ATSPackage user manual

Vanessa McHale

March 2, 2018

Contents

1 Introduction

ATSpackage is a build tool for ATS written in Haskell. There are three things it accomplishes:

1. Distributed builds. ATSPackage allows users depend on libraries that are hosted elsewhere.
2. Simplified builds. As ATSPackage contains scripts to download the compiler, builds are easier for potential contributors.
3. Haskell integration. ATSPackage has first-class support for building ATS code that depends on Haskell libraries.

With that in mind, it is worthwhile to enumerate some things that it does *not* accomplish:

1. Full flexibility of C. As ATSPackage is intended to simplify builds, it does not expose everything. This will likely not cause problems, provided that the libraries dependend on are written in C, ATS, or Haskell.

2 Package Resolution

As ATS is a statically typed language, some form of dependency resolution is necessary if we'd like to be able to share data structures between packages.

3 Builds

ATSPackage supports three build types: binary, dynamic library, and static library.

3.1 Binary Builds

ATSPackage allows

3.2 Binary Builds with Haskell Dependencies

ATSPackage allows binary builds with Haskell dependencies by allowing a package to depend on an object file generated by GHC. The object file can be generated by cabal, allowing full use of the Haskell ecosystem.

ATSPackage can also generate data types for ATS based on Haskell types. You can use this to eliminate some of the work involved in writing FFI bindings, and particularly to avoid ATS' lack of generics.