

Remote Development with EMACS

James E. Prewett

January 18, 2009

Remote Editing with TRAMP

Remote REPL with SWANK
ssh tunnels

Remote Editing with TRAMP

TRAMP allows us to *transparently* edit remote files!

- ▶ TRAMP: 'Transparent Remote (file) Access, Multiple Protocol'
- ▶ <http://www.gnu.org/software/tramp/>
- ▶ uses:
 - ▶ rsh, rcp
 - OR
 - ▶ ssh, scp

Set up TRAMP for SSH/SCP & edit a remote file

in your .emacs file:

- ▶ (require 'tramp)
- ▶ (setq tramp-default-method "scp")

to edit your .emacs file on the remote machine, simply open the file: */hostname:path/to/file*

Remote REPL with SWANK

SWANK is a part of SLIME.

On the remote machine, in a running LISP image:

- ▶ Load SWANK: `(load "/path/to/swank-loader.lisp")`
- ▶ Start SWANK: `(swank:create-server :dont-close t :port 4005)`
- ▶ Needed for SSH tunneling:
`(setf swank:*use-dedicated-output-stream* nil)`

connect-to-host helper function

```
(defvar *current-tramp-path* nil)
(defun connect-to-host (path)
  (setq *current-tramp-path* path)
  (setq slime-translate-from-lisp-filename-function
    (lambda (f)
      (concat *current-tramp-path* f)))
  (setq slime-translate-to-lisp-filename-function
    (lambda (f)
      (substring f (length *current-tramp-path*)))))
  (slime-connect "localhost" 4005))
(defvar *noctool-tramp-path*
  "/ssh:root@noctool..unm.edu:")
(defun noctool-slime ()
  (interactive)
  (connect-to-host *noctool-tramp-path*))
```

setting up an ssh tunnel

```
ssh <hostname> -L [bind_address:]port:host:hostport] \  
    user@host
```

eg:

```
ssh -L localhost:4005:localhost:4005 root@noctool
```

connect to the REPL!

... then M-x noctool-slime