

url.sty version 3.3

Donald Arseneau
edited as a L^AT_EX document by Robin Fairbairns

2006-04-12, documentation 2010-01-20

The package defines a form of `\verb` command that allows linebreaks at certain characters or combinations of characters, accepts reconfiguration, and can usually be used in the argument to another command. It is intended for email addresses, hypertext links, directories/paths, etc., which normally have no spaces. The font used may be selected using the `\urlstyle` command, and new url-like commands may be defined using `\urldef`.

Usage	Conditions
<code>\url{ }</code>	If the argument contains any “%”, “#” or “^”, or ends with “\”, it can’t be used in the argument to another command. The argument must not contain unbalanced braces.
<code>\url </code>	where “ ” is any character not used in the argument and not “{” or a space. The same restrictions apply as above except that the argument may contain unbalanced braces.
<code>\xyz</code>	for “\xyz” a defined-url; such a command can be used anywhere, no matter what characters it contains.

The “`\url`” command is fragile, and its argument is likely to be very fragile, but a defined-url is robust.

1 Package options

Package Option: `obeyspaces`

Ordinarily, all spaces are ignored in the url-text. The “[`obeyspaces`]” option allows spaces, but may introduce spurious spaces when a url containing “\” characters is given in the argument to another command.

So if you need to obey spaces you can say “`\usepackage[obeyspaces]{url}`”, and if you need both spaces and backslashes, use a defined-url.

Package Option: `hyphens`

Ordinarily, breaks are not allowed after “-” characters because this leads to confusion. (Is the “-” part of the address or just a hyphen?)

The package option “[`hyphens`]” allows breaks after explicit hyphen characters. The `\url` command will **never ever** hyphenate words.

Package Option: `spaces`

Likewise, breaks are not usually allowed after spaces under the “[`obeyspaces`]” option, but if you give the options “[`obeyspaces,spaces`]”, `\url` will allow breaks at those spaces.

Note that it seems logical to allow the sole option “[spaces]” to let input spaces indicate break points, but not to display them in the output. This would be easy to implement, but is left out to avoid(?) confusion.

Package Option: `lowtilde`

Normal treatment of the ~ character is to use the font’s “`\textasciitilde`” character, if it has one (or claims to). Otherwise, the character is faked using a mathematical “`\sim`”. The “[lowtilde]” option causes a faked character to be used always (and a bit lower than usual).

2 Defining a defined-url

Take for example the email address “`myself%node@gateway.net`” which could not be given (using “`\url`” or “`\verb`”) in a caption or parbox due to the percent sign. This address can be predefined with

```
\urldef{\myself}\url{myself%node@gateway.net} or
\urldef{\myself}\url|myself%node@gateway.net|
```

and then you may use “`\myself`” instead of “`\url{myself%node@gateway.net}`” in an argument, and even in a moving argument like a caption because a defined-url is robust.

3 Style

You can switch the style of printing using “`\urlstyle{xx}`”, where “`xx`” can be any defined style. The pre-defined styles are “`tt`”, “`rm`”, “`sf`” and “`same`” which all allow the same linebreaks but use different fonts — the first three select a specific font and the “`same`” style uses the current text font. You can define your own styles with different fonts and/or line-breaking by following the explanations below. The “`\url`” command follows whatever the currently-set style dictates.

4 Alternate commands

It may be desirable to have different things treated differently, each in a pre-defined style; e.g., if you want directory paths to always be in typewriter and email addresses to be roman, then you would define new url-like commands as follows:

```
\DeclareUrlCommand\email{\urlstyle{rm}}
\DeclareUrlCommand\directory{\urlstyle{tt}}
```

In fact, this `\directory` example is exactly the `\path` definition which is pre-defined in the package. If you look above, you will see that `\url` is defined with

```
\DeclareUrlCommand\url{}
```

I.e., using whatever `\urlstyle` and other settings are already in effect.

You can make a defined-url for these other styles, using the usual `\urldef` command as in this example:

```
\urldef{\myself}{\email}{myself%node.domain@gateway.net}
```

which makes `\myself` act like `\email{myself%node.domain@gateway.net}`, if the `\email` command is defined as above. The `\myself` command would then be robust.

5 Defining styles

Before describing how to customize the printing style, it is best to mention something about the unusual implementation of `\url`. Although the material is textual in nature, and the font specification required is a text-font command, the text is actually typeset in *math* mode. This allows the context-sensitive linebreaking, but also accounts for the default behavior of ignoring spaces. Now on to defining styles.

To change the font or the list of characters that allow linebreaks, you could redefine the commands `\UrlFont`, `\UrlBreaks`, `\UrlSpecials`, etc., directly in the document, but it is better to define a new ‘url-style’ (following the example of `\url@ttstyle` and `\url@rmstyle`) which defines all of `\UrlBigbreaks`, `\UrlNoBreaks`, `\UrlBreaks`, `\UrlSpecials`, and `\UrlFont`.

5.1 Changing font

The `\UrlFont` command selects the font. The definition of `\UrlFont` done by the pre-defined styles varies to cope with a variety of L^AT_EX font selection schemes, but it could be as simple as `\def\UrlFont{\tt}`. Depending on the font selected, some characters may need to be defined in the `\UrlSpecials` list because many fonts don’t contain all the standard input characters.

5.2 Changing linebreaks

The list of characters that allow line-breaks is given by `\UrlBreaks` and `\UrlBigBreaks`, which have the format `\do\c` for each character *c*.

The differences are that ‘BigBreaks’ usually have a lower penalty and have different breakpoints when in sequence (as in `http://`): ‘BigBreaks’ are treated as mathrels while ‘Breaks’ are mathbins (see *The TeXbook*, p.170). In particular, a series of ‘BigBreak’ characters will break at the end and only at the end; a series of ‘Break’ characters will break after the first and after every following *pair*; there will be no break after a ‘Break’ character if a ‘BigBreak’ follows. In the case of `http://` it doesn’t matter whether `:` is a ‘Break’ or ‘BigBreak’ — the breaks are the same in either case; but for *DECnet* nodes with `::` it is important to prevent breaks *between* the colons, and that is why colons are ‘BigBreaks’.

It is possible for characters to prevent breaks after the next following character (I use this for parentheses). Specify these in `\UrlNoBreaks`.

You can do arbitrarily complex things with characters by making them active in math mode (mathcode hex-8000) and specifying the definition(s) in

`\UrlSpecials`. This is used in the `rm` and `sf` styles for OT1 font encoding to handle several characters that are not present in those computer-modern style fonts. See the definition of `\Url@do`, which is used by both `\urlrmstyle` and `\url@sfstyle`; it handles missing characters via `\UrlSpecials`. The nominal format for setting each special character `c` is: `\do\c{<definition>}`, but you can include other definitions too.

If all this sounds confusing ... well, it is! But I hope you won't need to redefine breakpoints — the default assignments seem to work well for a wide variety of applications. If you do need to make changes, you can test for breakpoints using regular math mode and the characters `+=(a`.

You can allow some spacing around the breakable characters by assigning

```
\Urlmuskip = 0mu plus 1mu
```

You can change the penalties used for `BigBreaks` and `Breaks` by assigning

```
\mathchardef\UrlBreakPenalty=100
\mathchardef\UrlBigBreakPenalty=100
```

The default penalties are `\binoppenalty` and `\relpenalty`. These have such odd non-L^AT_EX syntax because I don't expect people to need to change them often.

6 Yet more flexibility

You can also customize the verbatim text by defining `\UrlRight` and/or `\UrlLeft`, e.g., for ISO formatting of urls surrounded by `< >`, define

```
\DeclareUrlCommand\url{\def\UrlLeft{<url:\ } \def\UrlRight{>}%
\urlstyle{tt}}
```

The meanings of `\UrlLeft` and `\UrlRight` are *not* reproduced verbatim. This lets you use formatting commands there, but you must be careful not to use TeX's special characters (`\^_~#&&{}` etc.) improperly. You can also define `\UrlLeft` to reprocess the verbatim text, but the format of the definition is special:

```
\def\UrlLeft#1\UrlRight{ ... do things with #1 ... }
```

Yes, that is `#1` followed by `\UrlRight` then the definition. For example, to put a hyperTeX hypertext link in the DVI file:

```
\def\UrlLeft#1\UrlRight{\special{html:<a href="#1">}#1\special{html:</a>}}
```

Using this technique, `url.sty` can provide a convenient interface for performing various operations on verbatim text. You don't even need to print out the argument! For greatest efficiency in such obscure applications, you can define a null `url-style` where all the lists like `\UrlBreaks` are empty.

```
%
% Revision History:
% ver 1.1 6-Feb-1996:
% Fix hyphens that wouldn't break and ligatures that weren't suppressed.
```

```

% ver 1.2 19-Oct-1996:
% Package option for T1 encoding; Hooks: "\UrlLeft" and "\UrlRight".
% ver 1.3 21-Jul-1997:
% Prohibit spaces as delimiter characters; change ascii tilde in OT1.
% ver 1.4 02-Mar-1999:
% LaTeX license; moving-argument-error
% ver 1.5 28-Mar-1999:
% possibility of spacing around break characters; re-settable penalties
% ver 1.6 20-Jun-2002:
% un-double #, fix obeyed-spaces, ignore trailing %, hook for hyperref
% (\Url@HyperHook), no macros in pre-processed url string (in \Url@String),
% limit catcode change of ~.
% ver 3.0 June 2003/Nov 2003:
% \DeclareUrlCommand; make font encoding automatic (only a few inputenc characters
% are supported yet - needs refactoring); reverse penalties.
% ver 3.1 Mar 2004:
% Remove spurious spaces in \url@XXstyle commands.
% ver 3.2 June 2005:
% Fix cmsy-symbols in tt bug (from 3.0); LY1 encoding bug; Enable plain
% with miniltx (again); Define the \urldef for hyperref; Lower "sim" tilde
% a little; fix \lowercase error in \UrlSpecials handling.
% ver 3.3 April 2006:
% Fix some encoding bugs and remove 8-bit characters. lowtilde option
% The End

```