disponibile tramite il tuo gestore di pacchetti. CMake non supporta altre opzioni, come lo strumento di generazione Ninja, che può essere utilizzato da kdesrc-build tramite l'opzione del file di configurazione [custom-build-command](#).
- Infine, avrai bisogno delle librerie Qt™ corrette (pacchetti di sviluppo inclusi) per la versione del software KDE che stai generando. kdesrc-build non supporta ufficialmente la generazione di Qt™ 5 (l'attuale versione principale), dunque consigliamo di usare i pacchetti di sviluppo della tua distribuzione o di consultare la pagina del wiki KDE Community [self-building Qt 5](#).

NOTA

La maggior parte delle distribuzioni di sistemi operativi ha un modo di installare facilmente gli strumenti di sviluppo necessari. Consulta la pagina wiki della Comunità [Required devel packages](#) per vedere se queste istruzioni sono già disponibili.

IMPORTANTE

Alcuni di questi pacchetti sono divisi in librerie (o programmi) e pacchetti di sviluppo. Ti serviranno almeno il programma o la libreria, e il suo pacchetto di sviluppo.

2.1.3 Configurare kdesrc-build

2.1.3.1 Installare kdesrc-build

Gli sviluppatori di KDE apportano modifiche frequenti al file kdesrc-build in modo da mantenerlo sincronizzato con i cambiamenti ai processi di sviluppo in KDE, incluse migliorie alla configurazione kdesrc-build raccomandata, moduli aggiunti, migliorie dei flag CMake, ecc.

A causa di questo, consigliamo di recuperare kdesrc-build direttamente dal deposito del suo codice sorgente e di aggiornarlo con frequenza.

Puoi recuperare kdesrc-build dal deposito del suo codice sorgente eseguendo:

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

Sostituisci ~/kdesrc-build con la cartella dove lo vuoi installare.

Puoi aggiornare kdesrc-build in seguito eseguendo:

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

SUGGERIMENTO

Raccomandiamo di aggiungere la cartella di installazione di kdesrc-build alla variabile di ambiente PATH, in modo da poter avviare kdesrc-build senza dover digitare ogni volta il suo percorso completo.

2.1.3.2 Preparare il file di configurazione

kdesrc-build usa un [file di configurazione](#) per controllare quali moduli sono costruiti, dove sono installati, ecc. Questo file si trova in ~/.config/kdesrc-buildrc (\$XDG_CONFIG_HOME/kdesrc-buildrc, se \$XDG_CONFIG_HOME è impostato).

Puoi usare kdesrc-build --generate-config per preparare una semplice configurazione di kdesrc-build. Puoi quindi modificare il file di configurazione ~/.config/kdesrc-buildrc ottenuto per fare le modifiche che ti servono.

2.1.3.2.1 Impostazione manuale del file di configurazione

Puoi anche impostare manualmente il tuo file di configurazione, copiando il file di configurazione di esempio incluso kdesrc-buildrc-kf5-sample in ~/.config/kdesrc-buildrc e poi modificando il file. [capitolo 4](#) sarà un utile riferimento per questo, in particolare la sua [tabella delle opzioni di configurazione](#).

kdesrc-build contiene molti file di configurazione consigliati per il supporto di KDE Frameworks 5, Plasma 5, e altre applicazioni KDE. Consulta Sezione [4.1.2](#) per informazioni su come usare altri file di configurazione dalla tua copia di kdesrc-buildrc.

Puoi trovare maggiori informazioni sulla sintassi del [file di configurazione](#) nel [Sezione 2.2](#) e nel [capitolo 4](#).

2.2 Impostare i dati della configurazione

Per usare kdesrc-build, dovresti avere un file nella tua cartella ~/.config (o in \$XDG_CONFIG_HOME, se impostato) chiamato .kdesrc-buildrc, che imposta le opzioni generali e specifica i moduli che vuoi scaricare e generare.

NOTA

È possibile usare diversi file di configurazione per kdesrc-build, come descritto nel capitolo 4. Se hai bisogno di usare configurazioni multiple, vedi quella sezione. Qui, presumeremo che la configurazione sia salvata in `~/ .config/kdesrc-buildrc`.

Il modo più semplice di procedere è usare il file `kdesrc-buildrc-kf5-sample` come modello, cambiando le opzioni globali per adattarle alle tue esigenze e cambiando anche l'elenco dei moduli da generare.

Le impostazioni predefinite dovrebbero essere appropriate per generare KDE. Alcune impostazioni che potresti voler cambiare includono:

- `kdedir`, che cambia la cartella di destinazione in cui il tuo software KDE sarà installato. Il valore predefinito è `~/kde/usr`, che è un'installazione per un solo utente.
- `branch-group`, che puoi usare per scegliere il ramo di sviluppo appropriato per tutti i moduli KDE nel loro insieme. Sono supportate molte configurazioni di generazione ma probabilmente sceglierai `kf5-qt5`, dunque kdesrc-build scaricherà l'ultimo codice basato su Qt™ 5 e KDE Frameworks 5.

SUGGERIMENTO

Se non ne scegli uno, kdesrc-build utilizzerà un gruppo di rami predefinito, ma questo cambierà nel tempo; è dunque meglio sceglierne uno in modo che il gruppo di rami non cambi in modo inaspettato.

- `source-dir`, per controllare la cartella di kdesrc-build utilizza per scaricare il codice sorgente, avviando il processo di generazione e salvando i registri. Questo è impostato in modo predefinito in `~/kde/src`.
- `cmake-options`, che imposta le opzioni da passare al comando CMake quando si generano i moduli. Tipicamente questo è utilizzato per impostare uno tra i modelli di generazione 'debug' o 'release', per abilitare (o disabilitare) le funzioni opzionali o per passare informazioni al processo di generazione riguardo la posizione delle librerie richieste.
- `make-options`, che imposta le opzioni utilizzate quando si esegue realmente il comando make per generare ogni modulo (una volta che CMake ha stabilito il sistema di generazione).

L'opzione più tipica è `-jN`, dove *N* deve essere sostituito col numero massimo di operazioni di compilazione che vuoi consentire. Un numero più alto (fino al numero di CPU logici disponibili nel tuo sistema) porta a generazioni più rapide ma richiede maggiore impegno di risorse del sistema.

SUGGERIMENTO

kdesrc-build imposta l'opzione `num-cores` al numero rilevato di core di elaborazione disponibili. Puoi utilizzare questo valore nel tuo file di configurazione in modo da evitare di impostarlo manualmente.

Example 2.1 Configurare Make per utilizzare tutte le CPU disponibili, con eccezioni

```
global
# Questa variabile d'ambiente viene utilizzata automaticamente da make, ←
  inclusi
# i comandi make non eseguiti direttamente da kdesrc-build, come il ←
  configure di 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 # Numero ridotto di operazioni di generazione solo per ←
    questi moduli
end module-set
```

NOTA

Alcuni depositi Git molto grandi potrebbero affossare il tuo sistema se provi a compilare con troppe operazioni di generazione simultanee, in particolare i depositi tipo Qt™ WebKit e Qt™ WebEngine. Per mantenere l'interattività del sistema potresti dover ridurre il numero di operazioni di generazione per i moduli specifici. Esempio 2.1 fornisce un esempio su come farlo.

Potresti voler selezionare diversi moduli da generare, come descritto in Sezione 2.6.2.

2.3 Usare lo script kdesrc-build

Una volta stabiliti i dati di configurazione, sei pronto per avviare lo script. Anche se devi ancora fare alcune regolazioni o approfondire alcune letture, è una buona idea caricare almeno i metadati del progetto KDE.

2.3.1 Caricare i metadati del progetto

Da una finestra di terminale, accedi come l'utente che vuoi usare per compilare il software KDE ed esegui lo script:

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

Questo comando imposta la cartella sorgente e connette ai depositi Git di KDE per scaricare la relativa banca dati e la banca dati dei metadati delle dipendenze, senza eseguire ulteriori modifiche. È bene avviarlo separatamente dato che questi metadati sono utili per altri comandi kdesrc-build.

2.3.2 Eseguire un'anteprima di ciò che avverrà quando avvierai kdesrc-build

Con i metadati del progetto installati, è possibile fare un'anteprima di cosa farà kdesrc-build una volta avviato. Ciò può essere fatto con l'opzione a riga di comando `--pretend`.

Manuale dello script kdesrc-build

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

Dovrebbe comparire un messaggio che ti avvisa che alcuni pacchetti sono stati generati correttamente (sebbene niente di ciò sia stato realmente fatto). Se non compaiono problemi seri, puoi procedere all'avvio reale dello script.

```
% kdesrc-build
```

Questo comando scaricherà il codice sorgente corretto, compilerà e installerà in ordine ciascun modulo. Successivamente dovresti osservare un output simile a quello in Esempio 2.2:

Example 2.2 Output esemplificativo di un'esecuzione di 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 Risoluzione di errori di generazione

A seconda di quanti moduli stai scaricando, è possibile che kdesrc-build non riesca la prima volta che compili il software KDE. Non disperare!

kdesrc-build registra l'output di ogni comando che esegue. Come impostazione predefinita, i file di registro sono tenuti in `~/kdesrc/log`. Per vedere cosa ha causato un errore in un modulo l'ultima volta che si è avviato kdesrc-build, è di solito sufficiente guardare in `~/kdesrc/log/latest/nome-modulo/error.log`.

SUGGERIMENTO

Il modo più facile di trovare quale errore ha causato l'interruzione della generazione di un modulo è forse cercare partendo dalla fine, senza distinguere le maiuscole, la parola `error`. Una volta trovata, scorri in alto per assicurarti che non ci siano altri messaggi d'errore nei paraggi. Il primo messaggio d'errore in un gruppo è di solito il problema fondamentale.

In quel file, vedrai l'errore che ha impedito la generazione del modulo. Se il file dice (in fondo) che ti mancano dei pacchetti, prova ad installarli (includendo i pacchetti `-dev` appropriati) prima di generare quel modulo. Assicurati, quando fai ripartire `kdesrc-build`, di passare l'opzione `--reconfigure` in modo che `kdesrc-build` costringa il modulo a ricontrollare i pacchetti mancanti.

Se l'errore sembra essere un errore di generazione (come un errore di sintassi, 'prototipo errato', 'tipo sconosciuto'), è probabilmente un errore del codice sorgente di KDE, che si spera venga risolto nel giro di pochi giorni. Se non viene risolto, manda pure un messaggio alla lista kde-devel@kde.org (è possibile ti venga richiesto di iscriverti) per segnalare l'errore di generazione.

Puoi trovare altri esempi comuni di cose che possono andare storte e le relative soluzioni, oltre a suggerimenti e strategie generali per generare software KDE, nella guida [Build from Source](#) (in inglese).

Invece, se tutto è andato bene, dovresti avere un nuovo KDE installato sul computer, e ora è solo questione di farlo partire, come descritto nella Sezione [2.5](#).

NOTA

Per maggiori informazioni sulle funzionalità di registro di `kdesrc-build`, vedi Sezione [3.2](#).

2.4 Generazione di moduli specifici

Piuttosto che generare tutte le volte ciascun modulo, potresti voler generare solo un singolo modulo o altri piccoli sottoinsiemi. Anziché modificare il tuo file di configurazione, puoi semplicemente passare i nomi dei moduli o gli insiemi di moduli da generare alla riga di comando.

Example 2.3 Output esemplificativo della generazione di un modulo specifico in kdesrc-build

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

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

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

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

  ...

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

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

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

In questo caso, sebbene sia stata specificata solo l'applicazione *dolphin*, il flag `--include-dependencies` ha indicato a `kdesrc-build` di includere le dipendenze elencate per *dolphin* (impostando l'opzione [include-dependencies](#)).

NOTA

La risoluzione delle dipendenze in questo caso ha funzionato solo perché *dolphin* risultava specificato nell'insieme di moduli basato su `kde-projects` (in questo esempio chiamato `base-apps`). Vedi Sezione [2.6.3.2](#).

2.5 Impostare l'ambiente per far funzionare il desktop Plasma di KDE

Supponendo che tu stia usando un utente apposito per generare Plasma di KDE, e che abbia già installato una versione di Plasma, far funzionare il nuovo Plasma potrebbe essere un po' difficile, perché il nuovo deve prendere la precedenza sul vecchio. Devi cambiare le variabili d'ambiente dei tuoi script d'accesso per assicurarti che venga usato il desktop appena generato.

2.5.1 Installare automaticamente un driver d'accesso

A partire dalla versione 1.16, kdesrc-build cercherà di installare un driver d'accesso appropriato che permetterà di accedere al desktop KDE generato da kdesrc-build dal gestore degli accessi. Questo può essere disabilitato con l'opzione `install-session-driver` nel file di configurazione.

NOTA

La configurazione della sessione non avviene quando kdesrc-build è eseguito in modalità di prova (opzione `--pretend`).

Questo driver funziona impostando un tipo di sessione 'xsession' personalizzato. Questo tipo di sessione dovrebbe normalmente funzionare con il gestore d'accesso sddm (dove compare come la sessione 'Custom'), ma altri gestori (come LightDM e gdm) potrebbero richiedere l'installazione di altri file per abilitare il supporto per xsession.

2.5.1.1 Aggiungere il supporto per xsession in varie distribuzioni

I gestori d'accesso predefiniti per alcune distribuzioni potrebbero richiedere l'installazione di pacchetti aggiuntivi per poter supportare gli accessi con xsession.

- La distribuzione Linux® [Fedora](#) richiede l'installazione del pacchetto `xorg-x11-xinit-session` per il supporto degli accessi con xsession personalizzati.
- [Debian](#) e le distribuzioni Linux® da esso derivate dovrebbero supportare gli accessi con xsession personalizzati, ma richiedono l'impostazione dell'opzione `allow-user-xsession` nel file `/etc/X11/Xsession.options`. Vedi anche la [documentazione su come configurare la sessione X](#) di Debian.
- Per le altre distribuzioni, vai alla Sezione [2.5.1.2](#).

2.5.1.2 Aggiungere il supporto per xsession manualmente

Se non ci sono istruzioni specifiche per la tua distribuzione nella Sezione [2.5.1.1](#), puoi aggiungere manualmente una voce per l'accesso con xsession personalizzato' all'elenco di tipi di sessione della tua distribuzione come segue:

NOTA

Questa procedura probabilmente richiederà privilegi amministrativi per essere effettuata.

1. Crea il file `/usr/share/xsessions/kdesrc-build.desktop`.
2. Assicurati che il file appena creato contenga il quanto segue:

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

- La voce `$HOME` deve essere sostituita dal percorso completo alla tua cartella HOME (per esempio, `/home/utente`). La specifica delle voci desktop non permette file generici per tutti gli utenti.
3. Al riavvio del gestore degli accessi, dovrebbe essere mostrato un nuovo tipo di sessione nell'elenco, 'KDE Plasma Desktop (unstable; kdesrc-build)', che dovrebbe cercare di eseguire il file `.xsession` installato da kdesrc-build se viene selezionato all'accesso.

NOTA

La cosa più facile potrebbe essere riavviare il computer per riavviare il gestore degli accessi, se il gestore non tiene traccia degli aggiornamenti alla cartella `/usr/share/xsessions`.

2.5.2 Configurare l'ambiente manualmente

Questa documentazione includeva precedentemente istruzioni su quali variabili d'ambiente impostare per caricare il desktop appena generato. Queste istruzioni sono state spostate nell'appendice (Sezione B.1).

Se vuoi configurare il tuo supporto per l'accesso, puoi consultare questa appendice o il file `kde-env-master.sh.in` incluso nel codice sorgente di `kdesrc-build`.

2.6 Organizzazione e selezione dei moduli

2.6.1 Organizzazione dei programmi di KDE

I programmi di KDE sono divisi in diversi componenti, molti dei quali possono essere generati da `kdesrc-build`. Comprimerne l'organizzazione aiuterà a selezionare correttamente i moduli che vuoi generare.

1. Al livello più basso c'è la libreria Qt™, che è una serie di strumenti molto potenti e multiplatforma. KDE si basa su Qt™, e anche alcune delle librerie esterne richieste da KDE si basano su Qt™. `kdesrc-build` può generare Qt™ o usarne una versione già installata sul tuo sistema se è abbastanza recente.
2. Sopra Qt™ ci sono le librerie necessarie al funzionamento dei programmi di KDE. Alcune di queste non sono considerate parte di KDE per la loro natura generica, ma sono comunque essenziali alla piattaforma di KDE. Queste librerie sono raccolte in un raggruppamento di moduli `kdesupport` ma non sono considerate parte delle librerie 'Frameworks'.
3. Sopra queste librerie essenziali si trova [KDE Frameworks](#), di solito abbreviato in KF5, composto dalle librerie essenziali per il desktop KDE Plasma, KDE Applications e altro software di terze parti.
4. Sopra Frameworks ci sono diverse cose:
 - Applicazioni di 'terze parti'. Queste sono applicazioni che usano KDE Frameworks o sono progettate per funzionare sotto KDE Plasma ma non sono scritte da o associate al progetto KDE.
 - Plasma, che è un ambiente desktop completo basato sugli 'spazi di lavoro'. Questo è quello che gli utenti vedono quando 'accedono a KDE'.
 - La suite KDE Application. Questa è una raccolta di software incluso con la piattaforma e il desktop Plasma, raggruppato in moduli individuali, incluso i programmi di utilità come Dolphin, giochi come KSudoku e software per la produttività, come Kontact, rilasciato da KDE.
 - Infine, c'è una raccolta di programmi (anch'essi organizzati in moduli) il cui sviluppo è supportato dalle risorse di KDE (traduzione, controllo del codice sorgente, tracciamento degli errori, eccetera) ma non è rilasciato da KDE come parte di Plasma o della suite Applications. Questi moduli sono noti come 'Extragear'.

2.6.2 Selezionare i moduli da generare

La selezione di quali generare tra i moduli possibili è controllata dal [file di configurazione](#). Dopo la sezione `global` c'è un elenco di moduli da generare, delimitato da righe `module` e `end module`. Una voce esemplificativa per un modulo viene mostrata nell'Esempio 2.4.

Example 2.4 Voce esemplificativa di un modulo nel file di configurazione

```
module kdesrc-build-git
  # Le opzioni del modulo vanno qui, per esempio:
  repository kde:kdesrc-build
  make-options -j4 # Esegue 4 compilazioni alla volta
end module
```

NOTA

Nella pratica, il costrutto di questo modulo non viene in genere usato direttamente. La maggior parte dei moduli, al contrario, è specificata tramite gruppi di moduli come descritto più avanti.

Quando usi solo le voci `module`, `kdesrc-build` le genera nell'ordine elencato e non tenta di scaricare altri depositi oltre quelli che hai direttamente specificato.

2.6.3 Insiemi di moduli

Il codice sorgente di KDE è decomposto in un gran numero di depositi basati su Git relativamente piccoli. Per rendere più semplice la gestione del gran numero di depositi coinvolti in una qualsiasi installazione utile basata su KDE, `kdesrc-build` supporta il raggruppamento multiplo dei moduli e tratta il gruppo come un 'insieme di moduli'.

2.6.3.1 Il concetto di insieme di moduli di base

Usando un insieme di moduli, puoi dichiarare rapidamente molti moduli di Git da scaricare e generare, come se avessi scritto una dichiarazione del modulo a sé per ciascuno. L'opzione `repository` viene gestita specificatamente per configurare da dove ogni modulo viene scaricato, e ciascuna altra opzione dell'insieme di moduli viene copiata in ogni modulo generato in questo modo.

Example 2.5 Usare gli insiemi di moduli

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

module qt
  # Opzioni rimosse per brevità
end module

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

# Altri moduli secondo necessità...
module kdesupport
end module
```

Nel Esempio 2.5 viene mostrato un breve insieme di moduli. Quando `kdesrc-build` lo rileva, si comporta come se, per ogni modulo dato sotto `use-modules`, sia dichiarato un modulo a sé, con

un deposito (*repository*) uguale a quello dell'insieme di moduli subito seguito dal nome del modulo dato.

Inoltre, si possono passare altre opzioni in un insieme di moduli, le quali vengono copiate a ciascun modulo creato in questo modo. Usando `module-set` è possibile creare rapidamente molti moduli Git tutti basati sullo stesso URL del deposito. Inoltre è possibile dare agli insiemi un nome (come mostrato nell'esempio), che permette di riferirsi a tutto il gruppo di moduli dalla riga di comando.

2.6.3.2 Supporto speciale per gli insiemi di moduli KDE

Il supporto per gli insiemi di moduli descritto finora è generico per qualsiasi modulo basato su Git. Per i depositi Git di KDE `kdesrc-build` include funzioni aggiuntive che facilitano il lavoro per gli utenti e gli sviluppatori. Questo supporto è abilitato specificando `kde-projects` come *repository* per l'insieme di moduli.

`kdesrc-build` normalmente genera solo i moduli elencati nel file di configurazione, nell'ordine che hai stabilito. Ma con un insieme di moduli `kde-projects`, `kdesrc-build` può eseguire una risoluzione delle dipendenze dei moduli specifici di KDE e, in aggiunta, includere automaticamente i moduli nella generazione, anche se specificati solo indirettamente.

Example 2.6 Usare gli insiemi di moduli di `kde-projects`

```
# Aggiunge solo un modulo per juk (il deposito kde/kdemultimedia/juk)
module-set juk-set
  repository kde-projects
  use-modules juk
end module-set

# Aggiunge tutti i moduli presenti in kde/multimedia/*, incluso juk,
# ma nessun'altra dipendenza
module-set multimedia-set
  repository kde-projects
  use-modules kde/multimedia
end module-set

# Aggiunge tutti i moduli presenti in kde/multimedia/*, e tutte le ←
# dipendenze
# kde-projects che si trovano fuori da kde/kdemultimedia
module-set multimedia-deps-set
  repository kde-projects
  use-modules kde/multimedia
  include-dependencies true
end module-set

# Tutti i moduli creati al di fuori di questi tre gruppi di moduli sono ←
# automaticamente inseriti
# nell'ordine di dipendenza corretto, a prescindere dalle impostazioni di ←
include-dependencies
```

SUGGERIMENTO

Questo costrutto di insiemi di moduli `kde-projects` è il metodo principale per dichiarare quali moduli vuoi generare.

Tutti gli insiemi di moduli utilizzano le opzioni `repository` e `use-modules`. Gli insiemi di moduli `kde-projects` hanno un valore `repository` predefinito, ma gli altri tipi di insiemi utilizzeranno anche l'opzione `git-repository-base`.

2.6.4 La banca dati dei moduli ufficiale di KDE

I depositi Git di KDE consentono il raggruppamento dei moduli collegati a Git all'interno di collezioni di moduli collegati (per es. `kdegraphics`). Git non riconosce questi raggruppamenti, ma `kdesrc-build` lo può fare utilizzando gli **insiemi di moduli** con un'opzione `repository` impostata a `'kde-projects'`.

`kdesrc-build` riconoscerà che il deposito `kde-projects` va trattato in modo speciale, e modificherà il processo di generazione conseguentemente. Tra le altre cose, `kdesrc-build`:

- Scaricherà la più recente banca dati dei moduli dall'[archivio git di KDE](#).
- Cercherà di trovare in quella banca dati un modulo con il nome dato nell'impostazione `use-modules` dell'insieme di moduli.
- Per ciascun modulo trovato `kdesrc-build` cercherà nella banca dati il corretto deposito, basato sull'impostazione `branch-group` in vigore. Se per il gruppo del ramo esiste un deposito ed è attivo, `kdesrc-build` utilizzerà automaticamente quello per scaricare o aggiornare il codice sorgente.

NOTA

Nella banca dati attuale, alcuni gruppi di moduli non solo hanno una raccolta di moduli, ma dichiarano *anche* il loro deposito Git. In queste situazioni, `kdesrc-build` al momento preferisce il deposito Git del gruppo invece di includere i depositi dei figli.

L'esempio seguente mostra come usare la banca dati dei moduli di KDE per installare la libreria multimediale Phonon.

```
module-set media-support
  # Questa opzione deve essere kde-projects per usare la banca dati dei ←
  moduli.
  repository kde-projects

  # Questa opzione sceglie quali moduli cercare nella banca dati.
  use-modules phonon/phonon phonon-gstreamer phonon-vlc
end module-set
```

SUGGERIMENTO

Si usa `phonon/phonon` perché (con la presente banca dati dei progetti) `kdesrc-build` dovrebbe altrimenti decidere tra il gruppo di progetti chiamato `'phonon'` e il progetto individuale omonimo. Attualmente `kdesrc-build` sceglierebbe il primo, che genererebbe molti più motori del necessario.

L'esempio seguente è forse più realistico, e mostra una funzionalità disponibile solo con la banca dati dei moduli di KDE: generare tutte le applicazioni grafiche di KDE con una sola dichiarazione.

```
module-set kdegraphics
  # Questa opzione deve essere kde-projects per usare la banca dati dei ←
  moduli.
  repository kde-projects
```

```
# Questa opzione sceglie quali moduli cercare nella banca dati.
use-modules kdegraphics/libs kdegraphics/*
end module-set
```

Due importanti capacità sono qui dimostrate:

1. kdesrc-build permette di specificare moduli discendenti di un dato modulo, senza generare il modulo antenato, usando la sintassi **nome-modulo /***. In questo caso è anche necessario, perché il modulo base, kdegraphics, è indicato come disattivato in modo che non sia generato per sbaglio con i suoi moduli figli. Specificare i moduli discendenti permette a kdesrc-build di saltare il modulo disattivato.
2. kdesrc-build, inoltre, non aggiungerà un certo modulo all'elenco di generazione più di una volta. Questo permette di impostare manualmente di generare prima kdegraphics/libs, prima del resto di kdegraphics, senza prima cercare di generare due volte kdegraphics/libs. Ciò era di solito richiesto per la gestione corretta delle dipendenze, e oggi resta un'opzione di ripiego nel caso in cui la banca dati del progetto KDE non trovi i metadati delle dipendenze.

2.6.5 Scartare dei moduli del progetto KDE

Potresti decidere di voler generare tutti i programmi di un certo gruppo di moduli di KDE *tranne* uno in particolare.

Per esempio, il gruppo kdeutils contiene un programma chiamato kremotecontrol. Se la tua macchina non ha l'hardware necessario per ricevere i segnali inviati da un telecomando, potresti decidere di non voler scaricare, generare e installare kremotecontrol ogni volta che aggiorni kdeutils.

Puoi farlo utilizzando l'opzione di configurazione [ignore-modules](#). Sulla riga di comando l'opzione `--ignore-modules` esegue la stessa cosa, ma è più comoda per filtrare un modulo solo una volta.

Example 2.7 Esempio per ignorare un modulo del progetto KDE in un gruppo

```
module-set utils
  repository kde-projects

  # Questa opzione sceglie quali moduli cercare nella banca dati.
  use-modules kdeutils

  # Questa opzione «sottrae» i moduli da quelli scelti da use-modules ←
  sopra.
  ignore-modules kremotecontrol
end module-set

module-set graphics
  repository kde-projects

  # Questa opzione sceglie quali moduli cercare nella banca dati.
  use-modules extragear/graphics

  # Questa opzione «sottrae» i moduli da quelli scelti da use-modules ←
  sopra.
  # In questo caso, sia extragear/graphics/kipi-plugins che
  # extragear/graphics/kipi-plugins/kipi-plugins-docs vengono ignorati.
  ignore-modules extragear/graphics/kipi-plugins
end module-set
```

2.7 Conclusioni del capitolo Per cominciare

Queste sono le funzionalità e i concetti principali necessari per iniziare a usare kdesrc-build

Per informazioni aggiuntive, puoi continuare a leggere questa documentazione. In particolare, sono riferimenti utili l'[elenco delle opzioni della riga di comando](#) e la [tabella delle opzioni del file di configurazione](#).

La comunità KDE mantiene anche una [guida di riferimento in linea su wiki su come compilare il codice sorgente](#) (in inglese), che tratta kdesrc-build e include suggerimenti e altre linee guida su come utilizzare questo strumento.

Capitolo 3

Funzionalità dello script

3.1 Panoramica delle funzionalità

Le funzionalità di kdesrc-build includono:

- Puoi 'far finta' di fare le operazioni. Se passi le opzioni `--pretend` o `-p` sulla riga di comando, lo script darà una descrizione dettagliata dei comandi che sta per eseguire, senza però eseguirli. Però, se non hai mai eseguito kdesrc-build, potresti dover usare prima il comando `kdesrc-build --metadata-only` in modo che `--pretend` funzioni.

SUGGERIMENTO

Per una descrizione ancora più prolissa di cosa sta facendo kdesrc-build, prova ad usare l'opzione `--debug`.

- kdesrc-build ti permette di ritirare i moduli rapidamente. Se il modulo che stai ritirando è già stato ritirato altre volte, kdesrc-build ne scaricherà una copia e la preparerà per l'uso sul tuo computer.

SUGGERIMENTO

In generale non c'è bisogno di alcun preparativo speciale per effettuare il ritiro iniziale di un modulo di Git, visto che viene comunque scaricato tutto il deposito di Git, quindi è facile per il server decidere cosa inviare.

Ciò è più veloce per te, e aiuta a ridurre il carico sui server Git anonimi di kde.org.

- Un altro modo di fare più velocemente è avviare il processo di generazione di un modulo non appena il codice sorgente di quel modulo è stato scaricato (disponibile dalla versione 1.6).
- Supporto eccellente per la generazione delle librerie Qt™ (nel caso che i programmi di KDE che stai generando dipendano su una versione recente delle Qt™, non disponibile nella tua distribuzione).
- kdesrc-build non richiede un'interfaccia grafica per funzionare. Quindi, puoi generare software KDE senza bisogno di un ambiente grafico.
- Supporta l'impostazione di opzioni predefinite per tutti i moduli (come le impostazioni di compilazione o le opzioni di configurazione). Queste opzioni possono anche essere cambiate per dei moduli specifici.

Inoltre, kdesrc-build [aggiungerà dei flag standard](#) appropriati per risparmiarti la fatica e i possibili errori derivanti dallo scriverli da te. NB: questo non si applica quando una toolchain (personalizzata) è configurata, per es., mediante: [cmake-toolchain](#)

- kdesrc-build può ritirare un [tag o ramo](#) particolare di un modulo. Puoi anche assicurarti che sia ritirata una particolare [revisione](#) di un modulo.
- kdesrc-build può passare automaticamente a una cartella sorgente di un altro deposito, ramo o tag. Ciò viene fatto automaticamente quando cambi un'opzione che cambia l'URL del deposito, ma devi usare l'opzione `--src-only` per far sapere a kdesrc-build che è accettabile fare il passaggio.
- kdesrc-build può [ritirare anche solo alcune porzioni di un modulo](#), per i casi in cui ti serve solo un programma da un grande modulo.
- Per gli sviluppatori: kdesrc-build vi [ricorderà](#) se usate `git+ssh://` ma `ssh-agent` non è in esecuzione, visto che ciò porterebbe a richieste di password multiple da SSH.
- Può [eliminare la cartella di generazione](#) di un modulo dopo la sua installazione per risparmiare spazio a spese del tempo di una successiva compilazione.
- Gli indirizzi delle cartelle usate da kdesrc-build sono configurabili (anche per modulo).
- Può usare Sudo, o un altro comando specificato dall'utente, per [installare i moduli](#) in modo che kdesrc-build non richieda di essere eseguito dall'amministratore.
- kdesrc-build, come impostazione predefinita, gira a [priorità ridotta](#) per permetterti di usare il computer mentre kdesrc-build sta lavorando.
- Ha supporto per usare i [tag e i rami](#) di KDE.
- C'è supporto per [riprendere una generazione](#) da un modulo dato. Puoi anche [ignorare temporaneamente alcuni moduli](#) di una certa generazione.
- kdesrc-build mostrerà [l'avanzamento della generazione](#) quando si usa CMake, e terrà sempre conto del tempo di generazione, in modo da sapere quanto ci ha messo.
- Ha già un insieme di opzioni predefinite appropriate per generare un'installazione monoutente di KDE dai depositi anonimi del codice sorgente.
- Espansione della tilde per le opzioni di configurazione. Per esempio, puoi specificare:

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

- Configura automaticamente un sistema di generazione, con la cartella sorgente non corrispondente alla cartella di generazione, in modo da tenere la prima pulita.
- Puoi specificare che le opzioni globali si applichino a ogni modulo da ritirare, e puoi anche specificare opzioni da applicare a singoli moduli.
- Rigenerazioni complete forzate, eseguendo kdesrc-build con l'opzione `--refresh-build`.
- Puoi specificare varie variabili d'ambiente da usare durante la generazione, incluse `KDEDIR`, `QTDIR`, `DO_NOT_COMPILE` e `CXXFLAGS`.
- I registri dei comandi hanno data e ora, in modo da avere sempre un registro dell'esecuzione dello script. Inoltre, uno speciale collegamento simbolico chiamato `latest` viene creato e punta sempre al file più recente della cartella dei registri.

3.2 Registro della generazione di kdesrc-build

3.2.1 Panoramica del registro

La registrazione è una funzionalità di kdesrc-build grazie alla quale l'output di ogni comando eseguito da kdesrc-build viene salvato in un file per essere analizzato in seguito, se necessario. Ciò si fa perché è spesso necessario avere l'output di questi programmi in caso di un errore di generazione, perché ci sono molti motivi per cui una generazione potrebbe non riuscire.

3.2.1.1 Schema della cartella di registro

I registri sono sempre tenuti nella cartella di registro. L'indirizzo della cartella di registro è controllato dall'opzione `log-dir`, che ha come valore predefinito `${cartella-sorgente} /log` (dove `${cartella-sorgente}` è il valore dell'opzione `source-dir`); in questa sezione, verrà indicato con `${cartella-registro}`).

Sotto `${cartella-registro}` c'è una serie di cartelle, una per ogni volta che è stato eseguito `kdesrc-build`. Ogni cartella ha per nome la data e il numero di esecuzione. Per esempio, la seconda volta che `kdesrc-build` è stato eseguito il 26 maggio 2004 verrebbe creata la cartella `2004-05-26-02`, dove `2004-05-26` è per la data e `-02` per il numero dell'esecuzione.

Per comodità, `kdesrc-build` creerà anche un collegamento ai registri dell'ultima esecuzione, chiamato `latest`. Quindi i registri dell'ultima esecuzione di `kdesrc-build` saranno sempre sotto `${cartella-registro} /latest`.

Ogni cartella di un'esecuzione di `kdesrc-build` conterrà a sua volta una serie di cartelle, una per ogni modulo di KDE che `kdesrc-build` ha cercato di generare. Inoltre, la cartella conterrà un file `build-status`, che ti permette di determinare quali moduli sono stati generati correttamente e quali no.

NOTA

Se il modulo contiene un sottomodulo (come `extragear/multimedia`, `playground/utils`, o `KDE/kdelibs`), ci sarà una struttura corrispondente nelle cartelle di registro. Per esempio, i registri di `KDE/kdelibs` dopo l'ultima esecuzione di `kdesrc-build` saranno in `${cartella-registro}latest/KDE/kdelibs`, e non in `${cartella-registro} /latest/kdelibs`.

In ogni cartella di registro dei moduli troverai una serie di file per ogni operazione effettuata da `kdesrc-build`. Se `kdesrc-build` aggiorna un modulo, potresti vedere dei file come `git-checkout-update.log` (per il ritiro di un modulo o per l'aggiornamento di un modulo che è stato ritirato in precedenza). Se è stato eseguito il comando **configure**, dovresti aspettarti di vedere un `configure.log` nella cartella.

Se si è verificato un errore, dovresti poterne vedere una spiegazione in uno dei file. Per aiutarti a capire quale file contiene l'errore, `kdesrc-build` creerà un collegamento dal file contenente l'errore (come `build-1.log`) a un file chiamato `error.log`.

Insomma, per vedere perché un modulo non è stato generato dopo l'ultima esecuzione di `kdesrc-build` la prima cosa che devi guardare è il file `${cartella-registro} /latest/ nome-modulo /error.log`.

SUGGERIMENTO

Se il file `error.log` è vuoto (soprattutto dopo un'installazione), forse non c'è davvero stato nessun errore. Alcuni degli strumenti usati dal sistema di generazione di KDE segnaleranno errori per sbaglio anche se in realtà non ce ne sono.

Inoltre, alcuni comandi evaderanno il reindirizzamento dell'output di `kdesrc-build` ed eviteranno il file di registro in certe circostanze (di solito quando si effettua il primo ritiro da Git), e l'output di errore in quel caso non è nel file di registro ma nella Konsole o nel terminale dove hai eseguito `kdesrc-build`.

Capitolo 4

Configurare kdesrc-build

4.1 Panoramica della configurazione di kdesrc-build

Per usare lo script, devi avere un file chiamato `.kdesrc-buildrc` nella tua cartella Home, il quale descrive i moduli che vuoi scaricare e generare, e tutte le opzioni e parametri di configurazione da usare per questi moduli.

4.1.1 Disposizione del file di configurazione

4.1.1.1 Configurazione globale

Il file di configurazione parte con le opzioni globali, specificate come segue:

```
global
nome-opzione valore-opzione
[...]
end global
```

4.1.1.2 Configurazione dei moduli

Seguono quindi una o più sezioni per i moduli, specificate in una delle due seguenti forme:

- ```
module nome-modulo
nome-opzione valore-opzione
[...]
end module
```

- ```
module-set nome-insieme-moduli
  repository kde-projects or git://host.org/percorso/a/deposito.git
  use-modules nomi-moduli

# Si possono impostare anche altre opzioni
nome-opzione valore-opzione
[...]
end module-set
```

IMPORTANTE

Nota che la seconda forma, con gli insiemi di moduli, *funziona solo con i moduli basati su Git*.

Per i moduli di Git, *nome-modulo* deve essere un modulo del deposito Git di KDE (per esempio, kdeartwork o kde-wallpapers).

Per i moduli di Git, il nome del modulo può essere fondamentalmente qualsiasi cosa vuoi, finché non duplica altri nomi di moduli nella configurazione. Tieni presente che la disposizione delle cartelle del codice sorgente e di generazione sarà basata sul nome del modulo se non usi l'opzione [dest-dir](#).

Tuttavia, per gli *insiemi di moduli* di Git, i *nomi-moduli* devono corrispondere a dei veri moduli Git nel deposito (repository) scelto. Vedi [git-repository-base](#) o [use-modules](#) per maggiori informazioni.

4.1.1.3 Elaborazione dei valori delle opzioni

In generale, l'intero contenuto della riga dopo *nome-opzione* viene utilizzato come *valore-opzione*.

Una modifica che kdesrc-build esegue è la sequenza `${nome-della-opzione}` sostituita dal valore di quella opzione preso dalla configurazione globale. Questo ti permette di fare riferimento al valore di opzioni esistenti, incluse le opzioni già impostate da kdesrc-build.

Per osservarne un esempio d'uso, vedi Esempio 2.1.

4.1.1.4 Gruppi 'options'

C'è un ultimo tipo di voce del file di configurazione, i gruppi `options`, che possono essere presenti ovunque ci sia un `module` o un `module-set`.

```
options nome-modulo
nome-opzione valore-opzione
[...]
end options
```

Un gruppo `options` può includere opzioni impostate per sé stesso come qualsiasi dichiarazione di modulo, e viene associato al modulo corrente. Tutte le opzioni impostate in questo modo verranno usate per *sostituirsi* alle opzioni impostate nel modulo associato.

IMPORTANTE

Il nome del modulo associato *deve* corrispondere al nome dato nella dichiarazione `options`. Fai attenzione agli errori di battitura.

Questo è utile a dichiarare un intero `module-set` in cui tutti i moduli devono usare le stesse opzioni, e poi usare i gruppi `options` per fare modifiche individuali.

I gruppi `options` si applicano anche agli insiemi di moduli specificati. Questo consente agli utenti esperti di usare un file di configurazione comune (che include le dichiarazioni `module-set`) come base, e poi eseguire modifiche alle opzioni utilizzate da questi insiemi nei file di configurazione che utilizzano il comando `include` per fare riferimento alla configurazione di base.

Example 4.1 Esempio dell'uso di options

In questo esempio scegliamo di generare tutti i moduli del raggruppamento di programmi multimediali di KDE. Però, vogliamo usare una versione diversa dell'applicazione KMix (forse per verificare la correzione di un errore). Funziona così:

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

# kmix fa parte del gruppo kde/kdemultimedia, e anche se non lo abbiamo mai
# nominato fin qui nel file, kdesrc-build capirà la modifica.
options kmix
  branch KDE/4.12
end options
```

Adesso, quando esegui kdesrc-build, tutti i programmi multimediali di KDE verranno generati dal ramo 'master' del deposito sorgente, ma KMix verrà generato dal ramo precedente 'KDE/4.12'. Usando options non hai dovuto elencare individualmente tutti gli *altri* programmi multimediali di KDE per impostarli al ramo giusto.

NOTA

Nota che questa funzionalità è disponibile solo a partire dalla versione 1.16 di kdesrc-build, o usando la versione di sviluppo di kdesrc-build a partire dal 12 gennaio 2014.

4.1.2 Includere altri file di configurazione

Nel file di configurazione puoi fare riferimento ad altri file usando la parola chiave include con un file, il che sarà come inserire il file riferito nel file di configurazione in quel punto.

Per esempio, potresti avere qualcosa di simile:

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

  # Inserisci qui opzioni specifiche.

end global
```

NOTA

Se non specifichi il percorso completo al file da includere, il file verrà cercato a partire dalla cartella contenente il file sorgente. Questo funziona anche ricorsivamente.

4.1.3 Opzioni di configurazione comunemente usate

Segue un elenco di opzioni comunemente usate. Fai clic sull'opzione per saperne di più. Per vederne l'elenco completo, vedi Sezione 4.2.

- [cmake-options](#), per definire quali flag con cui configurare un modulo usando CMake.

- `branch`, per ritirare da un ramo invece che `master`.
- `configure-flags`, per definire quali flag con cui configurare Qt™.
- `kddir`, per impostare la cartella in cui installare KDE.
- `make-options`, per passare opzioni al programma Make (come il numero di processori da usare).
- `qtdir`, per impostare il percorso alle librerie Qt™.
- `source-dir`, per cambiare da dove scaricare il codice sorgente.

4.2 Tabella delle opzioni di configurazione disponibili

Ecco una tabella delle varie opzioni, contenente le informazioni seguenti:

- Il nome dell'opzione;
- Una descrizione di come `kdesrc-build` reagisce se l'opzione è impostata sia nella sezione globale che nella sezione del modulo del [file di configurazione](#) quando si genera un modulo;
- Commenti specifici sullo scopo e sull'uso dell'opzione.

Nome dell'opzione	Descrizione
<code>async</code>	<p>Tipo Valore predefinito Disponibile da</p> <p>Questa opzione attiva la modalità di operazione asincrona, nella quale l'aggiornamento del codice sorgente e il processo di generazione saranno effettuati in parallelo, invece che attendere la fine di tutti gli aggiornamenti prima di avviare il processo di generazione.</p>
<code>colorful-output</code>	<p>Tipo Valore predefinito</p> <p>Imposta questa opzione a false (falso) per disattivare l'output colorato di <code>kdesrc-build</code>. Nota che <code>kdesrc-build</code> non passerà codici di colore a nient'altro che a un terminale (come <code>xterm</code>, <code>Konsole</code>, o la normale console di Linux®).</p>
<code>disable-agent-check</code>	<p>Tipo Valore predefinito</p> <p>Se stai usando SSH per scaricare i sorgenti da Git (come nel caso in cui stessi usando il protocollo <code>git+ssh</code>), questa opzione controlla se <code>kdesrc-build</code> cercherà di assicurarsi che se stai usando <code>ssh-agent</code>, questo stia effettivamente gestendo delle identità SSH. Questo è per cercare di prevenire che SSH chieda la password per ogni modulo.</p>

Logico (booleano)
True
1.6

Logico (booleano)
True

Logico (booleano)
False

<p>git-desired-protocol</p>	<p>Tipo Valore predefinito Informazioni sulla cronologia</p> <p>Questa opzione si applica solo ai moduli di un deposito del progetto KDE. Quello che fa questa opzione è configurare quale protocollo di rete preferire quando si esegue il push del codice sorgente per questi moduli. Normalmente si usa l'efficientissimo protocollo <code>git</code>, ma potrebbe essere bloccato in alcune reti (come intranet aziendali o WiFi pubbliche). Un protocollo alternativo molto meglio supportato è il protocollo <code>https</code> usato per i siti Web su Internet. Se stai usando una di queste reti ristrette puoi impostare questa opzione a <code>http</code> per preferire le comunicazioni <code>https</code>.</p> <div style="border: 1px solid black; padding: 5px;"> <p>SUGGERIMENTO Potresti anche aver bisogno dell'opzione <code>http-proxy</code> se è necessario un proxy HTTP per il traffico di rete.</p> </div> <p>In qualsiasi altra situazione non dovresti impostare questa opzione, perché il protocollo predefinito è il più efficiente.</p>
-----------------------------	--

Stringa

`git`

Questa opzione è d 1.16. Prima de a utilizzata per c anziché l'URL de per l'aggiorname utilizzato il proto

install-environment-driver	<p>Tipo Valore predefinito Disponibile da</p> <p>Installa uno script di shell che può essere originato in script di configurazione del profilo di un utente, in modo da stabilire le variabili d'ambiente necessarie per avviare il desktop Plasma costruito da kdesrc-build. Questo driver altera le seguenti righe:</p> <ul style="list-style-type: none"> • <code>\$XDG_CONFIG_HOME/kde-env-master.sh</code> (normalmente situato in <code>~/.config/kde-env-master.sh</code>). • <code>\$XDG_CONFIG_HOME/kde-env-user.sh</code> (normalmente situato in <code>~/.config/kde-env-user.sh</code>). <p><code>kde-env-user.sh</code> è facoltativo. Serve per le personalizzazioni utente (per esempi di impostazioni personalizzabili consulta la sezione Troubleshooting and Debugging della KDE UserBase), ma queste impostazioni possono essere impostate dall'utente altrove negli script di configurazione del profilo esistenti. Puoi disabilitare questa funzionalità impostando questa opzione a <code>false</code> e assicurandoti che sia disabilitata anche l'opzione install-session-driver.</p>	Logico (booleano) True 17.08
<p>SUGGERIMENTO kdesrc-build non sovrascriverà i file preesistenti a meno che si passi l'opzione da riga di comando <code>--delete-my-settings</code>.</p>		

<p>install-session-driver</p>	<p>Tipo Valore predefinito Disponibile da</p> <p>Se abilitata, kdesrc-build prova ad installare un driver per il gestore degli accessi grafico che permetta di accedere al desktop KDE generato con kdesrc-build. Questo driver altera le seguenti righe:</p> <ul style="list-style-type: none"> • <code>~/.xsession</code> • <code>\$XDG_CONFIG_HOME/kde-env-master.sh</code> (normalmente situato in <code>~/.config/kde-env-master.sh</code>). • <code>\$XDG_CONFIG_HOME/kde-env-user.sh</code> (normalmente situato in <code>~/.config/kde-env-user.sh</code>). <p>Se mantieni il tuo driver d'accesso personale, puoi disattivare questa funzionalità impostando questa opzione a <code>false</code>. Se abilitata, essa abilita anche la funzionalità install-environment-driver.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>SUGGERIMENTO kdesrc-build non sovrascriverà i file preesistenti a meno che si passi l'opzione da riga di comando <code>--delete-my-settings</code>.</p> </div>	<p>Logico (booleano) True 1.16</p>
<p>niceness</p>	<p>Tipo Valore predefinito</p> <p>Imposta questa opzione a un numero tra 20 e 0. Maggiore è il numero, minore la priorità che kdesrc-build si assegnerà, ossia più alto è il numero, più «cortese» è il programma.</p>	<p>Intero 10</p>
<p>num-cores</p>	<p>Tipo Valore predefinito Disponibile da</p> <p>Questa opzione è definita da kdesrc-build (quando utilizzi kdesrc-build --generate-config), impostata al valore del numero delle CPU disponibili (come indicato dall'applicazione esterna nproc). Se kdesrc-build non è in grado di rilevare il numero di CPU, il valore viene impostato a 4.</p> <p>Per un esempio d'uso dell'opzione vedi Esempio 2.1.</p>	<p>Intero Dipende dal sistema 20.07</p>

<p>num-cores-low-mem</p>	<p>Tipo Valore predefinito Disponibile da</p> <p>Questa opzione è definita da kdesrc-build (quando utilizzi kdesrc-build --generate-config), impostata al valore del numero delle CPU ritenuto adeguato per moduli «pesanti» o comunque con elevata complessità, tipo <code>qtwebengine</code>, al fine di evitare l'esaurimento delle memoria durante la generazione.</p> <p>Il calcolo tipico è un core di CPU per ogni 2 gigabyte (GiB) di memoria totale. Andrà specificato almeno un core e non più di <code>num-cores</code> core.</p> <p>Sebbene questa opzione sia pensata per supportare i moduli Qt™, puoi utilizzarla per qualsiasi altro tuo modulo nello stesso modo in cui viene utilizzato <code>num-cores</code>. Se kdesrc-build non è in grado di rilevare la memoria disponibile, il valore verrà impostato a 2.</p>	<p>Intero Dipende dal sistema 20.07</p>
<p>persistent-data-file</p>	<p>Tipo Disponibile da</p> <p>Usa questa opzione per cambiare dove kdesrc-build memorizza i suoi dati permanenti. La posizione predefinita è un file chiamato <code>.kdesrc-build-data</code>, localizzato nella stessa cartella del file di configurazione in uso. Se il file di configurazione globale è in uso, sarà salvato in <code>~/.local/state/kdesrc-build-data</code> (<code>\$XDG_STATE_HOME/kdesrc-build-data</code>, se <code>\$XDG_STATE_HOME</code> è impostato). Se hai più configurazioni disponibili nella stessa cartella, potresti voler impostare manualmente questa opzione in modo che configurazioni differenti non producano dati permanenti in conflitto tra loro.</p>	<p>Stringa 1.15</p>
<p>ssh-identity-file</p>	<p>Tipo Disponibile da</p> <p>Imposta questa opzione per controllare quale file chiave SSH viene passato al comando ssh-add quando kdesrc-build scarica il codice sorgente dai depositi che richiedono un'autenticazione. Vedi anche Sezione 6.4.1.</p>	<p>Stringa 1.14.2</p>

use-idle-io-priority	<p>Tipo Valore predefinito Disponibile da</p> <p>Usa la priorità minore per l'accesso al disco e ad altro I/O, che può significativamente migliorare i tempi di reazione del resto del sistema al prezzo di tempi di esecuzione di kdesrc-build leggermente più lunghi.</p>	<p>Logico (booleano) False 1.12</p>
use-inactive-modules	<p>Tipo Valore predefinito</p> <p>Consente a kdesrc-build anche di clonare e estrarre dai depositi contrassegnati come inattivi.</p>	<p>Logico (booleano) False</p>

Tabella 4.1: Opzioni solo per gli ambiti globali

Nome dell'opzione	Descrizione	
binpath	<p>Tipo</p> <p>Imposta questa opzione per impostare la variabile d'ambiente <code>PATH</code> durante la generazione. Non puoi sostituire questa impostazione nelle opzioni di un modulo. Il valore predefinito è quello di <code>\$PATH</code> così com'è all'avvio dello script. Questa variabile d'ambiente dovrebbe includere i percorsi, separati da due punti, della toolchain di sviluppo. I percorsi <code>\$KDEDIR /bin</code> e <code>\$QTDIR /bin</code> sono aggiunti automaticamente. Puoi usare la tilde (<code>~</code>) per qualsiasi percorso che aggiungi con questa opzione.</p>	<p>Stringa</p>
branch	<p>Tipo Valore predefinito</p> <p>Ritira il ramo specificato anziché quello predefinito.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTA</p> <p>Per la maggior parte dei moduli di KDE, vorrai probabilmente usare piuttosto l'opzione <code>branch-group</code>, e usare questa di volta in volta per le eccezioni.</p> </div>	<p>Stringa master</p>

<p>branch-group</p>	<p>Tipo Disponibile da</p> <p>Imposta questa opzione a un gruppo generale da cui vuoi scegliere i moduli. Per i tipi di moduli su Git supportati, kdesrc-build determinerà automaticamente il ramo effettivo da usare in base a delle regole stese dagli sviluppatori di KDE (queste sono visibili nel deposito del codice sorgente <code>kde-build-metadata</code> nella tua cartella del codice). Quando viene determinato un ramo, questo viene usato come se lo avessi specificato tu con l'opzione <code>branch</code>.</p> <p>Questo è utile se stai solo cercando di tenerti aggiornato sullo sviluppo senza dover prestare attenzione a tutti i cambi dei nomi dei rami.</p> <p>Nota che se scegli un ramo <i>da te</i> con <code>branch</code>, questo si sostituirà alle impostazioni appena presentate. Lo stesso vale per qualsiasi altra opzione di selezione dei rami come <code>tag</code>.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTA Questa opzione si applica solo ai moduli Git <code>kde-projects</code> (il caso normale). Vedi anche Sezione 2.6.4.</p> </div>	<p>Stringa 1.16-pre2</p>
---------------------	---	------------------------------

<p>build-dir</p>	<p>Tipo Valore predefinito</p> <p>Usa questa opzione per cambiare la cartella usata in cui contenere le sorgenti generate. Si può utilizzare in tre modi differenti:</p> <ol style="list-style-type: none"> 1. Relativa alla cartella dei sorgenti di KDE Git (vedi l'opzione source-dir). È il metodo predefinito, e viene selezionato se digiti il nome di una cartella che non inizia con una tilde (~) o una barra (/). 2. Percorso assoluto. Se specifichi un percorso che inizia per /, allora quel percorso viene utilizzato direttamente. Per esempio, <code>/tmp/kde-obj-dir/</code>. 3. Relativa alla tua cartella home. Se specifichi un percorso che inizia per ~, allora è utilizzato relativamente alla tua cartella Home, in modo analogo all'espansione di tilde della shell. Per esempio, <code>~/builddir</code> imposterebbe la cartella di generazione a <code>/home/user-name/builddir</code>. <p>Questa opzione può essere, sorprendentemente, cambiata per modulo.</p>	<p>Stringa ~/kde/build</p>
------------------	--	--------------------------------

<p>build-when-unchanged</p>	<p>Tipo Valore predefinito</p> <p>Controlla se kdesrc-build debba sempre cercare di generare un modulo che non ha avuto alcun aggiornamento al codice sorgente.</p> <p>Impostando <code>build-when-unchanged</code> a true, kdesrc-build cercherà sempre di generare un modulo, anche se il modulo non ha avuto alcun aggiornamento al codice sorgente. Con questo valore sarà portato con più probabilità a una generazione corretta.</p> <p>Impostando <code>build-when-unchanged</code> a false, kdesrc-build cercherà di generare il modulo solo se ha ricevuto degli aggiornamenti, o in altre situazioni in cui è probabile che una rigenerazione sia effettivamente necessaria. Questo può far risparmiare tempo, specialmente se esegui kdesrc-build ogni giorno, o anche più spesso.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>IMPORTANTE</p> <p>Questa funzionalità viene fornita solo come ottimizzazione. Come per molte altre ottimizzazioni, ci sono dei compromessi sulla correttezza dell'installazione. Per esempio, le modifiche ai moduli qt o kdelibs potrebbero causare una rigenerazione di tutti gli altri moduli, anche se il codice sorgente non è cambiato affatto.</p> </div>
<p>cmake-generator</p>	<p>Tipo Valore predefinito</p> <p>Specifica quale generatore deve essere usato con CMake. Al momento sono supportati Ninja e Unix Makefiles e pure i generatori aggiuntivi basati su di essi, come Eclipse CDT4 - Ninja . I valori non validi (non supportati) vengono ignorati e trattati come non impostati.</p> <p>È da notare che se viene specificato anche un generatore valido mediante <code>cmake-options</code>, esso ignorerà il valore per <code>cmake-generator</code>.</p>

Logico (booleano)
True

Stringa
Makefile Unix

cmake-toolchain	<p>Tipo</p> <p>Specifica un file di toolchain da usare con CMake.</p> <p>Quando viene configurato un file di toolchain valido, <i>kdesrc-build non imposterà più le variabili d'ambiente in modo automatico.</i></p> <p>Puoi utilizzare i parametri set-env, binpath e libpath per correggere manualmente l'ambiente, se il tuo file di toolchain non funziona subito con kdesrc-build. Per maggiori informazioni, consulta la panoramica dei flag standard aggiunti da kdesrc-build.</p> <p>È da notare che se una toolchain valida è specificata anche mediante cmake-options, questo valore verrà usato al posto di quello di <code>cmake-toolchain</code>.</p>	Stringa
-----------------	--	---------

<p>cmake-options</p>	<p>Tipo</p> <p>Si aggiunge alle opzioni globali per il sistema di generazione predefinito, si sostituisce ad esse per gli altri sistemi di generazione.</p> <p>Usa questa opzione per specificare quali flag passare a CMake quando crei il sistema di generazione per il modulo. Quando usata come opzione globale, viene applicata a tutti i moduli generati da questo script. Quando usata come opzione per un modulo, viene aggiunta in fondo alle opzioni globali. Ciò ti permette di specificare opzioni comuni di CMake nella sezione globale.</p> <p>Questa opzione non si applica a qt (che non usa CMake). Usa invece configure-flags. Se viene specificato un generatore valido tra le opzioni elencate, esso ignorerà il valore del cmake-generator. I generatori non validi (non supportati) vengono ignorati e non saranno passati a CMake. Se viene specificato un file di toolchain valido tra le opzioni elencate, esso ignorerà il valore del cmake-toolchain. Le toolchain non valide saranno ignorate e non saranno passate a CMake.</p> <p>Siccome queste opzioni sono passate direttamente alla riga di comando di CMake, dovrebbero essere date come sarebbero passate a CMake. Per esempio:</p> <pre>cmake-options -DCMAKE_BUILD_TYPE=RelWithDebInfo</pre> <p>Siccome è un po' fastidioso, kdesrc-build si dà da fare per assicurarsi che, fintanto che le altre opzioni sono correttamente impostate, dovresti poter lasciare questa opzione in bianco. In altre parole, i parametri <i>necessari</i> di CMake sono impostati automaticamente.</p>	<p>Stringa</p>
<p>compile-commands-export</p>	<p>Tipo</p> <p>Valore predefinito</p> <p>Abilita la generazione del comando <code>compile_commands.json</code> tramite CMake all'interno della cartella di generazione.</p>	<p>Logico (booleano)</p> <p>True</p>

Manuale dello script kdesrc-build

<p>compile-commands-linking</p>	<p>Tipo Valore predefinito Abilita la creazione di collegamenti simbolici da <code>compile_commands.json</code> generato tramite CMake all'interno della cartella di generazione alla corrispondente cartella sorgente.</p>	<p>Logico (booleano) False</p>
<p>configure-flags</p>	<p>Tipo Si aggiunge alle opzioni globali per il sistema di generazione predefinito, si sostituisce ad esse per gli altri sistemi di generazione. Usa questa opzione per specificare quali flag passare a <code>./configure</code> quando crei il sistema di generazione per il modulo. Quando usata come opzione globale, viene applicata a tutti i moduli generati da questo script. <i>Questa opzione funziona solo per qt.</i> Per cambiare le impostazioni di configurazione dei moduli di KDE, vedi cmake-options.</p>	<p>Stringa</p>
<p>custom-build-command</p>	<p>Tipo Questa opzione può essere impostata per eseguire un altro comando (al posto di make, per esempio) per effettuare il processo di generazione. <code>kdesrc-build</code> di solito dovrebbe fare la cosa giusta, quindi non ci dovrebbe essere bisogno di questa opzione. È comunque utile per usare metodi di generazione alternativi. Il valore di questa opzione si usa come la riga di comando da eseguire, modificata dall'opzione make-options come normalmente.</p>	<p>Stringa</p>

<p>cxxflags</p>	<p>Tipo</p> <p>Si aggiunge alle opzioni globali per il sistema di generazione predefinito, si sostituisce ad esse per gli altri sistemi di generazione.</p> <p>Usa questa opzione per specificare quali flag usare per generare il modulo. Questa opzione viene specificata qui invece che con configure-flags o cmake-options perché questa opzione imposterà anche la variabile d'ambiente <code>CXXFLAGS</code> durante il processo di generazione.</p> <p>Nota che per KDE 4 e ogni altro modulo che usi CMake è necessario impostare l'opzione <code>CMAKE_BUILD_TYPE</code> a none quando si configura il modulo. Questo si può fare usando l'opzione cmake-options.</p>	<p>Stringa</p>
<p>dest-dir</p>	<p>Tipo</p> <p>Usa questa opzione per cambiare il nome che viene dato a un modulo sul disco. Per esempio, se il modulo era <code>extragear-network</code>, usando questa opzione puoi rinominarlo a <code>extragear-network</code>.</p> <p>Nota che sebbene ciò cambi il nome del modulo sul disco, non è una buona idea includere cartelle o separatori di cartelle nel nome, perché ciò interferirà con le opzioni build-dir o source-dir.</p>	<p>Stringa</p>
<p>do-not-compile</p>	<p>Tipo</p> <p>Usa questa opzione per selezionare un insieme specifico di cartelle da non generare in un modulo (invece che tutte). Le cartelle da non generare dovrebbero essere separate da spazi.</p> <p>Nota che il codice sorgente dei programmi sarà comunque scaricato.</p> <p>Per esempio, per disabilitare la generazione delle cartelle <code>codeeditor</code> e <code>minimaltest</code> del framework <code>syntaxhighlighting</code>, dovresti aggiungere la compilazione do-not-compile codeeditor minimaltest, e aggiungere «do-not-compile juk kscd» alle tue opzioni <code>syntaxhighlighting</code>.</p> <p>Vedi Sezione 6.3.1.1 per un esempio.</p>	<p>Stringa</p>

<p>git-user</p>	<p>Tipo Disponibile da</p> <p>Questa opzione è per gli sviluppatori di KDE. Se impostata, verrà usata per configurare automaticamente le informazioni di identificazione per il programma di controllo del codice sorgente Git per i moduli Git <i>scaricati da quel momento in avanti</i>, inclusi la maggior parte dei moduli di KDE.</p> <p>In particolare, il nome e l'indirizzo di posta dell'utente di ogni deposito di Git vengono configurati con i valori impostati da questa opzione.</p> <p>Il valore deve essere specificato nella forma Nome <posta@esempio.it>.</p> <p>Per esempio, lo sviluppatore 'Pinco Pallino' con indirizzo 'pinco@pallino.it' scriverebbe:</p> <pre>git-user Pinco Pallino <pinco@pallino.it></pre>	<p>Stringa 15.09</p>
<p>http-proxy</p>	<p>Tipo Disponibile da</p> <p>Questa opzione, se impostata, usa l'URL specificato come server proxy da usare per tutte le comunicazioni su reti HTTP (per esempio, quando si scarica la banca dati dei progetti di KDE).</p> <p>Inoltre, kdesrc-build cercherà di assicurarsi che gli strumenti da cui dipende usino quel server proxy, se possibile, impostando la variabile d'ambiente <code>http_proxy</code> al server indicato, <i>se quella variabile non è già impostata</i>.</p>	<p>Stringa 1.16</p>

<p>directory-layout</p>	<p>Tipo Valori validi</p> <p>Questa opzione è utilizzata per configurare lo schema che kdesrc-build deve utilizzare durante la creazione di cartelle sorgente e di generazione.</p> <p>Lo schema flat è il valore predefinito e raggruppa tutti i moduli direttamente al di sotto delle cartelle sorgenti e di generazione di livello superiore. Per esempio, <code>source/extragear/network/telepathy/ktp-text-ui</code> nello schema metadata diventerebbe invece <code>source/ktp-text-ui</code> utilizzando lo schema flat.</p> <p>Lo schema invent crea una gerarchia di cartelle che rispecchia i percorsi relativi dei repository contenuti in invent.kde.org. Per esempio, <code>source/kde/applications/kate</code> nello schema metadata diventerebbe invece <code>source/utilities/kate</code> utilizzando lo schema invent. Questo schema riguarda solo i progetti KDE. È una buona scelta per i programmatori alle prime armi con kdesrc-build.</p> <p>Infine, lo schema metadata è lo stesso schema del precedente comportamento predefinito. Questo schema organizza i progetti KDE in base ai percorsi di progetto specificati nei metadati del progetto per questi moduli. È una buona scelta se vuoi uno schema di cartelle che tenga traccia di determinati processi KDE, ma tieni presente che questo percorso non è sempre stabile. Come risultato, kdesrc-build potrebbe lasciare da parte una vecchia copia del repository di un progetto e clonarne una nuova a causa di modifiche ai metadati del progetto.</p>
-------------------------	---

Stringa
flat, invent, m

<p>generate-vscode-project-config</p>	<p>Tipo Valore predefinito L'impostazione del modulo ha precedenza su quella globale Imposta questa opzione a true per far creare a kdesrc-build i file del progetto VS Code (cartella .vscode) all'interno della cartella sorgente del modulo. La cartella .vscode sarà creata all'interno della cartella sorgente del progetto, se non esiste già. Le configurazioni abilitano l'uso di LSP, la generazione, il debug e l'esecuzione del progetto dall'interno di VS Code. La configurazione raccomanda anche le estensioni da installare, utili per lavorare sulla maggior parte dei progetti KDE. Puoi anche utilizzare il flag a riga di comando <code>--generate-vscode-project-config.</code></p>	<p>Logico (booleano) False</p>
<p>include-dependencies</p>	<p>Tipo Valore predefinito Controlla se kdesrc-build includerà le dipendenze note di questo modulo nella sua generazione, senza richiederne menzione (neanche indiretta).</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>NOTA Questa opzione funziona solo per moduli basati su kde-project, e presume l'accuratezza dei metadati mantenuti dagli sviluppatori di KDE per il gruppo di rami selezionato.</p> </div> <p>Serve per supportare le applicazioni di compilazione che necessitano di versioni di Qt™ o Plasma più recenti rispetto a quelle supportate dai comuni sistemi operativi.</p>	<p>Logico (booleano) True</p>
<p>install-after-build</p>	<p>Tipo Valore predefinito Questa opzione viene usata per installare il pacchetto a seguito di una generazione riuscita.</p>	<p>Stringa True</p>
<p>kdedir</p>	<p>Tipo Valore predefinito Questa opzione imposta la cartella in cui KDE sarà installato dopo essere stato generato. Se la cambi a una cartella che richiede accesso come amministratore di sistema, potresti essere interessato anche all'opzione <code>make-install-prefix</code>.</p>	<p>Stringa ~/kde/usr</p>

libname	<p>Tipo Valore predefinito</p> <p>Imposta questa opzione per cambiare il nome predefinito della cartella delle librerie installate all'interno di \$KDEDIR e \$QTDIR. In molti sistemi è «lib» o «lib64». Viene avviato il rilevamento automatico per impostare il nome corretto, ma se il rilevamento fallisce può essere cambiato tramite questa impostazione.</p>	Stringa Rilevata automaticamente
libpath	<p>Tipo</p> <p>Imposta questa opzione per impostare la variabile d'ambiente LD_LIBRARY_PATH. Non puoi sostituire questa impostazione per un singolo modulo. Il valore predefinito è vuoto, ma i percorsi \$KDEDIR /\$LIBNAME e \$QTDIR /\$LIBNAME vengono aggiunti automaticamente. Puoi usare la tilde (~) per qualsiasi percorso che aggiungi con questa opzione.</p>	Stringa
log-dir	<p>Tipo</p> <p>Usa questa opzione per cambiare la cartella usata per contenere i file di registro generati dallo script.</p>	Stringa
make-install-prefix	<p>Tipo</p> <p>Imposta questa variabile a un elenco separato da spazi, che verrà interpretato come un comando con le sue opzioni con cui far precedere il comando make install usato per installare i moduli. È comodo per installare i pacchetti con per esempio Sudo, ma sii cauto quando usi i privilegi di amministratore.</p>	Stringa
make-options	<p>Tipo</p> <p>Imposta questa variabile per passare le opzioni della riga di comando al comando make. È utile per programmi come distcc o per sistemi con più di un nucleo nel processore.</p> <p>Nota che non tutti i sistemi di generazione supportati usano make. Per i sistemi di generazione che usano ninja (come il sistema Meson), consulta le impostazioni ninja-options.</p>	Stringa

Manuale dello script kdesrc-build

<p>manual-build</p>	<p>Tipo Valore predefinito Imposta il valore dell'opzione a true (vero) per impedire al processo di cercare di generare questo modulo. Sarà comunque aggiornato quando si aggiorna da Git. Questa opzione è esattamente equivalente all'opzione da riga di comando <code>--no-build</code>.</p>	<p>Logico (booleano) False</p>
<p>manual-update</p>	<p>Tipo Valore predefinito Imposta il valore dell'opzione a true (vero) per impedire che il processo di generazione cerchi di aggiornare (e di conseguenza generare o installare) questo modulo. Se imposti questa opzione per un modulo, lo stai in pratica escludendo.</p>	<p>Logico (booleano) False</p>
<p>ninja-options</p>	<p>Tipo Imposta questa variabile per passare le opzioni a riga di comando al comando di generazione ninja. Questo può essere utile per abilitare l'output 'verboso' o ridurre manualmente il numero di operazioni parallele di generazione che ninja potrebbe utilizzare.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTA Nota che questa impostazione controlla solo ninja quando utilizzato da kdesrc-build. Il modulo 'webengine' di Qt™ usa ninja indirettamente, ma lo supporta ufficialmente solo se generato da make. In questa situazione, puoi impostare <code>NINJAFLAGS</code> affinché make passi i flag appropriati quando chiama ninja in un momento successivo, utilizzando <code>make-options</code>.</p> <pre>options qtwebengine # Imponi a make e ninja di utilizzare non più di 6 operazioni # quando sono disponibili più CPU, in modo da evitare l make-options -j6 NINJAFLAGS=-j6 end options</pre> </div>	<p>Stringa</p>
<p>no-src</p>	<p>Tipo Valore predefinito Se questa opzione è impostata a <i>true</i>, kdesrc-build non aggiornerà automaticamente il codice sorgente del modulo. Cercherà ancora di generare il modulo se ci avrebbe provato comunque in condizioni normali.</p>	<p>Logico (booleano) False</p>

override-build-system

Tipo
 Valore predefinito
 Valori validi
 Disponibile da

Stringa
 Rilevata automaticamente
 KDE, Qt, qmake,
 1.16

Normalmente kdesrc-build rileverà il sistema di generazione giusto da usare per un modulo dopo averlo scaricato. Questo avviene controllando l'esistenza di certi file nella cartella del codice sorgente del modulo.

Alcuni moduli potrebbero avere più di una serie di file richiesti, che potrebbe confondere il rilevamento automatico. In questo caso puoi specificare manualmente il tipo di generazione giusto.

I tipi di generazione attualmente supportati che possono essere impostati sono:

KDE

Usato per generare i moduli di KDE. In realtà può essere usato per generare quasi ogni modulo che usi CMake, ma è meglio non farvi affidamento.

Qt

Usato per generare la libreria Qt™.

qmake

Usato per generare moduli Qt™ che usano file `.pro` nello stile di qmake.

generic

Usato per generare moduli che usano dei semplici Makefile e che non richiedono alcuna configurazione particolare.

autotools

Questo è lo strumento di configurazione standard usato dalla maggior parte dei programmi liberi e open source che non siano nelle altre categorie.

meson

Questo è uno [strumento relativamente nuovo](#) che sta acquistando popolarità come sostituto di autotools e potrebbe essere richiesto da alcuni moduli non

Manuale dello script kdesrc-build

<p>prefix</p>	<p>Tipo Questa opzione controlla dove installare il modulo (normalmente si usa il valore di <code>kdedir</code>). Usare questa opzione permette di installare un modulo in una cartella diversa da dove sono installate le librerie della piattaforma di KDE, come se stessi usando <code>kdesrc-build</code> solo per generare le applicazioni. Puoi usare <code>\${MODULE}</code> o <code>{MODULE}</code> nel percorso per farle espandere nel nome del modulo.</p>	<p>Stringa</p>
<p>purge-old-logs</p>	<p>Tipo Valore predefinito Questa opzione decide se le vecchie cartelle dei registri debbano essere automaticamente eliminate o no.</p>	<p>Logico (booleano) True</p>
<p>qmake-options</p>	<p>Tipo Disponibile da Tutte le opzioni qui specificate vengono passate al comando qmake, per i moduli che usano il sistema di generazione qmake. Per esempio, puoi usare l'opzione PREFIX=/percorso/a/qt per imporre a qmake dove installare il modulo.</p>	<p>Stringa 1.16</p>
<p>qtdir</p>	<p>Tipo Imposta questa opzione per impostare la variabile d'ambiente <code>QTDIR</code> durante la generazione. Se non la specifichi, <code>kdesrc-build</code> presumerà che Qt™ sia fornito dal sistema operativo.</p>	<p>Stringa</p>

<p>remove-after-install</p>	<p>Tipo Valori validi Valore predefinito</p> <p>Se hai poco spazio su disco, potrebbe interessarti questa opzione per eliminare automaticamente la cartella di generazione (o entrambe le cartelle del codice sorgente e di generazione per le installazioni singole) dopo che sarà riuscita l'installazione del modulo.</p> <p>I valori possibili di questa opzione sono:</p> <ul style="list-style-type: none"> • none - Non eliminare nulla. • builddir - Elimina la cartella di generazione, ma non il codice sorgente. • all - Elimina sia la cartella del codice sorgente che quella di generazione. <p>Nota che l'uso di questa opzione può avere un significativo impatto negativo sul tuo uso di banda (se usi <i>all</i>) e sul tempo necessario a compilare software KDE, perché kdesrc-build non potrà effettuare generazioni incrementali.</p>	<p>Stringa none, builddir, all none</p>
<p>repository</p>	<p>Tipo Disponibile da</p> <p>Questa opzione si usa per specificare il deposito Git da cui scaricare il codice sorgente del modulo. Qt™ (e quindi qt) avrebbero bisogno di questa opzione, oltre a diversi moduli di KDE che sono nel processo di conversione a Git.</p>	<p>Stringa 1.10</p>
<p>revision</p>	<p>Tipo Disponibile da</p> <p>Se questa opzione è impostata a un valore diverso da zero, kdesrc-build costringerà l'aggiornamento del codice sorgente a portare un modulo all'esatta revisione data, anche se ci sono opzioni come branch. Se il modulo è già alla revisione data non sarà aggiornato di nuovo, a meno che questa opzione sia cambiata o rimossa dalla configurazione.</p>	<p>Stringa 1.16</p>

run-tests	<p>Tipo Valore predefinito</p> <p>Se impostata a true, il modulo sarà generato con supporto per eseguire i suoi test, che verranno eseguiti come parte del processo di generazione. kdesrc-build visualizzerà un breve rapporto dei risultati dei test. Ciò è utile per gli sviluppatori o coloro che vogliono assicurarsi che il loro sistema sia correttamente configurato.</p>	Logico (booleano) False
set-env	<p>Tipo</p> <p>Questa opzione accetta un insieme di valori separati da spazi, dove il primo è la variabile d'ambiente da impostare, e il resto i valori a cui vuoi impostarla. Per esempio, per impostare la variabile PINCO a <i>pallino</i>, inseriresti questo comando nella sezione opportuna:</p> <pre>set-env PINCO pallino</pre> <p>Questa opzione è speciale perché può essere ripetuta senza sostituire le impostazioni di set-env precedenti nella stessa sezione del file di configurazione. In questo modo puoi impostare diverse variabili d'ambiente per modulo (o globalmente).</p>	Stringa
source-dir	<p>Tipo Valore predefinito</p> <p>Questa opzione si usa per impostare la cartelle in cui tenere il codice sorgente Git di KDE sul tuo computer. Puoi usare la tilde (~) per rappresentare la cartella Home usando questa opzione.</p>	Stringa ~/kde/src
stop-on-failure	<p>Tipo Valore predefinito</p> <p>L'impostazione di questa opzione a false consente di continuare l'esecuzione dello script a seguito del verificarsi di un errore durante il processo di generazione o installazione.</p>	Logico (booleano) True
tag	<p>Tipo Disponibile da</p> <p>Usa questa opzione per scaricare un rilascio specifico di un modulo.</p> <p><i>Nota:</i> è molto probabile che usare questa opzione <i>non è quello che vuoi fare</i>. I rilasci di KDE sono disponibili come tarball dal sito degli scaricamenti di KDE.</p>	Stringa 1.16

Manuale dello script kdesrc-build

use-clean-install	<p>Tipo Valore predefinito Disponibile da Imposta questa opzione a true per far eseguire make uninstall a kdesrc-build appena prima di eseguire make uninstall. Questo può essere utile per assicurarsi che non ci siano delle vecchie librerie abbandonate, metadati di CMake, eccetera, che possono causare problemi in installazioni di KDE di lunga data. Questo però funziona solo sui sistemi di generazione che supportano make uninstall.</p>
-------------------	--

Logico (booleano)
 False
 1.12

Tabella 4.2: Opzioni di tutti gli ambiti (modulo, insieme di moduli e globali)

Nome dell'opzione	Descrizione
-------------------	-------------

<p>git-repository-base</p>	<p>Tipo Disponibile da</p> <p>Questa opzione si usa per creare un nome breve per riferirsi a un particolare URL di un deposito di Git nelle dichiarazioni di insiemi di moduli seguenti, il che è utile per dichiarare molti moduli di iGit da generare. Devi specificare due cose (separate da uno spazio): il nome da assegnare all'URL di base, e il suo nome esatto. Per esempio:</p> <pre>global # altre opzioni # Questo è il percorso comune a tutti i moduli su server git-repository-base kde-git kde: end global # Dichiarazioni di moduli module-set # Adesso puoi usare l'alias prima definito, ma solo in u repository kde-git use-modules modulo1.git modulo2.git end module-set</pre> <p>L'opzione <code>use-modules</code> dell'insieme di moduli ha creato due moduli interni, e <code>kdesrc-build</code> si è comportato come se avesse letto:</p> <pre>module modulo1 repository kde:modulo1.git end module module modulo2 repository kde:modulo2.git end module</pre> <p>Il prefisso del deposito Git <code>kde:</code> usato sopra è una scorciatoia che verrà configurata automaticamente da <code>kdesrc-build</code>. Vedi l'articolo di Techbase sulla rinomina degli URL per maggiori informazioni. Nota che al contrario della maggior parte delle altre opzioni, questa può essere specificata più volte per creare tanti alias quanti sono richiesti.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>SUGGERIMENTO</p> <p>Non è necessario usare questa opzione per usare gli insiemi di moduli: questa opzione esiste per facilitare l'uso dello stesso deposito in diversi insiemi di moduli.</p> </div>	<p>Stringa 1.12.1</p>
----------------------------	--	---------------------------

<p>ignore-modules</p>	<p>Tipo Disponibile da</p> <p>Attualmente è utilizzabile solo nell'insieme di moduli, vedi il Bug 472917.</p> <p>I moduli indicati da questa opzione, che sarebbero altrimenti stati scelti da kdesrc-build tramite un'opzione use-modules, sono invece completamente saltati. Usa questa opzione se vuoi generare un intero gruppo del progetto kde-projects <i>tranne</i> alcuni moduli specifici.</p> <p>Il valore dell'opzione non deve necessariamente fare direttamente il nome del modulo. Qualsiasi modulo che contenga delle parti contigue e complete del suo percorso nei progetti di KDE corrispondenti a uno dei valori dell'opzione verrà ignorato; in questo modo puoi ignorare più moduli alla volta.</p> <p>Per esempio, un valore dell'opzione come <i>libs</i> farebbe escludere sia <i>kde/kdegraphics/libs</i> che <i>playground/libs</i> (ma non <i>kde/kdelibs</i> perché è la parte intera 'kdelibs' che viene considerata).</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>SUGGERIMENTO Vedi anche Esempio 2.7.</p> </div>	<p>Stringa 1.16</p>
-----------------------	---	-------------------------

<p>use-modules</p>	<p>Tipo Disponibile da Applicabile solamente nell'insieme di moduli Questa opzione permette di specificare facilmente molti moduli da generare in un solo punto del file di configurazione. Ogni identificativo passato a questa opzione viene convertito internamente in un modulo di kdesrc-build, con un'opzione repository impostata al deposito dell'insieme di moduli combinato con l'identificativo per configurare il deposito finale da cui scaricare. Tutte le altre opzioni assegnate all'insieme di moduli sono passate ai moduli generati senza modifica. L'ordine in cui i moduli vengono definiti in questa opzione è importante, perché è anche l'ordine in cui kdesrc-build elaborerà i moduli generati durante aggiornamento, generazione e installazione. Tutti i moduli definiti nell'insieme di moduli dato saranno gestiti prima che kdesrc-build passerà al modulo seguente dopo l'insieme di moduli. Se devi cambiare le opzioni di un modulo generato, basta ridichiararlo dopo averlo definito nell'insieme, e impostare le opzioni come necessario. Anche se cambierai le opzioni del modulo in questo modo, il modulo sarà ancora aggiornato e generato nell'ordine impostato dall'insieme di moduli (non puoi quindi riordinare la sequenza di generazione in questo modo).</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>IMPORTANTE Il nome da usare per il modulo, in questo caso, è semplicemente il nome passato a <code>use-modules</code>, con l'eccezione che ogni estensione <code>.git</code> viene rimossa.</p> </div> <p>Vedi Sezione 2.6.3 e git-repository-base per avere una descrizione di come si usa e un esempio.</p>	<p>Stringa 1.12.1</p>
--------------------	---	---------------------------

Tabella 4.3: Opzioni per gli ambiti globali e l'insieme di moduli

Capitolo 5

Opzioni da riga di comando e variabili d'ambiente

5.1 Uso dalla riga di comando

kdesrc-build è progettato per essere usato come segue:

```
kdesrc-build [--options...] [moduli da generare...]
```

Se nessun modulo viene specificato sulla riga di comando, kdesrc-build genererà tutti i moduli definiti nel suo file di configurazione, nell'ordine elencato in quel file (anche se ciò è soggetto a modifiche da parte di diverse opzioni del file di configurazione).

5.1.1 Opzioni dalla riga di comando di uso comune

L'elenco completo delle opzioni dalla riga di comando è dato nella Sezione 5.3. Tra le opzioni più comunemente usate ci sono:

--pretend (o -p)

Questa opzione fa indicare a kdesrc-build le azioni che effettuerebbe, senza implementarle davvero. Può essere utile per assicurarsi che i moduli che pensi di generare siano effettivamente quelli che saranno generati.

--refresh-build

Questa opzione costringe kdesrc-build a generare i moduli dati da un punto di partenza completamente nuovo. Qualsiasi cartella di generazione esistente per un modulo viene rimossa e il modulo viene rigenerato. Questa opzione è utile se incontri errori nella generazione di un modulo, e a volte è necessaria quando cambiano le librerie di Qt™ o KDE.

--no-src

Questa opzione salta il processo di aggiornamento del codice sorgente. Potresti usarla se hai aggiornato il codice sorgente da poco (con kdesrc-build o magari manualmente), ma vuoi lo stesso rigenerare qualche modulo.

--no-build

Questa opzione è simile a `--no-src` sopra, ma stavolta è il processo di generazione che viene saltato.

5.1.2 Specificare i moduli da generare

In generale, specificare i moduli da generare è facile come passare il loro nome come definito nel file di configurazione. Puoi anche passare dei moduli che fanno parte di un insieme, come indicato in [use-modules](#) o il nome di tutto l'insieme, se gliene hai dato uno.

Nel caso specifico di insiemi di moduli basati sulla [banca dati dei progetti di KDE](#), kdesrc-build espanderà i componenti dei nomi dei moduli per determinare il modulo che vuoi. Per esempio, la voce dei progetti di KDE di kdesrc-build indica il progetto come `extragear/utils/kdesrc-build`. Potresti specificare uno qualsiasi dei seguenti per generare kdesrc-build:

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

NOTA

I comandi nell'esempio precedente mettevano un + davanti al nome del modulo. Questo fa interpretare il nome come un modulo della banca dati dei progetti di KDE, anche se quel modulo non è definito nel tuo file di configurazione.

Fai attenzione se specifichi progetti molto grandi (per esempio `extragear/utils`), perché può portare alla generazione di molti moduli. Dovresti usare l'opzione `--pretend` prima di generare un nuovo modulo per assicurarti di generare solo i moduli che vuoi.

5.2 Variabili d'ambiente supportate

kdesrc-build non usa variabili d'ambiente. se devi impostare delle variabili d'ambiente per il processo di generazione o installazione, vedi l'opzione [set-env](#).

5.3 Parametri da riga di comando supportati

Lo script accetta le seguenti opzioni dalla riga di comando:

--async

Abilita la [modalità asincrona](#), che può effettuare gli aggiornamenti al codice sorgente e generazione dei moduli allo stesso tempo. Questa è predefinita, devi specificare l'opzione solo se l'hai disattivata nella configurazione.

--help (o -h)

Mostra soltanto un semplice aiuto sull'uso di questo script.

--version (o -v)

Mostra la versione del programma.

--show-info

Visualizza informazioni su kdesrc-build e il sistema operativo, che si mostrano utili nelle segnalazioni di errori o quando richiedi aiuto nei forum o nelle mailing list.

Disponibile dalla versione 18.11.

Manuale dello script kdesrc-build

--initial-setup

Consente a kdesrc-build di eseguire una prima configurazione necessaria per preparare il sistema affinché kdesrc-build possa operare e il software KDE appena installato possa avviarsi.

Questo include:

- l'installazione delle dipendenze conosciute (sulle distribuzioni Linux[®] supportate)
- l'aggiunta delle variabili d'ambiente richieste in `~/.bashrc`
- l'impostazione di un [file di configurazione](#)

Questa opzione è esattamente uguale all'uso concomitante di `--install-distro-packages --generate-config --update-shellrc`.

--install-distro-packages

Installa i pacchetti della distribuzione (nelle distribuzioni Linux[®] supportate) necessari a preparare il sistema al funzionamento di kdesrc-build e all'esecuzione del software KDE installato di recente.

Vedi anche `--initial-setup`

--generate-config

Genera il file di configurazione di kdesrc-build.

Vedi anche `--initial-setup`

--update-shellrc

Modifica `«~/.bashrc»` (o un altro file rc di shell) per aggiungere kdesrc-build alla tua variabile `$PATH` e abilitare il completamento automatico.

Vedi anche `--initial-setup`

--author

Mostra informazioni per contattare l'autore.

--color

Abilita l'output colorato (è predefinito per i terminali interattivi).

--generate-vscode-project-config

Genera una cartella `.vscode` con configurazioni per la generazione e il debug in Visual Studio Code.

--nice=valore

Questo valore regola la priorità di processore richiesta da kdesrc-build, e dovrebbe essere tra 0 e 20. 0 è la priorità massima (perché è la meno 'cortese', in inglese *nice*), 20 la minima. kdesrc-build usa un valore predefinito di 10.

--no-async

Disabilita la [modalità asincrona](#) di aggiornamento. L'aggiornamento verrà invece interamente effettuato prima dell'inizio della generazione. Questa opzione rallenterà il processo complessivo, ma, se si verificano errori IPC durante l'esecuzione di kdesrc-build, prova ad usare questa opzione e invia una [segnalazione di errore](#).

--no-color

Disabilita l'output colorato.

--pretend (o -p)

kdesrc-build farà il processo di aggiornamento e generazione, ma invece di aggiornare e generare davvero scriverà cosa avrebbe fatto (per esempio comandi da eseguire, passi vari, eccetera).

NOTA

Dei semplici comandi di sola lettura (come leggere informazioni dei file) potrebbero comunque essere eseguiti per rendere l'output più significativo (come simulare correttamente se il codice sorgente debba essere ritirato o aggiornato).

IMPORTANTE

Questa opzione richiede la disponibilità di alcuni metadati, i quali di norma vengono scaricati; ma gli scaricamenti sono disattivati in questa modalità di prova. Se non hai mai eseguito kdesrc-build, e quindi non hai questi metadati, dovresti eseguire prima **kdesrc-build --metadata-only** per scaricare i metadati necessari.

--quiet (o -q)

Non scrivere tanto output. Con questa opzione vengono scritte solo le cose fondamentali.

--really-quiet

scrivi solo avvertimenti ed errori.

--verbose

Scrivi molte cose su quello che sta succedendo, e su cosa sta facendo kdesrc-build.

--src-only (o -s)

Fai solo l'aggiornamento del codice sorgente.

--build-only

Fai solo la generazione.

--install-only

Se questa è l'unica opzione dalla riga di comando, cerca di installare tutti i moduli contenuti nella cartella `log/latest/build-status`. Se sono specificate opzioni da riga di comando dopo `--install-only`, sono tutte considerate essere moduli da installare (anche se non sono stati generati correttamente l'ultima volta).

--metadata-only

Effettua solo il processo di scaricamento dei metadati. kdesrc-build normalmente lo gestisce automaticamente, ma potresti usarlo per far funzionare l'opzione `--pretend`.

--rebuild-failures

Usa questa opzione per generare solo quei moduli la cui generazione non è riuscita in una precedente esecuzione di kdesrc-build. Questo è utile se si è verificato un numero notevole di generazioni errate miste a riuscite. Dopo aver corretto ciò che causa gli errori, puoi quindi facilmente rigenerare solo i moduli che prima non sono riusciti.

NOTA

Nota che l'elenco di 'moduli precedentemente non riusciti' viene azzerato ogni volta che un'esecuzione di kdesrc-build si conclude con errori. Tuttavia non viene azzerato in caso di generazione perfettamente riuscita, quindi puoi rigenerare correttamente un modulo o due e questa opzione funzionerà ancora.

Questa opzione è stata aggiunta a kdesrc-build 15.09.

--include-dependencies (o -d), --no-include-dependencies (o -D)

Questa opzione fa sì che kdesrc-build includa automaticamente gli altri moduli KDE e Qt™ nella compilazione, se richiesto per i moduli che hai indicato da compilare nella riga di comando o nel tuo [file di configurazione](#).

I moduli inclusi vengono aggiunti come registrati all'interno del sistema di gestione del codice sorgente di KDE. Vedi Sezione 2.6.4.

L'opzione del file di configurazione corrispondente è [include-dependencies](#).

Puoi anche usare `--no-include-dependencies`, che disattiva l'inclusione automatica dei moduli delle dipendenze aggiuntivi.

`--ignore-modules (0 -!)`

Non includere i moduli passati al resto della riga di comando nel processo di aggiornamento e generazione (ciò è utile se vuoi generare la maggior parte dei moduli nel [file di configurazione](#) e saltarne qualcuno).

`--no-src (0 -S)`

Salta la connessione al server Git.

`--no-build`

Salta il processo di generazione.

`--no-metadata`

Non scaricare automaticamente i metadati necessari per i moduli git di KDE. Gli aggiornamenti del codice sorgente dei moduli stessi avverrà comunque, a meno che passi anche `--no-src`.

Questo può essere utile quando esegui più volte `kdesrc-build`, visto che i metadati non cambiano spesso. Nota però che molte altre funzionalità richiedono la presenza dei metadati. Potresti considerare di eseguire `kdesrc-build` con l'opzione `--metadata-only` una volta, e usare questa opzione per le esecuzioni seguenti.

`--no-install`

Non installare automaticamente i pacchetti dopo averli generati.

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

Questa opzione disabilita esplicitamente il salto del processo di generazione (un'ottimizzazione controllata dall'opzione [build-when-unchanged](#)). Ciò è utile per far eseguire a `kdesrc-build` la generazione quando hai cambiato qualcosa che `kdesrc-build` non può sapere.

`--force-build` fa esattamente la stessa cosa, e forse è più facile da ricordare.

`--debug`

Abilita la modalità di debug per lo script. Attualmente ciò vuol dire che tutto l'output sarà riversato sullo standard output oltre ad essere registrato nella cartella di registro, come sarebbe normale. Inoltre, molte funzioni sono molto più verbose riguardo a quello che stanno facendo in modalità di debug.

`--query=mode`

Questo comando fa richiedere a `kdesrc-build` un parametro dei moduli nell'elenco di compilazione (o passato dalla riga di comando o letto dal file di configurazione), inviando il risultato allo schermo (un modulo per riga).

Questa opzione deve essere fornita con 'modalità query', che deve essere una delle seguenti:

- `source-dir`: `kdesrc-build` restituisce il percorso completo in cui è memorizzato il codice sorgente del modulo.
- `build-dir`: `kdesrc-build` restituisce il percorso completo in cui si è compiuto il processo di generazione del modulo.
- `install-dir`, `kdesrc-build` restituisce il percorso completo in cui il modulo sarà installato.
- `project-path`: `kdesrc-build` restituisce la posizione del modulo all'interno della gerarchia dei depositi di codice sorgente di KDE. Consulta Sezione 2.6.4 per maggiori informazioni su questa gerarchia.

- `branch`: `kdesrc-build` restituisce il ramo git risolto che verrà usato per ciascun modulo, basato sulle impostazioni effettive di `tag`, `branch` e `branch-group`.
- In alternativa, possono essere usati i nomi delle opzioni, validi per i moduli, contenuti nel [file di configurazione](#), i cui valori risolti saranno elencati per ciascun modulo.

Se un singolo modulo viene passato alla riga di comando, allora il risultato sarà semplicemente il valore del parametro che è stato ricercato. Se vengono passati più moduli (multipli o no) sulla riga di comando, allora ciascuna riga sarà prefissata dal nome del modulo. In ogni caso, `kdesrc-build` si ferma una volta che vengono restituiti tutti i valori.

Questa opzione è stata aggiunta in `kdesrc-build` 16.05.

Per esempio, il comando `'kdesrc-build --query branch kactivities kdepim'` potrebbe concludersi con un risultato tipo:

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

`--refresh-build (o -r)`

Ricrea il sistema di generazione e compila da zero.

`--reconfigure`

Riesegui `cmake` (per i moduli di KDE) o `configure` (per Qt™), senza ripulire la cartella di generazione. Normalmente non avresti bisogno di specificarlo, perché `kdesrc-build` rileverà il cambiamento di opzioni rilevanti e rieseguirà automaticamente la configurazione. Questa opzione è sottintesa se si usa `--refresh-build`.

`--resume-from (o --from oppure -f)`

Questa opzione è usata per riprendere la generazione a partire dal modulo dato, che dovrebbe essere la prossima opzione sulla riga di comando. Non dovresti specificare altri nomi di moduli sulla riga di comando.

NOTA

Questa opzione in precedenza aggiungeva `--no-src`, ma non lo fa più (da `kdesrc-build` 1.13). Se vuoi evitare gli aggiornamenti del sorgente quando riprendi, basta che passi `--no-src` insieme alle altre opzioni.

Vedi anche `--resume-after (o --after oppure -a)` e Sezione 6.3.6.1. Probabilmente preferiresti usare questa opzione da riga di comando se hai corretto l'errore di generazione e vuoi che `kdesrc-build` la completi.

`--resume-after (o --after oppure -a)`

Questa opzione è usata per riprendere la generazione da dopo il modulo dato, che dovrebbe essere la prossima opzione sulla riga di comando. Non dovresti specificare altri nomi di moduli sulla riga di comando.

NOTA

Questa opzione in precedenza aggiungeva `--no-src`, ma non lo fa più (da `kdesrc-build` 1.13). Se vuoi evitare gli aggiornamenti del sorgente quando riprendi, basta che passi `--no-src` insieme alle altre opzioni.

Vedi anche `--resume-from (o --from oppure -f)` e Sezione 6.3.6.1. Probabilmente preferiresti usare questa opzione da riga di comando se hai corretto l'errore di generazione e hai anche generato e installato il modulo da te, e vuoi che `kdesrc-build` riparta dal modulo seguente.

--resume

Questa opzione può essere usata per eseguire kdesrc-build dopo che si è verificato un errore di generazione.

Riprende la generazione dal modulo che non è riuscito, usando l'elenco dei moduli in attesa di essere generati, e disattiva gli aggiornamenti del codice sorgente e dei metadati. L'uso tipico è quando un banale errore o una dipendenza mancante causano l'errore di generazione. Una volta che l'errore è stato corretto, puoi rapidamente tornare alla generazione dei moduli senza dover configurare --resume-from e --stop-before.

Questa opzione è stata aggiunta in kdesrc-build 1.16.

--stop-before (o --until)

Questa opzione da riga di comando si usa per fermare il normale processo di generazione appena *prima* della prevista generazione di un modulo.

Per esempio, se l'elenco di generazione normale era moduloA, moduloB, moduloC, --stop-before=moduloB farà generare a kdesrc-build solo moduloA.

Questa opzione da riga di comando è stata aggiunta in kdesrc-build 1.16.

--stop-after (o --to)

Questa opzione da riga di comando si usa per fermare il normale processo di generazione appena *dopo* la generazione di un modulo.

Per esempio, se l'elenco di generazione normale era moduloA, moduloB, moduloC, --stop-before=moduloB farà generare a kdesrc-build moduloA e moduloB.

Questa opzione da riga di comando è stata aggiunta in kdesrc-build 1.16.

--stop-on-failure, --no-stop-on-failure

Questa opzione controlla se la generazione sarà interrotta non appena si verifica un errore. Il comportamento predefinito è --stop-on-failure. È possibile ignorarla se vuoi procedere con i moduli rimanenti della generazione, per evitare di perdere tempo nel caso in cui il problema fosse in un solo modulo.

Questa opzione è stata rimossa in kdesrc-build 1.16. Vedi anche l'opzione del file di configurazione [stop-on-failure](#).

--rc-file

Questo interpreta il prossimo parametro della riga di comando come il file da cui leggere le opzioni di configurazione. Il valore predefinito di questo parametro è kdesrc-buildrc (spuntato nell'attuale cartella di lavoro). Se questo file non esiste, sarà usato invece ~/.config/kdesrc-buildrc (\$XDG_CONFIG_HOME/kdesrc-buildrc, se \$XDG_CONFIG_HOME è impostato). Vedi anche capitolo 4.

--print-modules

Prende tutte le azioni specificate sulla riga di comando (o nel file di configurazione) fino al riordino delle dipendenze dei moduli incluso, stampa i moduli che verrebbero elaborati uno per riga, e quindi esce senza fare altro.

I metadati di kde-project vengono scaricati per primi (però, vedi --pretend o --no-src).

Il risultato non è pienamente compatibile con l'uso di script, perché potrebbero essere generati altri messaggi prima dell'elenco dei moduli.

Questo è utile soprattutto per capire velocemente quali dipendenze kdesrc-build determina per un modulo, il che vuol dire che è utile solo per i moduli di [kde-projects](#). Questa opzione è compatibile con --resume-from, --resume-after, --stop-before, e --stop-after.

--list-build

Elenca i moduli che devono essere compilati, in ordine di compilazione. Se applicabile, l'elenco dell'output cita pure quale commit/ramo/tag deve essere selezionato per il checkout.

Questa opzione è simile a --print-modules. Per maggiori informazioni su come i moduli si collegano l'un l'altro, consulta anche: --dependency-tree.

--dependency-tree

Restituisce le informazioni di dipendenza sui moduli che devono essere compilati, utilizzando un formato ad albero (ricorsivo). Le informazioni elencate includono anche da quale commit/ramo/tag dipendono e se la dipendenza deve essere compilata o no. Nota: l'output generato può diventare molto grande per le applicazioni con molte dipendenze.

--run

Questa opzione interpreta il prossimo elemento sulla riga di comando come un programma da eseguire; kdesrc-build finirà di leggere il file di configurazione, aggiornerà l'ambiente come normalmente, e poi eseguirà il programma dato.

Nella maggior parte dei casi, tuttavia, ciò non funzionerà per avviare una shell con l'ambiente di kdesrc-build, perché le shell interattive di solito azzerano almeno parte delle variabili d'ambiente (come PATH e KDEDIRS) nella sequenza di avvio.

SUGGERIMENTO

Se vuoi vedere l'ambiente usato da kdesrc-build, puoi eseguire il comando **printenv**:

```
$ 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
... eccetera.
```

--prefix=</percorso/a/kde>

Permette di cambiare la cartella in cui sarà installato KDE dalla riga di comando. Questa opzione implica **--reconfigure**, ma potrebbe ancora essere richiesto di usare **--refresh-build**.

--revision

Questa opzione fa ritirare a kdesrc-build la revisione corrispondente a un certo numero per ogni modulo di Git, sostituendosi a qualsiasi opzione di **ramo**, **tag** o **revisione** già impostata per questi moduli.

Questa opzione probabilmente non è una buona idea, ed è supportata per retrocompatibilità con vecchi script.

--build-system-only

Questa opzione fa interrompere a kdesrc-build la generazione di un modulo appena prima dell'esecuzione del comando **make**. Ciò viene supportato solo per retrocompatibilità con vecchie versioni; questo effetto non è utile per l'attuale sistema di generazione di KDE.

--delete-my-patches

Questa opzione si usa per far eliminare a kdesrc-build le cartelle sorgente che potrebbero contenere dati dell'utente, in modo che i moduli possano essere scaricati di nuovo. Questo è normalmente utile solo agli sviluppatori di KDE (che potrebbero avere delle modifiche locali da eliminare).

Normalmente non dovresti usare questa opzione; kdesrc-build richiederà di venire eseguito una seconda volta, se necessario.

--delete-my-settings

Questa opzione si usa per far sovrascrivere a kdesrc-build tutti i file preesistenti che potrebbero contenere dati dell'utente.

Al momento si usa soltanto per la configurazione di xsession per il gestore degli accessi. Normalmente non dovresti usare questa opzione; kdesrc-build richiederà di venire eseguito una seconda volta, se necessario.

Manuale dello script kdesrc-build

--<nome_opzione>=

Puoi usare questa opzione per sostituire un'opzione nel tuo [file di configurazione](#) per ogni modulo. Per esempio, per sostituire l'opzione `log-dir`, scriveresti: `--log-dir= /percorso/a/cartella`.

NOTA

Questa funzionalità può essere usata solo per i nomi di opzioni già note a kdesrc-build che non siano già supportate da appropriate opzioni da riga di comando. Per esempio, l'opzione del file di configurazione `async` ha le apposite opzioni da riga di comando `--async` e `--no-async`, che vengono preferite da kdesrc-build.

--set-module-option-value=<nome-modulo>, <nome-opzione>, <valore-opzione>

Puoi usare questa opzione per sostituire un'opzione nel [file di configurazione](#) di un modulo particolare.

Tutte le altre opzioni da riga di comando sono considerate come moduli da aggiornare e generare. Non confondere la generazione con l'installazione.

Capitolo 6

Usare kdesrc-build

6.1 Prefazione

Normalmente, usare kdesrc-build dopo essere passati dal capitolo 2 è facile quanto fare quanto segue da un terminale:

```
% kdesrc-build
```

kdesrc-build quindi scaricherà il codice sorgente di KDE, cercherà di configurarlo e generarlo, e infine di installarlo.

Continua a leggere per scoprire come fa kdesrc-build, e cos'altro puoi fare con questo strumento.

6.2 Funzionalità di base di kdesrc-build

6.2.1 Supporto per qt

kdesrc-build supporta la generazione delle librerie Qt™ usate dal software di KDE come comodità per gli utenti. Questo supporto viene gestito da un modulo speciale chiamato qt.

NOTA

Qt™ è sviluppato in un deposito separato dal software di KDE, che si trova a <http://code.qt.io/cgit/qt/>.

Per generare Qt™, dovresti assicurarti che l'impostazione `qtdir` sia impostata alla cartella in cui vorresti installare Qt™, come descritto nella Sezione 2.2.

Dovresti quindi assicurarti che il modulo qt sia aggiunto a `.kdesrc-buildrc` prima di ogni altro modulo nel file. Se stai usando il file di configurazione campione, puoi semplicemente decommentare la voce di qt ivi presente.

Dovresti verificare che le opzioni `repository` e `branch` siano impostate correttamente:

1. La prima possibilità è generare Qt™ con un mirror gestito dai depositi di codice sorgente di KDE (non vengono applicate modifiche, visto che è semplicemente un clone del sorgente ufficiale). Questo è decisamente raccomandato a causa di problemi occasionali nel clonare l'intero modulo di Qt™ dal suo deposito ufficiale.

Puoi impostare l'opzione `repository` per il modulo qt a `kde:qt` per usare questa opzione.

2. Altrimenti, per generare le Qt™ standard, imposta l'opzione `repository` a `git://gitorious.org/qt/qt.git`. Nota che potresti avere problemi ad effettuare il clone iniziale di Qt™ da questo deposito.

In entrambi i casi, l'opzione `branch` dovrebbe essere impostata a `master` (a meno che volessi generare un altro ramo).

6.2.2 Flag standard aggiunti da kdesrc-build

NB: questa sezione non si applica ai moduli per i quali hai configurato una toolchain personalizzata, utilizzando per esempio `cmake-toolchain`.

Per risparmiarti tempo, `kdesrc-build` aggiunge alcuni percorsi standard al tuo ambiente:

- I percorsi alle librerie KDE e Qt™ vengono automaticamente aggiunti alla variabile `LD_LIBRARY_PATH`. Ciò vuol dire che non devi modificare `libpath` per includerle.
- I percorsi ai programmi di sviluppo di KDE e Qt™ vengono automaticamente aggiunti alla variabile `PATH`. Ciò vuol dire che non devi modificare `binpath` per includerli.
- Il percorso al programma `pkg-config` fornito da KDE viene aggiunto automaticamente a `PKG_CONFIG_PATH`. Ciò vuol dire che non devi modificare `set-env` per includerlo.
- L'impostazione di `kdedir` è automaticamente propagata alla variabile d'ambiente `KDEDIR` durante la generazione. `KDEDIRS` non viene modificata.
- L'impostazione di `qtdir` è automaticamente propagata alla variabile d'ambiente `QTDIR` durante la generazione.

6.2.3 Cambiare le priorità di generazione di kdesrc-build

I programmi possono girare con diversi livelli di priorità sui sistemi operativi, inclusi Linux® e BSD. Ciò permette al sistema di allocare le risorse per i diversi programmi a seconda della loro importanza.

Normalmente `kdesrc-build` si assegnerà una bassa priorità, in modo che gli altri programmi sul sistema non ne siano influenzati e girino normalmente. Con questa tecnica, `kdesrc-build` userà le risorse quando saranno disponibili.

`kdesrc-build` manterrà comunque un livello di priorità abbastanza alto da girare prima dei processi di routine e prima di programmi di donazione di risorse come `Seti@Home`.

Per modificare `kdesrc-build` in modo che usi permanentemente un livello di priorità più alto (o più basso), devi regolare il valore di `niceness` nel `file di configurazione`. L'impostazione di `niceness` controlla quant'è 'cortese' `kdesrc-build` con gli altri programmi. In parole povere, un valore di `niceness` più alto dà a `kdesrc-build` una priorità minore. Perciò, per dare a `kdesrc-build` una priorità maggiore, devi ridurre il valore di `niceness` (e viceversa). Il valore di `niceness` può andare da 0 (per nulla cortese, priorità massima) a 20 (massima cortesia, priorità minima).

Puoi anche cambiare temporaneamente la priorità di `kdesrc-build` usando l'opzione da riga di comando `--nice`. Il valore dell'opzione è usato esattamente come per il valore di `niceness`.

NOTA

Per alcuni programmi eseguiti dall'amministratore è possibile avere un valore di `niceness` negativo, con una corrispondentemente maggiore priorità. Impostare un valore negativo (o anche zero) di `niceness` per `kdesrc-build` non è una buona idea, perché non migliorerà granché il tempo di generazione ma renderà molto lento il computer se lo dovessi ancora usare.

Manuale dello script kdesrc-build

Per eseguire kdesrc-build con un valore di `nicesness` di 15 (una priorità minore del normale):

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

Oppure puoi cambiare il [file di configurazione](#) per rendere la modifica permanente:

```
nicesness 15
```

SUGGERIMENTO

L'opzione `nicesness` influenza solo l'uso del processore. Un altro effetto notevole sulle prestazioni di un computer è dato dall'uso dell'input e dell'output (I/O) fatto da un programma. Per controllare quanto I/O viene utilizzato da un programma, i moderni sistemi Linux[®] supportano uno strumento simile, chiamato `ionice`. `kdesrc-build` supporta `ionice` (solo per abilitarlo o disabilitarlo completamente), con l'opzione `use-idle-io-priority`, dalla versione 1.12.

6.2.4 Installazione come amministratore

Potresti voler far girare `kdesrc-build` con privilegi di amministratore. Questo potrebbe essere per la (non raccomandata) installazione per tutto il sistema. È tuttavia utile anche quando si usa una generazione di KDE a utente singolo: questo perché alcuni moduli (soprattutto `kdebase`) installano programmi che richiederanno brevemente dei permessi elevati per girare; questi ultimi non possono ottenere questi livelli di permessi a meno di essere installati con permessi elevati.

Potresti semplicemente eseguire `kdesrc-build` direttamente come amministratore, ma ciò non è raccomandato, perché il programma non è stato verificato per un uso del genere. Anche se fa girare il programma in questo modo non dovrebbe essere pericoloso, è meglio evitare di usare i privilegi di amministratore quando è possibile.

Per provvedere a ciò, `kdesrc-build` fornisce l'opzione `make-install-prefix`. Puoi usare questa opzione per specificare un comando da usare per effettuare l'installazione come un altro utente. Il modo raccomandato per usare questo comando è con il programma `Sudo`, che eseguirà il comando di installazione come amministratore.

Per esempio, per installare tutti i moduli usando `Sudo`, potresti fare qualcosa di simile:

```
global
  make-install-prefix sudo
  # Altre opzioni
end global
```

Per usare `make-install-prefix` per un solo modulo, questo dovrebbe funzionare:

```
module nome_modulo_qualcosa
  make-install-prefix sudo
end module
```

6.2.5 Mostrare il progresso della generazione di un modulo

Questa funzionalità è sempre disponibile, ed è attivata automaticamente quando possibile. Quello che fa è visualizzare l'avanzamento stimato durante la generazione di un modulo; in questo modo puoi sapere quanto tempo ci vorrà per generarlo.

6.3 Funzionalità avanzate

6.3.1 Generare parzialmente un modulo

È possibile generare solo parti di un singolo modulo di KDE. Per esempio, potresti voler compilare solo un programma di un modulo. `kdesrc-build` ha delle funzionalità per facilitare questa scelta; ci sono diversi modi per farlo.

6.3.1.1 Rimuovere delle cartelle da una generazione

È possibile scaricare un intero deposito ma far trascurare al sistema alcune cartelle durante la generazione. Questo richiede che il modulo utilizzi CMake e che il sistema di generazione del modulo permetta che la cartella da rimuovere sia opzionale.

Questa azione è controllata con l'opzione `do-not-compile`.

IMPORTANTE

Questa opzione richiede almeno che il sistema di generazione del modulo sia riconfigurato dopo averlo cambiato. Questo si fa usando il comando `kdesrc-build --reconfigure modulo`.

Per rimuovere la cartella `python` dal processo di generazione di `kdebindings`:

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

NOTA

Questa funzione dipende da alcune convenzioni comuni usate nella maggior parte dei moduli di KDE. Perciò potrebbe non funzionare con tutti i programmi.

6.3.2 Supporto per rami e tag in `kdesrc-build`

6.3.2.1 Cosa sono i rami e i tag?

Git supporta la gestione della cronologia del codice di KDE. KDE usa questo supporto per creare rami di sviluppo e per segnare il deposito ogni volta che c'è un nuovo rilascio.

Per esempio, gli sviluppatori di KMail potrebbero lavorare a una nuova funzionalità in un ramo diverso per non interferire con la versione usata dagli altri sviluppatori. Questo ramo contiene dei lavori in corso, mentre il ramo principale (detto `master`) potrebbe averne altri.

Un tag, invece, è un punto specificato nel codice sorgente in un certo momento. Ciò viene usato dalla squadra di amministrazione di KDE per segnare una versione del codice sorgente adatta al rilascio, lasciando allo stesso tempo che gli sviluppatori possano lavorare sul codice.

6.3.2.2 Come usare rami e tag

Il supporto per rami e tag viene gestito da una serie di opzioni, che vanno da una richiesta generica di una versione allo scaricamento di un URL specifico per gli utenti avanzati.

Il modo più semplice è usare le opzioni `branch` e `tag`. Devi solo usare l'opzione insieme al nome del ramo o tag voluto del modulo, e `kdesrc-build` cercherà di determinare la posizione giusta

all'interno del deposito di KDE da cui scaricare. Per la maggior parte dei moduli di KDE funziona tutto bene.

Per scaricare kdelibs di KDE 4.6 (che è noto semplicemente come il ramo 4.6):

```
module kdelibs
  branch 4.6
  # altre opzioni...
end module
```

Oppure, per scaricare kdemultimedia per il rilascio KDE 4.6.1:

```
module kdemultimedia
  tag 4.6.1
  # altre opzioni...
end module
```

SUGGERIMENTO

Puoi specificare un valore globale per il ramo. Se lo fai, non dimenticarti di specificare un ramo diverso per i moduli che non dovrebbero usare quello globale!

6.3.3 Fermare prima la generazione

6.3.3.1 La generazione normalmente prosegue anche se si verificano problemi

kdesrc-build aggiornerà, genererà e installerà normalmente tutti i moduli presenti nell'elenco dei moduli specificato da generare, anche se un modulo restituisce un errore durante la generazione. Questa è di solito una comodità che ti permette di aggiornare i pacchetti software anche se viene fatto un semplice errore in uno dei depositi dei sorgenti durante lo sviluppo che causa la rottura della generazione.

Tuttavia, potresti voler fermare le attività di kdesrc-build quando un modulo non riesce a generare e installare. Questa operazione ti permette di risparmiare il tempo che perderesti nel tentare di fare progressi quando i moduli restanti nell'elenco di generazione non sono in grado di generare correttamente o, in special modo, se non hai ancora generato correttamente i moduli dell'elenco.

6.3.3.2 Non fermare prima con `--no-stop-on-failure`

Il metodo principale per eseguire questa operazione è usare l'opzione a riga di comando `--no-stop-on-failure` quando esegui kdesrc-build.

Questa opzione può essere impostata anche nel [file di configurazione](#) affinché diventi la modalità operativa normale.

È anche possibile istruire kdesrc-build al tempo di esecuzione per fermare la generazione *dopo* il completamento del modulo corrente su cui sta lavorando. È l'opposto dell'interruzione di kdesrc-build con un comando come **Ctrl+C**, che interrompe kdesrc-build immediatamente, perdendo però l'avanzamento del modulo corrente.

IMPORTANTE

L'interruzione di kdesrc-build durante l'installazione di un modulo, quando è abilitata l'opzione `use-clean-install`, significa che il modulo interrotto non sarà disponibile fino a quando kdesrc-build non sarà in grado di generare correttamente il modulo.

Se hai la necessità di interrompere kdesrc-build, in questa situazione, spegnendolo in modo non elegante, prova almeno a evitare di farlo mentre kdesrc-build sta installando un modulo.

6.3.3 Fermare kdesrc-build con eleganza quando stop-on-failure è «false»

Come detto sopra, è possibile spegnere prima kdesrc-build in modo elegante una volta che ha completato il modulo su cui sta lavorando. Per farlo, devi inviare il segnale HUP di POSIX a kdesrc-build.

Puoi farlo con un comando come **pkill** (nei sistemi Linux[®]), nel modo seguente:

```
$ pkill -HUP kdesrc-build
```

Una volta eseguito correttamente, nell'output di kdesrc-build comparirà un messaggio simile a:

```
[ build ] recv SIGHUP, will end after this module
```

NOTA

kdesrc-build potrebbe mostrare più volte questo messaggio a seconda del numero di singoli processi di kdesrc-build attivi. Ciò è normale e non è un'indicazione di errore.

Una volta che kdesrc-build ha riconosciuto il segnale, fermerà l'elaborazione dopo che il modulo corrente è generato e installato. Se kdesrc-build sta ancora aggiornando il codice sorgente quando viene ricevuta la richiesta, kdesrc-build si fermerà dopo che viene completato l'aggiornamento del codice sorgente del modulo. Una volta che entrambi i processi di aggiornamento e generazione vengono fermati prima, kdesrc-build restituirà i risultati parziali e si chiuderà.

6.3.4 Come kdesrc-build cerca di far riuscire la generazione

6.3.4.1 Rigenerazioni automatiche

kdesrc-build includeva delle funzionalità per cercare automaticamente di rigenerare un modulo dopo un errore (e a volte questo tentativo funzionava, a causa di imperfezioni nel sistema di generazione del tempo). Grazie al passaggio a CMake, il sistema di generazione non presenta più questi problemi, e quindi kdesrc-build non cercherà di generare un modulo più di una volta. Ci sono tuttavia situazioni in cui kdesrc-build cercherà autonomamente di fare qualcosa:

- Se cambi le opzioni `configure-flags` o `cmake-options` di un modulo, kdesrc-build se ne accorgerà e rieseguirà automaticamente `configure` o `cmake` per quel modulo.
- Se il sistema di generazione non esiste (anche se kdesrc-build non l'ha eliminato), kdesrc-build lo ricreerà automaticamente. Ciò è utile per poter effettuare un `--refresh-build` completo per un modulo particolare senza doverlo fare su altri.

6.3.4.2 Rigenerare un modulo manualmente

Se cambi le impostazioni di un'opzione di un modulo, o cambia il codice sorgente del modulo in un modo che kdesrc-build non riconosce, potresti dover rigenerare il modulo a mano.

Per farlo basta eseguire `kdesrc-build --refresh-build modulo`.

Se volessi invece che kdesrc-build rigenerasse il modulo automaticamente durante il prossimo aggiornamento normale della generazione, puoi creare un file speciale. Ogni modulo ha una cartella di generazione. Se crei un file di nome `refresh-me` nella cartella di generazione di un modulo, kdesrc-build rigenererà il modulo la prossima volta che partirà il processo di generazione, anche se normalmente avrebbe effettuato la più veloce generazione incrementale.

SUGGERIMENTO

Come impostazione predefinita, la cartella di generazione è `~/kde/build/ modulo /`. Se cambi l'impostazione dell'opzione `build-dir`, usa quella invece di `~/kde/build`.

Rigenerazione usando `refresh-me` per il modulo `kdelibs`:

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

6.3.5 Cambiare le impostazioni delle variabili d'ambiente

Normalmente `kdesrc-build` usa l'ambiente presente all'avvio quando esegue i programmi per effettuare aggiornamenti e generazioni. Ciò è utile per quando si esegue `kdesrc-build` dalla riga di comando.

Tuttavia, potresti voler cambiare le impostazioni delle variabili d'ambiente per cui `kdesrc-build` non fornisce direttamente un'opzione (per esempio, per configurare delle variabili d'ambiente richieste se chiami `kdesrc-build` da un programma come Cron). Ciò è possibile con l'opzione `set-env`.

Al contrario della maggior parte delle opzioni, può essere impostata più volte, e accetta due voci, separate da spazi. La prima è il nome della variabile d'ambiente da impostare, e il resto della riga è il suo valore.

Impostare `DISTRO=BSD` per tutti i moduli:

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

6.3.6 Riprendere le generazioni

6.3.6.1 Riprendere una generazione non riuscita o annullata

Puoi dire a `kdesrc-build` di iniziare a generare da un modulo diverso da quello che sceglierebbe normalmente. Ciò può essere utile quando una serie di moduli non è riuscita, o se hai annullato una generazione a metà. Puoi controllare questo comportamento con le opzioni `--resume-from` e `--resume-after`.

NOTA

Alcune versioni precedenti di `kdesrc-build` saltavano l'aggiornamento del sorgente alla ripresa di una generazione. Questa non è più l'impostazione predefinita, ma puoi sempre usare l'opzione da riga di comando `--no-src` per saltare l'aggiornamento.

Riavvio della generazione a partire da `kdebase`:

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

Riprendere la generazione dopo `kdebase` (nel caso che hai corretto il problema manualmente e hai installato il modulo da te):

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

Se l'ultima generazione di `kdesrc-build` si è conclusa con un errore, puoi usare anche l'opzione `--resume`, che riprende l'ultima generazione a partire dal modulo che ha provocato l'errore. Gli aggiornamenti di codice sorgente e metadati vengono saltati (ma se ti servono, è di solito meglio usare `--resume-from`).

6.3.6.2 Ignorare dei moduli in una generazione

Analogamente a come si può [riprendere la generazione di un modulo](#), si può invece scegliere di aggiornare e generare tutto normalmente, ma ignorando una serie di moduli.

Puoi farlo usando l'opzione `--ignore-modules`. Questa opzione dice a kdesrc-build di ignorare tutti i moduli sulla riga di comando durante l'aggiornamento e la generazione.

Ignorare extragear/multimedia e kdereview in un'esecuzione completa:

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

6.3.7 Cambiare le opzioni dalla riga di comando

6.3.7.1 Cambiare le opzioni globali

Puoi cambiare le impostazioni delle opzioni lette dal [file di configurazione](#) direttamente dalla riga di comando. Questa modifica si sostituirà alle impostazioni del file di configurazione, ma solo temporaneamente. Ha effetto solo fintanto che è presente sulla riga di comando.

kdesrc-build ti permette di cambiare opzioni con un nome come *nome-opzione* passando un argomento sulla riga di comando nella forma `--nome-opzione=valore`. kdesrc-build capirà se l'opzione non è tra le proprie, e ne cercherà il nome nel suo elenco di opzioni. Se non lo troverà, ti avvertirà, altrimenti ricorderà il valore impostato e sostituirà qualsiasi impostazione del file di configurazione.

Impostare l'opzione `source-dir` a `/dev/null` per fare una prova:

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

6.3.7.2 Cambiare le opzioni dei moduli

È anche possibile cambiare le opzioni solo per un modulo specifico. La sintassi è simile: `--modulo,nome-opzione=valore`.

Questa modifica si sostituisce a qualsiasi impostazione simile per il modulo trovata nel [file di configurazione](#), e si applica solo quando l'opzione viene passata dalla riga di comando.

Usare una diversa cartella di generazione per il modulo kdedu:

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

6.4 Funzionalità per gli sviluppatori di KDE

6.4.1 Controlli dell'agente SSH

kdesrc-build può assicurare che gli sviluppatori di KDE che usano SSH per accedere al deposito del codice sorgente di KDE non dimentichino di tenere attivato l'agente SSH. Ciò potrebbe far fermare indefinitamente kdesrc-build in attesa della password SSH dello sviluppatore, quindi, come impostazione predefinita, kdesrc-build controllerà se l'agente è in esecuzione prima di aggiornare il codice sorgente.

NOTA

Questo è concepito solo per gli sviluppatori di KDE che utilizzano SSH.

Potresti voler disattivare il controllo dell'agente SSH, nel caso che si verificano situazioni in cui kdesrc-build rilevi erroneamente la presenza dell'agente. Per disattivare il controllo dell'agente, imposta l'opzione `disable-agent-check` a **true**.

Disattivare il controllo dell'agente SSH:

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

6.5 Altre funzionalità di kdesrc-build

6.5.1 Cambiare la quantità di output da kdesrc-build

kdesrc-build ha diverse opzioni per controllare la quantità di output generato dallo script. In ogni caso, gli errori saranno sempre segnalati.

- L'opzione `--quiet` (la sua forma breve è `-q`) fa stare kdesrc-build per lo più zitto. Solo i messaggi importanti, gli avvertimenti e gli errori saranno mostrati. Quando disponibile, il progresso della generazione sarà ancora mostrato.
- L'opzione `--really-quiet` (senza forma breve) fa visualizzare a kdesrc-build solo gli avvisi importanti o gli errori durante l'esecuzione.
- L'opzione `--verbose` (la sua forma breve è `-v`) fa generare a kdesrc-build dell'output molto dettagliato.
- L'opzione `--debug` è solo per scopi di prova e correzione, fa funzionare kdesrc-build come se `--verbose` fosse attiva, mette i comandi nell'output del terminale, e visualizzerà informazioni di debug per molte funzioni.

6.5.2 Output colorato

Anche se viene eseguito da Konsole o da un altro terminale, kdesrc-build sarà normalmente visualizzato con del testo colorato.

Puoi disattivarlo usando l'opzione `--no-color` sulla riga di comando, o impostando l'opzione `colorful-output` nel file di configurazione a **false**.

Disattivare l'output colorato nel file di configurazione:

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

6.5.3 Rimuovere cartelle non necessarie dopo la generazione

Se hai poco spazio su disco ma vuoi comunque avere l'ultimo codice sorgente di KDE, kdesrc-build può ridurre l'occupazione del disco durante la generazione di KDE da Git.

NOTA

Tieni presente che generare KDE porta via molto spazio. Ci sono diverse parti ingombranti durante l'uso di kdesrc-build:

1. Il codice sorgente stesso può portar via un bel po' di spazio. I moduli predefiniti portano via 1,6 gigabyte di spazio su disco. Puoi ridurre questa quantità assicurandoti di generare solo i moduli che vuoi davvero. kdesrc-build non eliminerà il codice sorgente dal disco anche se elimini la voce corrispondente dal [file di configurazione](#), quindi assicurati di eliminare il codice sorgente inutilizzato. Nota che i file sorgenti sono scaricati da Internet, e *non* dovresti eliminarli se li stai usando, almeno fino a quando non hai finito di usare kdesrc-build.

Inoltre, se hai già una versione di Qt™ installata dalla tua distribuzione (e ce ne sono buone probabilità), probabilmente non hai bisogno di installare il modulo qt. Ciò risparmierà circa 200 megabyte di spazio su disco.

2. kdesrc-build creerà una cartella di generazione separata per generarci il codice sorgente. A volte kdesrc-build dovrà copiare una cartella di codice sorgente in una cartella di generazione finta. Quando questo avviene, si usano dei collegamenti simbolici per risparmiare spazio, per cui non dovrebbe creare problemi d'ingombro. La cartella di generazione sarà di solito molto più grande della cartella del codice sorgente di un modulo. Per esempio, la cartella di generazione di kdatabase è di circa 1050 megabyte, mentre il codice sorgente di kdatabase è solo di circa 550 megabyte.

Fortunatamente, la cartella di generazione non è necessaria dopo che un modulo è stato correttamente generato e installato. kdesrc-build può rimuovere automaticamente la cartella di generazione dopo l'installazione di un modulo; vedi gli esempi sotto per maggiori informazioni. Nota che questo passo renderà impossibile a kdesrc-build l'uso di veloci generazioni incrementali.

3. Infine, c'è lo spazio necessario per l'installazione di KDE stessa, che non va dalla cartella di generazione. Questa usa tipicamente meno spazio della cartella di generazione. È però difficile avere numeri esatti.

Come si riducono i requisiti di spazio di KDE? Un modo è usare i flag di compilazione giusti, ottimizzando per ridurre lo spazio invece che per la velocità. Un altro modo che può avere un effetto notevole è rimuovere le informazioni di debug dalla generazione di KDE.

ATTENZIONE

Dovresti sapere bene quello che stai facendo prima di decidere di rimuovere le informazioni di debug. Usare programmi di ultimissima generazione vuol dire che stai usando programmi potenzialmente molto più inclini ad andare in crash di un rilascio stabile. Se stai usando programmi senza informazioni di debug, potrebbe essere molto difficile creare una buona segnalazione di errori per far risolvere il problema, e probabilmente dovrai riabilitare il debug per l'applicazione affetta e rigenerarla per aiutare uno sviluppatore a correggere il problema. Quindi, rimuovi le informazioni di debug a tuo rischio e pericolo!

Rimuovere la cartella di generazione dopo l'installazione di un modulo. La cartella del codice sorgente è tenuta, e il debug è abilitato:

```
global
  configure-flags      --enable-debug
  remove-after-install builddir      # Rimuovi la cartella di generazione ←
    dopo l'installazione
end global
```

Rimuovere la cartella di generazione dopo l'installazione, senza informazioni di debug, con ottimizzazione per le dimensioni.

```
global
  cxxflags             -Os             # Ottimizza per le dimensioni
  configure-flags      --disable-debug
  remove-after-install builddir      # Rimuovi la cartella di generazione ←
    dopo l'installazione
end global
```

Capitolo 7

CMake, il sistema di generazione di KDE

7.1 Introduzione a CMake

Nel marzo 2006, il programma CMake sconfisse diversi concorrenti e fu selezionato per essere il sistema di generazione di KDE 4, sostituendo il sistema basato sugli Autotools che KDE aveva usato fin dall'inizio.

È disponibile un'introduzione a CMake sul [wiki della Comunità di KDE](#). Fondamentalmente, invece di eseguire `make -f Makefile.cvs`, quindi `configure`, e quindi `Make`, dobbiamo semplicemente eseguire CMake e poi `Make`.

kdesrc-build supporta CMake. Alcune funzionalità di kdesrc-build erano in realtà funzionalità per il sistema di generazione sottostante, incluse `configure-flags` e `do-not-compile`. Quando sono disponibili funzionalità equivalenti, sono fornite. Per esempio, l'equivalente dell'opzione `configure-flags` è `cmake-options`, e anche l'opzione `do-not-compile` è supportata per CMake nella versione 1.6.3 di kdesrc-build.

Capitolo 8

Riconoscimenti e licenza

Traduzione di Federico Zenith federico.zenith@member.fsf.org e Valter Murav valtermura@gmail.com

Questa documentazione è concessa in licenza sotto i termini della [GNU Free Documentation License](#).

Appendice A

Moduli di KDE e organizzazione del codice sorgente

A.1 Il 'modulo'

KDE raggruppa i suoi programmi in 'moduli' di varie dimensioni. Questi erano inizialmente un vago raggruppamento di alcuni grandi moduli, ma con l'introduzione dei [depositi di codice sorgente](#) basati su [Git](#) questi grandi moduli sono stati divisi in molti moduli più piccoli.

Anche kdesrc-build usa questo concetto di modulo. Essenzialmente, un 'modulo' è un raggruppamento di codice che può essere scaricato, generato, testato e installato.

A.1.1 Moduli singoli

È facile far generare un singolo modulo a kdesrc-build. L'elenco seguente è un esempio di come può essere scritta una dichiarazione per un modulo basato su Git nel [file di configurazione](#).

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

SUGGERIMENTO

Questo è un modulo basato su Git perché non usa un'opzione [repository](#). Inoltre, l'opzione `cmake-options` è elencata solo come esempio e non è necessaria.

A.1.2 Gruppi di moduli collegati

La maggior parte dei moduli di KDE sono oggi basati su Git, e sono normalmente combinati in gruppi di moduli.

kdesrc-build supporta pertanto anche i gruppi di moduli, con gli [insiemi di moduli](#). Per esempio:

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

SUGGERIMENTO

Se preferisci, puoi lasciare il nome dell'insieme di moduli (*moduli-base*). L'impostazione di `repository` dice a `kdesrc-build` da dove scaricare il codice sorgente, ma puoi anche usare un URL `git://`.

Una funzionalità speciale di 'repository `kde-projects`' è che `kdesrc-build` includerà automaticamente tutti i moduli Git raggruppati sotto i moduli da te elencati (nella banca dati dei progetti di KDE).

A.1.3 'Gruppi di rami' dei moduli

Estendendo il concetto di [gruppo di moduli](#), gli sviluppatori di KDE hanno notato che sincronizzare il nome dei rami di Git in molti depositi stava diventando difficile, soprattutto durante il lavoro per le nuove infrastrutture di KDE per Qt™ 5.

È stato quindi sviluppato il concetto di 'gruppi di rami', per permettere a utenti e sviluppatori di selezionare uno tra pochi gruppi, e permettere allo script di selezionare automaticamente il ramo di Git appropriato.

`kdesrc-build` supporta questa funzionalità dalla versione 1.16-pre2, attraverso l'opzione [branch-group](#).

Example A.1 Esempio dell'uso dei gruppi di rami

I gruppi di rami si possono usare nel file di configurazione come segue:

```
global
  # Seleziona KDE Frameworks 5 e altre applicazioni basate su Qt5
  branch-group kf5-qt5

  # Altre opzioni globali vanno qui...
end global

module-set
  # branch-group funziona solo per kde-projects
  repository kde-projects

  # branch-group viene ereditato da quello impostato globalmente,
  # ma potrebbe essere specificato qui.

  use-modules kdelibs kde-workspace
end module-set

# Il ramo di kdelibs sarà «frameworks»
# Il ramo di kde-workspace sarà «master» (nell'agosto 2013)
```

In questo caso lo stesso `branch-group` produce diversi nomi di rami per ogni modulo di Git.

Questa funzionalità richiede alcuni dati gestiti dagli sviluppatori di KDE in un deposito Git chiamato `kde-build-metadata`, però questo modulo verrà incluso automaticamente da `kdesrc-build` (anche se lo potresti veder apparire nell'output dello script).

SUGGERIMENTO

I moduli di KDE che non hanno il nome di un ramo impostato per il gruppo di rami che scegli saranno predefiniti al nome di un ramo appropriato, come se non avessi specificato `branch-group`.

Appendice B

Procedure sostituite di configurazione dei profili

B.1 Configurare in profilo d'accesso di KDE

Queste istruzioni indicano come configurare il profilo necessario per assicurare che il computer possa accedere al desktop Plasma di KDE appena generato. `kdesrc-build` cercherà normalmente di farlo da sé (vedi Sezione 2.5.1). Questa sezione dell'appendice può essere utile per chi non può usare il supporto di `kdesrc-build` per la configurazione del profilo di accesso. Tuttavia le istruzioni potrebbero non sempre essere aggiornate, e può essere utile consultare il file `kde-env-master.sh.in` incluso nel codice sorgente di `kdesrc-build`.

B.1.1 Cambiare le impostazioni del profilo di avvio

IMPORTANTE

Il file `.bash_profile` contiene le impostazioni di accesso alla popolare shell Bash usata da molte distribuzioni Linux[®]. Se usi una shell diversa, potresti dover correggere i campioni forniti in questa sezione per la tua shell.

Apri o crea il file `.bash_profile` nella cartella Home con un qualsiasi editor di testo. Se stai generando il modulo qt (che è l'impostazione predefinita), aggiungi:

```
QTDIR=(percorso a cartella qt) # Come ~/kdesrc/build/qt, predefinito
KDEDIR=(percorso a cartella kde) # Come ~/kde/usr, predefinito
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH
MANPATH=$QTDIR/doc/man:$MANPATH

# Fai la cosa giusta se LD_LIBRARY_PATH non è già impostata.
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
```


Manuale dello script kdesrc-build

Oppure, se non stai generando qt (e stai usando le librerie Qt™ del sistema), aggiungi:

```
KDEDIR=(percorso a kdedir) # Come ~/kde/usr, predefinito.
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH

# Fai la cosa giusta se LD_LIBRARY_PATH non è già impostata.
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
```

Se non stai usando un utente apposito, imposta una diversa variabile \$KDEHOME per il nuovo ambiente nel file `.bash_profile`:

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

# Crealo se necessario
[ ! -e ~/.kde-git ] && mkdir ~/.kde-git
```

NOTA

Se in seguito il menu K è vuoto o troppo affollato da applicazioni della tua distribuzione, potresti dover impostare le variabili d'ambiente XDG nel tuo `.bash_profile`:

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

B.1.2 Avviare KDE

Adesso che hai adattato le impostazioni d'ambiente per usare il KDE giusto, è importante assicurarsi che sia usato anche lo script `startkde` giusto.

Apri il file di testo `.xinitrc` dalla cartella Home, o crealo se necessario. Aggiungi la riga:

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

IMPORTANTE

Su alcune distribuzioni, potrebbe essere necessario effettuare gli stessi passi con il file `.xsession`, anch'esso nella cartella Home. Ciò è vero in particolare quando si usano gestori di accesso grafici come `sddm`, `gdm` o `xdm`.

Adesso, avvia il tuo nuovo KDE: nei sistemi BSD e Linux® con supporto per i terminali virtuali, le combinazioni di tasti da **Ctrl+Alt+F1** a **Ctrl+Alt+F12** vengono usate per selezionare le console virtuali dalla 1 alla 12. Ciò permette di avere più di un ambiente desktop allo stesso tempo. Le prime sei sono terminali di testo, e le seguenti sono terminali grafici.

Se invece quando avvii il computer ti viene presentato un gestore grafico, puoi usare il nuovo ambiente KDE anche se non è elencato nelle opzioni. La maggior parte dei gestori d'accesso

Manuale dello script kdesrc-build

grafici, sddm incluso, hanno l'opzione di usare una 'sessione personalizzata' all'accesso. Con questa opzione, vengono caricate le tue impostazioni di sessione dal file `.xsession` nella tua cartella Home. Se hai già modificato il file come descritto sopra, questa opzione dovrebbe farti entrare nella nuova installazione di KDE.

Se non lo fa, c'è qualcos'altro da provare che di solito funziona: Premi **Ctrl+Alt+F2**, e ti troverai davanti a un terminale testuale. Accedi con l'utente dedicato e scrivi:

```
start -- :1
```

SUGGERIMENTO

Puoi usare il KDE dal codice sorgente e il vecchio KDE allo stesso tempo! accedi usando il tuo utente regolare, avvia il desktop stabile di KDE. Premi **Ctrl+Alt+F2** (o **F1**, **F3**, eccetera) e ti sarà presentato un terminale testuale. Accedi con l'utente Git dedicato a KDE e scrivi:

```
start -- :1
```

Puoi tornare al desktop KDE del tuo utente regolare premendo il tasto scorciatoia per il desktop già in esecuzione. Questo è normalmente **Ctrl+Alt+F7**, ma potrebbe anche essere **F6** o **F8**. Per tornare al tuo KDE compilato con `kdesrc-build`, devi usare la stessa sequenza, ma con il prossimo tasto funzione. Per esempio, se hai dovuto inserire **Ctrl+Alt+F7** per andare al KDE normale, devi premere **Ctrl+Alt+F8** per tornare al KDE di `kdesrc-build`.