

Inhalt

1. Hintergründe zu SCons
2. Installation von SCons
3. Die Datei "SConstruct"
4. Erstellen von Programmen
 1. Keyword Arguments
 2. Mehrere Quelldateien
3. Erstellen von Bibliotheken
4. Einbinden von Bibliotheken
5. Abhängigkeiten erkennen
6. Auf Abhängigkeiten testen
 1. Construction Environments
 2. Auf Bibliotheken/Header/Funktionen/typedefs testen
3. Zum Schluss

1. Hintergründe zu SCons

SCons ist ein Open Source Build-System, welches in [Python](#) implementiert wurde und seine Wurzeln im auf Perl basierenden Cons hat. Bei der Entwicklung von SCons wurde auf Korrektheit, Geschwindigkeit und besondere Einfachheit Wert gelegt. Wie Python selbst ist SCons sehr einfach zu benutzen, sowohl für einfache Aufgaben, als auch für komplizierte Builds.

Um SCons zu benutzen, braucht es nur minimale Python-Kenntnisse. Erwerben kann man diese mit dem Tutorial auf der Python-Homepage.

SCons kann sowohl für C bzw. C++ als auch für Java verwendet werden. In diesem Artikel werde ich allerdings nur auf C++ eingehen und die Java-Anhänger auf das offizielle [Tutorial](#) verweisen.

Da es von den Herstellern von SCons bereits ein sehr gutes [Tutorial](#) gibt, werde ich bloß auf die grundlegenden Funktionen eingehen und diese relativ schnell abhandeln.

2. Installation von SCons

Geschrieben von:

Sonntag, den 12. Februar 2006 um 20:10 Uhr - Aktualisiert Sonntag, den 19. August 2007 um 22:13 Uhr

SCons ist, da auf Python basierend, plattformunabhängig. Um mit SCons zu arbeiten, muss Python installiert sein. Man kann entweder die vorkompilierten Pakete installieren oder es selbst kompilieren. Die unten erwähnten Pakete kann man in der Download-Section von <http://www.scons.org/> finden.

Für Windows gibt es, wie gewohnt, einen Installer, der wohl keiner weiteren Erklärung bedarf. ;)

Auf auf RPM basierenden Systemen geht die Installation relativ simpel:

```
rpm -uvh scons-x.x-x.noarch.rpm
```

Auch bei auf Debian basierenden Systemen ist es nicht sehr schwer:

```
apt-get install scons
```

Wer SCons aus den Quellen übersetzen möchte, lädt sich den Tarball (oder die Zip-Datei) herunter, entpackt sie und wechselt ins scons Verzeichnis. Dort einfach

```
python setup.py install
```

eingeben und SCons wird in die üblichen Verzeichnisse (/usr/local/) installiert. Dies erfordert allerdings Administratorrechte; besitzt man diese nicht, so kann man folgendermaßen ein anderes Verzeichnis angeben:

```
python setup.py install --prefix=$HOME
```

3. Die Datei "SConstruct"

Diese Datei kann man in etwa mit einem Makefile oder einer Ant-XML-Datei vergleichen, denn in ihr stehen die Tasks und Targets, die SCons dann ausführt. Um die Aufgaben zu beschreiben, wird Python verwendet, d.h., dass diese Datei eigentlich ein Python-Skript ist. Im Unterschied zu Python-Skripten wird diese Datei aber nicht unbedingt geordnet abgearbeitet (siehe unten), sondern so, wie es die Erfüllung der Targets erfordert. Die "SConstruct" sollte man an der Wurzel im Projektverzeichnis ablegen.

4. Erstellen von Programmen

Um zu zeigen, wie man mit SCons Programme erstellen kann, werden wir das allseits beliebte "Hallo Welt"-Beispiel verwenden:

Das Build-System SCons

Geschrieben von:

Sonntag, den 12. Februar 2006 um 20:10 Uhr - Aktualisiert Sonntag, den 19. August 2007 um 22:13 Uhr

```
[code xml:lang="cpp"]// hello_world.cpp #include <iostream>
int main(int argc, char **argv) {
    std::cout
```