

AcroTeX.Net

The artthreads Package

Creating article threads for multi-column documents

D. P. Story

Stop! Before you read this documentation, take a moment to change your preferences: On your menu system of Adobe Acrobat/Reader, select Edit> Preferences > General, and *clear the checkbox* for the item named Make Hand tool select text & images, as listed under Basic Tools on that dialog box. (Ctrl+K or Command+K is the shortcut to the Preferences dialog box on Windows or Mac OS, respectively.) This makes reading an article thread much easier. Look for the hand icon with a down arrow on it.


Article threads are an Adobe Reader/Acrobat feature. To my knowledge, they have not been implemented in any other PDF viewer. So, a PDF containing article threads must be viewed in Adobe Reader or Acrobat 

Table of Contents

1	Introduction	3
2	Navigating an article thread	3
3	Creating article threads	4
4	Other navigation commands	8
5	My retirement	10

1. Introduction

Through the commands of the `artthreads` package, you can create *article threads*, a concept/feature of Adobe Reader/Acrobat that has been around since the product's beginning. The use of article threads typically only makes sense in a document in which the text is in a multi-column format, or, perhaps, a single narrow column.

When a document uses a multiple column format, it is often difficult, or at the very least inconvenient, to read it. When document page is fit into the viewing window, the text may be too small to read. This requires the reader, that's you, to adjust the magnification (zoom) level of the

page to fit a column of text comfortably in the application window. Once the column is fitted, additional navigation is needed to move the window view down one column, then up to the top of the next column. Think of trying to read a digital newspaper with many columns. Such navigation is tedious and distracts from reading the information being presented.

Set the article threads after the composition of the document is completed. The method of setting the threads is *very visual* not automatic as in the `threadcol` package of Scott Patkin.

2. Navigating an article thread

Perhaps you've noticed the 'Intro' button on exhibit in this previous section or the 'Navigate' button in this section as well. Clicking these buttons reveals the **Articles panel** in the left-hand navigation pane. Listed there are all the articles created for this documentation. Right-clicking on any of the listed article names brings up a context menu; select Read Article to begin reading or choose Properties to see the thread info, which (may) include the title, author, subject, and keywords.

Begin an article thread. There are two of ways to begin reading a thread: (1) first open the Article pane (View > Show/Hide > Navigation Panes > Articles), left-click an article title and select Read Article, or simple double-click an article title; or (2) place your mouse pointer over text, if it changes to a hand with an arrow on it (), the text is part of an article thread, simply click on it to begin.

Navigating a thread. Once you have initiated Read Article, you can navigate

through the article thread using any of the following methods: (1) click in an area when the mouse is a hand icon with an arrow on it () to move forward (shift-click to move back ()); (2) press the Enter key to move forward (or shift+Enter to move back); or (3) the page down (or page up) keys will also move you around within the thread. At the end of the thread, you'll see this hand icon .

There may be problems in getting the hand icon with the arrow on it to appear, if this is the case, open the Preferences menu, Edit > Preferences > General, and *clear the checkbox* for the item named Make Hand tool select text & images, as listed under Basic Tools on that dialog box. (Ctrl+K or Command+K is the shortcut to the Preferences dialog box on Windows and Mac OS, respectively.)

Exploring threads. Now that you have explored the operation of article threads and see how they work, the next thing to do is to see how they are created with the `artthreads` package.

It should be noted that in recent versions of DC, the thread icons are changed from `\artthread`, `\artthread*`, `\artthread*`, and `\artthread*` to `\artthread`, `\artthread*`, `\artthread*`, and `\artthread*`, respectively.

3. Creating article threads

We begin by including the `artthreads` package in your document.

```
\usepackage[options]{artthreads}
```

The `artthreads` package brings in the `fitr` package (<http://ctan.org/pkg/fitr>), many commands from that package are used. All options of `artthreads` are passed to `fitr`.

Package options. The package has ten options: six driver options and four viewing options.

- Driver Options:** The major driver options are `dvips` (the default), `pdftex`, `luatex`, and `xetex`; there are two minor driver options, these are `dvipdfm` and `dvipdfmx`. If you specify `dvips`, it is assumed the PDF creator is Adobe Distiller.¹

The `artthreads` package uses the driver of the `web` package, if it is loaded; otherwise `artthreads` auto-detects for `pdflatex`, `lualatex` and `xelatex`. If no driver is passed and no driver is automatically detected, the `dvips` driver is used.
- Viewing Options:** When you specify `preview`, the bounding boxes for the article threads are shown in the dvi-previewer (or the PDF document); you can turn off this preview by specifying `!preview` (or removing `preview` entirely from the option list). The other option type is `viewMagWin`. When this option is specified, the viewing window, a rectangular region, becomes visible in the dvi-previewer (or in the PDF document); specifying `!viewMagWin` turns off this type of preview.

You'll notice the two rectangular regions around the left column. The inner one is made visible by the `preview` option (or locally by `\previewOn`), you can see that it covers 'precisely' the text of the item. The outer rectangle is exposed by the `viewMagWin` option (or locally by `\viewMagWinOn`). When you view the thread - and you do not define an 'extra' border around the viewing area - the inner rectangle is used. The text in this inner rectangle may not be entirely adequate for reading as it is very 'tight' around the text. It is for this reason, we expand the viewing area by a specified amount. Compare the reading comfort of the left column versus the right column. (Try viewing the column thread with a narrow application window.)

You may have noticed that the viewer returns to the same page view the page was in prior to reading the thread.

The article info. The specifications for an article thread require it to have a title, and to optionally have key-values for author, subject, and keywords.

```
\setThreadInfo{title=text,author=text,
subject=text,keywords={\list}}
```

¹You can use `dvips -> ps2pdf` if there are no document JavaScripts.

If no value of `title` is provided, `artthreads` uses a generated value of `thread-⟨num⟩`. The `\setThreadInfo` command must appear prior to the opening the thread (see `\bArticle` below) to which it applies, if it does not, a \LaTeX error occurs. The drivers `dvips` and `xetex` (`dvipdfm`, `dvipdfmx`) support all four keys; however, `pdftex` and `lualatex` only supports the `title` key.

The article thread commands. There is one command (`\bArticle`) for beginning an article thread, and another (`\cArticle`) to continue that thread.

```
\bArticle{KV-pairs}
\cArticle{KV-pairs}
```

Before illustrating `\bArticle` and `\cArticle`, the *KV-pairs* are described first.

- `width=⟨length⟩`: The value of `width` sets the width of the thread to `⟨length⟩`; `⟨length⟩` is usually, but not always, `\linewidth`. Dimension arithmetic is supported with the value of `width`; for example, `width=\linewidth+3pt`.
- `height=⟨length⟩`: The value of `height` sets the height of the thread to `⟨length⟩`. As with the `width` key, dimension arithmetic is supported by `height`; for example, `height=3in+2\baselineskip`.
- `lift=⟨length⟩`: This key-value lifts (raises) the article thread window up (or down) by an amount of `⟨length⟩`; for example, `lift=15pt` (or `lift=-15pt`). The default is a lift of `0pt`. Dimension arithmetic is supported.
- `shift=⟨length⟩`: The amount of horizontal shift of the article thread; positive to the right, negative to the left. For example, `shift=-1in` shifts the button/viewing window 1 inch to the left. The default is `0pt`. Dimension arithmetic is supported.

One last command before the examples.

Setting the expanded (add to) border. It is often convenient to set the basic thread dimensions (`width=\linewidth,height=3in`), but is harder to expand and position the thread to a wider or higher thread region. Use the `\setAddToBorder` command to expand, or add to, the dimensions of the thread rectangular dimensions.

```
\setAddToBorder{addtow=⟨length⟩,addtoh=⟨length⟩}
```

The `addtow` key adds `⟨length⟩` to the left and right sides of the rectangular boundary of the thread; `addtoh` adds `⟨length⟩` to the top and bottom of the rectangular boundary of the thread. The default dimensions for these two keys are `0pt`.

```
\setAddToBorder{addtow=10bp,
  addtoh=10bp}
```

In this column we've set a 10bp border around the thread. When you click on the reading area, the viewer should zoom to the width of the wider rectangle.

```
\setAddToBorder{addtow=4bp,
  addtoh=4bp}
```

In this column, we've set a 4bp border. Again, when you click on the reading area, the viewer should zoom to the width of the wider rectangle.

Tip: The values of `addtow` and `addtoh` should be in big points (bp) for greater accuracy as these are converted to Postscript or PDF dimensions, depending on the driver.

Methodology with examples. Now we've come to final part of this documentation, a description of the methodology, accompanied by examples. As pointed out early in this documentation, the method for applying the article threads is very visual. Compose your document content first, and then insert the document threads.

We illustrate the techniques in a single column with a generous width. First we declare our add to border dimension and article info

```
\setAddToBorder{addtow=4bp,addtoh=4bp}
\setThreadInfo{title=Methodology,author=D. P. Story}
```

There are two places to begin a thread: (1) at the very beginning of a paragraph; and (2) from within a paragraph. Method (1) is preferred. We begin the next paragraph with

```
\noindent\bArticle{lift=\baselineskip,
  width=\linewidth,height=1in+3\baselineskip}%
Now we begin our deep thoughts...
```

❶ Now we begin our deep thoughts. During this development phase, the options `preview` and `viewMagWin` should be in effect. In a DVI viewer, we can see the bounding rectangles. We can see it is too low, too long and so on. For users of `pdflatex`, `xelatex`, etc., view your documents in the DVI previewer first (or view them as a PDF). I've used `lift=\baselineskip` to raise up the thread rectangle by that mount to enclose the first line. The value of `width` is usually `\linewidth`, but the value of `height` may have to be adjusted, in this example, I've "tweaked" the `1in` by `3\baselineskip`.

❷ Naturally, after you're satisfied, you then remove `preview` and `viewMagWin` or change them to `!preview` and `!viewMagWin`. Another possible location is from within a paragraph, here, we continue the previous thread using the `\cArticle` command in conjunction with the `\vadjust` TeX primitive. Between the end of the word 'primitive' and the period (.) ending the sentence, I've placed,

```
primitive\vadjust{\noindent\cArticle{lift=5\baselineskip,
  width=\linewidth,height=2in+\baselineskip}}. Between the
```

We chose `lift=5\baselineskip` because the original positioning of the thread rectangle was too low by about five lines. The value of `height` was chosen to cover the paragraph once the complete paragraph was composed.

We now declare `\previewOff\viewMagWinOff`.

- ③ Of course, in this example, it was entirely unnecessary to insert the `\cArticle` mid-paragraph, it would have been better placed at the beginning of the paragraph. Placing `\cArticle` mid-paragraph is useful for multi-column formats. When the paragraph flows from the first column the second column, use the `\vadjust` technique to cover the portion of the text that flows to the right column.

By the way, notice the use of `\noindent`, this is oftentimes needed to position the article thread in the left margin of the text block.

Continuing a thread from one page to the next. To continue reading across a page boundary, continue the thread on the previous page using `\cArticle`. For example, the most recent thread begins with paragraph ① (`\bArticle`) and continues with paragraph ② and ③ (`\cArticle`). Paragraphs ① and ② are on the previous page, while paragraph ③ is on this page. As you read this thread, paragraph ① is presented first, followed by paragraph ②, and finally, paragraph ③ on the subsequent page.

The demo files. The demo file `article_tech.tex` reproduces the discussion on methods, it is found in the `examples` folder.

In this documentation, I've covered the content with many article threads; normally, for multi-column content, one thread per column is sufficient. The thread for this article continues on to the next page, as this *Lipsum* text illustrates.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pul-

vinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta

tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per incepto hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristisque mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristisque sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consetetur odio sem sed wisi.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consetetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis

commodo.

Etiam ac leo a risus tristisque nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

The demo file. The demo file for this package is `article_tst.tex`, available in the `examples` folder.

4. Other navigation commands

There are several commands for creating links or push button actions that open the Articles pane and/or begin reading a specified article.

Manipulating the Articles pane. The following two commands represent actions that toggle the Articles pane (`\shArticlesPaneActn`) or just open/show the Articles pane (`\sArticlesPaneActn`).

<code>\shArticlesPaneActn</code>	(Toggle Articles pane action)
<code>\sArticlesPaneActn</code>	(Show Articles pane action)
<code>\shArticlesPaneReadActn</code>	(Toggle Articles pane and read article action)
<code>\sArticlesPaneReadActn</code>	(Show Articles pane and read article action)

For example,

```
Toggle Articles pane
\setLink[\A{\shArticlesPaneActn}]{Toggle \textsf{Articles} pane}
Show Articles pane
\setLink[\A{\sArticlesPaneActn}]{Show \textsf{Articles} pane}
```

These two actions can be used by push buttons as well.

Reading a thread. To create an action to read a thread with a title of `<thread-title>`, use `\Thread{<thread-title>}`. For example,

```
Read about the Package options.
\setLink[\A{\Thread{Package options}}]{Package options}.
```

Again notice that the viewer returns you to the jumping off point after you've finished reading the thread. The `\Thread{<thread-title>}` action can be part of a push button action as well.

Some push buttons. `artthreads` has some push buttons with common actions, these are,

```
\toggleArticlePane[<KV-pairs>]{<wd>}{<ht>}
\toggleArticlePaneRead[<KV-pairs>]{<wd>}{<ht>}
\showArticlePane[<KV-pairs>]{<wd>}{<ht>}
\showArticlePaneRead[<KV-pairs>]{<wd>}{<ht>}
```

The names of the commands describe the action they perform. However, the commands `\toggleArticlePaneRead` and `\showArticlePaneRead` require special comment. Each begins an article thread, but which article thread?

```
\showArticlePaneRead{}{11bp}
```

Without any further qualifiers, this button begins reading the thread as determined by the value of the `title` key of the last `\setThreadInfo` command declared. So, the above push button should begin the thread titled "Final Thread". However, to program the button to begin reading the thread titled `<thread-title>`, use the following syntax:

```
\showArticlePaneRead[\cmd{readArticle{<thread-title>}}]{11bp}
```

We illustrate by using `\showArticlePaneRead` to read the thread with title “Package options,” and the same action in a link.

```
\showArticlePaneRead[\cmd{\readArticle{Package options}}]{}{11bp}
Show Articles pane and read the Package options article
\setLink[\cmd{\readArticle{Package options}}
  \A{\sArticlesPaneReadActn}]{Show \textsf{Articles} pane
  and read the Package options article}
```

Tooltips for push buttons. The tooltips for the commands `\toggleArticlePane` and `\showArticlePane` are Toggle Articles Pane and Show Articles Pane, respectively. These can be localized using the `\TU` key if eforms; for example,

```
\showArticlePane[\TU{Umschalten der Artikel Schaltfl}{a}{che}]{}{11bp}
```

which may be German for Toggle Articles Pane.

For the other two, `\toggleArticlePaneRead` and `\showArticlePaneRead`, two convenience commands are defined.

```
\tooltipTogglePaneRead{<text>}
\tooltipShowPaneRead{<text>}}
```

The values of `<text>` allows for the inclusion of the command `\threadTitle`, which expands to the current title of the thread article. The default definitions are,

```
\tooltipTogglePaneRead{Toggle Articles pane and
  read the article \threadTitle}
\tooltipShowPaneRead{Show Articles pane and
  read the article \threadTitle}
```

If you roll over the button at the top of the this page, its tooltip will read Show Articles pane and read the article Package options. The two commands may be declare with an appropriate language localization.

Beyond Basic ASCII. Begin by reading [Jürgen's Thread](#). The examples of this reference and, for the most part in the sample files, use Basic ASCII as the thread titles; however, the full PDFDocEncoding character set is also supported. The sample file `article_tech.tex` provides a more complete discussion of this topic, as well as a more complicated example. Check out the [Articles](#) pane to see the title of this thread listed.

5. My retirement

Now, I simply must get back to it. ~~DS~~