(Analysis) Release Building and Distribution

Attila Krasznahorkay

Disclaimer

- Not meant as a technical discussion about the build procedure
  - Alex will talk about that in the afternoon
- Meant as an overview of
  - How we provide (analysis) code to the users
  - How we could think to do it in the future

Analysis Releases

- One of the big successes of the Run 2 analysis model
- Providing centrally managed releases of all the combined performance/trigger software packages needed to analyse ATLAS data
- Comes in two flavours:
  - AnalysisBase
    - O(100) packages built using RootCore
    - Built for `x86_64-slc6-gcc48-opt` and `x86_64-mac1010-clang70-opt`
  - AthAnalysisBase
    - O(300) packages built using CMT
    - Built just for `x86_64-slc6-gcc48-opt`

Release Distribution

- Done exclusively on CVMFS

  - Allows us to precisely/easily control the externals used

- Local installation of AnalysisBase supported, by making it easy to download all packages from SVN, and build them using RootCore

  - Only requires a C++11 compiler and ROOT 6 to be present

  - Works reasonably well for non-novice users

```
cd ATLAS/sw/releases/
rcSetupLocal -b Base,2.3.34
```

- Building AthAnalysisBase by hand is not supported at the moment at all

  - Even though some users would have been interested in it

CMake

- Want to migrate all software building infrastructure to use CMake for release 21.

- ~On track for this

  - The full offline release build is being migrated in a dedicated migration nightly http://atlas-nightlies-browser.cern.ch/~platinum/nightlies/info?tp=g&nightly=CMAKE

  - The same code can be used to build AnalysisBase releases with some effort

    - Convenient utilities are sorely missing just yet

- Packages will have a single `CMakeLists.txt` file, describing their build in all release environments

  - Hopefully making it easier to pull in analysis code developers into offline projects

External Software

- O(50) packages picked up from <u>LCG releases</u>
  - Only available for SLC6 and CC7, with GCC 4.8 or GCC 4.9
  - Even for these platforms not trivial to install from a central repository
  - Is it expected that SFT will ever provide LCG for other platforms? They did provide some Mac builds in the (distant) past…
- O(10) additional externals packaged with offline releases specifically for ATLAS
  - Will become part of the AtlasExternals project's build to make release distribution easier
- RootCore already behaves similar to how the CMake builds should later on
  - Can check if an external is available on the build system already
  - If not, it downloads and builds the external itself
    - Works for everything beside ROOT itself

Binary Distributions

- CMake/CPack helps with creating binary installers in many flavours
  - Simple .tgz files
  - RPM/DEB linux packages
  - MacOS X installers
  - Windows installers
- We should provide such things if possible
  - Simplify the installation of analysis releases for novice users
  - Save on CPU time necessary for building the release on every analyser's laptop
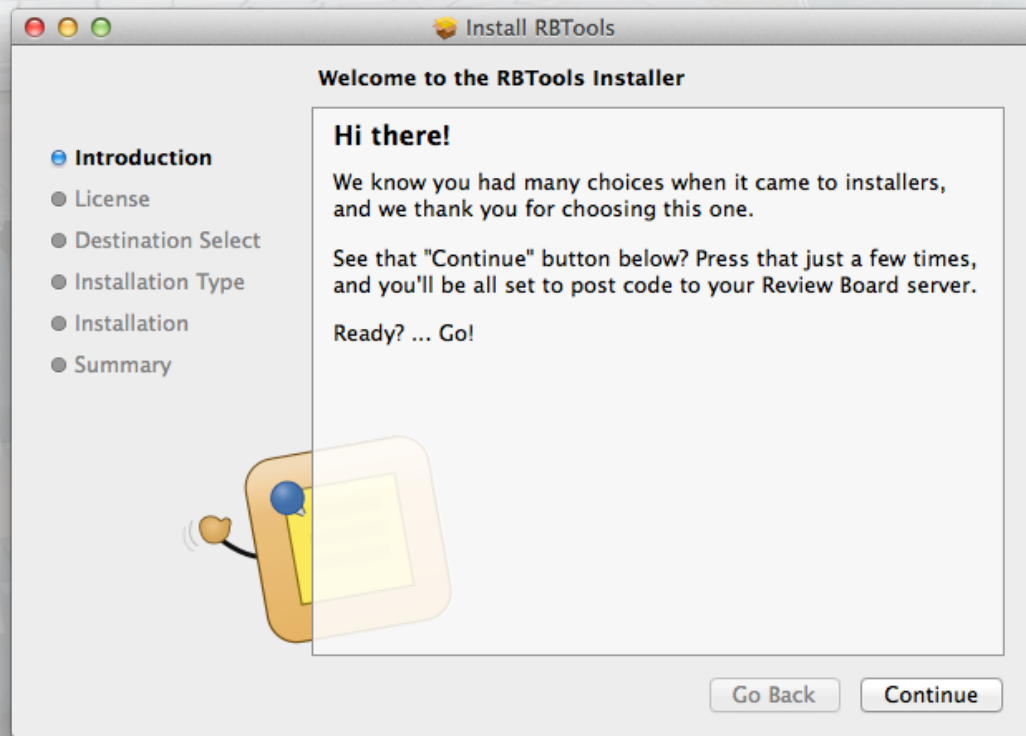
Binary Installers

- For linux provide RPMs/DEBs for:
  - All externals used in the build
  - A single RPM/DEB for the analysis release
  - Installing everything under a fixed directory, like
    - /atlas/analysisbase/3.2.3/
    - /atlas/analysisbase/externals/ROOT/v6-08-10/
    - /atlas/athanalysisbase/3.2.3/
    - /atlas/athanalysisbase/externals/LCG_90/
    - …

- For MacOS X provide graphical installers that installs all these things. Under let's say:
  - /Applications/ATLAS/AnalysisBase/3.2.3/
  - /Applications/ATLAS/Externals/ROOT/v6-08-10/
  - …
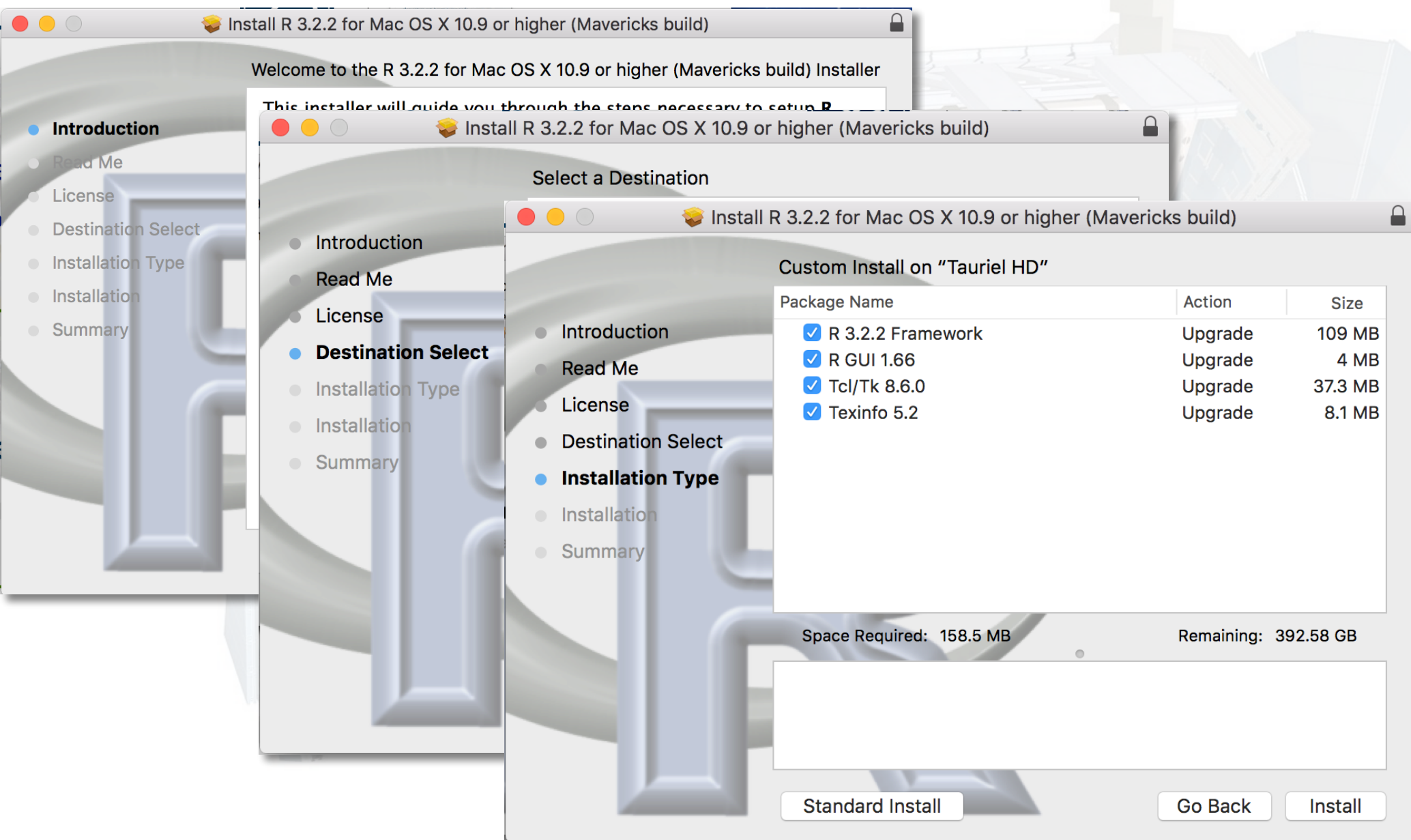
A Good Example
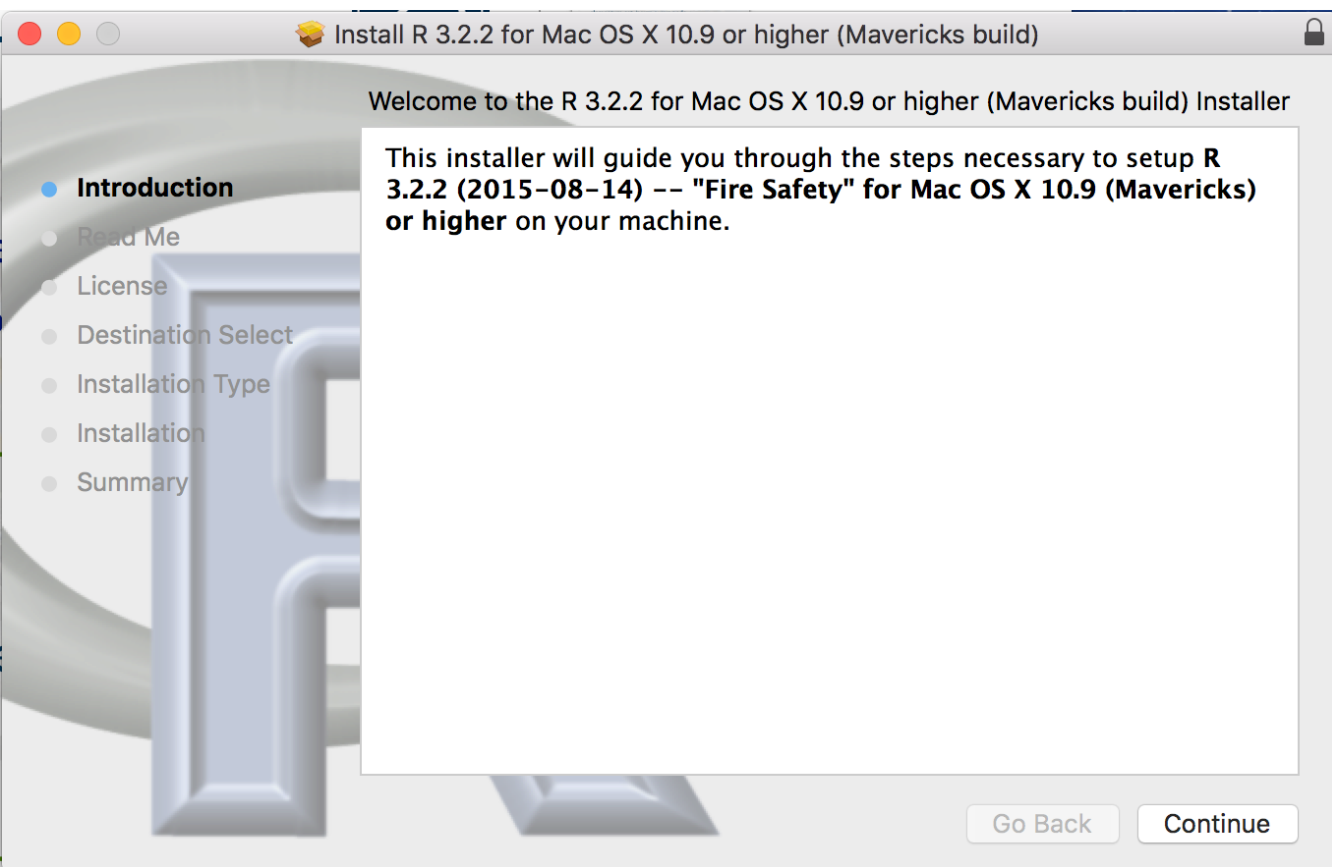
A Good Example
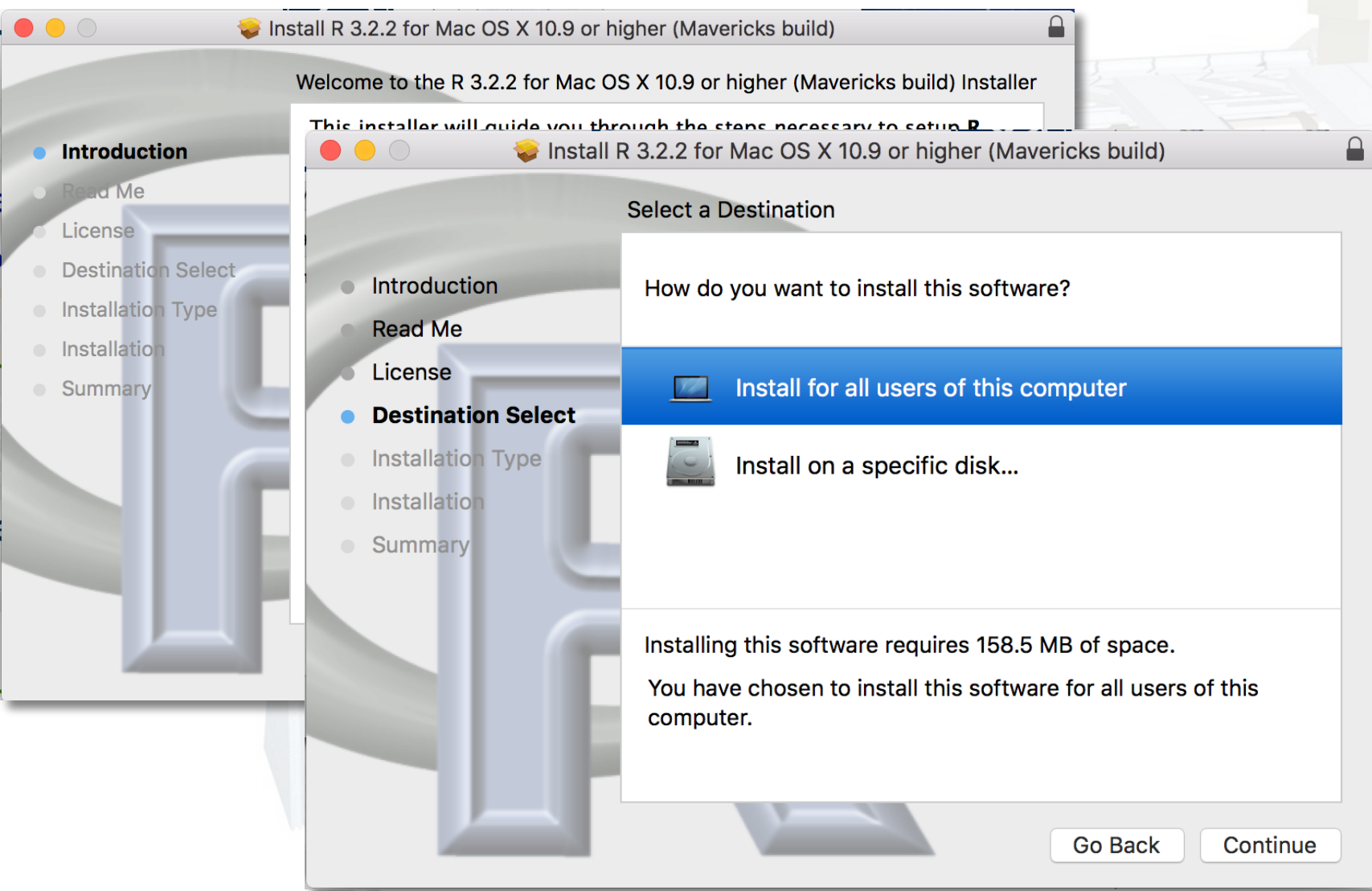
A Good Example

Summary

- Installing externals should definitely be made as simple as installing any software natively on the platform
  - RPM/DEB packages on linux, .pkg files on MacOS X
- If at all possible, analysis/offline releases should be installable in the same way
- With some simple rules for relative paths between projects this is not too difficult to set up