

AcroTeX.Net

## The aeb\_mlink Package

### AeB Pro Family

D. P. Story

## Table of Contents

<b>1 Introduction</b>	<b>3</b>
<b>2 Alternate package name: aeb-mlink</b>	<b>3</b>
<b>3 Requirements</b>	<b>3</b>
<b>4 Package options</b>	<b>3</b>
<b>5 Multi-line link commands</b>	<b>4</b>
5.1 \mlhypertext (\mlsetLink) . . . . .	4
5.2 \mlhyperlink . . . . .	5
5.3 \mlhyperref . . . . .	6
5.4 \mlnameref . . . . .	6
5.5 \mlNameref . . . . .	7
5.6 \mlhref . . . . .	7
5.7 \mlurl . . . . .	8
<b>6 Setting the appearance and action</b>	<b>8</b>
<b>7 Fixing the quad points</b>	<b>10</b>
7.1 Discussion of \mlnameref, \mlNameref, and \mlhypertext . . . . .	13
<b>8 My Retirement Home</b>	<b>15</b>

## 1. Introduction

This package creates multi-line links, a dream of mine for a number of years. The package `hyperref` does create links, but generally these links cannot be broken across lines, unless the application `pdflatex` (or `xelatex`) is used to create a PDF; however, since my workflow is primarily through Acrobat Distiller, this does me little good.

This package uses the `QuadPoints` entry of the link annotation dictionary to create a bounding region; consequently, this package requires Acrobat Distiller or the `ps2pdf` converter to create a PDF. The `QuadPoints` entry in the link annotation is a PDF 1.6 feature, so these multi-line links will work in Adobe Reader 7.0 or later. If viewed in a version of Adobe Reader previous to 7.0, the viewer will use the underlying bounding box, which is the smallest rectangle containing the link text.

Version 2.3.6 (dated 2020/07/12 or later) now allows an `aeb_mlink` document to be compiled by `pdflatex`, `lualatex`, and `xelatex` without errors; however, all commands defined in the package do nothing other than to display their `<text>` with no links created. This makes it nice for the lovers of these drivers to compile the document with their favorite driver and view the PDF in their favorite PDF previewer, such as SumatraPDF. Later, the document author can switch over to a `latex -> dvips -> (distiler|ps2pdf)` workflow to acquire the multi-line link version.

## 2. Alternate package name: `aeb_mlink`

This package is known on CTAN as `aeb_mlink` ([ctan.org/pkg/aeb\\_mlink](http://ctan.org/pkg/aeb_mlink)), even though there is no such package by that name, until now. Included within this distribution is the ‘dummy’ package `aeb_mlink` which loads `aeb_mlink` and passes all options to it.

## 3. Requirements

The following  $\LaTeX$  packages are the major ones required by `aeb_mlink`.

- `eforms` provides form and link support, à la  $\AcroTeX$  ([ctan.org/pkg/acrotex](http://ctan.org/pkg/acrotex)).
- `hyperref` provides link support, including the commands `\hyperref`, `\hyperlink`, `\href`, `\nameref`, `\Nameref`, and `\href`.
- `soul` a very nice package by Melchior FRANZ, used by `aeb_mlink` to break the text of a link into its syllables ([ctan.org/pkg/soul](http://ctan.org/pkg/soul)).

## 4. Package options

There are only two package options—the name of the dvi-to-ps converter application to be used—, these are `dvips` and `dvipson`.<sup>1</sup> The default is `dvips`.

```

\documentclass{article}
\usepackage{xcolor}
\usepackage[colorlinks]{hyperref}
\usepackage[<options>]{aeb_mlink}
\documentclass{article}
\usepackage[designi]{web}
\usepackage[<options>]{aeb_mlink}

```

<sup>1</sup>`dvipson.exe` is the dvi-to-ps converter of the Y&Y  $\TeX$  System, no longer available, but is used by this author.

Above are two typical preambles for the use of `aeb_mlink`:

- The one on the left inputs `xcolor` (recommended) and `hyperref` explicitly. The `colorlinks` option is recommended for `hyperref`. These two are followed by `aeb_mlink`.
- The preamble representation on the right uses the `web` package, which internally inputs both `xcolor` (if available) and `hyperref` (with `colorlinks` option). Again, `aeb_mlink` comes last.

**Options of `aeb_mlinks`.** These are,

`dvips` Use this option when the `dvips` application is used to create the Postscript file.

`dvipsone` Use this option when the `dvipsone` application is used to create the Postscript file. (Only a few people worldwide use this option, it could be as small as two.)

`dbllevel=<0|1|2>` To obtain information on the multi-line links being created. Each value gives more information about what is going on. Used for development.

*<eforms-options>* Any options other than the ones listed above are passed to the `eforms` package.

Since `eforms` is a required package, the `acrotex` bundle (AeB) is installed; consequently, if you've set up the `web.cfg` configuration file (that accompanies AeB) with either `\ExecuteOptions{dvips}` or `\ExecuteOptions{dvipsone}`, the specification of the driver (`dvips` or `dvipsone`) is not normally needed.

## 5. Multi-line link commands

We define seven multi-line hypertext link commands: `\mlhypertext`, `\mlhyperlink`, `\mlhyperref`, `\mlnameref`, `\mlNameref`, `\href`, and `\mlurl`. The first is a basic command that is used in the definitions of the other six.

The demo file is `mlink_tst.tex`, it illustrates all the commands - with the exception of `\mlurl` - of this package. The demo file for `\mlurl` is a separate file and is mentioned in [Section 5.7](#).

### 5.1. `\mlhypertext` (`\mlsetLink`)

A general purpose multi-line link command.

```
\mlhypertext[<opts>]{<text>} \mlsetLink[<opts>]{<text>}
```

**Parameter Description:** The first parameter is optional, and is used to set the appearance and the actions of the link. The second parameter is the text around which the link is constructed.

**Command Description:** The eforms package defines `\mlsetLink` to `\mlhypertext` (this is to conform to the naming conventions of eforms). Either command sets a hypertext link having a default appearance, and no action. The default appearance is the same as that of `\hyperlink`, the command from `hyperref`. If `colorlinks` option was taken in the `hyperref` package, the text is colored `\linkcolor` (usually red); otherwise, the text is the default color at the time, and a boundary rectangle is created by `AcroForms` with color `\linkbordercolor` (usually red).<sup>2</sup>

For example, the following link is constructed using `\mlhypertext`: [Acrobat Distiller, version 7.0 or greater](#) was used to create this PDF document.

```
\mlhypertext[\A{\JS{app.alert("AcroTeX rocks!")}}]{%
  {Acrobat Distiller, version~7.0 or greater}}
```

In the above example, the link, shown in green is broken across lines. Though is not shown in this example, these links can break a line at a hyphenation point as well. The optional argument is used to create an alert box that appears when the link is clicked.

## 5.2. `\mlhyperlink`

A multi-line version of the `\hyperlink` command.

```
\mlhyperlink[<opts>]{<named-dest>}{<text>}
```

**Parameter Description:** The first parameter is optional, and is used to set the appearance, but *not the actions* of the link. The second parameter is a named destination, perhaps created by the `\hypertarget` command of `hyperref`. The third parameter is the text around which the link is constructed.

**Command Description:** This command is the multi-line version of the `hyperref` command `\hyperlink`. The default appearance is the same as `\hyperlink`, as described above in `\mlhypertext`. The action of this link is to jump to a named destination; the action cannot be changed by defining a different action through the first parameter.

This link uses the `\mlhyperlink` command: [Welcome, ... to my humble retirement residence, such as it is!](#)

```
\mlhyperlink{retire}{Welcome, $\dots$ to my humble
retirement residence, such as it is!}
```

In this example, the line is broken at a hyphenation, cool! We jump to the named destination, `retire`. On the last page, the following text appears

```
My humble retirement \hypertarget{retire}{home}.
```

creating the named destination of `retire`.

<sup>2</sup>The web package changes the default coloring of `hyperref`.

### 5.3. `\mhyperref`

A multi-line version of the `\hyperref` command.

```
\mhyperref[⟨opts⟩]{⟨label⟩}{⟨text⟩}
```

**Parameter Description:** The first parameter is optional, and is used to set the appearance, but *not the actions* of the link. The second parameter is a label name created by the `\label` command of  $\text{\LaTeX}$ . The third parameter is the text around which the link is constructed.

**Command Description:** This is the multi-line version of `\hyperref`, the command defined by the `hyperref` package. The default appearance is the same as `\hyperref`. The action of this link is to jump to a destination created by the `\label` command of the cross-referencing system of  $\text{\LaTeX}$ ; the action cannot be changed by defining a different action through the first parameter.

This link uses the `\mhyperref` command: [Welcome! Welcome to my humble retirement residence, such as it is!](#)

```
\mhyperref[\ui{linktxtcolor=red}]{retirement}
{Welcome! Welcome to my humble retirement
residence, such as it is!}
```

The user-friendly interface is used to change the appearance of the color of the link text. The second argument is `retirement`, which is the label affixed to the last section of this document,

```
\section{My \textbf{Retirement} Home}\label{retirement}
```

See [‘My Retirement Home’ on page 15](#).

### 5.4. `\mnameref`

A multi-line version of the `\nameref` command.

```
\mnameref[⟨opts⟩]{⟨label⟩}
```

**Parameter Description:** The first parameter is optional, and is used to set the appearance, but *not the actions* of the link. The second parameter is a name of a  $\text{\LaTeX}$  label.

**Command Description:** This is the multi-line version of the `hyperref` (`nameref`) command, `\nameref`. The default appearance is the same as `\hyperref`. The action of this link is to jump to a destination created by the `\label` command of the cross-referencing system of  $\text{\LaTeX}$ ; the action cannot be changed by defining a different action through the first parameter.

This link uses the `\mnameref` command: Click the link to witness [My Retirement Home](#). The code for this link is seen below:

```
\mnameref{retirement}
```

### 5.5. `\m1Nameref`

A multi-line version of the `hyperref` command, `\Nameref`.

```
\m1Nameref[⟨opts⟩]{⟨label⟩}
```

**Parameter Description:** The first parameter is optional, and is used to set the appearance, but *not the actions* of the link. The second parameter is a name of a  $\LaTeX$  label.

**Command Description:** This is the multi-line version of the `hyperref` (`nameref`) command, `\Nameref`. The default appearance is the same as `\hyperref`. The action of this link is to jump to a destination created by the `\label` command of the cross-referencing system of  $\LaTeX$ ; the action cannot be changed by defining a different action through the first parameter.

This link uses the `\m1Nameref` command: Click the link or go to the page, ‘**My Retirement Home**’ on page 15. The code for the link is seen below:

```
\m1Nameref{retirement}
```

### 5.6. `\m1href`

A multi-line version of the `hyperref` command, `\href`.

```
\m1href[⟨opts⟩]{⟨url⟩}{⟨text⟩}
```

**Parameter Description:** The first parameter is optional, and is used to set the appearance, but *not the actions* of the link. The second parameter is an URL, while the third parameter is the link text.

**Command Description:** This is the multi-line version of the `hyperref` command, `\href`. The default appearance is the same as `\hyperref`. There is one of three actions, depending on the third argument, just as with `\href`. The action cannot be changed by defining a different action through the first parameter.

The next link uses the `\m1href` command with URL: **See the educational offerings of the Acro $\TeX$  Web Site at the University of Akron.**

```
\m1href{%
http://www.math.uakron.edu/~dpstory/acrotex.html#educational}
{See the educational offerings of the {Acro\negthinspace{\TeX}} Web
Site at the University of Akron}.
```

Use the `\m1href` command with `mailto` URL: My email address at Acro $\TeX$ .Net, **dpstory at acrotex dot net.**

```
\m1href{mailto:dpstory@acrotex.net}{dpstory at acrotex dot net}
```

If the second argument is a PDF file, the PDF viewer will open it from your local file system, or if you are in a browser, will open it from the web, provided it can find the relative path

```
\mhref{shameless_ad.pdf}{Here is a
shameless advertisement for Acro\negthinspace\TeX}.
```

If the key word ‘run:’ is used, the link is set up to launch an application. The PDF viewer will attempt to open the file with the application associated with the extension of the file provided.

```
\mhref{run:\jobname.tex}{This is the source file of this document}.
```

### 5.7. \murl

A multi-line version of the `\url` command of the `url` package. The `hyperref` modifies this command to make it into an unbreakable link (at least for those using Distiller).

```
\murl[{opts}]{{url}}
```

**Parameter Description:** The first parameter is optional, and is used to set the appearance, but *not the actions* of the link. The second parameter is a URL, the text of the link is the same as the URL.

**Command Description:** This is the multi-line version of the `url/hyperref` command, `\hurl`. The default appearance is the same as set by `\hyperref`.

The next link is an example of `\murl`: <http://www.math.uakron.edu/~dpstory/acrotex.html#educational>. The code for this link is found below:

```
\murl{http://www.math.uakron.edu/~dpstory/acrotex.html#educational}
```

The `url` package typesets the URL in math mode, which makes it very difficult to wrap it to the next line with a hypertext link around it. In math mode, spaces are ignored; consequently, if the URL contains spaces, use `\%20` (with backslash), the URL encoding for the space character. See the Internet reference [http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp) for a discussion of the **HTML URL ENCODING REFERENCE**.

The demo file that illustrate all the commands to follow is `m1ink_tst_url.tex`. Additional details and examples are found in this example file.

## 6. Setting the appearance and action

The appearance and the action (at least for `\m1hypertex`) can be set through the first optional parameter. The optional parameter takes key-value pairs of two types:

**Key-value Pairs.** The optional first parameter of each of the form and link commands take two styles of key-values:

1. eForms KVP: This is the key-value system originally developed, where each KVP has the form `\<key>\{<val>\}`. Through these keys, the appearance and action of a link—if otherwise not restricted—may be set. For example, [AcroTeX can be found at www.acrotex.net](http://www.acrotex.net), as close as your nearest web browser.



```
\m\hypertext[\S{U}\W{1}\Color{0 0.6 0}\linktxtcolor{webgreen}
\A{\JS{app.alert("AcroTeX rocks!")}}]{AcroTeX...browser.}
```

An alternative link format is to provide a visible rectangle; Use a link with a non-zero boundary size along with `\S{S}`, which is the default.

```
\m\hypertext[\W1\linktxtcolor{}\Color{red}
\A{\JS{app.alert("Wrap the link!")}}]{Use a ... boundary size.}
```

The eForms KVP system is explained in the eforms manual, *Support for AcroForms and Links, and for Document JavaScript and Open Page Events*,<sup>3</sup> where a complete listing of all supported KVPs is found in the appendices.

2. xkeyval KVP: When the `useui` option is taken of the eforms package, key-value pairs are defined of the form `<key> [= <value>]`. The key-value pairs are actually a value of a special eForms key, `\ui`. The value of `\ui` consists of a comma-delimited list of xkeyval key-value pairs. For example, [AcroTeX can be found at www.acrotex.net](http://www.acrotex.net), as close as your nearest web browser.

```
\m\setLink[\ui{%
border=visible,
linestyle=underlined,
bordercolor={0 0.6 0},
linktxtcolor=webgreen,
js={app.alert("AcroTeX rocks")}] {AcroTeX...browser.}
```

The xkeyval KVP system is described in the document *Support for AcroForms and Links, and for Document JavaScript and Open Page Events*.<sup>4</sup>

**Keys special to the `aeb_mlink` package.** The following links are defined within eforms, but are only obeyed from within the option list of one of the `\m\<cmdname>` commands.

- `\m\fix{<y|n>}` (xkeyval KVP: `m\fix=<yes|no>`)  
If `\m\fix{<y>}` is specified, contiguous bounding rectangles are combined; otherwise they are not. See Section 7 for more information.
- `\m\strut{<some-strut>}` (xkeyval KVP: `m\strut=<some-strut>`)  
The value of this key sets the height of the bounding rectangles; the default is `\strut`. See Section 7 for details and examples.
- `\m\crackat{<num>}` (xkeyval KVP: `m\crackat=<num>`)  
This key is use to crack (or break) an hypertext or url into two links. The application is to enable the link to break across a page boundary. Refer to the paragraph **Multi-line links that break across pages**, beginning on page 12 for more information on `\m\crackat`.

<sup>3</sup><http://www.math.uakron.edu/~dpstory/acrotex/eformman.pdf>


<sup>4</sup>ibid.

- `\m1hyph{<y|n>}` (xkeyval KVP: `m1hyph=<yes|no>`)  
When the `\m1crackat` option is used, an hyphen is inserted provided `\m1hyphy`, otherwise, no hyphen is inserted. Refer to the paragraph [Fixing a cross-page link](#), page 12, for an example.
- `\m1crackinsat{<text>}` (xkeyval KVP: `m1crackinsat=<text>`)  
The value `<text>` of this key is inserted after the break point, when `\m1crackat` is used. Normally, nothing is inserted after the brake point.

## 7. Fixing the quad points

In versions of `aeb_mlink` prior to 2018/03/19, the each syllable (possible break point) of multi-line text was enclosed in a bounding rectangle (described by the coordinates of its four corners). For example, it is well known that the acrotex website is free to all who dare to enter. Now you see the individual bounding rectangles; they are all linked together, if you click on one of them, they all react.

Beginning with the version of `aeb_mlink` dated 2018/03/19, Postscript code was introduced to fix up these bounding boxes so that all syllables *on the same line* are enclosed in a single bounding rectangle. If we turn on this feature and declare the acrotex website is free to all who dare to enter. Very cool!

-  The demonstration document for these feature is found on the [AcroTeX Blog](#) web site. Refer to the article titled [Fixing multi-line link boxes](#).

**Turning the fix on and off.** The feature of fixing up these little rectangles into one rectangle per line is controlled by the commands `\m1fixOn` and `\m1fixOff`; the first turns on the fix up, the second turns it off. The default is `\m1fixOn`. These two commands turn on or off the fix up for the document (or within a group), to turn the fix on or off for an individual link, use the key-value pairs `\m1fix{y}` to turn on the fix and `\m1fix{n}` to turn it off. These key-values are placed as an optional argument of any of the `\m1<cmdname>` commands. For example,

```
\m1hypertext[\m1fix{n}]A{\JS{app.alert("And so it is");}}
\linktxtcolor{\W1\H{0}\S{S}\Color{red}}{the acrotex website is
free to all}
```

The user-friendly counterpart is `m1fix=<yes|no>`.

**Changing the vertical size of the rectangles.** The default height and depth of the boxes are determined by `\strut`, in the current font size. A *new key is introduced*, `\m1strut{<some-strut>}`. You can simple increase the vertical dimensions of the boxes by `\m1strut{\large\strut}` or `\m1strut{\huge\strut}`. The acrotex website is free to all link should have vertically expanded rectangles.

```
\m1hypertext[\m1strut{\large\strut}
A{\JS{app.alert("And so it is");}}
\linktxtcolor{\W1\H{0}\S{S}\Color{red}}{The acrotex website
is free to all}
```

When the link is underlined ( $\S{U}$ ) the depth is set to 1bp. These changes to the vertical size of the rectangles do not affect any text positioning of the content, only the bounding boxes.

`\mlstrut=<some-strut>` is the counterpart to `\mlstrut{<some-strut>}` in the user-friendly key-value system.

The `<some-strut>` argument can be anything that takes up vertical space, any horizontal space is ignored. For example, “The integral  $\int_0^\pi \sin(x) dx$  has a value of a crummy little 2.”

```
\mlhypertext[\mlstrut{\displaystyle\int_0^\pi}\linktxtcolor{\W1\H{0}}
\S{S}\Color{red}]{‘The integral {\displaystyle\int_0^\pi \sinx\,dx$}
... $2$.’}
```

**Important:** When the feature of fixing the rectangles is active (with `\mlfixOn` or locally with `\mlfix{y}`), you need to *compile the document at least three times* before distilling the document; otherwise distiller fails. There are commands written to the AUX file that needs to be up to date. This is only needed after you modify any of the multi-line hypertext links. There are  $\text{\LaTeX}$  warning messages written to the  $\text{\TeX}$  log to inform you if the links are not fully formed; there are also messages in the distiller log. Setting the package option `dbllevel=1` can increase the amount of feedback you get through the distiller log.

**Marking the links.** You can run into trouble with one or more of the links. There may be tex log or distiller log messages referring to the link number (e.g., `mLink30`), but where is this link? By expanding `\mlMarksOn`, you turn on the marking of the links. I’m turning on link marking for the rest of the document. Look for ‘`L<num>`’ beginning in section 7.1. Turn marking back off with `\mlMarksOff`. Above the starting point of each link will be a tiny ‘`L<num>`’. The format for this little ‘`L<num>`’ is determined by the internal command `\ml@MrkLnk`; it may be redefined as desired.

**Viewing the syllable numbers.** You can display the syllable numbers of multi-line hypertext by expanding `\turnSyllbCntOn`; `\turnSyllbCntOff` turns off this feature again. For example, we expand `\turnSyllbCntOn` in a group:

```
l221 We have set \turnSyllbCntOn in a group so the effects are not felt after the
18 group.
```

```
\begin{quote}\turnSyllbCntOn
\mlhypertext{We have set {\cs{turnSyllbCntOn}} in a
group so the effects are not felt after the group}.
\end{quote}
```

The sentence has 10 syllables, as counted by `soul` and `aeb_mlink`. It should be noted that `\turnSyllbCntOn` also executes `\mlMarksOn`. This feature may be useful for determining where to break a link across pages, see the next paragraph.

Notice we’ve enclosed `\cs{turnSyllbCntOn}` in braces, as a result, `soul` treats the contents as a single syllable. If we want to break this long command across a line (or page), we would have to register the command `\cs` with `\soulregister`:

Register \cs with \soulregister to have the command name \turnSyll  
bCntOn break across lines (or pages).

```
\begin{quote}\turnSyllbCntOn\soulregister{\cs}{1} % global declaration
\mlhypertext{Register \cs{cs} with \cs{soulregister} to have
the command name \cs{turnSyllbCntOn} break across lines (or pages).}
\end{quote}
```

**Multi-line links that break across pages.** This phenomenon is not supported by the PDF Reference. These kinds of links can be created by  $\LaTeX$  when the link is near the bottom of the page and  $\LaTeX$  moves a portion of it to the next page. `aeb_mlinks` detects this when the document is distilled and places a message in the distiller (or `ps2pdf`) log. If you have a link that crosses page boundaries, you need to reword the paragraph to avoid this. The message reads,

```
Beginning of page: 3
!!-----
Warning:
The text of mLink4 has crossed a page boundary from page 2 to 3.
Cross page links are not supported by the PDF Specification.
This link is not constructed, please fix it.
Break point is after syllable number 4.
Use the \mlcrackat{4} option with this link.
!!-----
```


and can't be missed. The link is not constructed. You can see here, there is a reference to `mLink4`. You need to turn on link marking (`\mlMarksOn`) or just go to page 2 and look at the bottom of the page.

**Fixing a cross-page link.** There are a couple of action choices you can take to fix a link that has crossed a page boundary: (1) rewrite the paragraph so the link does not cross the page boundary; (2) manually break the link; (3) do as the log file suggests, you can use the `\mlcrackat{4}` option. In this latter case, for example,

```
\mlhypertext[\mlcrackat{4}\mlhyph{y}<other-opts>]
{Doctor Story created the acrotex website.}

<latex-content>.....Doctor Story cre-
                                     (page break)
ated the acrotex website.
```

Two links are created, the first consisting of the text 'Doctor Story cre-' and the second on 'ated the acrotex website.'; the second link (should) be inserted on the next page. The link option `\mlhyp{y}` inserts a hyphen at the break point; for URLs, a hyphen may not be appropriate. **Question:** What is the missing `<other-opts>` in the above example?

 **A working document:** The demonstration document for breaking links across pages is found on the [AcroTeX Blog](#) web site. Please refer to the article titled [Crossing page boundaries with multi-line links](#). Many other articles on AeB and PDF can be found at the [AcroTeX Blog](#) web site, as well.

**Reverting to the old style.** The restriction of having to compile multiple times before distillation may be a burden when it comes to developing a document. You can declare `\OldStyleBoxesOn` in the preamble to revert to the old style that does not use the information written to the auxiliary file. After document development and the text of all links are stable, you can then say `\OldStyleBoxesOff` (or comment out `\OldStyleBoxesOn`) in the preamble. Compiling then yields the better formed and combined bounding rectangles.

**Minimally sized /Rect.** For multi-line links, the link annotation contains two entries containing dimensions of rectangles, the `/Rect` and `/QuadPoints` entries. Our focus in this paragraph is on `/Rect`; there are two possible values for `/Rect`: (1) the dimensions of the entire page, which has been the default value since the inception of this package, or (2) the dimensions of the smallest rectangle containing all the hypertext, the new default.

For example, the following link is constructed using `\m1hypertext: Acrobat Distiller, version 7.0 or greater` was used to create this PDF document. The blue rectangle is a representation of the ‘unseen’ `/Rect` dimensions. It is the smallest rectangle that contains the hypertext. You can revert to the old default (the whole page) by expanding `\useSmallRectOff`, the default is `\useSmallRectOn`.

**The importance of small rectangles.** In the current age we live in, web browsers have their own ‘native’ PDF viewers. These viewers generally do not obey JavaScript actions but they do recognize basic actions such as jumping to the destination in the document; however, they ignore the `/QuadPoints` entry in a link, and respect only the `/Rect` entry. In a ‘native’ viewer, following a multi-line link may be problem when the `/Rect` entry is set the whole page. Now, the `/Rect` is the minimal rectangle, and following a link should not be an issue as long as *one rectangle is not contained within another*. Multi-line links are fully supported by Adobe Acrobat Reader.

**Tricky bits.** The use of `\m1nameref` and `\m1Nameref` can be tricky when the referenced label contains special commands or font changes. Two tricks are to enclose the bad bits in braces (`{}`) and/or `\protect` commands. Section or paragraph commands that end up as PDF bookmark need to use the `hyperref` command `\texorpdfstring`. To illustrate, an dummy subsection is introduced, and the discussion on this topic is continued there.

### 7.1. Discussion of `\m1nameref`, `\m1Nameref`, and `\m1hypertext`

To illustrate, we reference the section title “<sup>128</sup>Discussion of `\m1nameref`, `\m1Nameref`, and `\m1hypertext`.” The verbatim listing of this subsection is

```
\subsection{Discussion of problems with \texorpdfstring
  {{\protect\cs{m1nameref}}, {\protect\cs{m1Nameref}},
   and {\protect\cs{m1hypertext}}}}
  {\textbackslash{m1nameref}, \textbackslash{m1Nameref},
   and \textbackslash{m1hypertext}}}\label{ss:probs}
```

Observe the use of `\texorpdfstring`, the first argument is typeset, the second one is placed in the bookmarks.

- On the first argument of `\texorpdfstring`, we focus our comments on the first `{\protect\cs{m\nameref}}` of the three commands referenced. Note that it is enclosed in braces that signals the `soul` package, which does the parsing, to treat it as an unbreakable unit. The control sequence `\cs` is protected by `\protect`. Finally `\cs` is a command that produces a backslash in front of its argument in monospace font.

There are two ‘common’ definitions of `\cs`-style commands:

- `\def\cs#1{\texttt{\char'\#1}}`
- `\def\cs#1{\texttt{\@backslashchar#1}}` (used in this manual)

The latter definition must be the one used within a section title, not the first one.

The `aeb_mlinks` defines `\mlcs` using the second version; that is

```
\def\mlcs#1{\texttt{\@backslashchar#1}}
```

is defined. In this way both definitions for `\cs` are available. The `\mlcs` version must be used in text that can be referenced by `\mlnameref` or `\mlNameref`.

In the above title declaration, if any of the components (the braces, the `\protect`, and the special definition of `\cs`)  $\LaTeX$  compilation errors occur.

Similar comments are made for `\mlhypertext`, `\mlhyperref`, and `\mlhyperlink`. These three take a text phrase as its ‘hypertext’ string. For example, “<sup>129</sup>The exciting world of `\mlhyperref` is average wonderful.”

```
\mlhyperref[\linkxtcolor{\Color{red}\S{S}\W1}{ss:\mlhyperref}{The
exciting world of {\protect\mlcs{\mlhyperref}} is average wonderful.}
```

Notice the braces, the `\protect`, and the use of `\mlcs`; they work together for a successful link construction.

## 8. My Retirement Home

My humble retirement home.



**Click image to return from jump**

Now, I simply must get back to my retirement. ~~DS~~