

Help and Documentation Utilities

Version 6.0

February 18, 2014

This section describes utilities designed to connect to documentation and to support bug reports. See also [scribble/xref](#).

1 Searching in the Documentation

```
(require help/search)      package: scribble-lib

(send-main-page [#:sub sub
                 #:notify notify
                 #:fragment fragment
                 #:query query]) → any
sub : path-string? = "index.html"
notify : (-> path? void) = void
fragment : (or/c #f string?) = #f
query : (or/c #f string?) = #f
```

Visits the documentation file *sub* in the user's browser.

This function builds a URL that points into the main collection documentation or into the user-specific documentation, depending on the *sub* argument. Once it finds the path, `send-main-page` passes the path to *notify*. The *fragment* and *query* arguments are passed to `send-url/file`, along with the URL.

```
(perform-search str [context]) → void?
str : string?
context : (or/c #f
           string?
           (list/c string? string?)) = #f
```

Searches for *str* in the documentation. The *context* argument supplies a context for the search or, if it is two strings, a context for the search and a label for that context.

2 Connecting to racket

```
(require help/help-utils)    package: racket-doc
```

The `help/help-utils` library is dynamically loaded by the `help` form that is available by default in racket.

```
(search-for strs) → void?  
  strs : (listof string?)
```

Calls `perform-search` after concatenating the elements of `strs` and adding spaces between them.

```
(find-help/lib id lib) → void?  
  id : symbol?  
  lib : module-path?
```

Visits the documentation page for `id` as an export of `lib`.

```
(find-help id) → void?  
  id : identifier?
```

Visits the documentation for `id`.

```
(go-to-main-page) → void?
```

Visits the main entry page for the documentation.