

IDRIS Workshop - Installation

Carsten König

Devopenspace Leipzig 2016

Idris

Windows

Ich empfehle die **Binaries** zu benutzen.

Die Dateien durch Aufruf und am Besten den Hauptpfad (in dem die `idris.exe` liegt) in die `PATH` Umgebungsvariable aufnehmen.

Ein Aufruf von `idris` aus der Eingabeaufforderung sollte dann die REPL liefern:

```
C:\>idris
```

```

      /---\_/--(\)---
     / //   / /---/
    _// // / / / ( )
   /---\_,/_/ /_----/

Version 0.12.3
http://www.idris-lang.org/
Type :? for help
```

Idris is free software with ABSOLUTELY NO WARRANTY.

For details type :warranty.

```
Idris>
```

MacOS

Leider kann ich diese Option nicht testen, es gibt aber [ein Binaries-pkg hier](#) und die Installation sollte der Windows-Variante ähnlich sein.

Ansonsten sollte die Linux Variante über **stack** auch funktionieren.

Linux

Sofern ihr **stack (Haskell)** installiert habt (falls nicht bitte tun), funktioniert das normalerweise über **LTS Haskell 6.18** ganz gut.

Dazu einfach außerhalb irgendeines Projekts `stack install` verwenden.

Ich hatte ein einziges Problem mit der *dependency* auf `trifecta(==1.6.*)` das ich durch gleichzeitige Installation:

```
stack install trifecta.1.6 idris.0.12.3
```

beheben konnte.

Mit dem aktuellen LTS Haskell 7.1 (ghc-8.0.1) Snapshot sollte `idris.0.12.3` (die zum 23.09 aktuelle Version) direkt gehen - das habe ich allerdings noch nicht versucht.

Falls ihr nicht genau wisst, wo ihr den Resolver einstellt - der steht in einer Datei `stack.yaml` in eurem *stack project-root*. Wo dieses Verzeichnis liegt bekommt ihr einfach heraus, indem ihr `stack path` aufruft und nach `project-root` (recht weit oben) sucht.

Die Datei sieht üblicherweise so aus:

```
# This is the implicit global project's config file, which is only used when
# 'stack' is run outside of a real project. Settings here do _not_ act as
# defaults for all projects. To change stack's default settings, edit
# 'D:\Stack\config.yaml' instead.
#
# For more information about stack's configuration, see
# http://docs.haskellstack.org/en/stable/yaml_configuration/
#
flags: {}
extra-package-dbs: []
packages: []
extra-deps: []
resolver: lts-6.18
```

wobei ihr ganz unten den `resolver` überschreiben könnt.

Bei üblicher `stack` Installation sollten die entsprechenden Dateien automatisch in einem `PATH` Verzeichnis gelandet sein (im `local-bin` das ihr ebenfalls über `stack path` finden könnt).

Die Dokumentation im `idris-lang/Idris-dev/wiki` auf Github scheint hier etwas veraltet zu sein - ich würde den Weg über `cabal install` jedenfalls nicht empfehlen.

Atom

Das *Package* hat den Namen `language-idris` und sollte nach Installation direkt funktionieren. Falls nicht bitte checken, ob ausführbare `idris` bzw. `idris.exe` Datei im Pfad liegt oder deren Pfad in den *Settings* des Atom-Packages unter `Idris Location` eintragen.

Emacs

Einfach den `idris-mode` über `package-install` von MELPA installieren.