Troy's Home Page : slrn Howto

I have spent hours tinkering and customising slrn, my favourite newsreader. Here are some tips and tricks I have picked up along the way. There's nothing difficult about it. If I can do it, so can you. I've provided code so you can hack to suit your purposes or even modify for other scripts.

Tables of Contents:

Other howtos yet to be completed:

What is slrn?

slrn is a console mode newsreader, which is a program that reads articles on Usenet newsgroups. It is highly customisable, has excellent scoring capability, and complies well with the Good NetKeeping Seal of Approval.

Macros can be written using the S-Lang macro language. S-Lang is by the original author of slrn, John E Davis.

Back to top

How to use macros.

They are called from $HOME/.slrnrc by adding the line
`interpret "macro_name.sl"`

You can either put all your macros in one file, just adding more and more as your write/collect them. Or you can put them all in separate files as I do. I find this makes them easier to manage with them in separate files based on their functionality.

The sequence of macros is important. Make sure you don't double up on function names or key bindings.

Here is my current $HOME/.slrnrc :

```
% Filename: $HOME/.slrnrc
% Author: Troy Piggins

% .slrn is a directory in $HOME

% "include" is used to extend the slrnrc file format.

% troy.sl contains variables/settings that I have changed
include   ".slrn/troy.sl"

% I keep my colour settings separate
% colors.sl is slrn's standard attribute's colours
include   ".slrn/colors.sl"
% score-color.sl is special for Rudy Taraschi's score col
include   ".slrn/score-color.sl"

% "interpret" is used to interpret s-lang macros.

% third party macros:
interpret ".slrn/onekey-score.sl"
interpret ".slrn/t-prot.sl"
interpret ".slrn/t-prot-cfg"
interpret ".slrn/group_sort.sl"
interpret ".slrn/get-by-mid.sl"
interpret ".slrn/rtfFAQ-1.01.sl"
interpret ".slrn/notify_postponed.sl"
interpret ".slrn/stickytags-1.72.sl"
interpret ".slrn/savex.sl"
% http://strg-alt-entf.org/slrn.html
%interpret ".slrn/slrnface

% my macros:
interpret ".slrn/display_filter.sl"
interpret ".slrn/create_msg_id.sl"
interpret ".slrn/macros.sl"
interpret ".slrn/demime.sl"
interpret ".slrn/view_manuals.sl"
interpret ".slrn/view_my_posts.sl"
interpret ".slrn/fetchnews.sl"

% put key bindings last to ensure any macro bindings occu
% defined
include   ".slrn/binds.sl"
```

Back to top

How to view my saved posts.

slrn has no inbuilt way of viewing posts saved to file, or posts you've sent or replies. Not a problem with a simple macro that uses mutt to view them. Just hit the (editable) key defined in the macro and mutt opens up showing all of your posts, quit mutt and you are back in slrn.

Download the macro view_my_posts.sl and interpret it.

```
% Filename: view_my_posts.sl
% Version: 0.9.1
% Author: Troy Piggins
```

```
% assumes you are have mutt mail reader
% check the paths to "save_posts" and "save_replies" are

define view_my_posts() {

% path to your posts, ie the value of "save_posts" in you
  variable postpath= "$HOME/news/My_Posts";

  () = system ("mutt -f "+postpath);

}

definekey ("view_my_posts", "^T", "article");
definekey ("view_my_posts", "^T", "group");

define view_my_mails() {

% path to your mails, ie the value of "save_replies" in y
  variable mailpath= "$HOME/news/My_Replies";

  () = system ("mutt -f "+mailpath);

}

definekey ("view_my_mails", "^R", "article");
definekey ("view_my_mails", "^R", "group");
```

Back to top

How to read the slrn manuals and references from within slrn.

Updated 9/9/07 version 0.9.3 - can now specify user-defined pagers instead of hard-coded to 'less'.

Another simple macro to view slrns manuals, man pages, or any other reference docs you like from within slrn via a menu that looks like this:

Download the macro view_manuals.sl and interpret it.

```
% Filename: view_manuals.sl
% Version: 0.9.3
% Author: Troy Piggins

% 0.9.3 - added user defineable pager
% 0.9.2 - added slang help files
% 0.9.1 - initial release

define view_manuals() {

% set the path to your favourite pager here
  variable pager="/usr/bin/less";
%  variable pager="/usr/local/share/vim/vim70/macros/less
%  variable pager="/usr/local/bin/most";

% be sure to check the path to your slrn docs with traili
% default is /usr/local/share/doc/slrn/
```

```
  variable docpath="/usr/local/share/doc/slrn/";

  variable man=select_list_box( "slrn docs", "man page",
    "slrn manual", "score help", "slrnfuns help",
    "slang help", "slangfuns help", 6, 1);

  switch (man)
    { case "man page" : system( "man slrn");}
    { case "slrn manual" : system( pager+" "+docpath+"man
    { case "score help" : system( pager+" "+docpath+"scor
    { case "slrnfuns help" : system( pager+" "+docpath+"s
    { case "slang help" : system( pager+" /usr/share/doc/
    { case "slangfuns help" : system( pager+" /usr/share/
    { message ("Error in doc selection");}

}

definekey ( "view_manuals", "<f1>", "article");
definekey ( "view_manuals", "<f1>", "group");
```

Back to top

How to create a custom Message-ID header.

By default you will get a Message-ID similar to slrnfkhn0n.ovr.*username@posting_host*. I like to create my own and here's how I do it.

```
% Filename: create_msg_id.sl
% Version: 0.9.2
% Author: Troy Piggins
% Thanks: Sven Guckes from the slrn-user mailing list, an
% the slang-user mailing list

% returns a string of the format YYYYMMDDhhmmss.xyz@subdo
% eg for my setup I get:
% Message-ID: 20070220143000.xyz@usenet.piggo.com
% where xyz are random characters
% you need to change the value of sd and hostname to suit

% Pseudo-random function provided by John E Davis on the
% list, because from what I can tell at the time of writi
% way of generating random numbers in s-lang without the
% people won't have.  Possible in slang 2.1.4.

private variable Random;
define srandom (r)
{
   Random = r;
}
srandom (_time() * getpid());
define random ()
{
  Random = (Random*69069U + 1013904243U)&0xFFFFFFFFU;
  return Random;
}
% end pseudo-random function

define create_msg_id()
{

   variable sd= "usenet";
   variable hostname= "piggo.com";
```

```
    variable tm= localtime( _time ());

% some random chars for "true uniqueness" as recommended
%   in

    variable rnd=  random() mod 1000;

    return sprintf ("%d%02d%02d%02d%02d%02d.%d@%s.%s",
                    tm.tm_year + 1900, tm.tm_mon + 1, tm.t
                    tm.tm_hour, tm.tm_min, tm.tm_sec,
                    rnd,
                    sd, hostname);

}

define post_hook ()
{
  set_string_variable ( "custom_headers",
                        sprintf ("Message-ID: <%s>", creat
}

define followup_hook()
{

  set_string_variable ("followup_custom_headers",
                        sprintf ("Message-ID: <%s>", creat
}

define reply_hook()
{
  set_string_variable ("reply_custom_headers",
                        sprintf ("Message-ID: <%s>", creat
}
```

Download the create_msg_id.sl macro and interpret it. Be careful if you already have, or intend to have, other post_hooks etc. You will need to comment out the ones in this script and call multiple functions somewhere common to all macros using them.

Back to top

How to create more than one custom header.

You can see from the custom Message-ID above how to add custom headers. Trouble is you can only use the command once. Adding more custom headers is as simple as adding a '\n' between them. I use:

```
define post_hook ()
{
  set_string_variable ( "custom_headers",
    sprintf ("Message-ID: <%s>\nX-NNTP-Proxy: Leafnode-2
}
```

to get a "Message-ID:" header followed by a "X-NNTP-Proxy:" header.

Back to top

How to have different settings/identities for different groups.

No - this is not about anonymity. No - this is not for deception. I always post under my real name in technical groups. Because I play online games, I post under my gaming alias "PiG$$" in games groups. My ISP's news policy always includes my IP address, my Message-IDs and other headers are always a similar format, and I'm not interested in hiding my identity. Enough about what this

isn't for, and on with what it is for.

Want different signatures for different groups? Want a different Reply-To address for different groups? Here's how I do it.

[still being edited]

Back to top

How to handle PGP and/or multipart MIME attachments.

slrn has very few negatives. Not correctly handling MIME attachments is one of them, but fortunately there is a workaround.

Firstly you need to download something to handle the MIME parts. One solution is demime, a perl script designed to convert MIME messages into plain text. It does this by unconditionally removing all attachments leaving plain text only. At the time of writing this (7 Feb 07) version 1.1d was the latest and has been since late 2003. There is a patch to update this to version 1.1e which I recommend. For some bizarre reason the author hasn't provided 1.1e as full download, but the patched version works fine for my purposes. I've saved a patched copy of demime.1.1e and you can use that if you like but I take no responsibility for it. I saved this to /usr/local/bin/demime.1.1e, and also made a symbolic link to it called /usr/local/bin/demime.

Alternatively you could use unmime.py, a python script written by Grant Edwards from the slrn-user mailing list. Treat it the same way as demime.

Next step is to download the demime.sl macro and interpret it. Note that if you use unmime.py instead of demime, you need to change the line

```
   fp = popen ("/usr/local/bin/demime --quiet - <
"+demime_tmp, "r");
```

to

```
   fp = popen ("/usr/local/bin/unmime.py - < "+demime_tmp,
"r");
```

Final step is to register the macro. Simplest way would be to just add or uncomment the line

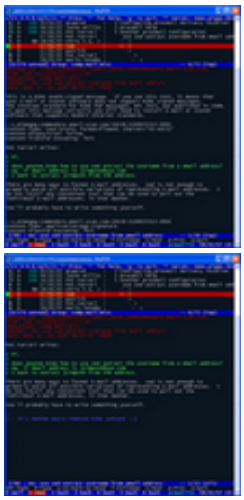
```
() = register_hook ("read_article_hook", "demime");
```

to the end of demime.sl after all the other code.

I do it a little more complicated than that, because I use more than one display filter and like to be able to turn them on and off. See the section on display filters to see how I handle that.

Here are some screenshots showing a MIME multipart html message before and after.

And here is a MIME PGP signed message before and after.



Back to top

How to create a custom display filter.

I spend a lot of time reading email and news, so I want it to be easy on the eye. You can see I don't like seeing MIME PGP and HTML messages. Extending that method I use a custom display filter that can pipe the raw message through any program. My current setup uses sed only, but I have even piped posts through spamassassin just to see if it worked to handle USENET spam. It did.

Download the display_filter.sl macro and interpret it. This file does not actually do the filtering, it merely pipes the raw message through the display filter slrn_display_filter and passes the filtered message back to slrn.

You need to register the macro. Simplest way would be to just add or uncomment the line
`() = register_hook ("read_article_hook", "display_filter");`
to the end of display_filter.sl after all the other code.

Because I use more than one display filter and like to be able to turn them on and off, I actually use something like this to register several at once:

```
define register_display_filters () {
  () = register_hook ("read_article_hook", "demime");
  () = register_hook ("read_article_hook", "t_prot");
  () = register_hook ("read_article_hook", "display_filte
  () = register_hook ("read_article_hook", "show_xface");
}

define unregister_display_filters () {
  () = unregister_hook ("read_article_hook", "demime");
  () = unregister_hook ("read_article_hook", "t_prot");
  () = unregister_hook ("read_article_hook", "display_fil
}

% display filters are on by default
register_display_filters ();

definekey ("register_display_filters", "\e6", "article");
definekey ("unregister_display_filters", "\e7", "article"
```

Download the sample slrn_display_filter script and save it somewhere in your $PATH.

You can use the filter's sed part to make almost and changes you like. To include something like spamassassin, you could edit the filter thus:

```
spamc         \
| sed -e '

# fix incorrect sig delimiters

s/^--$/-- /

# ...

'
```

There you have it. Let your imagination run wild.

Back to top

How to colourise certain words in the body of a message.

I have long been missing a feature in slrn to colourise parts of the body of a message. Other than quoted text, URLs, signatures, and emphasized text there isn't a feature that enables that.

So I thought to myself "Why not use the emphasized text colouring feature in conjunction with the display filter?" And it works. Of course you are restricted to only the 3 colours for bold, italics, and underline text.

For example, to colourise email addresses and the word *slrn* in the body of a message, add this to slrn_display_filter above:

```
sed -e '

# fix incorrect sig delimiters

s/^--$/-- /

# underline each side of matched text - this case email a

s/[$a-zA-Z0-9.-]*@[a-zA-Z0-9.-]*/_&_/g

# puts italics chars each side of matched text

s/slrn/\/&\//

'
```

Email addresses are the underlined colour and the word *slrn* will appear the italics colour in messages, although the original message is not actually changed. Hope this makes sense.

Back to top

How to run system/shell commands like slrnpull or fetchnews for local news spools/servers.

If you run a local news spool or cache, you probably have a cron job set up to fetch new articles at set periods like this for slrnpull:

```
  @hourly sudo /usr/local/bin/slrnpull -h newsserver
@weekly sudo /usr/local/bin/slrnpull --expire
```

or perhaps this for leafnode:

```
  */15 * * * * sudo /usr/local/sbin/fetchnews
@daily sudo /usr/local/sbin/texpire
```

Sometimes it is convenient to check for new articles intermediately. This can be done simply with the following macro fetchnews.sl . Just hit 'G' to fetch

new articles, or 'g' to post only if you are using Leafnode or Leafnode 2.

```
% Filename: fetchnews.sl
% Version: 0.9.1
% Author: Troy Piggins

define fetchnews () {

% Uncomment following line if using slrnpull.  Check path
%  () = system ("sudo /usr/local/bin/slrnpull -h newsserv

% following line is for leafnode
  () = system ("sudo /usr/local/sbin/fetchnews -vvv");

  call("refresh_groups");
}

define post_only ()
{

% following line is for leafnode
  () = system ("sudo /usr/local/sbin/fetchnews -vvv -P");

}

definekey ("fetchnews", "G", "group");
definekey ("post_only", "g", "group");
```

Back to top

How to get over the UTF-8 bad display problem.

I recently downloaded and compiled the CVS version of slrn, and it correctly displays properly declared UTF-8 and legacy character set messages. With the addition of some patches by Thomas Wiegner, which go a long way towards detecting improperly declared character sets, I can say that every message I have viewed since installing has been correctly displayed.

At the time of writing this (28 July 2007), Thomas Wiegner has just compiled a Debian package for Ubuntu Feisty which includes the CVS version and his patches. I have not used it, but this promises to be a very good package for Debian-based slrn users. This version should also work on Debian unstable, Debian Etch, Debian Lenny, Ubuntu Dapper and probably other Debian-based distributions. If you don't use a Debian-based distro, read on.

Here's how I compiled and installed it, with the help of Peter J Ross and Andrew Strong. By the way, Andrew has written a very helpful page on this issue also.

Get the CVS version of slrn:

```
$ cd
$ mkdir cvs
$ cd cvs
$ cvs -d:pserver:anonymous@slrn.cvs.sourceforge.net:/cvsr
$ cvs -z3 -d:pserver:anonymous@slrn.cvs.sourceforge.net:/
```

Make sure you have automake 1.10 and autoconf 2.61, which are needed to compile this version of slrn.

```
$ automake --version
automake (GNU automake) 1.10
...
$ autoconf --version
autoconf (GNU Autoconf) 2.61
...
```

If you don't, and I didn't on my Dapper Drake Ubuntu 6.06, remove the old versions and get the new ones. Automake versions can be downloaded from http://ftp.gnu.org/gnu/automake/ and autoconf from http://ftp.gnu.org/gnu/autoconf/

```
$ cd
$ mkdir downloads
$ cd downloads
$ sudo apt-get remove automake
$ wget http://ftp.gnu.org/gnu/automake/automake-1.10.tar.
$ tar xzvf automake-1.10.tar.gz
$ cd automake-1.10
$ ./configure
$ make
$ sudo make install
$ sudo apt-get remove autoconf
$ wget http://ftp.gnu.org/gnu/autoconf/autoconf-2.61.tar.
$ tar xzvf autoconf-2.61.tar.gz
$ cd autoconf-2.61
$ ./configure
$ make
$ sudo make install
```

Now we can compile slrn :-)

```
$ cd
$ cd cvs/slrn
$ ./autogen.sh
$ ./configure
$ make all
$ sudo make install
```

If you wish to install slrnpull or add in other compile-time options, add the appropriate switches to the `./configure` line above.

The above will give you the latest and greatest slrn unreleased as a tarball. Want to improve it? Of course you do. No let's install Thomas' patches to try to handle incorrectly declared character sets.

```
$ cd
$ cd cvs/slrn
$ make clean
$ wget http://www.foory.de/thw/slrn/fallback_charset.patc
$ wget http://www.foory.de/thw/slrn/us-ascii_override.pat
$ patch -p0 < fallback_charset.patch
$ patch -p0 < us-ascii_override.patch
$ ./autogen.sh
$ ./configure
$ make all
$ sudo make install
```

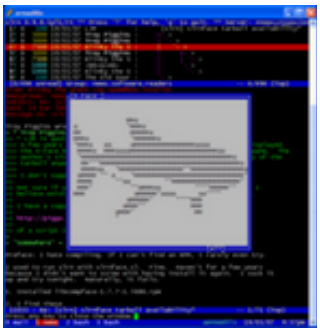
To implement the above patches you need to add the following lines to your $HOME/.slrnrc file:

```
set fallback_charset iso-8859-1
set usascii_override 1
```

Back to top

How to show X-Faces in consoles.

Personally, I don't have a use for X-Faces. But a while ago I was curious to see what they were and tinkered with them a bit. Here is how I got them to appear in my screen terminal over ssh connection. They look like this:



First I downloaded a shell script called view-x-face by Roland Rosenfeld, saved it in my $PATH (in /usr/local/bin) and made it executable. Make sure that if you save it somewhere other than /usr/local/bin/, you will need to edit the slrnface.sl macro below to correct the path.

Looking at the comments in the beginning of the script, there are some pre-requisites - 'uncompface' from the compface package, 'icontopbm' from the pmbplus or netpbm package, and because I don't have X11 running, 'image2ascii' shell script again by Roland Rosenfeld which call on some commands provided by ImageMagick. On my Ubuntu system I simply did something like this:

```
sudo apt-get install compface
```

and so on.

Next thing was to get a slrn macro called 'slrnface'. It was provided by Jurriaan Kalkman on the slrn-user mailing list on 20/1/02. Download slrnface.sl here. Check the path to view-x-face, because it is coded to /usr/local /bin/view-x-face by default.

Once you've interpretted slrnface, that's it. Just hit 'x' to view the X-Face. If you want to automatically view X-Faces upon opening articles, simply register it as a read_article_hook.

By default, view-x-face displays the image full screen width. If you want to decrease the size of it, just edit this line in view-x-face:

```
VIEWER=image2ascii
```

to this:

```
VIEWER="image2ascii -geometry 64x64"
```

You lose some of the X-Face quality, but they were never meant to be so large.

Back to top

NNTP-Posting-Host whois macro.

Updated 9/9/07 version 0.9.1 - can now specify user-defined pagers instead of hard-coded to 'less'.

Just threw together a quick macro for slrn that extracts the NNTP-Posting-Host header's IP address or hostname, runs it through whois, and displays the output all within slrn.

I find sometimes I do whois lookups to distinguish legit vs imposter posts, and decided I do it enough to make writing a small macro to make life easier. It's not bad.

Just download whois.sl and interpret it.

```
% Filename: whois.sl
% Version: 0.9.1
% Author: Troy Piggins

% 0.9.1 - added user defineable pager

% You need 'whois' and a pager for this macro to work.  P
% path below.
```

```
define whois_postinghost () {

% set the path to your favourite pager here
  variable pager="/usr/bin/less";
%  variable pager="/usr/local/share/vim/vim70/macros/less
%  variable pager="/usr/local/bin/most";

  variable status= is_article_visible ();

  if ( status & 1) {
    variable postinghost= extract_article_header ( "NNTP-

    variable ph_exists= strlen ( postinghost);

    if ( ph_exists > 0) {
      () = system ( "whois "+postinghost+" | "+pager);
    } else {
      error ("This article doesn't contain a NNTP-Posting
%     throw InvalidParmError, "This article doesn't conta
    }

  } else {
     error ("You need to display the article to check");
%    throw InvalidParmError, "You need to display the art
  }

}

definekey ( "whois_postinghost", "\eH", "article");
```

The definekey statement binds the macro to [ESC]-H, but you can change that to whatever you want.

Back to top

Opening a bash shell within slrn.

The macro opens a bash shell from within slrn. You easily, of course, extend the 'shell' variable to any other shell of your choosing. Download shell.sl and interpret it.

```
% Filename: shell.sl
% Version: 0.9.0
% Author: Troy Piggins

% This macro opens a bash shell from within slrn.

% I had the idea for when I visit groups like grc.news.la
% update for something is announced.  I just hit my 'brow
% default, on the open article which usually provides a l
% software to download.  Once it's downloaded I run this
% shell, change to the dir for source code, compile and i
% hit.  Once finished I just 'ctrl-d' to quit the shell a
% slrn.  I'm lazy so I figure I will be more likely to ke
% as I read them than going back and doing it later.

define shell () {

  variable shell="/bin/bash";

  () = system ( shell);

}
```

```
definekey ( "shell", "\eb", "article");
definekey ( "shell", "\eb", "group");
```

Back to top

How to get different coloured subjects for different scores -> Rudy Taraschi's score color patch.

Rudy has written a patch that allows you to assign a different color for different score ranges. I've been using this patch for years and find it invaluable.

I intend to write more about this, with screenshots, but for the moment I'll just post a link to his download page:

http://www.taraschi.net/slrn

Make sure the patch you download corresponds to the version of slrn you are compiling. Things are moving pretty fast again with slrn development working towards version 0.9.9.

Back to top

Links.

The slrn homepage.
news.software.readers group - be sure to include 'slrn' in the subject of your post to attract suitable responses.
slrn-user mailing list or just browse the slrn-user archives
The S-Lang Library Documentation Page for reference while writing macros.
Usenet Netiquette on Wikipedia.
Andrew Strong's Setting up slrn under linux

Back to top
