

Підручник зі скрипту kdesrc-build

Michael Pyne

Carlos Woelz

Переклад українською: Юрій Чорноіван



Підручник зі скрипту kdesrc-build

Зміст

1	Вступ	8
1.1	Короткий вступ до kdesrc-build	8
1.1.1	Для чого призначено kdesrc-build?	8
1.1.2	Робота kdesrc-build «у пісочниці»	8
1.2	Огляд документації	9
2	Початок роботи	10
2.1	Підготовка системи до збирання KDE	10
2.1.1	Створення нового облікового запису користувача	10
2.1.2	Перевірка готовності вашої системи до збирання програмного забезпечення KDE з початкових кодів	10
2.1.3	Налаштування kdesrc-build	12
2.1.3.1	Встановлення kdesrc-build	12
2.1.3.2	Підготовка файлу налаштувань	12
2.1.3.2.1	Ручне визначення параметрів у файлі налаштувань	12
2.2	Визначення даних налаштувань	13
2.3	Використання скрипту kdesrc-build	14
2.3.1	Завантаження метаданих проєкту	14
2.3.2	Попередній перегляд того, що трапиться, якщо запустити kdesrc-build	14
2.3.3	Усування проблем під час збирання	15
2.4	Збирання окремих модулів	16
2.5	Налаштовування середовища на запуск зібраного стільничного середовища Плазми KDE	17
2.5.1	Автоматичне встановлення драйвера входу до системи	18
2.5.1.1	Додавання підтримки xsession у дистрибутив	18
2.5.1.2	Додавання підтримки xsession вручну	18
2.5.2	Налаштовування середовища вручну	19
2.6	Впорядкування та вибір модулів	19
2.6.1	Структура модулів програмного забезпечення KDE	19
2.6.2	Вибір модулів для збирання	20
2.6.3	Набори модулів	20
2.6.3.1	Концепція наборів базових модулів	20
2.6.3.2	Особлива підтримка для наборів модулів KDE	21
2.6.4	Офіційна база даних модулів KDE	22
2.6.5	Відфільтровування модулів проєкту KDE	23
2.7	Початкові висновки	23

3	Можливості скрипту	25
3.1	Огляд можливостей	25
3.2	Запис журналів збирання kdesrc-build	27
3.2.1	Огляд ведення журналу	27
3.2.1.1	Компонування каталогів журналу	27
4	Налаштування kdesrc-build	28
4.1	Огляд налаштування kdesrc-build	28
4.1.1	Компонування файла налаштувань	28
4.1.1.1	Загальні налаштування	28
4.1.1.2	Налаштування модулів	28
4.1.1.3	Модулі «options»	29
4.1.2	Включенні інших файлів налаштувань	30
4.1.3	Типові параметри налаштування	30
4.2	Таблиця параметрів налаштування	31
5	Параметри командного рядка і змінні середовища	72
5.1	Користування за допомогою командного рядка	72
5.1.1	Типові параметри командного рядка	72
5.1.2	Визначення модулів для збирання	73
5.2	Підтримувані змінні середовища	73
5.3	Параметри командного рядка	73
6	Користування kdesrc-build	81
6.1	Передмова	81
6.2	Основні можливості kdesrc-build	81
6.2.1	Підтримка qt	81
6.2.2	Стандартні прапорці, які додаються kdesrc-build	82
6.2.3	Зміна пріоритету збирання kdesrc-build	82
6.2.4	Встановлення від імені суперкористувача	83
6.2.5	Показ поступу збирання модуля	83
6.3	Додаткові можливості	84
6.3.1	Часткове збирання модуля	84
6.3.1.1	Отримання частин модуля	84
6.3.1.2	Вилучення каталогів з процесу збирання	84
6.3.2	Підтримка версій і випусків у kdesrc-build	85
6.3.2.1	Що таке версія (branch) і випуск (tag)?	85
6.3.2.2	Як користуватися версіями і випусками?	85
6.3.2.3	Додаткові параметри підтримки версій	86
6.3.3	Способи, у які kdesrc-build намагається забезпечити успішне збирання	86
6.3.3.1	Автоматичне перезбирання	86
6.3.3.2	Перезбирання модуля вручну	87
6.3.4	Зміна параметрів змінних середовища	87

6.3.5	Поновлення збирання	88
6.3.5.1	Поновлення збирання, яке завершилося помилкою або було скасоване	88
6.3.5.2	Ігнорування модулів під час збирання	88
6.3.6	Зміна параметрів за допомогою командного рядка	88
6.3.6.1	Зміна загальних параметрів	88
6.3.6.2	Зміна параметрів для модулів	89
6.4	Можливості, призначені для розробників KDE	89
6.4.1	Перевірки агента SSH	89
6.5	Інші можливості kdesrc-build	89
6.5.1	Зміна рівня докладності виводу kdesrc-build	89
6.5.2	Виведення у кольорі	90
6.5.3	Вилучення непотрібних каталогів після збирання	90
7	CMake, система збирання KDE	92
7.1	Вступ до CMake	92
8	Подяки і ліцензування	93
A	Модулі KDE та структура початкових кодів	94
A.1	«Модуль»	94
A.1.1	Окремі модулі	94
A.1.2	Групи пов'язаних модулів	94
A.1.3	Модуль «групи гілок»	95
B	Процедури налаштування профілю, які заміщуються програмою	96
B.1	Налаштування профілю входу до системи KDE	96
B.1.1	Зміна ваших параметрів профілю запуску	96
B.1.2	Запуск KDE	97

Перелік таблиць

4.1	Таблиця параметрів	71
-----	------------------------------	----

Анотація

`kdesrc-build` — це скрипт, який збирає і встановлює програми KDE безпосередньо зі сховищ кодів проекту KDE.

Розділ 1

Вступ

1.1 Короткий вступ до kdesrc-build

1.1.1 Для чого призначено kdesrc-build?

kdesrc-build — скрипт, який допомагає учасникам спільноти KDE встановлювати програмне забезпечення KDE зі сховищ [Git](#) та [Subversion](#) та неперервно його оновлювати. Цей скрипт, зокрема, призначено для тих, хто хоче тестувати та розробляти програмне забезпечення KDE, включно з користувачами, які хочуть переконатися у виправленні певної вади, та розробниками, які працюють над новими можливостями.

kdesrc-build може бути налаштовано для супроводу окремого модуля, всього середовища Плазми з набором програм KDE або якогось проміжного варіанта.

Для початку, ознайомтеся з розділом розділ 2 чи читайте підручник далі, щоб докладніше розібратися із принципами роботи kdesrc-build та питаннями, розібраними у цьому підручнику.

1.1.2 Робота kdesrc-build «у пісочниці»

kdesrc-build працює з використанням інструментів, доступ до яких користувач може отримати з командного рядка із використанням тих самих інтерфейсів, які доступні користувачу. Під час роботи kdesrc-build виконується така послідовність дій:

1. kdesrc-build читає параметри [командного рядка](#) та [файли налаштувань](#) для визначення частин, які слід зібрати, параметрів збирання, місця встановлення тощо.
2. kdesrc-build виконує оновлення початкових кодів для кожного з [модулів](#). Оновлення продовжується, доки всі модулі не буде оновлено. Модулі, які не вдасться оновити у звичайному режимі, не зупинять збирання, — програма просто повідомить наприкінці збирання, які модулі не вдалося оновити.
3. Модулі, які було успішно оновлено, буде зібрано, для них буде запущено відповідні комплекси тестування, а потім їх буде встановлено. Для зменшення загального часу обробки kdesrc-build типово запускатиме збирання коду, щойно завершиться оновлення першого з модулів. Решта модулів при цьому оновлюватиметься у фоновому режимі.

ПІДКАЗКА

Чудовий огляд збирання модулів KDE із інформативними діаграмами наведено у [статті із обговоренням збирання програми KDE Krita](#). Саме у такий спосіб kdesrc-build автоматизує збирання усіх модулів KDE.

1.2 Огляд документації

У цьому підручнику подано огляд таких аспектів роботи kdesrc-build:

- [Огляд](#) початкових кроків.
- Основні [можливості](#).
- Синтаксис та параметри [файла налаштувань](#).
- [Команди та параметри](#).

Крім того, задокументовано кроки, які вам слід виконати за допомогою сторонніх інструментів (іншими словами, кроків, які kdesrc-build не виконує у автоматичному режимі).

Розділ 2

Початок роботи

У цій главі ми продемонструємо, як скористатися `kdesrc-build` для отримання модулів зі сховища KDE і їх збирання. Тут ми наведемо базові пояснення щодо структури системи керування версіями KDE і кроків, які вам слід виконати до запуску скрипту.

Всі питання, обговорені у цій главі, докладніше викладено у [статті щодо збирання з початкових кодів](#), розміщеній на [вебсайті вікі спільноти KDE](#). Якщо ви компілюєте KDE вперше, вам варто прочитати цей підручник або проконсультуватися з цим довідником. Там ви знайдете докладні відомості щодо інструментів пакування та потрібних пакунків, типових проблем з компіляцією, стратегій і запуску нововстановленого KDE.

2.1 Підготовка системи до збирання KDE

2.1.1 Створення нового облікового запису користувача

Рекомендуємо вам скористатися окремим обліковим записом користувача для збирання, встановлення і запуску програмного забезпечення KDE, оскільки таким чином ви зменшите об'єм роботи зі зміни прав доступу та уникнете перешкод, пов'язаних з псуванням налаштувань пакунків з вашого дистрибутива. Якщо у системі вже встановлено пакунки KDE, найкращим вибором буде створення окремого облікового запису для збирання і запуску нового KDE.

ПІДКАЗКА

Якщо ви не чіпатимете загальносистемного KDE, у вас завжди буде запасний варіант на випадок, якщо помилка у програмному кодї зробить зібрані вами програми непридатними до користування.

Якщо хочете, можете усе так налаштувати, щоб встановлення відбувалося до загальносистемного каталогу (наприклад `/usr/src/local`). Тут ми не обговорюватимемо цей тип встановлення, оскільки ми припускаємо, що ви ознайомлені зі наслідками своїх дій.

2.1.2 Перевірка готовності вашої системи до збирання програмного забезпечення KDE з початкових кодів

Перш ніж почати користування скриптом `kdesrc-build` (або будь-яким іншим інструментом для збирання), вам слід встановити інструменти для розробки і бібліотеки, потрібні для KDE. Майже повний список потрібних інструментів можна знайти на сторінці [вікі спільноти KDE](#).

Ось список деяких з потрібних для збирання програм:

- Вам знадобиться CMake. Цю програму KDE використовує для обробки налаштувань початкового коду під час збирання та створення специфічних команд збирання для вашої системи. Номер потрібної вам версії залежить від версій програмного забезпечення KDE, які ви збираєте (подробіці див. на TechBase). Якщо ваш дистрибутив достатньо сучасний, включеного до нього пакунка CMake має бути достатньо.
- Крім того, вам слід встановити клієнтське програмне забезпечення для отримання початкових кодів KDE. Це означає, що вам слід встановити такі пакунки:
 - Вам слід встановити **систему керування кодом Git**, яка використовується для **початкового коду** усіх проєктів KDE.
 - Хоча збирання відповідного модуля (libdbusmenu-qt) і не є обов'язковим, вам може знадобитися система керування версіями **Bazaar** (цей модуль потрібен для збирання бібліотек KDE). Для більшості користувачів цю бібліотеку можна встановити за допомогою системи керування пакунками дистрибутива, але у kdesrc-build передбачено і можливість її збирання. Для збирання libdbusmenu-qt вам слід встановити Bazaar.
- Скриптова мова Perl потрібна для роботи kdesrc-build, взаємодії із деякими сховищами KDE та Qt™ (якщо ви збираєте цю бібліотеку з початкових кодів).

Пакунки Perl, які є частиною вашого дистрибутива, мають підійти (потрібна Perl версії не нижчої за 5.14), але вам також знадобляться декілька додаткових модулів (kdesrc-build попередить вас, якщо у системі їх не буде виявлено):

 - IO::Socket::SSL
 - JSON::PP або JSON::XS
 - YAML::PP, YAML::XS або YAML::Syck
- Вам знадобиться усе середовище розробки мовою C++ (компілятор, стандартна бібліотека, середовище виконання та усі пов'язані пакунки для розробки). Мінімальна потрібна версія може бути різною — усе залежить від модуля KDE: збірка KDE Frameworks 5 може бути зібрана найдавнішими компіляторами, а Плазма 5 KDE та програми KDE можуть вимагати для збирання найновіших версій компілятора.

Мінімальними рекомендованими версіями є GCC 4.8 або Clang 4. У багатьох дистрибутивах передбачено просте встановлення цих інструментів за допомогою метапакунка «build-essentials». Крім того, можна встановити «залежності для збирання» Qt™ або подібні набори пакунків. На вікі спільноти KDE є [сторінка зі списком рекомендованих пакунків для основних дистрибутивів](#).
- Вам знадобиться засіб збирання, який, власне, виконуватиме кроки компіляції (які було визначено CMake). Рекомендуємо скористатися GNU Make, який має бути доступним до встановлення за допомогою програми для керування пакунками вашого дистрибутива. У CMake передбачено підтримку інших засобів збирання, зокрема Ninja. Цими засобами збирання можна скористатися з kdesrc-build за допомогою параметра файлу налаштувань `custom-build-command`.
- Нарешті, вам знадобляться відповідні бібліотеки Qt™ (разом із пакунками для розробки) для версії програмного забезпечення KDE, яку ви збираєте. Офіційно, у kdesrc-build не передбачено підтримки збирання Qt™ 5 (поточної основної версії бібліотеки), отже, рекомендуємо вам скористатися пакунками для розробки з вашого дистрибутива або ознайомитися із настановами з вікі спільноти KDE щодо [збирання Qt 5 власноруч](#).

ПРИМІТКА

У більшості операційних систем передбачено спосіб простого встановлення потрібних інструментів для розробки. Зверніться до [сторінки вікі спільноти щодо потрібних для розробки пакунків](#), щоб ознайомитися з готовими настановами.

ВАЖЛИВО

Деякі з цих пакунків поділено на бібліотеки (або програми) і пакунки для розробки. Вам знадобляться принаймні програма або бібліотека і її пакунок для розробки.

2.1.3 Налаштування kdesrc-build

2.1.3.1 Встановлення kdesrc-build

Розробники KDE постійно вдосконалюють kdesrc-build, підтримуючи його синхронізацію із поступом у розробці KDE, зокрема, поліпшуючи рекомендовані налаштування kdesrc-build, додаючи модулі, удосконалюючи прапорці CMake тощо.

Через це, ми рекомендуємо отримати код kdesrc-build безпосередньо з початкового сховища коду, а потім періодично оновлювати програму.

Ви можете отримати kdesrc-build зі сховища його коду за допомогою такої команди:

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

Замініть ~/kdesrc-build на адресу каталогу, до якого слід встановити програму.

Згодом, ви можете оновити код kdesrc-build за допомогою такої команди:

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

ПІДКАЗКА

Ми рекомендуємо додати каталог встановлення kdesrc-build до вашої змінної середовища PATH, щоб ви могли запускати kdesrc-build, не вказуючи шлях до цього каталогу під час кожного запуску скрипту.

2.1.3.2 Підготовка файлу налаштувань

kdesrc-build використовує [файл налаштувань](#) (зберігається у ~/.kdesrc-buildrc) для визначення модулів, які слід зібрати, місць, куди слід встановлювати ці модулі, тощо.

Ви можете скористатися програмою, включеною до складу kdesrc-build з назвою kdesrc-build-setup з метою приготування простих налаштувань kdesrc-build. Після цього ви можете внести зміни до ~/.kdesrc-buildrc відповідно до ваших уподобань.

Програма kdesrc-build-setup запускається з терміналу (замість використання графічного інтерфейсу), подібно до kdesrc-build, отже ви можете скористатися нею, навіть якщо графічний інтерфейс недоступний.

2.1.3.2.1 Ручне визначення параметрів у файлі налаштувань

Крім того, ви можете створити власний файл налаштувань вручну. Для цього скопіюйте включений зразок файлу налаштувань kdesrc-buildrc-kf5-sample до ~/.kdesrc-buildrc, а потім внесіть до нього зміни. Корисні настанови щодо цього викладено у розділі розділ 4. Особливо корисною може бути [таблиця параметрів налаштування](#).

До складу kdesrc-build включено багато рекомендованих файлів налаштувань, у яких реалізовано підтримку KDE Frameworks 5, Плазми 5 та інших програм KDE. Скрипт kdesrc-build-setup посилається на ці файли у створеному ним файлі налаштувань, але ви можете також зробити це власноруч. Див. розділ Розділ 4.1.2, щоб дізнатися більше про те, як використати інші файли налаштувань у вашому власному ~/.kdesrc-buildrc.

Докладніші відомості щодо синтаксису [файла налаштувань](#) можна знайти у розділах Розділ 2.2 і розділ 4.

2.2 Визначення даних налаштувань

Щоб користуватися kdesrc-build, вам слід мати файл з назвою `.kdesrc-buildrc` у вашому домашньому каталозі, у цьому файлі визначаються загальні параметри і перелік модулів, які ви бажаєте отримати і зібрати.

ПРИМІТКА

Ви можете скористатися іншим файлами налаштувань kdesrc-build, відповідну процедуру описано у розділі розділ 4. Якщо у вас є потреба у використанні декількох наборів налаштувань, будь ласка, ознайомтеся з цим розділом. Надалі ми вважатимемо, що налаштування зберігаються у файлі `~/kdesrc-buildrc`.

Найпростішим способом початкового налаштування є використання файла `kdesrc-buildrc-kf5-sample` як зразка. Достатньо змінити загальні параметри так, щоб вони відповідали вашим потребам, а також змінити список модулів, які ви бажаєте зібрати.

Типових параметрів буде достатньо для збирання KDE. Ймовірно, що серед параметрів, які ви захочете змінити, можуть бути:

- `kdedir`, який змінює каталог призначення, куди буде встановлено програмне забезпечення KDE. Типовим є каталог `~/kde`, тобто середовище буде встановлено для одного (вашого) користувача.
- `branch-group` — параметр, яким можна скористатися для вибору відповідної гілки розробки модулів KDE у цілому. Передбачено підтримку багатьох варіантів збирання, але, найімовірніше, вам потрібна гілка `kf5-qt5`. Якщо використано саме цю гілку, kdesrc-build отримає найсвіжіший код на основі Qt™ 5 та KDE Frameworks 5.

ПІДКАЗКА

Якщо ви не виберете якоїсь гілки, kdesrc-build використовуватиме типову, але типова гілка з часом змінюватиметься, тому варто вибрати гілку власноруч, щоб зміна гілки не стала для вас несподіванкою.

- `source-dir` — керування каталогом, який kdesrc-build використовує для отримання початкового коду, запуску процедури збирання та збереження журналу. Типовим каталогом є `~/kdesrc`.
- `cmake-options` — параметр, який встановлює параметри, які слід передати програмі CMake під час збирання кожного з модулів. Типово, використовується для встановлення варіантів «debug» або «release», тобто вмикання (або вимикання) додаткових можливостей, або передавання даних щодо розташування потрібних бібліотек до процесу збирання.
- `make-options` — параметр, який визначає параметри, які використовуватимуться для запуску самої програми make для збирання кожного з модулів (щойно CMake визначить систему збирання).

Найтипівішим параметром є `-jN`, де *N* слід замінити на максимальну кількість завдань із компіляції, які ви хочете виконувати одночасно. Чим більшою буде кількість (аж до кількості логічних процесорів, які доступні у вашій системі), тим швидшою буде компіляція, але тим більшим буде навантаження на систему.

Example 2.1 Налаштування Make на компіляцію у 4 потоки з виключеннями

```
global
    make-options -j4
    &#8230;
end global

&#8230;

module-set big-module-set
    repository kde-projects
    use-modules calligra
    make-options -j2 # Зменшити кількість завдань зі збирання лише для ц ←
        их модулів
end module-set
```

ПРИМІТКА

У деяких великих сховищах Git ваша система може загрузнути у потоці багатьох паралельних завдань із збирання. Особливо цим відзначаються сховища Qt™ WebKit та Qt™ WebEngine. Щоб зберегти можливість працювати у системі, варто зменшити кількість завдань зі збирання для певних модулів. Приклад 2.1 є прикладом того, як це зробити.

Ймовірно, вам захочеться вибрати для збирання інші модулі. Спосіб визначення цих модулів описано у розділі Розділ 2.6.2.

2.3 Використання скрипту kdesrc-build

Після визначення даних налаштувань ви будете готові до запуску самого скрипту. Навіть якщо ви все ще хочете дещо налаштувати додатково або ознайомитися із додатковою довідковою інформацією, все ж варто одразу принаймні завантажити метадані проєктів KDE.

2.3.1 Завантаження метаданих проєкту

У вікні терміналу увійдіть до системи від імені користувача, якого ви використаєте для компіляції програмного забезпечення KDE, і виконайте команду:

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

Ця команда налаштує каталог з початковими кодами і з'єднає його зі сховищами Git KDE для отримання бази даних сховищ git та бази метаданих залежностей без компіляції чи внесення змін до коду. Корисно віддати цю команду окремо, оскільки ці метадані корисні для виконання інших команд kdesrc-build.

2.3.2 Попередній перегляд того, що трапиться, якщо запустити kdesrc-build

Якщо встановлено метадані проєкту, ви можете попередньо переглянути дії, які буде виконано kdesrc-build після запуску програми. Зробити це можна за допомогою параметра рядка команди --pretend.

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

Підручник зі скрипту kdesrc-build

Ви маєте побачити повідомлення щодо того, що певні пакунки успішно зібрано (хоча насправді нічого не було зібрано). Якщо не буде показано повідомлення про якісь значні проблеми, ви зможете перейти до безпосереднього запуску скрипту для збирання.

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

Ця команда призведе до отримання відповідного початкового коду та збирання і встановлення кожного модуля за порядком, але її виконання буде зупинено, якщо якийсь із модулів не вдасться зібрати (через використання параметра `--stop-on-failure`). Зрештою, ви маєте побачити виведені дані, подібні до даних у розділі Приклад 2.2:

Example 2.2 Приклад даних, виведених під час запуску 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 Усування проблем під час збирання

Залежно від кількості модулів, які ви вибрали для отримання, може так статися, що `kdesrc-build` не зможе з першого разу зібрати програмне забезпечення KDE. Але вам не слід впадати у відчай!

`kdesrc-build` записує до журналу дані, виведені всіма виконаними скриптом командами. Типово, файли журналу зберігаються у теці `~/kdesrc/log`. Щоб дізнатися про причину помилки для модуля у останній команді `kdesrc-build`, зазвичай, досить зазирнути до файлу `~/kdesrc/log/latest/ назва модуля /error.log`.

ПІДКАЗКА

Ймовірно, найшвидшим способом виявлення повідомлення про помилку, що призвела до неможливості збирання модуля, є пошук від кінця файла без врахування регістру слова `error`. Після виявлення такого рядка пошукайте інші повідомлення щодо помилок поряд з ним. На джерело проблем має вказати перше повідомлення про помилку у виявленій групі.

У цьому файлі ви знайдете повідомлення про помилку, яка призвела до невдалої спроби збирання модуля. Якщо у нижній частині цього файла ви бачите повідомлення про те, що не вдалося знайти певні пакунки, спробуйте встановити пакунки (зокрема відповідні пакунки `-dev`) перед повторним запуском `kdesrc-build`, і передайте `kdesrc-build` параметр `--reconfigure` після встановлення пакунків, яких не вистачає.

Або, якщо помилка є помилкою збирання (повідомлення про синтаксичні помилки, помилки щодо некоректного прототипу («`incorrect prototype`»), повідомлення про невідомий тип («`unknown type`») тощо), ймовірно, що помилку пов'язано з вадою вихідних кодів KDE, яку, як ми сподіваємося, буде усунуто протягом декількох днів. Якщо цю помилку не буде усунуто протягом цього часу, не вагайтеся і напишіть листа до списку листування kde-devel@kde.org (можливо, вам доведеться оформити підписку на цей список листування), щоб повідомити розробникам про невдалу спробу збирання.

Ви можете знайти приклади помилок і знайти способи їх усунування, а також загальні підказки та стратегії збирання програмного забезпечення KDE на сторінці [покрокових настанов зі збирання KDE 4](#).

З іншого боку, якщо все гаразд, у вашій системі встановлено нове KDE. Тепер його слід запустити. Опис потрібних для цього дій наведено далі, у розділі [Розділ 2.5](#).

ПРИМІТКА

Для отримання докладніших відомостей щодо можливостей журналювання `kdesrc-build`, будь ласка, зверніться до розділу [Розділ 3.2](#).

2.4 Збирання окремих модулів

Окрім повного збирання набору модулів, у вас може виникнути потреба у збиранні окремого модуля або невеликого набору модулів. Замість редагування вашого файла налаштувань, ви можете просто передати скрипту назви модулів або наборів модулів для збирання у рядку команди.

Example 2.3 Приклад даних, виведених kdesrc-build під час збирання окремого модуля

```
% 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
```

У нашому прикладі, хоча вказано лише програму *dolphin*, використання параметра `--include-dependencies` призведе до того, що `kdesrc-build` включить до процесу збирання залежності для *dolphin* (встановленням параметра `include-dependencies`).

ПРИМІТКА

Розв'язання залежностей у цьому випадку працює лише тому, що *dolphin* вказано у наборі модулів на основі `kde-projects` (у цьому прикладі він має назву `base-apps`). Див. Розділ [2.6.3.2](#).

2.5 Налаштовування середовища на запуск зібраного стільничного середовища Плазми KDE

Припустімо, що ви увійшли до системи від імені призначеного для збирання Плазми KDE користувача, і що у системі вже встановлено якусь версію KDE. В такому випадку запуск вашої копії Плазми може бути дещо ускладненим, оскільки зібрана Плазма має отримати пріоритет над старою версією. Вам слід змінити змінні середовища у ваших скриптах входу до середовища, щоб забезпечити запуск щойно зібраної версії середовища.

2.5.1 Автоматичне встановлення драйвера входу до системи

Починаючи з версії 1.16, kdesrc-build намагається встановити відповідний драйвер входу до системи, за допомогою якого ви зможете входити до вашого зібраного за допомогою kdesrc-build стільничного середовища KDE з загальної програми для керування сеансами. Вимкнути цей драйвер можна за допомогою параметра `install-session-driver` у файлі налаштувань.

ПРИМІТКА

У режимі імітації kdesrc-build налаштування сеансу не відбувається.

Робота драйвера забезпечується встановлення нетипового типу сеансу «`xsession`». Цей тип сеансу типово має працювати з програмою для керування сеансами `kdm` (де йому відповідатиме сеанс «`Custom`»), інші ж програми для керування сеансами (зокрема `LightDM` і `gdm`) можуть потребувати встановлення додаткових файлів для вмикання підтримки `xsession`.

2.5.1.1 Додавання підтримки xsession у дистрибутив

Для роботи з типовими програмами для керування сеансами деяких дистрибутивів може знадобитися встановлення додаткових пакунків з метою забезпечення підтримки входів у систему `xsession`.

- Для забезпечення підтримки нетипових входів `xsession` у дистрибутиві [Fedora Linux](#)[®] слід встановити пакунок `xorg-x11-xinit-session`.
- У [Debian](#) та заснованих на Debian дистрибутивах [Linux](#)[®] типово має бути передбачено підтримку нетипових входів `xsession`, але для того, щоб з такими входами можна було працювати, слід вказати параметр `allow-user-xsession` у `/etc/X11/Xsession.options`. Також варто ознайомитися з [документацією щодо налаштування сеансів графічного середовища Debian](#).
- Настанови щодо налаштування інших дистрибутивів можна знайти у розділі [Розділ 2.5.1.2](#).

2.5.1.2 Додавання підтримки xsession вручну

Якщо у розділі [Розділ 2.5.1.1](#) немає специфічних для вашого дистрибутива настанов, ви можете додати до списку сеансів вашого дистрибутива пункт «`Custom xsession login`» у такий спосіб:

ПРИМІТКА

Для виконання цих настанов вам, ймовірно, знадобляться права доступу адміністратора.

1. Створіть файл `/usr/share/xsessions/kdesrc-build.desktop`.
2. Додайте до створеного файла такий текстовий фрагмент:

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

- ❶** Запис `$HOME` слід замінити на повний шлях до вашого домашнього каталогу (наприклад `/home/користувач`). У специфікації запису стільниці не передбачено використання універсальних для всіх користувачів файлів.
3. Після перезапуску програми для керування сеансами у її меню має бути показано новий тип сеансу — «`KDE Plasma Desktop (unstable; kdesrc-build)`». У разі вибору цього пункту буде виконано спробу запуснути файл `.xsession`, встановленого kdesrc-build.

ПРИМІТКА

Якщо програма для керування сеансами не стежить за оновленням у каталозі `/usr/share/xsessions`, найпростішим способом її перезапуску є перезапуск комп'ютера.

2.5.2 Налаштовування середовища вручну

У цьому документі наведено настанови щодо тих змінних середовища, які слід встановити для завантаження щойно зібраного стільничного середовища. Ці настанови можна знайти у додатку (Розділ Б.1).

Якщо ви маєте намір власноруч налаштувати підтримку входу до системи, ви можете звернутися до відповідного додатка або переглянути файл `sample-kde-env-master.sh`, який є частиною початкових кодів `kdesrc-build`.

2.6 Впорядкування та вибір модулів

2.6.1 Структура модулів програмного забезпечення KDE

Програмне забезпечення KDE поділено на декілька компонентів, значну кількість яких можна зібрати за допомогою `kdesrc-build`. Знання зі структури компонентів допоможуть вам визначитися з модулями програмного забезпечення, які слід зібрати.

1. На найнижчому рівні структури перебуває бібліотека Qt™, яка дуже потужним набором інструментів, здатних працювати на багатьох програмних платформах. KDE засновано на Qt™, крім того, деякі з бібліотек, які не є частиною KDE, але потрібні для роботи KDE, також засновано на Qt™. `kdesrc-build` може зібрати Qt™ або скористатися вже зібраною бібліотекою, яку встановлено у системі, якщо ця бібліотека є достатньо новою.
2. Над Qt™ надбудовано бібліотеки, які потрібні для роботи програмного забезпечення KDE. Частина з цих бібліотек не є частиною самого KDE через свою загальну спрямованість, але вони є потрібними для роботи платформи KDE. Ці бібліотеки зібрано у єдиний модуль `kdesupport`, вони не вважаються частиною бібліотек «Frameworks».
3. Над цими основними бібліотеками надбудовано [KDE Frameworks](#), які іноді позначають аббревіатурою KF5. Це основні бібліотеки стільниці Плазми KDE, програм KDE та іншого стороннього програмного забезпечення.
4. На основі Frameworks побудовано декілька інших частин:
 - «Сторонні» програми. Ці програми використовують KDE Frameworks або створені для запуску у середовищі Плазми KDE, але вони не є частиною або взагалі не пов'язані з проектом KDE.
 - Плазма, яка є «робочим простором» стільничного середовища. Це те, що користувачі зазвичай бачать після «входу до KDE».
 - Комплект програм KDE. Це збірка корисного програмного забезпечення разом із платформою для розробки та стільницею Плазми, згрупована за окремими модулями, включно з допоміжними програмами, зокрема Dolphin, іграми, зокрема KSudoku, та офісним програмним забезпеченням KDE зокрема Contact.
 - Нарешті, існує збірка програмного забезпечення (також розподілена на модулі), підтримку розробки якого здійснює команда KDE (зокрема забезпечує переклад, обслуговування сховища коду, надання системи стеження за вадами тощо). Випуски цього програмного забезпечення не збігаються у часі з випусками KDE, воно не є частиною Плазми або основного комплекту програм. Ці модулі називають «Extragear».

2.6.2 Вибір модулів для збирання

Вибором переліку модулів, які ви бажаєте зібрати, можна керувати за допомогою [файла налаштування](#). Після розділу `global` у цьому файлі можна вказати список модулів, які слід зібрати, між рядком `module` та рядком `end module`. Приклад запису модуля наведено у розділі [Приклад 2.4](#).

Example 2.4 Приклад запису модуля у файлі налаштувань:

```
module kdesrc-build-git
    # Тут слід вказати параметри цього модуля. Приклад:
    repository kde:kdesrc-build
    make-options -j4 # Запустити компіляцію у 4 потоки
end module
```

ПРИМІТКА

На практиці, така конструкція модуля зазвичай не використовується безпосередньо. Замість цього, більшість модулів визначаються наборами модулів, як це описано нижче.

Якщо використано лише записи `module`, `kdesrc-build` збиратиме модулі у вказаному у списку порядку і не намагатиметься отримати дані будь-яких інших сховищ, які не було вказано явним чином.

2.6.3 Набори модулів

Початковий код KDE розділено між багатьма відносно малими сховищами на основі Git. Щоб цією великою кількістю сховищ, які є частиною будь-якого хоч трохи корисного комплексу програмного забезпечення KDE, було простіше керувати, у `kdesrc-build` передбачено підтримку групування декількох модулів і обробки групи як «набору модулів».

2.6.3.1 Концепція наборів базових модулів

За допомогою набору модулів ви можете оголосити декілька модулів Git, дані яких слід отримати і зібрати, подібно до окремих оголошень модулів. Для визначення кожної з адрес отримання даних модуля використовується параметр [repository](#), визначену за його допомогою адресу буде скопійовано до адрес всіх модулів з набору.

Example 2.5 Використання наборів модулів

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

module qt
    # Параметри вилучено для спрощення
end module

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

# Інші модулі, якщо потрібно...
module kdesupport
end module
```

У розділі Приклад 2.5 наведено короткий набір модулів. kdesrc-build обробляє цей набір модулів подібно до того, як виконується обробка кожного з модулів у `use-modules`, якщо кожен з модулів було оголошено окремо. Значення параметра `repository` буде визначено на основі значення параметра набору модулів `repository`, до якого буде додано вказану назву модуля.

Крім того, набору модулів можна передавати і інші параметри, які буде скопійовано до кожного з нових модулів, створених за його допомогою. За допомогою набору модулів можна скоротити роботу з оголошення багатьох модулів Git, які зберігаються за однією адресою. Також можна надавати наборам модулів назви (як це показано у прикладі). Назви можна використовувати для посилання на цілу групу модулів у командному рядку.

2.6.3.2 Особлива підтримка для наборів модулів KDE

Описана вище підтримка наборів модулів є загальною для будь-яких модулів на основі Git. Для сховищ git KDE у kdesrc-build передбачено додаткові можливості, якщо спрощують життя користувачів і розробників. Підтримка цих можливостей вмикається визначенням `kde-projects` як `repository` для набору модулів.

Зазвичай, kdesrc-build збирає лише модулі, які ви вказали у вашому файлі налаштувань, за порядком, у якому їх було вказано. Якщо ж визначено набір модулів `kde-projects`, kdesrc-build може виконувати розв'язування залежностей специфічних для KDE модулів і додатково автоматично включати модулі до збирання, навіть якщо їх не було безпосередньо вказано.

Example 2.6 Використання наборів модулів kde-projects

```
# Додає лише модуль для juk (сховище kde/kdemultimedia/juk)
module-set juk-set
    repository kde-projects
    use-modules juk
end module-set

# Додає усі модулі у kde/multimedia/*, зокрема juk,
# але без інших залежностей
module-set multimedia-set
    repository kde-projects
    use-modules kde/multimedia
end module-set

# Додає усі модулі у kde/multimedia/* та усі залежності kde-projects
# поза kde/kdemultimedia
module-set multimedia-deps-set
    repository kde-projects
    use-modules kde/multimedia
    include-dependencies true
end module-set

# Усі модулі, створені поза цими трьома наборами модулів буде
# автоматично упорядковано, незалежно від встановлення include- ←
dependencies
```

ПІДКАЗКА

Ця конструкція набору модулів `kde-projects` є основним способом оголошення про те, які модулі ви хочете зібрати.

В усіх наборах модулів використовуються параметри `repository` та `use-modules`. Набори модулів `kde-projects` мають попередньо визначенен значення `repository`, але інші типи наборів модулів також використовують параметр `git-repository-base`.

2.6.4 Офіційна база даних модулів KDE

Сховища git KDE забезпечують групування пов'язаних модулів git у збірки пов'язаних модулів (наприклад kdegraphics). Git не розпізнає цього групування, але kdesrc-build може визначити такі групи, використовуючи **набори модулів**, у яких значенням repository є «kde-projects».

kdesrc-build визначить, що сховище kde-projects потребує особливої обробки і скоригує процес збирання відповідним чином. Окрім виконання інших дій, kdesrc-build буде:

- Отримувати найсвіжішу базу даних модулів з [архіву git KDE](#).
- Намагатиметься знайти модуль з вказаною назвою у списку параметра набору модулів use-modules з цієї бази даних.
- Для кожного виявленого модуля kdesrc-build шукатиме відповідне сховище у базі даних із врахуванням параметра branch-group. Якщо сховище існує і є активним для групи гілок, kdesrc-build автоматично скористається ним для отримання і оновлення початкового коду.

ПРИМІТКА

У поточній базі даних деякі з записів груп модулів містять не лише збірки модулів, але і дані щодо власного сховища. У таких випадках поточна версія kdesrc-build намагатиметься скористатися сховищем Git групи, а не дочірніми сховищами записів групи.

Наведений нижче приклад демонструє використання бази даних модулів KDE для встановлення мультимедійної бібліотеки Phonon.

```
module-set media-support
# Значенням цього параметра має бути kde-projects, щоб використати б ←
азу даних модулів.
repository kde-projects

# За допомогою цього параметра можна визначити модулі, які слід шука ←
ти у базі даних.
use-modules phonon/phonon phonon-gstreamer phonon-vlc
end module-set
```

ПІДКАЗКА

phonon/phonon використовується, оскільки інакше, у разі використання поточної бази даних проекту, kdesrc-build доведеться визначитися між групою проектів з назвою «phonon» і окремим проектом з назвою «phonon». У поточній версії kdesrc-build вибере перший, що призведе до збирання багатьох модулів, які можуть виявитися непотрібними.

Наступний приклад є, можливо, реалістичнішим. Він демонструє можливість, доступну лише у разі використання бази даних модулів KDE: збирання всіх програм KDE з графічним інтерфейсом за допомогою єдиної команд.

```
module-set kdegraphics
# Значенням цього параметра має бути kde-projects, щоб можна було ск ←
ористатися базою даних модулів.
repository kde-projects

# За допомогою цього параметра можна вказати модулі, які слід шукати ←
у базі даних.
use-modules extragear
end module-set
```

Тут продемонстровано дві важливі можливості:

1. kdesrc-build надає вам змогу вказати дочірні модулі вказаного модуля, без збирання батьківського модуля за допомогою синтаксичної конструкції `назва-модуля /*`. У цьому випадку таке визначення є необхідним, оскільки основний модуль, `kdegraphics`, позначено як неактивний, щоб його не було випадково зібрано разом з дочірніми модулями. Визначення дочірніх модулів надає змогу kdesrc-build пропустити вимкнений модуль.
2. kdesrc-build також не додаватиме вказаний модуль до списку збирання декілька разів. Це надасть змогу нам вручну встановити збирання `kdegraphics/libs` першим, до решти `kdegraphics`, без спроби збирання `kdegraphics/libs` двічі. Так зроблено для належної обробки залежностей. Зараз цей спосіб визначення порядку збирання лишається резервним для випадків, коли у базі даних проєктів KDE пропущено відповідні метадані щодо залежностей.

2.6.5 Відфільтровування модулів проєкту KDE

Можливо, у вас виникне потреба у збиранні всіх програм у певному модулі KDE, *окрім* якоїсь програми.

Наприклад, у групі `kdeutils` міститься програма з назвою `kremotecomtrol`. Якщо на вашому комп'ютері не встановлено обладнання для роботи з пристроями дистанційного керування, ви можете вирішити, що отримувати, збирати і встановлювати `kremotecomtrol` під час кожного оновлення `kdeutils` не варто.

Досягти потрібного результату можна за допомогою параметра налаштування `ignore-modules`.

Example 2.7 Приклад ігнорування модуля `kde-project` у групі

```
module-set utils
  repository kde-projects

  # За допомогою цього параметра можна визначити модулі, які слід шукати ←
  #   ти у базі даних.
  use-modules kdeutils

  # За допомогою цього параметра можна «вилучити» модулі зі списку вик ←
  #   ористаних модулів, визначеного вище.
  ignore-modules kremotecomtrol

module-set graphics
  repository kde-projects

  # За допомогою цього параметра визначаються модулі, які слід шукати ←
  #   у базі даних.
  use-modules extragear/graphics

  # За допомогою цього параметра «вилучаються» модулі, вибрані вище за ←
  #   допомогою use-modules.
  # У нашому прикладі ігноруються одразу extragear/graphics/kipi- ←
  #   plugins i
  # extragear/graphics/kipi-plugins/kipi-plugins-docs
  ignore-modules extragear/graphics/kipi-plugins
end module-set
```

2.7 Початкові висновки

Це основні відомості та поняття, потрібні для того, щоб розпочати роботу з kdesrc-build

Підручник зі скрипту kdesrc-build

Додаткові відомості можна знайти у цьому підручнику. Зокрема, корисними будуть [список параметрів командного рядка](#) та [таблиця параметрів файлу налаштувань](#).

Також спільнота KDE постійно оновлює [інтернет-довідник щодо збирання з початкового коду](#), у якому використовується kdesrc-build і наводяться підказки та інші настанови щодо використання цього скрипту.

Розділ 3

Можливості скрипту

3.1 Огляд можливостей

Серед можливостей kdesrc-build:

- Ви можете вказати програмі, що слід виконати фальшиву послідовність операцій. Якщо ви передасте параметр `--pretend` або `-p` до рядка команди, скрипт надасть вам докладний опис команд, які потрібно виконати скрипту, без справжнього виконання команд. Втім, якщо ви ще ніколи не запускали kdesrc-build, вам варто спочатку віддати команду `kdesrc-build --metadata-only`, інакше `--pretend` не працюватиме як слід.

ПІДКАЗКА

Якщо вам потрібні ще докладніші пояснення щодо дій kdesrc-build, спробуйте скористатися параметром `--debug`.

- kdesrc-build може (за допомогою сервера FTP KDE) забезпечити пришвидшення отримання певних модулів Subversion. Якщо модуль, який ви отримуєте, вже було запаковано на вебсайті, kdesrc-build житить знімок і підготує його до використання на вашому комп'ютері.

ПІДКАЗКА

Загалом, для початкового отримання модуля Git не потрібно виконувати ніяких попередніх дій, оскільки за будь-яких умов доведеться отримувати весь вміст сховища Git. Тому серверу доволі просто визначити, що саме ви бажаєте отримати.

Це пришвидшить роботу і полегшить вам отримання з анонімних серверів Subversion kde.org.

- Додаткове пришвидшення можна отримати, якщо почати процес збирання модуля одразу після того, як вихідні коди цього модуля буде отримано. (Ця можливість доступна починаючи з версії 1.6)
- Чудова підтримка збирання бібліотеки Qt™ (якщо програмне забезпечення KDE, яке ви намагаєтеся зібрати, залежить від найсвіжшої версії Qt™, доступу до пакунків якої у вашому дистрибутиві ще не передбачено).
- kdesrc-build не потребує наявності графічного інтерфейсу для роботи. Отже, ви можете зібрати програмне забезпечення KDE без потреби у переходу до графічного середовища.
- Підтримка встановлення типових параметрів для всіх модулів (зокрема параметрів компіляції і налаштування). Крім того, такі параметри, зазвичай, можна змінити для окремих модулів.

Крім того, kdesrc-build **додасть стандартні прапорці**, потрібні для зберігання повідомлень про труднощі і помилки: вам не доведеться копіювати їх для пошуку у мережі власноруч. Варто зауважити, що це не стосується (нетипових) наборів інструментів, якщо вказано, наприклад, за допомогою `cmake-toolchain`

- kdesrc-build може отримувати певні **гілки або версії** модуля. Ви можете вказати для отримання навіть окрему **версію** певного модуля.
- kdesrc-build може автоматично перемикає каталог вихідних кодів для отримання з іншого сховища, гілки або випуску. Цю дію буде виконано автоматично, якщо ви зміните параметр, який вказує якою має бути адреса URL сховища, але вам доведеться скористатися параметром `--src-only`, щоб надати змогу kdesrc-build дізнатися про те, що скрипту можна виконувати таке перемикання.
- kdesrc-build може **отримувати лише частини модуля** у тих випадках, коли вам потрібна лише одна програма з великого модуля.
- Для розробників: kdesrc-build **нагадає вам** про те, що ви використовуєте `svn+ssh://`, але `ssh-agent` не запущено, оскільки таке використання призведе до повторних запитів на пароль від SSH.
- Можливість **вилучення каталогу збирання** модуля після його встановлення для економії місця за рахунок збільшення наступного часу збирання.
- Адреси каталогів, які використовуються kdesrc-build можна налаштувати (навіть для окремих модулів).
- Можливість використання Sudo, або іншої вказаної користувачем команди для **встановлення модулів**, призначеної для того, щоб kdesrc-build не потрібно було запускати від імені привілейованого користувача.
- kdesrc-build типово запускається зі **зниженими правами доступу**, це надає вам змогу використовувати kdesrc-build під час роботи з іншими програмами.
- Підтримка використання **випусків і гілок KDE**.
- Передбачено підтримку **поновлення збирання** з вказаного модуля. Ви навіть можете **ігнорувати певні модулі** тимчасово для вказаного сеансу збирання.
- kdesrc-build покаже **ступінь збирання** за використання CMake, і завжди стежитиме за часом процесу збирання, отже ви знатимете про тривалість збирання.
- Постається з вбудованим набором типових параметрів, щоб відповідають збиранню базового середовища KDE для одного користувача з анонімного сховища коду.
- Використання тильди у параметрах налаштування. Наприклад, ви можете вказати:

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

- Автоматичне встановлення системи збирання з каталогом вихідних кодів, який відрізняється від каталогу збирання, що забезпечить чистоту каталогу вихідних кодів.
- Ви можете вказати загальні параметри, які буде застосовано до всіх отриманих модулів, крім того, ви можете визначити параметри для окремих модулів.
- Примусове повне повторне збирання: для цього слід запустити kdesrc-build з параметром `--refresh-build`.
- Ви можете вказати значення різноманітних змінних середовища, які буде використано під час збирання, зокрема `KDEDIR`, `QTDIR`, `DO_NOT_COMPILE` і `CXXFLAGS`.
- Запис команд до журналу. Журнали датуватимуться і нумеруватимуться так, що у вас завжди буде журнал запуску скрипту. Крім того, буде створено символічне посилання з назвою `latest`, яке завжди вказуватиме на найсвіжіший запис журналу у каталозі журналу.
- Ви можете отримати лише частину модуля Subversion KDE. Наприклад, ви можете отримати лише `taglib` з `kdesupport`.

3.2 Запис журналів збирання kdesrc-build

3.2.1 Огляд ведення журналу

Ведення журналу — це можливість kdesrc-build, за використання якої дані, виведені будь-якої командою, яку запускає kdesrc-build, буде збережено до файла для наступного вивчення, якщо це буде потрібно. Так зроблено, тому що ці дані часто потрібні для визначення причин помилки під час збирання, оскільки існує багато причин, через які збирання може бути невдалим.

3.2.1.1 Компонування каталогів журналу

Журнали завжди зберігаються у каталозі журналів. Адреса каталогу журналів визначається параметром `log-dir`, типовим значенням якого є `source-dir /log` (де `source-dir` є значенням параметра `source-dir`. Надалі у цьому розділі ми посилатимемося на значення відповідного параметра як `log-dir`).

У каталозі `log-dir` буде створено набір каталогів, по одному на кожен запуск kdesrc-build. Кожен з каталогів матиме назву, що відбиватиме дату запуску скрипту і номер запуску. Наприклад, для другого запуску kdesrc-build 26 травня 2009 року буде створено каталог з назвою `2009-05-26-02`, де `2009-05-26` позначатиме дату а `-02` порядковий номер запуску.

Крім того, для зручності, kdesrc-build створить посилання на журнали останнього запуску з назвою `latest`. Отже, журнали для найостаннішого запуску kdesrc-build мають завжди зберігатися у теці `log-dir /latest`.

Крім того, кожен з каталогів запусків kdesrc-build сам містить набір каталогів, по одному на кожен модуль KDE, який kdesrc-build намагається зібрати. У цьому каталозі також міститиметься файл `build-status`, за допомогою якого ви зможете визначити, які модулі було зібрано, а які зібрати не вдалося.

ПРИМІТКА

Якщо у самому модулі міститься підмодуль (наприклад `extragear/multimedia`, `playground/utlils`, або `KDE/kdelibs`), йому відповідатиме власний підкаталог у каталозі журналу. Наприклад, журнали `KDE/kdelibs` після останнього запуску kdesrc-build можна буде знайти у `log-dir /latest/KDE/kdelibs`, а не у каталозі `log-dir /latest/kdelibs`.

У кожному з каталогів модулів ви знайдете набір файлів для кожної з дій, які виконуватиме kdesrc-build. Якщо kdesrc-build оновлює модуль, ви побачите файли з назвами на зразок `vn-so.log` (отримання модуля) або `svn-up.log` (якщо відбувалося оновлення модуля, який вже було отримано). Якщо було запущено скрипт `configure`, слід очікувати появи журналу `configure.log` у цьому каталозі.

Якщо сталася помилка, ви побачите пояснення її причину у одному з файлів. Щоб вам легше було знайти файл, у якому міститься помилка, kdesrc-build створить посилання з файла, що містить помилку (на зразок `build-1.log` на файл з назвою `error.log`).

Результатом всього цього буде те, що ви знатимете чому не вдалося зібрати модуль після останнього запуску kdesrc-build, для цього вам слід для початку знайти файл з назвою `log-dir /latest/ назва модуля /error.log`.

ПІДКАЗКА

Якщо файл `error.log` є порожнім (особливо після встановлення), ймовірно, жодних помилок не сталося. Деякі з інструментів, які використовуються системою збирання KDE, іноді помилково повідомляють про помилку, коли насправді помилок не сталося.

Крім того, деякі з команд уникатимуть переспрямування виведення даних kdesrc-build і за певних обставин обходять файл журналу (зазвичай, під час виконання першого отримання Subversion), в такому випадку помилки не можна буде знайти файл журналу, — всі помилки буде виведено до Konsole або програми терміналу, з якої ви запустили kdesrc-build.

Розділ 4

Налаштування kdesrc-build

4.1 Огляд налаштування kdesrc-build

Щоб скористатися цим скриптом, вам потрібен файл з назвою `.kdesrc-buildrc`, розташований у вашому домашньому каталозі, у цьому файлі описано модулі, які ви бажаєте отримати і зібрати, та усі параметри використання цих модулів.

4.1.1 Компонування файла налаштувань

4.1.1.1 Загальні налаштування

Файл налаштувань буде запущено з загальними параметрами, вказаними у такий спосіб:

```
global
назва_параметра значення_параметра
[...]
end global
```

4.1.1.2 Налаштування модулів

Після цього повинні йти один або декілька розділів модулів, вказаних у одному з двох форматів:

- ```
module назва_модуля
назва_параметра значення_параметра
[...]
end module
```

- ```
module-set назва-набору-модулів
  repository kde-projects або git://host.org/path/to/repo.git
  use-modules назви-модулів

# Можна встановити також інші параметри
назва-параметра значення-параметра
[...]
end module-set
```

ВАЖЛИВО

Зауважте, що другий формат, набори модулів, *працює лише для модулів, код яких зберігається у Git*.

Для модулів Subversion параметр *назва_модуля* має бути назвою модуля зі сховища Subversion KDE (наприклад kdeartwork або kde-wallpapers). Крім того, можна змінити значення вручну визначенням адреси Subversion.

Для модулів Git назвою може бути будь-який рядок, якщо цей рядок не дублює інших назв модулів у налаштуваннях. Пам'ятайте, що розташування каталогів коду і збирання буде визначено на основі назви модуля, якщо ви не використовуєте параметра `dest-dir`.

Але для *наборів модулів* Git *назви-модулів* мають відповідати справжнім модулям git modules у вибраному сховищі (параметр `repository`). Щоб дізнатися більше, ознайомтеся з довідкою щодо `git-repository-base` і `use-modules`.

4.1.1.3 Модулі «options»

Цей останній тип записів файлу налаштувань, групи `options`, яким можна скористатися для записів `module` або `module-set`.

```
options назва_модуля
назва_параметра значення_параметра
[...]
end options
```

У групі `options` можна встановлювати значення параметрів так само, як у оголошенні модуля. Цю групу має бути пов'язано з наявним модулем. Усі параметри, встановлені за допомогою цієї групи, буде використано для *перевизначення* набору параметрів пов'язаного модуля.

ВАЖЛИВО

Пов'язана назва модуля має збігатися з назвою, даною у оголошенні `options`. Тому слід не уважно стежити за тим, щоб у назві не було друкарських помилок.

Це корисно, якщо треба оголосити повний набір `module-set` для модулів з використанням однакових параметрів, а потім скористатися групами `options` для внесення окремих змін.

Групи `options` можна також застосовувати до іменованих наборів модулів. Ця можливість надає змогу досвідченим користувачам використовувати загальний файл налаштувань (який включено до оголошення `module-set`) як основу і вносити зміни до параметрів, використовуваних наборами модулів, у файлах налаштувань, використовуючи команду `include` для посилання на основні налаштування.

Example 4.1 Приклад використання options

У нашому прикладі визначено збирання усіх модулів з групи мультимедійного програмного забезпечення KDE. Крім того, визначено використання іншої версії для програми KMix (наприклад, щоб перевірити, чи виправлено певну ваду). Це працює так:

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

# kmix є частиною групи kde/kdemultimedia, навіть якщо kmix раніше у файлі ←
# не згадувався, kdesrc-build врахує зміну.
options kmix
  branch KDE/4.12
end options
```

Тепер, якщо ви запустите kdesrc-build, усі мультимедійні програми KDE буде зібрано на основі коду з гілки «master» сховища з кодом, KMix буде зібрано зі старішої гілки «KDE/4.12». Оскільки використано options, нам не довелося створювати окремий список усіх інших мультимедійних програм KDE, щоб визначити для них правильне значення параметра гілки коду.

ПРИМІТКА

Зауважте, що цю можливість реалізовано лише з версії kdesrc-build 1.16, а також у тестовій версії kdesrc-build для розробників з 12 січня 2014 року.

4.1.2 Включенні інших файлів налаштувань

У межах файла налаштувань ви можете посилатися на інші файли за допомогою ключового слова include з файлом. Все працюватиме так, наче вміст відповідного файла вставлено у місці включення.

Наприклад, ви можете вказати щось таке:

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

  # Тут вкажіть специфічні параметри.
end global
```

ПРИМІТКА

Якщо вами не буде вказано шлях до файла, який слід включити, повністю, програма шукатиме цей файл, починаючи з каталогу, щоб містить початковий файл. Пошук буде виконано рекурсивно.

4.1.3 Типові параметри налаштування

Далі наведено список параметрів, якими ви можете скористатися. Натисніть позначку параметра, щоб дізнатися про нього більше. Повний список параметрів наведено у розділі Розділ 4.2.

- `cmake-options`, щоб визначити параметри налаштування модуля за допомогою CMake.
- `branch`, щоб отримати код для стабільної версії, а не версії, що розробляється (`/trunk` для Subversion або `master` для Git).
- `configure-flags`, щоб визначити параметри, згідно яких буде налаштовано модуль Qt™.
- `kdedir`, щоб вказати каталог, куди буде встановлено KDE.
- `make-options`, щоб передати параметри програмі Make (зокрема кількість процесорів, які слід використовувати).
- `qtdir`, щоб вказати шлях до бібліотек Qt™.
- `source-dir`, щоб змінити каталог призначення для отримання початкових кодів.

4.2 Таблиця параметрів налаштування

Нижче наведено таблицю параметрів програми, у якій ви знайдете такі дані:

- Назву параметра
- Опис реакції `kdesrc-build` під час збирання модуля на встановлення значення параметра у загальному розділі та розділі модуля [файла налаштувань](#).
- Коментарі щодо призначення та способів використання параметра.

Назва параметра	Модуль -> Загальна поведінка	Нотатки
<code>apidox</code>		Цей параметр використовувався для вмикання збирання документації модуля програмного інтерфейсу KDE. Його було вилучено у версії <code>kdesrc-build 1.6.3</code> через брак підтримки. Ви можете знайти мережеву документацію з програмного інтерфейсу за адресою kde.org . Крім того, можна зібрати документацію з програмного інтерфейсу KDE за допомогою скрипту <code>kdedoxygen.sh</code> , включеного до модуля <code>kde-dev-scripts</code> . Докладніше про це можна дізнатися на сайті KDE TechBase .
<code>apply-qt-patches</code>		Цей параметр було вилучено з версії <code>kdesrc-build 1.10</code> . Щоб отримати відповідний йому результат, зверніться до Розділ 6.2.1 та параметра repository .

<p>async</p>	<p>Неможливо перевизначити</p>	<p>За допомогою цього параметра можна увімкнути асинхронний режим роботи, за якого оновлення кодів програм і процес збирання виконуватимуться паралельно, — програма не чекатиме на завершення оновлення всіх кодів програм перед запуском процесу збирання. Типовим значенням цього параметра є значення, що вмикає асинхронний режим. Щоб вимкнути асинхронний режим, встановіть значення параметра false. Цей параметр додано починаючи з версії 1.6 програми.</p>
<p>binpath</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Встановіть для цього параметра значення, щоб визначити значення змінної середовища PATH на час збирання. Ви не зможете перевизначити значення цього параметра для окремого модуля. Типовим значенням параметра є \$PATH, змінна, значення якої встановлюється під час запуску. Значенням цієї змінної середовища має бути відокремлений двокрапками список шляхів до ваших каталогів для збирання. Шляхи \$KDEDIR /bin і \$QTDIR /bin буде додано автоматично. Ви можете додати тильду (~) для всіх шляхів, які ви додаєте за допомогою цього параметра.</p>

<p>branch</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Встановіть значення для цього параметра, щоб програма отримала певну версію KDE, а не типову гілку <i>master</i> (для Git) або <i>trunk</i> (для Subversion), у якій розробляється KDE. Наприклад, щоб отримати версію KDE 4.6, вам слід встановити значення цього параметра <i>4.6</i>.аметра <i>3.4</i>. Якщо kdesrc-build не вдається належним чином отримати версію з цим параметром, ймовірно, вам доведеться вручну вказати адресу URL, з якої слід виконати отримання за допомогою параметра module-base-path або override-url.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>ПРИМІТКА Для більшості модулів KDE, ймовірно, варто користуватися замість цього параметра параметром branch-group, а цей параметр використовувати лише для створення виключень з правил.</p> </div>
---------------	--	--

<p>branch-group</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Для цього параметра слід встановити значення загальної групи, з якої слід вибирати модулі. Для підтримуваних типів модулів Git kdesrc-build визначити справжню гілку, яку слід автоматично використовувати на основі створених розробниками KDE правил (ці правила можна переглянути у підкаталозі kde-build-metadata каталогу початкового коду kdesrc-build). Після визначення гілки цю гілку буде використано так, наче ви вказали усі гілки власноруч за допомогою параметра branch. Цей параметр корисний, якщо ви просто хочете підтримувати актуальність певної гілки розробки, не бажаючи приділяти надмірну увагу усім змінам у назвах гілок. У поточній версії (станом на 11 серпня 2013 року) групами гілок є:</p> <ul style="list-style-type: none"> • stable-qt4 — група для стеження за виправленнями вад у заснованих на Qt™ 4 бібліотеках та програмах KDE. • latest-qt4 — група для стеження за розробкою і новими можливостями у заснованих на Qt™ 4 бібліотеках і програмах KDE. • kf5-qt5 — група для стеження за найсвіжішими течіями у розробці ще не випущених бібліотек KDE Frameworks 5, заснованих на Qt™ 5, робочими просторами Плазми 2 тощо. <p>Зауважте, що якщо ви <i>суборете</i> значення branch</p>
---------------------	--	--

<p>build-dir</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Скористайтеся цим параметром, щоб змінити каталог, у якому міститимуться вихідні коди для збирання. Існує три різних способи скористатися ним:</p> <ol style="list-style-type: none">1. Шлях відносно каталогу вихідних кодів з Subversion KDE (див. параметр source-dir). Це типовий спосіб. Цей режим буде вибрано, якщо ви введете назву каталогу, яка не починається з тильди (~) або похилої риски (/). Типовим значенням є <code>build</code>.2. Абсолютний шлях. Якщо ви вкажете шлях, назва якого починається з /, цей шлях буде використано безпосередньо. Наприклад, <code>/tmp/kde-obj-dir/</code>.3. Шлях відносно вашого домашнього каталогу. Якщо ви вкажете шлях, назва якого починатиметься з ~, буде використано шлях відносно вашого домашнього каталогу, подібно до використання тильди у командній оболонці. Наприклад, <code>~/build-dir</code> визначить каталог збирання <code>/home/user-name/build-dir</code>. <p>Як це не дивно, ви можете змінити значення цього параметра для кожного з модулів окремо.</p>
------------------	--	--

<p>build-when-unchanged</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>За допомогою цього параметра можна керувати тим, чи буде kdesrc-build завжди намагатися зібрати модуль, у якому не було оновлення початкових кодів.</p> <p>Якщо для build-when-unchanged буде встановлено значення true, kdesrc-build завжди виконуватиме збирання модуля, навіть якщо ніякого оновлення коду модуля не відбулося. Така поведінка є типовою, оскільки у разі її використання збільшується ймовірність належного збирання.</p> <p>Якщо ви встановите для build-when-unchanged значення false, kdesrc-build виконуватиме спроби збирання модуля, лише якщо у коді модуля відбулися зміни, або у інших випадках, коли повторне збирання є насправді необхідним. Подібна поведінка може зекономити вам час, особливо якщо ви запускаєте kdesrc-build щоденно або навіть частіше.</p> <div data-bbox="997 1507 1334 2003" style="border: 1px solid black; padding: 5px;"><p>ВАЖЛИВО</p><p>Цей параметр призначено лише для оптимізації збирання. Подібно до значної частини оптимізацій, його використання може позначитися на коректності збирання. Наприклад, зміни, внесені до модулів qt або kdelibs, можуть призвести до необхідності повторного збирання інших модулів, навіть якщо їх початкові коди не зазнали змін взагалі.</p></div>
-----------------------------	--	---

<p>checkout-only</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Встановіть значення для цього параметра, щоб отримати вихідні коди Subversion файл за файлом. Значенням має бути відокремлений пробілами список каталогів, які слід отримати. Хоча цей параметр може переотримувати загальний параметр, пам'ятайте, що встановлення цього параметра на загальному рівні не має ніякого сенсу. Зауважте, що цей параметр не впливає на модулі Git через спосіб роботи системи керування кодом Git. Приклад можна знайти у розділі Розділ 6.3.1.1.</p>
<p>cmake-generator</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>За допомогою цього параметра можна визначити, який засіб створення даних буде використано у поєднанні із CMake. У поточній версії передбачено підтримку як Ninja так і файлів Make Unix. Некоректні (непідтримувані) значення буде проігноровано — значення вважатиметься невстановленим. Якщо не встановлено, типово буде використано файли Make Unix. Зауважте, що якщо коректний засіб обробки також вказано за допомогою cmake-options, він матиме пріоритет над цим значенням для cmake-generator.</p>

<p>cmake-toolchain</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Скористайтеся цим параметром, щоб вказати файл набору інструментів, яким слід скористатися у CMake.</p> <p>Якщо налаштовано коректний набір інструментів, kdesrc-build <i>не встановлюватиме змінні середовища автоматично</i>. Для виправлення середовища вручну ви можете скористатися set-env, binpath та libpath, якщо файл набору інструментів не працює у kdesrc-build без вашого втручання. Щоб дізнатися про це більше, зверніться до огляду стандартних прапорців, які додаються kdesrc-build.</p> <p>Зауважте, що якщо коректний набір інструментів також вказано за допомогою cmake-options, він матиме пріоритет над цим значенням для <code>cmake-toolchain</code>.</p>
------------------------	--	---

<p>smake-options</p>	<p>Дописує дані до загальних параметрів для типової системи збирання і перевизначає загальні параметри для інших систем збирання.</p>	<p>Скористайтеся цим параметром для того, щоб визначити, які параметри слід передати CMake під час створення системи збирання модуля. Якщо цей параметр використано як загальний, його буде застосовано до всіх модулів, які збирає цей скрипт. Якщо параметр визначено на рівні модуля, його буде додано в кінець загальний параметрів. За допомогою цього параметра можна вказати загальні параметри CMake у загальному розділі (global). Значення цього параметра не стосується qt (CMake там не використовується). Скористайтеся для збирання qt configure-flags. Якщо серед параметрів цього списку буде вказано коректний засіб створення даних, він матиме пріоритет над значенням smake-generator. Некоректні (або невідтримувані) значення запису засобу створення даних буде проігноровано — їх не буде передано CMake. Якщо серед параметрів цього списку буде вказано коректний файл набору інструментів, він матиме пріоритет над значенням smake-toolchain. Некоректні (або невідтримувані) значення запису набору інструментів буде проігноровано — їх не буде передано CMake. Оскільки ці параметри передаються безпосередньо до командного рядка CMake, їх слід вказувати у тому самому вигляді, у якому їх буде передано CMake. Приклад:</p>
----------------------	---	--

<p>colorful-output</p>	<p>Неможливо перевизначити</p>	<p>Встановіть для цього параметра значення <code>false</code>, щоб вимкнути розфарбування даних, які виводитиме kdesrc-build. Типовим значенням цього параметра є <code>true</code>. Зауважте, що kdesrc-build не виводитиме розфарбовані дані до інших, відмінних від терміналу (на зразок xterm, Konsole або звичайної консолі Linux[®]), виводів.</p>
<p>configure-flags</p>	<p>Дописує дані до загальних параметрів для типової системи збирання і перевизначає загальні параметри для інших систем збирання.</p>	<p>Скористайтеся цим параметром, щоб вказати параметри, які слід передати <code>./configure</code> під час створення системи збирання модуля. Якщо цей параметр використовується на загальному рівні, його буде застосовано до всіх модулів, які збиратимуться цим скриптом. <i>Цей параметр потрібен лише для qt-sourc..</i> Щоб змінити параметри налаштування модулів KDE, ознайомтеся з розділом щодо <code>cmake-options</code>.</p>

<p>custom-build-command</p>	<p>Параметр модуля перевизначає загальний параметр (параметри системи збирання)</p>	<p>За допомогою цієї команди можна наказати системі виконати іншу команду (відмінну від, наприклад, make) з метою збирання. Зазвичай, програма kdesrc-build здатна сама вибрати команду для збирання, отже немає потреби у визначенні цього параметра. Втім, він може бути корисним у разі використання альтернативних систем збирання. Значення цього параметра використовується як команда, яку слід виконати. Зміни до команди вносяться за допомогою параметра make-options у звичайному режимі.</p>
<p>sxxflags</p>	<p>Дописує дані до загальних параметрів для типової системи збирання і перевизначає загальні параметри для інших систем збирання.</p>	<p>Скористайтеся цим параметром, щоб вказати, які параметри слід використовувати для збирання модуля. Цей параметр визначається саме тут, а не у параметрах configure-flags і cmake-options, оскільки цей параметр також встановлює змінну середовища CXXFLAGS під час процесу збирання. Зауважте, що у KDE 4 і всіх інших модулів, які використовують CMake, потрібно встановити значення параметра CMAKE_BUILD_TYPE рівним none під час налаштування модуля. Зробити це можна за допомогою параметра cmake-options.</p>

<p>dest-dir</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Скористайтеся цим параметром, щоб змінити назву, яку буде дано модулю на диску. Наприклад, якщо вам потрібен модуль extragear/network, за допомогою цього параметра ви можете перейменувати його на extragear-network. Зауважте, що не варто включати до назви модуля каталоги роздільники каталогів, оскільки такі назви можуть призвести до конфлікту з одним з параметрів build-dir або source-dir.</p>
<p>disable-agent-check</p>	<p>Неможливо перевизначити</p>	<p>Зазвичай, якщо ви використовуєте SSH для отримання вихідних кодів Subversion (наприклад, якщо ви використовуєте протокол svn+ssh), kdesrc-build намагатиметься перевірити і зробити так, щоб, якщо ви використовуєте агент ssh, цей агент справді керував певними профілями SSH. Так зроблено, щоб SSH не надсилав, за можливості, запитів на пароль для кожного з ваших модулів. Ви можете вимкнути таку перевірку встановленням параметра <code>disable-agent-check</code> у значення <code>true</code>.</p>

do-not-compile	Параметр модуля перевизначає загальний параметр	Скористайтеся цим параметром, щоб вибрати певний набір каталогів, які не буде зібрано у модулі (замість збирання всіх каталогів). Відокремлювати каталоги у списку слід за допомогою пробілів. Зауважте, що вихідні коди програм все ж буде отримано. Ви можете скористатися параметром checkout-only , щоб вказати список тек, які ви бажаєте отримати. Наприклад, щоб вимкнути збирання JuK і KsCD у модулі kdemultimedia, вам слід додати рядок "do-not-compile juk kscd" до параметрів kdemultimedia. Приклад можна знайти у розділі Розділ 6.3.1.2 .
email-address	Неможливо перевизначити	Цей параметр було вилучено у версії kdesrc-build 1.14.
email-on-compile-error	Неможливо перевизначити	Цей параметр було вилучено у версії kdesrc-build 1.14.
inst-apps		Цей параметр було вилучено у версії 1.10

git-desired-protocol

Неможливо перевизначити

Цей параметр стосується модулів зі сховища [проєкту KDE](#).

Насправді, за допомогою цього параметра можна вказати, якому з протоколів роботи у мережі слід надавати перевагу під час оновлення початкових кодів відповідних модулів.

Зазвичай, використовується дуже ефективний протокол `git`, але його може бути заблоковано у деяких мережах (наприклад у корпоративних мережах, відкритих мережах Wi-Fi). Альтернативним протоколом з ширшою підтримкою є `https`, який використовується для доступу до сайтів інтернету.

Якщо ви користуєтеся якоюсь з мереж з обмеженнями, ви можете встановити для цього параметра значення `http`, щоб наказати програмі надавати перевагу передаванню даних за допомогою `https`.

ПІДКАЗКА

Крім того, у вас може виникнути потреба у встановленні значення для параметра `http-proxy`, якщо для роботи з даним використовується проксі-сервер HTTP.

У всіх інших випадках не варто встановлювати цей параметр, оскільки типовий протокол є найефективнішим. Цей параметр було додано у `kdesrc-build 1.16`. До версії 20.06 цей параметр використовувався для налаштування адреси отримання даних, а не адреси надсилання даних. З версії 20.06 для

<p>git-repository-base</p>	<p>Неможливо перевизначити</p>	<p>Цей параметр, доданий у версії 1.12.1, використовується для скорочення посилань на базову адресу певного сховища Git у наступних оголошеннях наборів модулів. Параметр корисний для оголошення збирання декількох модулів Git. Вам слід вказати два значення (відокремлених пробілами): назву, яку буде пов'язано з базовою адресою, та саму базову адресу. Приклад:</p> <pre> global # 1110;1085;1096;1110; 1087;10 # 1047;1072;1075;1072;1083;110 git-repository-base kde-git kde: end global # 1054;1075;1086;1083;1086;1096; module-set # 1058;1077;1087;1077;1088; 10 # 1091; 1084;1077;1078;1072;10 repository kde-git use-modules 1084;1086;1076;1091; end module-set</pre> <p>Параметр use-modules набору модулів створив два записи модулів.</p> <p>kdesrc-build поводитиметься так, наче було вказано:</p> <pre> module 1084;1086;1076;1091;1083;1 repository kde:1084;1086;1076;109 end module module 1084;1086;1076;1091;1083;1 repository kde:1084;1086;1076;109 end module</pre> <p>Префікс сховища Git kde:, використаний вище, є скороченням, яке буде автоматично встановлено kdesrc-build. Щоб дізнатися більше, ознайомтеся зі статтею TechBase щодо</p>
----------------------------	--------------------------------	--

<p>git-user</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Цей параметр призначено для розробників KDE. Якщо його встановлено, значення буде використано для автоматичного налаштування даних профілю системи керування версіями Git для <i>нових отриманих</i> модулів Git (більшості модулів KDE). Зокрема, вказаними за допомогою цього параметра значеннями буде заповнено поля імені користувача та адреси електронної пошти для кожного нового сховища Git. Значення має бути вказано у форматі Ім'я користувача <адреса @електронної.пошти>. Наприклад, якщо розробника звати «Bozna Khto», а адресою є «bozna@abc.xyz», слід вказати таке:</p> <pre>git-user Bozna Khto <bozna@abc.xyz></pre> <p>Цей параметр було додано у kdesrc-build 15.09.</p>
-----------------	--	---

<p>http-proxy</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Цей параметр, якщо його встановлено, використовується для визначення адреси проксі-сервера, яким слід користуватися для обміну даними у мережі за допомогою HTTP (наприклад під час отримання знімків нових модулів або баз даних проєктів KDE). Крім того, kdesrc-build намагатиметься використати проксі-сервер для роботи інших програм, від дій яких залежить працездатність програми. Отже, для змінної середовища <code>http_proxy</code> буде встановлене значення позначеного сервера, <i>якщо для цієї змінної ще не встановлено жодного значення</i>. Цей параметр було додано у kdesrc-build 1.16.</p>
<p>ignore-kde-structure</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>За допомогою цього параметра можна зберегти недоторканими початковий код і зібрати файли безпосередньо у модулі. Наприклад, замість <code>source/extragear/network/telepathy/ktp-text-ui</code> буде використано <code>source/ktp-text-ui</code>. Типово цей параметр вимкнено. Якщо вам потрібно увімкнути цей параметр, встановіть для нього значення <code>true</code>. Цей параметр було додано у kdesrc-build 1.16.</p>

<p>ignore-modules</p>	<p>Неможливо перевизначити</p>	<p>Модулі, названі у цьому параметрі, які мало б бути вибрано kdesrc-build через визначення параметра <code>use-modules</code>, буде повністю пропущено. Цим параметром можна скористатися, якщо ви хочете зібрати всі групи у проєкті <code>kde-projects</code>, <i>окрім</i> певних модулів. Значення параметра не повинне обов'язково вказувати безпосередньо на назву модуля. Будь-який з модулів, повна частина шляху до модуля у проєктах KDE якого збігається з одним зі значень параметра, буде проігноровано. Отже, таким чином можна ігнорувати одразу декілька модулів. Наприклад, значення параметра <code>libs</code> призведе до того, що буде виключено <code>kde/kdegraphics/libs</code> і <code>playground/libs</code> (але не <code>kde/kdelibs</code>, оскільки «kdelibs» не збігається повністю зі значенням параметра).</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>ПІДКАЗКА Див. також розділ Приклад 2.7.</p> </div> <p>Цей параметр було додано у kdesrc-build 1.16.</p>
-----------------------	--------------------------------	---

<p>include-dependencies</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>За допомогою цього параметра, якщо встановлено значення true, можна наказати kdesrc-build включити відомі залежності модуля до його налаштувань збирання, без потреби у явному визначенні цих залежностей (навіть опосередковано).</p> <div data-bbox="997 684 1341 961" style="border: 1px solid black; padding: 5px;"> <p>ПРИМІТКА Цей параметр працюватиме лише для модулів на основі kde-project і потребує від розробників KDE точного визначення метаданих для вибраної вами групи гілок.</p> </div> <p>Типово, цей параметр увімкнено для забезпечення підтримки збирання програм, які потребують версій Qt™ або Плазми, які є новішими за ті, підтримку яких передбачено у типових операційних системах.</p>
<p>install-after-build</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Цей параметр використовується для встановлення пакунка після його успішного збирання. Типово, цей параметр увімкнено. Якщо ви бажаєте вимкнути його, вам слід встановити для нього значення false у файлі налаштувань. Крім того, ви можете скористатися параметром командного рядка --no-install.</p>

install-environment-driver

Неможливо перевизначити

Типово, kdesrc-build встановити скрипт оболонки, яким можна скористатися у скриптах налаштування профілю користувача для спрощення визначення бажаних змінних середовища для запуску стільниці Плазми, зібраної за допомогою kdesrc-build. Цим драйвером буде внесено зміни до таких файлів:

- `$XDG_CONFIG_HOME/kde-env-master.sh` (зазвичай зберігається за адресою `~/.config/kde-env-master.sh`).
- `$XDG_CONFIG_HOME/kde-env-user.sh` (зазвичай зберігається за адресою `~/.config/kde-env-user.sh`).

Скрипт `kde-env-user.sh` є необов'язковим. Його призначено для налаштування користувачем (див. розділ [Troubleshooting and Debugging](#) сторінки UserBase KDE, де наведено приклади параметрів налаштування), але ці параметри можна встановити у інший спосіб користувачем у скриптах налаштування наявного профілю.

Ви можете вимкнути драйвер встановленням для цього параметра значення `false` і та вимиканням параметра [install-session-driver](#).

Цей параметр було додано у kdesrc-build 17.08.

ПІДКАЗКА

kdesrc-build не перезаписуватиме вже створених файлів (якщо такі існують), якщо ви не передасте програмі параметр командного

<p>install-session-driver</p>	<p>Неможливо перевизначити</p>	<p>Якщо увімкнено, kdesrc-build намагається встановити драйвер до графічної програми для керування сеансами для того, щоб уможливити ваш вхід до зібраного за допомогою kdesrc-build стільничного середовища KDE.</p> <p>Цим драйвером буде внесено зміни до таких файлів:</p> <ul style="list-style-type: none"> • <code>~/.xsession</code> • <code>\$XDG_CONFIG_HOME/kde-environment-master.sh</code> (зазвичай зберігається за адресою <code>~/.config/kde-environment-master.sh</code>). • <code>\$XDG_CONFIG_HOME/kde-environment-user.sh</code> (зазвичай зберігається за адресою <code>~/.config/kde-environment-user.sh</code>). <p>Якщо ви користуєтеся власним драйвером для входу до середовища, ви можете вимкнути драйвер встановленням для цього параметра значення <code>false</code>. Якщо її увімкнено, ця можливість також вмикає можливість install-environment-driver. Цей параметр було додано у kdesrc-build 1.16.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>ПІДКАЗКА kdesrc-build не перезаписуватиме вже створених файлів (якщо такі існують), якщо ви не передасте програмі параметр командного рядка <code>--delete-my-settings</code>.</p> </div>
-------------------------------	--------------------------------	--

kdedir	Параметр модуля перевизначає загальний параметр	За допомогою цього параметра можна вказати каталог, до якого буде встановлено KDE після збирання. Типовим значення параметра є <code>~/kde</code> . Якщо ви зміните значення так, що воно вказуватиме на каталог, для доступу до якого потрібні права користувача <code>root</code> , ймовірно, вам слід прочитати і про параметр make-install-prefix .
--------	---	---

<p>kde-languages</p>	<p>Неможливо перевизначити</p>	<p>За допомогою цього параметра ви можете вказати скрипту, що слід отримати і встановити пакунки локалізації разом з KDE. Це слід зробити, якщо ви не є мешканцем США і бажаєте, щоб скористатися KDE вашою рідною мовою. Щоб скористатися цим параметром, вам слід вказати для нього значення-список мов, які слід встановити, коди мов слід відокремлювати пробілами. З кожною з мов пов'язано певний код, знайти потрібний код можна на сторінці http://l10n.kde.org/teams-list.php. Можна вибрати і лише одну мову. Типово, не буде отримано жодної мови, що означає, що інтерфейс і довідку KDE буде показано лише американською англійською. Наприклад, якщо вам потрібно встановити пакунок української мови, вам слід встановити значення цього параметра десь так: <code>kde-languages uk</code>. Крім того, ймовірно, для того, щоб середовище використовувало українську, вам слід належним чином налаштувати його за допомогою Центру керування (KDE 3) або Системних параметрів (KDE 4).</p>
----------------------	--------------------------------	--

libpath	Параметр модуля перевизначає загальний параметр	Встановіть значення для цього параметра значення, щоб змінити значення змінної середовища <code>LD_LIBRARY_PATH</code> на час збирання. Ви не зможете перевизначити цей параметр на рівні модулів. Типовим є порожнє значення, але шляхи <code>\$KDEDIR /lib</code> і <code>\$QTDIR /lib</code> буде додано автоматично. Ви можете скористатися тильдою (<code>~</code>) у будь-якому зі шляхів, які ви додасте за допомогою цього параметра.
log-dir	Параметр модуля перевизначає загальний параметр	Скористайтеся цим параметром, щоб змінити каталог, у якому зберігатимуться файли журналу, створені скриптом.
make-install-prefix	Параметр модуля перевизначає загальний параметр	Встановіть для цієї змінної значення відокремленого пробілами списку, який скрипт вважатиме командою і її параметрами, які передують команді <code>make install</code> , призначеній для встановлення модулів. Наприклад, цей параметр буде корисним для встановлення пакунків за допомогою <code>Sudo</code> , але, будь ласка, будьте обережними з використанням прав доступу адміністративного користувача.

<p>make-options</p>	<p>Параметр модуля перевизначає загальний параметр (параметри системи збирання)</p>	<p>Встановіть значення для цієї змінної, щоб передати параметри командного рядка команді make. Подібні параметри будуть корисним для програм на зразок distcc або у системах з декількома процесорними ядрами. Зауважте, що make використовується не в усіх системах збирання. Якщо для збирання використовується ninja (так, зокрема, побудовано систему збирання Meson), скористайтеся параметром ninja-options.</p>
<p>manual-build</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Встановіть для цього параметра значення true, щоб під час процесу збирання скрипт не намагався зібрати цей модуль. Модуль все ж буде оновлено під час оновлення зі сховища Subversion. Цей параметр у точності збігається з параметром командного рядка --no-build.</p>
<p>manual-update</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Встановіть для цього параметра значення true, щоб процес збирання не намагався оновити (і отже зібрати і встановити) цей модуль. Якщо ви встановите цей параметр для певного модуля, вам варто закоментувати його.</p>

<p>module-base-path</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Встановіть цей параметр, щоб перевизначити шлях до типової теки kdesrc-build для потрібного модуля. Цим параметром можна, наприклад, скористатися, якщо потрібно використати певні гілки або стабільні версії бібліотек. У визначення правильного шляху неоціненну поміч вам може надати Переглядач кодів KDE. Зауважте, що kdesrc-build конструює остаточний шлях відповідно до такого шаблону: <code>\$svn-server /home/kde/ \$module-base-path</code>. Типовим значенням є або <code>trunk/ \$module</code>, або <code>trunk/KDE/ \$module</code>, залежно від назви модуля.</p> <div data-bbox="997 1129 1338 1310" style="border: 1px solid black; padding: 5px;"> <p>ПІДКАЗКА Ви можете скористатися варіантами <code>branch</code> або <code>tag</code>, якщо їх можна застосовувати.</p> </div>
<p>niceness</p>	<p>Неможливо перевизначити</p>	<p>Значенням цього параметра є ціле число у діапазоні від 20 до 0. Чим вищим буде це число, тим нижчим встановлюватиметься пріоритет процесу kdesrc-build, тобто, чим вищим є число, тим вищим є «пріоритет» програми. Типовим значенням є 10.</p>

<p>ninja-options</p>	<p>Параметр модуля перевизначає загальний параметр (параметри системи збирання)</p>	<p>Встановіть значення для цієї змінної, щоб передати параметри командного рядка для команди збирання ninja. За допомогою цього значення можна увімкнути «докладний» режим виведення даних або вручну зменшити кількість паралельних завдань зі збирання для ninja.</p> <div style="border: 1px solid black; padding: 5px;"> <p>ПРИМІТКА Зауважте, що значення цього параметра використовується для керування ninja лише у kdesrc-build. Модуль «webengine» Qt™ використовує ninja опосередковано, а офіційно збирається лише за допомогою make. У таких випадках ви можете, скориставшись make-options, встановити значення змінної NINJAFLAGS, щоб make передала відповідні прапорці під час наступного виклику ninja.</p> <pre>options qtwebengine # 1054;1073;1084;1077;1078;1085;1072;1074;1110;1090;1111 make-options -j6 NINJAFLAGS=-j6 end options</pre> </div>
<p>no-svn</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Якщо для цього параметра встановлення значення «true», kdesrc-build не оновлюватиме код модуля автоматично. Скрипт все ж намагатиметься зібрати цей модуль, якщо згідно налаштувань його слід спробувати зібрати.</p>
<p>no-rebuild-on-fail</p>		<p>Цей параметр було вилучено у версії 1.10, оскільки подібна поведінка вже не допомагає, оскільки виправлено основну систему збирання.</p>

<p>override-build-system</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Цей параметр для досвідчених користувачів було додано у kdesrc-build 1.16.</p> <p>Зазвичай, kdesrc-build автоматично визначає систему збирання, яку слід використати для модуля, після отримання його коду. Визначення виконується за допомогою пошуку специфічних файлів у каталозі з кодами модуля.</p> <p>Деякі модулі можуть містити більше за один набір файлів, що може призвести до помилок під час автоматичного визначення. У таких випадках слід вказати тип системи збирання вручну. Серед типів систем збирання, передбачених у поточній версії, слід відзначити такі:</p> <p>KDE</p> <p>Використовується для збирання модулів KDE. Насправді, можна використовувати для збирання майже будь-яких модулів, де використовується CMake, але краще на це не покладатися.</p> <p>Qt</p> <p>Призначено для збирання самої бібліотеки Qt™.</p> <p>qmake</p> <p>Використовується для збирання модулів Qt™, у яких використано файли qmake, .pro.</p> <p>generic</p> <p>Використовується</p>
------------------------------	--	---

<p>override-url</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Якщо встановите цей параметр, kdesrc-build використовуватиме його значення як адресу URL, яку слід передати Subversion <i>повністю незмінною</i>. Загалом, вам слід використовувати цей параметр, якщо ви бажаєте отримати певний випуск, але kdesrc-build не може визначити, який випуск ви мали на увазі у параметрі branch.</p>
<p>persistent-data-file</p>	<p>Неможливо перевизначити</p>	<p>За допомогою цього параметра можна змінити місце, у якому зберігатимуться постійні дані kdesrc-build. Типово, ці дані зберігатимуться у файлі з назвою <code>.kdesrc-build-data</code>, який зберігатимуться у тому самому каталозі, що і файл налаштувань. Якщо ви користуєтеся декількома наборами налаштувань, вам варто встановити значення цього параметра вручну, щоб різні набори налаштувань не призводили до конфлікту між наборами постійних даних. Цей параметр було додано у kdesrc-build 1.15.</p>

<p>prefix</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>За допомогою цього параметра можна вказати місце, куди слід встановити модуль (зазвичай, буде використано значення параметра <code>kdedir</code>). Цей параметр надає вам змогу встановити модуль у каталог, що не збігається з каталогом бібліотек Платформи KDE, якщо ви, наприклад, використовуєте <code>kdesrc-build</code> лише для збирання певних програм. Ви можете замінити назву модуля у шляхах за допомогою синтаксичної конструкції <code>#{MODULE}</code> або <code>\$MODULE</code>.</p>
<p>purge-old-logs</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>За допомогою цього параметра можна наказати скрипту автоматично вилучати старі каталоги журналів або не робити цього. Типовим значенням є <code>true</code> (вилучати).</p>
<p>qmake-options</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Усі вказані тут параметри буде передано програмі qmake для модулів, у яких використовується система збирання <code>qmake</code>. Наприклад, ви можете скористатися параметром <code>qmake PREFIX=/шлях/до/qt</code>, для перевизначення місця, куди буде встановлено модуль. Цей параметр було додано у версії <code>kdesrc-build 1.16</code>.</p>
<p>qtdir</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Вкажіть значення для цього параметра, щоб визначити змінну середовища <code>QTDIR</code> на час збирання. Якщо ви не вкажете значення для параметра, <code>kdesrc-build</code> вважатиме, що слід використовувати бібліотеки Qt™ з операційної системи.</p>

<p>remove-after-install</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Якщо на жорсткому диску вашого комп'ютера мало місця, ймовірно, вам слід скористатися цим параметром для того, щоб автоматично вилучити каталог збирання (або одночасно каталоги кодів зі збирання для одноразового встановлення) після того, як модуль буде успішно встановлено.</p> <p>Можливими значення цього параметра є:</p> <ul style="list-style-type: none"> • none — Нічого не вилучати (Типове значення). • builddir — Вилучити каталог збирання, але не початкових кодів. • all — Вилучити каталоги початкових кодів і збирання. <p>Зауважте, що використання цього параметра призведе до збільшення використання вашого каналу з'єднання (якщо ви використовуєте <i>all</i>) і часу, який буде використано для збирання програмного забезпечення KDE, оскільки kdesrc-build не зможе виконувати покрокове збирання.</p>
<p>repository</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Цей параметр було додано у версії 1.10. За його допомогою можна вказати сховище Git, звідки слід отримувати коди модуля. Цей параметр потрібен для Qt™ (і отже qt), а також різноманітних модулів KDE, які перебувають у стані переходу на використання Git.</p>

<p>revision</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Якщо встановити для цього параметра значення, відмінне від 0 (нуля), kdesrc-build примусово виконає оновлення початкового коду до точно вказаної версії модуля, навіть якщо діють параметри на зразок <code>branch</code>. Якщо ви вже маєте модуль вказаної версії, його не буде оновлено до того часу, доки цей параметр налаштування не буде змінено або вилучено.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>ПРИМІТКА Цей параметр не працював для модулів з кодом у git (зокрема модулів <code>kde-projects</code>) до версії kdesrc-build 1.16.</p> </div>
<p>run-tests</p>	<p>Параметр модуля перевизначає загальний параметр (параметри системи збирання)</p>	<p>Якщо значенням цього параметра буде <code>true</code>, модуль буде зібрано з підтримкою запуску комплексу тестів, комплекс тестів буде виконано у процесі збирання. kdesrc-build покаже вам спрощений звіт щодо результатів тестування. Цей параметр буде корисним для розробників або тих, хто бажає переконатися, що систему налаштовано належним чином.</p>

<p>set-env</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Цей параметр приймає розділений пробілами набір значень, де перше значення є змінною середовища, яку слід встановити, а решта значень вважатиметься значенням змінної. Наприклад, щоб встановити для змінної RONALD значення McDonald, вам слід вказати у відповідному розділі таку команду:</p> <pre>set-env RONALD McDonald</pre> <p>Цей параметр має особливість: його можна повторювати без перевизначення раніше встановлених параметрів set-env у одному і тому ж розділі файла налаштувань. Таким чином, ви можете встановити декілька змінних середовища у одному модулі (або загальних змінних).</p>
<p>source-dir</p>	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Цей параметр використовується для встановлення каталогу, у якому Subversion слід зберігати коди KDE. Якщо ви не вкажете це значення, його типовим значенням буде <code>~/kdesrc</code>. Ви можете скористатися тильдою (<code>~</code>) для позначення домашнього каталогу.</p>

ssh-identity-file	Неможливо перевизначити	<p>Встановіть значення цього параметра, якщо потрібно вказати, який саме файл ключа SSH слід передати програмі ssh-add під час отримання за допомогою kdesrc-build коду зі сховищ, для доступу до яких потрібно пройти розпізнавання. Також варто ознайомитися з розділом Розділ 6.4.1. Цей параметр було додано у версії 1.14.2.</p>
stop-on-failure	Параметр модуля перевизначає загальний параметр	<p>Встановіть для цього параметра значення true, щоб зупинити виконання скрипту, якщо під час збирання або встановлення трапляться помилки. Типово, цей параметр вимкнено.</p>
svn-server	Параметр модуля перевизначає загальний параметр	<p>Цей параметр використовується для того, щоб вказати сервер, який слід використати для отримання з Subversion. Типовим є анонімне сховище Subversion, <code>svn://anonsvn.kde.org/</code></p> <div data-bbox="998 1318 1334 1591" style="border: 1px solid black; padding: 5px;"> <p>ПРИМІТКА Якщо ви є розробником KDE, ви можете скористатися власним обліковим записом Subversion, який ви отримали разом з обліковим записом розробника, замість анонімного запису.</p> </div>

tag	<p>Параметр модуля перевизначає загальний параметр</p>	<p>Використовуйте цей параметр для отримання певної версії модуля. <i>Зауваження:</i> ймовірно, вам <i>не доведеться</i> використовувати цей параметр. Випуски KDE можна отримати у форматі архівів з сайта отримання пакунків KDE.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>ПРИМІТКА Підтримку цього параметра передбачено лише для модулів з кодом у git з версії kdesrc-build 1.16.</p> </div>
use-clean-install	<p>Параметр модуля перевизначає загальний параметр (параметри системи збирання)</p>	<p>Встановіть для цього параметра значення true, якщо kdesrc-build слід виконати make uninstall безпосередньо перед виконанням команди make install. Цей параметр може бути корисним для забезпечення вилучення застарілих файлів бібліотек, метаданих CMake тощо, які можуть призвести до проблем у екземплярах KDE, які мають довгу історію. Втім, цей параметр допоможе позбутися зайвих даних лише у системах збирання, у яких передбачено підтримку виконання команди make uninstall. Цей параметр було додано у kdesrc-build 1.12, але не було задокументовано до версії kdesrc-build 1.16.</p>
use-cmake		<p>Цей параметр було вилучено у kdesrc-build 1.4, оскільки для збирання всіх модулів KDE 4 потрібен CMake, а використання CMake не дозволене для будь-яких інших модулів.</p>

Підручник зі скрипту kdesrc-build

use-idle-io-priority	Неможливо перевизначити	За допомогою цього параметра, доданого у kdesrc-build 1.12, може зменшити пріоритет використання скриптом диска та інших інструментів вводу-виводу, це може значно покращити швидкість решти системи під час роботи скрипту за рахунок збільшення часу роботи kdesrc-build. Типово таку поведінку вимкнено, щоб зменшити пріоритет використання диска, встановить значення true .
----------------------	-------------------------	--

use-modules

Можна використовувати лише у [module-set](#)

За допомогою цього параметра, доданого у kdesrc-build 1.12.1, вам буде простіше вказати набір модулів, які збиратимуться на певному етапі, у [файлі налаштування](#).

Використовувати цей параметр можна *лише* у межах визначення **module-set**. Всі ідентифікатори, передані цьому параметру, перетворюються у розгорнуту форму у модулі kdesrc-build.

Значення параметра **repository** буде встановлено на основі даних щодо сховища набору модулів і назви-ідентифікатора модуля. Всі інші параметри, визначені у наборі модулів також буде скопійовано до створених записів модулів без змін.

Порядок, у якому модулі визначено у цьому параметрі, є важливим, оскільки це саме той порядок, у якому kdesrc-build оброблятиме створені записи модулів під час оновлення, збирання і встановлення. Всі модулі, визначені у певному наборі модулів, буде оброблено до переходу kdesrc-build до наступного після набору модулів модуля.

Якщо вам потрібно змінити параметри створеного модуля, просто оголосіть модуль ще раз після його визначення у наборі модулів і вкажіть нові параметри. Хоча таким чином ви внесете зміни до параметри модуля, модуль все одно буде оновлено і зібрано у порядку, вказаному у наборі модулів (тобто ви не зможете таким чином

use-qt-builddir-hack	Параметр модуля перевизначає загальний параметр	Цей параметр було вилучено через покращення у системі збирання Qt™.
use-stable-kde	Неможливо перевизначити	Цей параметр є застарілим і його буде вилучено (знову) у наступному випуску програми. Будь ласка, користуйтеся замість цього параметра параметром branch-group , який надає ширші можливості вибору.

Табл. 4.1: Таблиця параметрів

Розділ 5

Параметри командного рядка і змінні середовища

5.1 Користування за допомогою командного рядка

kdesrc-build має бути запущено так:

```
kdesrc-build [--параметри...] [модулі, які слід зібрати...]
```

Якщо у командному рядку не буде вказано жодних модулів, kdesrc-build збиратиме всі модулі, визначені у файлі налаштувань, за вказаним у файлі порядком (таку поведінку можна змінити за допомогою різних параметрів файла налаштувань).

5.1.1 Типові параметри командного рядка

Повний список параметрів командного рядка наведено у Розділ 5.3. Ось найпоширеніші параметри:

--pretend (або -p)

За допомогою цього параметра kdesrc-build можна вказати, які дії слід виконати, без їхнього виконання. Це може бути корисним, якщо вам потрібно переконатися, що модулі, які вам потрібно зібрати, буде насправді зібрано.

--refresh-build

За допомогою цього параметра можна наказати kdesrc-build зібрати вказані модулі від початку. Всі вже створені каталоги збирання модуля буде вилучено, після чого його збирання буде повторено. Цей параметр корисний, якщо під час збирання модуля сталися помилки. Іноді ним доведеться користуватися у разі зміни бібліотек Qt™ або KDE.

--no-src

За допомогою цього параметра можна пропустити процедуру оновлення коду. Ви можете скористатися ним, якщо у вас є нещодавно оновлений початковий код (ви могли зробити це вручну або під час нещодавнього запуску kdesrc-build), але вам перезібрати певні модулі.

--no-build

Цей параметр подібний до наведеного вище параметра **--no-src**, але у разі його використання процес збирання буде пропущено.

5.1.2 Визначення модулів для збирання

Загалом, визначення модулів для збирання доволі просте: достатньо передати назву модуля, яку визначено у файлі налаштувань. Ви також можете передати модулі, які є частиною набору модулів у форматі назв `use-modules` або назви всього набору модулів, якщо ви дали йому назву.

У окремому випадку наборів модулів, заснованих на [базі даних проєктів KDE](#), kdesrc-build розгорне компоненти назв модулів для визначення потрібного вам модуля. Наприклад, запис проєкту KDE kdesrc-build відповідає проєкту у `extragear/utils/kdesrc-build`. Ви можете вказати будь-що з вказаного нижче для збирання kdesrc-build:

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

ПРИМІТКА

Команди у наведеному вище прикладі вказано перед назвою модуля з префіксом `+`. Це наказує програмі вважати назву модуля назвою з бази даних проєктів KDE, навіть якщо цей модуль не було визначено у вашому файлі налаштувань.

Будьте обережні з визначенням дуже загальних проєктів (наприклад `extragear/utils`), оскільки це може призвести до збирання надмірної кількості модулів. Вам слід скористатися параметром `--pretend` перед збиранням нового набору модулів, щоб було зібрано лише потрібні вам модулі.

5.2 Підтримувані змінні середовища

kdesrc-build не використовує змінних середовища. Якщо у вас виникне потреба у встановленні змінних середовища для процесів збирання або встановлення, будь ласка, скористайтеся параметром `set-env`.

5.3 Параметри командного рядка

Цей скрипт сприймає такі параметри командного рядка:

`--async`

Вмикає [асинхронний режим](#), за якого можливе одночасне виконання оновлення кодів та збирання модулів. Така поведінка є типовою, вам слід вказувати значення цього параметра, лише якщо ви вимкнули асинхронний режим у налаштуваннях.

`--help`

Показати лише короткі довідкові відомості про скрипт.

`--version` (або `-v`)

Показує версію програми.

`--show-info`

Показати відомості щодо kdesrc-build та операційної системи. Ці відомості можуть бути корисними для звітів щодо вад або запитань на форумах чи у списках листування.

Доступний з версії 18.11.

Підручник зі скрипту kdesrc-build

--initial-setup

Наказує kdesrc-build виконати одноразове початкове налаштування, яке потрібне для приготування системи до роботи з kdesrc-build та забезпечення можливості запуску нововстановленого програмного забезпечення KDE.

Це зокрема:

- Встановлення відомих залежностей (для підтримуваних дистрибутивів Linux[®])
- Додавання потрібних змінних середовища до `~/.bashrc`
- Визначення параметрів [файла налаштувань](#)

Доступний з версії 18.11.

--author

Показує інформацію для контактів з автором.

--color

Вмикає виведення інформації у кольорі. (Типова поведінка для інтерактивних терміналів).

--nice=значення

За допомогою цього параметра можна скоригувати пріоритет використання процесора kdesrc-build. Значенням має бути число у діапазоні 0-20. 0 відповідає найвищому пріоритету, 20 — найнижчому. Типовим значенням для kdesrc-build є 10.

--no-async

Вмикає [асинхронний режим](#) оновлення. Натомість оновлення перед початком збирання відбуватиметься повне оновлення вихідних кодів. Використання цього параметри уповільнить процес збирання, але, якщо ви бачите помилки щодо ІРС під час виконання скрипту kdesrc-build, спробуйте скористатися цим параметром, а також надішліть [звіт про ваду](#).

--no-color

Вмикає виведення інформації у кольорі.

--pretend (або -p)

kdesrc-build почне процес оновлення і збирання, але замість виконання будь-яких дій з оновлення чи збирання, скрипт виведе перелік дій, які мав би виконати (наприклад, які команди буде виконано, загальні кроки процесу тощо).

ПРИМІТКА

Скрипт виконає прості команди читання (зокрема читання даних щодо файлів), щоб виведені дані відповідали поточній конфігурації (наприклад, щоб правильно імітувати дію: повне отримання чи просте оновлення коду).

ВАЖЛИВО

Для забезпечення працездатності цього параметра потрібні деякі метадані, які зазвичай отримуються у автоматичному режимі, але у режимі фіктивної обробки отримання даних вимкнено. Якщо ви до цього ще ніколи не запускали kdesrc-build (а тому не маєте відповідних метаданих), вам слід віддати команду **kdesrc-build --metadata-only**, щоб програма спочатку отримала потрібні їй метадані.

--quiet (або -q)

Виводити лише скорочену інформацію. За використання цього параметра буде виведено лише базові відомості.

--really-quiet

Повідомляти лише про попередження і помилки.

Підручник зі скрипту kdesrc-build

`--verbose`

Виводити докладну інформацію про те, що відбувається, і про дії, які виконує kdesrc-build.

`--src-only` (або `--svn-only`)

Виконати лише оновлення коду (параметр `--svn-only` підтримується лише з міркувань сумісності зі старішими версіями).

`--build-only`

Виконати лише процес збирання.

`--install-only`

Виконати лише процес встановлення.

`--metadata-only`

Виконали лише дії з отримання метаданих. Зазвичай, kdesrc-build виконує отримання метаданих у автоматичному режимі, тому цей параметр потрібен, лише якщо вам хочеться уможливити виконання команди з параметром `--pretend`.

`--rebuild-failures`

Цим параметром можна скористатися для збирання лише тих модулів, які не вдалося зібрати під час попереднього запуску kdesrc-build. Це корисно, якщо одночасно із успішним збиранням декількох модулів не вдалося зібрати значну кількість інших модулів. Після виправлення помилки, що спричинила неможливість збирання, ви можете зібрати усі ті модулі, які не вдалося зібрати раніше.

ПРИМІТКА

Зауважте, що список модулів, які не вдалося зібрати раніше, переписується кожного разу, коли запуск kdesrc-build завершується із помилками під час збирання модулів. Втім, такий список не змінюється, якщо збирання було повністю успішним, тому ви можете успішно повторно зібрати один або два модулі, а цей параметр все одно працюватиме.

Цей параметр було додано у версії kdesrc-build 15.09.

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

Використання цього параметра призводить до того, що kdesrc-build автоматично включає інші модулі KDE та Qt™ до процедури збирання, якщо вони потрібні для модулів, збирання яких було вказано за допомогою параметрів командного рядка або вашого [файла налаштувань](#).

Перелік доданих модулів визначатиметься записами у системі керування початковими кодами програм KDE. Див. Розділ [2.6.4](#).

Відповідним параметром у файлі налаштувань є параметр `include-dependencies`.

Ви також можете скористатися параметром `--no-include-dependencies`, який вимикає автоматичне включення додаткових модулів-залежностей.

`--ignore-modules`

Не включати модулі, передані як аргумент параметра командного рядка, у процес оновлення або збирання (корисно, якщо ви бажаєте зібрати більшість модулів у вашому [файлі налаштування](#) і просто пропустити збирання малої частини списку модулів).

`--no-src` (або `--no-svn`)

Пропустити встановлення зв'язку з сервером Subversion (параметр `--no-svn` підтримується лише з міркувань сумісності зі старішими версіями скрипту).

`--no-build`

Пропустити процес збирання.

Підручник зі скрипту kdesrc-build

`--no-metadata`

Не отримувати автоматично додаткових метаданих для модулів git KDE. Оновлення самого коду модулів все одно відбуватиметься, якщо ви не додасте параметр `--no-src`.

Цей параметр корисний, якщо ви часто перезапускаєте `kdesrc-build`, оскільки метадані змінюються доволі нечасто. Втім, слід пам'ятати, що метадані потрібні для роботи багатьох інших можливостей програми. Ви можете просто запустити `kdesrc-build` з параметром `--metadata-only` один раз, а потім користуватися цим параметром під час наступних запусків.

`--no-install`

Не встановлювати автоматично пакунки після збирання.

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

За допомогою цього параметра можна явним чином вимкнути пропуски у процесі збирання (оптимізацію, якою керує `build-when-unchanged` параметр). Параметр буде корисним для запуску збирання за допомогою `kdesrc-build` у разі внесення вами змін, які не може бути автоматично виявлено `kdesrc-build`.

`--force-build` призначено для виконання того ж завдання, крім того, можливо, цей параметр легше запам'ятати.

`--debug`

Вмикає режим усунення вад для скрипту. У поточній версії це означає, що всі виведені скриптом дані буде спрямовано у стандартний вивід оболонки на додачу до їх запису до каталогу журналу у звичайному режимі. Крім того, у такому режимі багато з функціональних можливостей скрипту надають у цьому режимі значно докладніші відомості про свою роботу.

`--query` =режим

За допомогою цієї команди можна наказати `kdesrc-build` визначити значення параметра модулів у списку збирання (з командного рядка або з файлу налаштувань) і вивести результат на екран (один рядок для кожного з модулів).

Разом із цим параметром слід вказати «режим запиту», одне з таких значень:

- `source-dir` — наказує `kdesrc-build` вивести повний шлях до каталогу, де зберігається код модуля.
- `build-dir` — наказує `kdesrc-build` вивести повний шлях до каталогу, де відбуватиметься процес збирання.
- `install-dir` — наказує `kdesrc-build` вивести повний шлях до каталогу, куди буде встановлено модуль.
- `project-path` — наказує `kdesrc-build` вивести розташування модуля у ієрархії сховищ із початковим кодом програм KDE. Докладний опис цієї ієрархії наведено у розділі Розділ 2.6.4.
- `branch` — наказує `kdesrc-build` вивести назву визначеної гілки git, яку буде використано для кожного з модулів на основі поточних значень параметрів `tag`, `branch` та `branch-group`.
- Крім того, можна використовувати коректні варіанти параметрів для модулів з файлу налаштувань, де наведено виявлені значення для кожного з модулів.

Якщо за допомогою командного рядка передано назву одного модуля, виведено буде лише значення вказаного параметра. Якщо ж у командному рядку вказано декілька модулів (або не вказано жодного модуля), до кожного рядка буде додано префікс із назвою модуля. У обох випадках `kdesrc-build` завершити роботу одразу після виведення вказаних значень.

Цей параметр було додано у версії `kdesrc-build` 16.05.

Наприклад, команда «`kdesrc-build --query branch kactivities kdepim`» призведе до виведення даних, подібних до цих:

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

--no-rebuild-on-fail

Не намагатися перебудувати модулі, які не вдалося зібрати спочатку. kdesrc-build ніколи не намагатиметься виконувати збирання модуля, який скрипт вже намагався зібрати.

--refresh-build

Повторно створити систему збирання і виконати збирання з початку.

--reconfigure

Виконати **cmake** (для модулів KDE) або **configure** (для Qt™) знову, без спорожнення каталогу збирання. Зазвичай, потреби у такій поведінці немає, оскільки kdesrc-build сам може виявити зміни у відповідних параметрах і автоматично перезапустити налаштування збирання. Цей параметр буде використано, якщо ви скористаєтеся параметром ***--refresh-build***.

--resume-from

Цей параметр використовується для поновлення збирання, починаючи з вказаного модуля, назва якого має бути наступним параметром командного рядка. Крім того, вам не слід вказувати інших модулів у командному рядку.

ПРИМІТКА

У попередніх версія використання цього параметра автоматично додавало параметр ***--no-src***, але тепер програма цього не робить (з версії kdesrc-build 1.13). Якщо ви бажаєте уникнути оновлення кодів програма під час поновлення збирання, просто додайте до інших параметрів ***--no-src***.

Див. також ***--resume-after*** and Розділ 6.3.5.1. Вам слід надавати перевагу використанню цього параметра командного рядка, якщо ви виправили помилку збирання і бажаєте, щоб скрипт kdesrc-build завершив збирання.

--resume-after

Цей параметр використовується для поновлення збирання після вказаного модуля, назва якого має бути наступним параметром командного рядка. Крім того, вам не слід вказувати інших модулів у командному рядку.

ПРИМІТКА

У попередніх версія використання цього параметра автоматично додавало параметр ***--no-src***, але тепер програма цього не робить (з версії kdesrc-build 1.13). Якщо ви бажаєте уникнути оновлення кодів програма під час поновлення збирання, просто додайте до інших параметрів ***--no-src***.

Див. також ***--resume-from*** and Розділ 6.3.5.1. Вам слід надавати перевагу використанню цього параметра командного рядка, якщо ви виправили помилку збирання, а також збрали і встановили модуль власноруч, і бажаєте, щоб скрипт kdesrc-build розпочав з обробки наступного модуля.

--resume

Цим параметром можна скористатися для запуску kdesrc-build після аварійного завершення роботи програми під час збирання.

За допомогою цього параметра можна відновити збирання з модуля, під час збирання якого сталася критична помилка, на основі попереднього списку модулів, які очікують на збирання. Оновлення коду та метаданих буде вимкнено. Цей параметр призначено для випадків, коли проста помилка або невстановлена залежність призводить до неможливості зібрати модуль. Щойно помилку буде виправлено, а залежність встановлено, ви можете швидко повернутися до збирання модулів без додаткових вправ зі встановлення належних аргументів для параметрів ***--resume-from*** та ***--stop-before***.

Цей параметр зручно використовувати разом з параметром ***--stop-on-failure***, особливо, якщо це відбувається на початковому кроці налаштування середовища для розробки.

Цей параметр було додано у версії kdesrc-build 1.16.

--stop-before

Цим параметром командного рядка можна скористатися для того, щоб зупинити звичайне збирання *до того*, як розпочнеться збирання вказаного модуля.

Наприклад, якщо порядок у звичайному списку збирання був таким: модульА, модульВ, модульС, і вказано **--stop-before=модульВ**, kdesrc-build виконає збирання лише модуля модульА.

Цей параметр було додано у kdesrc-build 1.16.

--stop-after

Цим параметром командного рядка можна скористатися для того, щоб зупинити звичайне збирання *після того*, як розпочнеться збирання вказаного модуля.

Наприклад, якщо порядок у звичайному списку збирання був таким: модульА, модульВ, модульС, і вказано **--stop-after=модульВ**, kdesrc-build виконає збирання модулів модульА та модульВ.

Цей параметр було додано у kdesrc-build 1.16.

--stop-on-failure

За допомогою цього параметра можна наказати програмі припинити збирання за першої з критичної помилки. Корисний, якщо виконується початкове налаштування середовища для розробки. Без цього параметра kdesrc-build намагатиметься продовжити збирання решти модулів у наборі, щоб уникнути втрати часу, якщо проблема виникла лише з одним з цих модулів.

Цей параметр було додано у версії kdesrc-build 1.16. Див. також параметр [stop-on-failure](#).

--rc-file

Інтерпретує наступний параметр командного рядка, як файл, з якого слід читати параметри налаштування. Типовим значенням цього параметра є `kdesrc-buildrc` (скрипт шукатиме його у поточному каталозі), якщо його буде знайдено, або `~/.kdesrc-buildrc`, якщо цей файл не вдасться знайти. Див. також розділ 4.

--print-modules

Виконує усі дії, вказані до цього параметра, зокрема перевпорядкування залежностей модулів, вказане у командному рядку (або файлі налаштувань), виводить список модулів, обробку яких буде виконано, по одному на рядок, потім завершує роботу без виконання подальших дій.

Спочатку буде отримано метадані `kde-project` (втім, див. [--pretend](#) або [--no-src](#)).

Виведені дані не повністю сумісні з тими, які використовуються скриптами, оскільки до списку модулів може бути виведено додаткові повідомлення.

Параметр корисний для пришвидшення визначення того, що kdesrc-build вважає залежностями модуля, тобто найкорисніший для модулів `kde-projects`. Цей параметр сумісний з параметрами [--resume-from](#), [--resume-after](#), [--stop-before](#), [--stop-after](#).

--list-build

Виводить список модулів, які має бути зібрано. Порядок модулів у списку відповідатиме порядку, у якому ці модулі буде зібрано. Якщо такі дані є застосовними, у виведеному списку буде також вказано внесок, гілку і мітку, які буде вибрано для отримання даних модулів.

Цей параметр є подібним до [--print-modules](#). Докладний опис зв'язку модулів один з одним можна знайти у розділі щодо [--dependency-tree](#).

--dependency-tree

Виводить дані щодо залежностей модулів, які має бути зібрано, з використанням рекурсивного ієрархічного формату. У виведених даних буде зазначено внесок, гілку і мітку залежностей та те, чи має бути зібрано залежність. Зауваження: виведені дані можуть бути доволі об'ємними для програм із багатьма залежностями.

Підручник зі скрипту kdesrc-build

--run

Цей параметр інтерпретує наступний елемент командного рядка як назву програми, яку слід запустити. `kdesrc-build` завершить читання файлу налаштувань, відновить попередні налаштування середовища, а потім виконає вказану програму.

Але цим параметром здебільшого не можна користуватися для запуску оболонки за допомогою середовища `kdesrc-build`, оскільки інтерактивні оболонки типово змінюють принаймні частину змінних середовища (зокрема `PATH` і `KDEDIRS`) під час процедури свого запуску.

ПІДКАЗКА

Якщо ви бажаєте переглянути всі параметри середовища, які буде використано `kdesrc-build`, ви можете скористатися командою **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
... тощо.
```

--prefix=<шлях/до/kde>

За допомогою цього параметра можна змінити каталог, куди KDE буде встановлено з командного рядка. За використання цього параметра автоматично використовується параметр `--reconfigure`, але можливо доведеться окремо скористатися параметром `--refresh-build`.

--revision

За допомогою цього параметра можна наказати `kdesrc-build` отримати певну версію кожного з модулів Subversion, незалежно від параметрів `branch`, `tag` та `revision` вже встановлених для цих модулів.

Вам не варто користуватися цим параметром, він підтримується лише з міркувань сумісності зі старішими версіями скрипту.

--build-system-only

За допомогою цього параметра можна наказати `kdesrc-build` перервати збирання модуля безпосередньо перед виконанням команди **make**. Параметр підтримується лише з міркувань сумісності зі старішими версіями скрипту, у ньому немає користі за поточної системи збирання KDE.

--install

Якщо цей параметр буде єдиним, скрипт намагатиметься встановити всі модулі, вказані у `log/latest/build-status`. Якщо після `--install` буде вказано інші параметри командного рядка, всі ці параметри вважатимуться назвами модулів, які слід встановити (навіть якщо їх не вдалося зібрати під час попереднього запуску).

--no-snapshots

Використання цього параметра призведе до того, що `kdesrc-build` завжди виконуватиме звичайне початкове отримання коду модуля замість використання початкового архіву зі знімком (можна використовувати лише для модулів Git зі сховища `kde-projects`). Зауважте, що цим параметром варто користуватися, лише якщо вам не вдається скористатися знімком з кодом, оскільки такі знімки зменшуються навантаження на сховища з кодом KDE.

ПРИМІТКА

Знімки модулів є самі версіями каталогів. Вам не варто використовувати цей параметр, його використання виправдане лише з метою усунення вад.

--delete-my-patches

Цей параметр призначено для того, щоб дозволити kdesrc-build вилучати каталоги початкового коду, у яких можуть міститися записані користувачем дані, щоб модуль можна було повністю переотримати. Зазвичай, така можливість буде корисною лише для розробників KDE, тобто тих, хто вносить до коду зміни локально і хоче, щоб ці зміни було згодом вилучено.

У поточній версії цей параметр використовується лише для отримання коду модулів, які було перенесено з Subversion до Git. Зазвичай, потреби у використанні цього параметра немає. Якщо у параметрі виникне потреба, kdesrc-build попросить вас перезапустити програму з ним.

--delete-my-settings

Цим параметром можна скористатися для того, щоб дозволити kdesrc-build перезаписувати вже створені файли, у яких можуть міститися дані, записані туди користувачем.

У поточній версії цей параметр використовується лише для налаштування xsession у програмі для керування сеансами. Зазвичай, потреби у використанні цього параметра немає. Якщо у параметрі виникне потреба, kdesrc-build попросить вас перезапустити програму з ним.

--<назва параметра>=

Ви можете скористатися цим параметром, щоб перевизначити відповідний параметр у вашому [файлі налаштувань](#) для кожного з модулів. Наприклад, щоб перевизначити значення параметра `log-dir`, вам слід вказати: `--log-dir=шлях/до/каталогу`.

ПРИМІТКА

Цією можливістю можна скористатися лише для тих назв параметрів, які вже розпізнаються kdesrc-build і ще не підтримуються відповідними параметрами командного рядка. Наприклад, параметр файла налаштувань `async` має специфічні параметри командного рядка, `--async` та `--no-async`, яким надається перевага у kdesrc-build.

---set-module-option-value=<назва модуля>, <назва параметра>, <значення параметра>

Скористайтеся цим параметром, щоб перевизначити параметр з вашого [файл налаштувань](#) для окремого модуля.

Всі інші параметри командного рядка вважатимуться назвами модулів, які слід оновити і зібрати. Будь ласка, не плутайте збирання зі встановленням.

Розділ 6

Користування kdesrc-build

6.1 Передмова

Зазвичай, використання kdesrc-build після виконання кроків розділу розділ 2 зводиться до виконання такої команди у терміналі:

```
% kdesrc-build
```

Після цього kdesrc-build отримає коди програм KDE, спробує налаштувати (configure) і зібрати їх, а потім встановити програми.

Далі ми поговоримо про те, як kdesrc-build робить це, а також про те, які дії ви можете виконати за допомогою цього інструменту.

6.2 Основні можливості kdesrc-build

6.2.1 Підтримка qt

kdesrc-build підтримує зручне для користувачів збирання набору інструментів Qt™, який використовується у програмному забезпеченні KDE. Підтримку забезпечує обробка особливого модуля з назвою qt.

ПРИМІТКА

Qt™ розробляється у окремому сховищі, не пов'язаному з програмним забезпеченням KDE. Це сховище розташоване за адресою <http://code.qt.io/cgit/qt/>.

Щоб зібрати Qt™, вам слід переконаватися, що параметр `qtdir` вказує на каталог, куди б ви бажали встановити Qt™, як це описано у розділі Розділ 2.2.

Після цього вам слід переконаватися, що модуль qt додано до вашого файла `.kdesrc-buildrc` перед будь-якими іншими модулями у цьому файлі. Якщо ви користуєтеся зразком файла налаштувань, ви можете просто зняти позначку коментаря з запису модуля qt.

Тепер вам слід перевірити, чи встановлено належним чином параметри `repository` і `branch`:

1. Перший варіант полягає у збиранні Qt™ на основі коду з дзеркала зі сховищ коду KDE (ніяких змін до коду не внесено, це просто клон офіційного сховища). Ми рекомендуємо цей варіант через можливі проблеми з клонуванням модуля Qt™ з офіційного сховища повністю.

Ви можете встановити параметр `repository` для модуля qt у значення `kde:qt`, щоб скористатися цим параметром.

2. Іншим варіантом буде збирання стандартного Qt™. У такому разі слід встановити значення параметра `repository` рівним `git://gitorious.org/qt/qt.git`. Зауважте, що у вас можуть виникнути проблеми з початковим клонуванням Qt™ зі цього сховища.

У обох випадках параметр `branch` слід встановити у значення `master` (якщо ви не маєте наміру спеціально зібрати якусь іншу гілку).

6.2.2 Стандартні прапорці, які додаються kdesrc-build

Варто зауважити, що цей розділ не стосується модулів, для яких налаштовано нетиповий набір інструментів, наприклад, за допомогою `cmake-toolchain`.

Щоб зекономити вам час, `kdesrc-build` додасть деякі з стандартних шляхів до вашого середовища:

- Шлях до бібліотек KDE і Qt™ автоматично додається до змінної середовища `LD_LIBRARY_PATH`. Отже, вам не потрібно змінювати `libpath`, щоб включити відповідні каталоги.
- Шлях до програм підтримки розробки KDE і Qt™ буде автоматично додано до змінної середовища `PATH`. Отже, вам не потрібно змінювати `binpath`, щоб включити відповідна каталоги.
- Шлях до програми `pkg-config` KDE буде автоматично додано до `PKG_CONFIG_PATH`. Отже, вам не потрібно використовувати `set-env` для додавання відповідних каталогів.
- Це значення `kdedir` буде автоматично передано змінній середовища `KDEDIR` на час збирання. (Змінна `KDEIRS` залишатиметься незмінною.)
- Це значення для `qtdir` буде автоматично передано змінній середовища `QTDIR` на час збирання.

6.2.3 Зміна пріоритету збирання kdesrc-build

У сучасних операційних системах ви можете запускати програми з різним рівнем пріоритетності, це зокрема можна робити у Linux® і BSD. Система розподілятиме час процесора між різними програмами відповідно до того, наскільки пріоритетними є ці програми.

Зазвичай, `kdesrc-build` надає собі низький пріоритет так, щоб решта програм вашої системи не відчули змін і могли виконуватися у звичному режимі. Таким чином, `kdesrc-build` використовуватиме лише залишковий час процесора.

Але `kdesrc-build` підтримуватиме достатньо великий рівень пріоритету, а отже скрипт виконуватиметься з вищим пріоритетом, ніж пакетні процеси і програми з надання процесорного часу, на зразок `Seti@Home`.

Ви можете змінити рівень пріоритетності `kdesrc-build` на вищий (або нижчий), — для цього вам достатньо змінити значення параметр `nice` у файлі налаштувань. Значення параметра `nice` керує тим, наскільки «поблажливим» `kdesrc-build` буде до інших програм. Іншими словами, встановлення більшого значення параметра `nice` надасть `kdesrc-build` менший пріоритет. Отже, щоб надати `kdesrc-build` вищий пріоритет, зменшіть значення `nice` (і навпаки). Значеннями параметра `nice` можуть бути цілі числа від 0 (ніяких компромісів, — найвищий пріоритет) до 20 (найкомпромісніше виконання, — найнижчий пріоритет). і числа від 0 (ніяких компромісів, — найвищий пріоритет) до 20 (найкомпромісніше виконання, — найнижчий пріоритет).

Крім того, ви можете тимчасово змінити пріоритет `kdesrc-build` за допомогою параметра командного рядка `--nice`. Значення цього параметра таке саме, як і значення `nice`.

ПРИМІТКА

Деякі з програм, які виконуються від імені адміністративного користувача, можна виконувати з від'ємним рівнем `nice`, такі програми матимуть ще вищий рівень пріоритетності. Встановлювати від'ємного (або навіть нульового) значення `nice` для `kdesrc-build` не варто, подібні значення не зможуть значно пришвидшити роботу програми, але значно сповільнять виконання інших завдань системи, які будуть вам потрібні.

Підручник зі скрипту kdesrc-build

Щоб запустити kdesrc-build з пріоритетом nice 15 (пріоритетом, нижчим за звичайний), виконайте команду:

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

Крім того, ви можете змінити [файл налаштувань](#), щоб зробити зміни сталими:

```
niceness 15
```

ПІДКАЗКА

За допомогою параметра `niceness` можна керувати лише використанням процесорів комп'ютера. Іншим важливим фактором, який впливає на швидкість системи, є ступінь використання програмою каналів вводу і виводу даних (I/O). З метою керування використанням I/O програмою у сучасних операційних системах Linux[®] передбачено підтримку подібного до nice інструменту з назвою `ionice`. kdesrc-build підтримує роботу з `ionice` (але лише у режимі «увімкнути-вимкнути») за допомогою параметра `use-idle-io-priority`, починаючи з версії kdesrc-build 1.12.

6.2.4 Встановлення від імені суперкористувача

Ймовірно, у вас може виникнути потреба у запуску kdesrc-build встановлення з правами доступу адміністративного користувача. Така потреба може виникнути під час загальносистемного встановлення, яке, втім, не рекомендується авторами програми. Ця можливість також корисна і під час рекомендованого збирання для окремого користувача KDE. Причиною цього є те, що деякі у деяких модулях (особливо у модулі kdebase) програми мають встановлюватися з підвищеними правами доступу до ресурсів системи. Ці програми не зможуть отримати потрібних дозволів, якщо ви не встановите їх з підвищеними правами.

Ви можете просто запустити kdesrc-build від імені адміністративного користувача, але ми не рекомендуємо вам цього робити, оскільки скрипт не було достатньо випробувано запуском у такому режимі. Хоча запуск програми у такий спосіб і не повинен призвести до шкідливих наслідків, вам краще за можливості уникати запуску скрипту від імені адміністративного користувача.

Для надання скрипту додаткових прав доступу у kdesrc-build передбачено параметр `make-install-prefix`. Ви можете скористатися цим параметром для того, щоб вказати команду, яка використовується для встановлення від імені іншого користувача. Рекомендуємо вам скористатися для цього програмою Sudo, ця програма надає змогу виконувати команди від імені адміністративного користувача.

Наприклад, щоб встановити всі модулі за допомогою Sudo, ви можете вказати щось таке:

```
global
  make-install-prefix sudo
  # Інші параметри
end global
```

Для того, щоб скористатися `make-install-prefix` для окремого модуля, вкажіть у файлі налаштувань рядки на зразок:

```
module назва-svn-модуля
  make-install-prefix sudo
end module
```

6.2.5 Показ поступу збирання модуля

Ця можливість завжди доступна, за можливості скрипт автоматично вмикає її. Вона надає вам змогу слідкувати за поступом збирання модуля, а отже ви знатимете скільки часу залишилося до завершення збирання модуля.

6.3 Додаткові можливості

6.3.1 Часткове збирання модуля

Існує також можливість збирання окремих фрагментів окремого модуля KDE. Наприклад, ви можете зібрати лише одну з програм модуля. У kdesrc-build передбачено засоби для полегшення такого збирання. Існує декілька шляхів виконати подібне збирання.

6.3.1.1 Отримання частин модуля

Ймовірно, це найкращий спосіб виконати завдання. Якщо він працюватиме, ви зекономите час на отримання і простір на диску. За використання цього способу kdesrc-build отримуватиме лише частини модуля, який ви вкажете. Досягти цього можна за допомогою використання параметра `checkout-only` для модуля, — цей параметр визначає список каталогів, які слід отримати.

Підказка

Якщо ви ще не знаєте, які саме частини модуля вам слід отримати, вам варто спочатку переглянути компонування Subversion цього модуля за допомогою [WebSVN](#).

Для того, щоб отримати лише KUser і KSystemlog з kdeadmin, ви можете скористатися `checkout-only` ось так:

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

ВАЖЛИВО

Каталоги збиратимуться у тому порядку, у якому ви вкажете їх за допомогою цього параметра. Якщо у одному з каталогів буде потрібен компонент, який ще не було зібрано, вам слід не забути вказати відповідні каталоги у рядку `checkout-only` і переконатися, що потрібний компонент вказано перед назвою іншого компонента, для збирання якого потрібен перший компонент.

Крім того, іноді певна програма може потребувати якихось інших компонентів, а визначити назви цих компонентів досить складно, — ця процедура може потребувати використання методу проб і помилок з постійним додаванням каталогів до аргументів параметра, щоб виявити потрібні залежності. Працездатність цього параметра залежить від підтримки з боку системи збирання модуля, отже ним можна буде скористатися, лише якщо модуль є збіркою окремих програм.

Нарешті, останнє зауваження щодо цього параметра: якщо ви зміните значення цього параметра, вам доведеться виконати команду `kdesrc-build --refresh-build назва_модуля`, щоб забезпечити правильне переналаштування модуля. Крім того, kdesrc-build ніколи не буде вилучати існуючих файлів, якщо ви усуватимете певні каталоги зі списку аргументів параметра `checkout-only` або додаватимете параметр до модуля, який вже було отримано.

6.3.1.2 Вилучення каталогів з процесу збирання

Замість обмеження списку об'єктів для отримання ви можете наказати скрипту отримати всі каталоги і зібрати всі з них, окрім декількох. Такий підхід може бути корисним, якщо під час збирання одного з каталогів завжди трапляються помилки і цей каталог не потрібен для збирання решти модуля.

Подібним збиранням можна керувати за допомогою параметра `do-not-compile`. Цей параметр працює подібно до параметра `checkout-only`, про який ми писали раніше, — йому передається список каталогів, які не слід збирати.

ВАЖЛИВО

Крім того, подібно до [checkout-only](#), для того, щоб зміни, внесені цим параметром, набули чинності, вам слід запустити скрипт налаштування системи збирання відповідного модуля ще раз. Це можна зробити за допомогою команди `kdesrc-build --reconfigure назва_модуля`.

Щоб вилучити каталог `python` з процесу збирання `kdebindings`, додайте такі рядки:

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

ПРИМІТКА

Працездатність цієї функції залежить від виконання деяких стандартів, які використовуються у більшості модулів KDE. Саме тому вона може не працювати для деяких програм.

6.3.2 Підтримка версій і випусків у kdesrc-build

6.3.2.1 Що таке версія (branch) і випуск (tag)?

Subversion підтримує керування журналом змін коду KDE. KDE використовує цю підтримку для створення версій, які розробляються, і створення сховищ випусків для зберігання коду випущених версій середовища.

Наприклад, розробники KMail можуть працювати над новою можливістю у іншій версії, щоб уникнути можливої непрацездатності версії, над якою працює більшість інших розробників. У цій версії відбувається розробка, хоча розробка також відбувається і у основному сховищі версій (яке називається `/trunk`).

З іншого боку, випуск є моментальним знімком сховища коду у певний момент часу. Таке сховище використовується командою адміністрування KDE для позначення версії коду, придатної для випуску, а також надає розробникам можливість доопрацьовувати їх код.

У Subversion немає ніяких відмінностей між версіями, випусками і експериментальною версією (`trunk`) коду. Подібне відокремлення є лише наслідком певної домовленості між розробниками. Це ускладнює належну підтримку версій і випусків у `kdesrc-build`. Все ж деякі речі можна робити і за допомогою скрипту.

6.3.2.2 Як користуватися версіями і випусками?

Підтримка версій і випусків забезпечується набором параметрів, які призначено для виконання дій у діапазоні від загального запису на певну версію, до визначення адреси URL для отримання досвідченими користувачами.

Найпростішим способом є використання параметрів `branch` і `tag`. Ви можете просто скористатися параметром з аргументом-назвою бажаної версії або випуску модуля, — `kdesrc-build` спробує визначити належну адресу у сховищі KDE для отримання коду. Для більшості модулів KDE таких способів працює дуже добре.

Щоб отримати `kdelibs` зі складу KDE 4.6 (відповідне сховище відоме як версія 4.6):

```
module kdelibs
  branch 4.6
  # інші параметри...
end module
```

Або, щоб отримати `kdemultimedia` у вигляді, у якому цей модуль було випущено у KDE 4.6.1:

```
module kdemultimedia
  tag 4.6.1
  # інші параметри...
end module
```

ПІДКАЗКА

Ви можете вказати загальне значення версії (branch). Але, якщо ви це зробите, не забудьте вказати іншу версію для модулів, які не використовують загальної нумерації версій!

6.3.2.3 Додаткові параметри підтримки версій

kdesrc-build підтримує два параметри, призначені для обробки ситуацій, коли параметри `branch` і `tag` неправильно визначають шлях до сховища: `module-base-path` і `override-url`.

- Параметр `module-base-path` використовується для того, щоб допомогти kdesrc-build заповнити відсутню частину шляху до модуля. У сховищі KDE всіх шляхи виглядають так: Корінь_svn/базовий_шлях_до_модуля/ назва модуля. Зазвичай, kdesrc-build може визначити відповідну середню частину самостійно. Якщо програма не може цього зробити, ви можете скористатися `module-base-path`, ось так:

```
module kdesupport
  # kdesupport підтримує різноманітні мітки, які спрощують
  # впорядкування програмного забезпечення для певного випуску Платф ←
  орми KDE.
  module-base-path tags/kdesupport-for-4.5
end module
```

Таким чином ви накажете kdesrc-build отримати kdesupport з (у нашому прикладі), `svn://anonsvn.kde.org/home/kde/ tags/kdesupport-for-4.5`.

ПІДКАЗКА

У попередніх випусках kdesrc-build обробка `module-base-path` виконувалася у дещо інший спосіб. Якщо у вас виникають труднощі з використанням застарілого визначення `module-base-path`, ймовірно, вам слід перевірити, чи є адреса саме тією, яка насправді потрібна kdesrc-build, за допомогою параметра `--pretend`.

- З іншого боку, параметр `override-url` вимагає від вас вказати точний шлях для отримання. Але він надає вам змогу взяти шляхи з попередніх версій kdesrc-build, звідки вже не можна отримати дані. У поточній версії параметра `module-base-path` має бути достатньо для будь-якої адреси Subversion кодів KDE.

ВАЖЛИВО

kdesrc-build не змінюватиме і не виправлятиме значення, яке ви вкажете для `override-url`, отже, якщо ви зміните ваш параметр `svn-server`, може виникнути потреба і у зміні цього параметра.

6.3.3 Способи, у які kdesrc-build намагається забезпечити успішне збирання

6.3.3.1 Автоматичне перезбирання

У попередніх версіях kdesrc-build було передбачено можливість для автоматичного повторення спроби збирання модуля після невдалої спроби (оскільки іноді така повторна спроба буває

успішною, причиною можуть бути вади у системі збирання). Завдяки переходу на CMake, система збирання вже не страждає на подібні вади, отже kdesrc-build не намагатимуться зібрати модуль повторно. Але можливі випадки, коли kdesrc-build виконає повторне збирання автоматично:

- Якщо ви зміните `configure-flags` або `cmake-options` для певного модуля, kdesrc-build виявить ці зміни і повторно запустить скрипт `configure` або `cmake` для відповідного модуля.
- Якщо системи збирання не існує (навіть якщо kdesrc-build не вилучав її), kdesrc-build автоматично створить її повторно. Це буде корисним, якщо ви виконаєте повне повторне збирання (`--refresh-build`) для певного модуля без виконання повторного збирання для інших модулів.

6.3.3.2 Перезбирання модуля вручну

Якщо ви змінили параметри певного модуля або код модуля змінився таким чином, що kdesrc-build не може його розпізнати, у вас може виникнути потреба у повторному збиранні цього модуля вручну.

Ви можете зробити це простою командою `kdesrc-build --refresh-build назва_модуля`.

Якщо замість цього ви бажаєте, щоб скрипт kdesrc-build автоматично перезібрав модуль під час наступного звичайного оновлення збирання, ви можете створити спеціальний файл. Кожен з модулів має власний каталог для збирання. Якщо ви створите у цьому каталозі для збирання файл з назвою `.refresh-me`, kdesrc-build повторно збере модуль під час наступного сеансу збирання, навіть якщо скрипт виконує швидке нарощувальне збирання.

ПІДКАЗКА

Типово, каталогом збирання буде `~/kdesrc/build/ назва модуля /`. Якщо ви зміните значення параметра `build-dir`, буде замість `~/kdesrc/build` буде використано значення цього параметра.

Повторне збирання за допомогою `.refresh-me` модуля *kdelibs*:

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

6.3.4 Зміна параметрів змінних середовища

Зазвичай kdesrc-build використовує поточні параметри середовища для запуску програм, призначених для виконання оновлень та збирання. Це корисно, якщо ви запускаєте kdesrc-build з командного рядка.

Крім того, у вас може виникнути потреба у зміні тих змінних середовища, для яких у kdesrc-build не передбачено параметрів. (Наприклад, щоб визначити будь-які змінні середовища під час запуску kdesrc-build за таймером, наприклад Cron.) Це можна зробити за допомогою параметра `set-env`.

На відміну від інших параметрів, цей параметр можна визначити декілька разів, він приймає два записи, відокремлені пробілом. Першим з записів є назва змінної середовища, яку слід встановити, решта рядка вважається значенням цієї змінної.

Встановити значення змінної `DISTRO=BSD` для всіх модулів:

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


6.3.5 Поновлення збирання

6.3.5.1 Поновлення збирання, яке завершилося помилкою або було скасоване

Ви можете наказати kdesrc-build почати збирання з модуля, відмінного від того, з якого скрипт зазвичай розпочинає збирання. Такі налаштування можуть стати у пригоді, якщо спроба збирання певного набору модулів завершилася невдало або якщо ви перервали попередню спробу збирання. Вказати модуль можна за допомогою параметрів `--resume-from` та `--resume-after`.

ПРИМІТКА

У старіших версіях kdesrc-build під час поновлення збирання оновлення кодів програм не відбувалося. Тепер програма типово цього не робить, але ви можете передати їй параметр командного рядка `--no-src`, щоб уникнути оновлення кодів програм.

Відновлення збирання з модуля kdebase:

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

Відновлення збирання, починаючи з місця після kdebase (якщо ви вручну виправили ваду і встановили модуль власноруч):

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

Якщо остання спроба збирання за допомогою kdesrc-build завершилася критичною помилкою, ви також можете скористатися параметром `--resume`. Використання цього параметра надасть змогу відновити останній сеанс збирання з модуля, під час збирання якого сталася критична помилка. Крім того, буде пропущено кроки оновлення коду та метаданих модулів (якщо ви хочете, щоб програма виконала ці кроки, краще скористатися параметром `--resume-from`).

6.3.5.2 Ігнорування модулів під час збирання

Подібно до способу, у який ви можете поновити збирання, починаючи з певного модуля, ви можете і оновити зібрати компоненти у звичайному режимі, але без певного набору модулів.

Виконати це завдання можна за допомогою параметра `--ignore-modules`. За допомогою цього параметра ви можете повідомити скрипту kdesrc-build про те, що слід ігнорувати всі передані як аргументи у командному рядку модулі під час виконання оновлення і збирання.

Ігнорування extragear/multimedia і kdereview під час повноцінного запуску:

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

6.3.6 Зміна параметрів за допомогою командного рядка

6.3.6.1 Зміна загальних параметрів

Ви можете змінити набір параметрів з [файла налаштувань](#) безпосередньо з командного рядка. Ця зміна перевизначить параметри файла налаштувань, але лише тимчасово. Ці параметри працюватимуть лише протягом виконання вказаної у командному рядку команди.

kdesrc-build надає вам змогу змінити параметри з назвою *назва-параметра* передаванням аргументу у командний рядок у формі `--назва-параметра=значення`. kdesrc-build визначає чи відомий скрипту параметр і намагається знайти назву у власному списку назв параметр. Якщо програмі не вдається знайти назву, вона попередить вас про це, у іншому випадку скрипт запам'ятає встановлене вами значення і перевизначає значення, встановлення у файлі налаштувань.

Встановлення параметра `source-dir` у значення `/dev/null` для перевірки:

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


6.3.6.2 Зміна параметрів для модулів

Крім того, можна змінити параметри, вказані для окремого модуля. Синтаксис команди, подібний до синтаксису попередньої команди: `--назва_модуля,назва-параметра=значення`.

Ця зміна перевизначає параметр відповідного модуля у файлі налаштувань і діє лише протягом виконання рядка команди.

Використання іншого каталогу збирання для модуля kdedu:

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

6.4 Можливості, призначені для розробників KDE

6.4.1 Перевірки агента SSH

kdesrc-build може забезпечити вимикання інструмента агента SSH, якщо розробник KDE, який використовується SSH для доступу до сховища коду KDE, забуде його вимкнути. Таким чином, kdesrc-build може нескінченно очікувати на введення розробником пароля SSH, отже, типово, kdesrc-build перевіряє, чи запущено агент перед виконанням оновлення коду.

ПРИМІТКА

Ці дії виконуються лише для розробників KDE, які використовують SSH. Причиною цього є те, що для типового анонімного доступу для отримання паролів не потрібен. Subversion оброблятиме паролі для іншого можливого способу доступу для розробників KDE, `https`.

Ймовірно, ви захочете вимкнути перевірку агента SSH у випадку, коли kdesrc-build помилково визначає присутність агента. Щоб вимкнути перевірку агента, встановіть для параметра `disable-agent-check` значення `true`.

Вимикання перевірки SSH-агента:

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

6.5 Інші можливості kdesrc-build

6.5.1 Зміна рівня докладності виводу kdesrc-build

У kdesrc-build передбачено декілька параметрів для керування об'ємом даних, які виводить скрипт. У будь-якому випадку, помилки завжди виводяться.

- Запуск kdesrc-build з параметром `--quiet` (скорочена форма `-q`) виводить мінімум даних. Буде показано лише важливі повідомлення, попередження або повідомлення про помилки. Відомості щодо поступу збирання, у разі їх наявності, також буде показано.
- За використання параметра `--really-quiet` (не має скороченої форми) kdesrc-build під час виконання покаже лише важливі повідомлення і повідомлення про помилки.
- Використання параметра `--verbose` (скорочена форма `-v`) призведе до дуже детального виводу даних kdesrc-build.
- Параметр `--debug` призначено лише для усування вад, за його використання kdesrc-build працюватиме так, наче увімкнено `--verbose`, команди виводитимуть дані до терміналу, крім того, буде показано відомості, призначені для усування вад для багатьох функціональних можливостей.

6.5.2 Виведення у кольорі

Якщо ви запустите kdesrc-build з Konsole або іншої програми емуляції терміналу, скрипт виводитиме розфарбований текст.

Ви можете вимкнути це розфарбування за допомогою параметра `--no-color`, доданого до командного рядка або встановленням для параметра `colorful-output` у файлі налаштувань у значення `false`.

Вимикання виведення у кольорі у файлі налаштувань:

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

6.5.3 Вилучення непотрібних каталогів після збирання

У вас мало місця на диску, але велике бажання працювати у найсвіжішій версії KDE? kdesrc-build може допомогти вам зекономити на просторі на диску під час збирання KDE з Subversion.

ПРИМІТКА

Пам'ятайте, що для збирання KDE потрібно досить багато місця. Ось декілька з компонентів, які використовують багато місця на диску, коли ви користуєтеся kdesrc-build:

1. На самі отримані коди може бути витрачено досить багато місця на диску. На збірку типових модулів буде витрачено близько 1,6 гігабайтів місця на диску. Ви можете зменшити це значення, якщо збиратимете лише потрібні вам модулі. kdesrc-build не вилучатиме вихідні коди з диску, навіть якщо ви вилучите відповідний запис з [файла налаштувань](#), отже не забувайте вилучати непотрібні отримані коди з каталогу початкових кодів. Пам'ятайте про те, що файли кодів отримуються з інтернету, — вам *не слід* вилучати їх, якщо ви ними користуєтеся, принаймні, доки ви не припините користуватися kdesrc-build.

Крім того, якщо у вашій системі вже встановлено Qt™ з вашого дистрибутива (і вас все у ньому влаштовує), ймовірно, вам не потрібно встановлювати модуль qt. Таким чином ви можете зекономити близько 200 мегабайтів місця на диску.

Ще одна річ, пов'язана зі способом, у який працює Subversion: для кожного зі отриманих файлів зі сховища зберігається дві копії. kdesrc-build не може одразу після отримання намагатися мінімізувати розмір коду, — код ще просто не використано.

2. kdesrc-build створює окремий каталог для збирання початкового коду. Іноді скрипт kdesrc-build має скопіювати каталог кодів для того, щоб створити фіктивний каталог збирання. За такої потреби скрипт створює для економії місця символічні посилання, отже значної витрати місця на диску не буде. Каталог збирання, зазвичай, матиме набагато більші розміри за каталог початкових кодів модуля. Наприклад, каталог збирання kdatabase має розміри близько 1050 мегабайтів, а каталог початкових кодів kdatabase — лише близько 550 мегабайтів.

На щастя, після успішного збирання і встановлення модуля потреба у каталозі збирання зникає. kdesrc-build може автоматично вилучати каталог збирання після встановлення модуля, докладніше про це можна дізнатися з наведених нижче прикладів. Зауважте, що виконання цього кроку зробить неможливим kdesrc-build нарощувальне збирання, яке могло б заощадити ваш час, якщо ви пізніше забажаєте знову зібрати модуль.

3. Нарешті, для встановлення самого KDE поза межами каталогу збирання також потрібне місце на диску. Розмір цього місця, зазвичай, значно менший за розмір каталогу збирання, все ж точно співвідношення важко вказати.

Як зменшити вимоги щодо місця на диску для KDE? Одним зі способів зменшення цього місця є використання належних параметрів компіляції, вони нададуть вам змогу зменшити розмір програм за рахунок зменшення швидкості їх виконання. Іншим, набагато ефективнішим, способом є вилучення відомостей, призначених для усунення вад, з зібраного вами KDE.

ЗАСТЕРЕЖЕННЯ

Вам слід добре все зважити, перш ніж вилучати відомості, призначені для усунення вад. Робота з експериментальним програмним забезпеченням означає, що ви користуєтеся програмою, яка потенційно має більш ймовірність відмов, ніж програма зі стабільного випуску. Якщо ви користуєтеся програмами без відомостей, призначених для усунення вад, вам буде дуже важко створити гарний звіт про помилку, за яким легко буде усунути ваду. Ймовірно вам доведеться знову увімкнути відомості для усунення вад для відповідної програми і повторно її зібрати, щоб допомогти розробникові усунути причину відмови. Отже, вирішуйте самі, чи слід вилучати відомості для усунення вад!

Вилучення каталогу збирання після встановлення модуля. Каталог початкових кодів не вилучатиметься, до зібраних програм буде додано інформацію для усунення вад:

```
global
  configure -flags      --enable-debug
  remove-after-install builddir      # Вилучити каталог збирання після ↵
    встановлення
end global
```

Вилучення каталогу збирання після встановлення, без інформації для усунення вад, з оптимізацією за розміром.

```
global
  cxxflags      -Os      # Оптимізувати за розміром
  configure -flags      --disable-debug
  remove-after-install builddir      # Вилучити каталог збирання після ↵
    встановлення
end global
```

Розділ 7

СMake, система збирання KDE

7.1 Вступ до СMake

У березні 2006 року програма СMake виборола серед декількох інших програм право бути вибраною системою збирання для KDE 4, вона замінила систему, засновану на autotools, яку KDE використовувало спочатку.

Сторінку вступних відомостей щодо СMake можна знайти у [вікі спільноти KDE](#). У загальних рисах, замість запуску `make -f Makefile.cvs`, з наступним запуском `configure` і `Make`, слід просто запустити СMake, а потім `Make`.

У `kdesrc-build` передбачено підтримку СMake. Деякі з параметрів `kdesrc-build` насправді є параметрами самої системи збирання, зокрема `configure-flags` і `do-not-compile`. Якщо доступні рівнозначні можливості, скрипт надає вам їх у розпорядження. Наприклад, відповідником параметра `configure-flags` є параметр `cmake-options`, а параметр `do-not-compile` підтримується і для СMake, починаючи з версії `kdesrc-build 1.6.3`.

Розділ 8

Подяки і ліцензування

Переклад українською: Юрій Чорноіван yurchor@ukr.net

Цей документ поширюється за умов дотримання [GNU Free Documentation License](#).

Додаток А

Модулі KDE та структура початкових кодів

А.1 «Модуль»

У KDE програмне забезпечення згруповано за «модулями» різних розмірів. Спочатку таке групування передбачало декілька сховищ великих модулів, але із впровадженням заснованих на [Git сховищ початкових кодів](#) ці великі модулі було поділено на багато дрібніших модулів.

У kdesrc-build також використовується ця концепція модулів. Загалом кажучи, «модуль» — збірка коду, яку можна отримати, зібрати, перевірити та встановити.

А.1.1 Окремі модулі

Наказати kdesrc-build зібрати окремий модуль доволі просто. Нижче наведено код прикладу оголошення модуля, код якого зберігається у Subversion, у [файлі налаштувань](#).

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

ПІДКАЗКА

Цей модуль засновано на Subversion, оскільки тут не використовується параметр [repository](#). Крім того, для прикладу використано параметр `cmake-options`, насправді він не потрібен.

А.1.2 Групи пов'язаних модулів

Зараз, більшість модулів коду KDE зберігається у Git проекту KDE, і зазвичай поєднуються у групи модулів.

Через це у kdesrc-build передбачено підтримку груп модулів, за допомогою визначень [наборів модулів](#). Приклад:

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

ПІДКАЗКА

Ви можете, якщо хочете, вказати порожню назву набору модулів (*base-modules* у нашому прикладі). У такому разі параметр `repository` повідомить `kdesrc-build` про те, звідки слід отримати код, але ви також можете скористатися адресою `git://`.

Однією особливою можливістю використання «`repository kde-projects`» є те, що `kdesrc-build` автоматично включатиме всі модулі Git, які згруповано у вказаних вами модулях (у базі даних проєктів KDE).

A.1.3 Модуль «групи гілок»

Працюючи з концепцією **групи модулів**, розробники KDE зрештою виявили, що синхронізація у Git великої кількості гілок в усіх сховищах є доволі марудною справою, особливо у процесі розробки нових бібліотек KDE Frameworks для Qt™ 5.

Щоб уникнути зайвої роботи, було створено концепцію «груп гілок». Користувачі і розробники можуть вибрати одну певну групу, а скрипт автоматично вибере відповідні гілки у кожному зі сховищ Git.

У `kdesrc-build` передбачено підтримку цієї можливості з версії 1.16-pre2. Підтримку реалізовано за допомогою параметра `branch-group`.

Example A.1 Приклад використання `branch-group`

`branch-group` можна скористатися у файлі налаштувань так:

```
global
    # Вибрати KDE Frameworks 5 та інші програми, засновані на Qt5
    branch-group kf5-qt5

    # Інші загальні параметри ...
end global

module-set
    # branch-group працює лише для kde-projects
    repository kde-projects

    # branch-group успадковується з одного набору повсюдно, але
    # його можна вказати і тут.

    use-modules kdelibs kde-workspace
end module-set

# гілкою kdelibs буде "frameworks"
# гілкою kde-workspace буде "master" (станом на серпень 2013 року)
```

У цьому випадку одне значення `branch-group` дасть різні назви гілок для кожного з модулів Git.

Для роботи з цією можливістю потрібні певні дані, супровід яких здійснюють розробники KDE у сховищі Git з назвою `kde-build-metadata`. Втім, цей модуль буде автоматично включено `kdesrc-build` (хоча його і буде показано у даних, виведених скриптом).

ПІДКАЗКА

Для модулів KDE, для яких не встановлено назви гілки для вибраної вами групи гілок, типово буде вибрано відповідну назву гілки, ту, яку було б вибрано, якщо б ви не вказували параметра `branch-group`.

Додаток Б

Процедури налаштування профілю, які заміщуються програмою

Б.1 Налаштування профілю входу до системи KDE

Нижче наведено настанови щодо налаштування профілю, потрібного для забезпечення можливості входу до зібраної вами стільниці Плазми KDE. Зазвичай, kdesrc-build намагається зробити це у автоматичному режимі (див. Розділ 2.5.1). Дані у цьому додатку можуть бути корисним для тих, хто не може скористатися підтримкою налаштування входу до системи kdesrc-build. Втім, можливо, ці настанови застаріли, тому вам також варто ознайомитися з вмістом файла `kde-env-master.sh`, який є частиною початкових кодів kdesrc-build.

Б.1.1 Зміна ваших параметрів профілю запуску

ВАЖЛИВО

Файл `.bash_profile` є файлом параметрів входу до системи популярної оболонки `bash`, яка використовується у багатьох дистрибутивах Linux[®]. Якщо ви використовуєте іншу оболонку, можливо, вам доведеться скоригувати приклади, наведені у цьому розділі, так, щоб вони відповідали оболонці, яку ви використовуєте.

Відкрийте або створіть файл `.bash_profile` у домашньому каталозі за допомогою вашого улюбленого редактора, і додайте в кінець файла наступні рядки. Якщо ви збираєте модуль `qt` (це типова поведінка), додайте:

```
QTDIR=(шлях до qtdir) # Типовим є ~/kdesrc/build/qt.
KDEDIR=(шлях до kdedir) # Типовим є ~/kde.
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH
MANPATH=$QTDIR/doc/man:$MANPATH

# Дійте у відповідності до того, чи встановлено змінну LD_LIBRARY_PATH.
if [ -z $LD_LIBRARY_PATH ]; then
    LD_LIBRARY_PATH=$KDEDIR/lib:$QTDIR/lib
else
```


Підручник зі скрипту kdesrc-build

```
LD_LIBRARY_PATH=$KDEDIR/lib:$QTDIR/lib:$LD_LIBRARY_PATH
fi

export QTDIR KDEDIRS PATH MANPATH LD_LIBRARY_PATH
```

або, якщо ви не збираєте qt (а замість неї використовуєте системну бібліотеку Qt™), додайте такі рядки:

```
KDEDIR=(шлях до каталогу kde) # Наприклад, типово ~/kde.
KDEDIRS=$KDEDIR
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH

# Діяти відповідно, якщо ще не встановлено LD_LIBRARY_PATH.
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
```

Якщо ви не використовуєте призначеного користувача, встановіть іншу змінну середовища \$KDEHOME для вашого нового середовища у вашому `.bash_profile`:

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

# Створити каталог, якщо це потрібно
[ ! -e ~/.kde-svn ] && mkdir ~/.kde-svn
```

ПРИМІТКА

Якщо пізніше ваше меню KDE буде порожнім, або переповненим записами програм з вашого дистрибутива, можливо, вам доведеться встановити змінну середовища XDG у вашому файлі `.bash_profile`:

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

Б.1.2 Запуск KDE

Тепер, коли ви налаштували параметри вашого середовища на використання належної версії KDE, вам також слід переконаватися, що ви використовуєте правильний скрипт `startkde`.

Відкрийте текстовий файл `.xinitrc` з вашого домашнього каталогу або за потреби створіть його. Додайте до файла такий рядок:

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

ВАЖЛИВО

У деяких дистрибутивах, ймовірно, вам доведеться виконати певні дії з файлом `.xsession`, який також знаходиться у вашому домашньому каталозі. Такі дії знадобляться, якщо ви використовуєте графічні інструменти керування входом до системи, такими як `kdm`, `gdm` або `xdm`.

Тепер запустіть вашу свіжу версію KDE: у системах BSD і Linux[®] з підтримкою віртуальних терміналів натискання комбінацій клавіш **Ctrl+Alt+F1 ... Ctrl+Alt+F12** використовується для перемикання до віртуальних консолей з номерами від 1 до 12. Таким чином, ви можете запускати декілька стільничних середовищ одночасно. Перші шість терміналів є текстовими, а наступні шість — графічними.

Якщо після запуску вашого комп'ютера ви бачите інструмент графічного входу до системи, ви можете скористатися новим середовищем KDE, навіть якщо його пункту немає у списку. У більшості інструментів входу до системи, зокрема у `kdm`, є пункт, призначений для «Не-типового сеансу». За використання цього пункту параметри вашого сеансу буде завантажено з файла `.xsession` у вашому домашньому каталозі. Якщо ви вже змінили цей файл у спосіб, описаний вище, використання цього пункту завантажить ваш новий встановлений KDE.

Якщо це не так, ви можете спробувати інший спосіб, який, зазвичай, працює: натисніть комбінацію клавіш **Ctrl+Alt+F2** і ви побачите текстовий термінал. Введіть ваші реєстраційні дані вашого призначеного користувача і введіть команду:

```
startx -- :1
```

ПІДКАЗКА

Ви можете запускати KDE, зібраний з вихідних кодів, і ваш старий KDE одночасно! Увійдіть до системи за допомогою реєстраційних даних вашого звичайного користувача, запустіть стабільну версію стільниці KDE. Натисніть комбінацію клавіш **Ctrl+Alt+F2** (або **F1, F3** тощо), — ви побачите текстовий термінал. Увійдіть до системи від імені вашого призначеного користувача Subversion KDE і введіть команду:

```
startx -- :1
```

Ви можете повернутися до стільниці KDE вашого звичайного користувача натисканням клавіатурного скорочення у вже запущеній стільниці. Цією комбінацією, зазвичай, є **Ctrl+Alt+F7**, можливо, у вашому випадку останньою клавішею буде **F6** або **F8**. Щоб повернутися до вашої зібраної `kdesrc-build` версії KDE, ви можете скористатися подібною комбінацією клавіш, але у ній останньою буде інша функціональна клавіша. Наприклад, якщо вам потрібно було ввести **Ctrl+Alt+F7**, щоб перемкнутися на вашу звичайну версію KDE, вам доведеться ввести **Ctrl+Alt+F8**, щоб повернутися до вашого зібраного `kdesrc-build` KDE.