APT Quick Reference

10th April 2004

Title

- You can include Title, Author, and Date information in the title block
- The title block must be indented (preferably centred), and must start with at least three dashes. Each element is surrounded by dashes. For example:

```
------
Title
------
Author
------
Date
------
```

Section

- You can have up to five levels of sectioning
- Create a section title by entering the title, left justified. Section levels two to five must have an asterisk in front on them. For example:

```
Section 1 Title

* Section 2 Title

** Section 3 Title

*** Section 4Title

**** Section 5 Title
```

Paragraphs

- You must indent paragraphs. This is how the aptconvert utility recognizes paragraphs
- The amount of indentation does not matter. A good rule of thumb is four spaces of indentation. If your text editor has an automatic indentation function, then use this function to do the indenting for you
- Separate paragraphs with empty lines. For example:

```
This is merely the first paragraph of my document.  Nothing Earth shattering
here, but it does demonstrate what to do.

This is the second paragraph.  Like the first, there's nothing important
here.  It's for demo purposes only.
```

Lists

- You must indent a list, and start each list item with the proper list delimiter
- APT supports three types of lists:
  - **bullet list**, which starts with `*`
  - **numbered list**, which starts with `[[n]]` -- n can be number, upper or lower case letter, or upper or lower case "I" (for upper or lower Roman numerals)
  - **definition list**, which starts with `[term]` -- `[term]` is the defined term, which is followed by the definition
- You can nest both paragraphs and other lists in a list. You must indent nested items more than the parent list.
- You can end a list with a section title, by adding extra space after a list, or by putting [] after the last list item.

```
* List item 1

* List item 2

* List item 3
```

```
* List item 1

* List item 2

  * List item 2a

  * List item 2b

* List item 3
```

Verbatim Text

- Verbatim text is set in a monospaced font and retains all formatting like preformatted text in HTML or the verbatim environment in LaTeX
- A block of verbatim text begins and ends with three unindented dashes
- You can replace first and last dashes with plus signs to surround the verbatim text with a box, as shown below:

```
int main()
        {
        printf("Enter degrees (e.g. 38): ");
        scanf("%d", &degrees);
        printf("Enter minutes (e.g. 28): ");
        scanf("%d", &minutes);
        printf("Enter seconds (e.g. 53): ");
        scanf("%d", &seconds);
```

```
    int main()
            {
            printf("Enter degrees (e.g. 38): ");
            scanf("%d", &degrees);
            printf("Enter minutes (e.g. 28): ");
            scanf("%d", &minutes);
            printf("Enter seconds (e.g. 53): ");
            scanf("%d", &seconds);
```

Images

- Image block are not indented
- The name of the image is surrounded by square brackets

- You can add a caption to the image by typing the caption text immediately after the image block
- You do not need to add extension to image name. APT tries all supported image types

```
[/home/scott/images/mclaren] Mika Hakkinen's 1999 McLaren
```

Tables

- Tables are not indented
- Tables start with an asterisk, followed by at least two dashes
- Each new cell starts with one of the following:
  - `*`, signifying the column is centred
  - `+`, signifying the column is left aligned
  - `\:`, signifying the column is right aligned
- Start new rows start with an asterisk followed by at least two dashes
- Cells are separated by |
- You can include a table caption at the end of the table

```
*-----*------*
cell | cell
*-----*------*
cell | cell
*-----*------*

*----------*-------------+----------------:
| Centered | Left-aligned | Right-aligned |
| cell 1,1 | cell 1,2     | cell 1,3      |
*----------*-------------+----------------:
| cell 2,1 | cell 2,2     | cell 2,3      |
*----------*-------------+----------------:
Table caption
```

Horizontal rule

- Horizontal rules are like the <hr> tag in HTML
- You created them with at least three equal signs that are not indented

```
==========
```

Page break

- Insert a form feed/page break character using text editor function. How you do this depends on the text editor that you are using. Emacs users, for example, use the **C-q C-l** command.

Italicize, bold, and monospaced text

- *Italicized text* is surrounded by single angle brackets
- `Bold text` is surrounded by double angle brackets
- `Monospaced text` is surrounded by triple angle brackets

```
<italics>

<<bold>>

<<<monospaced>>>
```

Anchors and links

- You specify an anchor between curly braces
- You specify a link between double curly braces
- You can specify text for a link using extra braces
- You can link internally (to an anchor), or externally using various HTML links (http:, mailto:, ftp:, etc.)

```
{Anchor}

Link to {{Anchor}}

{{http://www.scottnesbitt.net}}

{{{http://www.scottnesbitt.net}My home page}}
```

Line breaks

- Insert a backslash at point where want to break the line
- You cannot add line breaks to a title or table

Special characters

- You must escape special characters using a backslash. For example, \~
- Other characters, like copyright symbol (?), are denoted by `\nn` or `\unn` (if Unicode), where `\nn` is a hexadecimal or Unicode code. For example:

```
\251

\u00a9
```

Running APT

- APT can output documents in the following formats:
  - LaTeX
  - DocBook
  - HTML
  - Rich Text Format (RTF)
  - Postscript
  - PDF
- To generate output, you must run the `aptconvert` command, along with various processing instructions, at the command line prompt

- Sample commands for each format are given below. Check the APT user guide for details
- Because processing instructions can be quite long, you should consider putting them into a script

LaTeX

```
aptconvert -nonum -pi latex documentclass [11pt]article -pi latex classic no foo.tex foo.apt
```

- `-nonum` indicates that sections in the document will not be numbered
- `-pi latex documentclass [11pt]article` is a processing instruction that tells aptconvert to format the document as a LaTeXarticle, with an 11 point font
- `-pi latex classic no` is a processing instruction that tells aptconvert to improve the look of the document using Postscript fonts. If you want to use the standard LaTeXfonts, set this option to `classic yes`

DocBook

```
aptconvert -nonum -pi docbook publicId "-//OASIS//DTD/DocBook XML V4.0//EN" foo.xml foo.apt
```

- `-nonum` indicates that sections in the document will not be numbered
- `-pi docbook publicId "-//OASIS//DTD/DocBook XML V4.0//EN"` is a processing instruction that indicates which DocBook identifier to use

HTML

```
aptconvert -nonum -pi html paging 0 -pi html css docs.css foo.html foo.apt
```

- `-nonum` indicates that sections in the document will not be numbered.
- `-pi html paging 0` is a processing instruction that tells aptconvert to keep the output in one HTML file. You can break the page into separate, linked documents by specifying heading levels 1 to 5

- `-pi html css docs.css` is a processing instruction that tells aptconvert to format the HTML with the Cascading Style Sheet you specify (in this case, docs.css)

Rich Text Format (RTF)

```
aptconvert -nonum -pi rtf topmargin 2.5 -pi rtf bottommargin 2.5 -pi rtf
leftmargin 2.5 -pi rtf rightmargin 2.5 -pi rtf fontsize 11 -pi rtf spacing 12 foo.rtf foo.apt
```

- `-nonum` indicates that sections in the document will not be numbered
- `-pi html paging 0` is a processing instruction that tells aptconvert to keep the output in one HTML file. You can break the page into separate, linked documents by specifying heading levels 1 to 5
- `-pi rtf` is a processing instruction that tells aptconvert to generate a file in Rich Text Format (RTF), which can be read by most word processors
- `-pi rtf topmargin 2.5 -pi rtf bottommargin 2.5 -pi rtf leftmargin 2.5 -pi rtf rightmargin 2.5` are a set of processing instructions that set the margins of each page to 2.5 cm (approximately one inch)
- `-pi rtf fontsize 11` is a processing instruction that sets the body text of the document to 11 points
- `-pi rtf spacing 12` is a processing instruction that sets the spacing between lines to 12 points

Postscript

An APT document is converted to LaTeXand then typeset before it is converted to Postscript using the dvips utility

```
aptconvert -pi ps pass1 latex doc -pi ps pass2 dvips -o %0 doc foo.apt
```

- `-pi ps pass1 latex doc` is a processing instruction that tells aptconvert to to convert the document to a LaTeXsource file and then generate a DVI file
- `-pi ps pass2 dvips -o %0 doc` is a processing instruction that tells aptconvert to use dvips to convert the LaTeXdocument to Postscript

PDF

An APT document is converted to LaTeXand then typset before it is converted to Postscript using the dvips utility. From there, it is converted to PDF using the ps2pdf utility that is part of the Ghostscript distribution.

```
aptconvert -pi pdf pass1 latex doc -pi pdf pass2 dvips -o %0 doc -pi pdf
pass3 ps2pdf foo.ps %0 foo.apt
```

- `-pi pdf pass1 latex doc` is a processing instruction that tells aptconvert to convert the document to a LaTeXsource file and then generate a DVI file
- `-pi pdf pass2 dvips -o %0 doc` is a processing instruction that tells aptconvert to use dvips to convert the LaTeXdocument to Postscript using dvips
- `-pi pdf pass3 ps2pdf foo.ps %0` is a processing instruction that tells aptconvert to use ps2pdf to convert the Postscript document to PDF