
The SCons reST tool

Dirk Baechle

2012-04-25

Table of Contents

1. Basics	1
1.1. Install	1
1.2. How to activate	1
1.3. Requirements	1
2. Processing documents	2
3. All builders	2
4. Need more?	2

1. Basics

This tool tries to make working with docutils in SCons a little easier. It provides several toolchains for creating different output formats, like HTML or LaTeX, from a reST (reStructuredText) input file.

1.1. Install

Installing it, requires you to copy (or, even better: checkout) the contents of the package's `rest` folder to

1. `"/path_to_your_project/site_scons/site_tools/rest"`, if you need the reST Tool in one project only, or
2. `"~/scons/site_scons/site_tools/rest"`, for a system-wide installation under your current login.

For more infos about this, please refer to

- the SCons User's Guide, chap. 19.7 "Where to put your custom Builders and Tools" and
- the SCons Tools Wiki page at <http://scons.org/wiki/ToolsIndex>.

1.2. How to activate

For activating the tool "rest", you have to add its name to the Environment constructor, like this

```
env = Environment(tools=['rest'])
```

On its startup, the reST tool tries to find an installed version of the `docutils` package by looking for the command-line tool `rst2html`. So make sure that it is added to your system's environment `PATH` and can be called directly, without specifying its full path.

1.3. Requirements

For the most basic processing of reST to HTML/LaTeX/ODT, you need to have installed

- the Python `docutils` package, and optionally

- the `pygment` module for syntax highlighting in code blocks.

2. Processing documents

Creating a HTML or LaTeX document is very simple and straightforward. Say

```
env = Environment(tools=['rest'])
env.Rst2Html('manual.html', 'manual.rst')
env.Rst2Latex('manual.ltx', 'manual.rst')
```

to get both outputs from your source file `manual.rst`. As a shortcut, you can give the stem of the filenames alone, like this:

```
env = Environment(tools=['rest'])
env.Rst2Html('manual')
env.Rst2Latex('manual')
```

and get the same result. Target and source lists are also supported:

```
env = Environment(tools=['rest'])
env.Rst2Html(['manual.html', 'reference.html'], ['manual.rst', 'reference.rst'])
```

or even

```
env = Environment(tools=['rest'])
env.Rst2Html(['manual', 'reference'])
```

Important

Whenever you leave out the list of sources, you may not specify a file extension! The Tool uses the given names as file stems, and adds the suffixes for target and source files accordingly.

The rules given above are valid for all the Builders `Rst2Html`, `Rst2Latex` and `Rst2Odt`.

3. All builders

A simple list of all builders currently available:

Rst2Html	Single HTML file.
Rst2Latex	Outputs a LaTeX file.
Rst2Odt	OpenOffice output.

4. Need more?

This work is still in a very basic state and hasn't been tested well with things like variant dirs, yet. There will definitely arise the need for adding features, or a variable. Let us know if you can think of a nice improvement or have worked on a bugfix/patch with success. Enter your issues at the Bitbucket bug tracker for the ReST Tool, or write to the User General Discussion list of SCons at users@scons.tigris.org.