I. twidge Manual

twidge

<jgoerzen@complete.org>

John Goerzen

Name

`twidge` — Microblogging client for Twitter, Identica

Synopsis

twidge [-d] [-c *FILE*] *command* [*command_args*]

twidge --help

Description

twidge is a client for microblogging sites such as Twitter[1] and Identica[2] (identi.ca). Microblogging sites let you post short one-paragraph updates, follow the updates that your friends post, and interact with everyone in the site in a conversation style.

twidge is a client to make working with microblogging sites faster and easier. It is extremely versatile, and can be customized to work the way you want to work, and combined with other tools to do just about anything.

twidge can be used quite nicely interactively from the shell. It is useful directly as-is, and with simple shell aliases can make a highly efficient system to do exactly what you want. It is perfectly capable of being your only client for microblogging.

twidge also can be used in an automated way, via **cron(1)**, or it can even integrate with your email system.

A full list of twidge features, along with numerous suggestions on how to use it, can be found at the twidge website at http://software.complete.org/twidge.

But, if you'd like to get going RIGHT NOW, skip to the Quick Start section below. For now, I'll summarize a few twidge features; a more comprehensive list is on the website.

Feature List

- Convenient, easy to learn, and fast command-line interface (it's simple to do simple things, and advanced things are possible)

[1] http://twitter.com/
[2] http://identi.ca/

- This extensive manual, and numerous examples on the website
- Compatible with any microblogging service that implements the Twitter API. Tested with Twitter and Identica (identi.ca). Should be compatible with any other system.
- Full support for reading the activity of your friends, replies to you, and your own activity.
- Optional capability to remember what you have seen already and suppress those updates in future runs.
- Optional automatic shortening of long URLs via is.gd
- Optional integration with your email system -- send and receive updates via email.
- Specifically designed to be friendly to use in shell scripts.
- Shell scripting makes it easy to do many things, such as scheduling future updates (with **at**), ignoring certain updates (with **grep**), etc.
- Easy use of multiple accounts by having multiple configuration profiles. Mechanism lends itself to shell aliases and tools.
- Small, minimalist footprint
- Output formats are easily parsed. Input formats are easily generated. Configuration file format is liberal and easy to generate.
- Robust error detection and recovery throughout.

Quick Start

This section will describe how a first-time twidge user can get up and running quickly. It assumes you already have twidge compiled or installed on your system. If not, please follow the instructions in the `INSTALL` file in the source distribution.

To get started, simply run **twidge setup** at your shell prompt. twidge will lead you through the first-time configuration -- which is very quick and completely self-explanatory!

Now, let's start exploring. **twidge lsrecent -su** will show you the 20 most recent updates from the people you follow. After the first time, it will remember what you've seen and only show you new updates. In place of **lsrecent**, you could substitue **lsreplies** to replies to you. You can also run **twidge lsrecent --help**, or refer to this document, for more details on the command. In short, -s tells the system to save what you've seen, and -u tells it to only show you unseen items.

You can subscribe to updates from friends by running **twidge follow *`nick`***. To subscribe to updates for twidge itself, you'd run **twidge follow unixtwidge**. If you've had enough of updates from someone, just use the **twidge unfollow** command.

Now, how about posting your own updates? Just type **twidge update**. Type your update, and press Enter when done. Your update may scroll past the end of the screen; don't worry about it. Just don't hit Enter until you're done. You can also pass your update on the command line, taking care to quote it if it contains spaces or shell characters.

That's the quick tour. For many more examples, refer to the website.

Options

twidge always is invoked with the name of a specific operation, such as `update` or `lsrecent`. In twidge, these operations are called *commands*. Each command has its own options, which are given after the command on the twidge command line. A full summary of each command's options is given later in this manual.

You may obtain a list of all commands with **twidge lscommands**. Help is available for any individual command with **twidge *`command`* --help**. Global help is available with **twidge --help**.

Global Options

These options may be specified *before* any command.

-c *`FILE`*
--config=*`FILE`*

> By default, twidge uses `~/.twidgerc` for configuration. With this option, you can specify an alternate file. twidge will read configuration information from it. In certain cases, twidge will also write to it, such as with the **twidge setup** command, or the -s option to one of the ls commands.

-d
--debug

> Enables debugging output. This verbose output helps you learn what twidge is doing every step of the way and diagnose any problems you may encounter.

Commands in Twidge

twidge has many different commands. You must specify a command when using twidge. This section will discuss each command in detail. Note that all commands are case-sensitive and should be *given in lowercase*.

All commands support the option `--help`. Running **twidge *`command`* --help** will display information about the command and its options. Since all commands support this, it won't be explicitly listed for each command below.

Commands to Display Updates

twidge ls* [-a] [-l] [-s] [-u]

Several commands can display updates from yourself or others. They all share a common syntax and common set of options. The commands are **lsarchive**, **lsdm**, **lsdmarchive**, **lsrecent**, **lsreplies**, **lsrt**, **lsrtarchive**, and **lsrtreplies**. Here are the common options:

-a
--all

Normally, microblogging sites return one "page" of results -- typically, 20 or 100 results, depending on the operation. Specifying --all requests twidge to continually send requests to the microblogging site until it receives an entire set of information. Normally this is a bad idea, but when used in combination with -u -- especially if you have used -s before -- it can be a useful way to poll for all new updates.

If not used carefully, this command can generate many requests to the server and can use a significant amount of bandwidth. Microblogging sites have traffic limits, and will not take kindly to this. Be careful when you use --all.

-e `command`
--exec=`command`

Causes any normal output to be suppressed. Instead, for each item that is retrieved, call the given command. The command will be passed exactly four arguments. Care should be exercised when using them in shell scripts due to the likelihood of the presence of shell metacharacters. The arguments are:

1. The update ID for the given update.
2. The username that created the update
3. A suggested Message-ID for representing the update in email. Contains the enclosing angle brackets.
4. The update text itself. Likely contains spaces and other shell metacharacters.

-l
--long

The default output format shows a nickname and a message, formatted nicely for display. Using this option causes output to include these columns: message ID, sender nick, recipient nick (empty unless a direct message), update text, and date created. The output format is suitable for machine parsing. horizontal tab character \t separates the columns, and is guaranteed to appear nowhere else. It is also guaranteed that \n will appear nowhere within a line when using -l.

The columns as presented are anticipated to remain stable. However, additional columns could be added to the right of these columns in future versions of twidge. Well-behaved parsers should be prepared to discard any surplus columns to the right of those specified in this document.

-m `address`
--mailto=`address`

Causes any normal output to be suppressed. Instead, for each item that is retrieved, generate an email and send it to the given address. The subject line will contain the sender nickname and the first 30 characters of the text. The body will contain the entire text. The message is sent by using the program listed as sendmail in the configuration file. If mailfrom is listed in the configuration file, it is listed as the from address and the name of the sender is listed as the from comment.

A Message-ID designed to uniquely identify this update is generated and included. In addition, headers beginning with X-Twidge- will be added, with various metadata.

It should be noted that a simple script could be used with --exec to achieve this same purpose, but with greater flexibility. The functionality is built into twidge as well for convenience.

-s
--saveid

Saves the ID of the most recent fetched message for future use with -u. Requires write access to your configuration file.

-u
--unseen

Causes the command to show only items since the last use of -s with this particular command. If -s has never before been used with this command, has no effect.

Some of these options don't make sense together; for instance, --long, --exec, and --mailto are mutually exclusive. Results are undefined if you use such options simultaneously.

lsarchive

Lists your own posts. With the one `lsarchive` option, you can also list the posts of a specific other user:

-U *`USERNAME`*
--username=*`USERNAME`*

Instead of showing your own updates, instead show those of a differerent user.

lsdm

Lists direct messages sent to you. Note: identi.ca does not support direct messages, and this will cause an error with that service.

lsdmarchive

Lists direct messages you have sent. Note: identi.ca does not support direct messages, and this command will cause an error with that service.

lsrecent

Lists recent posts made by you or the people you follow.

lsreplies

Lists recent replies made by anyone on the microblogging site to you.

lsrt

Lists new-style retweets made by those that you follow.

Note: identi.ca doesn't support new-style retweets and will return an error on this command.

lsrtarchive

Lists the new-style retweets you made yourself.

Note: identi.ca doesn't support new-style retweets and will return an error on this command.

lsrtreplies

List retweets of your statuses made by others.

Note: identi.ca doesn't support new-style retweets and will return an error on this command.

Commands to Display Other Information

These commands honor --all and --long like the other commands, but do not honor --saveid or --unseen, as these options make no sense.

lsfollowers

twidge lsfollowers [*options*] [*username*]

Will slow a list of people following you. If given an optional username, shows the people that follow that user. If given --long, shows the IDs for each person. You may need to use --all to get a complete list, but again this may generate significant traffic.

lsfollowing

twidge lsfollowing [*options*] [*username*]

Shows a list of the people you follow. Arguments and actions are the same as with lsfollowers.

Action Commands

These commands perform an action on the server.

dmsend

twidge dmsend *recipient* [*status*]

Causes twidge to send a new direct message to the designated recipient. If a status is given on the command line, it must be presented as a single argument to twidge by the use of shell quoting if needed. If no status is given on the command line, a single line will be read from stdin and used as the status.

The maximum length of updates on various microblogging sites is 140 characters. Twidge will abort with an error if your update attempt exceeds that length.

By default, Twidge will attempt to shorten URLs in your updates via the is.gd service. You can disable this by setting shortenurls = no in the [DEFAULT] or [dmsend] section of your configuration file.

follow

twidge follow *username*

This command will cause twidge to request that the microblogging site add the given username to the list of people you follow. It is considered an error to attempt to add someone you already follow.

setup

twidge setup

Enters the interactive first-time setup tool.

unfollow

twidge unfollow *username*

This command will cause twidge to request that the microblogging site remove the given username from the list of people you follow. It is considered an error to attempt to remove someone you do not follow.

update

twidge update [*status*]

twidge update -r

Causes twidge to post a new status to the server. If a status is given on the command line, it must be presented as a single argument to twidge by the use of shell quoting if needed. If no status is given on the command line, a single line will be read from stdin and used as the status.

The maximum length of updates on various microblogging sites is 140 characters. Twidge will abort with an error if your update attempt exceeds that length.

By default, Twidge will attempt to shorten URLs in your updates via the TinyURL.com service. You can disable this by setting shortenurls = no in the [DEFAULT] or [update] section of your configuration file.

When -r is given, twidge expects to read an RFC2822 email message in stdin. The body of the message will be used as the content of the update. The References header of the message will be briefly scanned, and if appropriate, the reply-to attribute of the update will reflect the Message-ID referred to therein. This way, if you use **twidge lsrecent --mailto** to email recent items to you, then use your email reply button to reply to them, twidge will link the two on Twitter. This linkage can be done even if you omit @reply, but will likely confuse many people because most other clients can't do this. You should still include @reply.

It should be noted that, in an effort to minimize size and complexity, twidge has an extremely simple email parser. You should send it only plain text messages. Do not use HTML, RTF, attachments, or anything that would cause MIME headers to appear in the RFC2822 body. You probably want to turn off your signature as well. twidge will convert newline characters to spaces when processing your message body.

Other Command

lscommands

twidge lscommands

This command will display a list of all available twidge commands along with a brief description of each.

twidge Configuration File

twidge has a configuration file in which you can set various options. This file normally lives under `~/.twidgerc`.

The configuration file has multiple sections. Each section has a name and is introduced with the name in brackets. Each section has one or more options.

The section named DEFAULT is special in that it provides defaults that will be used whenever an option can't be found under a different section. If you specify no section names, DEFAULT is the assumed section. Some items, such as urlbase, are assumed to be in DEFAULT.

Let's start by looking at an example file, and then proceed to examine all the options that are available.

```
[DEFAULT]

; If your password contains a percent sign (%), list it twice (%%)

; Path to server API interface -- no trailing slash
urlbase = https://twitter.com
```

```
shortenurls = yes

; Last IDs seen by different commands.  Written to with -s.

[lsarchive]
lastid = 917383569

[lsrecent]
lastid = 914881241
```

Whenever twidge looks for information about a particular option, it first checks to see if it can find that option in a section for that option. If not, it checks the `DEFAULT` section. If it still doesn't find an answer, it consults its built-in defaults.

General Options

These are specified in the `DEFAULT` section.

urlbase

> The URL to access the API of the microblogging site. The default, `https://twitter.com` is for Twitter. To use Identica, you would specify `http://identi.ca/api`. Do not put a trailing slash on this option.

oauthrequesttoken

> The URL to access the oAuth request token interface. The default, `%(urlbase)s/oauth/request_token`, will work with most environments.

oauthaccesstoken

> The oAuth access token URL. Default is `%(urlbase)s/oauth/access_token`.

oauthauthorize

> The oAuth authorize URL. Default is `%(urlbase)s/oauth/authorize`.

oauthconsumerkey

> The oAuth consumer key. Twidge is registered with Twitter and identi.ca and will supply a reasonable default for those two based on the content of urlbase.

oauthconsumersecret

> The oAuth consumer secret. A default is provided as with oauthconsumerkey.

oauthdata

> Automatically written by **twidge setup**. Do not alter.

Per-Command Options

These options may be specified in `DEFAULT` or in a per-option section. If placed in `DEFAULT`, they will apply to all relevant options unless overridden.

lastid

Stores the last ID seen for commands that support -s or -u.

mailfrom

Gives the email address to use for the From: header on messages generated by twidge. Normally specified in [DEFAULT] so as to impact all commands. If not given, twidge specifies no From: line, leaving the system's mail transport to assign a default one.

Should contain a bare email address only -- no brackets, parens, quotes, or comments.

When given, Twidge uses the nickname and this address for the From line. When not given, the nickname is prepended to the Subject line.

savelast

For one of the "ls" class of commands, implies -s on every invocation, rather than require it to be manually given. This option need only be present; the value you give it doesn't mater.

sendmail

Stores the path to the sendmail executable on your system, used for sending mail. Normally specified in [DEFAULT] so as to impact all commands. Defaults to /usr/sbin/sendmail.

shortenurls

You may specify `shortenurls` in the `[update]` section. It defaults to yes. If set to no, will cause twidge to omit its attempt to shorten URLs in your updates via `tinyurl.com`.

URL Shortening Options

To enable the bit.ly or j.ump URL shorteners, you must add a `[bitly]` or `[jmp]` section to the configuration file. This should contain two entries: `login` and `apikey` as found in http://bit.ly/account/. For example:

```
[jmp]
login: bitlyapidemo
apikey: R_0da49e0a9118ff35f52f629d2d71bf07
```

Aliases

You can add an `[alias]` section to the config file which will effectively create new commands. For example:

```
[alias]
recent: lsrecent -u
replies: lsreplies -u
```

Tips & Hints

Here are a few tips and hints to make twidge more pleasant for you.

Going Through a Proxy

If your connections must go through a proxy, you'll need to set an environment variable. If you use an environment variable, your settings may will also impact other applications -- and that's probably what you want. See the Environment section later for tips on doing that.

Environment

twidge does not read any environment variables directly. However, it does pass on the environment to the programs it calls, such as Curl. This can be useful for specifying proxies. Please see curl(1) for more details.

Conforming To

- The Extensible Markup Language (XML)[3] standard (W3C)

Copyright

If the GPL is unacceptable for your uses, please e-mail me; alternative terms can be negotiated for your project.

[3] http://www.w3.org/XML/

Author

twidge, its modules, documentation, executables, and all included files, except where noted, was written by John Goerzen <`jgoerzen@complete.org`> and copyright is held as stated in the COPYRIGHT section.

See Also

curl(1)

The twidge homepage at http://software.complete.org/twidge.