

**fish**

**Jörg Walter  
Brad Hards**



fish

fish

## Contents

## fish

Allows you to access another computer's files using the Secure *SHell* (SSH) protocol. The remote computer needs to be running the SSH daemon, but the remainder of the protocol uses standard commandline tools as discussed below.

You can use the fish kiosklike like this: **fish:// hostname** or **fish:// username@hostname**.

### NOTE

You need to use double forward slashes.

You can omit the *username* (and the trailing @ symbol) if you have the same username on both computers.

You can add a password in the format: **fish:// username:password@hostname** but it is not necessary as you will be prompted for one if it is not supplied.

If you are running the SSH daemon on a non-standard port, you can specify that port using the normal URL syntax as shown below: **fish:// hostname:portnumber**.

Fish should work with any roughly POSIX compatible UNIX<sup>®</sup> based remote computer. It uses the shell commands **cat**, **chgrp**, **chmod**, **chown**, **cp**, **dd**, **env**, **expr**, **grep**, **ls**, **mkdir**, **mv**, **rm**, **rmdir**, **sed**, and **wc**. Fish starts **/bin/sh** as its shell and expects it to be a Bourne shell (or compatible, like **bash**). If the **sed** and **file** commands are available, as well as a `/etc/apache/magic` file with MIME type signatures, these will be used to guess MIME types.

If Perl is available on the remote machine, it will be used instead. Then only **env** and **/bin/sh** are needed. Using Perl has the additional benefit of being faster.

Fish may even work on Windows<sup>®</sup> machines, if tools like Cygwin are installed. All the above utilities must be in the system `PATH`, and the initial shell must be able to process the command **echo FISH:;/bin/sh** correctly.