

SCons Command Line Installer

The command line installer for the SCons Release Manager is located in the [SCons-installer package](#) in the CVS repository. It is written in python and uses the same database interface as the GUI installer.

Purpose: Provide a command line interface to download and install release packages build by the SCons release manager.

Options and parameters

-h, --help - print a short help message

-c, --command - the command to execute

-o, --os - the operating system to get the package for

-p, --package - the package to work with (ScienceTools, GlastRelease, etc)

-v, --variant - the variant of the built to get (Debug or Optimized)

-s, --version - the version to get (08-15-00, 3104, etc)

-t, --versionType - the type of build to get (LATEST, HEAD, RELEASE)

-e, --extDir - the location where external libraries are to be installed. Defaults to \$GLAST_EXT

-i, --instDir - the location where the packages are to be installed. Default is the local directory.

-x, --extOnly - only install the external libraries

-k, --pkgOnly - only install the packages, don't install the external libraries

-d, --distribution - Which distribution for the selected package to install (user, developer or source)

-r, --pretend - don't actually install anything, just list the actions that would have been taken

-f, --keepFiles - do not delete downloaded files once installation is complete

Option descriptions

-h, --help - this parameter causes a short help message listing all available options to be printed to the screen

-c, --command <action> - This is the command to be executed by the script. The format is derived from the current CMT command line installer. Available possibilities for the value of <action> are:

- install
- listOSes
- listPackages
- listVariants
- listVersions
- listVersionTypes
- listDistributions

with the exception of the install action, all of these actions can be provided without any additional parameters. In that situation, a list of all the possible values for the option specified is given. However, not all of the values may be available for the specific case the user is interested in. It is much more advisable to provide the necessary qualifying options to get specific information. If not used without any options each of the actions require the following additional options to be provided:

Action	Required Options
listPackages	- none -
listVersionTypes	--package
listVersions	--package, --versionType
listVariants	--package, --versionType, --version
listOSes	--package, --versionType, --version, --variant
listDistributions	--package, --versionType, --version, --variant, --os
install	--package, --versionType, --version, --variant, --os, --distribution

-p, --package <name> - The package to work with (ScienceTools, GlastRelease, etc). This option is required for all later commands if specific information is desired or a package is to be installed. The current possible values for <name> are:

- ScienceTools
- GlastRelease
- CHS

-t, --versionType <type> - The type of build you want to work with. Current possible values for <type> are:

- Integration (can also use LATEST)
- Release Candidate (can also use HEAD)
- Release

-s, --version <version> - The version of the software to get. This is the numeric part of the tag. For release builds it is of the form XX-YY-ZZ and for HEAD and LATEST builds it is simply a number

-v, --variant <variant> - The variant of the built to get. You only have two choices here: Debug and Optimized

-o, --os <osname> - This option specifies the operating system we want the software for. Since the release manager builds for a variety of different operating systems you need to specify the one you want. The exact OS choices is strongly dependent of the package, versionType, version and variant choices. Possible values of <osname> are:

- redhat4-i686-32bit-gcc34
- redhat4-x86_64-64bit-gcc34
- redhat5-i686-32bit-gcc41
- redhat3-i686-32bit-gcc32
- tiger-i386-32bit-gcc40
- Windows-i386-32bit-vc71
- redhat5-x86_64-64bit-gcc41

-d, --distribution <distName> - Which distribution for the selected package to install. <distName> can have three possible values

- **source** - This installs just the source code for the distribution. Use this if you want to build the software yourself.
- **user** - This installs all the binaries for the distribution. This is distribution for end users that just want to run the software
- **developer** - This installs everything, the source code and the binaries and is intended for use by those working to actively develop the software

This parameter is optional. If not present it defaults to user

-e, --extDir <dirName> - the location where external libraries are to be installed. Defaults to \$GLAST_EXT and then the local directory if \$GLAST_EXT is not defined.

-i, --instDir <dirName> - the location where the packages are to be installed. Defaults to \$INST_DIR and then the local directory if \$INST_DIR is not defined.

-x, --extOnly - only install the external libraries, don't install the tools

-k, --pkgOnly - only install the packages, don't install the external libraries

-r, --pretend - don't actually install anything, just list the actions that would have been taken. This only affects the install command Giving this option causes everything to happen except for the actual downloading and installation of the files. The installer will still perform all the check to make sure the selected installation package is available.

-f, --keepFiles - do not delete downloaded files once installation is complete. The default is to remove each installation file after the package has been installed. If this option is give, the files will remain on disk. They are downloaded to the /tmp directory.