



EBook Gratuito

APPENDIMENTO

SCons

Free unaffiliated eBook created from
Stack Overflow contributors.

#scons

Sommario

Di.....	1
Capitolo 1: Iniziare con SCons.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Iniziare.....	2
Capitolo 2: C ++.....	4
Examples.....	4
Una semplice costruzione.....	4
Specifica di varie opzioni di costruzione.....	4
Capitolo 3: Ottenere SCons in esecuzione.....	5
introduzione.....	5
Examples.....	5
Installazione su Linux.....	5
Installazione su Windows.....	5
In esecuzione dalla fonte.....	5
Installazione con pip Python.....	6
Capitolo 4: SCons esegue le fasi.....	7
introduzione.....	7
Examples.....	7
Ispezionando le fasi di SCons.....	7
Titoli di coda.....	8

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [scons](#)

It is an unofficial and free SCons ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official SCons.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con SCons

Osservazioni

SCons è un sistema di costruzione. Ci vuole un sacco di file di input e strumenti su di essi per produrre output. SCons è scritto in puro Python, funziona allo stesso modo su Linux, Windows e OS X e può essere eseguito senza installazione.

I file SConstruct di `sConstruct` sono script Python con comandi incorporati che creano un `build tree`. SCons esegue il processo di compilazione in fasi. Prima sta leggendo i file e costruendo un albero di costruzione. Il secondo è attraversare l'albero per costruire i file di destinazione.

Versioni

Versione	Data di rilascio
2.5.1	2017/11/03

Examples

Iniziare

Dopo aver [avviato SCons](#), crea un file chiamato `sConstruct`:

```
print('..Building World')
```

Ora esegui `scons`:

```
$ scons
scons: Reading SConscript files ...
..Building World
scons: done reading SConscript files.
scons: Building targets ...
scons: `.` is up to date.
scons: done building targets.
```

`sConstruct` è uno script Python con funzioni SCons aggiuntive.

```
Zip('archive', ['SConstruct'])
```

Lo script sopra si inserisce in `archive.zip` usando la funzione `Zip()` fornita da SCons. `Zip` è un **Builder** - costruisce il **target** specificato dal primo argomento da più **fonti**, che viene come secondo argomento ai Costruttori per convenzione.

SCons I builder iniziano con lettere maiuscole e operano su oggetti **Environment**, che

memorizza la configurazione di build. SCons fornisce **Ambiente** predefinito, ma può essere creato esplicitamente per separare le variabili di costruzione, scegliere diversi strumenti, ecc.

```
env = Environment()
env.Zip('archive', ['SConstruct'])
```

Si noti che quando si esegue lo script per la seconda volta, non viene generato nulla. SCons ricostruisce gli obiettivi solo quando cambiano i file di origine. Modifica `SConstruct` ed esegui `scons nuovo` per vedere la differenza.

SCons è progettato per essere estensibile. Aggiungi i tuoi metodi Builder allegandoli all'Ambiente, che può essere trattato in argomenti successivi.

Leggi Iniziare con SCons online: <https://riptutorial.com/it/scons/topic/4780/iniziare-con-scons>

Capitolo 2: C ++

Examples

Una semplice costruzione

È molto semplice creare un semplice progetto C ++. Ecco un esempio di un file di `sConstruct` che lo fa:

```
env=Environment ()  
  
env.Program('hello', Glob('src/*.cpp'))
```

Questo crea l'eseguibile `hello` composto da tutte le fonti in `src` con estensione `cpp`.

Specifica di varie opzioni di costruzione

Questo esempio mostra impostazioni di build più dettagliate:

```
env=Environment (  
    CPPPATH='/usr/include/boost/',  
    CPPDEFINES=['foo'],  
    LIBS=['bar'],  
    SCONS_CXX_STANDARD='c++11')  
  
env.Program('hello', Glob('src/*.cpp'))
```

Questo crea l'eseguibile `hello` da tutti i file `cpp` in `src`, con le seguenti impostazioni:

- Il percorso di ricerca è ``/usr/include/boost'`
- Il `FOO` costante è definito
- I collegamenti eseguibili con la `bar`
- C ++ 11 è usato come standard

Leggi C ++ online: <https://riptutorial.com/it/scons/topic/6158/c-plusplus>

Capitolo 3: Ottenere SCons in esecuzione

introduzione

SCons è scritto in Python 2 e non ha bisogno di alcuna dipendenza per funzionare. Puoi semplicemente copiare i suoi script nell'albero dei sorgenti del progetto ed eseguire da qui. O potresti voler usare la versione pacchettizzata per il tuo sistema operativo.

Examples

Installazione su Linux

Su Debian o Ubuntu, puoi installare SCons usando

```
$ sudo apt-get install scons
```

Sui sistemi basati su [YUM](#), usare

```
$ sudo yum install scons
```

È possibile installare utilizzando un [RPM](#) scaricandolo, quindi eseguendo

```
$ sudo rpm -Uvh http://prdownloads.sourceforge.net/scons/scons-2.5.0-1.noarch.rpm
```

Installazione su Windows

Prendi il programma di installazione da <http://scons.org/pages/download.html>

Oppure prova lo strumento di installazione di `pip` fornito con Python:

```
pip install scons
```

Se non è possibile trovare `scons` dopo questo, assicurati che Python `Scripts/` folder sia aggiunto a `PATH` per l'installazione di Python.

In esecuzione dalla fonte

Se hai delle modifiche da condividere o vuoi semplicemente provare una nuova versione in sviluppo.

```
$ hg clone https://bitbucket.org/scons/scons
$ python scons/src/script/scons.py
```

Installazione con pip Python

```
pip install scon
```

Se non si desidera eseguire `scons` dalla riga di comando, verificare che la directory degli script Python sia stata aggiunta a PATH per l'installazione.

Se vuoi giocare con API, `import SCons` da Python non funzionerà, perché SCons 2.5.xe sotto consente di installare più versioni side-by-side. Questo era necessario per passare tra diverse versioni di SCons durante lo sviluppo e la risoluzione dei problemi. Ora il modo più comune per farlo è usare `virtualenv` o semplicemente [eseguirlo dal sorgente](#) .

Leggi [Ottenere SCons in esecuzione online](https://riptutorial.com/it/scons/topic/9377/ottenere-scons-in-esecuzione): <https://riptutorial.com/it/scons/topic/9377/ottenere-scons-in-esecuzione>

Capitolo 4: SCons esegue le fasi

introduzione

SCons è un sistema di costruzione multi-step. Per prima cosa legge tutti gli `SConstruct` e `SConscript` per eseguire il codice Python e creare un `build graph` con obiettivi. Quindi esegue la scansione del filesystem per rilevare quali target del `build graph` devono essere aggiornati, dopodiché esegue il comando per creare target obsoleti.

Examples

Ispezionando le fasi di SCons

`scons` descrive le fasi in esecuzione. Eseguendolo su un `SConstruct` vuoto `SConstruct` ottiene questo:

```
$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
scons: `.` is up to date.
scons: done building targets.
```

Per sopprimere i messaggi di fase, aggiungere l'opzione `-Q`. `--tree=all` permette di vedere l'albero delle dipendenze per il target attuale che `scons` costruito durante la costruzione.

```
$ scons -Q --tree=all
scons: `.` is up to date.
+-.
+-SConstruct
```

`.` è il target predefinito, che significa "build `SConstruct` nella directory corrente". `SConstruct` è quindi una dipendenza per la costruzione del target predefinito.

Leggi SCons esegue le fasi online: <https://riptutorial.com/it/scons/topic/10170/scons-esegue-le-fasi>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con SCons	Ami Tavory , anatoly techtonik , bdbaddog , Community
2	C ++	Ami Tavory
3	Ottenere SCons in esecuzione	Ami Tavory , anatoly techtonik
4	SCons esegue le fasi	anatoly techtonik