

- [\(TOP\)](#)
- [Lynx Users Guide](#)
- [Key-stroke Commands](#)
- [Line Editor](#)
- [Supported URLs](#)
- About
  - [Lynx](#)
  - [Lynx-Dev](#)
- Configuration
  - [by Category](#)
  - [by Names](#)

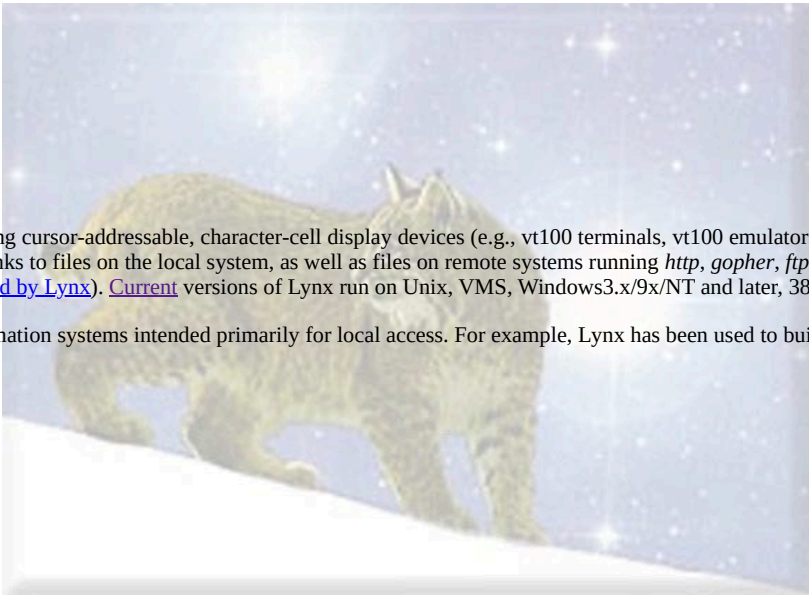
Lynx is a fully-featured *World Wide Web* (WWW) client for users running cursor-addressable, character-cell display devices (e.g., vt100 terminals, vt100 emulators running on PCs or Macs, or any other character-cell display). It will display *Hypertext Markup Language* (HTML) documents containing links to files on the local system, as well as files on remote systems running *http*, *gopher*, *ftp*, *wais*, *nntp*, *finger*, or *cso/ph/qi* servers, and services accessible via logins to *telnet*, *tn3270* or *rlogin* accounts (see [URL Schemes Supported by Lynx](#)). [Current](#) versions of Lynx run on Unix, VMS, Windows3.x/9x/NT and later, 386DOS and OS/2 EMX.

Lynx can be used to access information on the *WWW*, or to build information systems intended primarily for local access. For example, Lynx has been used to build several *Campus Wide Information Systems* (CWIS). In addition, Lynx can be used to build systems isolated within a single LAN.

## Table of Contents

- [Lynx online help](#)
- [Viewing local files with Lynx](#)
- [Leaving Lynx](#)
- [Starting Lynx with a Remote File](#)
- [Starting Lynx with the WWW\\_HOME environment variable](#)
- [Navigating hypertext documents with Lynx](#)
- [Printing, Mailing, and Saving rendered files to disk](#)
- [Viewing the HTML document source and editing documents](#)
- [Downloading and Saving source files](#)
- [Reloading files and refreshing the display](#)
- [Lynx searching commands](#)
- [Lynx Options Menu](#)
- [Comments and mailto: links](#)
- [USENET News posting](#)
- [Lynx bookmarks](#)
- [Jump command](#)
- [Directory Editing](#)
- [Using Color & the Mouse](#)
- [Scrolling and Other useful commands](#)
- [Lynx and HTML Forms](#) | [Lynx and HTML Images](#)
- [Lynx and HTML Tables](#) | [Lynx and HTML Tabs](#)
- [Lynx and HTML Frames](#) | [Lynx and HTML Banners](#)
- [Lynx and HTML Footnotes](#) | [Lynx and HTML Notes](#)
- [Lynx and HTML Lists](#)
- [Lynx and HTML Quotes](#)
- [Lynx and HTML Internationalization: 8bit, UNICODE, etc.](#)
- [Lynx and Client-Side-Image-Maps](#)
- [Lynx and Client-Side-Pull](#)
- [Lynx and State Management](#) (Me want cookie!)
- [Lynx and Cached Documents](#)
- [Lynx and Sessions](#)
- [The Lynx command line](#)
- [Environment variables used by Lynx](#)
- [Main configuration file lynx.cfg](#)
- [Lynx development history](#)

## Lynx online help



Online help is available while viewing any document. Press the “?” or “H” key (or the “h” key if vi-like key movement is not on) to see a list of help topics. See the section titled [Navigating hypertext documents with Lynx](#) for information on navigating through the help files.

In addition, a summary description of all the Lynx keystroke commands and their key bindings is available by pressing the “K” key (or the “k” key if vi-like key movement is not on).

If you want to recall recent status-line messages, you can do so by entering the “g” command, followed by “LYNXMESSAGES:”.

[\[ToC\]](#)

## Viewing local files with Lynx

Lynx can be started by entering the Lynx command along with the name of a file to display. For example these commands could all be used to display an arbitrary ASCII text or HTML file:

```
UNIX
lynx filename
lynx /home/my-dir/filename
lynx ~/filename

VMS
lynx filename
lynx dua5:[my-directory]filename
lynx /dua5/my-directory/filename
lynx ~/filename
lynx sys$login:filename
lynx /sys$login/filename

Win32/DOS
lynx file:///filename
lynx filename
lynx c:/dir/filename
lynx //n/dir/filename
```

When executed, Lynx will clear the screen and display as much of the specified file as will fit on the screen. Pressing a *down-arrow* will bring up the next screen, and pressing an *up-arrow* will bring up the previous screen. If no file is specified at startup, a default file will be displayed, depending on settings e.g., in *lynx.cfg*.

Lynx will display local files written in the *HyperText Markup Language (HTML)*, if the file's name ends with the characters *.html*, *.htm*, *.shtml*, *.htmlx*, *.html3*, or *.ht3*. HTML is a file format that allows users to create a file that contains (among other things) hypertext links to other files. Several files linked together may be described as a *hypertext document*. If the filename does not have one of the suffixes mapped by Lynx to HTML, the *-force\_html* command line option can be included to force treatment of the file as hypertext.

When Lynx displays an HTML file, it shows links as "bold face" text, except for one link, which is shown as "highlighted" text. Whether "boldface" or "highlighted" text shows up as reverse video, boldface type, or a color change, etc. depends on the display device being used (and the way in which that device has been configured). Lynx has no control over the exact presentation of links.

The one link displayed as "highlighted" text is the currently "selected" link. Lynx will display the file associated with the selected link when a *right-arrow* or a *Return* key is pressed. To select a particular link, press the *up-arrow* or *down-arrow* keys until the desired link becomes "highlighted," and then press the *right-arrow* or *Return* key to view the linked information. Information included in the HTML file tells Lynx where to find the linked file and what kind of server will provide it (i.e., HTTP, Gopher, etc.).

Lynx renders HTML files and saves the rendition (and the source, if so configured in the [lynx.cfg](#) file) for initial display and should you select the link again. If you do select a link again and have reason to desire a new fetch and rendering of the file, use the NOCACHE command, normally mapped to “X” and “X”, instead of the *right-arrow* or *Return* key when positioned on the link. You also can force a new fetch and rendering of the currently displayed document via the RELOAD command, normally mapped to *Control-R*.

When a binary file is encountered Lynx will ask the user if he/she wishes to download the file or cancel. If the user selects “D” for download, Lynx will transfer the file into a temporary location and present the user with a list of options. The only default option is *Save to disk*, which is disabled if Lynx is running in anonymous mode. Additional download methods may be defined in the [lynx.cfg](#) file. Programs like *kermit*, *zmodem* and *FTP* are some possible options.

[\[ToC\]](#)

## Leaving Lynx

To exit Lynx use the “q” command. You will be asked whether you really want to quit. Answering “y” will exit and “n” will return you to the current document. Use “Q” or *Control-D* to quit without verification.

[\[ToC\]](#)

## Starting Lynx with a Remote File

if you wish to view a remote file (that is, a file located on some computer system other than the one upon which you are running Lynx) without first viewing a local file, you must identify that file by using a Uniform Resource Locator (URL). URLs take the general form:

PROTOCOL :// HOST / PATH

where

PROTOCOL

identifies the communications protocol (*scheme*) used by the server that will provide the file. As mentioned earlier, Lynx (and any WWW client) can interact with a variety of servers, each with its own protocol.

HOST

is the Internet address of the computer system on which the server is running, and

PATH

is a scheme-specific field which for some schemes may correspond to a directory path and/or filename.

Here are some sample URLs.

HTTP (HyperText Transfer Protocol)

`https://invisible-island.net/lynx/`

Gopher

`gopher://gopher.micro.umn.edu/11/`

FTP (File Transfer Protocol)

`ftp://ftp2.cc.ukans.edu/pub/lynx/README`

WAIS (Wide Area Information Service protocol)

`wais://cnidr.org/directory-of-servers`

A URL may be specified to Lynx on the command line, as in:

`lynx http://kufacts.cc.ukans.edu/cwis/kufacts_start.html`

Lynx also will attempt to create a complete URL if you include adequate portions of it in the startfile argument. For example:

<code>wfbr</code>	will be expanded to:
<code>http://www.wfbr.edu/</code>	and:
<code>ftp.more.net/pub</code>	will be expanded to:
<code>ftp://ftp.more.net/pub</code>	

See [URL Schemes Supported by Lynx](#) for more detailed information.

[\[ToC\]](#)

## Starting Lynx with the WWW\_HOME environment variable.

You may also specify a starting file for Lynx using the WWW\_HOME environment variable,

UNIX

`ksh`  
`export WWW_HOME=http://www.w3.org/`

`csh`  
`setenv WWW_HOME http://www.w3.org/`

VMS

`define "WWW_HOME" "http://www.w3.org/"`

win32

`WWW_HOME=http://www.w3.org/ [or in registry]`

Note that on VMS the double-quoting *must* be included to preserve casing.

[\[ToC\]](#)

## Navigating hypertext documents with Lynx

The process of moving within a hypertext web, selecting and displaying links is known as "navigation." With Lynx almost all navigation can be accomplished with the arrow keys and the numeric keypad.

arrow keys				<table><tr><td>TOP</td><td>/ \</td><td>Page</td><td></td></tr><tr><td>of</td><td></td><td>UP</td><td></td></tr><tr><td>text 7</td><td></td><td>8</td><td>9</td></tr></table>				TOP	/ \	Page		of		UP		text 7		8	9												
TOP	/ \	Page																													
of		UP																													
text 7		8	9																												
<table><tr><td>SELECT</td><td>/ \</td><td></td><td></td></tr><tr><td>prev</td><td></td><td></td><td></td></tr><tr><td>link</td><td></td><td></td><td></td></tr></table>				SELECT	/ \			prev				link				<table><tr><td>&lt;---</td><td></td><td></td><td>---&gt;</td></tr><tr><td></td><td>4</td><td>5</td><td>6</td></tr></table>				<---			--->		4	5	6				
SELECT	/ \																														
prev																															
link																															
<---			--->																												
	4	5	6																												
<table><tr><td>BACK</td><td>SELECT</td><td>DISPLAY</td><td></td></tr><tr><td>&lt;-- prev</td><td>next</td><td>sel. --&gt;</td><td></td></tr><tr><td>doc.</td><td>link \ /</td><td>link</td><td></td></tr></table>				BACK	SELECT	DISPLAY		<-- prev	next	sel. -->		doc.	link \ /	link		<table><tr><td>END</td><td></td><td>Page</td><td></td></tr><tr><td>of</td><td></td><td>DOWN</td><td></td></tr><tr><td>text 1</td><td>\ / 2</td><td></td><td>3</td></tr></table>				END		Page		of		DOWN		text 1	\ / 2		3
BACK	SELECT	DISPLAY																													
<-- prev	next	sel. -->																													
doc.	link \ /	link																													
END		Page																													
of		DOWN																													
text 1	\ / 2		3																												

There are also a few other keyboard commands to aid in navigation. The Control and Function keys used for navigation within the current document are described in [Scrolling and Other useful commands](#).

Some additional commands depend on the fact that Lynx keeps a list of each link you visited to reach the current document, called the [History Page](#), and a list of all links visited during the current Lynx session, called the [Visited Links Page](#).

- The HISTORY keystroke command, normally mapped to *Backspace* or *Delete*, will show you the *History Page* of links leading to your access of the current document. Any of the previous documents shown in the list may be revisited by selecting them from the history screen.
- The VLINKS keystroke command, normally mapped to uppercase “V”, will show the *Visited Links Page*, and you similarly can select links in that list.
- The MAIN\_MENU keystroke command, normally mapped to “m” and “M”, will take you back to the starting document unless you specified the *-homepage=URL* option at the command line.
- Also, the LIST and ADDRLIST keystroke commands, normally mapped to “l” and “A” respectively, will create a compact lists of all the links in the current document, and they can be selected via those lists.

The “i” key presents an index of documents. The default index offered contains many useful links, but can be changed in *lynx.cfg* or on the command line using the *-index=URL* switch.

If you choose a link to a server with active access authorization, Lynx will automatically prompt for a username and a password. If you give the correct information, you will then be served the requested information. Lynx will automatically send your username and password to the same server if it is needed again.

[\[ToC\]](#)

## Printing, Mailing, and Saving rendered files to disk.

Rendered HTML documents, and plain text files, may be printed using the “p” command while viewing the document. After pressing the “p” key a menu of *Print Options* will be displayed. The menu will vary according to several factors. First, some sites set up special accounts to let users run Lynx to access local information systems. Typically these accounts require no passwords and do not require users to identify themselves. As a result such accounts are called "anonymous" accounts, and their users are considered "anonymous" users. In most configurations, all Lynx users (including anonymous users) are able to mail files to themselves and print the entire file to the screen.

Additional print options are available for users who are using Lynx from their own accounts (that is, so-called "non-anonymous users"). In particular, the *Save to a local file* option allows you to save the document into a file on your disk space. Additional print options may also be available as configured in the [lynx.cfg](#) file.

Some options, such as *Save to a local file*, involve prompting for an output filename. All output filename entries are saved in a circular buffer, and any previous entries can be retrieved for re-use by pressing the *up-arrow* or *down-arrow* keys at the prompt.

Note that if you want exact copies of text files without any expansions of TAB characters to spaces you should use the [Download](#) options.

[\[ToC\]](#)

## Viewing the HTML document source and editing documents

When viewing HTML documents it is possible to retrieve and display the unrendered (i.e., the original HTML) source of the document by pressing the “\” (backslash) key. Lynx usually caches only the rendering of the document and does not keep the source (unless it is configured to do so in the [lynx.cfg](#) file), so to display the source unrendered, Lynx must reload it from the server or disk. When viewing unrendered documents you may print them as any normal document.

Selecting the *Print to a local file* option from the Print Menu makes it possible to save the source of the document to disk so that you may have a local copy of the document source, but it is better to [Download](#) the source.

NOTE: When saving an HTML document it is important to name the document with a *.html* or *.htm* extension, if you want to read it with Lynx again later.

Lynx can allow users to edit documents that reside on the local system. To enable editing, documents must be referenced using a "file:" URL or by specifying a plain filename on the command line as in the following two examples:

Command

```
lynx file://localhost/FULL/PATH/FILENAME
```

lynx path/filename.html

In addition, the user must also specify an editor in the *Options Menu* so that Lynx knows which editor to use. If the file is specified correctly and an editor is defined, then you may edit documents by using the “e” command. When the “e” command is entered your specified editor is spawned to edit the file. After changes are completed, exit your editor and you will return to Lynx. Lynx will reload and render the file so that changes can be immediately examined.

[\[ToC\]](#)

## Downloading and Saving source files.

If the DOWNLOAD keystroke command (“d” or “D”) is used when positioned on a link for an HTML, plain text, or binary file, Lynx will transfer the file, without rendering, into a temporary location and present the user with a list of options, just as it does when a link for a binary file of a type for which no viewer has been mapped is activated.

There is a default *Download option of Save to disk*. This is disabled if Lynx is running in anonymous mode. Any number of download methods such as kermit and zmodem may be defined in addition to this default in the *lynx.cfg* file. Using the *Save to disk* option under the PRINT command after viewing the source of an HTML with the VIEW SOURCE (\) command will result in a file which differs from the original source in various ways such as tab characters expanded to spaces. Lynx formats the source presentation in this mode. On the other hand, if the DOWNLOAD command is used, the only change will be that Lynx optionally puts

```
<!--X-URL: http://www.site.foo/path/to/file.html -->
<BASE href="http://www.site.foo/path/to/file.html">
```

at the start of the file so that relative URLs in the document will still work. Even this modification can be prevented by setting PREPEND\_BASE\_TO\_SOURCE:FALSE in *lynx.cfg*.

Some options, such as *Save to disk*, involve prompting for an output filename. All output filename entries are saved in a circular buffer, and any previous entries can be retrieved for re-use by pressing the *up-arrow* or *down-arrow* keys at the prompt.

[\[ToC\]](#)

## Reloading files and refreshing the display

The RELOAD (*Control-R*) command will reload and re-render the file that you are currently viewing. The REFRESH (*Control-L* or *Control-W*) command will refresh or wipe the screen to remove or correct any errors that may be caused by operating system or other messages.

The NOCACHE (“x” or “X”) command can be used in lieu of ACTIVATE (*Return* or *right-arrow*) to request an uncached copy and new rendition for the current link, or resubmission of a FORM, if a cache from a previous request or submission exists. The request or submission will include *Pragma: no-cache* and *Cache-Control: no-cache* in its headers. Note that FORMs with POST content will be resubmitted regardless of whether the NOCACHE or ACTIVATE command is used (see [Lynx and HTML Forms](#)).

[\[ToC\]](#)

## Lynx searching commands

Two commands activate searching in Lynx: “/” and “S”.

While viewing a normal document use the “/” command to find a word or phrase within the current document. The search type will depend on the search option setting in the [Options Menu](#). The search options are case sensitive and case insensitive. These searches are entirely local to Lynx.

Some documents are designated *index documents* by virtue of an ISINDEX element in their HEAD section. These documents can be used to retrieve additional information based on searches using words or phrases submitted to an index server. The Lynx statusline will indicate that you are viewing such a document, and if so, the “S” key will invoke a statusline prompt to enter a query string. The prompt can be specified via a PROMPT attribute in the ISINDEX element. Otherwise, Lynx will use an internally configured prompt. The address for submitting the search can be specified via an HREF or ACTION attribute. Otherwise, Lynx will use the current document’s URL and append your query string as a *?searchpart* (see [Supported URLs](#)).

All search words or strings which you have entered during a Lynx session are saved in a circular buffer, and can be retrieved for re-use by pressing the *up-arrow* or *down-arrow* keys at the prompt for a search word or string. Also, you can use the “n”ext command to repeat a search with the last-entered search word or phrase, starting from the current position in the document. The word or phrase matches will be highlighted throughout the document, but such highlighting will not persist for new documents, or if the current document is reloaded. The search cycles to the top of the document if the word or phrase is not located below your current position.

Although [HTML Forms](#) have largely replaced index documents for searches via http servers, they are still useful for performing searches directly via WAIS or Gopher servers in conjunction with the internal gateways for such servers. For example, an HTML index document can act as a *cover page* describing a WAIS database and how to formulate query strings for searching it, and include an element such as:

```
<ISINDEX PROMPT="Enter WAIS query:"
      HREF="wais://net.bio.net/biologists-addresses">
```

for submitting a search of the Biologist’s Addresses database directly to the net.bio.net WAIS server.

[\[ToC\]](#)

# Lynx Options Menu

The Lynx *Options Menu* may be accessed by pressing the “o” key. It allows you to change options at runtime, if you need to. Most changes are read from & saved to your .lynxrc file; those which are not are marked (!) in the form-based menu (as below). Many other options are stored in the [lynx.cfg](#) file.

Lynx supports two styles of Options Menu:

- [form-based](#)
- [key-based](#)

The form-based menu shown below is an HTML file generated at runtime, in which the user fills in choices as in any ordinary HTML form.

## [Options Menu \(Lynx Version 2.9.2\)](#)

Accept Changes - Reset Changes - Left Arrow cancels changes - HELP!

Save options to disk: [ ]  
(options marked with (!) will not be saved)

### General Preferences

User mode : [\[Advanced\]](#).  
Editor : [vile](#)  
Type of Search : [\[Case insensitive\]](#).

### Security and Privacy

Cookies : [\[ask user\]](#).  
Invalid-Cookie Prompting (!) : [\[prompt normally\]](#).  
SSL Prompting (!) : [\[prompt normally\]](#).

### Keyboard Input

Keypad mode : [\[Numbers act as arrows\]](#).  
Emacs keys : [\[OFF\]](#).  
VI keys : [\[OFF\]](#).  
Line edit style : [\[Bash-like Bindings\]](#).

### Display and Character Set

Use locale-based character set : [\[ON\]](#).  
Use HTML5 charset replacements(!): [\[OFF\]](#).  
Display character set : [\[UNICODE \(UTF-8\)\]](#).  
Assumed document character set(!): [\[iso-8859-1\]](#).  
Internationalized domain names(!): [\[IDNA TR46\]](#).  
Raw 8-bit : [\[OFF\]](#).  
X Display : [localhost:0.0](#)

### Document Appearance

Show color : [\[ON\]](#).  
Color style (!) : [\[lynx.lss\]](#)  
Default colors (!) : [\[ON\]](#).  
Show cursor : [\[OFF\]](#).  
Underline links (!) : [\[OFF\]](#).  
Show scrollbar : [\[ON\]](#).  
Popups for select fields : [\[ON\]](#).  
HTML error recovery : [\[strict \(SortaSGML mode\)\]](#).  
Bad HTML messages (!) : [\[Warn, point to trace-file\]](#).  
Show images : [\[ignore\]](#).  
Verbose images : [\[OFF\]](#).  
Collapse BR tags (!) : [\[OFF\]](#).  
Trim blank lines (!) : [\[trim-lines\]](#).

### Headers Transferred to Remote Servers

Personal mail address : [\\_\\_\\_\\_\\_](#)  
Personal name for mail : [\\_\\_\\_\\_\\_](#)  
Password for anonymous ftp : [\\_\\_\\_\\_\\_](#)

```

Preferred media type (!)      : \[Accept lynx's internal types\].
Preferred encoding (!)       : \[All\].
Preferred document character set : _____
Preferred document language   : en
HTTP protocol (!)            : \[HTTP 1.0\].
Send User-Agent header (!)    : \[X\].
User-Agent header (!)         : Lynx/2.9.2 libwww-FM/2.14 SSL-MM/1.4.

```

```

Listing and Accessing Files
Use Passive FTP (!)          : \[ON\].
FTP sort criteria             : \[By Date\].
Local directory sort criteria : \[Directories first\].
Local directory sort order    : \[By Date\].
Show dot files                : \[OFF\].
Pause when showing message (!) : \[ON\].
Show transfer rate           : \[Show KiB/sec \(2-digits\), ETA\].

```

```

Special Files and Screens
Multi-bookmarks              : \[ADVANCED\].
Review/edit Bookmarks files  : Goto multi-bookmark menu
Auto Session (!)             : \[OFF\].
Session file (!)             : _____
Visited Pages                 : \[By Last Visit Reversed\].

```

View the file lynx.cfg.

Accept Changes - Reset Changes - Left Arrow cancels changes

The key-based menu depends on key-strokes to identify options which the user wants to change. It is compiled into Lynx and is accessed by setting FORMS\_OPTIONS to TRUE in lynx.cfg.

### [Options Menu \(Lynx Version 2.9.2\)](#)

```

(E)ditor                      : emacs
(D)ISPLAY variable           : aixtest.cc.ukans.edu:0.0
mu(L)ti-bookmarks: OFF       B)ookmark file: lynx_bookmarks.html
(E)TP sort criteria          : By Filename
(P)ersonal mail address      : montulli@netscape.com
(S)earching type             : CASE INSENSITIVE
preferred document lan(G)uage: en
preferred document c(H)arset : NONE
display (C)haracter set      : Western (ISO-8859-1)
raw 8-bit or CJK m(O)de      : ON      show color (G) : OFF
(V)I keys: OFF   e(M)acs keys: OFF   sho(W) dot files: OFF
popups for selec(T) fields    : ON      show cursor (@) : OFF
(K)eypad mode                : Numbers act as arrows
li(N)e edit style            : Default Binding
l(I)st directory style       : Mixed style
(U)ser mode                   : Advanced      verbose images (I) : ON
user (A)gent                  : [User-Agent header]
local e(X)ecution links      : FOR LOCAL FILES ONLY

```

An option can be changed by entering the capital letter or character in parentheses for the option you wish to change (e.g., “E” for Editor or “@” for show cursor). For fields where text must be entered, simply enter the text by typing on the keyboard. The [Line Editor](#) can be used to correct mistakes, and *Control-U* can be used to erase the line. When you are done entering a change press the *Return* key to get back to the *Command?* prompt.

For fields where you must choose one of two choices, press any key to toggle the choices and press the *Return* key to finish the change.

For fields where you potentially have more than two choices, popup windows may be evoked which function homologously to those for select fields in [HTML Forms](#). The popup windows will be invoked only if you have popups for select fields set to ON (see below). Otherwise, your cursor will be positioned at the current choice, and you can press any key to cycle through the choices, then press the *Return* key to finish the change.

When you are done changing options use the “r” command to return to Lynx or the “>” command to save the options to a .lynxrc file and return to Lynx.

The following table describes the options available on the *Options Menu*:

Assumed document character set

This option changes the handling of documents which do not explicitly specify a charset. Normally Lynx assumes that 8-bit characters in those documents are encoded according to iso-8859-1 (the official default for the HTTP protocol). Unfortunately, many non-English web pages "forget" to include proper charset info; this option helps you to browse those broken pages if you know by some means what the charset is.

When the value given here or by an `-assume_charset` command line flag is in effect, Lynx will treat documents as if they were encoded accordingly. This option is active when "Raw 8-bit or CJK Mode" is OFF.

## Auto Session

Lynx can save and restore useful information about your browsing history. Use this setting to enable or disable the feature.

## Bad HTML messages

Suppress or redirect Lynx's messages about "Bad HTML":

Ignore

do not warn; no details are written to the trace-file.

Add to trace-file

add the detailed warning message to the trace-file.

Add to LYNXMESSAGES

add the detailed warning message to the message page at "LYNXMESSAGES:".

Warn, point to trace-file

show a warning message on the status line; the complete message is written to the trace-file.

## Bookmark file

When multi-bookmarks is OFF, this is the filename and location of your default personal bookmark file. Enter "B" to modify the filename and/or location via the [Line Editor](#). Bookmark files allow frequently traveled links to be stored in personal easy to access files.

Using the "a"dd bookmark link command (see [Lynx bookmarks](#)) you may save any link that does not have associated POST content into a bookmark file. All bookmark files must be in or under your account's home directory. If the location specified does not begin with a dot-slash (./), its presence will still be assumed, and referenced to the home directory.

When multi-bookmarks is STANDARD or ADVANCED, entering "B" will invoke a menu of up to 26 bookmark files (associated with the letters of the English alphabet), for editing their filenames and locations (*filepath*), and descriptions.

Lynx will create bookmark files, if they do not already exist, when you first "a"dd a bookmark link to them. However, if you've specified a subdirectory (e.g., ./BM/lynx\_bookmarks.html), that subdirectory must already exist. Note that on VMS you should use the URL syntax for the filepath (e.g., *not* [.BM]lynx\_bookmarks.html).

## Collapse BR tags

If *Collapse BR tags* is turned off, Lynx will not collapse serial BR tags. If turned on, i.e., "collapse", two or more concurrent BRs will be collapsed into a single line break. Note that the valid way to insert extra blank lines in HTML is via a PRE block with only newlines in the block.

## Cookies

This option allows you to tell how to handle cookies: *ignore*, *prompt (ask user)* or *accept all*.

## Display Character set

This option allows you to set up the default character set for your specific terminal. The display character set provides a mapping from the character encodings of viewed documents and from HTML entities into viewable characters. It should be set according to your terminal's character set so that characters other than 7-bit ASCII can be displayed correctly, using approximations if necessary. You must have the selected character set installed on your terminal. (Since Lynx supports a wide range of platforms it may be useful to note that cpXXX codepages used within IBM PC computers, and windows-xxxx within native MS-Windows apps.)

## Editor

The editor to be invoked when editing browsable files, when sending mail or comments, when preparing a news article for posting, and for external TEXTAREA editing. The full pathname of the editor command should be specified when possible.

If a full pathname is given, this helps Lynx provide for detecting if options were also provided in this field. In this case, it will also quote the pathname, allowing for embedded blanks and other special characters that might confuse the shell which starts the editor program.

## Emacs keys



If set to ON then the CTRL-P, CTRL-N, CTRL-F, and CTRL-B keys will be mapped to up-arrow, down-arrow, right-arrow, and left-arrow, respectively. Otherwise, they remain mapped to their configured bindings (normally UP\_TWO lines, DOWN\_TWO lines, NEXT\_PAGE, and PREV\_PAGE, respectively).

Note: this has no direct effect on the line-editor's key bindings.

## Execution links

This deals with execution of local scripts or links. Local execution is activated when Lynx is first set up. If it has not been activated you will not see this option in the *Options Menu*.

When a local execution script is encountered Lynx checks the users options to see whether the script can be executed. Users have the following options:

Always off

Local execution scripts will never be executed

For Local files only

Local execution scripts will only be executed if the script to be executed resides on the local machine, and is referenced by a URL that begins with *file://localhost*

Always on

All local execution scripts will be executed

If the users options permit the script to be executed Lynx will spawn a shell and run the script. If the script cannot be executed Lynx will show the script within the Lynx window and inform the user that the script is not allowed to be executed and will ask the user to check his/her options.

## FTP sort criteria

This option allows you to specify how files will be sorted within FTP listings. The current options include "By Filename", "By Size", "By Type", and "By Date".

## HTML error recovery

Select the [recovery mode](#) used by Lynx.

## HTTP protocol

Normally Lynx negotiates HTTP/1.0, because it does not support chunked transfer (a requirement for all HTTP/1.1 clients), although it supports several other features of HTTP/1.1. You may encounter a server which does not support HTTP/1.0 which can be used by switching to the later protocol.

## Internationalized domain names

Convert internationalized domain names to and from ASCII.

IDNA 2003

Convert using the older “transitional” scheme.

IDNA 2008

Convert using the newer “non-transitional” scheme.

IDNA TR46

Use IDNA 2008 with the amendments from Unicode [Technical Report 46](#).

IDNA Compatible

First try converting using IDNA 2008, and if unsuccessful, try IDNA 2003.

## Invalid-Cookie Prompting

This allows you to tell how to handle invalid cookies: *prompt normally* to prompt for each cookie, *force yes-response* to reply "yes" to each prompt, *force no-response* to reply "no" to each prompt.

## Keypad mode

This option gives the choice among navigating with the arrow keys, or having every link numbered so that the links may be selected or made current by numbers as well as using the arrow keys, or having every link as well as every form field numbered so that they can be selected or sought by numbers. See the

[Follow link \(or page\) number:](#) and

[Select option \(or page\) number:](#)

help for more information.

## Line edit style

This option allows you to set alternative key bindings for the built-in line editor, if alternative line-edit bindings have been compiled in. Otherwise, Lynx uses the [Default Binding](#).

## Local directory sort criteria

This applies to directory editing. Files and directories can be presented in the following ways:

#### Mixed style

Files and directories are listed together in alphabetical order.

#### Directories first

Files and directories are separated into two alphabetical lists. Directories are listed first.

#### Files first

Files and directories are separated into two alphabetical lists. Files are listed first.

### Local directory sort order

The Options Form also allows you to sort by the file attributes.

#### By name

by filename (the default)

#### By size

by file size, in descending order

#### By date

by file modification time, in descending order

#### By mode

by file protection

#### By type

by filename suffix, e.g., the text beginning with “.”

#### By user

by file owner's user-id

#### By group

by file owner's group-id

### Multi-bookmarks

Lynx supports a default bookmark file, and up to 26 total bookmark files (see below). When multi-bookmarks is OFF, the default bookmark file is used for the “v”iew bookmarks and “a”dd bookmark link commands. If multi-bookmark support is available in your account, the setting can be changed to STANDARD or ADVANCED. In STANDARD mode, a menu of available bookmarks always is invoked when you seek to view a bookmark file or add a link, and you select the bookmark file by its letter token (see *Bookmark file*, below) in that menu. In ADVANCED mode, you instead are prompted for the letter of the desired bookmark file, but can enter “=” to invoke the STANDARD selection menu, or *RETURN* for the default bookmark file.

### Password for anonymous ftp

If this is blank, Lynx will use your personal mail address as the anonymous ftp password. Though that is the convention, some users prefer to use some other string which provides less information. If the given value lacks a “@”, Lynx also will use your computer's hostname as part of the password. If both this field and the personal mail address are blank, Lynx will use your \$USER environment variable, or “WWWuser” if even the environment variable is unset.

### Pause when showing message

If set to “off”, this overrides the INFOSECS setting in lynx.cfg, to eliminate pauses when displaying informational messages, like the “-nopause” command line option.

### Personal mail address

This mail address will be used to help you send files to yourself and will be included as the From: address in any mail or comments that you send. It will also be sent as the From: field in HTTP or HTTPS requests if inclusion of that header has been enabled via the NO\_FROM\_HEADER definition in [lynx.cfg](#) (the compilation default is not to send the header), or via the *-from* command line toggle.

## Personal mail name

This mail name will be included as the "X-Personal\_Name" field in any mail or comments that you send if that header has not been disabled via the NO\_ANONYMOUS\_EMAIL definition in [lynx.cfg](#).

## Popups for select fields

Lynx normally uses a popup window for the OPTIONS in form SELECT fields when the field does not have the MULTIPLE attribute specified, and thus only one OPTION can be selected. The use of popup windows can be disabled by changing this setting to OFF, in which case the OPTIONS will be rendered as a list of radio buttons. Note that if the SELECT field does have the MULTIPLE attribute specified, the OPTIONS always are rendered as a list of checkboxes.

## Preferred document language

The language you prefer if multi-language files are available from servers. Use RFC 1766 abbreviations, e.g., en for English, fr for French, etc. Can be a comma-separated list, which may be interpreted by servers as descending order of preferences. You can also make your order of preference explicit by using q factors as defined by the HTTP protocol, for servers which understand it, for example: da, en-gb;q=0.8, en;q=0.7

## Preferred document charset

The character set you prefer if sets in addition to ISO-8859-1 and US-ASCII are available from servers. Use MIME notation (e.g., ISO-8859-2) and do not include ISO-8859-1 or US-ASCII, since those values are always assumed by default. Can be a comma-separated list, which may be interpreted by servers as descending order of preferences. You can also make your order of preference explicit by using q factors as defined by the HTTP protocol, for servers which understand it, for example: iso-8859-5, utf-8;q=0.8

## Preferred encoding

When doing a GET, lynx tells what types of compressed data it can decompress (the "Accept-Encoding:" string). This is determined by compiled-in support for decompression or external decompression programs. Use this option to select none, one or all of the supported decompression types.

## Preferred media type

When doing a GET, lynx lists the MIME types which it knows how to present (the "Accept:" string). Depending on your system configuration, the mime.types or other data given by the GLOBAL\_EXTENSION\_MAP may include many entries that lynx really does not handle. Use this option to select one of the built-in subsets of the MIME types that lynx could list in the Accept.

### Accept lynx's internal types

list only the types that are compiled into lynx.

### Also accept lynx.cfg's types

lists types defined in lynx.cfg, e.g., the VIEWER and Cern RULE or RULESFILE settings.

### Also accept user's types

lists types from the PERSONAL\_EXTENSION\_MAP setting in lynx.cfg

### Also accept system's types

lists types from the GLOBAL\_EXTENSION\_MAP setting in lynx.cfg

### Accept all types

adds the types that are in lynx's built-in tables for external programs that may be used to present a document.

## Raw 8-bit or CJK Mode

Tells Lynx whether 8-bit characters are assumed to correspond with the display character set and therefore are processed without translation via the chartrans conversion tables:

- Should be ON by default when the display character set is one of the Asian (CJK) sets and the 8-bit characters are Kanji multibytes.
- Should be OFF for the other display character sets, but can be turned ON when the document's charset is unknown (e.g., is not ISO-8859-1 and no charset parameter was specified in a reply header from an HTTP server to indicate what it is) but you know by some means that you have the matching display character set selected.
- Should be OFF when an Asian (CJK) set is selected but the document is ISO-8859-1 or another "assumed document character set".

The setting also can be toggled via the RAW\_TOGGLE command, normally mapped to "@", and at startup via the -raw switch.

## Send User-Agent header

Controls whether the user-agent string will be sent.

## Session file

Define the file name where lynx will store user sessions. This setting is used only when *Auto Session* is enabled.

#### Show color

This option will be present if color support is available. If set to ON or ALWAYS, color mode will be forced on if possible. If (n)curses color support is available but cannot be used for the current terminal type, selecting ON is rejected with a message. If set to OFF or NEVER, color mode will be turned off.

ALWAYS and NEVER are not offered in anonymous accounts. If saved to a *.lynxrc* file in non-anonymous accounts, ALWAYS will cause Lynx to set color mode on at startup if supported. If Lynx is built with the slang library, this is equivalent to having included the *-color* command line switch or having the *COLORTERM* environment variable set. If color support is provided by curses or ncurses, this is equivalent to the default behavior of using color when the terminal type supports it. If (n)curses color support is available but cannot be used for the current terminal type, the preference can still be saved but will have no effect.

A saved value of NEVER will cause Lynx to assume a monochrome terminal at startup. It is similar to the *-nocolor* switch, but (when the slang library is used) can be overridden with the *-color* switch.

If the setting is OFF or ON when the current options are saved to a *.lynxrc* file, the default startup behavior is retained, such that color mode will be turned on at startup only if the terminal info indicates that you have a color-capable terminal, or (when the slang library is used) if forced on via the *-color* switch or *COLORTERM* variable. This default behavior always is used in anonymous accounts, or if the *option\_save* restriction is set explicitly. If for any reason the startup color mode is incorrect for your terminal, set it appropriately on or off via this option.

#### Show cursor

Lynx normally hides the cursor by positioning it to the right and if possible the very bottom of the screen, so that the current link or OPTION is indicated solely by its highlighting or color. If show cursor is set to ON, the cursor will be positioned at the left of the current link or OPTION. This is helpful when Lynx is being used with a speech or braille interface. It also is useful for sighted users when the terminal cannot distinguish the character attributes used to distinguish the current link or OPTION from the others in the screen display.

#### Show dot files

If display/creation of hidden (dot) files/directories is enabled, you can turn the feature on or off via this setting.

#### Show images

This allows you to select the way in which Lynx shows image links. These are the available selections:

- *ignore* to suppress the links altogether,
- *as labels* to show the descriptive text for the link
- *as links*, which allows you to use an external viewer

#### Show scrollbar

This allows you to enable (show) or disable (hide) the scrollbar on the right-margin of the display. This feature is available with ncurses or slang libraries.

#### Show transfer rate

This allows you to select the way in which Lynx shows its progress in downloading large pages. It displays its progress in the status line. These are the available selections:

- Do not show rate
- Local directory sort order
- Show dot files
- Execution links
- Pause when showing message
- Show transfer rate

#### SSL Prompting

This allows you to tell how to handle errors detected in SSL connections *prompt normally* to prompt for each cookie, *force yes-response* to reply "yes" to each prompt, *force no-response* to reply "no" to each prompt.

#### Trim blank lines

If *Trim blank lines* is turned off, Lynx will not trim trailing blank lines from the document. Also, Lynx will not collapse BR-tags onto the previous line when it happens to be empty as part of the *Collapse BR tags* feature.

#### Type of Search

Searching type has two possible values: CASE INSENSITIVE (default) and CASE SENSITIVE. The searching type effects inter-document searches only, and determines whether searches for words within documents will be done in a case-sensitive or case-insensitive manner.

#### Use HTML5 charset replacements

This option allows lynx to treat pages with ISO-8859-1 (Latin1) or ASCII encoding as if they were Windows 1252. That allows a few punctuation characters to be shown.

## Use locale-based character set

This option allows you to request lynx to obtain a MIME name from the operating system which corresponds to your locale setting. If successful, it overrides the normal setting of the display character set.

## Underline links

Use underline-attribute rather than bold for links.

## Use Passive FTP

This allows you to change whether Lynx uses passive ftp connections.

## User-Agent header

The header string which Lynx sends to HTTP servers to indicate the User-Agent is displayed here. Changes may be disallowed via the *-restrictions* switch. Otherwise, the header can be changed temporarily to a string such as *L\_y\_n\_x/2.9.2* for access to sites which discriminate against Lynx based on checks for the presence of "Lynx" in the header. If the User-Agent header has been changed, it can be restored to the built-in default value by deleting the modified string in the Options Menu. Whenever the User-Agent header is changed, the current document is reloaded, with the no-cache flags set, on exit from the Options Menu. Changes of the header are not saved in the RC file.

*NOTE:* Some sites may regard misrepresenting the browser as fraudulent deception, or as gaining unauthorized access, if it is used to circumvent blocking that was intentionally put in place. Some browser manufacturers may find the transmission of their product's name objectionable. If you change the User-Agent string, it is your responsibility. The Options Menu issues a reminder whenever the header is changed to one which does not include "Lynx" or "L\_y\_n\_x".

## User Mode

There are four possible choices: Novice, Intermediate, Advanced, and Minimal.

### Novice

In Novice mode two lines of help are displayed at the bottom of the screen.

### Intermediate

Intermediate mode turns off the help lines.

### Advanced

Advanced mode displays the URL of the currently selected link at the bottom of the screen.

### Minimal

Minimal mode eliminates the URL on the bottom line, the forward/backward indicator in the upper left, and most status-line messages.

## Verbose Images

Controls whether or not Lynx replaces the [LINK], [INLINE] and [IMAGE] comments (for images without ALT) with filenames of these images. This is extremely useful because now we can determine immediately what images are just decorations (button.gif, line.gif) and what images are important. This setting can also be toggled on startup via the *-verbose* switch.

## VI keys

If set to ON then the lowercase h, j, k, and l keys will be mapped to left, down, up, and right arrow, respectively. The uppercase H, J, K, and L keys remain mapped to their configured bindings (normally HELP, JUMP, KEYMAP, and LIST, respectively).

Note: this has no effect on the line-editor's key bindings.

## Visited Pages

Enable several different views of the visited links:

- By First Visit
- By First Visit Reversed
- As Visit Tree
- By Last Visit
- By Last Visit Reversed

## X Display

[\[ToC\]](#)

## Comments and mailto: links

At any time while viewing documents within Lynx, you may use the “c” command to send a mail message to the owner of the current document if the author of the document has specified ownership.

Note to authors: if you want to assign the ownership to your document, you need to add into HEAD section a LINK element with appropriate value for the REL (HTML5) or REV (HTML4) attribute.

One value is recognized for REL: *author* (case-insensitive). For example:

```
<HEAD>
...
<LINK REL="author" HREF="mailto:user@somedomain.com">
...
</HEAD>
```

Two values are recognized for REV: *owner* and *made* (case insensitive). For example,

```
<HEAD>
...
<LINK REV="made" HREF="mailto:user@somedomain.com">
...
</HEAD>
```

You may also add a TITLE attribute with, for example, the name of your page) If no ownership is specified then comments are disabled. Certain links called [mailto:](#) links will also allow you to send mail to other people. Using the mail features within Lynx is straightforward.

Once you have decided to send a comment or have selected a *mailto:* link a new screen will appear showing you to whom you are sending the message. Lynx will ask for your name, your e-mail address, and the subject of the message. If you have filled in the "personal mail address" field in the *Options Menu*, your e-mail address will be filled in automatically. After entering the above information, if you have an editor defined in the *Options Menu* and you are not an anonymous user then your specified editor will be spawned for you so that you can enter your message. If you do not have an editor defined or you are an anonymous user, a simple line mode input scheme will allow you to enter your message.

To finish sending the message, exit your spawned editor or, if you are using the simple line mode input scheme, type a “.” (period) on a line by itself. You will be asked a final time whether to send the message. If you press “y”, you will be prompted whether to append your signature file if one was defined in [lynx.cfg](#) and is accessible, and then the message will be sent, whereas if you press “n” the message will be deleted. Entering Control-G in response to any prompts also will cancel the mailing.

[\[ToC\]](#)

## USENET News posting

While reading [news](#) articles with Lynx you should see a link that says *Reply to: user@host* and, if the nntp server from which you received the article supports posting from your site, a link that says *Followup to: newsgroup(s)*

Reply to user@host

user@host will correspond to the mail address of the person who posted the news article. Selecting the link will allow you to send a message to the person who wrote the message you are currently viewing. You will be given the option of including the original message in your reply.

Followup to newsgroup(s)

Selecting this link will allow you to post back to the newsgroup that you are currently reading and any newsgroups to which the message was cross-posted. You will be given the option of including the original message in your reply. Once you have typed in your message, you will be asked for confirmation of whether to proceed with the posting, and whether to append your signature file if one was defined in [lynx.cfg](#) and is accessible. See [Supported URLs](#) for more information about the URL schemes for posting or sending followups (replies) to nntp servers with Lynx. [\[ToC\]](#)

See also [RFC 977](#).

## Lynx bookmarks

Bookmarks are entries in your *bookmark file*, which record the URL of a document you may want to return to easily, with a name of your choice to identify the document. To use bookmarks you must first have specified a name for your bookmark file in [lynx.cfg](#) or via the *Options Menu*.

To save a bookmark to the document you wish to place in the bookmark file press the “a” key and you will be asked:

Save D)ocument or L)ink to bookmark file or C)ancel? (d,l,c):

Answer “d” to save a link to the document you are currently viewing or “l” to save the link that is currently selected on the page. Selecting “c” will cancel without saving anything to your bookmark file.

A bookmark file will be created in conjunction with acting on the “a”dd command if it does not already exist. Otherwise, the link will be added to the bottom of the pre-existing bookmark file. You must have created a bookmark file via the “a”dd command before you can view it.

Use the “v” command to view the list of bookmarks you have saved. While viewing the bookmark list you may select a bookmark as you would any other link.

You can remove a link from the bookmark list by pressing the “r” key when positioned on that link. You also can use a standard text editor (e.g., via the “e”dit command while viewing a bookmark file, if an external editor has been defined via the *Options menu*) to delete or re-order links in the bookmark file, or to modify a link name by editing the content of the Anchor element for the link, but you should not change the format within the line for the link, consisting of an *LI* element followed by the Anchor element, nor cause the line to become wrapped to a second line. You similarly can change the link destination by editing the double-quoted value for the *HREF* attribute in the Anchor start tag, but you should not otherwise change the spacing within the start tag, nor add other attributes. You can add a new link while editing by copying an existing line for a link, to ensure the proper format, and then modifying its *HREF* value and Anchor content, but you should not add any other HTML markup to the bookmark file. If the format and spacing (other than the Anchor content or *HREF* value) within lines is changed or other HTML markup is added, the “a”dd and “r”emove commands may not work properly.

When multi-bookmarks (see [Options Menu](#)) is OFF, you will always view or add links to the default bookmark file. When it is STANDARD, a menu of up to 26 bookmark files will be invoked, and you select the bookmark file by entering its letter token. When it is ADVANCED, you will be prompted for the letter token, but can enter “=” to invoke the STANDARD selection menu, or *RETURN* for the default bookmark file. [\[ToC\]](#)

## Jump Command

Similar to the bookmarks file is the jumps file: for an example, look in the *samples* subdirectory in the distribution package. To use the jumps command, create a *jumps file* with the same format as the sample file, but containing your own URLs & short-cut names. Once you have done that, typing “j” prompts you to enter a short-cut name, which will take you straight to the URL associated with the short-cut in the jumps file, much like using “g”. If you want to check which short-cuts are available, type “?” at the jump prompt for the full list.

You can set up a jumps file which makes Lynx prompt for parameters, e.g., as part of a search. Do this by putting a “%s” marker in the URL at each point where you want Lynx to fill in text. When you activate the corresponding jump, Lynx will prompt you for the parameters, one by one.

All jump short-cuts you have entered are saved in a circular buffer in the same way as with “g” and “/>” (search):

previous entries can be retrieved with *up-arrow* or *down-arrow*.

The jumps feature is especially useful for system administrators who have unsophisticated users to care for, but ordinary Lynx users who have a number of URLs they regularly visit while browsing may find using the jumps command speeds their movements.

For more advice how to set up the jumps command on your system and how to define short-cut names, read [lynx.cfg](#) .

[\[ToC\]](#)

## Directory Editing

Lynx offers extended Dired support on Unix (on VMS the more powerful CSwing program is recommended for character cell terminals, and can be offered via Lynx as a jump shortcut or execution link). When a local directory is accessed using a URL of the form *file://localhost/path/*, a new set of commands is available. With Dired support you can create, edit, delete, copy, and move files on your local system. The commands available in Dired mode are

### C)reate

Type “c” to create a new file. New file will be empty.

### D)ownload

Type “d” to download using one of the pre-defined options.

### E)dit

Type “e” to spawn the editor defined in *Options Menu* and load a selected file for editing.

### F)ull Menu

Type “f” to show full menu of options available for selection. Menu may vary according to type of file selected and compression facilities available.

### M)odify

Type “m” to modify the name or location of file. Then type “n” to rename the file or “l” to move the file to a different location.

### R)emove

Type “r” to remove the selected file or directory.

T)ag

Type “t” to tag highlighted file. Further operations will be performed on tagged files instead of highlighted ones.

U)pload

Type “u” to upload a file to the present directory. An uploading method must have been pre-defined in [lynx.cfg](#) .

[\[ToC\]](#)

## Using Color & the Mouse

A limited range of colors & mouse commands are available, if the user chooses: see *lynx.cfg* for details. [\[ToC\]](#)

## Scrolling and Other useful commands

A summary of all the keystroke commands and their key bindings can be invoked via the KEYMAP command, normally mapped to “k” and “K”. The following describes some of the most commonly used commands.

^A

*Control-A* jumps you to the beginning of the current document. It is a synonym for the Keypad *Home* key, and can be used also when *Links are numbered* mode is on. The *Find* Function key also is a synonym, and ideally the latter has been mapped to the Function key labeled *Home* if you are using an IBM Enhanced Keyboard.

^E

*Control-E* jumps you to the end of the current document. It is a synonym for the Keypad *End* key, and can be used also when *Links are numbered* mode is on. The *Select* Function key also is a synonym, and ideally the latter has been mapped to the Function key labeled *End* if you are using an IBM Enhanced Keyboard.

^B

*Control-B* normally jumps you to the previous page of the current document, and thus is a synonym for the Keypad and Function *Page-Up* keys. However, *Control-B* acts as *right-arrow* when emacs-like key movement is enabled (see [Lynx Options Menu](#)).

^F

*Control-F* normally jumps you to the next page of the current document, and thus is a synonym for the Keypad and Function *Page-Down* keys. However, *Control-F* becomes *right-arrow* when emacs-like key movement is enabled.

^N

*Control-N* normally jumps you forward two lines in the current document. The VT220 *Remove* Function key (labeled *Delete* on IBM Enhanced keyboards, and distinct from their *Backspace* key) is a synonym. *Control-N* becomes *down-arrow* when emacs-like key movement is enabled.

^P

*Control-P* normally jumps you back two lines in the current document. The *Insert* Function key is a synonym. *Control-P* becomes *up-arrow* when emacs-like key movement is enabled.

^K

*Control-K* invokes the [Cookie Jar Page](#) if it contains cookies.

^T

*Control-T* toggles Lynx trace mode on and off. This is useful for diagnosing bad html. If you get a *Bad HTML* statusline message when loading a document, enter *Control-T* and then *Control-R* to reload the document in trace mode. You may then examine the *Lynx Trace Log* file with the ; command if enabled (see below), watch out especially for lines marked with a number of asterisks “\*\*\*\*\*”. You also can submit the document for validation via links in the online help menu. If you are able to diagnose the problem, send a message about it to the document's author.

^X

*Control-X* invokes the [Cache Jar Page](#) if it contains cached documents.

E



The “E” command allows you to edit the URL (or ACTION) of the current link and then use that as a goto URL. Pressing the “E” command will bring up a prompt asking you to edit the current link’s URL. If you do not modify it, or completely delete it, or enter Control-G, the command will be cancelled. Otherwise, the request for the “E”dited URL will be sent with method GET, and will be entered into the circular buffer for goto URLs so that it can be accessed for further modification via the “g” command. Note that lower case “e” invokes the external editor for the current document.

g

The “g” command allows any URL to be viewed. Pressing the “g” command will bring up a prompt asking for a URL. Type in the URL that you wish to view. All previously entered goto URLs are saved in a circular buffer, and can be accessed at the prompt by pressing the *up-arrow* or *down-arrow* keys.

G

The “G” command allows you to edit the URL of the current document and then use that as a goto URL. Pressing the “G” command will bring up a prompt asking you to edit the current document’s URL. If you do not modify it, or completely delete it, or enter Control-G, the command will be cancelled. If the current document has POST content associated with it, an Alert will be issued. If you do edit that URL, and it does not simply involve a fragment change (for seeking a position in the current document), the modified URL will be submitted with method GET and no POST content. If a modification of the current document’s URL results in a submission, that modified URL will be entered into the circular buffer for goto URLs, and can be accessed for further modification via the “g” command.

z

Lynx supports completely interruptible I/O processes. Press the “z” key at any time during a connect or transfer process and the process will be halted. If any data was transferred before the interrupt, it will be displayed.

)

The ) command jumps you forward half a page in the current document.

(

The ( command jumps you back half a page in the current document.

#

The “#” command jumps you to the pseudo Toolbar or Banner if present in the current document. Use *left-arrow* to return from there to your previous position in the document.

!

When “!” is pressed your default shell will be spawned. When you quit or exit the shell you will return to Lynx (usually *exit* under Unix and *logout* under VMS). This command is usually disabled for anonymous users. On VMS, “\$” normally is a synonym.

=

The “=” command shows information about the current document and the currently selected link if there is one. The number of lines in the file, URL, title, owner, and type are shown.

Normally the information is shown formatted (with margins) for readability. You can make Lynx show the URL wrapped without margins, e.g., making it convenient for select/paste, by doing this:

- toggle line-wrapping off using “|”
- when line-wrapping is off, use the “=” command

;

The ; command shows the *Lynx Trace Log* (*Lynx.trace* in the home directory) if one has been started for the current session. If a log has not been started, any trace messages will be sent to the screen (and will disturb the normal display) unless the system supports piping and that was used to redirect stderr messages to a file. The log is started when Lynx trace mode is turned on via the *-trace* command line switch, or via the *Control-T* toggle, if Lynx has been compiled to log the trace and other stderr messages by default. If not, ability to create a log can be toggled on with the *-tlog* switch. Note that this ability is probably disabled in anonymous or validation accounts.

\*

The “\*” command toggles image\_links mode on and off. When on, links will be created for all images, including inline images. If you have an image viewer mapped to the image’s MIME type, you can activate such links to view an inline image. You should normally have this mode toggled off.

@

The “@” command toggles raw 8-bit or CJK mode on and off. When on, the charset is assumed to match the selected character set and 8-bit characters are not reverse translated with respect to the ISO-8859-1 conversion tables.

[

The “[” command toggles pseudo\_inlines mode on and off. When on, inline images which have no ALT string specified will have an *[INLINE]* pseudo-ALT string inserted in the Lynx display. When off, they will be treated as having ALT="" (i.e., they will be ignored). If image\_links mode is toggled on, the pseudo-ALT strings will be restored, to serve as links to the inline images’ sources.

]

The “]” command is used to send HEAD requests for the current document or link. It applies only to documents or links (or form submit buttons) of http servers. A statusline message will notify you if the context for this command was inappropriate. The HEAD requests always are sent to the http server, i.e., Lynx does not retrieve any previous server replies from its cache. Note that for form submissions, http servers vary in whether they'll treat HEAD requests as valid and return the CGI script's headers, or treat it as invalid and return an error message.

If the line-wrapping margin is wider than the terminal's display, scroll left by half of the display's width.

This feature is not available when Lynx is built using the slang library.

“|” toggles Lynx line-wrapping on/off. Normally Lynx fits text onto the screen, wrapping lines. With this feature, Lynx provides the ability to eliminate line-wrapping (up to an internal line-limit of 1000 characters). Lynx uses the curses “pad” feature to support left/right scrolling. You can scroll left and right in the screen to view the wide lines.

The popup menu for the command shows the other choices which extend the wrapping margin:

```
/-----\  
| Try to fit screen width      |  
| No line wrap in columns     |  
| Wrap columns at screen width|  
| Wrap columns at 3/4 screen width|  
| Wrap columns at 2/3 screen width|  
| Wrap columns at 1/2 screen width|  
| Wrap columns at 1/3 screen width|  
| Wrap columns at 1/4 screen width|  
\-----/
```

This feature is not available when Lynx is built using the slang library.

If the line-wrapping margin is wider than the terminal's display, scroll right by half of the display's width.

This feature is not available when Lynx is built using the slang library.

## numbers

Lynx offers other, advanced navigation features when numbers are used to invoke the [Follow Link \(or goto link or page\) number:](#) or [Select Pop-up Option Number:](#) prompts.

[\[ToC\]](#)

## Lynx and HTML Forms

This section describes the Lynx Forms Interface. HTML gives document providers the ability to create on-line forms which may be filled out when the document is viewed. When a form is submitted the information on the form can be used to search a database or complete a survey.

An HTML Form provides for the use of buttons to perform an action (such as *submit*), checkboxes, radio buttons or popups to select options from a list, and fields for entering text.

### Buttons:

Buttons are displayed in the same way that Lynx displays links in a document. To "push" the button press the *right-arrow* or *Return* key. If it is a form submission button, you also can use the NOCACHE (“x”) or DOWNLOAD (“d”) keystroke commands to "push" the button (see below).

### Checkboxes and Radio buttons

Checkboxes are displayed as square brackets: [ ] and radio buttons are displayed as parenthesis: ( ). When a box is checked or a button selected, an x appears in the brackets: [x] or an asterisk appears within the parenthesis: (\*). To check a box or select a radio button press the *right-arrow* or *Return* key.

### Selection Fields

Selection fields are displayed as brackets with the default option displayed between them: [default\_\_]. To select an option press the *right-arrow* or *Return* key. A box with a border of asterisks (or line-drawing characters) will pop up with the list of possible options listed within the box. Use the *up-arrow*, *down-arrow*, *page-up*, *page-down*, and other navigation keys to move the cursor among options, and the *right-arrow* or *Return* key to select an option. You also can use the “/” and “n”ext [searching](#) commands for navigating to options which contain particular strings. *NOTE* that the popup menu feature can be disabled via compilation and/or configuration options, or via the [Options Menu](#), in which case the selection field options will be converted to a list of radio buttons. The default setting for use of popups or radio button lists can be toggled via the *-popup* command line switch.

Text entry (INPUT) fields are displayed as a row of underscores the length of the entry field: \_\_\_\_\_. You may enter text directly by typing at the keyboard. Use the [Line Editor](#) keys to correct errors. If you try to input more text than the field can hold, the line editor will not accept the additional characters. If you fill a text field the cursor will not move off the field but remain at the last field position. Use the *up-arrow*, and *down-arrow*, *TAB* or *Return* keys to move up, or down from the text entry field. NOTE, however, that *Return* also will [submit](#) the form if the text entry field is the only non-hidden field in the form. If "Textfields Need Activation" mode is turned on (with the `-tna` command-line option or in [lynx.cfg](#)), then text entry fields do not become active immediately upon being selected, as normally. Keystrokes have their normal command meaning unless the Line Editor gets activated with *Return* or *Right Arrow*. This mode can be used to avoid "getting stuck" in input fields, especially by users who rarely fill out forms.

NOTE: If you have a text input field selected you will not have access to most of the Lynx keystroke commands, because they are interpreted by the [Line Editor](#) as either text entries or editing commands. Select a button or box when you want to use Lynx keystrokes; or prefix your keystroke with `^V` to temporarily escape from line editing.

Some flavors of UNIX, shells & terminal settings require that you enter `^V^V` in order to start the external editor, as they also use `^V` as default command-line quote key (called "lnext" in stty man pages and "stty -a" output); to avoid this, you can put "stty lnext undef" in your `.cshrc` file (or `.profile` or `.bashrc`, depending on what shell you use), or invoke Lynx with a wrapper script, e.g.

```
#!/bin/sh
stty lnext undef
$HOME/bin/lynx "$@"
stty lnext ^V
exit
```

NB when NOT in the Line Editor, `^V` is by default bound to the command to switch between SortaSGML and TagSoup HTML parsing (i.e., `SWITCH_DTD`). To avoid confusion, either of these separate functions could be changed (mapped away) with a `KEYMAP` directive in `lynx.cfg`. For example,

```
KEYMAP:^V:DO_NOTHING
KEYMAP:#:SWITCH_DTD
```

would map `SWITCH_DTD` away from `^V` to `#`, while leaving its default Line Editor function as a command escape in place. On the other hand,

```
KEYMAP:^V::NOP:1
KEYMAP:^_:LKCMD:1
```

would move `^V`'s Line Editor binding as command escape to `^_` for the first Line Edit style, letting `^V` still act as `SWITCH_DTD` outside of text input fields.

## TEXTAREA Fields

TEXTAREA fields are for most purposes handled as if they were a series of text entry (INPUT) fields for which successive lines imply a newline at the end of the preceding line. You enter text on each line to construct the overall message. Any blank lines at the bottom of the TEXTAREA field will be eliminated from the submission. The *up-arrow*, and *down-arrow* or *Return* keys move you to the preceding, or next line of the overall message, as for INPUT fields. The *TAB* key will move you down beyond the bottom of the TEXTAREA field, and *Back Tab* (if available, e.g., as Shift-Tab, and correctly mapped in the terminal description) will move backward to a link or field before the TEXTAREA.

## Editing TEXTAREA Fields and Special TEXTAREA Functions

TEXTAREA fields can be edited using an external editor. The statusline should tell you when this is possible and what key to use, it might for example say

**(Textarea) Enter text. [ . . . . . ] (^Xe for editor).**

An external editor has to be defined, for example in the [Options Menu](#), before you can start using this function.

A key to invoke external TEXTAREA editing is normally provided by the [Line-Editor Key](#) Bindings. A `KEYMAP` directive in `lynx.cfg` can also be used to make a different key invoke external editing; it will then normally be necessary to prefix that key with `^V` to "escape" from line-editing. Two variants exist,

```
KEYMAP:e:EDITTEXTAREA
```

or

```
KEYMAP:e:DWIMEDIT
```

(the first is only functional for TEXTAREA editing, while the second allows to use the same key for normal [file editing](#) as long as both functions do not conflict).

Please see the [note above](#) for details about `^V` behavior.

You can also use two other special TEXTAREA functions. Again, these are already bound to key sequences in the [Line-Editor Bindings](#), by default `^Xg` and `^Xi`. You can use different keys by adding `KEYMAP` bindings to your `lynx.cfg` file, e.g.

```
KEYMAP:$:GROWTEXTAREA
KEYMAP:#:INSERTFILE
```

With these bindings, (in a TEXTAREA only) `^V$` would add 5 lines to the TEXTAREA and `^V#` would prompt for the name of an existing file to be inserted into the TEXTAREA (above the cursorline). An automatic variation of `GROWTEXTAREA` is normally compiled in, so that hitting *Enter* with the cursor on the last line adds a new line to the TEXTAREA, with the cursor on it.

If you have some keys (or control keys) to spare that you need for the special functions (without requiring a prefix key). For example, to use the `^E` key for the DWIMEDIT action, and the `Insert` key for the INSERTFILE action, use  
KEYMAP:^E:DWIMEDIT:PASS  
KEYMAP:0x10C:INSERTFILE:PASS  
(see `lynx.cfg` for other keystroke codes to use).

Note that the default bindings that use `^X` as a prefix key *may* also work by substituting the `Escape` key for `^X`. If your keyboard has a modifier (Meta) key that gets transmitted as an ESC prefix, for example `Alt`, you can then even use `Alt-e` instead of `^Xe`, `Alt-g` instead of `^Xg`, and so on. But this does not work reliably everywhere (it depends on the way Lynx is compiled, including which libraries are used, and behavior of the connection and terminal type).

In general, you can move around the form using the standard Lynx navigation keys. The *up-arrow* and *down-arrow* keys, respectively, select the previous or next field, box, or button. The `TAB` key selects the next field, box, or button.

To *submit* the form press *right-arrow* or *Return* when positioned on the form's submit button. If you've submitted the form previously during the Lynx session, have not changed any of the form content, and the METHOD was `GET`, Lynx will retrieve from its cache what was returned from the previous submission. If you wish to resubmit that form to the server with the same content as previously, use the `NOCACHE` command ("`x`") when positioned on the submit button. The *right-arrow* and *Return* keys also will invoke a no-cache resubmission if the reply from a form submission included a `META` element with a no-cache `Pragma` or `Cache-Control` directive:

```
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">
```

or the server sent a "Pragma" or "Cache-Control" MIME header with a no-cache directive.

You also can use the `DOWNLOAD` ("`d`") keystroke command when positioned on a form submit button if you wish to download the server's reply to the submission instead of having Lynx render and display it.

Forms which have `POST` as the METHOD, or a [mailto:](#) URL as the ACTION, are always resubmitted, even if the content has not changed, when you activate the *submit* button. Lynx normally will not resubmit a form which has `POST` as the METHOD if the document returned by the form has links which you activated, and then you go back via the `PREV_DOC` (*left-arrow*) command or via the [History Page](#). Lynx can be compiled so that it resubmits the form in those cases as well, and the default can be changed via `lynx.cfg`, and toggled via the `-resubmit_posts` command line switch.

If the form has one *text entry* field and no other fields except, possibly, hidden `INPUT` fields not included in the display, then that field also serves as a *submit* button, and pressing *right-arrow* or *Return* on that field will invoke submission of the form. Be sure to use *up-arrow*, *down-arrow* or `TAB` to move off the text entry field, in such cases, if it is not your intention to submit the form (or to retrieve what was returned from an earlier submission if the content was not changed and the METHOD was `GET`).

Forms can have multiple *submit* buttons, if they have been assigned `NAME`s in the markup. In such cases, information about which one of the buttons was used to submit the form is included in the form content.

Inlined images can be used as submit buttons in forms: If such buttons are assigned `NAME`s in the markup, for graphic clients they can also serve as [image maps](#), and the x,y coordinates of the graphic client's cursor position in the image when it was *clicked* are included in the form content. Since Lynx cannot inline the image, and the user could not have moved a cursor from the origin for the image, if no alternatives are made available in the markup Lynx sends a 0,0 coordinate pair in the form content.

Document authors who use images as submit buttons, but have at least some concern for text clients and sight-challenged Webizens, should include `VALUE`s for the buttons in such markup. Lynx will then display the string assigned to the `VALUE`, as it would for a normal submit button.

- Some document authors incorrectly use an `ALT` instead of `VALUE` attribute for this purpose. Lynx "cooperates" by treating `ALT` as a synonym for `VALUE` when present in an `INPUT` tag with `TYPE="image"`.
- If neither a `VALUE` nor an `ALT` attribute is present, Lynx displays "[IMAGE]-Submit" as the string for such buttons.
- If clickable images is set, the "[IMAGE]" portion of the string is a link for the image, and the "Submit" portion is the button for submitting the form.

Otherwise, the entire string is treated as a submit button. If a `VALUE` or `ALT` attribute is present and clickable images is set, Lynx prepends "[IMAGE]" as a link for the image, followed by "`-`" and then the attribute's value as the displayed string for the submit button.

Early versions of Lynx would send a `name=value` pair instead of a 0,0 coordinate pair if a `TYPE="image"` submit button was `NAME`-ed, had a `VALUE` attribute in the `INPUT` tag, and was used to submit the form. The script which analyzes the form content thus could be made aware whether the submission was by a user with a graphic client and had image loading turned on, or by a user who did not see the image nor make a conscious choice within it. However, requests that this be included in HTML specifications consistently have fallen on deaf ears, and thus Lynx now "fakes" a 0,0 coordinate pair whether or not a `VALUE` or `ALT` attribute is present in the `INPUT` tag. Ideally, the script which analyzes the submitted content will treat the 0,0 coordinate pair as an indicator that the user did not see the image and make a conscious choice within it.

Forms can have *hidden* `INPUT` fields, which are not displayed, but have `NAME`s and `VALUE`s included in the content. These often are used to keep track of information across a series of related form submissions, but have the potential for including information about the user that might be considered to represent an invasion of privacy. NOTE, in this regard, that Lynx has implemented the [HTML 3.0 DISABLED](#) attribute for *all* of its form fields. These can be used to keep track of information across submissions, and to cast it unmodifiable in the current form, but keep the user aware that it will be included in the submission.

Forms most commonly are submitted to http servers with the content encoded as `ENCTYPE="application/x-www-form-urlencoded"` for analysis by a script, and Lynx treats that as the default if no `ENCTYPE` is specified in the `FORM` start tag. However, you can specify a [mailto](#) URL as the form's ACTION to have the form content sent, instead, to an email address. In such cases, you may wish to specify `ENCTYPE="text/plain"` in the form markup, so that the content will not be encoded, but remain readable as plain text.

Lynx also supports `ENCTYPE="application/sgml-form-urlencoded"` for which all reserved characters in the content will be hex escaped, as with `application/x-www-form-urlencoded`, but semicolons ("`;`") instead of ampersands ("`&`") will be used as the separator for `name=value` pairs in the form content. The use of semicolons is preferred for forms with the `GET` METHOD, because the `GET` METHOD causes the encoded form content to be appended as a *searchpart* for the form's ACTION, and if such URLs are used in *text/html* documents or bookmark files without conversion of the ampersands to SGML character references (`&amp;` or `&#38;`), their being followed by form field `NAME`s which might correspond to SGML entities could lead to corruption of the intended URL.

NOTE, in this regard, Lynx converts ampersands to &amp; when creating bookmarks, and thus the bookmark links do not be vulnerable to such corruptions. Also NOTE that Lynx allows you to save links in your bookmark file for documents returned by forms with the GET METHOD, and which thus have the content appended as a ?searchpart, but not if the METHOD was POST, because the content would be lost and the link thus would be invalid.

Lynx supports ENCTYPE="multipart/form-data" for sending form content with name=value pairs encoded as multipart sections with individual MIME headers and boundaries. However, Lynx does not yet support INPUTs with TYPE="file" or TYPE="range" and will set the DISABLED attribute for all of the form's fields if any INPUTs with either of those two TYPES are present, so that the form cannot be submitted. Otherwise, Lynx will submit the form with the multipart ENCTYPE.

A Content-Disposition: file; filename=name.suffix header can be used by CGI scripts to set the suggested filename offered by Lynx for "d"ownload and "p"rint menu options to save or mail the body returned by the script following submission of a FORM. Otherwise, Lynx uses the last symbolic element in the path for the FORM's ACTION, which is normally the script, itself, or a PATH\_INFO field, and thus might be misleading. This also can be done via a META element in any document:

```
<META HTTP-EQUIV="Content-Disposition"
      CONTENT="file; filename=name.suffix">
```

[ToC]

## Lynx and HTML Images

As a text browser, Lynx does not display images as such -- you need to define a viewer in lynx.cfg: see there -- , but users can choose a number of ways of showing their presence.

There are 3 choices in lynx.cfg, with 2 corresponding keys:

MAKE_LINKS_FOR_ALL_IMAGES	* IMAGE_TOGGLE
MAKE_PSEUDO_ALTS_FOR_INLINES	[ INLINE_TOGGLE
VERBOSE_IMAGES	no corresponding key

You can also use the Options Menu, as outlined below:

key	lynx.cfg	FM	KM	.lynxrc	variable in source
*	MAKE_LINKS_	Y	N	N	clickable_images
[	MAKE_PSEUDO_	Y	N	N	pseudo_inline_alts
	VERBOSE_	Y	Y	Y	verbose_img

FM = Form-based Menu ; KM = Key-based Menu ;  
in .lynxrc , VERBOSE\_IMAGES is called "verbose\_images":  
the other two cannot be saved between sessions.

In the Form-based Menu, the 3-way "Show images" selection combines the effects of the "\*" & "[" keys, as follows:

Ignore	clickable_images = FALSE, pseudo_inline_alts = FALSE
As labels	clickable_images = FALSE, pseudo_inline_alts = TRUE
As links	clickable_images = TRUE, pseudo_inline_alts = unchanged

## Lynx and HTML Tables

HTML includes markup for creating tables structured as arrays of cells aligned by columns and rows on the displayed page.

Lynx recognizes the TABLE element and all of its associated elements as described in RFC 1942 and will process any ID attributes in the start tags for handling as NAME-ed anchors, but does not create actual tables. Instead, it treats the TR start tag as a collapsible BR (line break), and inserts a collapsible space before the content of each TH and TD start tag. This generally makes all of the content of the table readable, preserves most of the intra-cell organization, and makes all of the links in the table accessible, but any information critically dependent on the column and row alignments intended for the table will be missed.

If inherently tabular data must be presented with Lynx, one can use PRE formatted content, or, if the table includes markup not allowed for PRE content, construct the table using HTML Tabs. An example table using TAB elements is included in the test subdirectory of the Lynx distribution.

Starting with version 2.8.3, Lynx renders some tables in tabular form. This tabular representation for simple tables (TRST) does not attempt to implement full support for any table model. Limitations are:

- All data constituting a table row generally has to fit within the display width without inserting line breaks.
- Cell contents have to be simple. In general, only inline markup is acceptable, no <P>, <BR> etc. (although <BR> may be ignored at the beginning of the first cell or at the end of the last cell of a row).
- When tables are nested, only the innermost level is a candidate for tabular representation.
- Most attributes are ignored, including borders, WIDTH, vertical alignment.

Horizontal alignments (LEFT, CENTER, RIGHT), COLSPAN, and ROWSPAN are interpreted according to HTML 4.01. (ROWSPAN can only reserve empty space in subsequent rows, because of the limitations above.) When TRST fails because a table is not "simple" enough, the representation falls back to the minimal handling described earlier. Many (but, unfortunately, by no means all) tables that represent inherently tabular material will thus be shown with

correct tabular formatting. Where table markup is used only for layout purposes (containing whole blocks of text and list within table cells) and not essential for understanding the textual contents, it remains basically ignored. Some more information on details is available in the file `README . TRST` of the source distribution.

For tabular display of more complex tables, Lynx users can make use of external scripts or programs. The normal Lynx distribution currently does not provide such scripts, but they can be written locally or downloaded from several sources. It is suggested to use one of Lynx's facilities for invoking external programs (see `DOWNLOADER`, `PRINTER`, `EXTERNAL`, `TRUSTED_LYNXCGI` in [lynx.cfg](#) and [lynxcgi](#): in *Supported URLs* for information on various ways for setting this up).

[\[ToC\]](#)

## Lynx and HTML Tabs

Lynx implements the [HTML 3.0](#) `TAB` element only when `LEFT` alignment is in effect. If the alignment is `CENTER` or `RIGHT` (`JUSTIFY` is not yet implemented in Lynx, and is treated as a synonym for `LEFT`), or if the `TAB` element indicates a position to the left of the current position on the screen, it is treated as a collapsible space. For purposes of implementing `TAB`, Lynx treats *en* units as half a character cell width when specified by the `INDENT` attribute, and rounds up for odd values (e.g., a value of either 5 or 6 will be treated as three spaces, each the width of a character cell). See the example *table* using `TAB` elements in the test subdirectory of the Lynx distribution as a model for using this functionality.

Note that this *Users Guide* and the [Supported URLs](#) page include `TAB` markup in a manner which *degrades gracefully* for WWW browsers which do not support it. Toggle to display of [source](#) and [search](#) for `<tab` to examine the use of `TAB` markup in these documents.

[\[ToC\]](#)

## Lynx and HTML Frames

Some implementations of HTML include markup, primarily designed for graphic clients, that is intended to create an array of simultaneously displayed, independently scrolling windows. Such windows have been termed *frames*.

Lynx recognizes the Netscape and Microsoft Explorer `FRAME`, `FRAMESET`, and `NOFRAMES` elements, but is not capable of windowing to create the intended positioning of *frames*. Instead, Lynx creates labeled links to the *frame* sources, typically positioned in the upper left corner of the display, and renders the `NOFRAMES` section. If the document provider has disregard for text clients and sight-challenged Webizens, and thus does not include substantive content in the `NOFRAMES` section or a link in it to a document suitable for text clients, you can usually guess from the labeling of the *frame* links which one has the substantive material (if there is any), or you can try each of those links to see if anything worthwhile is returned.

[\[ToC\]](#)

Some sites -- in ignorance of Lynx capabilities -- may tell you (for example) "to view this page you need Netscape Navigator". You can simply ignore such warnings and access the frames via the Lynx-generated links as above.

## Lynx and HTML Banners

Some implementations of HTML markup include provisions for creating a non-scrolling window to be positioned at the top of each page, containing links with brief, descriptive link names, analogous to a Windows toolbar. Such windows have been termed *banners*.

Lynx recognizes and processes all of the [HTML 3.0](#) `REL` attribute tokens in `LINK` elements for creating a *banner*, and a number of others which have subsequently been proposed. These *banner* tokens are

*Home, ToC, Contents, Index, Glossary, Copyright, Up, Next, Previous, Prev, Help, Search, Top, Origin, Navigator, Child, Disclaimer, Sibling, Parent, Author, Editor, Publisher, Trademark, Meta, URC, Hotlist, Begin, First, End, Last, Pointer, Translation, Definition, Chapter, Section, Subsection, Alternate, Documentation, Biblioentry, Bibliography, Start, Appendix, Bookmark and Banner.*

Any `LINK` elements with those tokens as the `REL` attribute value, and an `HREF` attribute value in the `LINK`, will invoke creation of a *banner* at the top of the first page, with the element's `HREF` as the link, and the token as the default link name. If a `TITLE` attribute is included in the `LINK`, its value will be used as the link name instead of the default. *Bookmark* and *Banner* are intended to be accompanied by a `TITLE` attribute, which in effect makes the namespace for `REL banner` tokens infinite.

If the special token *Help* is used as the `REL` value and no `HREF` is included in the `LINK`, Lynx will use its own *HELPPFILE* URL for that link. For the special token *Home* without an `HREF`, Lynx will use the default *STARTFILE* (i.e., derived from the configuration files or the `WWW_HOME` environment variable, *not* the command line *startfile* if one was used). However, if a *-homepage=URL* was specified on the command line, its URL will be used as the `HREF`. For the special token *Index* without an `HREF`, Lynx will use the *DEFAULT\_INDEX\_FILE* derived from the configuration files, or if an *-index=URL* was specified on the command line, its URL will be used as the `HREF`.

Lynx does not waste screen real estate maintaining the *banner* at the top of every page, but the Lynx `TOOLBAR` keystroke command ("`#`") will, any time it is pressed, position you on the *banner* so that any of its links can be activated, and pressing the *left-arrow* when in the *banner* will return you to where you were in the current document. The toolbar is indicated by a "`#`" preceding its first link when present on the screen, that is, when the first page of the document is being displayed. The availability of a toolbar is indicated by a "`#`" at the top, left-hand corner of the screen when the second or subsequent pages of the document are being displayed.

Lynx also recognizes the [HTML 3.0](#) `BANNER` container element, and will create a *banner* based on its content if one has not already been created based on `LINK` elements. Lynx treats the Microsoft `MARQUEE` element as a synonym for `BANNER` (i.e., presenting its markup as a static *banner*, without any horizontal scrolling of its content). Lynx does not prefix the `BANNER` or `MARQUEE` content with a "`#`" because the content need not be only a series of links with brief, descriptive link names, but does add a "`#`" at the top, left-hand corner of the screen when the content is not being displayed, to indicate its accessibility via the `TOOLBAR` keystroke command.

[\[ToC\]](#)

## Lynx and HTML Footnotes

Lynx implements the [HTML 3.0](#) FN element similarly to a named Anchor within the current document, and assumes that the footnotes will be positioned at the bottom of the document. However, in contrast to named Anchors, the FN container element is treated as a block (i.e., as if a new paragraph were indicated whether or not that is indicated in its content) with greater than normal left and right margins, and the block will begin with a *FOOTNOTE:* label. For example, if the document contains:

```
See the <A HREF="#fn1">footnote</A>.
```

activating that link will take you to the labeled rendering of:

```
<FN ID="fn1"><p>Lynx does not use popups for FN blocks.</p></FN>
```

i.e., position it at the top of the page. Then, upon reading the footnote, you can return to your previous position in the document by pressing the *left-arrow* key. The content of an FN element can be any HTML markup that is valid in the BODY of the document.

[\[ToC\]](#)

## Lynx and HTML Notes

Lynx implements the [HTML 3.0](#) NOTE element (*Admonishment*) as a labeled block, i.e., as if a new paragraph were indicated whether or not paragraphing markup is included in its content, with greater than normal left and right margins, and with the type of note indicated by an emphasized label based on the value of its CLASS or ROLE attribute. If no CLASS or ROLE attribute is included, the default label *NOTE:* will be used. Lynx recognizes the values *caution* and *warning*, for which, respectively, the labels *CAUTION:* or *WARNING:* will be used. The NOTE element can have an ID attribute, which will be treated as a named Anchor, as for [HTML Footnotes](#), but the NOTE block need not be placed at the bottom of the document. The content of a NOTE block can be any HTML markup that is valid in the BODY of the document. This is an example:

```
<NOTE CLASS="warning" ID="too-bad">
  <p>The W3C vendors did not retain NOTE in the HTML 3.2 draft.</p>
</NOTE>
```

It will *degrade gracefully* for WWW browsers which do not support NOTE, except for recognition of the ID attribute as a named Anchor.

[\[ToC\]](#)

## Lynx and HTML Lists

Lynx implements the [HTML 3.0](#) list elements UL (*Unordered List*), OL (*Ordered List*), and DL (*Definition List*), and their associated attributes, and elements (LH, LI, DT, and DD) for the most part as described in that specification. The lists can be nested, yielding progressively greater indentation, up to six levels. The [HTML 2.0](#) MENU and DIR elements *both* are treated as synonyms for UL with the PLAIN attribute (no *bullets*, see below). Note, thus, that neither DIR nor MENU yields a series of columns with 24-character spacing. A single nesting index is maintained, so that different types of List elements can be used for different levels within the nest. Also, the [HTML 3.0](#) FIG, CAPTION and CREDIT elements are treated as valid within list blocks. They will be rendered with indentation appropriate for the current nesting depth, and the CAPTION or CREDIT elements will have a *CAPTION:* or *CREDIT:* label beginning the first line of their content. The content of any APPLET or OBJECT elements in the lists also will be indented appropriately for the current nesting depth, but those will not invoke line breaks unless indicated by their content, and it should not include markup which is inappropriate within the list.

Lynx also supports the TYPE attribute for OL elements, which can have values of 1 for Arabic numbers, *I* or *i* for uppercase or lowercase Roman numerals, or *A* or *a* for uppercase or lowercase letters, that increment for successive LI elements in the list block. The CONTINUE attribute can be used to continue the ordering from the preceding list block when the nesting depth is changed.

Lynx treats the OL attributes START and SEQNUM as synonyms for specifying the ordering value for the first LI element in the block. The values should be specified as Arabic numbers, but will be displayed as Arabic, Roman, or alphabetical depending on the TYPE for the block. The values can range from -29997 to the system's maximum positive integer for Arabic numbers. For Roman numerals, they can range from 1 (*I* or *i*) to 3000 (*MMM* or *mmm.*). For alphabetical orders, the values can range from 1 (*A* or *a*) to 18278 (*ZZZ* or *zzz*). If the CONTINUE attribute is used, you do not need to specify a START or SEQNUM attribute to extend the ordering from a previous block, and you can include a TYPE attribute to change among Arabic, Roman, or alphabetical ordering styles, or their casing, without disrupting the sequence. If you do not include a START, SEQNUM or CONTINUE attribute, the first LI element of each OL block will default to 1, and if you do not include a TYPE attribute, Lynx defaults to Arabic numbers.

For UL blocks without the PLAIN attribute, Lynx uses \*, +, o, #, @ and - as *bullets* to indicate, progressively, the depth within the six nesting levels.

Lynx treats UL, OL, DIR, and MENU blocks as having the COMPACT attribute by default, i.e., single spaces between LH and LI elements within those blocks. For DL blocks, double spacing will be used to separate the DT and DD elements unless the COMPACT attribute has been specified.

[\[ToC\]](#)

## Lynx and HTML Quotes

The [HTML 3.0](#) and later specifications provide for two classes of quotation in HTML documents. Block quotes, designated by the BLOCKQUOTE element (or its abbreviated synonym BQ in HTML 3.0), have implied paragraph breaks preceding and following the start and end tags for the block. Character level quotes, designated by the Q element, in contrast are simply directives in the markup to insert an appropriate quotation mark.

Lynx renders block quotes with a greater than normal left and right indentation. Lynx does not support italics, and normally substitutes underlining, but does not underline block quotes so as not to obscure any explicit emphasis elements within the quotation. The BLOCKQUOTE or BQ block can include a CREDIT container element, whose content will be rendered as an implied new paragraph with a *CREDIT:* label at the beginning of its first line.

Lynx respects nested Q start and end tags, and will use ASCII double-quotes (") versus grave accent (`) and apostrophe ('), respectively, for even versus odd depths in the nest.



Any ID attributes in BLOCKQUOTE, BQ or Q elements can be the target of a hyperlink in the form URL#id. It is treated just like the NAME in Anchors.

[\[ToC\]](#)

## Lynx and HTML Internationalization: 8bit, UNICODE, etc.

Lynx has superior support for HTML 4.0/I18N internationalization issues. However, to see the characters other than 7bit properly you *should* set your [display\\_character\\_set](#) from Option Menu and save its value, this is a Frequently Asked Question. Fine-tuning is also available from [lynx.cfg](#)

[\[ToC\]](#)

## Lynx and Client-Side-Image-Maps

HTML includes markup, designed primarily for graphic clients, that treats inlined images as maps, such that areas of the image within which a mouse cursor was positioned when the mouse was *clicked* can correspond to URLs which should be retrieved. The original implementations were based on the client sending an http server the x,y coordinates associated with the *click*, for handling by a script invoked by the server, and have been termed *server-side-image-maps*. Lynx has no rational way of coping with such a procedure, and thus simply sends a 0,0 coordinate pair, which some server scripts treat as an instruction to return a document suitable for a text client.

Newer HTML markup provides bases for the client to determine the URLs associated with areas in the image map, and/or for a text client to process alternative markup and allow the user to make choices based on textual information. These have been termed *client-side-image-maps*.

Lynx recognizes and processes the MAP container element and its AREA elements, and will create a menu of links for the HREF of each AREA when the link created for the IMG element with a USEMAP attribute is activated. The menu uses the ALT attributes of the AREA elements as the link names, or, if the document's author has disregard for text clients and sight-challenged Webizens, and thus did not include ALT attributes, Lynx uses the resolved URLs pointed to by the HREF attributes as the link names. Lynx uses the TITLE attribute of the IMG element, or the TITLE attribute of the MAP, if either was present in the markup, as the title and main header of the menu. Otherwise, it uses the ALT attribute of the IMG element. If neither TITLE nor ALT attributes were present in the markup, Lynx creates and uses a *[USEMAP]* pseudo-ALT. The MAPs need not be in the same document as the IMG elements. If not in the same document, Lynx will fetch the document which contains the referenced MAP, and locate it based on its NAME or ID attribute. All MAPs encountered in documents during a Lynx session are cached, so that they need not be retrieved repeatedly when referenced in different documents.

If the IMG element also indicates a *server-side-image-map* via an ISMAP attribute, Lynx normally will create a link for that as well, using an *[ISMAP]* pseudo-ALT (followed by a hyphen to indicate its association with the *client-side-image-map*) rather than ignoring it, and will submit a 0,0 coordinate pair if that link is activated. Although, the *client-side-image-map* may be more useful for a client such as Lynx, because all of the URLs associated with the image map can be accessed, and their nature indicated via ALT attributes, Lynx-friendly sites can map 0,0 such that the server returns a for-text-client document homologous to the content of FIG elements (see below). Inclusion of such a link for submissions to the server can be disabled by default via the configuration file ([lynx.cfg](#)), and the default can be toggled via the *-ismap* command line switch.

Lynx also recognizes the [HTML 3.0](#) FIG and OVERLAY elements, and will handle them as intended for text clients. These are the ideal way to handle *client-side-image-maps*, because the FIG content provides complete alternative markup, rather than relying on the client to construct a relatively meager list of links with link names based on ALT strings.

The presently experimental OBJECT element encompasses much of the functionality of the FIG element for *client-side-image-maps*. Lynx will render and display the content of OBJECT elements which have the SHAPES attribute equivalently to its handling of FIG. Lynx also handles OBJECT elements with the USEMAP and/or ISMAP attributes equivalently to its handling of IMG elements with *client-side-image-maps* and/or *server-side-image-maps*.

[\[ToC\]](#)

## Lynx and Client-Side-Pull

HTML includes provision for passing instructions to clients via directives in META elements, and one such instruction, via the token *Refresh*, should invoke reloading of the document, fetched from a server with the same URL or a new URL, at a specified number of seconds following receipt of the current document. This procedure has been termed *client-side-pull*. An example of such an element is:

```
<META HTTP-EQUIV="Refresh" CONTENT="3; URL=http://host/path">
```

which instructs a client to fetch the indicated URL in 3 seconds after receiving the current document. If the *URL=* field is omitted, the URL defaults to that of the current document. A *no-cache* directive is implied when the *Refresh* is for the same URL.

Lynx recognizes and processes *Refresh* directives in META elements, but puts up a labeled link, typically in the upper left corner of the display, indicating the number of seconds intended before a refresh, and the URL for the refresh, instead of making the request automatically after the indicated number of seconds. This allows people using a braille interface any amount of time to examine the current document before activating the link for the next URL. In general, if the number of seconds indicated is short, the timing is not critical and you can activate the link whenever you like. If it is long (e.g., 60 seconds), a server process may be generating new documents or images at that interval, and you would be wasting bandwidth by activating the link at a shorter interval.

[\[ToC\]](#)

## Lynx State Management (Me want cookie!)

HTTP provides a means to carry state information across successive connections between a browser and an http server. Normally, http servers respond to each browser request without relating that request to previous or subsequent requests. Though the inclusion of INPUT fields with TYPE="hidden" can be used as a sort of state management by [HTML Forms](#), a more general approach involves exchanges of MIME headers between the server and browser. When replying to a request, the server can send a *Set-Cookie* MIME header which contains information (*cookies*) relevant to the browser's request, and in subsequent requests the browser can send a *Cookie* MIME header with information derived from previously received cookies.



State Management via cookie exchanges originally was implemented by Netscape, and such cookies are now designated as *Version 0*. A more elaborate format for cookies, designated as *Version 1*, was standardized by the IETF (Internet Engineering Task Force) as [RFC 2109](#). Lynx supports both *Version 0* and *Version 1* cookie exchanges. This support can be disabled by default via the SET\_COOKIES symbol in the compilation (*userdefs.h*) and/or run time (*lynx.cfg*) configuration files, and that default setting can be toggled via the *-cookies* command line switch. The SET\_COOKIES symbol can be further modified by the ACCEPT\_ALL\_COOKIES mode. If ACCEPT\_ALL\_COOKIES is TRUE, and SET\_COOKIES is TRUE, Lynx will accept all cookies. Additionally, the cookies that are automatically accepted or rejected by Lynx can be further modified with the COOKIE\_ACCEPT\_DOMAINS and COOKIE\_REJECT\_DOMAINS options in your .lynxrc file, each of which is a comma-separated list of domains to perform the desired action. The domain listed in these options must be identical to the domain the cookie comes from, there is no wildcard matching. If a domain is specific in both COOKIE\_ACCEPT\_DOMAINS and COOKIE\_REJECT\_DOMAINS, rejection will take precedence.

When cookie support is enabled, *Set-Cookie* MIME headers received from an http server invoke confirmation prompts with possible replies of “Y”es or “N”o for acceptance of the cookie, “A”lways to accept the cookie and to allow all subsequent cookies from that *domain* (server's Fully Qualified Domain Name, or site-identifying portion of the FQDN) without further confirmation prompts, or ne“V”er to never allow cookies from that *domain* to be accepted (silently ignore its *Set-Cookie* MIME headers). All unexpired cookies are held in a hypothetical *Cookie Jar* which can be examined via the COOKIE\_JAR keystroke command, normally mapped to *Ctrl-K*, for invoking the [Cookie Jar Page](#). If Lynx has been compiled with the *--enable-persistent-cookies* flag, then unexpired cookies will be stored between sessions in the filename set with the COOKIE\_FILE option in your .lynxrc.

A common use of cookies by http servers is simply to track the documents visited by individual users. Though this can be useful to the site's WebMaster for evaluating and improving the organization of links in the various documents of the site, if the user has configured Lynx to include a *From* MIME header with the user's email address in http requests, or has passed personal information to the server via a form submission, the tracking might be used to draw inferences, possibly incorrect, about that user, and may be considered by some as an invasion of privacy.

An example of worthwhile State Management via cookies is the setting of personal preferences, typically via a form submission to the site, which will then apply to all documents visited at that site.

If you accept cookies when accessing a site, but are given no indication about how they will be used in subsequent requests to that site, nor can infer how they will be used, you can *Gobble* (delete) the cookies and/or change the “allow” setting for its *domain* via the [Cookie Jar Page](#).

[\[ToC\]](#)

## Cached Documents

A list of documents which are in lynx's internal cache is accessible through hypothetical *Cache Jar* which can be examined via the CACHE\_JAR keystroke command, normally mapped to *Ctrl-X*.

Entries in the *Cache Jar* are ordered from oldest (at the top) to newest. The user can easily access any document which is in the cache, especially those which may be soon removed due to configurable limits on the maximum number of cached documents, as well as the maximum amount of memory used by the cache.

The structure of *Cache Jar* is simple:

- Each entry starts with its ordinal number (within the session), recently added documents in cache have a smaller number than documents which are added before, and are positioned at the end of *Cache Jar*
- Following its ordinal number is the document title, which is also a link. On activating this link, the user is prompted if they want to delete the document from *Cache Jar*. The document's address (also a link) follows the title. It is distinguished by a URL : label preceding the link. Activating this link, lynx displays the corresponding cached document.
- Below each cached document URL lynx shows the document properties which include:
  - Lines,
  - Size,
  - File-Cache,
  - Content-Type,
  - Content-Language,
  - Content-Encoding,
  - Content-Location,
  - Subject,
  - Owner,
  - Date,
  - Expires,
  - Last-Modified,
  - ETag,
  - Server, and
  - Source-Cache-File.

This feature can be enabled by default using the USE\_CACHEJAR symbol in the compilation (*userdefs.h*), as well as enabled in *lynx.cfg*

[\[ToC\]](#)

## Sessions

Lynx's current state (all information about the user's current activity with lynx) is called a session. Sessions are useful in particular if you are in the middle of exploring something on the web and you were forced to stop abruptly, losing any trace of your current work.

A session can be automatically restored as lynx starts after a clean exit. The session data is saved if lynx is invoked with the *-session=FILENAME* switch. The *FILENAME* is the name of the file where the session will be stored.

There are also switches for only restoring: *-sessionin=FILENAME* and for only saving: *-sessionout=FILENAME* sessions:

If you do not want to specify these options at each lynx startup, there is an option in *lynx.cfg* to enable automatic saving/restoring of session. To keep lynx startup/exit reasonably fast there is also an option in *lynx.cfg* specifying how much information about the current lynx session will be stored in file.

The syntax of the session file is simple. You can use a text editor to modify, add new entries, or remove URLs you no longer want.

[\[ToC\]](#)

## The Lynx command line

A summary of the Lynx command line options (switches) is returned to stdout if Lynx is invoked with the *-help* switch. A description of the options also should be available via the system man (Unix) pages or help (VMS) libraries. On Win32, typing lynx -help in a DOS window should display similarly. The basic syntax of the Lynx command line can be represented as one of the following:

Command  
lynx [options]  
lynx [options] startfile

where

startfile

is the file or URL that Lynx will load at start-up.

- If startfile is not specified, Lynx will use a default starting file and base directory determined during installation.
- If a specified file is local (i.e., not a URL) Lynx displays that file and uses the directory in which that file resides as the base directory.
- If a URL is specified, the file will be retrieved, and only the server base directory will be relevant to further accesses.
- If more than one local file or remote URL is listed on the command line, Lynx will open only the last interactively. All of the names (local files and remote URLs) are added to the Goto history.

options

Lynx uses only long option names. Option names can begin with double dash as well, underscores and dashes can be intermixed in option names (in the reference below options are with one dash before them and with underscores).

Lynx provides many command-line options. Some options require a value (string, number or keyword). These are noted in the reference below. The other options set boolean values in the program. There are three types of boolean options: set, unset and toggle. If no option value is given, these have the obvious meaning: set (to true), unset (to false), or toggle (between true/false). For any of these, an explicit value can be given in different forms to allow for operating system constraints, e.g.,

```
-center:off  
-center=off  
-center-
```

Lynx recognizes "1", "+", "on" and "true" for true values, and "0", "-", "off" and "false" for false values. Other option-values are ignored.

The default boolean, number and string option values that are compiled into lynx are displayed in the help-message provided by lynx -help. Some of those may differ according to how lynx was built; see the help message itself for these values. The -help option is processed before any option, including those that control reading from the lynx.cfg file. Therefore runtime configuration values are not reflected in the help-message.

Capitalized items in the option summary indicate that a substitution must be made. These are the options:

-

If the argument is only "-" (dash), then Lynx expects to receive the arguments from stdin. This is to allow for the potentially very long command line that can be associated with the *-get\_data* or *-post\_data* arguments (see below). It can also be used to avoid having sensitive information in the invoking command line (which would be visible to other processes on most systems), especially when the *-auth* or *-pauth* options are used. On VMS, the dash must be encased in double-quotes ("-") and the keyboard input terminated with *Control-Z* or the command file input terminated by a line that begins with "\$". On Unix, the keyboard input terminator is *Control-D*. On Win32, [???].

### **-accept\_all\_cookies**

accept all cookies.

### **-anonymous**

apply restrictions appropriate for an anonymous account, see *-restrictions* below for some details.

### **-assume\_charset=MIMENAME**

charset for documents that do not specify it.

### **-assume\_local\_charset=MIMENAME**

charset assumed for local files, i.e., files which lynx creates such as internal pages for the options menu.

**-assume\_unrec\_charset=*MIMENAME***

use this instead of unrecognized charsets.

**-auth=*ID:PW***

set authorization *identifier* and *password* for protected documents at startup. Be sure to protect any script files which use this switch.

**-base**

prepend a request URL comment and BASE tag to text/html outputs for -source dumps.

**-bibp=*URL***

specify a local bibp server (default <http://bibhost/>).

**-blink**

forces high intensity background colors for color mode, if available and supported by the terminal. This applies to the slang library (for a few terminal emulators), or to OS/2 EMX with ncurses.

**-book**

use the bookmark page as the startfile. The default or command line startfile is still set for the Main screen command, and will be used if the bookmark page is unavailable or blank.

**-buried\_news**

toggles scanning of news articles for buried references, and converts them to news links. Not recommended because email addresses enclosed in angle brackets will be converted to false news links, and uuencoded messages can be trashed.

**-cache=*NUMBER***

set the *NUMBER* of documents cached in memory. The default is 10.

**-center**

Toggle center alignment in HTML TABLE.

**-case**

enable case-sensitive string searching.

**-cfg=*FILENAME***

specifies a Lynx configuration file other than the default [lynx.cfg](#).

**-child**

exit on left-arrow in startfile, and disable save to disk.

**-child\_relaxed**

exit on left-arrow in startfile, but allow save to disk and associated print/mail options.

**-cmd\_log=*FILENAME***

write keystroke commands and related information to the specified file.

**-cmd\_script=*FILENAME***

read keystroke commands from the specified file. You can use the data written using the *-cmd\_log* option. Lynx will ignore other information which the command-logging may have written to the log- file. Each line of the command script contains either a comment beginning with "#", or a keyword:

**exit**

causes the script to stop, and forces lynx to exit immediately.

## **key**

the character value, in printable form. Cursor and other special keys are given as names, e.g., **Down Arrow**. Printable 7-bit ASCII codes are given as-is, and hexadecimal values represent other 8-bit codes.

## **set**

followed by a "name=value" allows one to override values set in the lynx.cfg file.

## **-color**

forces color mode on. This feature is only available if Lynx is built using the slang library. The slang library will send ANSI color sequences without regard to the type of terminal which is being used.

If color support is instead provided by a color-capable curses library such as ncurses, Lynx relies completely on the terminal description to determine whether color mode is possible, and this flag is not needed and thus unavailable.

A saved `show_color=always` setting found in a `.lynxrc` file at startup has the same effect, but the setting read from `.lynxrc` on startup is overridden by this flag.

## **-connect\_timeout=N**

Sets the connection timeout, where *N* is given in seconds.

## **-cookie\_file=FILENAME**

specifies a file to use to read cookies. If none is specified, the default value is `~/lynx_cookies` for most systems, but `~/cookies` for MS-DOS.

## **-cookie\_save\_file=FILENAME**

specifies a file to use to store cookies. If none is specified, the value given by **-cookie\_file** is used.

## **-cookies**

toggles handling of Set-Cookie headers.

## **-core**

toggles forced core dumps on fatal errors. (Unix only)

## **-crawl**

with *-traversal*, output each page to a file.

with *-dump*, format output as with *-traversal*, but to stdout.

## **-curses\_pads**

toggles the use of curses "pad" feature which supports left/right scrolling of the display.

## **-debug\_partial**

separate incremental display stages with MessageSecs delay

## **-display=DISPLAY**

set the display variable for X rexe-ced programs.

## **-display\_charset=MIMEname**

set the charset for the terminal output.

## **-dont\_wrap\_pre**

inhibit wrapping of text in `<pre>` when *-dump*'ing and *-crawl*'ing, mark wrapped lines in interactive session.

## **-dump**

dumps the formatted output of the default document or one specified on the command line to standard out. This can be used in the following way:

*lynx -dump http://www.w3.org/*

**-editor=EDITOR**

enable external editing using the specified *EDITOR*. (vi, ed, emacs, etc.)

**-emacskeys**

enable emacs-like key movement.

**-enable\_scrollback**

toggles behavior compatible with the scrollbar keys in some communications software (may be incompatible with some curses packages).

**-error\_file=FILENAME**

the status code from the HTTP request is placed in this file.

**-exec**

enable local program execution (normally not configured).

**-fileversions**

include all versions of files in local VMS directory listings.

**-find\_leaks**

toggles the memory leak checking off. Normally this is not compiled-into your executable, but when it is, it can be disabled for a session.

**-force\_empty\_hrefless\_a**

force HREF-less “A” elements to be empty (close them as soon as they are seen).

**-force\_html**

forces the first document to be interpreted as HTML.

**-force\_secure**

toggles forcing of the secure flag for SSL cookies.

**-forms\_options**

toggles whether the Options Menu is key-based or form-based.

**-from**

toggles transmissions of From headers to HTTP or HTTPS servers.

**-ftp**

disable ftp access.

**-get\_data**

properly formatted data for a get form are read in from stdin and passed to the form. Input is terminated by a line that starts with “---”.

**-head**

send a HEAD request for the mime headers.

**-help**

print this Lynx command syntax usage message.

**-hiddenlinks=option**

control the display of hidden links. Option values are:

## **merge**

hidden links show up as bracketed numbers and are numbered together with other links in the sequence of their occurrence in the document.

## **listonly**

hidden links are shown only on *List* screens and listings generated by **-dump** or from the *Print* menu, but appear separately at the end of those lists. This is the default behavior.

## **ignore**

hidden links do not appear even in listings.

## **-historical**

toggles use of “>” or “-->” as a terminator for comments.

## **-homepage=URL**

set homepage separate from start page. Will be used if a fetch of the start page fails or if it is a script which does not return a document, and as the *URL* for the “m”ain menu command.

## **-image\_links**

toggles inclusion of links for all images.

## **-ismap**

toggles inclusion of ISMAP links when client-side MAPs are present.

## **-index=URL**

set the default index file to the specified *URL*

## **-justify**

do justification of text.

## **-link=NUMBER**

starting count for lnk#.dat files produced by *-crawl*.

## **-localhost**

disable URLs that point to remote hosts.

## **-locexec**

enable local program execution from local files only (if lynx was compiled with local execution enabled).

## **-lss=FILENAME**

specify filename containing color-style information. The default is lynx.lss.

## **-mime\_header**

include mime headers and force source dump.

## **-minimal**

toggles minimal versus valid comment parsing. When minimal, any “-->” serves as a terminator for a comment element. When valid, pairs of “--” are treated as delimiters for series of comments within the overall comment element. If historical is set, that overrides minimal or valid comment parsing.

## **-nested\_tables**

toggles nested-tables logic (for debugging).

## **-newschunks=NUMBER**

number of articles in chunked news listings.

**-newsmaxchunk=NUMBER**

maximum news articles in listings before chunking.

**-nobold**

disable bold video-attribute.

**-nobrowse**

disable directory browsing.

**-nocc**

disable Cc: prompts for self copies of mailings. Note that this does not disable any CCs which are incorporated within a mailto URL or form ACTION.

**-nocolor**

force color mode off, overriding terminal capabilities and any *-color* flags, *COLORTERM* variable, and saved .lynxrc settings.

**-noexec**

disable local program execution. (DEFAULT)

**-nofilereferer**

disable transmissions of Referer headers for file URLs.

**-nolist**

disable the link list feature in dumps.

**-nolog**

disable mailing of error messages to document owners.

**-nomargins**

disable left/right margins in the default style sheet.

**-nomore**

disable -more- string in statusline messages.

**-nonrestarting\_sigwinch**

make window size change handler non-restarting. This flag is not available on all systems, Lynx needs to be compiled with HAVE\_SIGACTION defined. If available, this flag *may* cause Lynx to react more immediately to window changes when run within an xterm.

**-nopause**

disable forced pauses for statusline messages.

**-noprint**

disable most print functions.

**-noredir**

do not follow URL redirections

**-noreferer**

disable transmissions of Referer headers.

**-noreverse**

disable reverse video-attribute.

**-nosocks**

disable SOCKS proxy usage by a SOCKSified Lynx.

**-nostatus**

disable the retrieval status messages.

**-notitle**

disable title and blank line from top of page.

**-nounderline**

disable underline video-attribute.

**-number\_fields**

force numbering of links as well as form input fields.

**-number\_links**

force numbering of links.

**-partial**

toggles displaying of partial pages while loading.

**-partial\_thres=NUMBER**

number of lines to render before repainting display with partial-display logic.

**-pauth=ID:PW**

set authorization *identifier* and *password* for a protected proxy server at startup. Be sure to protect any script files which use this switch.

**-popup**

toggles handling of single-choice SELECT options via popup windows or as lists of radio buttons. The default configuration can be changed in userdefs.h or [lynx.cfg](#). It also can be set and saved via the “o”ptions menu. The command line switch toggles the default.

**-post\_data**

properly formatted data for a post form are read in from stdin and passed to the form. Input is terminated by a line that starts with “---”.

**-preparsed**

show source preparsed and reformatted when used with -source or in source view (“\”). May be useful for debugging of broken HTML markup to visualize the difference between SortaSGML and TagSoup [recovery modes](#), switched by “^V”.

**-prettysrc**

do syntax highlighting and hyperlink handling in source view.

**-print**

enable print functions. (default)

**-pseudo\_inlines**

toggles pseudo-ALTs for inline images with no ALT string.

**-raw**

toggles default setting of 8-bit character translations or CJK mode for the startup character set.

**-realm**



restricts access to URLs in the starting realm.

## **-reload**

flushes the cache on a proxy server (only the first document affected).

## **-restrictions**

allows a list of services to be disabled selectively and takes the following form:

*lynx -restrictions=[option][,option][,option]...*

The list of recognized options is printed if none are specified.

**?**

if used alone, lists restrictions in effect.

**all**

restricts all options listed below.

**bookmark**

disallow changing the location of the bookmark file.

**bookmark\_exec**

disallow execution links via the bookmark file.

**change\_exec\_perms**

disallow changing the eXecute permission on files (but still allow it for directories) when local file management is enabled.

**chdir**

disallow command which changes Lynx's working directory.

**default**

same as command line option *-anonymous*. Set default restrictions for anonymous users. All specific services listed are always restricted, except for: inside\_telnet, outside\_telnet, inside\_ftp, outside\_ftp, inside\_rlogin, outside\_rlogin, inside\_news, outside\_news, telnet\_port, jump, mail, print, exec, and goto. The settings for these, as well as additional goto restrictions for specific URL schemes that are also applied, are derived from definitions within userdefs.h.

Note that this is the only option value that may have the effect of *removing* some restrictions, if they have been set by other options, namely for those services that *are* allowed by default according to userdefs.h. However, if the separate command line option form (*-anonymous*) is used, Lynx takes care to set the default restrictions before handling additional *-restrictions=* options (even if they precede the *anonymous* option), so that this cannot happen.

**dired\_support**

disallow local file management.

**disk\_save**

disallow saving to disk in the download and print menus.

**dotfiles**

disallow access to, or creation of, hidden (dot) files.

**download**

disallow some downloaders in the download menu. This does *not* imply the disk\_save restriction. It also does not disable the DOWNLOAD command, and does not prevent "Download or Cancel" offers when a MIME type cannot otherwise be handled. Those are only disabled if additionally the disk\_save restriction is in effect *and* no download methods are defined in a [Lynx configuration file](#) that are marked as "always ENABLED" (or, alternatively, if the -validate switch is used).

**editor**

disallow external editing.

## **exec**

disable execution scripts.

## **exec\_frozen**

disallow the user from changing the local execution option.

## **externals**

disallow some "EXTERNAL" configuration lines, if support for passing URLs to external applications (with the EXTERN\_LINK or EXTERN\_PAGE command) is compiled in.

## **file\_url**

disallow using G)oto, served links or bookmarks for file: URLs.

## **goto**

disable the “g” (goto) command.

## **inside\_ftp**

disallow ftps for people coming from inside your domain.

## **inside\_news**

disallow USENET news reading and posting for people coming from inside you domain. This applies to "news", "nntp", "newspost", and "newsreply" URLs, but not to "snews", "snewspost", or "snewsreply" in case they are supported.

## **inside\_rlogin**

disallow rlogins for people coming from inside your domain.

## **inside\_telnet**

disallow telnets for people coming from inside your domain.

## **jump**

disable the “j” (jump) command.

## **lynxcgi**

disallow execution of Lynx CGI URLs.

## **mail**

disallow mailing feature.

## **multibook**

disallow multiple bookmarks.

## **news\_post**

disallow USENET News posting.

## **options\_save**

disallow saving options in .lynxrc.

## **outside\_ftp**

disallow ftps for people coming from outside your domain.

## **outside\_news**

disallow USENET news reading and posting for people coming from outside your domain. This applies to "news", "nntp", "newspost", and "newsreply" URLs, but not to "snews", "snewspost", or "snewsreply" in case they are supported.

### **outside\_rlogin**

disallow rlogins for people coming from outside your domain.

### **outside\_telnet**

disallow telnets for people coming from outside your domain.

### **print**

disallow most print options.

### **shell**

disallow shell escapes.

### **suspend**

disallow *Control-Z* suspends with escape to shell on Unix.

### **telnet\_port**

disallow specifying a port in telnet G)oto's.

### **useragent**

disallow modifications of the User-Agent header.

### **-resubmit\_posts**

toggles forced resubmissions (no-cache) of forms with method POST when the documents they returned are sought with the PREV\_DOC (*left-arrow*) command or from the *History Page*.

### **-rlogin**

disable recognition of rlogin commands.

### **-scrollbar**

toggles showing scrollbar.

### **-scrollbar\_arrow**

toggles showing arrows at ends of the scrollbar.

### **-selective**

require .www\_browsable files to browse directories.

### **-session=FILENAME**

resumes from specified file on startup and saves session to that file on exit.

### **-sessionin=FILENAME**

resumes session from specified file.

### **-sessionout=FILENAME**

saves session to specified file.

### **-short\_url**

show very long URLs in the status line with "..." to represent the portion which cannot be displayed. The beginning and end of the URL are displayed, rather than suppressing the end.

### **-show\_cursor**

If enabled the cursor will not be hidden in the right hand corner but will instead be positioned at the start of the currently selected link. Show cursor is the default for systems without FANCY\_CURSES capabilities. The default configuration can be changed in userdefs.h or [lynx.cfg](#). It also can be set and saved via the “o”ptions menu. The command line switch toggles the default.

### **-show\_rate**

If enabled the transfer rate is shown in bytes/second. If disabled, no transfer rate is shown. Use lynx.cfg or the options menu to select KiB/second and/or ETA.

### **-soft\_dquotes**

toggles emulation of the old Netscape and Mosaic bug which treated “>” as a co-terminator for double-quotes and tags.

### **-source**

works the same as dump but outputs HTML source instead of formatted text. For example

```
lynx -source . >foo.html
```

generates HTML source listing the files in the current directory. Each file is marked by an HREF relative to the parent directory. Add a trailing slash to make the HREF's relative to the current directory:

```
lynx -source ./ >foo.html
```

### **-stack\_dump**

disable SIGINT cleanup handler.

### **-startfile\_ok**

allow non-http startfile and homepage with *-validate*.

### **-stderr**

When dumping a document using **-dump** or **-source**, Lynx normally does not display alert (error) messages that you see on the screen in the status line. Use the **-stderr** option to tell Lynx to write these messages to the standard error.

### **-stdin**

read the startfile from standard input (UNIX only).

### **-syslog=text**

information for syslog call.

### **-syslog-urls**

log requested URLs with syslog.

### **-tagsoup**

initialize DTD with "TagSoup" tables, [more details](#).

### **-telnet**

disable recognition of telnet commands.

### **-term=TERM**

tell Lynx what terminal type to assume it is talking to. (This may be useful for remote execution, when, for example, Lynx connects to a remote TCP/IP port that starts a script that, in turn, starts another Lynx process.)

### **-timeout=N**

For win32, sets the network read-timeout, where *N* is given in seconds.

### **-tlog**

toggles use of a *Lynx Trace Log* for the session. The log is named *Lynx.trace* and is created in the home directory when Lynx trace mode is turned on via the *-trace* command line switch (see below), or via the TRACE\_TOGGLE (*Control-T*) keystroke command. Once a log is started for the session, all trace and other stderr messages are written to the log. The contents of the log can be examined during the session via the TRACE\_LOG (normally, “;”) keystroke command. If use of a Lynx Trace Log is turned off, any trace output will go to the standard error stream.

### **-tna**

turns on ["Textfields Need Activation"](#) mode.

### **-trace**

turns on Lynx trace mode. If a Lynx Trace Log (*Lynx.trace* in the home directory) has been started for the current session, all trace messages are written to that log, and can be examined during the session via the TRACE\_LOG (normally, “;”) command. If no Trace Log file is in use, trace messages go to stderr.

### **-trace\_mask=va lue**

turn on optional traces, which may result in very large trace files. Logically OR the values to combine options:

1

SGML character parsing states

2

color-style

4

TRST (table layout)

8

config (lynx.cfg and .lynxrc contents)

16

binary string copy/append, used in form data construction.

### **-traversal**

traverse all http links derived from startfile. When used with *-crawl*, each link that begins with the same string as startfile is output to a file, intended for indexing. See CRAWL.announce for more information.

### **-trim\_input\_fields**

trim input text/textarea fields in forms.

### **-underscore**

toggles use of `_underline_` format in dumps.

### **-update\_term\_title**

enables updating the title in terminal emulators. Use only if your terminal emulator supports that escape code. Has no effect when used with *-notitle*.

### **-use\_mouse**

turn on mouse support, if available.

### **-useragent=STRING**

set different Lynx User-Agent header. Lynx produces a warning on startup if the *STRING* does not contain "Lynx" or "L\_y\_n\_x", see the [note](#) in the Options Menu section for rationale.

### **-validate**

accept only http URLs (meant for validation).

This flag implies security restrictions generally more severe than *-anonymous*: restriction options as for *-restrictions=all*, with the notable exception that goto remains enabled for http and https URLs; in addition, the PRINT and DOWNLOAD commands are completely disabled, and use of a Trace Log file is forced off.

Any relaxing of restriction that might be implied by an also present (or implied) *-anonymous* flag is overridden, the only way to possibly relax *some* of the restrictions to the level applicable for "anonymous" accounts is with an explicit *-restrictions=default*.

### **-verbose**

toggles [LINK], [IMAGE] and [INLINE] comments with filenames of these images.

#### **-version**

print version information.

#### **-vikeys**

enable vi-like key movement.

#### **-wdebug**

enable Waterloo tcp/ip packet debug (print to watt debugfile). This applies only to DOS versions compiled with WATTCP or WATT-32.

#### **-width=NUMBER**

number of columns for formatting of dumps, default is 80.

#### **-with\_backspaces**

emit backspaces in output if -dumping or -crawling (like `man` does).

No options are required, nor is a startfile argument required. White space can be used in place of equal sign separators (“=”) appearing in the option list above. It can not be used in place of the equal signs in forms like "-option=on" and "-option=off" for simple switches and toggles, for which "-option" alone (without a value) is valid.

[\[ToC\]](#)

## ***Environment variables used by Lynx***

Lynx uses certain environment variables and sets a few of them. Please visit a [separate page](#) for this rather technical information.

[\[ToC\]](#)

## **Main configuration file lynx.cfg**

Lynx has several levels of customization: from the Options Menu (accessible on-line, and possibly stored in your local .lynxrc file), via command-line switches on startup (mainly for batch processing). The most important and numerous default settings are stored in the Lynx configuration file *lynx.cfg*.

If you are on a UNIX system you should have appropriate permissions to make changes there or ask your system administrator to modify lynx.cfg for your needs. This file provides default settings for all accounts on your system. It may be copied to your shell account and included with -cfg command line switch or via an environment variable LYNX\_CFG (if you have shell access). Starting with version 2.8.1 Lynx has an include facility so you can load the system-wide configuration file and easily add one or more settings from your local add-on configuration file. It is really cool to read lynx.cfg with its comments for hundreds of options, most of them commented out because they are built-in defaults. You may visit an index of options: [by category](#) or [by alphabet](#).

To view your current configuration derived from lynx.cfg and any included configuration files, press “g” and type in “lynxcfg:”. If you are using the forms-based *Options Menu*, you may press “o” for the Options Menu and follow the *Check your lynx.cfg*'s link near the bottom.

However, for those who have a restricted account many Lynx features may be disabled by the system administrator, you probably will not see your lynx.cfg.

[\[ToC\]](#)

## **Lynx development history**

Lynx grew out of efforts to build a campus-wide information system at The University of Kansas. The earliest versions of Lynx provided a user-friendly, distributed hypertext interface for users connected to multiuser (Unix and VMS) systems via curses-oriented display devices. A custom hypertext format was developed to support hypertext links to local files and files on remote Gopher servers. Using Gopher servers for distributed file service allowed information providers to publish information from a wide variety of platforms (including Unix, VMS, VM/CMS and Macintosh). In addition, Lynx became the most user-friendly Gopher client, although that was only an ancillary capability.

This distributed approach let providers retain complete control over their information, but it made communication between users and providers somewhat more difficult. Following the lead of Neal Erdwien, of Kansas State University, the Lynx hypertext format was extended to include links for including ownership information with each file. This information made it possible for users running Lynx clients to send comments and suggestions via e-mail to the providers.

This early version of Lynx was also augmented to support hypertext links to programs running on remote systems. It included the ability to open a Telnet connection, as well as the ability to start programs via rexec, inetd, or by direct socket connects. These capabilities were included to allow users to access databases or custom program interfaces.

A subsequent version of Lynx incorporated the World Wide Web libraries to allow access to the full list of WWW servers, along with the option to build hypertext documents in HTML, rather than the native Lynx format. HTML has become far more widely used, and the native format has been phased out. With the addition of the WWW libraries, Lynx became a fully-featured WWW client, limited only by the display capabilities offered in the curses environment.

Lynx was designed by Lou Montulli, Charles Rezac and Michael Grobe of Academic Computing Services at The University of Kansas. Lynx was implemented by Lou Montulli and maintained by Garrett Arch Blythe and Craig Lavender.

Members of the [lynx-dev](#) list have developed and supported Lynx since release of v2.3 in May 1994.

The Lynx2-3FM code set was released as v2.4 in June 1995.

The Lynx2-4FM code set was released as v2.5 in May 1996.

The Lynx2-5FM code set was released as v2.6 in September 1996.

The Lynx2-6FM code set was released as v2.7 in February 1997.

The v2-7FM code set was released as v2.7.1 in April 1997.

The v2-7-1FM code set was released as v2.7.2 in January 1998.

The 2.7.1 development set was released as v2.8 in March 1998.

The 2.8 development set was released as v2.8.1 in October 1998.

The 2.8.1 development set was released as v2.8.2 in June 1999.

The 2.8.2 development set was released as v2.8.3 in April 2000.

The 2.8.3 development set was released as v2.8.4 in July 2001.

The 2.8.4 development set was released as v2.8.5 in February 2004.

The 2.8.5 development set was released as v2.8.6 in October 2006.

The 2.8.6 development set was released as v2.8.7 in July 2009.

The 2.8.7 development set was released as v2.8.8 in February 2014.

The 2.8.8 development set was released as v2.8.9 in July 2018.

The 2.8.9 development set was released as v2.9.0 in January 2024.

Lynx has incorporated code from a variety of sources along the way. The earliest versions of Lynx included code from Earl Fogel of Computing Services at the University of Saskatchewan, who implemented HYPERREZ in the Unix environment. Those versions also incorporated libraries from the Unix Gopher clients developed at the University of Minnesota, and the later versions of Lynx rely on the WWW client library code developed by Tim Berners-Lee (and others) and the WWW community.

Contributors have generally been acknowledged in the CHANGES file. Earlier CHANGES files can be found in the docs/ subdirectory of this distribution.

Information on obtaining the most current version of Lynx is available at [the current distribution page](#).

[\[ToC\]](#)