Forecast Production Assistant Version 8

Graphics Product Generator Reference Manual

Forecast
Production
Assistant

August 2016

Contents

1 Introduction to GPGen 1

1.1 Product Definition Files 1

1.2 Control Characters and Quotes 3

1.3 Format of Latitudes and Longitudes 4

1.4 Format of Values in a Range 4

1.5 Format of Valid Time Identifiers 5

1.6 Input and Output File Locations and Setup File Parameters 6

1.7 Running Graphics Product Generator Applications Directly 7

1.8 Examples of Product Definition Files 7

1.9 Configuration File Parameters 8

1.10 Magic Attributes 8

1.11 Graphics Geography Files 11

1.12 Graphics Symbol Files 12

1.13 Graphics Pattern Files for Drawing Lines 13

1.14 Graphics Fonts and Spacing for Text 14

1.15 Graphics Colours for Text, Lines, and Areas 14

1.16 Graphics Arrow Heads and Tails 16

1.17 Graphics Wind Barbs 16

1.18 Errors and Warnings 16

2 PSMet/SVGMet 17

2.1 PSMet/SVGMet Input and Output Files 17

2.2 Special PSMet/SVGMet Directives 18

2.3 PSMet/SVGMet Display Directives 20

2.4 Differences Between PSMet and SVGMet 24

2.5 Complete Description of All PSMet/SVGMet Directives 24

3 TexMet 191

3.1 TexMet Input and Output Files 191

3.2 Special TexMet Directives 192

3.3 TexMet Display Directives 193

3.4 Complete Description of All TexMet Directives 194

GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a worldwide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See Copyleft.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

Chapter 1

Introduction to the Graphics Product Generator

The Graphics Product Generator offers users the ability to create tailored graphic or text products using data from the FPA. There are presently three applications available, PSMet, SVGMet and TexMet.

The PSMet and SVGMet applications allow customized sampling and display of FPA data as a PostScript file (for PSMet) or as a Scalable Vector Graphic (SVG), an open XML specification created by the World Wide Web Consortium (W3C) for describing two-dimensional vector graphics, (for SVGMet). The PSMet and SVGMet commands allow the user to re-display geographic features, contour fields, area fields, line fields, and point data from the FPA in various colours or styles. The commands also allow sampling of FPA fields or equations, and the results can be displayed as graphic symbols or text on a geographic background or in tabular form. Headers and graphic symbols can also be displayed.

The TexMet application allows customized sampling of FPA fields or equations for display in an ASCII tabular form. The TexMet commands are similar to those in PSMet and SVGMet, but with fewer display options. This section introduces some of the common features of the Graphics Product Generator applications.

1.1 Product Definition Files

The Graphics Product Generator applications control the display of FPA data through the use of FPA product definition files, usually referred to as "pdf" files, because they used to have the suffix ".pdf".

Note

When FPA "pdf" files were first introduced the ".pdf" extension did not have the association it does now. The extension ".pdf" has since become synonymous with Portable Document Format files, and as a result it has become difficult to share FPA "pdf" files, especially on Windows platforms. With the introduction of FPA Version 7, the extension of FPA product definition files has been changed to ".fpdf".

Each "pdf" file contains a series of directives, or commands, that allow for the sampling and formatted display of information from the FPA database.

The syntax for most directives in the product definition files is:

```
@directive { keyword = value(s); keyword = value(s); ..... }
```

or:

```
@directive
  {
  keyword = value(s);
  keyword = value(s);
  .....
  }
```

or any combination of the two syntaxes. The keywords for each directive set one or more values for parameters that are allowed within each command. Multiple values can be separated by spaces or tabs. There is no required order to the keywords, so long as the instructions within each directive are sufficient to fully describe the data to be sampled and the format of how the data will be displayed.

The only limit on the directives is that each "keyword = value(s)" instruction has a maximum length of 255 characters.

The value(s) for parameters may have one of the following formats:

string	an ASCII literal value
integer	an integer number
real	a real number
latlon	(see Format of Latitudes and Longitudes, (Section 1.3))
range	(see Format of Values in a Range, (Section 1.4))
vtime	(see Format of Valid Time Identifiers, (Section 1.5))

A few special directives have no keywords. These include the PSMet, SVGMet and TexMet directives **@file_close** **@include** **@loop_end** **@process** **@verbose** and **@version**, the PSMet and SVGMet directives **@filter** **@reset_contour_presentation** and **@reset_geo_presentation**, and **@gpgen_group_end**. The syntax for these directives is:

```
@directive { value }
```

or

```
@directive { }
```

The **@include** directive allows the user to split up large product definition files into a number of smaller files, which will then be processed sequentially when the Graphics Product Generator application is run.

There are a handful of special directives that allow output of language specific values. These directives begin with **@gpgen_** such as **@gpgen_group_begin** and **@gpgen_insert**. The syntax for these directives is one of:

```
@gpgen_directive { keyword = value(s); keyword = value(s); ... }
```

```
@gpgen_directive {
  keyword = value(s);
  keyword = value(s);
  ...
}
```

```
@gpgen_directive { value(s) }
```

or

```
@gpgen_directive { }
```

The keywords and values are assumed to be specific to the output format (SVG for example), and so are not checked against valid GPGEN keywords and values. Instead they are inserted as is into the output product.

A description of all directives and their keywords for the PSMet, SVGMet and TexMet applications can be found in PSMet/SVGMet, (Chapter 2) and TexMet, (Chapter 3).

1.2 Control Characters and Quotes

Certain characters found in the product definition files have special meaning. These include:

!	Remainder of line is comment only
{	Start of keyword block for a directive
}	End of keyword block for a directive
=	Divider between a keyword and its value
\	Line continuation for values of a keyword
;	End of values for a keyword

Keywords that allow text strings therefore require the use of the escape character "\" before any of the control characters to allow literal usage of the characters, instead of translating them as part of the directive. For example, "Time \= 00 Local\! (not GMT)" in a keyword text string would be displayed as Time = 00 Local! (not GMT), rather than treating the "=" as a "keyword/value" divider, or the "!" as the beginning of a comment.

Note

A text string with embedded blanks can be enclosed by single or double quotes. Quotation marks at the beginning and end of a text string are automatically stripped off. If the quotation marks are required, then they can be embedded within an outer set of quotation marks. For example, a keyword text string of "’GMT’" would be displayed as ’GMT’.

1.3 Format of Latitudes and Longitudes

Several keywords in the Graphics Product Generator applications accept the value of latitudes or longitudes in special formats that correspond to the formats allowed in the Graphics Metafile Standard. These are identified by a keyword type of latlon. These special formats include:

[+/-] ***DDD*** [N/S/E/W]	for whole degrees
[+/-] ***DDD.DDD*** [N/S/E/W]	for decimal degrees
[+/-] ***DDD:MM*** [:SS] [N/S/E/W]	for degrees, minutes, (seconds)

The default directions, with no signs present, are in degrees North for latitudes and in degrees East for longitudes.

The following are examples of latitude formats:

45	would be read as 45 degrees North
+45:30S	would be read as 45.5 degrees South

The following are examples of longitude formats:

65.75	would be read as 65.75 degrees East
-65:45W	would be read as 65.75 degrees East

1.4 Format of Values in a Range

Several keywords in the Graphics Product Generator applications accept minimum and maximum values in a special format that allows for inclusive or exclusive ranges. These are identified by a keyword type of range. This format is given by:

[***min*** , ***max***]	for a range ***min*** $\leq$ value $\leq$ ***max***
(***min*** , ***max***]	for a range ***min*** < value $\leq$ ***max***
[***min*** , ***max***)	for a range ***min*** $\leq$ value < ***max***
(***min*** , ***max***)	for a range ***min*** < value < ***max***

A colon (:) may be used in place of the comma (,).

An asterisk (*) may be used in place of a min or max value; it is interpreted as an unlimited end of a range.

The following are examples of range formats:

Example 1.1 defines a range 75 < value $\leq$ 105

```
( 75 , 105 ]
```

Example 1.2 defines a range value < 105

```
[ * : 105 )
```

1.5 Format of Valid Time Identifiers

Several keywords in the Graphics Product Generator applications use a special format to specify valid times. These are identified by a keyword type of vtime. These times are often calculated from **T0**, the zero hour time for the application, which usually corresponds to the zero hour in the FPA depiction sequence. The format of valid times is given by one of:

xx	or	***rr/hh***	or	***rr/hhL***
		yyyy/jjj/hh	or	***yyyy/jjj/hhL***
		yyyy/mm/dd/hh	or	***yyyy/mm/dd/hhL***

where:

xx	is the number of hours after **T0** (or before **T0** if negative)
rr	is the day (0 for the **T0** day, 1 for the next day, ...)
hh	is the GMT hour of day (0 to 23)
hhL	is the local hour of day (0 to 23)
yyyy	is the year (4 digits required)
jjj	is the Julian day of year (1 to 366)
mm	is the month (1 to 12)
dd	is the day of month (1 to 31)

Minutes are also optionally allowed, by replacing the ***xx*** format with ***xx***:***tt***, or by replacing the ***hh*** or ***hhL*** formats with ***hh***:***tt*** or ***hh***:***ttL***, where ***tt*** is the number of minutes after the hour (0 to 59).

The following are examples of valid time formats:

Example 1.3 defines a valid time 18 hours from T0

```
18
```

Example 1.4 defines a valid time at 10:30 local the day after T0

```
1/10:30L
```

Example 1.5 defines a valid time at 18 GMT, May 10, 2008

```
2008/131/18
```

1.6 Input and Output File Locations and Setup File Parameters

The Graphics Product Generator applications assume default locations for all input and output files. However, these locations can be overridden by setting parameters in the FPA setup file used by each application. The setup file also contains a list of PSMet, SVGMet and TexMet products that can be generated from the FPA graphical editor.

The **`directories`** block of the FPA setup file may contain the following entries identifying the directory locations for the input and output files for the Graphics Product Generator applications (the default location is given in brackets):

psmet	***`pdf_directory`***	(`$HOME/setup/pdf/psmet`)
psout	***`output_directory`***	(`PSOut` in the Data directory)
svgmet	***`pdf_directory`***	(`$HOME/setup/pdf/svgmet`)
svgout	***`output_directory`***	(`SVGOut` in the Data directory)
texmet	***`pdf_directory`***	(`$HOME/setup/pdf/texmet`)
texout	***`output_directory`***	(`TexOut` in the Data directory)

The ***`pdf_directory`*** names are the base directories for the product definition files for generating the PSMet, SVGMet or TexMet products, and are usually located relative to the local FPA setup directory.

The ***`output_directory`*** names are the base directories for the output graphics file from the PSMet or SVGMet applications, or the output ASCII file for the TexMet application. These are usually located relative to the FPA Data directory, the database directory for the setup file being used.

The **`[product.graphic]`** section of the **`interface`** block of the FPA setup file contains the entries which identify the PSMet, SVGMet and TexMet products that can be produced from the FPA. These products have the format:

```
"product_label" psmet          sub_directory     product_name
"product_label" svgmet         sub_directory     product_name
"product_label" texmet         sub_directory     product_name
```

where the ***`product_label`*** strings are the labels that appear in the **Products** → **Graphics** panel in the FPA depiction editor, and the ***`sub_directory`*** and ***`product_name`*** names specify the location of the product definition files for generating the PSMet, SVGMet or TexMet products, as in:

```
pdf_directory/sub_directory/product_name.fpdf
```

The sub-directory and product-names also specify the location of the output files for the PSMet, SVGMet or TexMet products, as in:

`output_directory/sub_directory/product_name.ps` (for PSMet)
or `output_directory/sub_directory/product_name.svg` (for SVGMet)
or `output_directory/sub_directory/product_name.txt` (for TexMet)

When the Graphics Product Generator applications are run from within the FPA depiction editor, the setup file information as well as the ***`sub_directory`***, ***`product_name`***, and the current **T0** depiction time are all passed to the applications to be used as required.

1.7 Running Graphics Product Generator Applications Directly

Graphics Product Generator applications are usually run from the FPA depiction editor using the **Products** → **Graphics** panel. This launches the requested application, and maintains communication with the FPA as the application is running.

However, all Graphics Product Generator applications can also be run directly, from outside the FPA depiction editor. The format of the run string is:

Example 1.6 PSMet

```
psmet setup_file sub_directory product_name T0
```

Example 1.7 SVGMet

```
svgmet  setup_file sub_directory product_name T0
```

Example 1.8 TexMet

```
texmet setup_file sub_directory product_name T0
```

where ***setup_file*** is the name of the local FPA setup file, ***sub_directory*** and ***product_name*** specify the location of the product definition files (as described in the section above), and **T0** is the zero hour time for the application. Note that the **T0** time can be given in one of the formats from the Format of Valid Time Identifiers, (Section 1.5), shown previously, such as ***yyyy/jjj/hh*** or ***yyyy/jjj/hhL*** or ***yyyy/mm/dd/hh*** or ***yyyy/mm/dd/hhL***. Note that minutes are also optionally allowed, by using ***hh:tt*** or ***hh:ttL*** in place of ***hh*** or ***hhL***. (Note that the **T0** time usually corresponds to the current zero hour in the FPA depictions.)

1.8 Examples of Product Definition Files

Examples of product definition files for the PSMet, SVGMet and TexMet applications can be found in the directories:

```
$FPA/setup/pdf/psmet/examples
$FPA/setup/pdf/svgmet/examples
$FPA/setup/pdf/texmet/examples
```

Users may copy any portion of these files to their local directories to develop their own graphics products.

Warning
The files in these directories may be overwritten by future revisions of FPA, so that any editing should be carried out on local copies of these files.

1.9 Configuration File Parameters

Some of the keywords in the Graphics Product Generator applications refer to parameters in the FPA configuration files, described in Appendix A of the Administrator's Guide. These include:

element, element_list
an ***element_name*** (or list of names) from the Elements block

attribute, category_attribute
an ***attribute_name*** from the attributes section of the Elements block

wind_crossref
a ***crossref_name*** from the Winds section of the CrossRefs block

level, level_list
a ***level_name*** (or list of names) from the Levels block

source
a ***source_name*** from the Sources block (Note that ***source_name***:***subsource_name*** is also an allowed format)

units
a ***unit_name*** from the Units block

1.10 Magic Attributes

The **attribute** or **category_attribute** keywords (and related keywords containing the **attribute** string) usually refer to attribute names defined in the FPA configuration files. However, there are other special parameters that can also be accessed, such as information from a look up table or from a label or sampled value. The attribute names used to access these parameters are referred to as magic attributes, and the common ones are described below. Each magic attribute referred to in an attribute keyword may also be accompanied by a format keyword, and the appropriate conversion format for each magic attribute is also given. The common magic attributes are:

GPGEN_All
used for matching ALL attribute values.

GPGEN_Missing
used for matching when an attribute is NOT available.

GPGEN_DoNotMatch *xxx*
used for matching when it is easier to use a list of attribute values to EXCLUDE rather than a list of attribute values to match, and ***xxx*** is one or more attribute values.

GPGEN_default_attribute
default sampling attribute, for each field type:

EVAL_spval
for continuous or vector fields (%f in **printf** format)

FPA_auto_label
for discrete fields (%s in **printf** format)

FPA_category
for line or scattered or link chain fields (%s in **printf** format)

GPGEN_ident
identifier from look up table or list (%s in **printf** format)

GPGEN_label
label from look up table (%s in **printf** format)

GPGEN_lat
latitude for location (%f in **printf** format)

GPGEN_lat_ddmm
latitude as ***DDD***:***MM***[N/S] where ***DDD*** is degrees and ***MM*** is minutes and N or S is appended (%s in **printf** format)

GPGEN_lon
longitude for location (%f in **printf** format)

GPGEN_lon_ddmm
longitude as ***DDD***:***MM***[E/W] where ***DDD*** is degrees and ***MM*** is minutes and E or W is appended (%s in **printf** format)

GPGEN_prog_time
difference in valid time from **T0** as hh:tt where hh is hours and tt is minutes (%s in **printf** format)

GPGEN_prog_time_hours
difference in valid time in hours from **T0** (%d in **printf** format)

GPGEN_prog_time_minutes
difference in valid time in minutes from **T0** (%d in **printf** format)

The following magic attributes use the **strftime** format to display time. See **@write_time** directive for more details.

GPGEN_gmt_time
valid time as GMT (use **strftime** format)

GPGEN_local_time
valid time as local standard (use **strftime** format)

GPGEN_T0_time
T0 time (use **strftime** format)

GPGEN_creation_time
product creation time (use **strftime** format)

The following magic attributes can only be used in **@label** and **@sample_field** directives.

GPGEN_bearing
Bearing of a line at the sampled point (%f in **printf** format)

GPGEN_line_direction_from
The direction from which a line feature is pointing at the label location (%f in **printf** format)

GPGEN_line_direction_to
the direction to which the line feature is pointing at the label location (%f in **printf** format)

GPGEN_line_length
Length of the sampled line in km (%f in **printf** format)

GPGEN_link_chain_direction
Direction of movement calculated from current link node to the next node in the link chain (%f in **printf** format).

GPGEN_link_chain_speed
Speed calculated from current link node to the next node in the link chain (%f in **printf** format).

GPGEN_link_chain_vector
Direction and speed calculated from current link node to the next node in the link chain (%f in **printf** format).

GPGEN_link_chain_length
Length of link chain. (%f in **printf** format)

The following magic attributes can only be used in **@cross_section_axis_label** directives.

GPGEN_cross_section_direction
For `space` or `space_route` or `time_route` cross sections, direction calculated from current label location to next label location on the cross section (%f in **printf** format)

GPGEN_cross_section_speed
For `space_route` or `time_route` cross sections, speed calculated from current label location to next label location on the cross section (%f in **printf** format)

GPGEN_cross_section_vector
For `space_route` or `time_route` cross sections, direction and speed calculated from current label location to next label location on the cross section (%f in **printf** format)

GPGEN_cross_section_length
For `space` or `space_route` or `time_route` cross sections, length of the cross section axis (%f in **printf** format)

The following magic attributes can only be used within an **@loop_begin/@loop_end** loop using features from **@label** and **@sample_field** directives.

GPGEN_feature_attribute
The value of attributes from the current feature from set in **@loop_begin**

The following magic attributes can only be used in **@cross_section_areas**, **@cross_section_curves** and **@sample_field** directives.

GPGEN_proximity
Proximity of the sample point to the nearest area, line, link chain or scattered point in the sampled field in km (%f in **printf** format)

GPGEN_negative_proximity
Proximity (in negative direction) of the sample point to the nearest area, line, link chain or scattered point in the sampled field in km (%f in **printf** format)

There are several magic attributes that can be used only with specific directives. These magic attributes also begin with the string **`GPGEN_`**, and are described with the corresponding directive.

1.11 Graphics Geography Files

The PSMet and SVGMet applications allow the user to specify geography to use in their output graphics products. The applications search for geography files from default directories, but the user can add another directory to search first, if desired.

The **`directories`** block of the FPA setup file may contain the following entry identifying the directory location for geography files:

```
directories
  {
  ...
  Maps                map_directory
  ...
  }
```

The PSMet and SVGMet applications will search for a named geography file from the following directories in the order given:

`map_directory` (if entered)
Maps (in the Data directory)
`$FPA/data/common/CommonMaps`

1.12 Graphics Symbol Files

The FPA directories contain a number of predefined symbols that can be displayed by the PSMet and SVGMet applications. The symbols are created as a set size, but can be scaled as desired. Each symbol is created as a set of drawing instructions, based on a centre location of (0,0). This centre location does not necessarily have to be at the physical centre of the symbol itself, but the symbol is always displayed relative to this centre location. The symbol location can, however, be modified by an absolute value , or relative to the size of the symbol. The following keywords can be used:

symbol, mark
the name of the symbol file

symbol_scale, mark_scale
scale the symbol by a certain percentage

x_off, attribute_x_off
shift symbol horizontally by a set amount

y_off, attribute_y_off
shift symbol vertically by a set amount

justification, attribute_justification
horizontal justification based on left or right edge of symbol

attribute_vertical_just
vertical justification based on top or bottom edge of symbol

Examples of predefined symbols for the PSMet and SVGMet applications are shown in Appendix A. The default colour for these symbols is Blue, and the default size is also shown.

The PSMet and SVGMet applications also allow the user to create and use their own symbols in the output graphics products. The applications search for symbol files from default directories, but the user can add another directory to search first, if desired.

The **`directories`** block of the FPA setup file may contain the following entries identifying directory locations for symbol files:

```
directories
  {
  ...
  psmet_symbols              symbol_directory
  svgmet_symbols             symbol_directory
  ...
  }
```

The PSMet application will search for a named symbol file from the following directories in the order given:

`symbol_directory` (if entered)
`$HOME/setup/pdf/psmet/common/ps`
`$FPA/setup/pdf/psmet/common/ps`

The SVGMet application will search from the following directories in the order given:

```
symbol_directory (if entered)
$HOME/setup/pdf/svgmet/common/svg
$FPA/setup/pdf/svgmet/common/svg
```

Predefined symbol files can be found in the directories:

```
$FPA/setup/pdf/psmet/common/ps
$FPA/setup/pdf/svgmet/common/svg
```

Descriptions of the symbols in these directories can be found in the files:

```
$FPA/setup/pdf/psmet/common/ps_symbol_list
$FPA/setup/pdf/svgmet/common/svg_symbol_list
```

1.13 Graphics Pattern Files for Drawing Lines

The FPA directories contain a number of predefined pattern files that can be used to draw lines in the PSMet and SVGMet applications. The pattern files are created with a set width and height of the pattern, but the pattern can be stretched in width or length as desired. The following keywords can be used:

pattern
the name of the pattern file

pattern_width
the width of the pattern perpendicular to the line

pattern_length
the length of the pattern along the line

Examples of predefined pattern files for the PSMet and SVGMet applications are shown in Appendix B. Appendix B also shows how the **`pattern_width`** and **`pattern_length`** keywords affect the display of a pattern.

The PSMet and SVGMet applications also allow the user to create and use their own patterns for drawing lines in the output graphics products. The applications search for pattern files from default directories, but the user can add another directory to search first, if desired.

The **`directories`** block of the FPA setup file may contain the following entry identifying the directory location for pattern files:

```
directories
  {
  ...
  patterns        pattern_directory
  ...
  }
```

The PSMet and SVGMet applications will search for a named pattern file from the following directories in the order given:

pattern_directory (if entered)
`$HOME/config/patterns`
`$FPA/config/patterns`

Predefined pattern files can be found in the directory:

```
$FPA/config/patterns
```

1.14 Graphics Fonts and Spacing for Text

The PSMet and SVGMet applications use various keywords to control the font and spacing used to display text. (Note that the values for these keywords default to the current presentation if they are not present in a particular directive.)

The size of text characters varies from font to font, and the **`width_scale`** and **`height_scale`** keywords are used to estimate the approximate size of characters (compared to a square box) so as to be able to position text strings with respect to the left, right, top or bottom edge of other text strings. The following keywords can be used:

font
the name of the text font style

text_size
size of text characters

width_scale, attribute_width_scale
approximate width of characters with respect to text_size

height_scale, attribute_height_scale
approximate height of characters with respect to text_size

Examples of available fonts for the PSMet and SVGMet applications are shown in Appendix C.

1.15 Graphics Colours for Text, Lines, and Areas

The PSMet and SVGMet applications use various keywords to control colours for display of text, lines, and areas. (Note that the values for these keywords default to the current presentation if they are not present in a particular directive.)

The characters for text fonts are described by an outline that can be drawn, as well as the interior of the outline that can be filled with colour. The following keywords can be used:

line_width
the thickness of the character outlines

outline
the colour of the character outlines

fill
the colour of the interior of the character outlines

outline_first
display the character outlines first (with a value of `yes`) or last (with a value of `no`).

Note that outlines for text fonts are usually drawn last, that is, after the interior of the outlines are filled with colour

Lines can be displayed as a continuous curve, or as a sequence of filled areas, depending on the pattern file used for displaying the line. Multicoloured lines require specification of the line colours more than once.

Lines drawn as a continuous curve use the following keywords:

line_width
the thickness of the curve

outline
the colour (or colours) of the curve

Lines drawn as a sequence of filled areas use the following keywords:

line_width
the thickness of the outlines for the sequence of areas

outline
the colour (or colours) of the outlines for the sequence

fill
the colour of the interior of the outlines for the sequence

Areas can be displayed with a line for the boundary (using any of the keywords given above for lines) as well as with the interior area filled with colour or symbols. The following keywords are used for the displaying the interior of areas:

interior_fill
the colour of the interior of the area

symbol_fill_name
the display for symbols within the interior of the area (defined in the directive **@symbol_fill_display**)

Examples of predefined X11 colour names for the PSMet and SVGMet applications are shown in Appendix D.

1.16 Graphics Arrow Heads and Tails

The PSMet and SVGMet applications use various keywords to allow lines to be displayed with various styles of arrow heads or arrow tails. The directives that allow arrow heads or tails contain the **`arrow_name`** keyword, and the keywords used to set the arrow display parameters are found in the **@arrow_display** directive.

Examples of arrow display for the PSMet and SVGMet applications are shown in Appendix E. Appendix E also shows how the various keywords affect the display of arrow heads and tails.

1.17 Graphics Wind Barbs

The PSMet and SVGMet applications use various keywords to allow winds or vector winds to be displayed as wind barbs. The directives that display wind barbs contain the **`format`** or **`attribute_format`** keyword, with a value of **`wind_barb`**. The keywords used to set the wind barb display parameters are found in the **@wind_barb_presentation** directive.

Examples of the display of wind barbs for the PSMet and SVGMet applications are shown in Appendix F. Appendix F also shows how the various keywords affect the display of wind barbs.

1.18 Errors and Warnings

The Graphics Product Generator applications are designed to process all product definition file directives sequentially. The applications will halt and output an error message to the FPA log file if they encounter a directive that contains an unacceptable keyword, or if a required keyword has not been entered. The applications may output an incomplete product if an error condition has been encountered.

The Graphics Product Generator applications may also output warnings to the FPA log file. The user should check these to ensure that the indicated warning is an acceptable condition.

Users developing new graphics products may wish to make use of the **@verbose** directive. Setting the verbose mode to **`yes`** within a product definition file will result in additional comments being output as the product is run, which may highlight unexpected problems. The verbose mode should be re-set to **`no`** at a later point in the file, to end output of these additional comments.

New versions of the Graphics Product Generator are designed to simplify changes and additions to directives and keywords. The **@version** directive controls the output of additional comments that will help developers to update "pdf" files to new versions of the Graphics Product Generator. When a "pdf" file with an older **@version** value is run, a set of warnings will be output in the FPA log file; but the desired product should still be generated. The warnings in the FPA log file will indicate which directives and keywords need to be changed. Once all the required changes have been made, the **@version** directive should be changed to the newer version. Using obsolete directives or keywords with an up-to-date **@version** will result in error messages in the FPA log file.

Chapter 2

PSMet/SVGMet

The PSMet and SVGMet applications allow the user to develop graphical products using data from the FPA. The graphical products can be customized to display the data on different geographical backgrounds, in different formats, and with various patterns, symbols, and headers. This section describes the PSMet and SVGMet files and commands in more detail.

2.1 PSMet/SVGMet Input and Output Files

The PSMet and SVGMet applications use the following input files:

Product Definition Files(.fpdf)
PSMet/SVGMet directives (commands) for sampling and displaying FPA data or other information in a graphical format. These files end with the suffix .fpdf

PostScript Format Graphics Files (.ps)
Graphical symbols used in PSMet, in PostScript format. These files end with the suffix .ps

Examples of these files can be found in the directory `$FPA/setup/pdf/psmet/common/ps`

Scalable Vector Graphic (SVG) Files (.svg)
Graphical symbols used in SVGMet, in SVG format. These files end with the suffix .svg

Examples of these files can be found in the directory `$FPA/setup/pdf/svgmet/common/svg`

FPA Metafiles
FPA files containing patterns for display of lines or outlines

Examples of these files can be found in the directory `$FPA/config/patterns`

FPA Metafiles
FPA files containing FPA data for sampling or display. These files are accessed automatically by the directives

Look Up Table Files (.tab)
Cross reference tables for display formats for values. These files end with the suffix .tab

Location Look Up Table Files (.ltab)
Cross reference tables for set locations (with optional times). These files end with the suffix .ltab

Vertical Look Up Table Files for Cross Sections (.ztab)
Cross reference tables for vertical levels for cross sections. These files end with the suffix .ztab

Wind Look Up Table Files (.wtab)
Cross reference tables for display formats for winds. These files end with the suffix .wtab

Vector Look Up Table Files (.vtab)
Cross reference tables for display formats for vector parameters. These files end with the suffix .vtab

The PSMet application produces the following output file(s):

PostScript Format Graphics File (.ps)
Graphic output in simple PostScript format.

The SVGMet application produces the following output file(s):

Scalable Vector Graphics File (.svg)
Graphic output in the Scalable Vector Graphics Format ("SVG").

2.2 Special PSMet/SVGMet Directives

The following directives have special locations or usages in the PSMet and SVGMet product definition files. A complete description of each of these directives is given in a following section.

The @version directive is always found at the beginning of the main PSMet/SVGMet product definition file; it must always be the first directive encountered! The @file_name directive identifies the output file to be used. This directive is normally found at the beginning of the main product definition file too, though it may also be used later in the product definition files to open another output file. The @file_close directive is used to close an output file.

@version
PSMet/SVGMet version number. This must be the first directive!

@file_name
Name and location of PSMet/SVGMet output file

@file_close
Close the PSMet/SVGMet output file

The @verbose directive can be used while developing new graphics products. It will provide extra diagnostic output to help determine why a particular directive does not produce the desired output.

@verbose
Provide extra diagnostic output during development

The @process directive can be used to launch an external process from within PSMet or SVGMet. For example, this directive could be used to display the PSMet or SVGMet output file.

@process
Launch an external process

The following directives can be used to simplify the directive and keyword commands in a set of product definition files. The @group @include directives are used to combine keywords or directives. The @loop_begin @loop_end directives allow a given sequence of directives to be repeated several times with changing keyword values.

@group
Combine a series of repeated keywords

@include
Process a series of directives from another "pdf" file

@loop_begin
Beginning of loop for repeating directives with changing keyword values

@loop_end
End of loop for repeating directives

@loop_location_look_up
Create a temporary location look up table from a point, line or link chain feature.

The following directives control positioning for the output display. The @size directive is used to set the actual page size of the output display. The @display_units directive sets the units to be used for all calculations of position on the output display. (The units can be whatever the user feels most comfortable dealing with.) The @anchor directive is used for defining positions on the output display. This directive can be reset any number of times.

@size
Size of the output display (the paper size)

@display_units
Units for positioning and display of graphics

@anchor
Define a reference position for displaying graphics

The @projection @mapdef @resolution @define_map_placement directives control the map background used for sampling or display. (Note that any background map should be completely contained by the map defined in the FPA setup file and the FPA metafiles, or errors could result when sampling or displaying the data. Errors can also occur if the locations for sampling data are beyond the limits of the background map defined by these directives.) These directives are usually found at the beginning of PSMet and SVGMet product definition files.

@projection
Map projection for the current map

@mapdef
Map definition for the current map

@resolution
Spline resolution for the current map

@define_map_placement
Size and location on the page for the current map

The @presentation directive sets the default display parameters for all subsequent directives. Many of the keywords in PSMet and SVGMet have default values that are set automatically when the application is run. These parameters are reset by keywords in the @presentation directive. However, many of the keywords found in the @presentation directive are also found in other directives. The effect of such keywords is to override the default display parameters for that directive only; the default display parameters will not be changed.

@presentation
Default display parameters for appearance of graphics

2.2.1 Special format specific directives available only in SVGMet

The SVG format allows for grouping of display objects, which can be useful for moving a complex label, for example, in a graphics post processor such as Adobe Illustrator. The @gpgen_group_begin @gpgen_group_end directives can be used to create an SVG grouping.

@gpgen_group_begin
Start grouping of display objects.

@gpgen_group_end
End grouping of display objects.

The SVG format has a number of powerful features, not all of which could be incorporated into SVGMet. The @gpgen_insert directive can be used to insert such features directly into the output SVG format file.

@gpgen_insert
Insert SVG commands directly into the output SVG format file.

2.3 PSMet/SVGMet Display Directives

The following directives control the display of all parameters in PSMet and SVGMet. A complete description of each of these directives is given in a following section.

The following directives are used to set special parameters used in display of graphics features:

@set_source
Set a time for extracting FPA Metafile data

@filter
Define a distance for sampling the number of locations to display along lines or outlines

@define_table
Define a table for displaying data

@table_site
Set locations in a table

@define_cross_section
Define a cross section for displaying data

@define_sample_grid
Define a grid of locations for sampling values

@define_sample_list
Define a list of locations for sampling values

The following directives are used to set parameters to control the format of graphics features to be displayed:

@perspective_view
Display graphics on the current map with a tilted perspective

@arrow_display
Set display parameters for arrow heads and tails on lines

@label_display
Set parameters for displaying a box, ellipse, or underline about a parameter

@symbol_fill_display
Set parameters for filling an area with symbols

@contour_presentation
Set parameters for displaying contours

@reset_contour_presentation
Remove all contour display parameters

@geo_presentation
Set parameters for displaying geographical features

@reset_geo_presentation
Remove all geographic display parameters

@wind_presentation
Set parameters for displaying winds as text strings or symbols

@wind_barb_presentation
Set parameters for displaying winds as wind barbs

@vector_presentation
Set parameters for displaying vector field parameters as text strings or symbols

The following directives are used to display symbols, objects, or text:

@add
Display a graphic symbol (or bitmap file) on the current map, in a table, or at an arbitrary location

@text
Display a text string (or file) on the current map, in a table, or at an arbitrary location

@legend
Display a graphic symbol or text string off the page (for use in a graphics post processor)

@write_time
Display a formatted time string

@draw_distance_scale
Display a distance scale.

@distance_scale_ticks
Display tick marks on a named distance scale.

@distance_scale_labels
Display labels on a named distance scale.

@box
Display a centred box on the current map, in a table, or at an arbitrary location

@ellipse
Display a centred ellipse on the current map, in a table, or at an arbitrary location

@background
Display an outline around the current map

@define_line
Define an arbitrary line

@draw_line
Display a defined line

@draw_table_line
Display a defined line in a table

@draw_cross_section_line
Display a defined line in a cross section

The following directives are used to display features or parameters extracted from FPA Metafiles on the current map:

@areas
Display outlines from FPA discrete fields

@contours
Display contours from FPA continuous or vector fields

@images
Display radar and satellite imagery from FPA image database.

@lines
Display patterned lines from FPA line fields

@label
Display labels from FPA fields as point values

@lchain_nodes
Display nodes from FPA link chain fields.

@lchain_tracks
Display tracks from FPA link chain fields.

@sample_field
Sample values from FPA fields

@sample_wind
Sample winds from FPA fields

@geography
Display geographic features from FPA Maps

The following directives are used to display features or parameters extracted from FPA Metafiles at an arbitrary location or in a table:

@label
Display labels from FPA fields as point values

@sample_field
Sample values from FPA fields

@sample_wind
Sample winds from FPA fields

The following directives are used to display features or parameters extracted from FPA Metafiles in a cross section:

@cross_section_axis_labels
Display parameters along horizontal or vertical cross section axes

@cross_section_areas
Display parameters from FPA fields as outlines in a cross section

@cross_section_contours
Display values from FPA fields at several levels as contours in a cross section

@cross_section_curves
Display values from FPA fields as a curve in a cross section

@sample_field
Sample values from FPA fields at cross section locations

@sample_wind
Sample winds from FPA fields at cross section locations

2.4 Differences Between PSMet and SVGMet

The directives and keywords used by the PSMet and SVGMet applications are virtually identical.

One difference in the two applications is in the definition of the origin for displaying graphics. PostScript sets the origin to the lower left corner of the output display, while the SVG format sets the origin to the upper left corner of the output display. All directives and keywords set the origin as the centre of the page, and so both applications require the @size directive to set the output display size (the paper size) so that the output will be properly located.

Another difference is that the SVG format allows for grouping of display objects, which can be useful for moving a complex label, for example, in a graphics post processor such as Adobe Illustrator. The two directives used for SVGMet grouping are @gpgen_group_begin (to begin grouping of display objects) and @gpgen_group_end (to end a grouping).

The SVG format has a number of powerful features, not all of which could be incorporated into SVGMet. The @gpgen_insert directive can be used to insert such features directly into the output SVG format file.

All other directives behave similarly, and generate similar graphical results in both applications.

2.5 Complete Description of All PSMet/SVGMet Directives

The following table consists of an alphabetical listing of all PSMet/SVGMet directives, a list of keywords that can be used within each directive, a brief description of the directives and keywords, and one or more examples of each directive.

Note that most examples are found in product definition files in directories "`$FPA/setup/pdf/psmet/examples`" or "`$FPA/setup/pdf/svgmet/examples`", and other examples are included to demonstrate additional functionality.

2.5.1 @add

Display a graphic symbol (or file) on the current map or in a table or at a given location.

KEYWORD	TYPE	DESCRIPTION
symbol	string	Name of the graphic symbol file to display (Note: full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)
scale	real	Scale factor wrt original graphic (in percent)
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
rotate_to_latitude	yes/no	Rotate parallel to latitude. **rotation** is appended. (Default is no).
or		
rotate_to_longitude	yes/no	Rotate parallel to longitude. **rotation** is appended. (Default is no).
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
map_x	real	Position on x-axis of map (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units**)
map_y	real	Position on y-axis of map (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units**)
map_units	real	Number of meters (or degrees) per map unit (Default is **map_units** from @mapdef directive)
location_ident	string	Identifier for location in **location_look_up**
For table display ...		
table_name	string	Table name (from @define_table) (Note: the object will be displayed once for each call to the @table_site directive)
For grid display ...		
grid_name	string	Grid name (from @define_sample_grid) (Note: the object will be displayed once for each location on the grid)

KEYWORD	TYPE	DESCRIPTION
For list display ...		
list_name	string	List name (from @define_sample_list) (Note: the object will be displayed once for each location in the list)
For map, table or list display ...		
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.1

```
@add
{
  symbol = lgticg;
  scale  = 12;
  x_off  = 0.25;
  y_off  = 1.75;
}
```

Display the graphic from file "lgticg" at a location offset 0.25 "units" to the right and 1.75 "units" above the current "anchor position". The graphic will be 12% of its original size.
(Note that the default PSMet file would be `$FPA/setup/pdf/psmet/common/ps/lgticg.ps`, and that the default SVGMet file would be `$FPA/setup/pdf/svgmet/common/svg/lgticg.svg`.)
(Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

Example 2.2

```
@add
{
  symbol = circle_mark;
  scale  = 50;
  lat    = 45:52N;
  lon    = 66:32W;
}
```

Display the graphic from file "circle_mark" at a latitude of 45 degrees and 52 minutes North, and a longitude of 66 degrees and 32 minutes West on the current map. The graphic will be 50% of its original size, and not offset.

(Note that the default PSMet symbol would be `$FPA/setup/pdf/psmet/common/ps/circle_mark.ps`, and the default SVGMet symbol would be `$FPA/setup/pdf/svgmet/common/svg/circle_mark.svg`.)

2.5.2 @anchor

Set a new reference location, or anchor position, for subsequent displays on the current map.

The following directives are affected by the anchor position:

@add @box @define_map_placement @define_table @define_cross_section @draw_line @ellipse @label @lchain_nodes @sample_field @sample_wind @text @write_time.

However, if the directive contains a **table_name** keyword, the anchor position for that directive defaults to the table location set in @define_table.

Setting an anchor position of **ref = map** is necessary for displaying sampled values at the correct location on the current map for directives: @add @box @ellipse @label @lchain_nodes @sample_field @sample_wind @text.

Note, however, that graphics displayed for the following directives are always anchored on the current map, regardless of the anchor position: @areas @background @contours @geography @lchain_tracks @lines.

KEYWORD	TYPE	DESCRIPTION	
ref	string	Starting point for new reference location	
		One of:	
		map	reset to centre of current map
		map_latlon	wrt lat/lon on current map
		absolute	wrt page centre
		lower_left	wrt current map
		centre_left	wrt current map
		upper_left	wrt current map
		lower_centre	wrt current map
		centre	wrt current map
		upper_centre	wrt current map
		lower_right	wrt current map
		centre_right	wrt current map
		upper_right	wrt current map
		current	wrt current anchor position
		xsect_lower_left	wrt current cross-section
		xsect_centre_left	wrt current cross-section
		xsect_upper_left	wrt current cross-section
		xsect_lower_right	wrt current cross-section
		xsect_centre_right	wrt current cross-section
		xsect_upper_right	wrt current cross-section

KEYWORD	TYPE	DESCRIPTION
x	real	Horizontal offset from starting point (in display units)
y	real	Vertical offset from starting point (in display units)
For **xsect_*** ...		
cross_section_name	string	Cross section name (from @define_cross_section)
For **map_latlon** ...		
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
or		
location_ident	string	Identifier for location in **location_look_up**
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)

Example 2.3

```
@anchor { ref = absolute; x = -3.7; y = 1.5; }
```

Set a new reference location offset -3.7 "units" horizontally and 1.5 "units" vertically from the centre of the page. (Note that "units" is set by @display_units.)

Example 2.4

```
@anchor { ref = map; }
```

Reset the reference location to the centre of the current map

Example 2.5

```
@anchor { ref = map_latlon; lat = 42:30N; lon = 71:00W; }
```

Reset the reference location to latitude 42 degrees and 30 minutes North and longitude 71 degrees West on the current map

2.5.3 @areas

Display outlines from a "discrete" field (or from more than one "discrete" field) on the current map.

The **category_attribute** and **category** keywords can be used one or more times to display only selected areas.

The **attribute** and **look_up** keywords can be used to choose different presentations for areas with different attributes.

<table>
<tr><th>KEYWORD</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td colspan="3">Note that <b>element</b> and <b>level</b> can be specified more than once in this directive.
Note that if a single element name is matched with a list of level names, or vice versa, the single name must be given first. Multiple element and level name lists should have the same number.</td></tr>
<tr><td><b>element</b></td><td>string</td><td>FPA element name</td></tr>
<tr><td><b>element_list</b></td><td>string</td><td>List of FPA element names</td></tr>
<tr><td><b>level</b></td><td>string</td><td>FPA level name</td></tr>
<tr><td><b>level_list</b></td><td>string</td><td>List of FPA level names</td></tr>
<tr><td colspan="3">Note that <b>category_attribute</b> and <b>category</b> can be specified more than once in this directive, and that <b>category_cascade</b> determines how to combine more than one instance.</td></tr>
<tr><td><b>category_cascade</b></td><td>string</td><td>How to combine more than one instance of <b>category_attribute</b> and <b>category</b>
One of: <b>and</b> or <b>or</b>
(default is <b>and</b>)</td></tr>
<tr><td><b>category_attribute</b></td><td>string</td><td>FPA attribute containing category names
(default is <b>FPA_category</b>)</td></tr>
<tr><td><b>category</b></td><td>string</td><td>FPA category name (or list of names) for selecting which objects to display
(Note: name(s) are matched with the value sampled from <b>category_attribute</b>)
(default is <b>All</b> names)</td></tr>
<tr><td><b>attribute</b></td><td>string</td><td>FPA attribute containing value to match with values in <b>look_up</b> table
(default is <b>FPA_auto_label</b>)</td></tr>
<tr><td><b>look_up</b></td><td>string</td><td>File name for the look up table
(Note: ".tab" suffix is optional)</td></tr>
<tr><td><b>area_type</b></td><td>string</td><td>One of:
<b>subareas</b> Draw each subarea of a divided area as a separate area with defined outline and fill (default)
<b>boundary</b> Draw the entire area with a single outline and fill
<b>divides</b> Draw only dividing lines</td></tr>
</table>

PRESENTATION KEYWORD	TYPE	DESCRIPTION
interior_fill	string	Colour for area interior (see @presentation)
symbol_fill_name	string	Symbol fill name for area interior One of: **none** or a symbol fill name (set in @symbol_fill_display)
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)

Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once in this directive, for drawing multi-coloured boundaries!

line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.6

```
@areas
{
  element            = weather_system;
  level              = surface;
  category_attribute = FPA_category;
  category           = cloud freezing frozen precip vis;
  attribute          = clds_and_wx;
  look_up            = weather_day;
}
```

Display all areas with a value in attribute FPA_category of cloud, freezing, frozen, precip, or vis from the "discrete" field surface system_weather, with a presentation for each category determined by matching the value in attribute clds_and_wx with values in the look up table "weather_day.tab".

2.5.4 @arrow_display

Set display parameters for drawing lines with arrow heads or arrow tails. See Appendix E for further explanation of keywords for this directive.

Note that arrows use the same presentation as the lines they are drawn on, and that arrows on multi-coloured lines use the last colour presentation specified.

KEYWORD	TYPE	DESCRIPTION
arrow_name	string	Arrow display name
arrow_length	real	Length of arrow head (in display units)
arrow_angle	real	Angle from line to arrowhead (in counter-clockwise decimal degrees)
return_angle	real	Angle from line to return arrowhead (in counter-clockwise decimal degrees)
length_offset	real	Distance from endpoint of line to tip of arrow (in display units along path of line)
width_offset	real	Distance from endpoint of line to tip of arrow (in display units to left of path of line)
head_length	real	Width of arrow head (as a fraction of **arrow_length** keyword value)
tail_length	real	Width of arrow tail (as a fraction of **arrow_length** keyword value)
arrow_features	string	One of: **head** (draw arrow head), **tail** (draw arrow tail), **both** (draw arrow head and tail), **both_tail_reversed** (draw arrow head and reversed tail)

Example 2.7

```
@arrow_display
{
  arrow_name     = arrowhead;
  arrow_length   = 0.10;
  arrow_angle    = 50;
  return_angle   = 35;
  length_offset  = 0.0;
  width_offset   = 0.0;
  head_length    = 0.2;
  tail_length    = 0.2;
  arrow_features = head;
}
```

Set parameters for an arrow display called arrowhead. The head of the arrow will be drawn 0.10 "units" long and 0.02 "units" wide, at an angle of 50 degrees to the line and a return angle of 35 degrees. The tip of the arrow head will be located at the end of the line and not offset.
(Note that "units" is set by @display_units.)

2.5.5 @background

Display an outline around the current map.

Note that the map size is set in the @define_map_placement directive.

KEYWORD	TYPE	DESCRIPTION
extra_x	real	Additional width for outline (in display units)
extra_y	real	Additional height for outline (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
interior_fill	string	Colour for area interior (see @presentation)

Example 2.8

```
@background
{
  interior_fill = CMYK 100 30 0 0;
  line_width    = 0.050;
  extra_x       = 0.030;
  extra_y       = 0.030;
}
```

Display an outline around the current map that is 0.030 "units" larger than the map size set in the @define_map_placement directive (or 0.015 "units" larger on each edge). The outline will be 0.050 "units" wide (or +/- 0.025 "units" wide) and thus will overlap the map by 0.010 "units". The outline style and colour will be set from the current defaults, but the outline will be filled a with a colour composed of 100% Cyan and 30% Magenta.
(Note that "units" is set by @display_units.)

2.5.6 @box

Display a centred box on the current map or in a table or at a given location.

KEYWORD	TYPE	DESCRIPTION
`width`	real	Width (in display units)
`height`	real	Height (in display units)
`rotation`	real	Angle of rotation (in counter-clockwise decimal degrees)
`rotate_to_latitude`	yes/no	Rotate parallel to latitude. **`rotation`** is appended. (Default is no).
or		
`rotate_to_longitude`	yes/no	Rotate parallel to longitude. **`rotation`** is appended. (Default is no).
`x_off`	real	Horizontal offset wrt current anchor position (in display units)
`y_off`	real	Vertical offset wrt current anchor position (in display units)
For map display ...		
`lat`	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
`lon`	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
`map_x`	real	Position on x-axis of map (Note: position is measured from **`xmin`** from @mapdef directive, in units of **`map_units`**)
`map_y`	real	Position on y-axis of map (Note: position is measured from **`ymin`** from @mapdef directive, in units of **`map_units`**)
`map_units`	real	Number of meters (or degrees) per map unit (Default is **`map_units`** from @mapdef directive)
`location_ident`	string	Identifier for location in **`location_look_up`**
For table display ...		
`table_name`	string	Table name (from @define_table) (Note: the object will be displayed once for each call to the @table_site directive)
For grid display ...		
`grid_name`	string	Grid name (from @define_sample_grid) (Note: the object will be displayed once for each location on the grid)

For list display ...

KEYWORD	TYPE	DESCRIPTION
list_name	string	List name (from @define_sample_list) (Note: the object will be displayed once for each location in the list)
For map, table or list display ...		
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
interior_fill	string	Colour for area interior (see @presentation)
symbol_fill_name	string	Symbol fill name for area interior One of: **none** or a symbol fill name (set in @symbol_fill_display)

Example 2.9

```
@box
{
  width         = 2.0;
  height        = 0.5;
  x_off         = -1.00;
  y_off         = -0.25;
  outline       = CMYK 0 0 0 100;
  interior_fill = CMYK 0 0 0 0;
  line_width    = .050;
}
```

Display a box 2.0 "units" wide by 0.5 "units" high centred at a location offset 1.00 "units" to the left and 0.25 "units" below the current "anchor position". (Therefore, the upper right corner of the box will be located at the anchor position.) The box outline will be 0.050 "units" wide in a colour composed of 100% Black, and the box will be filled with White.
(Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

2.5.7 @contour_presentation

Set the appearance of contours displayed with the @contours directive.

Note that each contour displayed is checked against the contour presentations defined for each value in the **values** keyword (in the order entered) followed by each range of values in the **range** keyword (in the order entered). Note that each subsequent @contour_presentation directive will add additional contour presentations to the **values** and **range** lists of values to check, or modify existing presentations for matching values.

Note that the @reset_contour_presentation directive resets all lists of contour presentations currently used.

KEYWORD	TYPE	DESCRIPTION
units	string	FPA units for contours to display (Note: **values** and **range** are given in these units)
values	real	Value (or list of values) for which the contour presentation applies
range	range	Minimum and maximum values for which the contour presentation applies (see Format of Values in a Range, (Section 1.4))

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
interior_fill	string	Colour for area interior (see @presentation)
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)

Example 2.10

```
@contour_presentation { units = mb; range = [ 800 : 1200 ];
  line_width = 0.015; line_style = solid; outline = CMYK 0 0 0 40; }
```

Display all contours from 800 mb to 1200 mb (inclusive) with a solid line 0.015 **units** wide. The line will be of a colour composed of 40% Black.
(Note that **units** is set by @display_units.)

Example 2.11

```
@contour_presentation { range = [ 3 : 5 ); units = m; line_width = 0.05;
  line_style = solid; outline = X11 Yellow; fill = none; }
```

Display all contours from 3 m to less than 5 m with a solid Yellow line 0.05 **units** wide.
(Note that **units** is set by @display_units.)

2.5.8 @contours

Display contours from a "continuous" field (or magnitude contours from a "vector" field) on the current map.

Note that the appearance of contours is usually given by the presentations set in the @contour_presentation directive, but that any presentation parameters set here will take precedence over those set in @contour_presentation.

Note that **display_as_areas = yes** is used to display colour banded contours. The **min** keyword value (or the minimum value for each range if **base** and **interval** are used) is used to determine the contour presentation to display.

KEYWORD	TYPE	DESCRIPTION
element	string	FPA element name
level	string	FPA level name
equation	string	FPA equation string for calculating a field (refer to Appendix B of Administrator's Guide for format of equations)
units	string	FPA units for contours to display (Note: **values min max base interval** are given in these units)
display_as_areas	yes/no	Display contour range as a colour band? (default is **no**)
values	real	Value (or list of values) of contours to display
or		
min	real	Minimum contour value to display
max	real	Maximum contour value to display
base	real	Base contour value to display (**interval** is measured from this value)
interval	real	Interval for contour values to display

PRESENTATION KEYWORD	TYPE	DESCRIPTION
interior_fill	string	Colour for area interior (see @presentation)
symbol_fill_name	string	Symbol fill name for area interior One of: **none** or a symbol fill name (set in @symbol_fill_display)
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)

Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once in this directive, for drawing multi-coloured contours!

line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.12

```
@contours
{
  element  = pressure;
  level    = msl;
  units    = mb;
  pattern  = simple;
  min      = 900;
  max      = 1060;
  base     = 1000;
  interval = 4;
}
```

Display contours from the msl pressure field at every multiple of 4 mb away from 1000 mb, beginning at 900 mb and ending at 1060 mb. The contours will be drawn with a simple line, and with a presentation previously set by the @contour_presentation directive.

Example 2.13

```
@contours
{
  <MaxTempAreas;
  min           = -35;
  max           = -30;
  interior_fill = X11 White;
}
```

Note that the keywords for the "macro" called "MaxTempAreas" are defined as an example in the @ps_group directive, given below, and are automatically included as part of this directive.
Display the -35 to -30 degreesC contour band from the surface max_temp field as an area (or areas) filled with White. The minimum and maximum limits of the contour range will not be drawn.

2.5.9 @cross_section_areas

Display outlines in a cross section. The outlines are created from two vertical attributes from a "discrete" field, or from a single vertical attribute from a "discrete", "line", "link chain" or "scattered" field (or from a vertical value from a "continuous" or "vector" field) and the top or bottom edge of the cross section.

Outlines are created by sampling the field along the horizontal axis of the cross section, and creating an enclosed area if parameter values are found for the two keywords **vertical_attribute_upper** and **vertical_attribute_lower** (or calculated using the equation from **vertical_equation**, if either keyword is missing). The vertical extent of the outline is determined by mapping the values for the two keywords to the vertical look up table **vertical_look_up**.

Note that entries in **location_look_up** table MUST be entries from the location look up table in directive @define_cross_section.

Note that the **category_attribute** and **category** keywords can be used one or more times to display only selected areas.

It is now possible to sample information from ASCII data files, and display the data on a cross section. Note that all data files must be arranged row by row, with each data parameter in a separate column. The keywords **vertical_data_file**, **vertical_data_file_format** and **vertical_data_file_units** provide control for this action. Note that all parameters in the data file are accessed by magic attributes, as follows:

GPGEN_ident	Identifier string
GPGEN_lat or **GPGEN_lat_ddmm**	Latitude location
GPGEN_lon or **GPGEN_lon_ddmm**	Longitude location
GPGEN_label	Data string
GPGEN_value	Data value

The keywords **proximity** and **proximity_units** allow sampling attributes from discrete areas, line features, link chains or scattered points within the given proximity of the sample location, and displaying the sampled values as areas. The magic attribute **GPGEN_proximity** can be used to extract the distance from the sample location to the feature in km.

KEYWORD	TYPE	DESCRIPTION
cross_section_name	string	Cross section name (from @define_cross_section)
vertical_element	string	FPA element name for vertical parameters
vertical_level	string	FPA level name for vertical parameters
vertical_attribute_upper	string	FPA attribute for upper value (Note: **GPGEN_cross_section_top** can be used for the top of the cross_section) (default is **GPGEN_default_attribute**)
vertical_attribute_lower	string	FPA attribute for lower value (Note: **GPGEN_cross_section_base** can be used for the bottom of the cross section) (default is **GPGEN_default_attribute**)

KEYWORD	TYPE	DESCRIPTION
vertical_equation	string	FPA equation string for calculating values (refer to Appendix B of Administrator's Guide for format of equations)
vertical_units	string	FPA units for field values
location_look_up	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional) (Default is **location_look_up** from @define_cross_section directive)
or		
location_distances	real(s)	A list of distances from look up table in @define_cross_section to sample at.
location_units	string	FPA units of **location_distances**
or		
location_times	real(s)	A list of times from look up table in @define_cross_section to sample at.
or		
location_fractions	real(s)	A list (0-1) of fractions along the horizontal axis from look up table in @define_cross_section to sample at.
vertical_look_up	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional) (Default is **vertical_look_up** from @define_cross_section directive)
display_function	string	One of: **linear** point by point **step_before** step function before **step_centre** step function centred **step_after** step function after **box** sequence of boxes
proximity	real	Only display sample if area, line, link chain or scattered feature is within the given proximity
proximity_units	string	FPA units for proximity

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)

KEYWORD	TYPE	DESCRIPTION
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
vertical_data_file	string	Identify the name of the ASCII data file to sample. May contain codewords: <default>, <psmet>, <psout>, <svgmet>, <svgout>, <texmet>, <texout> or <home> to refer to PSMet, SVGMet or TexMet directories in the same FPA database.
vertical_data_file_format	string	Identify the format of the data in the ASCII data file with a list of recognized format types separated by the same characters used to separate the values in the ASCII data file. Recognized types: **identifier** Identifier string **latitude** Latitude location **longitude** Longitude location **timestamp** Valid time **label** Data string **value** Data value **units** FPA units of data value **-** Placeholder in data file Note that either the **identifier** or the **latitude** and **longitude** must be present to identify the data locations, but all other format types are optional. Note that only the **value** format type can be used to set the vertical positions in a cross section.
vertical_data_file_units	string	FPA units for data values Note that all data values are automatically converted to MKS units. Either the **units** format type or the **vertical_data_file_units** keyword are required for values in the data file that are non-MKS.

KEYWORD	TYPE	DESCRIPTION
For box display ...		
box_width	real	Width of boxes (in display units)
x_box_off	real	Horizontal offset of boxes (in display units)
y_box_off	real	Vertical offset of boxes (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
interior_fill	string	Colour for area interior (see @presentation)
symbol_fill_name	string	Symbol fill name for area interior One of: **none** or a symbol fill name (set in @symbol_fill_display)
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)
Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once in this directive, for drawing multi-coloured boundaries!		
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.14 Example vertical_data_file_format

The **vertical_data_file_format** string is a list of recognized format types separated by the same characters used to separate the values in the ASCII data file.

The following format would indicate that the identifier string is located in the first column of the ASCII data file, the time stamp in the second column, a data file value in the fourth column, and that the columns are separated by white space (blanks or tabs).

```
vertical_data_file_format = identifier timestamp - value
```

The following format would indicate that the latitude and longitude are located in the first and second columns of the ASCII data file, a data file value in the fifth column, the FPA units for the value in the fourth column, and that the columns are separated by semi-colons.

```
vertical_data_file_format = latitude;longitude;-;units;value
```

Example 2.15

```
@cross_section_areas
{
  cross_section_name       = route_cross_section;
  vertical_element         = weather_local;
  vertical_level           = surface;
  vertical_attribute_upper = cloud_top;
  vertical_attribute_lower = cloud_base;
  vertical_look_up         = xsect_heights;
  line_width               = 0.020;
  outline                  = X11 CadetBlue;
  interior_fill            = X11 SkyBlue;
  pattern                  = scallop;
  pattern_width            = 0.050;
  pattern_length           = 0.300;
}
```

Display outlines on a cross section named route_cross_section (set in the @define_cross_section directive). The outlines are created from the surface weather_local field based on the field attributes cloud_top and cloud_base. The field is sampled at each location identified in the default location look up table (set in the @define_cross_section directive), and the attribute parameters are converted to vertical locations using the vertical look up table "`xsect_heights.ztab`". The outlines will be drawn with a line 0.020 "units" wide and with a colour of X11 CadetBlue, using the pattern from file "scallop". The pattern will be 0.050 "units" wide and 0.300 "units" long, and will then repeat. The outlines will be filled with a colour of X11 SkyBlue.
(Note that the default pattern file would be `$FPA/config/patterns/scallop`. Note that "units" is set by @display_units.)

Example 2.16 Sample from a data file

```
@cross_section_areas
{
  cross_section_name        = route_cross_section;
  location_look_up          = route_temps;
  vertical_look_up          = xsect_temps;
  vertical_data_file        = temp_file.txt;
  vertical_data_file_format = - latitude longitude timestamp value units;
  vertical_units            = degreesC;
  pattern                   = simple;
  outline                   = X11 Red;
}
```

Display an area on a cross section named route_cross_section (set in the @define_cross_section directive). The area is created by sampling the data file `temp_file.txt` at the locations in the location look up table `route_temps.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive), and the values are converted to vertical locations using the vertical look up table `xsect_temps.ztab`. The temperature value is located in the fifth column of the data file and the units of temperature in data file is located in the sixth column. The value will be displayed in degreesC (vertical_units).

2.5.10 @cross_section_axis_labels

Display parameters along the horizontal or vertical axis of a cross section.

If the `axis_for_display` is lower or upper, then one label will be displayed for each location in the location look up table `location_look_up`. If the `axis_for_display` is left or right, then one label will be displayed for each location in the vertical look up table `vertical_look_up`.

Note that only magic attributes can be displayed along cross section axes!

Note that entries in `location_look_up` table MUST be entries from the location look up table in directive @define_cross_section.

KEYWORD	TYPE	DESCRIPTION
`cross_section_name`	string	Cross section name (from @define_cross_section)
`axis_for_display`	string	Axis of the cross section for display One of: `lower` lower x axis `upper` upper x axis `left` left y axis `right` right y axis
`location_look_up`	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional) (Default is `location_look_up` from @define_cross_section directive)
or		
`location_distances`	real(s)	A list of distances from look up table in @define_cross_section to sample at.
`location_units`	string	FPA units of `location_distances`
or		
`location_times`	real(s)	A list of times from look up table in @define_cross_section to sample at.
or		
`location_fractions`	real(s)	A list (0-1) of fractions along the horizontal axis from look up table in @define_cross_section to sample at.
`vertical_look_up`	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional) (Default is `vertical_look_up` from @define_cross_section directive)
`attribute`	string	Magic attribute containing value to display, or to match with values in `look_up` table (default is `GPGEN_ident`)

KEYWORD	TYPE	DESCRIPTION
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)
time_zone	string	Display the time (from a magic attribute using **strftime** format) adjusted to another time zone (see @write_time for more information)
language	string	Display the time (from a magic attribute using **strftime** format) adjusted to another language (see @write_time for more information)
format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **symbol** ***<cform>*** use look up **text** ***<cform>*** use look up **vector_symbol** for vector values **vector_text** for vector values **none** (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the axis label value. This conversion format is applied to the value before accessing the **look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)
symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent)
text_size	real	Text height for label or text found in look up table (in display units)
display_name	string	Named display (from @label_display)
display_type	string	Type of display about label One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)
width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **60** is acceptable for most font types

KEYWORD	TYPE	DESCRIPTION
height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **100** is acceptable for most font types
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left centre right**
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)

Example 2.17

```
@cross_section_axis_labels
{
  cross_section_name = route_cross_section;
  location_look_up   = route_sub;
  axis_for_display   = lower;
  attribute          = GPGEN_label;
  format             = direct;
  text_size          = 0.17;
  x_off              = 0.00;
  y_off              = -0.20;
}
```

Display labels from the magic **GPGEN_label** attribute on the lower axis of a cross section named "route_cross_section" (set in the @define_cross_section directive). The labels will be displayed at each location in the location look up table `route_sub.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive). The labels will be displayed "as is", with a size of 0.017 "units" high and offset -0.20 "units" below each axis location.
(Note that "units" is set by @display_units.)

Complex (multi-attribute) axis labels allow the display of several axis label parameters within the same directive. The entire complex label is displayed with respect to the **rotation**, **x_off**, **y_off**, **display_name**, and **display_type** keywords set above.

Attribute keywords apply only to the attribute that they follow.

The position of an attribute in a complex axis label can be related to the position of other attributes through use of the **attribute_anchor**, **attribute_ref**, **attribute_justification**, and **attribute_vertical_just** keywords. The **attribute_ref** keyword defines the starting position of the attribute with respect to a location about the symbol or text identified by the **attribute_anchor** keyword. The **attribute_justification** and **attribute_vertical_just** keywords define the horizontal and vertical positioning of the attribute with respect to this starting position.

The display parameters of each attribute default to the values of the **look_up**, **justification**, **format**, **symbol_scale**, **text_size**, **width_scale**, and **height_scale** keywords and the **line_style**, **line_width**, **outline**, **outline_first**, **fill**, **font**, **font_weight**, **italics**, **char_space**, and **word_space** presentation keywords set above. These default values can be replaced by the appropriate attribute keyword listed below, to create multi-colour multi-format complex axis labels.

Note that the **attribute_show** keyword can be used to "turn off" the display of attributes, while still maintain positioning with respect to other attributes.

KEYWORD	TYPE	DESCRIPTION
attribute	string	Magic attribute containing value to display, or to match with values in **attribute_look_up**
attribute_look_up	string	File name for look up table (Note: ".tab" suffix is optional) (default is **look_up** set above)
attribute_show	yes/no	Display attribute? (default is yes)

KEYWORD	TYPE	DESCRIPTION
attribute_anchor	string	Reference "anchor" for displaying attribute One of: **none** or a previously listed **attribute** (default is **none**, which corresponds to the starting position of the complex label)
attribute_ref	string	Starting position for displaying attribute (a position on the "attribute_anchor" label) One of: **lower_left** **centre_left** **upper_left** **lower_centre** **centre** **upper_centre** **lower_right** **centre_right** **upper_right** (default is **lower_centre**)
attribute_justification	string	Attribute text justification (horizontal) One of: **left** **centre** **right** (default is **justification** set above)
attribute_vertical_just	string	Attribute text justification (vertical) One of: **bottom**, **centre**, **top** (default is **bottom**)
attribute_units	string	FPA units for label wind speeds or gusts
attribute_format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **symbol** ***<cform>*** use look up **text** ***<cform>*** use look up **vector_symbol** for vector values **vector_text** for vector values **none** do not display value (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the axis label value. This conversion format is applied to the value before accessing the **attribute_look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)
attribute_symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent) (default is **symbol_scale** set above)
attribute_text_size	real	Text height for attribute or text found in look up table (in display units) (default is **text_size** set above)
attribute_display_name	string	Named display (from @label_display)

KEYWORD	TYPE	DESCRIPTION
attribute_display_type	string	Type of display about attribute One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)
attribute_width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **width_scale** set above)
attribute_height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **height_scale** set above)
attribute_x_off	real	Horizontal offset of attribute display (in display units)
attribute_y_off	real	Vertical offset of attribute display (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
attribute_line_width	real	Line width (in display units) (default is **line_width** set above)
attribute_line_style	string	Line style (see @presentation) (default is **line_style** set above)
attribute_outline	string	Line outline colour (see @presentation)
attribute_fill	string	Line fill colour (see @presentation)
attribute_outline_first	yes/no	Display outline first? (default is **outline_first** set above)
attribute_font	string	Text font name recognized by output device or graphics post processor (default is **font** set above)
attribute_font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black** (default is **font_weight** set above)
attribute_italics	yes/no	Text in italics? (default is **italics** set above)
attribute_char_space	real	Text character spacing (as percent of average character width) (default is **char_space** set above)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
attribute_word_space	real	Text word spacing (as percent of average character width) (default is **word_space** set above)

Example 2.18

```
@cross_section_axis_labels
{
  cross_section_name       = route_cross_section;
  location_look_up         = route_sub;
  axis_for_display         = lower;
  format                   = direct;
  text_size                =  0.17;
  x_off                    =  0.00;
  y_off                    = -0.08;

  attribute                = GPGEN_label;
  attribute_anchor         = none;
  attribute_justification = centre;
  attribute_vertical_just = top;

  attribute                = GPGEN_prog_time;
  attribute_anchor         = GPGEN_label;
  attribute_ref            = lower_centre;
  attribute_justification = centre;
  attribute_vertical_just = top;
  attribute_format         = direct T%s;
  attribute_text_size      =  0.12
}
```

Display complex labels for the lower axis of a cross section named "route_cross_section" (set in the @define_cross_section directive). The labels will be displayed at each location in the location look up table `route_sub.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive). The magic attribute **GPGEN_label** will be displayed "as is" with text 0.017 "units" high, and the top and horizontal centre offset -0.08 "units" below each axis location. The magic attribute **GPGEN_prog_time** will be displayed as a string with the format T%s, with text 0.012 "units" high and with the top and horizontal centre of the string located at the lower centre of the **GPGEN_label** location.
(Note that "units" is set by @display_units.)

2.5.11 @cross_section_contours

Display contours determined from "continuous" field values at several levels (or determined from "vector" field magnitudes at several levels) in a cross section.

Contours are created by sampling several fields (or equations) at different levels along the horizontal axis of the cross section, and then contouring this pseudo grid of values by mapping the horizontal positions to the location look up table **location_look_up** and mapping the level parameters to the vertical locations in the vertical look up table **vertical_look_up**.

Note that entries in **location_look_up** table MUST be entries from the location look up table in directive @define_cross_section.

KEYWORD	TYPE	DESCRIPTION
cross_section_name	string	Cross section name (from @define_cross_section)

Note that **element** and **level** can be specified more than once in this directive.
Note that if a single element name is matched with a list of level names, or vice versa, the single name must be given first. Multiple element and level name lists should have the same number.

KEYWORD	TYPE	DESCRIPTION
element	string	FPA element name
element_list	string	List of FPA element names
level	string	FPA level name
level_list	string	List of FPA level names

Note that **equation** can be specified more than once in this directive.

KEYWORD	TYPE	DESCRIPTION
equation	string	FPA equation string for calculating a field (refer to Appendix B of Administrator's Guide for format of equations)
units	string	FPA units for contours to display (Note: **values min max base** and **interval** are given in these units)
location_look_up	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional) (Default is **location_look_up** from @define_cross_section directive)

or

KEYWORD	TYPE	DESCRIPTION
location_distances	real(s)	A list of distances from look up table in @define_cross_section to sample at.
location_units	string	FPA units of **location_distances**

or

KEYWORD	TYPE	DESCRIPTION
location_times	real(s)	A list of times from look up table in @define_cross_section to sample at.

or

KEYWORD	TYPE	DESCRIPTION
location_fractions	real(s)	A list (0-1) of fractions along the horizontal axis from look up table in @define_cross_section to sample at.

KEYWORD	TYPE	DESCRIPTION
`vertical_look_up`	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional) (Default is **`vertical_look_up`** from @define_cross_section directive)
`values`	real	Value (or list of values) of contours to display

or

KEYWORD	TYPE	DESCRIPTION
`min`	real	Minimum contour value to display
`max`	real	Maximum contour value to display
`base`	real	Base contour value to display (**`interval`** is measured from this value)
`interval`	real	Interval for contour values to display
`display_as_areas`	yes/no	Display contour range as a colour band? (default is **`no`**)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`interior_fill`	string	Colour for area interior (see @presentation)
`symbol_fill_name`	string	Symbol fill name for area interior One of: **`none`** or a symbol fill name (set in @symbol_fill_display)
`pattern`	string	Line pattern for boundary (see @presentation)
`pattern_width`	real	Height of pattern (in display units)
`pattern_length`	real	Length of pattern before repeating (in display units)

Note that **`line_width`**, **`line_style`**, **`outline`** and **`fill`** can be specified more than once in this directive, for drawing multi-coloured contours!

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`line_width`	real	Line width for boundary (in display units)
`line_style`	string	Line style (see @presentation)
`outline`	string	Line outline colour (see @presentation)
`fill`	string	Line fill colour (see @presentation)

Example 2.19

```
@cross_section_contours
{
  cross_section_name = route_cross_section;
  element            = wind_speed;
  level_list         = SFC 850 700 500 250;
  units              = knots;
  location_look_up   = route_contours;
  vertical_look_up   = xsect_heights;
  pattern            = simple;
  min = 10; max = 1000; base = 10; interval = 10;
}
```

Display contours on a cross section named **`route_cross_section`** (set in the @define_cross_section directive). The contours are created from the SFC wind_speed, 850 wind_speed, 700 wind_speed, 500 wind_speed, and 250 wind_speed fields. The field is sampled at each location identified in the location look up table `route_contours.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive), and the values are converted to vertical locations using the vertical look up table `xsect_heights.ztab`. The contours are displayed every 10 knots, between 10 knots and 1000 knots, with a simple line, and with a presentation previously set by the @contour_presentation directive.

2.5.12 @cross_section_curves

Display a curve in a cross section. The curve is created from a vertical attribute in a "discrete", "line", "link chain" or "scattered" field, or from a vertical value from a "continuous" or "vector" field.

The curve is created by sampling the field along the horizontal axis of the cross section, and extracting the value from the **vertical_attribute** (for a "discrete", "line", "link chain" or "scattered" field), or the value sampled from the field or equation (for a "continuous" or "vector" field). The vertical extent is determined by mapping the extracted value to the vertical look up table **vertical_look_up**.

Note that entries in **location_look_up** table MUST be entries from the location look up table in directive @define_cross_section.

Note that the **category_attribute** and **category** keywords can be used one or more times to display only selected areas.

It is now possible to sample information from ASCII data files, and display the data on a cross section. Note that all data files must be arranged row by row, with each data parameter in a separate column. The keywords **vertical_data_file**, **vertical_data_file_format** and **vertical_data_file_units** provide control for this action. Note that all parameters in the data file are accessed by magic attributes, as follows:

GPGEN_ident	Identifier string
GPGEN_lat or **GPGEN_lat_ddmm**	Latitude location
GPGEN_lon or **GPGEN_lon_ddmm**	Longitude location
GPGEN_label	Data string
GPGEN_value	Data value

The keywords **proximity** and **proximity_units** allow sampling attributes from discrete areas, line features, link chains or scattered points within the given proximity of the sample location, and displaying the sampled values as lines. The magic attribute **GPGEN_proximity** can be used to extract the distance from the sample location to the feature in km.

KEYWORD	TYPE	DESCRIPTION
cross_section_name	string	Cross section name (from @define_cross_section)
vertical_element	string	FPA element name for vertical parameters
vertical_level	string	FPA level name for vertical parameters
vertical_attribute	string	FPA attribute for vertical value (default is **GPGEN_default_attribute**)
vertical_equation	string	FPA equation string for calculating values (refer to Appendix B of Administrator's Guide for format of equations)
vertical_units	string	FPA units for field values

KEYWORD	TYPE	DESCRIPTION
location_look_up	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional) (Default is **location_look_up** from @define_cross_section directive)
or		
location_distances	real(s)	A list of distances from look up table in @define_cross_section to sample at.
location_units	string	FPA units of **location_distances**
or		
location_times	real(s)	A list of times from look up table in @define_cross_section to sample at.
or		
location_fractions	real(s)	A list (0-1) of fractions along the horizontal axis from look up table in @define_cross_section to sample at.
vertical_look_up	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional) (Default is **vertical_look_up** from @define_cross_section directive)
display_function	string	One of: **linear** point by point **step_before** step function before **step_centre** step function centred **step_after** step function after **box** sequence of boxes
proximity	real	Only display sample if area, line, link chain or scattered feature is within the given proximity
proximity_units	string	FPA units for proximity

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)

KEYWORD	TYPE	DESCRIPTION
category	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
vertical_data_file	string	Identify the name of the ASCII data file to sample. May contain codewords: <default>, <psmet>, <psout>, <svgmet>, <svgout>, <texmet>, <texout> or <home> to refer to PSMet, SVGMet or TexMet directories in the same FPA database.
vertical_data_file_units	string	FPA units for data values Note that all data values are automatically converted to MKS units. Either the **units** format type or the **vertical_data_file_units** keyword are required for values in the data file that are non-MKS.
vertical_data_file_format	string	Identify the format of the data in the ASCII data file with a list of recognized format types separated by the same characters used to separate the values in the ASCII data file. Recognized types: **identifier** Identifier string **latitude** Latitude location **longitude** Longitude location **timestamp** Valid time **label** Data string **value** Data value **units** FPA units of data value **-** Placeholder in data file Note that either the **identifier** or the **latitude** and **longitude** must be present to identify the data locations, but all other format types are optional. Note that only the **value** format type can be used to set the vertical positions in a cross section.

For box display ...

box_width	real	Width of boxes (in display units)
x_box_off	real	Horizontal offset of boxes (in display units)
y_box_off	real	Vertical offset of boxes (in display units)

<table>
<tr><th>PRESENTATION KEYWORD</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td><b>pattern</b></td><td>string</td><td>Line pattern for boundary (see @presentation)</td></tr>
<tr><td><b>pattern_width</b></td><td>real</td><td>Height of pattern (in display units)</td></tr>
<tr><td><b>pattern_length</b></td><td>real</td><td>Length of pattern before repeating (in display units)</td></tr>
<tr><td colspan="3">(Note that <b>line_width</b>, <b>line_style</b>, <b>outline</b> and <b>fill</b> can be specified more than once in this directive, for drawing multi-coloured curves!)</td></tr>
<tr><td><b>line_width</b></td><td>real</td><td>Line width for boundary (in display units)</td></tr>
<tr><td><b>line_style</b></td><td>string</td><td>Line style (see @presentation)</td></tr>
<tr><td><b>outline</b></td><td>string</td><td>Line outline colour (see @presentation)</td></tr>
<tr><td><b>fill</b></td><td>string</td><td>Line fill colour (see @presentation)</td></tr>
</table>

Example 2.20 Example vertical_data_file_format

The **vertical_data_file_format** string is a list of recognized format types separated by the same characters used to separate the values in the ASCII data file.

The following format would indicate that the identifier string is located in the first column of the ASCII data file, the time stamp in the second column, a data file value in the fourth column, and that the columns are separated by white space (blanks or tabs).

```
vertical_data_file_format = identifier timestamp - value
```

The following format would indicate that the latitude and longitude are located in the first and second columns of the ASCII data file, a data file value in the fifth column, the FPA units for the value in the fourth column, and that the columns are separated by semi-colons.

```
vertical_data_file_format = latitude;longitude;-;units;value
```

Example 2.21

```
@cross_section_curves
{
  cross_section_name = route_cross_section;
  vertical_element   = temperature;
  vertical_level     = surface;
  vertical_units     = degreesC;
  location_look_up   = route_temps;
  vertical_look_up   = xsect_temps;
  pattern            = simple;
  outline            = X11 Red;
  line_width         = 0.025;
  line_style         = 2 8 3;
}
```

Display a curve on a cross section named route_cross_section (set in the @define_cross_section directive). The curve is created by sampling the surface temperature field in units of degreesC at each location in the location look up table `route_temps.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive), and the values are converted to vertical locations using the vertical look up table `xsect_temps.ztab`. The curve will be drawn with a line 0.025 "units" wide and with a colour of X11 Red, using a simple line with a dash pattern of 8 times the line width followed by a space of 3 times the line width. (Note that "units" is set by @display_units.)

Example 2.22 Sample from a data file

```
@cross_section_curves
{
  cross_section_name        = route_cross_section;
  location_look_up          = route_temps;
  vertical_look_up          = xsect_temps;
  vertical_data_file        = temp_file.txt;
  vertical_data_file_format = - latitude longitude timestamp value units;
  vertical_units            = degreesC;
  pattern                   = simple;
  outline                   = X11 Red;
}
```

Display a curve on a cross section named route_cross_section (set in the @define_cross_section directive). The curve is created by sampling the data file `temp_file.txt` at the locations in the location look up table `route_temps.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive), and the values are converted to vertical locations using the vertical look up table `xsect_temps.ztab`. The temperature value is located in the fifth column of the data file and the units of temperature in data file is located in the sixth column. The value will be displayed in degreesC (vertical_units).

2.5.13 @define_cross_section

Set the size and location of a cross section. Cross sections in space, in time, and in space and time are allowed.

- A **`space`** type cross section samples through space at a set time.
- A **`time`** type cross section samples through time at a set location.
- A **`space_route`** or **`time_route`** type cross section samples through space and time.

The default locations and times for the horizontal axis of the cross section are defined in the location look up table.

The horizontal axis for a **`space`** or **`space_route`** type cross section is scaled with respect to the total distance in space, according to a map scale given by **`map_scale`**, or using an absolute width given by **`width`**.

The horizontal axis for a **`time`** or **`time_route`** type cross section is scaled with respect to the total time, using an absolute width given by **`width`**.

The default vertical locations and scaling for the vertical axis of the cross section are defined in the vertical look up table. The third column of the vertical look up table is entered as a percentage of an absolute height given by **`height`**.

KEYWORD	TYPE	DESCRIPTION
`cross_section_name`	string	Cross section name (from @define_cross_section)
`type`	string	Type of cross section One of: **`space`** constant time **`time`** constant location **`space_route`** scale wrt space **`time_route`** scale wrt time
`map_scale`	real	Scaling factor (10000000 corresponds to 1:10,000,000) applied to the horizontal axis of the cross section
`width`	real	Width (in display units) (horizontal axis)
`height`	real	Height (in display units) (vertical axis)
`location_look_up`	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional)
`vertical_look_up`	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional)

KEYWORD	TYPE	DESCRIPTION
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)

Example 2.23

```
@define_cross_section
{
  cross_section_name = route_cross_section;
  type               = time_route;
  map_scale          = 10000000;
  height             = 5.00;
  location_look_up   = route;
  vertical_look_up   = xsect;
}
```

Set the location of a cross section called **route_cross_section** to the current anchor location. The cross section will sample data through space and time. The horizontal axis locations are defined in the location look up table route.ltab and the horizontal axis will be scaled at 1:10,000,000 of the total distance covered by the location look up table. The vertical axis locations are defined in the vertical look up table xsect.ztab and the vertical axis will be scaled to 5.00 **units** (Note that **units** is set by @display_units.)

2.5.14 @define_line

Set parameters for a line to be displayed.

KEYWORD	TYPE	DESCRIPTION
`line_name`	string	Line name
`line`	r r ...	Locations (as x/y pairs) for drawing the line (in display units). Requires a minimum of 2 x/y pairs.

Example 2.24

```
@define_line { line_name = distance_scale; line = -0.985 0 0.985 0; }
```

Set parameters for a horizontal line called "distance_scale" to be drawn from a position 0.985 "units" to the left of the current position (-0.985 0) to a position 0.985 "units" to the right of the current position (0 0.985). The line will be drawn with the @draw_line directive. (Note that "units" is set by @display_units.)

Example 2.25

```
@define_line { line_name = vertical_ticks; line = 0.0 -0.1 0.0 0.1; }
```

Set parameters for a vertical line called "vertical_ticks" to be drawn from the a position 0.1 "units" below the current position (0.0 -0.1) to a position 0.1 "units" above the current position (0.0 0.1). The line will be drawn with the @draw_cross_section_line directive. (Note that "units" is set by @display_units.)

Example 2.26

```
@define_line { line_name = vertical_line; line = 0 0 0 0.5; }
```

Set parameters for a vertical line called "vertical_line" to be drawn from the current position (0 0) to a position 0.5 "units" higher (0 0.5). The line will be drawn with the @draw_table_line directive. (Note that "units" is set by @display_units.)

2.5.15 @define_map_placement

Set the size and location of the centre of the current map to be displayed.

Note that the map is either set to a given map scale using **map_scale**, or one of the axes is scaled to a given size using **size** and **axis_to_scale**.

Note that the current map is defined by the directives: @mapdef @projection @resolution

KEYWORD	TYPE	DESCRIPTION
map_scale	real	Scaling factor (10000000 corresponds to 1:10,000,000) applied to the current map
size	real	Length of scaled axis for the current map (in display units)
axis_to_scale	string	Axis of the current map to be scaled One of: **x_axis** scale x axis **y_axis** scale y axis **longest** scale longest axis **shortest** scale shortest axis
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)

Example 2.27

```
@define_map_placement
{
  size          = 6.5;
  axis_to_scale = x_axis;
  x_off         = 0.0;
  y_off         = 1.0;
}
```

Set the location of the centre of the current map to 1.0 "units" above the current anchor location. The map will have a size of 6.5 "units" horizontally, with all vertical dimensions scaled accordingly. (Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

2.5.16 @define_sample_grid

Define a grid of locations for sampling.

Grids can be either an array of latitude-longitude locations, or an array of map positions on the current map.

KEYWORD	TYPE	DESCRIPTION
grid_name	string	Grid name
For latlon grid ...		
lat_begin	latlon	Latitude to begin grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lat_end	latlon	Latitude to end grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lat_interval	latlon	Interval between latitudes in grid (see Format of Latitudes and Longitudes, (Section 1.3))
lon_begin	latlon	Longitude to begin grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lon_end	latlon	Longitude to end grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lon_interval	latlon	Interval between longitudes in grid (see Format of Latitudes and Longitudes, (Section 1.3))
For map grid ...		
map_x_begin	real	Position on x-axis of map to begin grid (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units** below)
map_x_end	real	Position on x-axis of map to end grid (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units** below)
map_x_interval	real	Interval between grid positions on x-axis of map (in units of **map_units** below)
map_y_begin	real	Position on y-axis of map to begin grid (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units** below)
map_y_end	real	Position on y-axis of map to end grid (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units** below)
map_y_interval	real	Interval between grid positions on y-axis of map (in units of **map_units** below)
map_units	real	Number of meters (or degrees) per map unit (Default is **map_units** from @mapdef directive)

	KEYWORD	TYPE	DESCRIPTION
For display off map ...			
	x_shift	real	Horizontal offset of each longitude or map_x location in the sample grid (in display units)
	y_shift	real	Vertical offset of each longitude or map_y location in the sample grid (in display units)

Example 2.28

```
@define_sample_grid
{
  grid_name     = grid_waves;
  lat_begin     = 22.5S;
  lat_end       = 60.0N;
  lat_interval = 5.0N;
  lon_begin     = 92.5W;
  lon_end       = 20.0W;
  lon_interval = 5.0E;
  x_shift       = 0.50;
  y_shift       = 0.30;
}
```

Define a latitude-longitude grid for sampling called "grid_waves". The grid will begin at latitude 22.5 degrees South and end at latitude 60.0 degrees North with sample positions every 5.0 degrees North. The grid will begin at longitude 92.5 degrees West and end at longitude 20.0 degrees West with sample positions every 5.0 degrees East. The sampling grid will therefore have 17 rows of latitudes by 15 columns of longitudes. If the sampled data is NOT displayed on the current map, then each latitude row will be offset by 0.30 "units" vertically, and each longitude column will be offset by 0.50 "units" horizontally. (Note that "units" is set by @display_units.)

2.5.17 @define_sample_list

Define a list of positions for sampling.

Lists can contain latitude-longitude locations, map positions on the current map, or identifiers for a location look up table.

KEYWORD	TYPE	DESCRIPTION
`list_name`	string	List name

For latlon positions ...
Note that **`lat_lon_ident`** can be specified more than once in this directive.

`lat_lon_ident`	l l [s]	Latitude, longitude and optional label for sample location (see Format of Latitudes and Longitudes, (Section 1.3))

For map positions ...
Note that **`map_x_y_ident`** and **`map_units`** can be specified more than once in this directive.

`map_x_y_ident`	r r [s]	Position on x-axis and y-axis of map, and optional label for sample location (Note: positions are measured from **`xmin`** and **`ymin`** from @mapdef directive, in units of **`map_units`** below)
`map_units`	real	Number of meters (or degrees) per map unit (Default is **`map_units`** from @mapdef directive)

For location idents ...
Note that **`location_ident`** and **`location_ident_list`** can be specified more than once in this directive.

`location_ident`	string	Identifier for location in **`location_look_up`** (Note: the location look_up table will be defined in a sampling directive) (Note: the magic parameters **`GPGEN_All`** and **`GPGEN_All_vtime`** can be used to display ALL locations in a location look up table
`location_ident_list`	string(s)	List of identifiers in a location look up table (Note: the location look_up table will be defined in a sampling directive)

For display off map ...

`x_shift`	real	Horizontal offset of each sample location (in display units)
`y_shift`	real	Vertical offset of each sample location (in display units)

KEYWORD	TYPE	DESCRIPTION
x_wrap	integer	Apply **x_shift** for this many values before resetting **x_shift** to 0 and applying **y_shift**. (Default is 1) Only one of **x_wrap** **y_wrap** may be set > 1.
or		
y_wrap	integer	Apply **y_shift** for this many values before resetting **y_shift** to 0 and applying **x_shift**. (Default is 1) Only one of **x_wrap** **y_wrap** may be set > 1.

Example 2.29

```
@define_sample_list
{
  list_name            = aerodrome_list;
  location_ident_list = CYFC CYYG CYHZ CYQY CYQX CYJT CYYY;
  location_ident_list = CYZV CYQB CYUL KBOS CJFK;
}
```

Define a list for sampling called "aerodrome_list". The list contains 12 location identifiers which will be matched to locations in a location look up table. The location look up table will be defined in a sampling directive which refers to the named list.

2.5.18 @define_table

Set the type and location for a table to be displayed.

Note that both column and row type tables are allowed. A column type table arranges the table data by columns, and each call to the @table_site directive will create another row of data in the table. A row type table arranges the table data by rows, and each call to the @table_site directive will create another column of data in the table.

KEYWORD	TYPE	DESCRIPTION
table_name	string	Table name
type	string	One of: **column** or **row**
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)

Example 2.30

```
@define_table
{
  table_name = T1;
  type       = column;
  x_off      =  0.00;
  y_off      = -0.75;
}
```

Set the location of a table called "T1" to 0.75 "units" below the current "anchor position". The table will arrange data by column, so that one row of data will be created for each call to the @table_site directive. (Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

2.5.19 @display_units

Set the units for sizes and distances for all maps.

Note that the default is "inches" if this directive is not used!

KEYWORD	TYPE	DESCRIPTION
`type`	string	One of: **`inches`** **`cm`** **`mm`** **`picas`** **`points`**
`scale_factor`	string	Amount to scale entire display (in percent)

Example 2.31

```
@display_units { type = inches; }
```

Sets units for all subsequent directives to inches.

Example 2.32

```
@display_units { type = inches; scale_factor = 65; }
```

Sets units for all subsequent directives to inches, but scale all displayed maps, features and labels to 65% of their defined size.

2.5.20 @distance_scale_labels

Display labels on a named distance scale.

KEYWORD	TYPE	DESCRIPTION
scale_name	string	Scale name (from @draw_distance_scale)
label_units	string	FPA units (default is **scale_units** from @draw_distance_scale)

Note that **label_location** and **label_string** may be specified more than once in this directive.

label_location	real	set the location of labels on distance scale in **label_units**
label_string	string	Text for label
label_justification	string	Label justification One of:**left centre right**
label_rotation	real	Angle of rotation of label (in counter-clockwise decimal degrees)
text_size	real	Text height for label or text found in look up table (in display units)
x_off	real	Horizontal offset of label with respect to the distance scale (in label_units)
y_off	real	Vertical offset of label with respect to the distance scale (in label_units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)

Example 2.33

```
@distance_scale_labels
{
  scale_name          = distance_scale1;
  label_location      = 250;
  label_string        = 250km;
  label_units         = km;
  label_justification = centre;
  text_size           =  0.15;
  outline             = none;
  fill                = X11 Black;
  x_off               =  0.00;
  y_off               = -0.20;
}
```

Display the label "250km" in black. It will be centred 250 units along the scale and offset 0.20 units below it.

2.5.21 @distance_scale_ticks

Display tick marks on a named distance scale.

KEYWORD	TYPE	DESCRIPTION
scale_name	string	Scale name (from @draw_distance_scale)
tick_units	string	FPA units (default is **scale_units** from @draw_distance_scale)
tick_length	real	length of tick in **tick_units**
Note that **tick_location** can be specified more than once in this directive!		
tick_location	real	distance along the scale to place tick line in **tick_units**
tick_rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
tick_justification	string	Tick justification One of:**left centre right**

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)

Example 2.34

```
@distance_scale_ticks
{
scale_name          = distance_scale1;
tick_location       = 100;
tick_location       = 200;
tick_location       = 300;
tick_location       = 400;
tick_length         = 15;
tick_units          = km;
tick_justification  = left;
tick_rotation       = 90.0;
outline             = X11 Black;
line_width          = 0.03;
}
```

Add tick lines 100km, 200km, 300km and 400km along the scale. The tick length is 15km relative to the length of the scale. They are left justified on the scale and rotated to be perpendicular to the scale.

2.5.22 @draw_cross_section_line

Display an axis, tick marks, or horizontal or vertical lines in a cross section.

An axis parameter will display either the lower or upper horizontal axis using the location look up table **location_look_up**, or the left or right vertical axis using the vertical look up table **vertical_look_up**.

A **ticks** parameter will display tick marks along the lower or upper horizontal axis, displaying one tick mark for each location in the location look up table **location_look_up**, or along the left or right vertical axis, displaying one tick mark for each location in the vertical look up table **vertical_look_up**. The tick marks are drawn with a line set in the @define_line directive.

A **horizontal_lines** parameter will display a horizontal line for each location in the vertical look up table "vertical_look_up", while a vertical_lines parameter will display vertical lines for each location in the location look up table **location_look_up**.

Note that entries in **location_look_up** table MUST be entries from the location look up table in directive @define_cross_section.

	KEYWORD	TYPE	DESCRIPTION
	cross_section_name	string	Cross section name (from @define_cross_section)
	line_to_draw	string	Parameter for cross section line to draw One of: **axis ticks horizontal_lines vertical_lines**
	axis_for_display	string	Axis of cross section to display One of: **lower upper left right**
	location_look_up	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional) (Default is **location_look_up** from @define_cross_section directive)
or			
	location_distances	real(s)	A list of distances from look up table in @define_cross_section to sample at.
	location_units	string	FPA units of **location_distances**
or			
	location_times	real(s)	A list of times from look up table in @define_cross_section to sample at.
or			
	location_fractions	real(s)	A list (0-1) of fractions along the horizontal axis from look up table in @define_cross_section to sample at.

KEYWORD	TYPE	DESCRIPTION
vertical_look_up	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional) (Default is **vertical_look_up** from @define_cross_section directive)
line_name	string	Line name (from @define_line)
arrow_name	string	Arrow display name (from @arrow_display)
x_off	real	Horizontal offset wrt cross section (in display units)
y_off	real	Vertical offset wrt cross section (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)

Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once in this directive, for drawing multi-coloured lines!

line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.35

```
@draw_cross_section_line
{
  cross_section_name = route_cross_section;
  line_to_draw       = axis;
  axis_for_display   = lower;
}
```

Display the lower axis of a cross section called "route_cross_section" (using the default location look up table from the @define_cross_section directive)

Example 2.36

```
@draw_cross_section_line
{
  cross_section_name = route_cross_section;
  line_to_draw       = ticks;
  axis_for_display   = lower;
  location_look_up   = route_sub;
  line_name          = vertical_ticks;
}
```

Display tick marks along the lower axis of a cross section called "route_cross_section". The tick marks will use a line called **`vertical_ticks`** (from the @define_line directive), displayed once for each location in the location look up table `route_sub.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive).

Example 2.37

```
@draw_cross_section_line
{
  cross_section_name = route_cross_section;
  line_to_draw       = vertical_lines;
  location_look_up   = route_sub;
}
```

Display vertical lines across a cross section called "route_cross_section". One vertical line will be displayed for each location in the location look up table `route_sub.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive).

2.5.23 @draw_distance_scale

Display a distance scale.

KEYWORD	TYPE	DESCRIPTION
scale_name	string	Scale name
scale_length	real	Length in **scale_units**
scale_units	real	FPA units of the scale
scale_justification	string	One of: **left centre right**
scale_rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)

Example 2.38

```
@draw_distance_scale
{
scale_name          = distance_scale1;
scale_length        = 500;
scale_units         = km;
scale_justification = centre;
scale_rotation      = 0.0;
x_off               = 1.4;
y_off               = -0.35;
outline             = X11 Black;
line_width          = 0.03;
}
```

Draw a horizontal 500km scale.

2.5.24 @draw_line

Display a line at a given location.

Note that the line parameters must be set in the @define_line directive.

KEYWORD	TYPE	DESCRIPTION
line_name	string	Line name (from @define_line)
arrow_name	string	Arrow display name (from @arrow_display)
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)

Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once in this directive, for drawing multi-coloured lines!

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.39

```
@draw_line
{
  line_name   = distance_scale;
  x_off       = 1.40;
  y_off       = 0.35;
  outline     = CMYK 0 0 0 100;
  line_width = 0.030;
  pattern     = simple;
}
```

Display a line called "distance_scale" (from the @define_line directive), offset 1.40 "units" to the right and 0.35 "units" above the current "anchor position". The line will be 0.030 "units" wide in a colour composed of 100% Black, and will be drawn with a simple pattern. (Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

2.5.25 @draw_table_line

Display a line in a table. The line is displayed once for each table site (set in the @table_site directive).

Note that the line parameters must be set in the @define_line directive.

KEYWORD	TYPE	DESCRIPTION
table_name	string	Table name (from @define_table)
last_site	yes/no	Draw line for last site in table?
line_name	string	Line name (from @define_line)
arrow_name	string	Arrow display name (from @arrow_display)
x_off	real	Horizontal offset wrt table location (in display units)
y_off	real	Vertical offset wrt table location (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)

Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once in this directive, for drawing multi-coloured lines!

line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.40

```
@draw_table_line
{
  line_name  = vertical_line;
  table_name = T1;
  last_site  = yes;
  x_off      = 1.90;
  y_off      = 0.00;
  outline    = CMYK 0 0 0 100;
  line_width = 0.03;
  pattern    = simple;
}
```

Display a line called "vertical_line" (from the @define_line directive) for a table called "T1" (from the @define_table, (Section 2.5.18) directive). The line will be displayed once for every site in the table (from the @table_site directive). The starting position of the line will be 1.90 "units" to the right of each site in the table. The line will be 0.03 "units" wide in a colour composed of 100% Black, and will be drawn with a simple pattern. (Note that "units" is set by @display_units.)

2.5.26 @ellipse

Display a centred ellipse on the current map or in a table or at a given location.

Note that a "pie-shaped" ellipse is drawn if **start_angle** and **end_angle** are not the same. Also note that the ellipse is drawn counter-clockwise from **start_angle** to **end_angle** before any **rotation** is applied.

KEYWORD	TYPE	DESCRIPTION
width	real	Width of ellipse (or length along x-axis) (in display units)
height	real	Height of ellipse (or length along y-axis) (in display units)
radius	real	Set width and height of ellipse to one value. (in display units)
diameter	real	Diameter for a circle (in display units)
start_angle	real	Start angle for drawing ellipse (in counter-clockwise decimal degrees from x-axis)
end_angle	real	End angle for drawing ellipse (in counter-clockwise decimal degrees from x-axis)
closed	yes/no	Should the ellipse be automatically closed? (Default is yes). If no then draw arc only.
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
rotate_to_latitude	yes/no	Rotate parallel to latitude. **rotation** is appended. (Default is no).
or		
rotate_to_longitude	yes/no	Rotate parallel to longitude. **rotation** is appended. (Default is no).
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)
For map display ...		
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
map_x	real	Position on x-axis of map (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units**)
map_y	real	Position on y-axis of map (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units**)

KEYWORD	TYPE	DESCRIPTION
map_units	real	Number of meters (or degrees) per map unit (Default is **map_units** from @mapdef directive)
location_ident	string	Identifier for location in **location_look_up**
For table display ...		
table_name	string	Table name (from @define_table) (Note: the object will be displayed once for each call to the @table_site directive)
For grid display ...		
grid_name	string	Grid name (from @define_sample_grid) (Note: the object will be displayed once for each location on the grid)
For list display ...		
list_name	string	List name (from @define_sample_list) (Note: the object will be displayed once for each location in the list)
For map, table or list display ...		
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
interior_fill	string	Colour for area interior (see @presentation)
symbol_fill_name	string	Symbol fill name for area interior One of: **none** or a symbol fill name (set in @symbol_fill_display)

Example 2.41

```
@ellipse
{
  width         = 1.0;
  height        = 1.5;
  rotation      = 45.0;
  x_off         = 2.25;
  y_off         = -1.75;
  outline       = CMYK 0 0 0 100;
  interior_fill = CMYK 0 0 0 0;
  line_width    = .050;
}
```

Display a ellipse 1.0 "units" wide by 1.5 "units" high centred at a location offset 2.25 "units" to the right and 1.75 "units" below the current "anchor position". The ellipse will be rotated 45.0 degrees counter-clockwise about the centre. The ellipse outline will be 0.050 "units" wide in a colour com- posed of 100% Black, and the ellipse will be filled with White. (Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

Example 2.42

```
@ellipse
{
  width         = 3.0;
  height        = 2.0;
  start_angle   = 270.0;
  end_angle     = 30.0;
  rotation      = 45.0;
  x_off         = 4.75;
  y_off         = -1.75;
  outline       = CMYK 0 0 0 100;
  interior_fill = CMYK 0 0 30 0;
  line_width    = .050;
}
```

Display a partial or "pie-shaped" ellipse 3.0 "units" wide by 2.0 "units" high centred at a location off- set 4.75 "units" to the right and 1.75 "units" below the current "anchor position". Only the portion of the ellipse between 270.0 degrees (measured counter-clockwise from the x-axis) and 30.0 degrees (measured counter-clockwise from the x-axis) will be drawn, and the partial ellipse will then be rotated 45.0 degrees counter-clockwise about the centre. The ellipse outline will be 0.050 "units" wide in a colour composed of 100% Black, and the ellipse will be filled with a colour composed of 30% Yellow. (Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

2.5.27 @file_close

Close the output graphics file.

Note that this directive has no keywords.

Example 2.43

```
@file_close { }
```

Close the PSMet or SVGMet output graphics file.

2.5.28 @file_name

Set the directory and path name of the output graphics file.

Note that the current output graphics file is closed and a new output graphics file is opened each time this directive is called. Note that the location of the PSMet "output_directory" is defined by the "psout" keyword in the local setup file, and the location of the SVGMet "output_directory" is defined by the "svgout" keyword in the local setup file.

KEYWORD	TYPE	DESCRIPTION
`dir`	string	Directory name for output graphics file (The directory name can include one of the codewords found in Table 2.1)
`name`	string	File name for output graphics file (The file name can include any of the codewords based on the name of the graphics product, the **T0** time when the product is run, the current "valid time" (**Tv**), or the actual "clock time" (**Tc**) for sampling. See Table 2.2)

Table 2.1: Directory Codewords

`<default>`	for path to sub-directory in the default output directory
`<psmet>`	for PSMet directory
`<psout>`	for PSMet output directory
`<svgmet>`	for SVGMet directory
`<svgout>`	for SVGMet output directory
`<texmet>`	for TexMet directory
`<texout>`	for TexMet output directory
`<home>`	for "home" directory defined in local setup file

Table 2.2: Filename Codewords

`<pdf>`	for the ***`product_name`*** of the controlling product definition file (without trailing ".fpdf")
`<iteration>`	for a @loop_begin iteration value.
`<iteration _attribute>`	for the value of the attribute in a @loop_begin iteration.
`<year>`	for **T0** year (such as 2008)
`<month>`	for **T0** month (01-12)
`<day>`	for **T0** day (01-31)
`<julian>`	for **T0** day of the year (001-366)
`<hour>`	for **T0** hour of the day (00-23)
`<minute>`	for **T0** minute in hour (00-59)
`<local>`	L if **T0** time is a local time
`<v_year>`	for **Tv** year (such as 2008)
`<v_month>`	for **Tv** month (01-12)
`<v_day>`	for **Tv** day (01-31)
`<v_julian>`	for **Tv** day of the year (001-366)
`<v_hour>`	for **Tv** hour of the day (00-23)
`<v_minute>`	for **Tv** minute in hour (00-59)
`<v_local>`	L if **Tv** time is a local time
`<p_hour>`	for **Tv-T0** in hours
`<p_minute>`	for **Tv-T0** in minutes
`<p_hr_min>`	for **Tv-T0** in hours and minutes (xx:tt)
`<c_year>`	for **Tc** year (such as 2008)
`<c_month>`	for **Tc** month (01-12)
`<c_day>`	for **Tc** day (01-31)
`<c_julian>`	for **Tc** day of the year (001-366)
`<c_hour>`	for **Tc** hour of the day (00-23)
`<c_minute>`	for **Tc** minute in hour (00-59)

Example 2.44

```
@file_name { dir = <default>; name = <pdf>;}
```

Sets the file name for the output graphics file in PSMet to: `output_directory/sub_directory/product_name.ps` where ***`output_directory`*** is identified by the directory tag "psout" in the local setup file, ***`sub_directory`*** is the PSMet directory name for this product, and ***`product_name`*** is base name of the controlling product definition file for this product. (Note that this example is the default file name for the output PostScript file, if this directive is not included!)

2.5.29 @filter

Set a minimum distance between points for drawing lines or outlines, to allow for editing within graphics post processor.

Note that it is important to reset the filter value to 0 when the required lines or outlines have been drawn.

Note that geographical outlines should be drawn with a filter value of 0.

VALUE		DESCRIPTION
filter_value	(real)	Minimum allowed distance between consecutive points drawn in a line or outline (in display units) (Note: a value of "0" will draw all points)

Example 2.45

```
@filter { 0.1 }
```

Sets the minimum allowed distance between consecutive points in a line to 0.1 "units". Points closer than this distance will be omitted. (Note that "units" is set by @display_units.)

Example 2.46

```
@filter { 0 }
```

Reset the filter value to display ALL points in a line or outline.

2.5.30 @geo_presentation

Set the appearance of geographic features displayed with the @geography directive.

Note that each geographic feature displayed is checked against the list of geographic presentations defined by each **geo_name** keyword. Note that each subsequent @geo_presentation directive will add an additional geographic presentation to the list, or modify existing presentations for matching names.

Note that the @reset_geo_presentation directive resets the list of geographic presentations currently used.

Note that patterned lines are not allowed for displaying geographic features.

KEYWORD	TYPE	DESCRIPTION
geo_name	string	Name for a geographic presentation

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
interior_fill	string	Colour for area interior (see @presentation)
symbol_fill_name	string	Symbol fill name for area interior One of: **none** or a symbol fill name (set in @symbol_fill_display)

Example 2.47

```
@geo_presentation
{
  geo_name      = land;
  outline       = none;
  interior_fill = CMYK 0 10 35 0;
}
```

Display geographic features named land without an outline but filled with a colour composed of 10% Magenta and 35% Yellow.

Example 2.48

```
@geo_presentation
{
  geo_name      = border_major;
  line_width    = 0.001;
  outline       = CMYK 0 0 0 80;
  interior_fill = none;
}
```

Display geographic features named border_major with an outline 0.001 "units" wide in a colour composed of 80% Black. (Note that "units" is set by @display_units.)

2.5.31 @geography

Display geographic features on the current map.

Note that the appearance of geographic features is given by matching the value of the "attribute" keyword for each feature with the named presentations set in the @geo_presentation directive.

Note that the **category_attribute** and **category** keywords can be used one or more times to display only selected geographic features.

KEYWORD	TYPE	DESCRIPTION
geo_file	string	FPA Metafile containing geography (Note: this must be the name of a file in the "Maps" directory or a full path name)
element	string	FPA element name
level	string	FPA level name

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with **geo_name** from @geo_presentation directive to select a presentation (default is **FPA_category**)

Example 2.49

```
@geography
{
  geo_file           = base_map.fpa;
  element            = land_water;
  level              = geography;
  category_attribute = FPA_category;
  category           = land;
  attribute          = FPA_category;
}
```

Display geographic features from the geography **land_water** fields in the file `base_map.fpa` in the "Maps" directory. Only the features with an attribute value for **FPA_category** of land will be chosen. The features will be displayed by matching the attribute value for **FPA_category** to the named presentations previously set by the @geo_presentation directive.

Example 2.50

```
@geography
{
  geo_file           = base_map.fpa;
  element            = boundaries;
  level              = geography;
  category_attribute = FPA_category;
  category           = border border_major;
  attribute          = FPA_category;
}
```

Display geographic features from the geography **boundaries** fields in the file `base_map.fpa` in the "Maps" directory. Only the features with an attribute value for **FPA_category** of border or border_major will be chosen. The features will be displayed by matching the attribute value for **FPA_category** to the named presentations previously set by the @geo_presentation directive.

2.5.32 @gpgen_group_begin

Force grouping of objects for an output graphics file.

Note that this directive is not used by PSMet!

This directive does not require keywords. You may include format specific keywords that will be added to the grouping tag of the output format.

Example 2.51 SVG keywords

```
@gpgen_group_begin { <keyGroup; visibility = 'visible'; }
```

Groups all subsequent graphic displays up to next @gpgen_group_end directive, and assigns the 'keyword = value' pairs from a group directive (defined in a @loop_begin or @group directives) and 'visibility' value to the grouped elements.
keywords in <keyGroup; and 'visibility' will not be valid GPGEN keywords, but they should be valid SVG keywords.

2.5.33 @gpgen_group_end

End forced grouping of objects for an output graphics file.

Note that this directive is not used by PSMet!

Note that this directive has no keywords.

Example 2.52

```
@gpgen_group_end { }
```

Ends grouping of graphic displays that began with @gpgen_group_begin directive.

2.5.34 @gpgen_insert

Insert format specific commands directly into the output file. This directive can interpret an number of special codewords that are replaced by appropriate values when the product is generated.

VALUE		DESCRIPTION
commands	string	valid output format specific code. (The command can include any of the codewords based on the name of the graphics product, the **T0** time when the product is run, the current "valid time" (**Tv**), or the actual "clock time" (**Tc**) for sampling. See Table 2.2)

Table 2.3: Codewords

<GPGEN_default>	path to ***sub_directory*** in the ***output_directory***
<GPGEN_svgout>	for SVGMet output directory

Table 2.3: (continued)

<GPGEN_home>	for "home" directory defined in local setup file
<GPGEN_year>	for **T0** year (such as 2008)
<GPGEN_month>	for **T0** month (01-12)
<GPGEN_day>	for **T0** day (01-31)
<GPGEN_julian>	for **T0** day of the year (001-366)
<GPGEN_hour>	for **T0** hour of the day (00-23)
<GPGEN_minute>	for **T0** minute in hour (00-59)
<GPGEN_local>	L if **T0** time is a local time
<GPGEN_v_year>	for **Tv** year (such as 2008)
<GPGEN_v_month>	for **Tv** month (01-12)
<GPGEN_v_day>	for **Tv** day (01-31)
<GPGEN_v_julian>	for **Tv** day of the year (001-366)
<GPGEN_v_hour>	for **Tv** hour of the day (00-23)
<GPGEN_v_minute>	for **Tv** minute in hour (00-59)
<GPGEN_v_local>	L if **Tv** time is a local time
<GPGEN_c_year>	for **Tc** year (such as 2008)
<GPGEN_c_month>	for **Tc** month (01-12)
<GPGEN_c_day>	for **Tc** day (01-31)
<GPGEN_c_julian>	for **Tc** day of the year (001-366)
<GPGEN_c_hour>	for **Tc** hour of the day (00-23)
<GPGEN_c_minute>	for **Tc** minute in hour (00-59)
<GPGEN_p_hour>	for **Tv-T0** in hours
<GPGEN_p_minute>	for **Tv-T0** in minutes
<GPGEN_p_hr_min>	for **Tv-T0** in hours and minutes (xx:tt)
<GPGEN_iteration>	for a @loop_begin iteration value.
<GPGEN_iteration_attribute>	for the value of the attribute in a @loop_begin iteration.

Example 2.53 SVG Product

```
@gpgen_insert { <use xlink:href="map.jpg" x="0" y="0"> }
```

Insert a JPEG background map into the SVG image.

2.5.35 @group

Set a list of keywords to be included as a "macro".

Note that the keywords are included in another directive by using the format: ***<group_name*** in the keyword list.

KEYWORD	TYPE	DESCRIPTION
group_name	string	Group name for keyword list
acceptable_keywords		Any keywords that are acceptable in the directives that this "macro" is included within can be set here

Example 2.54

```
@group
{
  group_name        = MaxTempAreas;
  element           = max_temp;
  level             = surface;
  units             = degreesC;
  display_as_areas = yes;
  outline           = none;
  fill              = none;
}
```

Set a "macro" called "MaxTempAreas" containing keywords used in the @contours directive, given in the example above.

2.5.36 @images

Insert imagery into output products.

KEYWORD	TYPE	DESCRIPTION
image_tag	string	Image file identifier from Image configuration file
image_tag_list	string(s)	List of image file identifiers from Image configuration file
colour_table	string	Name of colour table from Image configuration file
colour_table_list	string(s)	List of colour tables from Image configuration file
brightness	integer	Set the default brightness for corresponding image files. Values between 0 and 100.
brightness_list	integer(s)	List of brightness values for corresponding image files. Values between 0 and 100.
satellite_brightness	integer	Default brightness for all satellite images. Values between 0 and 100.
radar_brightness	integer	Default brightness for all radar images. Values between 0 and 100.
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)

Imagery may not always be available at fixed intervals. Sometimes it may be a few minutes early or late. The keywords: **match_time_before** and **match_time_after** allow you to define an acceptance window. The keywords **satellite_time_before**, **satellite_time_after**, **radar_time_before** and **radar_time_after** allow you to set different acceptance windows for satellite and radar.

KEYWORD	TYPE	DESCRIPTION
match_time_before	vtime	Lower limit of acceptance window (specify hours and minutes as in: ***hh:mm***)
match_time_after	vtime	Upper limit of acceptance window (specify hours and minutes as in: ***hh:mm***)
satellite_time_before	vtime	Lower limit of acceptance window for satellite images (specify hours and minutes as in: ***hh:mm***)
satellite_time_after	vtime	Upper limit of acceptance window for satellite images (specify hours and minutes as in: ***hh:mm***)
radar_time_before	vtime	Lower limit of acceptance window for radar images (specify hours and minutes as in: ***hh:mm***)
radar_time_after	vtime	Upper limit of acceptance window for radar images (specify hours and minutes as in: ***hh:mm***)

KEYWORD	TYPE	DESCRIPTION
blend_images	yes/no	Make radar imagery semi-transparent when displayed over satellite imagery
blend_ratio	integer	Degree of transparency
radar_range_rings	yes/no	Display radar range rings? (Note: If range rings are already part of the image they can not be removed)
radar_range_ring_interval	real	If the range rings are not already part of the image, sets the distance between range rings in **radar_range_ring_units**
radar_range_ring_units	string	FPA units for range ring interval
radar_range_ring_colour	string	Radar range ring colour One of: **none** **CMYK *c m y k*** **RGB *r g b*** **X11 *name*** where ***c,m,y,k*** and ***r,g,b*** are 0-100, and ***name*** is a valid X11 colour name (Note: If range rings are already part of the image they can not be changed)
radar_limit_ring	yes/no	Display radar limit ring? (Note: If range rings are already part of the image they can not be removed)
radar_limit_ring_colour	string	Radar limit ring colour One of: **none** **CMYK *c m y k*** **RGB *r g b*** **X11 *name*** where ***c,m,y,k*** and ***r,g,b*** are 0-100, and ***name*** is a valid X11 colour name (Note: If range rings are already part of the image they can not be changed)

Example 2.55

```
@images
{
  brightness              = 90;
  image_tag_list          = radar1 radar2 radar3 satellite1;
  match_time_before       = 00:15;
  match_time_after        = 00:20;
  radar_limit_ring        = yes;
  radar_limit_ring_colour = X11 Red;
  blend_images            = yes;
  blend_ratio             = 30;
}
```

Display three images blended with radar at 30% transparency. Match images up to 15 minutes early and 20 minutes late. Include a red radar limit ring.

2.5.37 @include

Process directives from another product definition file.

Note that the path for the product definition file should be relative to the directory containing the current product definition file.

Note that the ".fpdf" suffix on the file name is optional.

VALUE		DESCRIPTION
file_name	(string)	Path for the product definition file

Example 2.56

```
@include { av_map_setup2 }
```

Process the directives from the file: `av_map_setup2.fpdf`.
Note that this product definition file will be located in the same directory as the product definition file containing the @include directive

2.5.38 @label

Display location dependent information on the current map or in a table or at a given location. This information includes features labeled in the FPA, or parameters from locations in a "scattered" type field.

Complex (multi-attribute) labels are described at the end of this section.

Note that the **category_attribute** and **category** keywords can be used to display only selected labels.

Note that the **case** and **case_look_up** keywords can only be used for simple (single-attribute) labels.

Note that every label requires a **format** keyword. A style of **none** will display only a mark or label_display. A **direct** style format will display the value of the label attribute from FPA. A **symbol** or **text** style format also requires a **look_up** keyword to define a look up table, and the value of the label attribute from FPA is cross-referenced in this look up table to identify the symbol or text string to display. A **wind_barb**, **wind_text** or **wind_symbol** format displays winds using the presentation from the directives @wind_barb_presentation and @wind_presentation.

KEYWORD	TYPE	DESCRIPTION
element	string	FPA element name
level	string	FPA level name
units	string	FPA units for label wind speeds or gusts
geo_file	string	FPA Metafile for geography labels (Note: this must be the name of a file in the "Maps" directory or a full path name)

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)

Note that **case** and **case_look_up** can be specified more than once in this directive!

KEYWORD	TYPE	DESCRIPTION
case	string	Time of day or hemisphere in which to use **case_look_up** look up table (given below) One of:**day night dusk** **northern_hemisphere** **southern_hemisphere**
case_look_up	string	File name for the optional look up table (Note: ".tab" suffix is optional)
time_zone	string	Display the time (from a magic attribute using **strftime** format) adjusted to another time zone (see @write_time for more information)
language	string	Display the time (from a magic attribute using **strftime** format) adjusted to another language (see @write_time for more information)
format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **symbol** ***<cform>*** use look up **text** ***<cform>*** use look up **wind_barb** for wind values **wind_text** for wind values **wind_symbol** for wind values **none** do not display value (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)
mark	string	Name of graphic file to display as a marker (Note that the full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)
mark_scale	real	Scale factor wrt original mark (in percent)
symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent)
text_size	real	Text height for label or text found in look up table (in display units)
display_name	string	Named display (from @label_display)

KEYWORD	TYPE	DESCRIPTION
`display_type`	string	Type of display about label One of: **`sized_box`** scaled box **`fixed_box`** fixed size box **`sized_ellipse`** scaled ellipse **`fixed_ellipse`** fixed size ellipse **`sized_underline`** scaled underline **`fixed_underline`** fixed size line (Note: the ***`sized_...`*** display types scale the type of display to the size of the label)
`width_scale`	real	Approximate width of characters, for use with ***`sized_...`*** display types (as percent of **`text_size`** keyword value) The default value of **`60`** is acceptable for most font types
`height_scale`	real	Approximate height of characters, for use with ***`sized_...`*** display types (as percent of **`text_size`** keyword value) The default value of **`100`** is acceptable for most font types
`fit_to_map`	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)
`fit_to_map_ref`	string	Reference position to fit label to current map One of: **`none`** **`upper`** **`lower`** **`left`** **`right`** (default is **`none`**) (Note: this parameter usually indicates the position of the label **`mark`** wrt the rest of the label)
`rotation`	real	Angle of rotation (in counter-clockwise decimal degrees)
`rotate_to_latitude`	yes/no	Rotate parallel to latitude. **`rotation`** is appended. (Default is no).
or		
`rotate_to_longitude`	yes/no	Rotate parallel to longitude. **`rotation`** is appended. (Default is no).
or		
`rotation_attribute`	string	Set rotation based on value of attribute. **`rotation`** is appended.
`constrain_rotation`	yes/no	Used with **`rotate_to_latitude`**, **`rotate_to_longitude`**, or **`rotation_attribute`** to force label to always appear right-side up with respect to the bottom of the map. (Default is "yes")
`x_display_off`	real	Horizontal offset of entire label, including the location mark (in display units)
`y_display_off`	real	Vertical offset of entire label, including the location mark (in display units)

KEYWORD	TYPE	DESCRIPTION
x_off	real	Horizontal offset of label (in display units)
y_off	real	Vertical offset of label (in display units)
For display off map ...		
x_shift	real	Horizontal increment for display of labels off the map (in display units)
y_shift	real	Vertical increment for display of labels off the map (in display units)
x_wrap	integer	Apply **x_shift** for this many values before resetting **x_shift** to 0 and applying **y_shift**. (Default is 1) Only one of **x_wrap** **y_wrap** may be set > 1.
or		
y_wrap	integer	Apply **y_shift** for this many values before resetting **y_shift** to 0 and applying **x_shift**. (Default is 1) Only one of **x_wrap** **y_wrap** may be set > 1.

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none** **thin** **ultralight** **light** **normal** **medium** **demibold** **bold** **ultrabold** **black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left** **centre** **right**
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)

Example 2.57

```
@label
{
  element     = actual_wind;
  level       = surface;
  units       = knots;
  attribute   = EVAL_wind;
  format      = wind_barb;
  line_width = 0.04;
  outline     = X11 FireBrick;
  fill        = X11 FireBrick;
  x_off       = 0.00;
  y_off       = 0.00;
}
```

Display labels from the surface actual_wind field in the FPA as wind barbs scaled in units of knots. The wind barbs will be displayed in Firebrick with lines 0.04 "units" wide and not offset, according to the appearance given in the @wind_barb_presentation directive. (Note that "units" is set by @display_units.)

Example 2.58

```
@label
{
  element              = weather_system;
  level                = surface;
  category_attribute = wx_label_type;
  category             = public;
  attribute            = clds_and_wx;
  format               = symbol;
  look_up              = weather_day;
  case                 = night;
  case_look_up         = weather_ngt;
  symbol_scale         = 40;
  x_off                = 0.0;
  y_off                = 0.0;
  justification        = centre;
}
```

Display labels from the surface weather_system field which have a value of public for the FPA attribute wx_label_type. Labels will be displayed as symbols by matching the value from the FPA attribute clds_and_wx to the values in the look up table `weather_day.tab` (or by matching the FPA value to the values in the look up table `weather_ngt.tab` at nighttime). The symbols will be centred, not offset, and 40% of their original size.

Complex (multi-attribute) labels allow the display of several label parameters within the same directive. The entire complex label is displayed with respect to the **rotation**, **x_off**, **y_off**, **display_name**, and **display_type** keywords set above.

Attribute keywords apply only to the attribute that they follow.

The position of an attribute in a complex label can be related to the position of other attributes through use of the keywords **attribute_anchor**, **attribute_ref**, **attribute_justification** and **attribute_vertical_just**. The **attribute_ref** keyword defines the starting position of the attribute with respect to a location about the symbol or text identified by the **attribute_anchor** keyword. The **attribute_justification** and **attribute_vertical_just** keywords define the horizontal and vertical positioning of the attribute with respect to this starting position.

The display parameters of each attribute default to the values of the **look_up**, **justification**, **format**, **symbol_scale**, **text_size**, **units**, **width_scale**, and **height_scale** keywords and the **fill**, **line_style**, **line_width**, **outline**, **outline_first**, **font**, **font_weight**, **char_space**, **word_space**, and **italics** presentation keywords set above. These default values can be replaced by the appropriate attribute keyword listed below, to create multi-colour multi-format complex labels.

Note that the **attribute_show** keyword can be used to "turn off" the display of attributes, while still maintain positioning with respect to other attributes.

KEYWORD	TYPE	DESCRIPTION
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
attribute_look_up	string	File name for look up table (Note: ".tab" suffix is optional) (default is **look_up** set above)
attribute_show	yes/no	Display attribute? (default is yes)
attribute_anchor	string	Reference "anchor" for displaying attribute One of: **none** or a previously listed **attribute** (default is **none**, which corresponds to the starting position of the complex label)
attribute_ref	string	Starting position for displaying attribute (a position on the "attribute_anchor" label) One of: **lower_left centre_left upper_left lower_centre centre upper_centre lower_right centre_right upper_right** (default is **lower_centre**)
attribute_justification	string	Attribute text justification (horizontal) One of: **left centre right** (default is **justification** set above)
attribute_vertical_just	string	Attribute text justification (vertical) One of: **bottom**, **centre**, **top** (default is **bottom**)
attribute_units	string	FPA units for label wind speeds or gusts

KEYWORD	TYPE	DESCRIPTION
attribute_format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **symbol** ***<cform>*** use look up **text** ***<cform>*** use look up **wind_barb** for wind values **wind_text** for wind values **wind_symbol** for wind values **none** do not display value (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **attribute_look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format.)
attribute_symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent) (default is **symbol_scale** set above)
attribute_text_size	real	Text height for attribute or text found in look up table (in display units) (default is **text_size** set above)
attribute_display_name	string	Named display (from @label_display)
attribute_display_type	string	Type of display about attribute One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)
attribute_width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **width_scale** set above)
attribute_height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **height_scale** set above)

KEYWORD	TYPE	DESCRIPTION
attribute_x_off	real	Horizontal offset of attribute display (in display units)
attribute_y_off	real	Vertical offset of attribute display (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
attribute_line_width	real	Line width (in display units) (default is **line_width** set above)
attribute_line_style	string	Line style (see @presentation) (default is **line_style** set above)
attribute_outline	string	Line outline colour (see @presentation)
attribute_fill	string	Line fill colour (see @presentation)
attribute_outline_first	yes/no	Display outline first? (default is **outline_first** set above)
attribute_font	string	Text font name recognized by output device or graphics post processor (default is **font** set above)
attribute_font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black** (default is **font_weight** set above)
attribute_italics	yes/no	Text in italics? (default is **italics** set above)
attribute_char_space	real	Text character spacing (as percent of average character width) (default is **char_space** set above)
attribute_word_space	real	Text word spacing (as percent of average character width) (default is **word_space** set above)

Example 2.59

```
@label
{
  element                 = pressure;
  level                   = msl;
  category_attribute      = hilo_type;
  category                = high low;
  mark                    = hilo;
  mark_scale              = 10;
  symbol_scale            = 35;
  text_size               = 0.20;
  outline                 = X11 black;
  fill                    = X11 black;
  line_width              = 0.01;
  justification           = centre;

  attribute               = hilo_type;
  attribute_anchor        = none;
  attribute_ref           = lower_centre;
  attribute_vertical_just = centre;
  attribute_format        = symbol;
  attribute_look_up       = hilo;
  attribute_x_off         = 0.00;
  attribute_y_off         = 0.15;

  attribute               = EVAL_spval;
  attribute_anchor        = none;
  attribute_ref           = lower_centre;
  attribute_format        = direct %s;
  attribute_x_off         = 0.00;
  attribute_y_off         = -0.25;
  attribute_x_off         = 0.00;
  attribute_y_off         = -0.25;
  attribute_outline       = X11 white;
  attribute_line_width    = 0.03;
  attribute_outline_first = yes;
}
```

Display labels from the msl pressure field as complex labels if the value for the FPA attribute hilo_type is one of high or low. The label location will be marked by the symbol "hilo", scaled to 10% of its original size. The default size of attribute symbols will be 35% of their original size, and the default size of attribute text will be 0.20 "units". The default attribute colours will be black, the default line width will be 0.01 "units", and attributes will default to centre justification. The FPA attribute **`hilo_type`** will be displayed as a symbol, with the centre located 0.15 "units" above the label location, by matching the attribute value to the values within the look up table `hilo.tab`. The FPA attribute **`EVAL_spval`** will be displayed as a string in %s format located 0.25 "units" below the label location, outlined with a white line 0.03 "units" wide, and with the outline displayed first. (Note that the default PSMet mark would be `$FPA/setup/pdf/psmet/common/ps/hilo.ps`, and that the default SVGMet mark would be `$FPA/setup/pdf/svgmet/common/svg/hilo.svg`.) (Note that "units" is set by @display_units.)

2.5.39 @label_display

Set display parameters for drawing a box, ellipse, or underline around simple or complex labels or sampled values.

Note that the same display parameters can be used for different display types. The type of display to use is set by a **display_type** keyword in each directive.

KEYWORD	TYPE	DESCRIPTION
display_name	string	Display name
width	real	Width (in display units)
height	real	Height (in display units)
width_attribute	string	Set width based on value of attribute.
height_attribute	string	Set height based on value of attribute.
attribute_units	string	FPA units for value of **height_attribute** and **height_attribute**.
radius	real	Set width and height of fixed sized ellipse to one value. (in display units)
radius_attribute	real	Set width and height of ellipse to value of attribute. (in display units)
diameter	real	Set width and height of fixed sized ellipse to one value. (in display units)
diameter_attribute	real	Set width and height of ellipse to value of attribute. (in display units)
start_angle	real	Start angle for ellipse display (in counter-clockwise decimal degrees from x-axis)
end_angle	real	End angle for ellipse display (in count-clockwise decimal degrees from x-axis)
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
rotation_attribute	string	Set rotation based on value of attribute. **rotation** is appended.
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)
margin_left	real	Offset left edge of display leftwards (in display units)
margin_right	real	Offset right edge of display rightwards (in display units)
margin_top	real	Offset top edge of display upwards (in display units)
margin_bottom	real	Offset bottom edge of display downwards (in display units)
margin_width	real	Offset left edge of display leftwards and right edge of display rightwards (in display units)

KEYWORD	TYPE	DESCRIPTION
margin_height	real	Offset top edge of display upwards and bottom edge of display downwards (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
interior_fill	string	Colour for area interior (see @presentation)

Example 2.60

```
@label_display
{
  display_name  = label1;
  margin_left   = 0.05;
  margin_right  = 0.00;
  margin_top    = 0.00;
  margin_bottom = 0.05;
  line_width    = 0.02;
  outline       = CMYK 0 0 0 100;
  interior_fill = none;
}
```

Set parameters for a display called label1. The display outline will be 0.02 "units" wide in a colour composed of 100% Black, and "sized_..." type displays will have a left edge offset 0.05 "units" to the left of the outline, and a bottom edge offset 0.05 "units" below the outline. (Note that "units" is set by @display_units.)

Example 2.61

```
@label_display
{
  display_name  = under1;
  margin_bottom = 0.02;
  line_width    = 0.01;
  outline       = CMYK 0 0 0 100;
  interior_fill = none;
}
```

Set parameters for a display called under1. The display outline will be 0.01 "units" wide in a colour composed of 100% Black, and "sized_..." type displays will have a bottom edge offset 0.02 "units" below the outline. (Note that "units" is set by @display_units.)

2.5.40 @lchain_nodes

Display link chain node information on the current map.

Note that the **category_attribute** and **category** keywords can be used to display only selected nodes, and that the **track_category_attribute** and **track_category** keywords can be used to display only nodes from selected tracks.

Note that the **case** and **case_look_up** keywords can only be used for simple (single-attribute) labels.

Note that every node requires a **format** keyword. A style of **none** will display only a mark or label_display. A **direct** style format will display the value of the label attribute from FPA. A **symbol** or **text** style format also requires a **look_up** keyword to define a look up table, and the value of the label attribute from FPA is cross-referenced in this look up table to identify the symbol or text string to display. A **wind_barb**, **wind_text** or **wind_symbol** format displays winds using the presentation from the directives @wind_barb_presentation and @wind_presentation.

KEYWORD	TYPE	DESCRIPTION
Note that **element** and **level** can be specified more than once in this directive. Note that if a single element name is matched with a list of level names, or vice versa, the single name must be given first. Multiple element and level name lists should have the same number.		
element	string	FPA element name
element_list	string	List of FPA element names
level	string	FPA level name
level_list	string	List of FPA level names
node_speed_units	string	FPA units for motion of link chain at node.
node_speed_round	real	Precision of node speed.
node_stationary_max	real	Limit node labels based on node speed.
node_stationary_label	string	Label to use if node speed is less than **node_stationary_max**.
units	string	FPA units for node wind speeds or gusts (if required)
start_time	vtime	Nodes before this time will not be included. Default is the start of the track.
end_time	vtime	Nodes after this time will not be included. Default is the end of the track.
or		
times	vtime(s)	List of times to be included.
Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.		
category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)

KEYWORD	TYPE	DESCRIPTION
category	string	FPA category name (or list of names) for selecting which nodes to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)

Note that **track_category_attribute** and **track_category** can be specified more than once in this directive, and that **track_category_cascade** determines how to combine more than one instance.

track_category_cascade	string	How to combine more than one instance of **track_category_attribute** and **track_category** One of: **and** or **or** (default is **and**)
track_category_attribute	string	FPA attribute containing category names for tracks containing the nodes (default is **FPA_category**)
track_category	string	FPA category name (or list of names) for selecting which tracks contain the nodes to display (Note: name(s) are matched with the value sampled from **track_category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)

Note that **case** and **case_look_up** can be specified more than once in this directive!

case	string	Time of day or hemisphere in which to use **case_look_up** look up table (given below) One of:**day night dusk northern_hemisphere southern_hemisphere**
case_look_up	string	File name for the optional look up table (Note: ".tab" suffix is optional)
time_zone	string	Display the time (from a magic attribute using **strftime** format) adjusted to another time zone (see @write_time for more information)
language	string	Display the time (from a magic attribute using **strftime** format) adjusted to another language (see @write_time for more information)

KEYWORD	TYPE	DESCRIPTION
format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **symbol** ***<cform>*** use look up **text** ***<cform>*** use look up **wind_barb** for wind values **wind_text** for wind values **wind_symbol** for wind values **none** do not display value (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)
mark	string	Name of graphic file to display as a marker (Note that the full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)
mark_scale	real	Scale factor wrt original mark (in percent)
symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent)
text_size	real	Text height for label or text found in look up table (in display units)
display_name	string	Named display (from @label_display)
display_type	string	Type of display about label One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)
width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **60** is acceptable for most font types

KEYWORD	TYPE	DESCRIPTION
height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **100** is acceptable for most font types
fit_to_map	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)
fit_to_map_ref	string	Reference position to fit label to current map One of: **none upper lower left right** (default is **none**) (Note: this parameter usually indicates the position of the label **mark** wrt the rest of the label)
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
rotate_to_latitude	yes/no	Rotate parallel to latitude. **rotation** is appended. (Default is no).
or		
rotate_to_longitude	yes/no	Rotate parallel to longitude. **rotation** is appended. (Default is no).
or		
rotation_attribute	string	Set rotation based on value of attribute. **rotation** is appended.
constrain_rotation	yes/no	Used with **rotate_to_latitude**, **rotate_to_longitude**, or **rotation_attribute** to force label to always appear right-side up with respect to the bottom of the map. (Default is "yes")
x_stationary	integer	Horizontal offset for stationary labels.
y_stationary	integer	Vertical offset for stationary labels.
x_display_off	real	Horizontal offset of entire label, including the location mark (in display units)
y_display_off	real	Vertical offset of entire label, including the location mark (in display units)
x_off	real	Horizontal offset of node label (in display units)
y_off	real	Vertical offset of node label (in display units)
For display off map ...		
x_shift	real	Horizontal increment for display of nodes off the map (in display units)
y_shift	real	Vertical increment for display of nodes off the map (in display units)

	KEYWORD	TYPE	DESCRIPTION
	x_wrap	integer	Apply **x_shift** for this many values before resetting **x_shift** to 0 and applying **y_shift**. (Default is 1) Only one of **x_wrap y_wrap** may be set > 1.
or			
	y_wrap	integer	Apply **y_shift** for this many values before resetting **y_shift** to 0 and applying **x_shift**. (Default is 1) Only one of **x_wrap y_wrap** may be set > 1.

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left centre right**
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)

Example 2.62

```
@lchain_nodes
{
  element              = storm_tracks;
  level                = surface;
  end_time             = -0:10;
  mark                 = circle;
  mark_scale           = 60;
  line_width           = 0.01;
  outline              = X11 Grey90;
}
```

Display nodes for link chains with times up to 10 minutes before T0 from the storm_tracks surface field in the FPA as grey circle marks 60% original size.

Example 2.63

```
@lchain_nodes
{
  element             = storm_tracks;
  level               = surface;
  start_time          = 0:00;
  mark                = circle_filled;
  outline             = X11 Black;
}
```

Display nodes for link chains with times at or after T0 from the storm_tracks surface field in the FPA as black filled circle marks.

Example 2.64

```
@lchain_nodes
{
  element             = tropical_tracks;
  level               = surface;
  times               = -12 -6 0 6 12 18 24;
  format              = symbol;
  attribute           = cyclone_type;
  look_up             = tropical_storms;
  symbol_scale        = 15;
  fill                = X11 Blue;
  justification       = centre;
  rotate_to_latitude = yes;
}
```

Display nodes for link chains at given times from the tropical_tracks surface field in the FPA as symbols by matching the value from the FPA attribute cyclone_type to the values in the look up table `tropical_storms.tab`. The symbols will be centred, 15% of their original size, and rotated to be parallel to the latitude where they are located.

Complex (multi-attribute) labels allow the display of several label parameters within the same directive. The entire complex label is displayed with respect to the **rotation**, **x_off**, **y_off**, **display_name**, and **display_type** keywords set above.

Attribute keywords apply only to the attribute that they follow.

The position of an attribute in a complex label can be related to the position of other attributes through use of the keywords **attribute_anchor**, **attribute_ref**, **attribute_justification** and **attribute_vertical_just**. The **attribute_ref** keyword defines the starting position of the attribute with respect to a location about the symbol or text identified by the **attribute_anchor** keyword. The **attribute_justification** and **attribute_vertical_just** keywords define the horizontal and vertical positioning of the attribute with respect to this starting position.

The display parameters of each attribute default to the values of the **look_up**, **justification**, **format**, **symbol_scale**, **text_size**, **units**, **width_scale**, and **height_scale** keywords and the **fill**, **line_style**, **line_width**, **outline**, **outline_first**, **font**, **font_weight**, **char_space**, **word_space**, and **italics** presentation keywords set above. These default values can be replaced by the appropriate attribute keyword listed below, to create multi-colour multi-format complex labels.

Note that the **attribute_show** keyword can be used to "turn off" the display of attributes, while still maintain positioning with respect to other attributes.

KEYWORD	TYPE	DESCRIPTION
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
attribute_look_up	string	File name for look up table (Note: ".tab" suffix is optional) (default is **look_up** set above)
attribute_show	yes/no	Display attribute? (default is yes)
attribute_anchor	string	Reference "anchor" for displaying attribute One of: **none** or a previously listed **attribute** (default is **none**, which corresponds to the starting position of the complex label)
attribute_ref	string	Starting position for displaying attribute (a position on the "attribute_anchor" label) One of: **lower_left centre_left upper_left lower_centre centre upper_centre lower_right centre_right upper_right** (default is **lower_centre**)
attribute_justification	string	Attribute text justification (horizontal) One of: **left centre right** (default is **justification** set above)
attribute_vertical_just	string	Attribute text justification (vertical) One of: **bottom**, **centre**, **top** (default is **bottom**)
attribute_units	string	FPA units for label wind speeds or gusts

KEYWORD	TYPE	DESCRIPTION
attribute_format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **symbol** ***<cform>*** use look up **text** ***<cform>*** use look up **wind_barb** for wind values **wind_text** for wind values **wind_symbol** for wind values **none** do not display value (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **attribute_look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format.)
attribute_symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent) (default is **symbol_scale** set above)
attribute_text_size	real	Text height for attribute or text found in look up table (in display units) (default is **text_size** set above)
attribute_display_name	string	Named display (from @label_display)
attribute_display_type	string	Type of display about attribute One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)
attribute_width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **width_scale** set above)
attribute_height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **height_scale** set above)

KEYWORD	TYPE	DESCRIPTION
attribute_x_off	real	Horizontal offset of node attribute (in display units)
attribute_y_off	real	Vertical offset of node attribute (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
attribute_line_width	real	Line width (in display units) (default is **line_width** set above)
attribute_line_style	string	Line style (see @presentation) (default is **line_style** set above)
attribute_outline	string	Line outline colour (see @presentation)
attribute_fill	string	Line fill colour (see @presentation)
attribute_outline_first	yes/no	Display outline first? (default is **outline_first** set above)
attribute_font	string	Text font name recognized by output device or graphics post processor (default is **font** set above)
attribute_font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black** (default is **font_weight** set above)
attribute_italics	yes/no	Text in italics? (default is **italics** set above)
attribute_vertical_just	string	Attribute text justification (vertical) One of: **bottom**, **centre**, **top** (default is **bottom**)
attribute_char_space	real	Text character spacing (as percent of average character width) (default is **char_space** set above)
attribute_word_space	real	Text word spacing (as percent of average character width) (default is **word_space** set above)

2.5.41 @lchain_tracks

Display tracks from a "link chain" field (or from more than one "link chain" field) on the current map. Note that the **category_attribute** and **category** keywords can be used one or more times to display only selected link chains. Note that the **attribute** and **look_up** keywords can be used to choose different presentations for lines with different attributes.

KEYWORD	TYPE	DESCRIPTION
Note that **element** and **level** can be specified more than once in this directive. Note that if a single element name is matched with a list of level names, or vice versa, the single name must be given first. Multiple element and level name lists should have the same number.		
element	string	FPA element name
element_list	string	List of FPA element names
level	string	FPA level name
level_list	string	List of FPA level names
start_time	vtime	Nodes before this time will not be included. Default is the start of the track.
end_time	vtime	Nodes after this time will not be included. Default is the end of the track.
track_length_units	string	FPA units to use for track length.
track_length_min	real	Display tracks only if they exceed the minimum length.
Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.		
category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with values in **look_up** table
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)
arrow_name	string	Arrow display name (from @arrow_display)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
pattern	string	Line pattern for boundary (see @presentation)
pattern_width	real	Height of pattern (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
pattern_length	real	Length of pattern before repeating (in display units)

Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once in this directive, for drawing multi-coloured curves!

line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.65

```
@lchain_tracks
{
  element             = storm_tracks;
  level               = surface;
  line_width          = 0.02;
  line_style          = solid;
  outline             = X11 Grey90;
}
```

Display tracks from the storm_tracks surface field as a solid grey line.

2.5.42 @legend

Display a graphic symbol or text at a location away from the main display. Note that this directive is similar to the @add and @text directives, except that the graphics and text from this directive are grouped separately from the remainder of the display, where they can be easily copied or moved to appropriate locations. Note that the @legend display is ALWAYS anchored to the centre of the page!

KEYWORD	TYPE	DESCRIPTION
symbol	string	Name of the graphic symbol file to display (Note: full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)
string	string	Text to display (Note: use \ before all control characters)
scale	real	Scale factor wrt original graphic (in percent)
text_size	real	Text height for label or text found in look up table (in display units)
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
x_off	real	Horizontal offset wrt current anchor position (in display units)
y_off	real	Vertical offset wrt current anchor position (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left centre right**
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)

Example 2.66

```
@legend
{
  symbol = $FPA/setup/pdf/svgmet/common/svg/low;
  scale  = 25;
  x_off  = 5.0;
  y_off  = 0.5;
}
```

Display the graphic from file `$FPA/setup/pdf/svgmet/common/svg/low` 5.0 "units" to the right and 0.5 "units" above the centre of the page. The graphic will be displayed at 25% of its original size. (Note that "units" is set by @display_units.)

2.5.43 @lines

Display lines from a "line" field (or from more than one "line" field) on the current map. Note that the **`category_attribute`** and **`category`** keywords can be used one or more times to display only selected lines. Note that the **`attribute`** and **`look_up`** keywords can be used to choose different presentations for lines with different attributes.

KEYWORD TYPE DESCRIPTION

Note that **`element`** and **`level`** can be specified more than once in this directive.
Note that if a single element name is matched with a list of level names, or vice versa, the single name must be given first. Multiple element and level name lists should have the same number.

KEYWORD	TYPE	DESCRIPTION
`element`	string	FPA element name
`element_list`	string	List of FPA element names
`level`	string	FPA level name
`level_list`	string	List of FPA level names

Note that **`category_attribute`** and **`category`** can be specified more than once in this directive, and that **`category_cascade`** determines how to combine more than one instance.

KEYWORD	TYPE	DESCRIPTION
`category_cascade`	string	How to combine more than one instance of **`category_attribute`** and **`category`** One of: **`and`** or **`or`** (default is **`and`**)
`category_attribute`	string	FPA attribute containing category names (default is **`FPA_category`**)
`category`	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **`category_attribute`**) (default is **`All`** names)
`attribute`	string	FPA attribute containing value to match with values in **`look_up`** table
`look_up`	string	File name for the look up table (Note: ".tab" suffix is optional)
`arrow_name`	string	Arrow display name (from @arrow_display)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`pattern`	string	Line pattern for boundary (see @presentation)
`pattern_width`	real	Height of pattern (in display units)
`pattern_length`	real	Length of pattern before repeating (in display units)

Note that **`line_width`**, **`line_style`**, **`outline`** and **`fill`** can be specified more than once in this directive, for drawing multi-coloured curves!

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`line_width`	real	Line width for boundary (in display units)
`line_style`	string	Line style (see @presentation)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`outline`	string	Line outline colour (see @presentation)
`fill`	string	Line fill colour (see @presentation)

Example 2.67

```
@lines
{
  element            = fronts;
  level              = surface;
  category_attribute = FPA_category;
  category           = stationary;
  attribute          = FPA_category;
  pattern            = front.sfc.stationary;
  pattern_width      = 0.150;
  pattern_length     = 0.750;
  outline            = none;
  fill               = CMYK 100 100 0 0;
  fill               = CMYK 0 100 100 0;
}
```

Display all lines with a value in attribute **`FPA_category`** of stationary from the "line" field surface fronts, using the pattern from file "front.sfc.stationary". The pattern will be 0.150 "units" wide and 0.750 "units" long, and will then repeat. The pattern will be drawn as a filled line with a colour in the first segment composed of 100% Cyan and 100% Magenta, and a colour in the second segment composed of 100% Magenta and 100% Yellow. (Note that the default pattern file would be `$FPA/config/patterns/front.sfc.stationary`. Note that "units" is set by @display_units.)

2.5.44 @loop_begin

Begin a loop to repeat all directives from this @loop_begin directive up to the matching @loop_end directive. The keyword values for each group and keyword are reset to the next value in the list for each iteration. If the number of keyword values is less than the number of iterations, then the command reuses the list of keyword values as many times as is necessary. The number of iterations is set either by the keyword **number_of_iterations** or it is determined by the number of features that match the category specified for a particular field (element/level pair).

Note that the **category_attribute** and **category** keywords can be used one or more times to choose only selected features for a particular field (element/level pair).

Note that the keywords and their values are included in another directive by using the format <***group_name*** in the keyword list.

Note that @loop_begin and @loop_end directives may be embedded.

KEYWORD	TYPE	DESCRIPTION
number_of_iterations	string	Number of iterations of directives (up to the matching @loop_end directive)

or

element	string	Element to loop on
level	string	Level to loop on
field_type	string	Use **Scattered** to loop on field labels. Otherwise not required.
area_type	string	Use **subareas** to loop based on subareas of an area and subarea attributes or **boundary** to loop based on areas and default area attributes.

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	value to replace magic term **<iteration_attribute>**.

KEYWORD	TYPE	DESCRIPTION
Note that **group_name**, **keyword_name**, **keyword_value** and **keyword_value_list** can be specified more than once in this directive, and that a **group_name** may not require a **keyword_name** if the directive to be accessed requires only a value.		
group_name	string	Group name for keyword list
keyword_name	string	Keyword name
keyword_value	string	Keyword value for one iteration
keyword_value_list	string(s)	Keyword value for a number of iterations

Example 2.68

```
@loop_begin {
  number_of_iterations = 5;
  group_name    = source_group;
  keyword_name = source;     keyword_value = interp;
  keyword_name = valid_time; keyword_value_list = 00 06 12 18 24; }
```

Set a "macro" called "source_group" containing values for the keywords source and valid_time to be used in subsequent directives. (The value of the source keyword will be interp for each iteration.)

Example 2.69

```
@loop_begin {
  element            = storms;
  level              = surface;
  category_attribute = storm_type;
  category           = tornado;
  attribute          = storm_id;
  }
```

Loop through all the storms with storm_type = tornado. Where <iteration_attribute> is the value of storm_id.

2.5.45 @loop_end

End a loop to repeat directives. The directives between the matching @loop_begin directive and this @loop_end directive will be repeated for the number of iterations specified in the @loop_begin directive.

Note that @loop_begin and @loop_end directives may be embedded.

Note that this directive has no keywords

Example 2.70

```
@loop_end { }
```

End of looping begun by an @loop_begin directive.

2.5.46 @loop_location_look_up

Build an internal location look up table from the current line, point or link chain feature in a loop. Note that the current feature is extracted from the field identified in the @loop_begin directive.

Line and link chain features can be used for space type displays. The horizontal extent of the display is determined by the length of the feature, with look up locations set each **location_interval** along the length, or at every point in the line or every node on the link chain if the **location_interval** is not set.

Point features can be used for time type displays, with look up locations at the point location for each time given in the **times** keyword, or at the current valid time if the **times** keyword is not set.

Link chain features can be used for time type displays, with look up locations at every node on the link chain, except when modified by the keywords **times** or the keywords **start_time** and **end_time**.

Note that times in the **times** keyword must be listed in increasing order.

Note that when using the **start_time** and **end_time** keywords, the start time must be before the end time.

KEYWORD	TYPE	DESCRIPTION
location_look_up	string	Name for location look up table. (Note: This is an internal name that can be referred to by other directives, and does not require a ".ltab" suffix.)
location_interval	real	Distance along feature for each look up location from a line or link chain feature.
location_units	string	FPA units for **location_interval**.
times	vtime(s)	List of times to use in look up table from a point or link chain feature.
start_time	vtime	Start time for look up locations from a link chain feature.
end_time	vtime	End time for look up locations from a link chain feature.
labels	string(s)	Labels to match up with locations or times.

Example 2.71 Create location look up table for space type display

```
@loop_location_look_up
  {
  location_look_up  = space_intervals;
  location_interval = 20;
  location_units    = km;
  }
```

Build an internal location look up table called "space_intervals" from locations every 20 km from features extracted from the field identified in a @loop_begin directive.

Example 2.72 Create location look up table for time type display

```
@loop_location_look_up
  {
  location_look_up = time_intervals;
  times            = 00/12:00;
  labels           = 12Z;
  times            = 00/15:00;
  labels           = 15Z;
  times            = 00/18:00;
  labels           = 18Z;
  times            = 00/21:00;
  labels           = 21Z;
  times            = 01/00:00;
  labels           = 00Z;
  times            = 01/03:00;
  labels           = 03Z;
  times            = 01/06:00;
  labels           = 06Z;
  times            = 01/09:00;
  labels           = 09Z;
  times            = 01/12:00;
  labels           = 12Z;
  }
```

Build an internal location look up table called "time_intervals" from locations every 3 hours from 12:00 "today" until 12:00 "tomorrow" from features extracted from the field identified in a @loop_begin directive.

2.5.47 @mapdef

Set the map definition for the current map.

A more complete description of map definitions, including acceptable formats for all parameters, can be found in the FPA Graphics Metafile Standard.

Note that this map definition does not need to match the map definition from the FPA that generates the graphics product, but this map definition should be completely within the map definition from the FPA.

KEYWORD	TYPE	DESCRIPTION
olat	latlon	Latitude of the map "origin" (see Format of Latitudes and Longitudes, (Section 1.3))
olon	latlon	Longitude of the map "origin" (see Format of Latitudes and Longitudes, (Section 1.3))
rlon	latlon	Vertical reference longitude of the map (see Format of Latitudes and Longitudes, (Section 1.3))
xmin	real	Start point of the x-axis (in units of **map_units** keyword below)
ymin	real	Start point of the y-axis (in units of **map_units** keyword below)
xmax	real	End point of the x-axis (in units of **map_units** keyword below)
ymax	real	End point of the y-axis (in units of **map_units** keyword below)
map_units	real	Number of meters (or degrees) per map unit

Example 2.73

```
@mapdef
{
  olat      = 44:40N;
  olon      = 63:35W;
  rlon      = 80.0W;
  xmin      = -1300;
  ymin      = -700;
  xmax      = 1100;
  ymax      = 1100;
  map_units = 1000;
}
```

Set a map definition for the current map with an origin of 44 degrees and 40 minutes North and 63 degrees and 35 minutes West, oriented with the vertical axis parallel to 80 degrees West longitude. All distances in the map definition will be in units of 1000 m, or km. The map will extend from -1300 km to 1100 km horizontally from the origin, and from -700 km to 1100 km vertically from the origin. (Note that the origin in this example is near the centre of the map.)

2.5.48 @perspective_view

Display graphics on the current map with a tilted or stretched perspective.

Note that the map is tilted back from the top, as if hinged along the bottom edge. The **y_tilt_off** keyword can be used to adjust this tilting position up or down from the bottom edge.

Note that the "eye" is located at twice the map height in front of the centre of the map. The **x_eye y_eye z_eye** keywords can be used to adjust this position.

Note that only the positions of symbols or text are adjusted by perspective, the **scale_to_perspective** keyword can be used to adjust the size of symbols or text to a size appropriate for the perspective position.

KEYWORD	TYPE	DESCRIPTION
show_perspective_view	yes/no	(default is **no**)
scale_to_perspective	yes/no	Adjust size of symbols or text on the map to match the appropriate perspective location? (default is **no**)
y_title_off	real	Vertical offset of tilt position (in display units from bottom of map)
title_angle	real	Title angle for map (in decimal degrees back from vertical)
x_eye	real	Horizontal offset of "eye" location (in display units from centre of map)
y_eye	real	Vertical offset of "eye" location (in display units from centre of map)
z_eye	real	Offset of "eye" location away from map (in display units from twice the map height in front of the centre of map)
x_stretch	real	Stretch map features horizontally (in percent)
y_stretch	real	Stretch map features vertically (in percent)

Example 2.74

```
@perspective_view {
  show_perspective_view = yes; scale_to_perspective = yes;
  y_tilt_offset = 4.25; tilt_angle = 60; x_stretch = 90; y_stretch = 110; }
```

Display all features on the current map as if the map display were tilted backwards 60 degrees about a horizontal axis 4.25 "units" above the bottom of the map, with all features stretched horizontally by 90% and vertically by 110%. All symbols and text on the map will be scaled to a size appropriate to the perspective view, that is, symbols and text towards the top of the map will be proportionately smaller, and those towards the bottom of the map will be proportionately larger than normal. (Note that "units" is set by @display_units.)

2.5.49 @presentation

Set the appearance of graphical objects and text for the current map.

Note that the @presentation directive with no "name" keyword sets the default appearance of all graphical objects and text for all subsequent directives.

Note that presentation keywords can also be included in other directives, where their effect is to change the appearance of the graphical object or text for that directive only.

KEYWORD	TYPE	DESCRIPTION
name	string	Name for a special presentation
Note that **line_width**, **line_style**, **outline** and **fill** can be specified more than once for named presentations.		
line_width	real	Line width (in display units)
line_style	string	Line style One of: **solid** or ***n dl sl dl sl ...*** where ***n*** is the number of ***dl***s and ***sl***s (2 or 4 or ...), ***dl*** is the length of the dashes (as a multiple of the **line_width** value), and ***sl*** is the distance between dashes (as a multiple of the **line_width** value)
outline	string	Line outline colour One of: **none** **CMYK** ***c m y k*** **RGB** ***r g b*** **X11** ***name*** where ***c,m,y,k*** and ***r,g,b*** are 0-100, and ***name*** is a valid X11 colour name
fill	string	Line fill colour One of: **none** **CMYK** ***c m y k*** **RGB** ***r g b*** **X11** ***name*** where ***c,m,y,k*** and ***r,g,b*** are 0-100, and ***name*** is a valid X11 colour name
outline_first	yes/no	Display outline before fill? (default is **no**)
interior_fill	string	Interior fill colour One of: **none** **CMYK** ***c m y k*** **RGB** ***r g b*** **X11** ***name*** where ***c,m,y,k*** and ***r,g,b*** are 0-100, and ***name*** is a valid X11 colour name

KEYWORD	TYPE	DESCRIPTION
symbol_fill_name	string	Symbol fill name for area interior One of: **none** or a symbol fill name (set in @symbol_fill_display)
pattern	string	Line pattern One of: **simple** or FPA Metafile containing pattern (Note that full path names are allowed. See Graphics Pattern Files for Drawing Lines, (Section 1.13) for default pattern file locations.)
pattern_width	real	Height of pattern (in display units)
pattern_length	real	Length of pattern before repeating (in display units)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left centre right**
text_size	real	Text height for label or text found in look up table (in display units)
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)
line_space	real	Text line spacing (as percent of character height)

Example 2.75

```
@presentation { outline = CMYK 0 0 0 100; fill = CMYK 0 60 25 10;
  font = "Times-Roman"; font_weight = normal;
  italics = no; text_size = 0.500; justification = left;
  char_space = 0; word_space = 100; line_space = 100; }
```

Set a presentation, to be used for all subsequent directives, with an outline colour composed of 100% Black and a fill colour composed of 60% Magenta, 25% Yellow, and 10% Grey. The text font is set to normal "Times-Roman" style with no italics. The text size is set to 0.500 "units" high and left justified. Text will be displayed with no spacing between characters, 100% of the character width between words, and 100% of the character height between lines. (Note that "units" is set by @display_units.)

2.5.50 @process

Execute one or more UNIX or Linux system commands.

Note that codewords accepted in @file_name directive are also accepted here.

KEYWORD	TYPE	DESCRIPTION
commands	(string)	One or more UNIX or Linux commands (Note: commands can include any of the codewords in Table 2.4 based on the full name of the output file, output directories, "pdf" file names, the **T0** time when the product is run, the current "valid_time" (**Tv**) or the actual clock time (**Tc**).

Table 2.4: Codewords

<file_name>	for full output file name
<file_name_base>	for full output file name without the extension
<iteration>	for a @loop_begin iteration value.
<iteration _attribute>	for the value of the attribute in a @loop_begin iteration.
<SETUP>	for name of local setup
<RTIME>	for **T0** timestamp
<VTIME>	for **Tv** timestamp
<default>	path to ***sub_directory*** in the ***output_directory***
<psmet>	for PSMet directory
<psout>	for PSMet output directory
<svgmet>	for SVGMet directory
<svgout>	for SVGMet output directory
<texmet>	for TexMet directory
<texout>	for TexMet output directory
<home>	for "home" directory defined in local setup file
<pdf>	for the ***product_name*** of the controlling product definition file (without trailing ".fpdf")
<year>	for **T0** year (such as 2008)
<month>	for **T0** month (01-12)
<day>	for **T0** day (01-31)
<julian>	for **T0** day of the year (001-366)
<hour>	for **T0** hour of the day (00-23)
<minute>	for **T0** minute in hour (00-59)
<local>	L if **T0** time is a local time
<v_year>	for **Tv** year (such as 2008)
<v_month>	for **Tv** month (01-12)
<v_day>	for **Tv** day (01-31)
<v_julian>	for **Tv** day of the year (001-366)
<v_hour>	for **Tv** hour of the day (00-23)
<v_minute>	for **Tv** minute in hour (00-59)
<v_local>	L if **Tv** time is a local time

Table 2.4: (continued)

<p_hour>	for **Tv-T0** in hours
<p_minute>	for **Tv-T0** in minutes
<p_hr_min>	for **Tv-T0** in hours and minutes (xx:tt)
<c_year>	for **Tc** year (such as 2008)
<c_month>	for **Tc** month (01-12)
<c_day>	for **Tc** day (01-31)
<c_julian>	for **Tc** day of the year (001-366)
<c_hour>	for **Tc** hour of the day (00-23)
<c_minute>	for **Tc** minute in hour (00-59)

Example 2.76

```
@process
{
  display <file_name>;
  cp <file_name> <home>/transfer;
}
```

Display the output graphics file using the "display" (from ImageMagick) command, and then copy the output file to the "transfer" subdirectory of the "home" directory (defined in the local setup file).

2.5.51 @projection

Set the map projection for the current map.

A more complete description of map projections, including acceptable formats for all parameters, can be found in the FPA Graphics Metafile Standard.

KEYWORD	TYPE	DESCRIPTION
type	string	Keyword for projection (predefined) One of: **latitude_longitude** **plate_caree** **polar_stereographic** **lambert_conformal** **mercator_equatorial** **rotated_lat_lon** **oblique_stereographic**
ref1	string	Projection dependent reference value
ref2	string	Projection dependent reference value
ref3	string	Projection dependent reference value
ref4	string	Projection dependent reference value
ref5	string	Projection dependent reference value

Table 2.5: Projection types and associated parameters

Projection	Type	Parameters
Lambert Conformal	**lambert_conformal**	upper reference latitude lower reference latitude
Latitude-Longitude	**latitude_longitude**	
Mercator Equatorial	**mercator_equatorial**	
Oblique Stereographic	**oblique_stereographic**	central latitude central longitude secant angle [optional]
Plate-Caree	**plate_caree**	
Polar Stereographic	**polar_stereographic**	**north** or **south** "true" latitude
Rotated Latitude-Longitude	**rotated_lat_lon**	bottom axis latitude bottom axis longitude rotation angle [optional]

Example 2.77

```
@projection { type = polar_stereographic; ref1 = north; ref2 = 60; }
```

Set a map projection for the current map as north **polar_stereographic** true at 60 degrees North.

2.5.52 @reset_contour_presentation

Removes all current contour presentations set by calling the @contour_presentation directive.

Note that this directive has no keywords.

Example 2.78

```
@reset_contour_presentation { }
```

Removes all current contour presentations set by the @contour_presentation directive.

2.5.53 @reset_geo_presentation

Removes all current geographic presentations set by calling the @geo_presentation directive.

Note that this directive has no keywords.

Example 2.79

```
@reset_geo_presentation { }
```

Removes all current geographic presentations set by the @geo_presentation directive.

2.5.54 @resolution

Set the spline resolution for "continuous" or "vector" fields for the current map.

Note that **map_units** is automatically reset to the value of **map_units** in the @mapdef directive, and **res** is scaled accordingly.

Note that values determined by **xmax** - **xmin** and **ymax** - **ymin** keywords in the @mapdef directive should be multiples of the the value of the **res** keyword, given below.

KEYWORD	TYPE	DESCRIPTION
res	real	Spline resolution (in units of **map_units** keyword below)
map_units	real	Number of meters (or degrees) per map unit (Default is **map_units** from @mapdef directive)

Example 2.80

```
@resolution { res = 200; map_units = 1000; }
```

Set a spline resolution for "continuous" or "vector" fields to 200 distance units, where all distance units are expressed in 1000 m, or km.

2.5.55 @sample_field

Display values (or corresponding symbols or text) sampled from FPA fields on the current map or in a table or in a cross section or at a given location.

Complex (multi-attribute) samples are described at the end of this section.

Note that the **category_attribute** and **category** keywords can be used to sample only selected values.

Note that the **case** and **case_look_up** keywords can only be used for simple (single-attribute) samples.

Note that every sample requires a **format** keyword. A style of **none** will display only a mark or label_display. A **direct** style format will display the value sampled from FPA. A **symbol** or **text** style format also requires a **look_up** keyword to define a look up table, and the value sampled from FPA is cross-referenced in this look up table to identify the symbol or text string to display. A **wind_barb** format displays winds using the presentation from the directive @wind_barb_presentation. A **wind_text** or **wind_symbol** format displays winds using the presentation from the directive @wind_presentation. A **vector_text** or **vector_symbol** format displays "vector" field values using the presentation from the directive @vector_presentation.

Note that **space_route** or **time_route** cross sections also allow **format** styles of **tail_wind_barb tail_wind_text tail_wind_symbol** or styles of **cross_wind_barb cross_wind_text cross_wind_symbol**. These will display the tail wind or cross wind component of the sampled wind with respect to the motion along the cross section at the sampled location.

Note that for cross section samples, entries in **location_look_up** table MUST be entries from the location look up table in directive @define_cross_section.

It is now possible to sample information from ASCII data files, and display the data on a map or in a table. Note that all data files must be arranged row by row, with each data parameter in a separate column. The keywords **data_file**, **data_file_format**, **data_file_units**, **data_file_wind_units**, **vertical_data_file**, **vertical_data_file_format**, and **vertical_data_file_units** provide control for this action. Note that all parameters in the data file are accessed by magic attributes, as follows:

GPGEN_ident	Identifier string
GPGEN_lat or **GPGEN_lat_ddmm**	Latitude location
GPGEN_lon or **GPGEN_lon_ddmm**	Longitude location
GPGEN_label	Data string
GPGEN_value	Data value
GPGEN_wind	Data wind direction (degrees true)

The keywords **proximity**, **proximity_units** and **display_at_feature** allow sampling attributes from discrete areas, line features, link chains or scattered points within the given proximity of the sample location. The magic attribute **GPGEN_proximity** can be used to display the distance from the sample location to the feature in km.

KEYWORD	TYPE	DESCRIPTION
element	string	FPA element name
level	string	FPA level name

KEYWORD	TYPE	DESCRIPTION
equation	string	FPA equation string for calculating a field (refer to Appendix B of Administrator's Guide for format of equations)
field_type	string	Use **Scattered** to sample field labels. Otherwise not required.
units	string	FPA units for "continuous" or "vector" field samples
geo_file	string	FPA Metafile for sampling geography (Note: this must be the name of a file in the "Maps" directory or a full path name)
proximity	real	Only display sample if area, line, link chain or scattered feature is within the given proximity
proximity_units	string	FPA units for proximity
display_at_feature	yes/no	Display the attributes at the closest feature, or at the sample location respectively

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which objects to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)

Note that **case** and **case_look_up** can be specified more than once in this directive!

case	string	Time of day or hemisphere in which to use **case_look_up** look up table (given below) One of:**day night dusk** **northern_hemisphere** **southern_hemisphere**
case_look_up	string	File name for the optional look up table (Note: ".tab" suffix is optional)

KEYWORD	TYPE	DESCRIPTION
`time_zone`	string	Display the time (from a magic attribute using **strftime** format) adjusted to another time zone (see @write_time for more information)
`language`	string	Display the time (from a magic attribute using **strftime** format) adjusted to another language (see @write_time for more information)
`format`	string	Format of attribute to display One of: **`direct`** ***`<cform>`*** use value **`symbol`** ***`<cform>`*** use look up **`text`** ***`<cform>`*** use look up **`wind_barb`** for wind values **`wind_text`** for wind values **`wind_symbol`** for wind values **`vector_text`** for vector values **`vector_symbol`** for vector values **`none`** do not display value or additionally for **`route`** cross sections: **`tail_wind_barb`** for tail wind values **`tail_wind_text`** for tail wind values **`tail_wind_symbol`** for tail wind values **`cross_wind_barb`** for cross wind values **`cross_wind_text`** for cross wind values **`cross_wind_symbol`** for cross wind values (Note: ***`<cform>`*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **`look_up`** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***`<cform>`*** conversion format)
`data_file`	string	Identify the name of the ASCII data file to sample. May contain codewords: <default>, <psmet>, <psout>, <svgmet>, <svgout>, <texmet>, <texout> or <home> to refer to PSMet, SVGMet or TexMet directories in the same FPA database.

KEYWORD	TYPE	DESCRIPTION
data_file_format	string	Identify the format of the data in the ASCII data file with a list of recognized format types separated by the same characters used to separate the values in the ASCII data file. Recognized types: **identifier** Identifier string **latitude** Latitude location **longitude** Longitude location **timestamp** Valid time **label** Data string **value** Data value **units** FPA units of data value **wind_direction** Data wind direction (degrees true) **wind_speed** Data wind speed **wind_gust** Data wind gust **wind_units** FPA units of data wind speed or gusts **-** Placeholder in data file Note that either the **identifier** or the **latitude** and **longitude** must be present to identify the data locations, but all other format types are optional. Note that **label**, **value** and **wind** format types can all be present in the same **data_file_format** keyword.
data_file_units	string	FPA units for data values Note that all data values are automatically converted to MKS units. Either the **units** format type or the **data_file_units** keyword are required for values in the data file that are non-MKS.
data_file_wind_units	string	FPA units for wind speed or gust values Note that all wind speed and gust values are automatically converted to knots. Either the **wind_units** format type or the **data_file_wind_units** keyword are required for wind speeds or gusts in the data file that are not in knots.
vertical_data_file	string	Identify the name of the ASCII data file to sample. May contain codewords: <default>, <psmet>, <psout>, <svgmet>, <svgout>, <texmet>, <texout> or <home> to refer to PSMet, SVGMet or TexMet directories in the same FPA database.

KEYWORD	TYPE	DESCRIPTION
vertical_data_file_format	string	Identify the format of the data in the ASCII data file with a list of recognized format types separated by the same characters used to separate the values in the ASCII data file. Recognized types: **identifier** Identifier string **latitude** Latitude location **longitude** Longitude location **timestamp** Valid time **label** Data string **value** Data value **units** FPA units of data value **-** Placeholder in data file Note that either the **identifier** or the **latitude** and **longitude** must be present to identify the data locations, but all other format types are optional. Note that only the **value** format type can be used to set the vertical positions in a cross section.
vertical_data_file_units	string	FPA units for data values Note that all data values are automatically converted to MKS units. Either the **units** format type or the **vertical_data_file_units** keyword are required for values in the data file that are non-MKS.
mark	string	Name of graphic file to display as a marker (Note that the full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)
mark_scale	real	Scale factor wrt original mark (in percent)
symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent)
text_size	real	Text height for label or text found in look up table (in display units)
display_name	string	Named display (from @label_display)

KEYWORD	TYPE	DESCRIPTION
display_type	string	Type of display about label One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)
width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **60** is acceptable for most font types
height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **100** is acceptable for most font types
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
rotate_to_latitude	yes/no	Rotate parallel to latitude. **rotation** is appended. (Default is no).
or		
rotate_to_longitude	yes/no	Rotate parallel to longitude. **rotation** is appended. (Default is no).
or		
rotation_attribute	string	Set rotation based on value of attribute. **rotation** is appended.
constrain_rotation	yes/no	Used with **rotate_to_latitude**, **rotate_to_longitude**, or **rotation_attribute** to force label to always appear right-side up with respect to the bottom of the map. (Default is "yes")
x_display_off	real	Horizontal offset of entire label, including the location mark (in display units)
y_display_off	real	Vertical offset of entire label, including the location mark (in display units)
x_off	real	Horizontal offset of value, symbol or text wrt current sample location (in display units)
y_off	real	Vertical offset of value, symbol or text wrt current sample location (in display units)

KEYWORD	TYPE	DESCRIPTION
source	string	FPA source name for data directory for field to sample (Note: the format for this keyword would be ***source*** if the data directory has no sub sources, or ***source*:*subsource*** if the data directory has sub sources) (Default is current **source** from @set_source directive)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)
For map or point samples ...		
fit_to_map	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)
fit_to_map_ref	string	Reference position to fit label to current map One of: **none upper lower left right** (default is **none**) (Note: this parameter usually indicates the position of the label **mark** wrt the rest of the label)
For point samples ...		
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
map_x	real	Position on x-axis of map (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units**)
map_y	real	Position on y-axis of map (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units**)
map_units	real	Number of meters (or degrees) per map unit (Default is **map_units** from @mapdef directive)
location_ident	string	Identifier for location in **location_look_up**
grid_name	string	Grid name (from @define_sample_grid) (Note: the object will be displayed once for each location on the grid)
list_name	string	List name (from @define_sample_list) (Note: the object will be displayed once for each location in the list)

KEYWORD	TYPE	DESCRIPTION
For point or table samples ...		
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)
table_name	string	Table name (from @define_table) (Note: the object will be displayed once for each call to the @table_site directive)
For cross section samples ...		
cross_section_name	string	Cross section name (from @define_cross_section)
location_look_up	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional) (Default is **location_look_up** from @define_cross_section directive)
or		
location_distances	real(s)	A list of distances from look up table in @define_cross_section to sample at.
location_units	string	FPA units of **location_distances**
or		
location_times	real(s)	A list of times from look up table in @define_cross_section to sample at.
or		
location_fractions	real(s)	A list (0-1) of fractions along the horizontal axis from look up table in @define_cross_section to sample at.
vertical_look_up	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional) (Default is **vertical_look_up** from @define_cross_section directive)
vertical_element	string	FPA element name for vertical parameters
vertical_level	string	FPA level name for vertical parameters
vertical_equation	string	FPA equation string for calculating values (refer to Appendix B of Administrator's Guide for format of equations)
vertical_field_type	string	Use **Scattered** to sample field labels. Otherwise not required.
vertical_units	string	FPA units for field values
vertical_attribute	string	FPA attribute for vertical value (default is **GPGEN_default_attribute**)

KEYWORD	TYPE	DESCRIPTION
vertical_attribute_upper	string	FPA attribute for upper value (Note: **GPGEN_cross_section_top** can be used for the top of the cross_section) (default is **GPGEN_default_attribute**)
vertical_attribute_lower	string	FPA attribute for lower value (Note: **GPGEN_cross_section_base** can be used for the bottom of the cross section) (default is **GPGEN_default_attribute**)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left centre right**
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)

Example 2.81

```
@sample_field
{
  element = pressure;
  level   = msl;
  units   = mb;
  format  = %.1f mb;
  lat     = 35N;
  lon     = 40W;
  x_off   = 0;
  y_off   = 0;
  fill    = X11 red;
}
```

Sample the msl pressure field at latitude 35 degrees North and longitude 40 degrees West. The value will be displayed as a real number rounded to the nearest 0.1 mb and followed by " mb" (for example, "1004.6 mb"). The value will be displayed as characters filled with red line at the given latitude and longitude on the current map.

Example 2.82 Example data_file_format

The **`data_file_format`** string is a list of recognized format types separated by the same characters used to separate the values in the ASCII data file.

The following format would indicate that the identifier string is located in the second column of the ASCII data file, the time stamp in the fourth column, data file string in the seventh column, and that the columns are separated by white space (blanks or tabs).

```
data_file_format = - identifier - timestamp - - label
```

The following format would indicate that the latitude and longitude are located in the first and second columns of the ASCII data file, a data file value in the fifth column, the FPA units for the value in the fourth column, and that the columns are separated by commas.

```
data_file_format = latitude,longitude,-,units,value
```

See Example 2.86.

Example 2.83

```
@sample_field
{
  table_name    = T1;
  element       = weather_system;
  level         = surface;
  attribute     = clds_and_wx;
  format        = symbol;
  look_up       = weather_day;
  case          = night;
  case_look_up  = weather_ngt;
  symbol_scale  = 30;
  source        = interp;
  valid_time    = 01/04;
  x_off         = 2.30;
  y_off         = 0.25;
  justification = centre;
}
```

Sample the surface weather_system field from the interp source at 04 GMT, 01 days after the current day for each site (from the @table_site directive) of a table called "T1" (from the @define_table directive). Display the sampled value for the clds_and_wx attribute as a symbol by matching the attribute value to the values within the look up table `weather_day.tab` (or by matching the attribute value to the values within the look up table `weather_ngt.tab` at nighttime). The symbol will be 30% of its original size, and displayed 2.30 "units" to the right and 0.25 "units" above the current location in the table. The symbol will be centre justified. (Note that "units" is set by @display_units.)

Complex (multi-attribute) samples allow the display of several sampled parameters within the same directive. The entire complex sample is displayed with respect to the **display_name**, **display_type**, **rotation**, **x_off**, and **y_off** keywords set above.

Attribute keywords apply only to the attribute that they follow.

The position of an attribute in a complex sample can be related to the position of other attributes through use of the keywords **attribute_anchor**, **attribute_ref**, **attribute_justification**, and **attribute_vertical_just**. The **attribute_ref** keyword defines the starting position of the attribute with respect to a location about the symbol or text identified by the **attribute_anchor** keyword. The **attribute_justification** and **attribute_vertical_just** keywords define the horizontal and vertical positioning of the attribute with respect to this starting position.

The display parameters of each attribute default to the values of the **look_up**, **justification**, **units**, **format**, **symbol_scale**, **text_size**, **width_scale**, and **height_scale** keywords and the **fill**, **line_style**, **line_width**, **outline**, **outline_first**, **char_space**, **word_space**, **font**, **font_weight**, and **italics** presentation keywords set above. These default values can be replaced by the appropriate attribute keyword listed below, to create multi-colour multi-format complex labels.

Note that the **attribute_show** keyword can be used to "turn off" the display of attributes, while still maintaining positioning with respect to other attributes.

KEYWORD	TYPE	DESCRIPTION
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
attribute_look_up	string	File name for look up table (Note: ".tab" suffix is optional) (default is **look_up** set above)
attribute_show	yes/no	Display attribute? (default is yes)
attribute_anchor	string	Reference "anchor" for displaying attribute One of: **none** or a previously listed **attribute** (default is **none**, which corresponds to the starting position of the complex label)
attribute_ref	string	Starting position for displaying attribute (a position on the "attribute_anchor" label) One of: **lower_left centre_left upper_left lower_centre centre upper_centre lower_right centre_right upper_right** (default is **lower_centre**)
attribute_justification	string	Attribute text justification (horizontal) One of: **left centre right** (default is **justification** set above)
attribute_vertical_just	string	Attribute text justification (vertical) One of: **bottom**, **centre**, **top** (default is **bottom**)
attribute_units	string	FPA units for "continuous" or "vector" field samples

KEYWORD	TYPE	DESCRIPTION
attribute_format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **symbol** ***<cform>*** use look up **text** ***<cform>*** use look up **wind_barb** for wind values **wind_text** for wind values **wind_symbol** for wind values **vector_text** for vector values **vector_symbol** for vector values **none** do not display value or additionally for **route** cross sections: **tail_wind_barb** for tail wind values **tail_wind_text** for tail wind values **tail_wind_symbol** for tail wind values **cross_wind_barb** for cross wind values **cross_wind_text** for cross wind values **cross_wind_symbol** for cross wind values (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **attribute_look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)
attribute_symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent) (default is **symbol_scale** set above)
attribute_text_size	real	Text height for attribute or text found in look up table (in display units) (default is **text_size** set above)
attribute_display_name	string	Named display (from @label_display)
attribute_display_type	string	Type of display about attribute One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)

KEYWORD	TYPE	DESCRIPTION
attribute_width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **width_scale** set above)
attribute_height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **attribute_text_size** keyword value) (default is **height_scale** set above)
attribute_x_off	real	Horizontal offset of attribute display (in display units)
attribute_y_off	real	Vertical offset of attribute display (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
attribute_line_width	real	Line width (in display units) (default is **line_width** set above)
attribute_line_style	string	Line style (see @presentation) (default is **line_style** set above)
attribute_outline	string	Line outline colour (see @presentation)
attribute_fill	string	Line fill colour (see @presentation)
attribute_outline_first	yes/no	Display outline first? (default is **outline_first** set above)
attribute_font	string	Text font name recognized by output device or graphics post processor (default is **font** set above)
attribute_font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black** (default is **font_weight** set above)
attribute_italics	yes/no	Text in italics? (default is **italics** set above)
attribute_char_space	real	Text character spacing (as percent of average character width) (default is **char_space** set above)
attribute_word_space	real	Text word spacing (as percent of average character width) (default is **word_space** set above)

Example 2.84

```
@sample_field
{
  element                    = weather_system;
  level                      = surface;
  category_attribute         = FPA_category;
  category                   = freezing frozen precip vis cloud;
  format                     = direct;
  mark                       = triangle_up;
  mark_scale                 = 100;
  display_name               = label1;
  display_type               = sized_box;
  lat                        = 41.5N;
  lon                        = 67.5W;

  attribute                  = cloud_top;
  attribute_anchor           = none;
  attribute_display_name     = under1;
  attribute_display_type     = sized_underline;
  attribute_x_off            = -0.25;
  attribute_justification     = centre;

  attribute                  = cloud_base;
  attribute_anchor           = cloud_top;
  attribute_ref              = lower_centre;
  attribute_display_name     = over1;
  attribute_display_type     = sized_underline;
  attribute_justification     = centre;
  attribute_vertical_just    = top;
  .....
}
```

Sample the surface weather_system field at latitude 41.5 degrees North and longitude 67.5 degrees West and display the sampled value as a complex sample if the value for the attribute **FPA_category** is one of **freezing**, **frozen**, **precip**, **vis**, or **cloud**. The sample location will be marked by the symbol **none**. The full complex sample will be enclosed in a box scaled to the size of the sampled values, according to the display called label1 (from the @label_display directive). The attribute cloud_top will be displayed 0.25 "units" to the left of the sample location, and centre justified. The cloud_top attribute will be underlined according to a display called underline1 (from the @label_display directive). The top edge of the attribute cloud_base will be displayed at the lower_centre of the cloud_top attribute, and centre justified. The cloud_base attribute will be underlined according to a display called overline1 (from the @label_display directive).

(Note that if the value of the cloud_base attribute is blank, then any positions using an attribute_anchor of cloud_base will be positioned at the lower_centre of the cloud_top attribute!)

(Note that the remaining keywords from this example have not been included.)

(Note that the default PSMet mark would be `$FPA/setup/pdf/psmet/common/ps/triangle_up.ps`, and that the default SVGMet mark would be `$FPA/setup/pdf/svgmet/common/svg/triangle_up.svg`.)

(Note that **units** is set by @display_units.)

Example 2.85

```
@sample_field
{
  element                   = weather_system;
  level                     = surface;
  category_attribute        = FPA_category;
  category                  = freezing frozen precip;
  format                    = direct;
  cross_section_name        = route_cross_section;
  location_look_up          = route_samples;
  vertical_look_up          = xsect_heights;
  vertical_element          = weather_system;
  vertical_level            = surface;
  vertical_attribute_upper = cloud_top;
  vertical_attribute_lower = cloud_base;
  y_display_off             = -0.40;

  attribute                 = weather;
  attribute_anchor          = none;
  attribute_format          = symbol;
  attribute_symbol_scale   = 80;
  attribute_look_up         = weather_syno;
  attribute_justification  = left;
  attribute_vertical_just  = centre;
}
```

Sample the surface weather_system field for a cross section named route_cross_section (set in the @define_cross_section directive). The field is sampled at each location in the location look up table `route_samples.ltab` (whose locations must be a subset of the default location look up table set in the @define_cross_section directive). Display the sample only if the value for the attribute FPA_category is one of freezing, frozen or precip. Display the sample at a location 0.40 "units" below a point half way between the vertical location of the attributes cloud_top and cloud_base sampled from the surface weather_system field, and converted to vertical locations using the vertical look up table `xsect_heights.ztab`". Display the sampled value for the weather attribute as a symbol by matching the attribute value to the values within the look up table "weather_syno.tab". Display the symbol at 80% of its original size, and with the left edge and vertical centre of the symbol located at the current sample location. (Note that "units" is set by @display_units.)

Example 2.86 Sample from a data file

```
@sample_field
{
  data_file        = temp_file.txt;
  data_file_format = ident latitude longitude timestamp - value;
  data_file_units  = degreesF;
  location_look_up = sample_grid.ltab;

  attribute        = GPGEN_value;
  attribute_format = direct %f;
  attribute_justification = left;

  attribute        = GPGEN_ident;
  attribute_format = direct %s;
  attribute_justification = right;
}
```

Sample the data file `temp_file.txt` and display the site identifier (first column) and temperature value located in the (sixth column). The temperature value found in the data file is in degreesF and will be converted to MKS units.

2.5.56 @sample_wind

Display cross-referenced winds from FPA fields on the current map or in a table.

Note that the cross-reference "Fpa_Wind" refers to the FPA wind field.

Note that every sample requires a **format** keyword. A **wind_barb** format displays winds using the presentation from the directive @wind_barb_presentation, while a **wind_text** or **wind_symbol** format displays winds using the presentation from the directive @wind_presentation.

Note that **space_route** or **time_route** cross sections also allow **format** styles of **tail_wind_barb tail_wind_text tail_wind_symbol** or styles of **cross_wind_barb cross_wind_text cross_wind_symbol**. These will display the tail wind or cross wind component of the sampled wind with respect to the motion along the cross section at the sampled location.

Note that for cross section samples, entries in **location_look_up** table MUST be entries from the location look up table in directive @define_cross_section.

KEYWORD	TYPE	DESCRIPTION
wind_crossref	string	FPA wind cross-reference
units	string	FPA units for wind speeds or gusts to display
format	string	Format of attribute to display One of: **wind_barb** for wind values **wind_text** for wind values **wind_symbol** for wind values or additionally for **route** cross sections: **tail_wind_barb** for tail wind values **tail_wind_text** for tail wind values **tail_wind_symbol** for tail wind values **cross_wind_barb** for cross wind values **cross_wind_text** for cross wind values **cross_wind_symbol** for cross wind values
mark	string	Name of graphic file to display as a marker (Note that the full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)
mark_scale	real	Scale factor wrt original mark (in percent)
display_name	string	Named display (from @label_display)
display_type	string	Type of display about label One of: **sized_box** scaled box **fixed_box** fixed size box **sized_ellipse** scaled ellipse **fixed_ellipse** fixed size ellipse **sized_underline** scaled underline **fixed_underline** fixed size line (Note: the ***sized_...*** display types scale the type of display to the size of the label)

KEYWORD	TYPE	DESCRIPTION
width_scale	real	Approximate width of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **60** is acceptable for most font types
height_scale	real	Approximate height of characters, for use with ***sized_...*** display types (as percent of **text_size** keyword value) The default value of **100** is acceptable for most font types
rotation	real	Angle of rotation (in counter-clockwise decimal degrees)
rotate_to_latitude	yes/no	Rotate parallel to latitude. **rotation** is appended. (Default is no).
or		
rotate_to_longitude	yes/no	Rotate parallel to longitude. **rotation** is appended. (Default is no).
x_display_off	real	Horizontal offset of entire label, including the location mark (in display units)
y_display_off	real	Vertical offset of entire label, including the location mark (in display units)
x_off	real	Horizontal offset of wind text or barb wrt current sample position
y_off	real	Vertical offset of wind text or barb wrt current sample position
source	string	FPA source name for data directory for field to sample (Note: the format for this keyword would be ***source*** if the data directory has no sub sources, or ***source*:*subsource*** if the data directory has sub sources) (Default is current **source** from @set_source directive)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)

For map or point samples ...

fit_to_map	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)

KEYWORD	TYPE	DESCRIPTION
For point samples ...		
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
map_x	real	Position on x-axis of map (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units**)
map_y	real	Position on y-axis of map (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units**)
map_units	real	Number of meters (or degrees) per map unit (Default is **map_units** from @mapdef directive)
location_ident	string	Identifier for location in **location_look_up**
grid_name	string	Grid name (from @define_sample_grid) (Note: the object will be displayed once for each location on the grid)
list_name	string	List name (from @define_sample_list) (Note: the object will be displayed once for each location in the list)
For point or table samples ...		
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)
For table samples ...		
table_name	string	Table name (from @define_table) (Note: the object will be displayed once for each call to the @table_site directive)

KEYWORD	TYPE	DESCRIPTION
For cross section samples ...		
cross_section_name	string	Cross section name (from @define_cross_section)
location_look_up	string	File name for the location look up table for the cross section horizontal axis (Note: one point will be sampled for each location in the location look up table) (Note: ".ltab" suffix is optional) (Default is **location_look_up** from @define_cross_section directive)
or		
location_distances	real(s)	A list of distances from look up table in @define_cross_section to sample at.
location_units	string	FPA units of **location_distances**
or		
location_times	real(s)	A list of times from look up table in @define_cross_section to sample at.
or		
location_fractions	real(s)	A list (0-1) of fractions along the horizontal axis from look up table in @define_cross_section to sample at.
vertical_look_up	string	File name for the vertical look up table, for converting values to locations on the cross section vertical axis (Note: ".ztab" suffix is optional) (Default is **vertical_look_up** from @define_cross_section directive)
vertical_element	string	FPA element name for vertical parameters
vertical_level	string	FPA level name for vertical parameters
vertical_equation	string	FPA equation string for calculating values (refer to Appendix B of Administrator's Guide for format of equations)
vertical_units	string	FPA units for field values
vertical_attribute	string	FPA attribute for vertical value (default is **GPGEN_default_attribute**)
vertical_attribute_upper	string	FPA attribute for upper value (Note: **GPGEN_cross_section_top** can be used for the top of the cross_section) (default is **GPGEN_default_attribute**)
vertical_attribute_lower	string	FPA attribute for lower value (Note: **GPGEN_cross_section_base** can be used for the bottom of the cross section) (default is **GPGEN_default_attribute**)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left centre right**
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)

Example 2.87

```
@sample_wind { wind_crossref = Fpa_Wind;
  lat = 44:40N; lon = 63:35W;
   units = knots; format = wind_barb;
  outline = X11 Red; fill = X11 Red; line_width = 0.05; }
```

Sample the Fpa_Wind wind cross-reference at latitude 44 degrees and 40 minutes North and longitude 63 degrees and 35 minutes West. The wind will be displayed as a **wind_barb** scaled in units of knots at the given location on the current map. The wind barb will be drawn with Red line 0.05 "units" wide, and with a barb presentation given by a @wind_barb_presentation directive. Note that "units" is set by @display_units)

Example 2.88

```
@sample_wind { wind_crossref = Fpa_Wind;
  table_name = T1; units = knots; format = wind_text;
  source = depict; valid_time = 12;
  x_off = 3.0; y_off = 0; }
```

Sample the Fpa_Wind wind cross-reference from the depict source 12 hours after the current time for each site (from the @table_site directive) of a table called "T1" (from the @define_table directive). Display the sampled wind scaled in units of knots, and with a presentation given by a @wind_presentation directive. The sampled wind will be displayed 3.000 "units" to the right of the current location in the table. (Note that "units" is set by @display_units)

2.5.57 @set_source

Set the data directory and valid time for fields to sample.

Note that all times are relative to the **T0** time from FPA.

KEYWORD	TYPE	DESCRIPTION
source	string	FPA source name for data directory for field to sample (Note: the format for this keyword would be ***source*** if the data directory has no sub sources, or ***source*:*subsource*** if the data directory has sub sources)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5))

Example 2.89

```
@set_source { source = depict; valid_time = 00; }
```

Set the data directory for fields to sample to the FPA depict source at the current time

Example 2.90

```
@set_source { source = GEM; valid_time = 1/00; }
```

Set the data directory for fields to sample to the FPA GEM source at 00 GMT 1 day after the current time (that is, tomorrow).

2.5.58 @size

Set the size of the output display (the "paper" size).

KEYWORD	TYPE	DESCRIPTION
width	real	Width of output display (in display units)
height	real	Height of output display (in display units)

Example 2.91

```
@size { width = 11.00; height = 8.50; }
```

Set the size of the output display to 11.00 "units" wide by 8.50 "units" high. (Note that "units" is set by @display_units.)

2.5.59 @symbol_fill_display

Set display parameters for filling an area with symbols.

Note that symbol fills are always anchored to the centre of the current map, and all keywords are measured with respect to this location.

KEYWORD	TYPE	DESCRIPTION
symbol_fill_name	string	Symbol fill name
symbol	string	Name of the graphic symbol file to display (Note: full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)
symbol_scale	real	Scale factor wrt original symbol found in look up table (in percent)
symbol_rotation	real	Angle of rotation for graphic (in counter-clockwise decimal degrees)
x_off	real	Horizontal offset of initial graphic symbol wrt centre of current map (in display units)
y_off	real	Vertical offset of initial graphic symbol wrt centre of current map (in display units)
x_repeat	real	Horizontal spacing for subsequent "columns" of graphic symbols (in display units)
y_repeat	real	Vertical spacing for subsequent "rows" of graphic symbols (in display units)
x_shift	real	Horizontal offset between graphic symbols in adjoining "rows" (in display units)
y_shift	real	Vertical offset between graphic symbols in adjoining "columns" (in display units)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)

Example 2.92

```
@symbol_fill_display
  {
  symbol_fill_name = cross_fill;
  symbol           = cross;
  symbol_scale     = 40.0;
  symbol_rotation  =  0.0;
  x_off            =  0.0;
  x_repeat         =  0.2;
  x_shift          =  0.1;
  y_off            =  0.0;
  y_repeat         =  0.2;
  y_shift          =  0.0;
  line_width       =  0.005;
  outline          = X11 Red;
  fill             = none;
  }
```

Set parameters for a symbol fill display called cross_fill. The display will fill an outlined area with graphics from file "cross", beginning at a location at the centre of the current map, repeated every 0.2 "units" horizontally and every 0.2 "units" vertically, and offset 0.1 "units" horizontally between adjoining "rows" of symbols. The symbols will be drawn with a line 0.005 "units" wide in X11 Red. The graphic will be 40% of its original size. See Graphics Symbol Files, (Section 1.12) for default symbol location. (Note that "units" is set by @display_units.)

2.5.60 @table_site

Set parameters for locations in a table at which to sample data.

Note that the type of table is set in the @define_table directive. A column type table arranges the table data by columns, and each call to the @table_site directive will create another row of data in the table. A row type table arranges the table data by rows, and each call to the @table_site directive will create another column of data in the table.

Note that data in the table is "anchored" to the table location set in the @define_table directive according to the **x_off** or **y_off** keywords given below. In a column type table, data for each site forms another row in the table, offset vertically from the table location according to **y_off**. In a row type table, data for each site forms another column in the table, offset horizontally from the table location according to **x_off**.

Note that the data locations for sampling or display can be set directly as latitude-longitude or map grid locations, but can also be set indirectly as a location identifier, which is matched to a "location_look_up" table (set in the sampling or display directives) to determine the location.

Note that the presentation keywords are used for display of the **site_label**.

KEYWORD	TYPE	DESCRIPTION
table_name	string	Table name (from @define_table)
site_label	string	Site label for location.
text_size	real	Text height for label or text found in look up table (in display units)
rotation	real	Text angle of rotation for site label (in counter-clockwise decimal degrees)
x_label	real	Horizontal offset of the site label wrt **x_off**
y_label	real	Vertical offset of the site label wrt **y_off**
x_off	real	Horizontal offset of table data wrt table location (in display units)
y_off	real	Vertical offset of table data wrt table location (in display units)
For lat/lon samples ...		
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
For map samples ...		
map_x	real	Position on x-axis of map (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units**)
map_y	real	Position on y-axis of map (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units**)
map_units	real	Number of meters (or degrees) per map unit (Default is **map_units** from @mapdef directive)

KEYWORD	TYPE	DESCRIPTION
For location samples ...		
`location_ident`	string	Identifier for location in `location_look_up` (Note that the keyword `location_look_up` in the @add @box @ellipse @sample_field @sample_wind @text directives identifies the location look up table to use)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`line_width`	real	Line width for boundary (in display units)
`line_style`	string	Line style (see @presentation)
`outline`	string	Line outline colour (see @presentation)
`fill`	string	Line fill colour (see @presentation)
`outline_first`	yes/no	Display outline before fill? (Note that default is to display fill first!)
`font`	string	Text font name recognized by output device or graphics post processor
`font_weight`	string	Text display One of: `none` `thin` `ultralight` `light` `normal` `medium` `demibold` `bold` `ultrabold` `black`
`italics`	yes/no	Text in italics?
`justification`	string	Text justification One of: `left` `centre` `right`
`char_space`	real	Text character spacing (as percent of average character width)
`word_space`	real	Text word spacing (as percent of average character width)

Example 2.93

```
@table_site
{
  table_name     = T1;
  site_label     = Halifax;
  lat            = 44:40N;
  lon            = 63:35W;
  x_label        = 0.15;
  y_label        = 0.15;
  x_off          = 0.0;
  y_off          = -0.60;
  justification = left;
}
```

Set a location for a site called "Halifax" at latitude 44 degrees and 40 minutes North and longitude 63 degrees and 35 minutes West in a table called "T1" (from the @define_table directive). Since the table is of type "column", all sampled data for this site will be displayed 0.60 "units" below the table location (from the @define_table directive). The site label will be displayed 0.15 "units" above and 0.15 "units" to the right of the initial data location (which is 0.60 "units" below the table location), and left justified. (Note that "units" is set by @display_units.)

2.5.61 @text

Display text (including text from a file) on the current map or in a table or at a given location.

KEYWORD	TYPE	DESCRIPTION
`text_file`	string	Full path for file containing text to display
`string`	string	Text to display (Note: use \ before all control characters)
`attribute`	string	"Magic" attribute containing value to display
`format`	string	Format of "magic" attribute to display Must use: **`direct`** ***`<cform>`*** (Note: ***`<cform>`*** is an optional string which contains a **printf** conversion format for the "magic" attribute. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.)
`text_size`	real	Text height for label or text found in look up table (in display units)
`rotation`	real	Text angle of rotation (in counter-clockwise decimal degrees)
`rotate_to_latitude`	yes/no	Rotate parallel to latitude. **`rotation`** is appended. (Default is no).
or		
`rotate_to_longitude`	yes/no	Rotate parallel to longitude. **`rotation`** is appended. (Default is no).
`x_off`	real	Horizontal offset of text wrt current anchor position (in display units)
`y_off`	real	Vertical offset of text wrt current anchor position (in display units)
For map display ...		
`lat`	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
`lon`	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
`map_x`	real	Position on x-axis of map (Note: position is measured from **`xmin`** from @mapdef directive, in units of **`map_units`**)
`map_y`	real	Position on y-axis of map (Note: position is measured from **`ymin`** from @mapdef directive, in units of **`map_units`**)
`map_units`	real	Number of meters (or degrees) per map unit (Default is **`map_units`** from @mapdef directive)
`location_ident`	string	Identifier for location in **`location_look_up`**

KEYWORD	TYPE	DESCRIPTION
For table display ...		
table_name	string	Table name (from @define_table) (Note: the object will be displayed once for each call to the @table_site directive)
For grid display ...		
grid_name	string	Grid name (from @define_sample_grid) (Note: the object will be displayed once for each location on the grid)
For list display ...		
list_name	string	List name (from @define_sample_list) (Note: the object will be displayed once for each location in the list)
For map, table or list display ...		
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (Default is current **valid_time** from @set_source directive)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
line_width	real	Line width for boundary (in display units)
line_style	string	Line style (see @presentation)
outline	string	Line outline colour (see @presentation)
fill	string	Line fill colour (see @presentation)
outline_first	yes/no	Display outline before fill? (Note that default is to display fill first!)
font	string	Text font name recognized by output device or graphics post processor
font_weight	string	Text display One of: **none thin ultralight light normal medium demibold bold ultrabold black**
italics	yes/no	Text in italics?
justification	string	Text justification One of: **left centre right**
char_space	real	Text character spacing (as percent of average character width)
word_space	real	Text word spacing (as percent of average character width)
line_space	real	Text line spacing (as percent of character height)

Example 2.94

```
@text { string = Tonight;
  y_off = 1.7; x_off = -0.1; justification = centre; }
```

Display the text string "Tonight" 1.7 "units" above and 0.1 "units" to the left of the current "anchor position" and centre justified.

Example 2.95

```
@text { text_file = $FPA/setup/pdf/psmet/examples/example_file;
  font = "Times-Italic"; italics = yes; }
```

Display text from the file `$FPA/setup/pdf/psmet/examples/example_file` at the current "anchor position". The text font will be set to "Times-Italic" style using italics. (Note that **`units`** is set by @display_units, and "anchor position" is set by @anchor.)

2.5.62 @vector_presentation

Set the appearance of vector direction and speed displayed as text or symbols.

Note that each vector component requires a "`..._type`" keyword to indicate the appropriate section in the **`vector_look_up`** file to use in formatting the vector component to be displayed.

KEYWORD	TYPE	DESCRIPTION
`vector_look_up`	string	File name for the vector look up table (Note: ".vtab" suffix is optional)
`calm_type`	string	Type of display for "calm" vector speed One of: **`none`** do not display **`value`** display as value **`text`** display matching text **`symbol`** display matching symbol
`calm_size`	real	Text height for "calm" vector speed (in display units)
`calm_scale`	real	Scale factor wrt original symbol for "calm" vector speed (in percent)
`calm_justification`	string	Justification for "calm" vector speed One of: **`left centre right`**
`calm_format`	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
`x_calm`	real	Horizontal offset for "calm" vector speed (in display units)
`y_calm`	real	Vertical offset for "calm" vector speed (in display units)
`direction_type`	string	Type of display for vector direction One of: **`none`** do not display **`value`** display as value **`text`** display matching text **`uniform`** display uniform symbol **`proportional`** display scaled symbol
`direction_size`	real	Text height for vector direction (in display units)
`direction_scale`	real	Scale factor wrt original symbol for vector direction (in percent)
`direction_justification`	string	Justification for vector direction One of: **`left centre right`**
`direction_format`	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
`x_dir`	real	Horizontal offset for vector direction (in display units)
`y_dir`	real	Vertical offset for vector direction (in display units)

KEYWORD	TYPE	DESCRIPTION
speed_type	string	Type of display for vector speed One of: **none** do not display **value** display as value **text** display matching text **symbol** display matching symbol
speed_size	real	Text height for vector speed (in display units)
speed_scale	real	Scale factor wrt original symbol for vector speed (in percent)
speed_justification	string	Justification for vector speed One of: **left centre right**
speed_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_spd	real	Horizontal offset for vector speed (in display units)
y_spd	real	Vertical offset for vector speed (in display units)

Example 2.96

```
@vector_presentation {
  vector_look_up = waves;
  calm_type = symbol; calm_justification = centre; calm_size = 0.20;
  calm_scale = 80; x_calm = 0; y_calm = 0;
  direction_type = uniform; direction_justification = centre;
  direction_size = 0.20; direction_scale = 100; x_dir = 0; y_dir = 0;
  speed_type = value; speed_justification = centre; speed_size = .2;
  speed_scale = 80; x_spd = 0; y_spd = -.1;
}
```

Display waves according to the formats in the vector look up table `waves.vtab`. Calm waves will be formatted according to the section "calm symbol", with symbols scaled to 80% or their original size and centre justified. Wave directions will be formatted according to the section "direction uniform", with symbols scaled to 100% of their original size and centre justified. Wave speeds will be formatted according to the section "speed value", with text characters 0.20 "units" high, offset 0.10 "units" below the current location, and centre justified. (Note that "units" is set by @display_units.)

2.5.63 @verbose

Set "verbose" mode for graphics product definition file development.

VALUE	DESCRIPTION
mode	One of: **yes** or **no**

Example 2.97

```
@verbose { yes }
```

Set the verbose mode on for extra diagnostic output. This can be useful if problems are encountered in developing a new graphics product.

Example 2.98

```
@verbose { no }
```

Set verbose mode off for minimal diagnostic output.

2.5.64 @version

Set the PSMet or SVGMet version number.

Note that this must be the first directive!

VALUE	DESCRIPTION
psmet8.0	Current version number. (Note: psmet6.0 is also accepted for backwards compatibility
svgmet8.0	Current version number. (Note: svgmet6.0 is also accepted for backwards compatibility

Example 2.99

```
@version { psmet8.0 }
```

Set PSMet version number. Note that this must be the first directive!

2.5.65 @wind_barb_presentation

Set the appearance of winds displayed as wind barbs.

KEYWORD	TYPE	DESCRIPTION
shaft_length	real	Length of wind barb shaft (in display units)
barb_length	real	Length of 10 unit and 50 unit wind barbs (as fraction of **shaft_length**)
barb_width	real	Width of base of 50 unit wind barb (as fraction of **shaft_length**)
barb_space	real	Space between wind barbs (as fraction of **shaft_length**)
barb_angle	real	Angle from wind shaft to wind barbs (in counter-clockwise decimal degrees) (Note: clockwise in southern hemisphere)
speed_round	real	Round speed to the closest multiple of this value.
gust_above	real	Gust speed must be larger than wind speed by this amount to be displayed (in units of wind speed)
gust_size	real	Gust speed height (in display units)
gust_distance	real	Distance from wind shaft to gust speed (as fraction of **shaft_length**)
gust_angle	real	Angle from wind shaft to gust speed (in counter-clockwise decimal degrees) (Note: clockwise in southern hemisphere)
gust_justification	string	Gust speed justification One of: **left centre right**
gust_format	string	A **printf** conversion format for gust speed, applied to the value before displaying (Note: use \ before all control characters)
gust_round	real	Round gusts to the closest multiple of this value.
calm_max	real	Maximum wind speed for "calm" conditions (in units of wind speed)
calm_symbol	string	Name of the graphic symbol file to display for "calm" conditions (Note that full path names are allowed. see Graphics Symbol Files, (Section 1.12))
calm_scale	real	Scale factor wrt original graphic (in percent)
huge_min	real	Minimum wind speed for "very large" winds (in units of wind speed)
huge_symbol	string	Name of the graphic symbol file to display for "very large" winds (Note that full path names are allowed. See Graphics Symbol Files, (Section 1.12) for default file locations)

KEYWORD	TYPE	DESCRIPTION
huge_scale	real	Scale factor wrt original graphic (in percent)

Example 2.100

```
@wind_barb_presentation
{
  shaft_length       = 0.40;
  barb_length        = 0.50;
  barb_width         = 0.25;
  barb_space         = 0.20;
  barb_angle         = 60;
  gust_above         = 5;
  gust_size          = 0.20;
  gust_distance      = 0.50;
  gust_angle         = 90;
  gust_justification = centre;
  gust_format        = G%.0f;
  calm_max           = 5;
  calm_symbol        = circle_mark;
  calm_scale         = 50;
  huge_min           = 100;
  huge_symbol        = circle_mark;
  huge_scale         = 100;
}
```

Display wind barbs with a shaft 0.40 "units" long, barbs 0.20 "units" long, 50 unit barbs 0.10 "units" wide at the base, and barbs spaced 0.08 "units" apart at an angle of 60 degrees to the wind shaft. Display wind gusts more than 5 units greater than wind speed with a G followed by a %.0f formatted value at a location 0.20 "units" and an angle of 90 degrees to the wind shaft. The formatted gust speed will be 0.20 "units" high and centre justified. Display "calm" winds (less than 5 units of wind speed) by a "circle_mark" 50% of its original size. Display "very large" winds (more than 100 units of wind speed) by a "circle_mark" at normal size. (Note that the default PSMet symbol would be `$FPA/setup/pdf/psmet/common/ps/circle_mark.ps`, and the default SVGMet symbol would be `$FPA/setup/pdf/svgmet/common/svg/circle_mark.svg`.) (Note that "units" is set by @display_units, and wind units are set by sampling/display directive.)

2.5.66 @wind_presentation

Set the appearance of wind direction, wind speed and wind gust components displayed as text or symbols on the current map or in a table.

Note that each wind component requires a "`..._type`" keyword to indicate the appropriate section in the **`wind_look_up`** file to use in formatting the wind component to be displayed.

KEYWORD	TYPE	DESCRIPTION
`wind_look_up`	string	File name for the wind look up table (Note: ".wtab" suffix is optional)
`calm_type`	string	Type of display for "calm" winds One of: **`none`** do not display **`value`** display as value **`text`** display matching text **`symbol`** display matching symbol
`calm_size`	real	Text height for "calm" wind (in display units)
`calm_scale`	real	Scale factor wrt original symbol for "calm" wind (in percent)
`calm_justification`	string	Justification for "calm" wind One of: **`left centre right`**
`calm_format`	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
`x_calm`	real	Horizontal offset for "calm" wind (in display units)
`y_calm`	real	Vertical offset for "calm" wind (in display units)
`direction_type`	string	Type of display for wind direction One of: **`none`** do not display **`value`** display as value **`text`** display matching text **`uniform`** display uniform symbol **`proportional`** display scaled symbol
`direction_size`	real	Text height for wind direction (in display units)
`direction_scale`	real	Scale factor wrt original symbol for wind direction (in percent)
`direction_justification`	string	Justification for wind direction One of: **`left centre right`**
`direction_format`	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
`x_dir`	real	Horizontal offset for wind direction (in display units)
`y_dir`	real	Vertical offset for wind direction (in display units)

KEYWORD	TYPE	DESCRIPTION
speed_type	string	Type of display for wind speed One of: **none** do not display **value** display as value **text** display matching text **symbol** display matching symbol
speed_size	real	Text height for wind speed (in display units)
speed_scale	real	Scale factor wrt original symbol for wind speed (in percent)
speed_justification	string	Justification for wind speed One of: **left centre right**
speed_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_spd	real	Horizontal offset for wind speed (in display units)
y_spd	real	Vertical offset for wind speed (in display units)
gust_type	string	Type of display for wind gust One of: **none** do not display **value** display as value **text** display matching text **symbol** display matching symbol
gust_above	real	Gust speed must be larger than wind speed by this amount to be displayed (in units of wind speed)
gust_size	real	Text height for wind gust (in display units)
gust_scale	real	Scale factor wrt original symbol for wind gust (in percent)
gust_justification	string	Justification for wind gust One of: **left centre right**
gust_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_gust	real	Horizontal offset for wind gust (in display units)
y_gust	real	Vertical offset for wind gust (in display units)

Example 2.101

```
@wind_presentation
{
  wind_look_up            = winds;
  calm_type               = none;
  gust_type               = none;
  direction_type          = value;
  direction_justification = left;
  direction_size          = 0.20;
  direction_scale         = 80;
  x_dir                   = 0.05;
  y_dir                   = 0.05;
  speed_type              = value;
  speed_justification     = left;
  speed_size              = 0.20;
  speed_scale             = 80;
  x_spd                   = 0.05;
  y_spd                   = -0.20;
}
```

Display winds according to the formats in the wind look up table `winds.wtab`. Calm winds and wind gusts will not be displayed. Wind directions will be formatted according to the section "direction value", with text characters 0.20 "units" high, offset 0.05 "units" to the right of and 0.05 "units" above the current location, and left justified. Wind speeds will be formatted according to the section "speed value", with text characters 0.20 "units" high, offset 0.05 "units" to the right of and 0.20 "units" below the current location, and left justified. (Note that "units" is set by @display_units.)

2.5.67 @write_time

Display current clock, issue or valid time as formatted text on the current map.

Note that the issue time is the **T0** time from the FPA, while the valid time is the current valid time from the @set_source directive.

Note that **type**, **zone_type** and **format** are always required.

KEYWORD	TYPE	DESCRIPTION
text_size	real	Text height (in display units)
rotation	real	Text angle rotation (in counter-clockwise decimal degrees)
x_off	real	Horizontal offset of text wrt current anchor position
y_off	real	Vertical offset of text wrt current anchor position

Note that **type**, **zone_type**, **time_zone**, **language** and **format** can be specified more than once in this directive, and the resulting formatted strings are concatenated before display.

type	string	Type of time to display One of: **creation** ***<tadj>*** creation time **issue** ***<tadj>*** **T0** time **valid** ***<tadj>*** valid time (Note: ***<tadj>*** is an optional adjustment to the time to display, in ***hh*** (hours) or ***hh:mm*** (hours:minutes) format.)
zone_type	string	Type of time zone to display One of: **GMT** or **UTC** Coordinated Universal time **LMT** or **LCL** Local time
time_zone	string	Display the equivalent time in another time zone. (Default is set by environment variable **$TZ**) See the directory `/usr/share/zoneinfo` for valid timezones.
language	string	Display time parameters in another language. (Default is set by environment variable **$LANG**) See the file `/usr/share/locale/locale.alias` for language options.

KEYWORD	TYPE	DESCRIPTION
`format`	string	Format for time to display Note that replacement characters from the system function **strftime** are allowed as part of the format. These include: **`%a`** abbreviated weekday name **`%A`** full weekday name **`%b`** abbreviated month name **`%B`** full month name **`%c`** local date and time **`%d`** day of the month (01-31) **`%H`** hour (24 hour clock)(00-23) **`%I`** hour (12 hour clock)(1-12) **`%j`** day of the year (001-366) **`%m`** month (01-12) **`%M`** minute (00-59) **`%n`** for new line **`%p`** local equivalent of AM/PM (Note: midnight is AM, noon is PM) **`%P`** local equivalent of am/pm **`%t`** for tab space **`%u`** weekday (1-7, Monday is 1) **`%w`** weekday (0-6, Sunday is 0) **`%y`** year without century (00-99) **`%Y`** year with century **`%Z`** time zone name, if any **`%%`** for "%" (Note: use \ before all control characters)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`line_width`	real	Line width for boundary (in display units)
`line_style`	string	Line style (see @presentation)
`outline`	string	Line outline colour (see @presentation)
`fill`	string	Line fill colour (see @presentation)
`outline_first`	yes/no	Display outline before fill? (Note that default is to display fill first!)
`font`	string	Text font name recognized by output device or graphics post processor
`font_weight`	string	Text display One of: **`none`** **`thin`** **`ultralight`** **`light`** **`normal`** **`medium`** **`demibold`** **`bold`** **`ultrabold`** **`black`**
`italics`	yes/no	Text in italics?
`justification`	string	Text justification One of: **`left`** **`centre`** **`right`**
`char_space`	real	Text character spacing (as percent of average character width)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
word_space	real	Text word spacing (as percent of average character width)
line_space	real	Text line spacing (as percent of character height)

Example 2.102

```
@write_time
{
  type      = valid;
  zone_type = LMT;
  time_zone = EST5EDT;
  format    = Map Valid %A, %B %d, %Y %H:%M %Z;
  text_size = .25;
  justification = left;
  x_off     = 0;
  y_off     = .15;
}
```

Display valid time for the current graphics product in the Eastern time zone, displayed 0.25 "units" high, offset 0.15 "units" above the current "anchor position", and left justified. This particular **format** might produce, for example:

```
Map Valid Saturday, May 10, 2008 13:00 EDT
```

(Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

Example 2.103

```
@write_time
{
  type      = valid;
  format    = "CHART VALID %b%d/%Y %H:%M";
  type      = valid +12;
  format    = " - %b%d/%Y %H:%M UTC;
  zone_type = UTC;
  text_size = .25;
  justification = centre;
}
```

Display valid time range (from current valid time to 12 hours from current valid time) for the current graphics product in Coordinated Universal Time, displayed 0.25 "units" high at the current "anchor position", and centre justified. This particular **format** might produce, for example:

```
CHART VALID May10/2008 13:00 - May11/2008 01:00 UTC
```

(Note that "units" is set by @display_units, and "anchor position" is set by @anchor.)

2.5.68 Format of Look Up Tables

Look up tables allow conversion of FPA style attribute values or sampled values into symbols, text, or a presentation for display. They are accessed by the **look_up**, **case_look_up**, or **attribute_look_up keywords** in the @areas @cross_section_axis_labels @lines @label @sample_field directives. Files containing look up tables have a ".tab" suffix.

Look up tables contain lines with the following format:

```
    key(s)  :  symbol  :  text  :  presentation
    key(s)  :  symbol  :  text  :  presentation
    ...
```

where:

key(s) are one or more "value" strings separated by "white space" (such as blanks or tabs)

symbol is a graphic symbol file to display

text is text to display (quotes not required)

presentation is a named presentation (identified by the "name" keyword in the @presentation directive)

Note that one of symbol, text, or presentation must be non-blank, but that the others can be left blank if not required. Note also that lines without the **:** separators are considered comments, as are all parameters that follow the **!** or **#** characters in a line.

When an FPA field is sampled by one of the directives, each of the ***key(s)*** is tested against the resultant FPA "value" string. Each time one ***key*** is found within the FPA "value" string, one "hit" is registered, and the "best" match in the look up table is defined as the first line in the file where the maximum number of "hits" has been found. The symbol or text or presentation from that line would then used by the directive for displaying the sampled value. Two "special" ***key(s)*** are allowed.

The ***key*** ***missing*** will match against blank FPA "value" strings.
The ***key*** ***default*** will match against FPA "value" strings that register no "hits" in the look up table.

The type of display to use depends on each directive.

The @areas directive uses the named presentation given by presentation to display the outline and interior of each "discrete" area on the map.

The @lines directive uses the named presentation given by presentation to display the patterned line of each "line" feature on the map.

The @cross_section_axis_labels @label @sample_field directives display the values sampled at each location with a graphic symbol from the file symbol (if the keyword **format** is set to symbol) or with a text string given by text (if the keyword **format** is set to text).

The order of the lines in the look up table is extremely important. Lines with the most complicated "value" strings should always be included before lines with simpler "value" strings, and all lines with a single ***key*** should be included before lines with two ***key(s)***, and lines with two ***key(s)*** before lines with three ***key(s)***, and so on.

When several lines register the same number of "hits", the line that appears first in the look up table will be used. For example, the ***key(s)*** for rain "R" and light rainshowers "RW-" will both register one "hit" against the FPA "value" string for reduced visibility in light rainshowers "1-3RW-", since each **`key`** is found within the FPA "value" string. However, if the line for rain appears in the look up table before the line for light rainshowers, then the line for rain will ALWAYS be chosen first, and any special presentation for light rainshowers will never be accessed. Therefore, complicated "value" strings should always be listed before simple "value" strings.

The number of **`key(s)`** in a line does not affect the choice of a "best" match. One "hit" on a line with one **`key`** is equivalent to one "hit" on a line with several **`key(s)`**. For example, a line containing the **`key(s)`** for rain and snow "R S" and a line containing the **`key`** for rain alone "R" will both register one "hit" against the FPA "value" string for light rain "R-". However, if the line containing the **`key(s)`** for rain and snow appears in the look up table before the line containing the **`key`** for rain alone, then the line containing the **`key(s)`** for rain and snow will ALWAYS be chosen first, and the corresponding presentation for mixed rain and snow will be used. Therefore, lines containing fewer **`key(s)`** should always be listed first.

Care must also be taken to ensure that the combination of look up table **`key(s)`** and FPA "value" strings does not register incorrect "hits". For example, a line containing the **`key`** for snow "S" will register a "hit" against the FPA "value" string for risk of a thunderstorm "RISK TRW". In this case, the "hit" occurs because of the "S" in "RISK"! It may be necessary to constrain the use of some "value" strings within the FPA to avoid generating unwanted matches in the Graphics Product Generator applications.

The following example of a look up table and examples of matches with FPA "value" strings is based on look up tables that are no longer used, but the basic principles are still applicable.

Example 2.104 Look up file format

```
1 TRW           : thunder : Thunderstorm   : trw1
2 RW            : rain    : Showers        : shower1
3 R             : rain    : Rain           : rain1
4 S             : snow    : Snow           : snow1
5 RSF           : rs_mix  : Rain and Snow  : mix1
6 RISK TRW      : thunder : Risk Thunder   : trw2
7 RW RISK TRW+  : rain    : Rain / Thunder : rain2
8 RW RISK TRW   : rain    : Rain / Thunder : rain2
```

Example 2.105 FPA "value" string to match "RW"

One "hit" will be registered by **`key(s)`** in lines 2,3,5,7,8. Therefore, line 2 will be used.
If a symbol is required, the symbol file "rain" will be displayed.
If text is required, the text string "Showers" will be displayed.
If a "discrete" area is to be displayed, the named presentation "shower1" will be used.

Example 2.106 FPA "value" string to match "RW-F"

One "hit" will be registered by **`key(s)`** in lines 2,3,7,8, and two hits by **`key(s)`** in line 5. Therefore, line 5 will be used.
If a symbol is required, the symbol file "rs_mix" will be displayed.
If text is required, the text string "Rain and Snow" will be displayed.
If a "discrete" area is to be displayed, the named presentation "mix1" will be used.
(Note: an additional line with the **`key(s)`** "R F" or "RW F" may be necessary.)

Example 2.107 FPA "value" string to match "RISK TRW+"

One "hit" will be registered by ***key(s)*** in lines 1,2,3,4, and two hits by ***key(s)*** in lines 5,6,7,8. Therefore, line 5 will be used.
If a symbol is required, the symbol file "rs_mix" will be displayed.
If text is required, the text string "Rain and Snow" will be displayed.
If a "discrete" area is to be displayed, the named presentation "mix1" will be used.
(Note: line 5 should be relocated at the end of the look up table to avoid this error!)

2.5.69 Format of Location Look Up Tables

Location look up tables allow for fixed or moving locations to be identified in tabular form, for easy access by the PSMet or SVGMet; directives. They are accessed by the **location_look_up** keyword in the @add @anchor @box @cross_section_areas @cross_section_axis_labels @cross_section_contours @cross_section_curves @define_cross_section @draw_cross_section_line @ellipse @sample_field @sample_wind @text directives. Files containing location look up tables have a ".ltab" suffix.

Location look up tables contain lines with the following format:

```
    ident    lat lon    [valid_time]    [text_string]
    ident    lat lon    [valid_time]    [text_string]
    ...
```

where:

ident	is the location identifier (in string format)
lat	is the location latitude (in latlon format)
lon	is the location longitude (in latlon format)
valid_time	is an optional valid time (in vtime format)
text_string	is an optional text description (in string format)

Note that the **-** character can be used as a placeholder for any "missing" parameters.

Note that all parameters that follow the **!** or **#** characters in a line are considered comments.

The **location_ident** keyword from the directive is matched with the ***ident*** from a line of the location look up table to set the appropriate latitude and longitude for display or sampling. The optional ***valid_time*** is used to specify a moving location. If valid_time is included, then the line in the location look up table must also match the current valid time set by the **valid_time** keyword in the directive, or the valid time previously set in the @set_source directive. One "special" ident value is allowed. The ident ***default*** will match against any location identifier not found in the look up table.

2.5.70 Format of Vertical Look Up Tables

Vertical look up tables are used to set vertical positions in cross sections. They are accessed by the **vertical_look_up** keyword which can be found in the @cross_section_areas @cross_section_contours @cross_section_axis_labels @define_cross_section @draw_cross_section_line @sample_field @sample_wind directives. Files containing vertical look up tables have a ".ztab" suffix.

Vertical look up tables contain lines with the following format:

```
ident   value axis_location    [text_string]
ident   value axis_location    [text_string]
...
```

where:

ident is the location identifier (in string format)

value is a vertical value for the level (in real format)

axis_location is a percentage of cross section height, from the "height" keyword in the @define_cross_section directive (in real format)

text_string is an optional text description (in string format)

Note that the **-** character can be used as a placeholder for any "missing" parameters.

Note that all parameters that follow the **!** or **#** characters in a line are considered comments. The vertical parameters from the directive are matched with the set of ***value*** parameters from the lines of the vertical look up table to set the appropriate location along the vertical axis of the cross section, given by the set of ***axis_location*** parameters, for display or sampling.

2.5.71 Format of Wind Look Up Tables

Wind look up tables control the format for displaying wind direction, speed, and gusts as symbols or text, rather than as wind barbs. They are accessed by the **wind_look_up** keyword in the @wind_presentation directive. Files containing wind look up tables have a ".wtab" suffix.

Wind look up tables contain four sections. One section must specify the "calm" condition, since this may override both the direction and speed displays. The other sections specify the "direction" or "speed" or "gust" formats to use when displaying winds. Note that all wind directions are in degrees true, and all wind speeds gusts are converted to the units given by the "units" keyword of the sampling or display directive.

Note that all parameters that follow the **!** or **#** characters in a line are considered comments.

Note that the examples are taken from wind look up tables in the directories `$FPA/setup/pdf/psmet/examples` or `$FPA/setup/pdf/svgmet/examples`.

The "calm" section can include the following three formats:

```
calm    value
  min_spd max_spd round factor sdigits
  end

calm    text
  min_spd max_spd text
  end

calm    symbol
  min_spd max_spd symbol
  end
```

where:

min_spd is the minimum wind speed for "calm" conditions
max_spd is the maximum wind speed for "calm" conditions
round is the value to round the wind speed to the nearest multiple of
factor is the power of 10 to divide the wind speed by
sdigits is the significant digits of the wind speed to display
 0 - for the nearest integer
 >0 - for the nearest integer this long (with leading zeroes)
 <0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)
symbol is the symbol file to use

Example 2.108

```
calm value
    0 5 1 0 0
    end
```

Wind speeds from 0 to 5 are rounded to the nearest whole value, and all significant digits are displayed.

Example 2.109

```
calm text
    0 5 "CALM"
    end
```

Wind speeds from 0 to 5 are displayed using the text string "CALM".

Example 2.110

```
calm symbol
    0 5 wnd_calm
    end
```

Wind speeds from 0 to 5 are displayed using the symbol "wnd_calm".

The "direction" section can include the following four formats:

```
direction value
    min_spd max_spd round factor sdigits
    end

direction text
    min_dir max_dir text
    min_dir max_dir text
    ...
    end

direction symbol uniform
    min_dir max_dir symbol rotate
    min_dir max_dir symbol rotate
    ...
    end

direction symbol proportional
    min_spd max_spd symbol min_scale max_scale
    min_spd max_spd symbol min_scale max_scale
    ...
    end
```

where:

min_spd is the minimum wind speed

max_spd is the maximum wind speed

round is the value to round wind direction to the nearest multiple of

factor is the power of 10 to divide the wind direction by

sdigits is the significant digits of the wind direction to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place

min_dir is the minimum wind direction angle

max_dir is the maximum wind direction angle

text is the text to display (in double quotes)

symbol is the symbol file to use

rotate is the wind direction for displaying the "symbol" (Note: -1 will use the actual wind direction)

min_scale is a scale factor wrt original symbol (in percent) for min_spd

max_scale is a scale factor wrt original symbol (in percent) for max_spd

Example 2.111

```
direction value
    5 999 10 0 3
    end
```

Wind directions for wind speeds from 5 to 999 are rounded to the nearest 10 degrees, and displayed as the nearest integer 3 digits long. (That is, a wind direction of "53" degrees would be rounded to "50", and displayed as "050".)

Example 2.112

```
 direction text
    337.5  22.5 "N"
     22.5  67.5 "NE"
     67.5 112.5 "E"
    end
```

Wind directions from 337.5 to 22.5 degrees are displayed using the text string "N", wind directions from 22.5 to 67.5 degrees are displayed using the text string "NE", and wind directions from 67.5 to 112.5 degrees are displayed using the text string "E".

Note that a wind direction of -22.5 is equivalent to a wind direction of 337.5!

Example 2.113

```
direction symbol uniform
    22.5  67.5 wnd_arrow 45
    67.5 112.5 wnd_arrow 90
    end
```

Wind directions from 22.5 to 67.5 degrees are displayed using the symbol "wnd_arrow" rotated to 45 degrees, and wind directions from 67.5 to 112.5 degrees are displayed using the symbol "wnd_arrow" rotated to 90 degrees.

Example 2.114

```
direction symbol uniform
    0 360 wnd_arrow -1
    end
```

Wind directions from 0 to 360 degrees are displayed using the symbol "wnd_arrow" rotated to the actual wind direction.

Example 2.115

```
direction symbol proportional
     5  50 wnd_arrow  20 200
    50 100 wnd_arrow 200 500
    end
```

Wind directions for wind speeds from 5 to 50 are displayed using the symbol "wnd_arrow" rotated to the actual wind direction and scaled to between 20% and 200% of the original size of the graphic (depending on the wind speed), and wind directions for wind speeds from 50 to 100 degrees are displayed using the symbol "wnd_arrow" rotated to the actual wind direction and scaled to between 200% and 500% of the original size of the graphic (depending on the wind speed).

The "speed" section can include the following three formats:

```
speed value
    min_spd max_spd round factor sdigits
    end

speed text
    min_spd max_spd text
    min_spd max_spd text
    ...
    end

speed symbol
    min_spd max_spd symbol
    min_spd max_spd symbol
    ...
    end
```

where:

min_spd is the minimum wind speed
max_spd is the maximum wind speed
round is the value to round the wind speed to the nearest multiple of
factor is the power of 10 to divide the wind speed by
sdigits is the significant digits of the wind speed to display
  0 - for the nearest integer
  >0 - for the nearest integer this long (with leading zeroes)
  <0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)
symbol is the symbol file to use

Example 2.116

```
speed value
    5 999 5 0 0
    end
```

Wind speeds from 5 to 999 are rounded to the nearest 5, and the nearest integer wind speed is displayed.

Example 2.117

```
speed text
     5 15 "LIGHT"
    15 30 "STRONG"
    end
```

Wind speeds from 5 to 15 are displayed using the text string "LIGHT", and wind speeds from 15 to 30 are displayed using the text string "STRONG".

Example 2.118

```
speed symbol
     5 15 wnd_lgt
    15 30 wnd_str
    end
```

Wind speeds from 5 to 15 are displayed using the symbol "wnd_lgt", and wind speeds from 15 to 30 are displayed using the symbol "wnd_str".

The "gust" section can include the following three formats:

```
gust value
    min_gst max_gst round factor sdigits
    end

gust text
    min_gst max_gst text
    min_gst max_gst text
    ...
    end

gust symbol
    min_gst max_gst symbol
    min_gst max_gst symbol
    ...
    end
```

where:

min_gst is the minimum wind gust

max_gst is the maximum wind gust

round	is the value to round the wind gust to the nearest multiple of
factor	is the power of 10 to divide the wind gust by
sdigits	is the significant digits of the wind gust to display 0 - for the nearest integer >0 - for the nearest integer this long (with leading zeroes) <0 - for a real number with digits after the decimal place
text	is the text to display (in double quotes)
symbol	is the symbol file to use

Example 2.119

```
gust value
    5 999 5 0 0
    end
```

Wind gusts from 5 to 999 are rounded to the nearest 5, and the nearest integer wind gust is displayed.

Example 2.120

```
gust text
    15  30 "STRONG GUSTS"
    30 999 "VERY STRONG GUSTS"
    end
```

Wind gusts from 15 to 30 are displayed using the text string "STRONG GUSTS", and wind gusts from 30 to 999 are displayed using the text string "VERY STRONG GUSTS".

Example 2.121

```
gust symbol
    15  30 wnd_str
    30 999 wnd_gst
    end
```

Wind gusts from 15 to 30 are displayed using the symbol "wnd_str", and wind gusts from 30 to 999 are displayed using the symbol "wnd_gst".

2.5.72 Format of Vector Look Up Tables

Vector look up tables control the format for displaying vector speed and direction as symbols or text. They are set by the **vector_look_up** keyword in the @vector_presentation directive. Files containing vector look up tables have a ".vtab" suffix.

Vector look up tables contain three sections. One section must specify the "calm" condition, since this may override both the direction and speed displays. The other sections specify the "direction" or "speed" formats to use when displaying vector parameters.

Note that all parameters that follow the ! or # characters in a line are considered comments.

Note that examples are taken from vector look up tables in the directories `$FPA/setup/pdf/psmet/examples` or `$FPA/setup/pdf/svgmet/examples`.

The "calm" section can include the following three formats:

```
calm value
    min_spd max_spd round factor sdigits
    end
calm text
    min_spd max_spd text
    end
calm symbol
    min_spd max_spd symbol
    end
```

where:

min_spd is the minimum vector speed for "calm" conditions
max_spd is the maximum vector speed for "calm" conditions
round is the value to round the speed to the nearest multiple of
factor is the power of 10 to divide the vector speed by
sdigits is the significant digits of the vector speed to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)
symbol is the symbol file to use

Example 2.122

```
calm value
    0 5 1 0 0
      end
```

Vector speeds from 0 to 5 are rounded to the nearest whole value, and all significant digits are displayed.

Example 2.123

```
calm text
    0 5 "CALM"
    end
```

Vector speeds from 0 to 5 are displayed using the text string "CALM".

The "direction" section can include the following four formats:

```
direction value
    min_spd max_spd round factor sdigits
    end

direction text
    mid_dir max_dir text
    mid_dir max_dir text
    ...
    end

direction symbol uniform
    mid_dir max_dir symbol rotate
    mid_dir max_dir symbol rotate
    ...
    end

direction symbol uniform
    mid_dir max_dir symbol min_scale max_scale
    mid_dir max_dir symbol min_scale max_scale
    ...
    end
```

where:

min_spd is the minimum vector speed
max_spd is the maximum vector speed
round is the value to round the speed to the nearest multiple of
factor is the power of 10 to divide the vector direction by
sdigits is the significant digits of the vector direction to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)
symbol is the symbol file to use
rotate is the vector direction for displaying the "symbol". (Note: -1 will use the actual vector direction)
min_scale is a scale factor wrt original symbol (in percent) for min_spd
max_scale is a scale factor wrt original symbol (in percent) for max_spd

Example 2.124

```
direction value
    5 999 10 0 3
    end
```

Vector directions for vector speeds from 5 to 999 are rounded to the nearest 10 degrees, and displayed as the nearest integer 3 digits long. (That is, a vector direction of "53" degrees would be rounded to "50", and displayed as "050".)

Example 2.125

```
direction text
    337.5 22.5 "N"
    22.5  67.5 "NE"
    67.5 112.5 "E"
    end
```

Vector directions from 337.5 to 22.5 degrees are displayed using the text string "N", vector directions from 22.5 to 67.5 degrees are displayed using the text string "NE", and vector directions from 67.5 to 112.5 degrees are displayed using the text string "E".
Note that a vector direction of -22.5 is equivalent to a vector direction of 337.5!

Example 2.126

```
direction symbol uniform
    22.5 67.5  arrow_from 45
    67.5 112.5 arrow_from 90
    end
```

Vector directions from 22.5 to 67.5 degrees are displayed using the symbol "arrow_from" rotated to 45 degrees, and vector directions from 67.5 to 112.5 degrees are displayed using the symbol "arrow_from" rotated to 90 degrees.

Example 2.127

```
direction symbol uniform
    0 360 arrow_from -1
    end
```

Vector directions from 0 to 360 degrees are displayed using the symbol "arrow_from" rotated to the actual vector direction.

Example 2.128

```
direction symbol proportional
     1  10 wave  20  200
    10 100 waved 200 500
    end
```

Vector directions for vector speeds from 5 to 50 are displayed using the symbol "wave" rotated to the actual vector direction and scaled to between 20% and 200% of the original size of the graphic (depending on the vector speed), and vector directions for vector speeds from 50 to 100 degrees are displayed using the symbol "wave" rotated to the actual vector direction and scaled to between 200% and 500% of the original size of the graphic (depending on the vector speed).

The "speed" section can include the following three formats:

```
speed value
    min_spd max_spd round factor sdigits
    end

speed text
    min_spd max_spd text
    min_spd max_spd text
    ...
    end

speed symbol
    min_spd max_spd symbol
    min_spd max_spd symbol
    ...
    end
```

where:

min_spd is the minimum vector speed
max_spd is the maximum vector speed
round is the value to round the speed to the nearest multiple of
factor is the power of 10 to divide the vector speed by
sdigits is the significant digits of the vector speed to display
  0 - for the nearest integer
  >0 - for the nearest integer this long (with leading zeroes)
  <0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)
symbol is the symbol file to use

Example 2.129

```
speed value
    1 999 1 0 0
    end
```

Vector speeds from 1 to 999 are rounded to the nearest 1, and the nearest integer vector speed is displayed.

Example 2.130

```
speed text
     1       5    "LIGHT"
     5      10    "MODERATE"
    10     999    "HEAVY"
    end
```

Vector speeds from 1 to 5 are displayed using the text string "LIGHT", vector speeds from 5 to 10 are displayed using the text string "MODERATE", and vector speeds from 10 to 999 are displayed using the text string "HEAVY".

Example 2.131

```
speed symbol
     1      5    wave
     5     10    wave
    10    999    wave
    end
```

Vector speeds from 1 to 999 are displayed using the symbol "wave".

Chapter 3

TexMet

The TexMet application allows the user to develop ASCII file products using data from the FPA. The ASCII products can be customized to sample various types of data and display them as values, text, or tables. This section describes the TexMet files and commands in more detail.

3.1 TexMet Input and Output Files

The TexMet application uses the following input files:

Product Definition Files (fpdf)
TexMet directives (commands) for sampling and displaying FPA data or other information in a tabular format. These files end with the suffix .fpdf

FPA Metafiles
containing FPA data for sampling or display. (Note: these files are accessed automatically by the directives)

Look Up Table Files (tab)
Cross reference tables for display formats for values. These files end with the suffix .tab

Location Look Up Table Files (ltab)
Cross reference tables for set locations (with optional times) These files end with the suffix .ltab

Wind Look Up Table Files (wtab)
Cross reference tables for display formats for winds These files end with the suffix .wtab

Vector Look Up Table Files (vtab)
Cross reference tables for display formats for vector parameters These files end with the suffix .vtab

The TexMet application produces the following output file(s):

ASCII Output File (txt)
ASCII file containing sampled FPA data

3.2 Special TexMet Directives

The following directives have special locations or usages in the TexMet product definition files. A complete description of each of these directives is given in a following section.

The @version directive is always found at the beginning of the main TexMet product definition file; it must always be the first directive encountered! The @file_name directive identifies the output file to be used. This directive is normally found at the beginning of the main product definition file too, though it may also be used later in the product definition files to open another output file. The @file_close directive is used to close an output file.

@version
TexMet version number. This must be the first directive!

@file_name
Name and location of TexMet output file

@file_close
Close the TexMet output file

The @verbose directive can be used while developing new TexMet products. It will provide extra diagnostic output to help determine why a particular directive does not produce the desired output.

@verbose
Provide extra diagnostic output during development

The @process directive can be used to launch an external process from within TexMet. For example, this directive could be used to display the TexMet output file.

@process
Launch an external process

The following directives can be used to simplify the directive and keyword commands in a set of product definition files. The @group @include directives are used to combine keywords or directives. The @loop_begin @loop_end directives allow a given sequence of directives to be repeated several times with changing keyword values.

@group
Combine a series of repeated keywords

@include
Combine a series of directives

@loop_begin
Beginning of loop for repeating directives with changing keyword values

@loop_end
End of loop for repeating directives

@loop_location_look_up
Create a temporary location look up table from a point, line or link chain feature.

The following directives control positioning in the output file. The @size directive is used to set the actual size of the output file. The @anchor directive is used for defining positions in the output file. This directive can be reset any number of times.

@size
Size of the ASCII output file. This must occur before any display directives.

@anchor
Define a reference position for displaying parameters

The @projection @mapdef @resolution directives control the map background used for sampling. (Note that any background map should be completely contained by the map defined in the FPA setup file and the FPA metafiles, or errors could result when sampling the data. Errors can also occur if the locations for sampling data are beyond the limits of the background map defined by these directives.) These directives are usually found at the beginning of TexMet product definition files.

@projection
Map projection for sampling

@mapdef
Map definition for sampling

@resolution
Spline resolution for sampling

The @presentation directive sets the default display parameters for all subsequent directives. Many of the keywords in TexMet have default values that are set automatically when the application is run. These parameters are reset by keywords in the @presentation directive. However, many of the keywords found in the @presentation directive are also found in other directives. The effect of such keywords is to override the default display parameters for that directive only; the default display parameters will not be changed.

@presentation
Default display parameters

3.3 TexMet Display Directives

The following directives control the display of all parameters in TexMet. A complete description of each of these directives is given in a following section. The following directives are used to set special parameters used in display of various ASCII values:

@set_source
Set a time for extracting FPA Metafile data

@define_table
Define a table for sampling data

@table_site
Set locations in a table

@define_sample_grid
Define a grid of locations for sampling values

@define_sample_list
Define a list of locations for sampling values

The following directives are used to set parameters to control the format of ASCII values to be displayed:

@wind_presentation
Set parameters for displaying winds as text

@vector_presentation
Set parameters for displaying vector field parameters as text strings

The following directives are used to display ASCII text:

@text
Display a text string (or file)

@write_time
Display a formatted time string

The following directives are used to display ASCII values extracted from FPA Metafiles at an arbitrary location or in a table:

@label
Display labels from FPA fields

@lchain_nodes
Display nodes from FPA link chain fields.

@sample_field
Sample values from FPA fields

@sample_wind
Sample winds from FPA fields

3.4 Complete Description of All TexMet Directives

The following table consists of an alphabetical listing of all the TexMet directives, a list of keywords that can be used within each directive, a brief description of the directives and keywords, and one or more examples of each directive.

Note that most examples are found in product definition files in directory `$FPA/setup/pdf/texmet/examples`, and other examples are included to demonstrate additional functionality.

3.4.1 @anchor

Set a new reference location, or anchor position, for subsequent displays in the ASCII output file.

Note that the following directives are affected by the anchor position: @define_table @label @lchain_nodes @sample_field @sample_wind @text @write_time

Note, however, that if the directive contains a **table_name** keyword, the anchor position for that directive defaults to the table location set in @define_table.

KEYWORD	TYPE	DESCRIPTION
ref	string	Starting point for new reference location One of: **absolute** first column - first line **lower_left** first column - last line **centre_left** first column - centre line **upper_left** first column - first line **lower_centre** centre column - last line **centre** centre column - centre line **upper_centre** centre column - first line **lower_right** last column - last line **centre_right** last column - centre line **upper_right** last column - first line **current** wrt current anchor (default starting point is **absolute**)
column	integer	Column (character) offset from starting point
row	integer	Row (line) offset from starting point

Example 3.1

```
@anchor { ref = absolute; column = 5; row = 4; }
```

Set a new reference location offset 5 columns (characters) to the right of the first column and 4 rows (lines) below the first line of the ASCII output file.

3.4.2 @define_sample_grid

Define a grid of locations for sampling.

Grids can be either an array of latitude-longitude locations, or an array of map positions on the current map.

KEYWORD	TYPE	DESCRIPTION
grid_name	string	Grid name
For latlon grid ...		
lat_begin	latlon	Latitude to begin grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lat_end	latlon	Latitude to end grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lat_interval	latlon	Interval between latitudes in grid (see Format of Latitudes and Longitudes, (Section 1.3))
lon_begin	latlon	Longitude to begin grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lon_end	latlon	Longitude to end grid of positions (see Format of Latitudes and Longitudes, (Section 1.3))
lon_interval	latlon	Interval between longitudes in grid (see Format of Latitudes and Longitudes, (Section 1.3))
For map grid ...		
map_x_begin	real	Position on x-axis of map to begin grid (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units** below)
map_x_end	real	Position on x-axis of map to end grid (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units** below)
map_x_interval	real	Interval between grid positions on x-axis of map (in units of **map_units** below)
map_y_begin	real	Position on y-axis of map to begin grid (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units** below)
map_y_end	real	Position on y-axis of map to end grid (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units** below)
map_y_interval	real	Interval between grid positions on y-axis of map (in units of **map_units** below)
map_units	real	Number of meters (or degrees) per map unit (default is **map_units** from @mapdef directive)

KEYWORD	TYPE	DESCRIPTION
x_shift	integer	Column (character) offset of each sample location
y_shift	integer	Row (line) offset of each sample location

Example 3.2

```
@define_sample_grid
{
  grid_name    = grid_waves;
  lat_begin    = 22.5S;
  lat_end      = 60.0N;
  lat_interval = 5.0N;
  lon_begin    = 92.5W;
  lon_end      = 20.0W;
  lon_interval = 5.0E;
  x_shift      = 5;
  y_shift      = -1;
}
```

Define a latitude-longitude grid for sampling called "grid_waves". The grid will begin at latitude 22.5 degrees South and end at latitude 60.0 degrees North with sample positions every 5.0 degrees North. The grid will begin at longitude 92.5 degrees West and end at longitude 20.0 degrees West with sample positions every 5.0 degrees East. The sampling grid will therefore have 17 rows of latitudes by 15 columns of longitudes. Each row will be offset up 1 line, and each column will be offset to the right by 5 characters.

3.4.3 @define_sample_list

Define a list of positions for sampling.

Lists can contain latitude-longitude locations, map positions on the current map, or identifiers for a location look up table.

KEYWORD	TYPE	DESCRIPTION
`list_name`	string	List name

For latlon positions ...
Note that `lat_lon_ident` can be specified more than once in this directive.

`lat_lon_ident`	l l [s]	Latitude, longitude and optional label for sample location (see Format of Latitudes and Longitudes, (Section 1.3))

For map positions ...
Note that `map_x_y_ident` and `map_units` can be specified more than once in this directive.

`map_x_y_ident`	r r [s]	Position on x-axis and y-axis of map, and optional label for sample location (Note: positions are measured from `xmin` and `ymin` from @mapdef directive, in units of `map_units` below)
`map_units`	real	Number of meters (or degrees) per map unit (default is `map_units` from @mapdef directive)

For location idents ...
Note that `location_ident` and `location_ident_list` can be specified more than once in this directive.

`location_ident`	string	Identifier for location in `location_look_up` (Note: the location look_up table will be defined in a sampling directive) (Note: the magic parameters `GPGEN_All` and `GPGEN_All_vtime` can be used to display ALL locations in a location look up table)
`location_ident_list`	string(s)	List of identifiers in a location look up table (Note: the location look_up table will be defined in a sampling directive)

`x_shift`	integer	Column (character) offset of each sample location
`y_shift`	integer	Row (line) offset of each sample location

KEYWORD	TYPE	DESCRIPTION
x_wrap	integer	Apply **x_shift** for this many values before resetting **x_shift** to 0 and applying **y_shift**. (Default is 1) Only one of **x_wrap** **y_wrap** may be set > 1.
or		
y_wrap	integer	Apply **y_shift** for this many values before resetting **y_shift** to 0 and applying **x_shift**. (Default is 1) Only one of **x_wrap** **y_wrap** may be set > 1.

Example 3.3

```
@define_sample_list
{
  list_name           = aerodrome_list;
  location_ident_list = CYFC CYYG CYHZ CYQY CYQX CYJT CYYY;
  location_ident_list = CYZV CYQB CYUL KBOS CJFK;
}
```

Define a list for sampling called "aerodrome_list". The list contains 12 location identifiers which will be matched to locations in a location look up table. The location look up table will be defined in a sampling directive which refers to the named list.

3.4.4 @define_table

Set the type and location for a table to be displayed.

Note that both column and row type tables are allowed. A column type table arranges the table data by columns, and each call to the @table_site directive will create another row of data in the table. A row type table arranges the table data by rows, and each call to the @table_site directive will create another column of data in the table.

KEYWORD	TYPE	DESCRIPTION
table_name	string	Table name
type	string	One of: **column** or **row**
x_off	integer	Column (character) offset wrt current anchor position
y_off	integer	Row (line) offset wrt current anchor position

Example 3.4

```
@define_table
{
  table_name = T1;
  type       = column;
  x_off      = 0;
  y_off      = 4;
}
```

Set the location of a table called "T1" to 4 rows below the current "anchor position". The table will arrange data by column, so that one row (line) of data will be created for each call to the @table_site directive. (Note that "anchor position" is set by @anchor.)

3.4.5 @file_close

Close the ASCII output file.

Note that this directive has no keywords.

Example 3.5

```
@file_close { }
```

Close the TexMet ASCII output file.

3.4.6 @file_name

Set the directory and path name of the output ASCII file.

Note that the current output ASCII file is closed and a new output ASCII file is opened each time this directive is called. Note that the location of the TexMet "output_directory" is defined by the "texout" keyword in the local setup file.

KEYWORD	TYPE	DESCRIPTION
dir	string	Directory name for output ASCII file (The directory name can include one of the codewords found in Table 3.1)
name	string	File name for ASCII output file (The file name can include any of the codewords based on the name of the ASCII product, the **T0** time when the product is run, the current "valid time" (**Tv**), or the actual "clock time" (**Tc**) for sampling. See Table 3.2 for valid options)

Table 3.1: Directory Codewords

<default>	for path to sub-directory in the default output directory
<psmet>	for PSMet directory
<psout>	for PSMet output directory
<svgmet>	for SVGMet directory
<svgout>	for SVGMet output directory
<texmet>	for TexMet directory
<texout>	for TexMet output directory
<home>	for "home" directory defined in local setup file

Table 3.2: Filename Codewords

`<pdf>`	for the ***`product_name`*** of the controlling product definition file (without trailing ".fpdf")
`<iteration>`	for a @loop_begin iteration value.
`<iteration _attribute>`	for the value of the attribute in a @loop_begin iteration.
`<year>`	for **T0** year (such as 2008)
`<month>`	for **T0** month (01-12)
`<day>`	for **T0** day (01-31)
`<julian>`	for **T0** day of the year (001-366)
`<hour>`	for **T0** hour of the day (00-23)
`<minute>`	for **T0** minute in hour (00-59)
`<local>`	L if **T0** time is a local time
`<v_year>`	for **Tv** year (such as 2008)
`<v_month>`	for **Tv** month (01-12)
`<v_day>`	for **Tv** day (01-31)
`<v_julian>`	for **Tv** day of the year (001-366)
`<v_hour>`	for **Tv** hour of the day (00-23)
`<v_minute>`	for **Tv** minute in hour (00-59)
`<v_local>`	L if **Tv** time is a local time
`<p_hour>`	for **Tv-T0** in hours
`<p_minute>`	for **Tv-T0** in minutes
`<p_hr_min>`	for **Tv-T0** in hours and minutes (xx:tt)
`<c_year>`	for **Tc** year (such as 2008)
`<c_month>`	for **Tc** month (01-12)
`<c_day>`	for **Tc** day (01-31)
`<c_julian>`	for **Tc** day of the year (001-366)
`<c_hour>`	for **Tc** hour of the day (00-23)
`<c_minute>`	for **Tc** minute in hour (00-59)

Example 3.6

```
@file_name { dir = <default>; name = <pdf>_<year>:<month>:<day>:<hour>;}
```

Sets the file name for the ASCII output file in TexMet to: `output_directory/sub_directory/product_name_yyyy:mm:dd:hh.txt` where ***`output_directory`*** is identified by the directory tag "texout" in the local setup file, ***`sub_directory`*** is the TexMet directory name for this product, ***`product_name`*** is base name of the controlling product definition file for this product, and ***`yyyy, mm, dd, hh`*** refer to the numeric year, month, day and hour of the **T0** run time for this product.

3.4.7 @group

Set a list of keywords to be included as a "macro".

Note that the keywords are included in another directive by using the format: ***<group_name*** in the keyword list.

KEYWORD	TYPE	DESCRIPTION
group_name	string	Group name for keyword list
acceptable_keywords		Any keywords that are acceptable in the directives that this "macro" is included within can be set here

Example 3.7

```
@group
{
  group_name        = sample_winds;
  wind_crossref     = Fpa_Wind;
  units             = knots;
  format            = wind_text;
  table_name        = T1;
  x_off             = 1;
}
```

Set a "macro" called "sample_winds" containing keywords used in the @sample_wind directive, given in the example below.

3.4.8 @include

Process directives from another product definition file.

Note that the path for the product definition file should be relative to the directory containing the current product definition file.

Note that the ".fpdf" suffix on the file name is optional.

VALUE		DESCRIPTION
file_name	(string)	Path for the product definition file

Example 3.8

```
@include { av_map_setup2 }
```

Process the directives from the file: `av_map_setup2.fpdf`.
Note that this product definition file will be located in the same directory as the product definition file containing the @include directive

3.4.9 @label

Display location dependent information in a table or at a given location. This information includes features labeled in the FPA, or parameters from locations in a "scattered" type field.

Complex (multi-attribute) labels are described at the end of this section.

Note that the **category_attribute** and **category** keywords can be used to display only selected labels.

Note that the **case** and **case_look_up** keywords can only be used for simple (single-attribute) labels.

Note that every label requires a **format** keyword. A **direct** style format will display the value of the label attribute from FPA. A **text** style format also requires a **look_up** keyword to define a look up table, and the value of the label attribute from FPA is cross-referenced in this look up table to identify the text string to display. A **wind_text** format displays winds using the presentation from the directive @wind_presentation.

KEYWORD	TYPE	DESCRIPTION
element	string	FPA element name
level	string	FPA level name
units	string	FPA units for label wind speeds or gusts
geo_file	string	FPA Metafile for geography labels (Note: this must be the name of a file in the "Maps" directory or a full path name)

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which samples to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)

KEYWORD	TYPE	DESCRIPTION

Note that **case** and **case_look_up** can be specified more than once in this directive!

case	string	Time of day or hemisphere in which to use **case_look_up** look up table (given below) One of:**day night dusk** **northern_hemisphere** **southern_hemisphere**
case_look_up	string	File name for the optional look up table (Note: ".tab" suffix is optional)
time_zone	string	Display the time (from a magic attribute using **strftime** format) adjusted to another time zone (see @write_time for more information)
language	string	Display the time (from a magic attribute using **strftime** format) adjusted to another language (see @write_time for more information)
format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **text** ***<cform>*** use look up **wind_text** for wind values (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)
fit_to_map	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)
x_off	integer	Column (character) offset of label
y_off	integer	Row (line) offset of label

For display as a grid ...

x_shift	integer	Column (character) increment for display of labels
y_shift	integer	Row (line) increment for display of labels
x_wrap	integer	Apply **x_shift** for this many values before resetting **x_shift** to 0 and applying **y_shift**. (Default is 1) Only one of **x_wrap y_wrap** may be set > 1.

or

KEYWORD	TYPE	DESCRIPTION
`y_wrap`	integer	Apply **`y_shift`** for this many values before resetting **`y_shift`** to 0 and applying **`x_shift`**. (Default is 1) Only one of **`x_wrap`** **`y_wrap`** may be set > 1.

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`justification`	string	Text justification One of: **`left`** **`centre`** **`right`**

Example 3.9

```
@label
{
  element            = pressure;
  level              = msl;
  category_attribute = FPA_label_type;
  category           = labeled_low labeled_high;
  attribute          = hilo_name;
  format             = direct;
  x_shift            = 0;
  y_shift            = 1;
  justification      = centre;
}
```

Display labels from the msl pressure field which have a value of labeled_low or labeled_high for the FPA attribute FPA_label_type. Labels will be displayed as centred text using the value in the FPA attribute hilo_name, and each label will be displayed 1 row below the previous label.

Complex (multi-attribute) labels allow the display of several label parameters within the same directive. The entire complex label is displayed with respect to the **`x_off`**, **`y_off`** keywords set above.

Attribute keywords apply only to the attribute that they follow.

The position of an attribute in a complex label can be related to the position of other attributes through use of the keywords **`attribute_anchor`**, **`attribute_ref`**, **`attribute_justification`** and **`attribute_vertical_just`**. The **`attribute_ref`** keyword defines the starting position of the attribute with respect to a location about the text identified by the **`attribute_anchor`** keyword. The **`attribute_justification`** and **`attribute_vertical_just`** keywords define the horizontal and vertical positioning of the attribute with respect to this starting position.

The display parameters of each attribute default to the values of the **`units`**, **`format`**, **`look_up`**, and **`justification`** keywords set above. These default values can be replaced by the appropriate attribute keyword listed below, to create multi-format complex labels.

Note that the **`attribute_show`** keyword can be used to "turn off" the display of attributes, while still maintain positioning with respect to other attributes.

KEYWORD	TYPE	DESCRIPTION
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
attribute_look_up	string	File name for look up table (Note: ".tab" suffix is optional) (default is **look_up** set above)
attribute_show	yes/no	Display attribute? (default is **yes**)
attribute_anchor	string	Reference "anchor" for displaying attribute One of: **none** or a previously listed **attribute** (default is **none**, which corresponds to the starting position of the complex label)
attribute_ref	string	Starting position for displaying attribute (a position on the "attribute_anchor" label) One of: **lower_left** **centre_left** **upper_left** **lower_centre** **centre** **upper_centre** **lower_right** **centre_right** **upper_right** (default is **lower_centre**)
attribute_justification	string	Attribute text justification (horizontal) One of: **left**, **centre**, **right** (default is **justification** set above)
attribute_vertical_just	string	Attribute text justification (vertical) One of: **bottom**, **centre**, **top** (default is **bottom**)
attribute_units	string	FPA units for label wind speeds or gusts
attribute_format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **text** ***<cform>*** use look up **wind_text** for wind values (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **attribute_look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format.)
attribute_x_off	integer	Column (character) offset of attribute display
attribute_y_off	integer	Row (line) offset of attribute display

Example 3.10

```
@label
{
  element              = pressure;
  level                = msl;
  category_attribute = FPA_label_type;
  category             = labeled_low labeled_high cyclone_at_min;
  x_shift              = 0;
  y_shift              = 1;
  justification        = left;

  attribute            = hilo_type;
  attribute_anchor     = none;
  attribute_format     = text;
  attribute_look_up    = hilo;

  attribute            = cyclone_type;
  attribute_anchor     = none;
  attribute_format     = text;
  attribute_look_up    = cyclone;

  attribute            = EVAL_spval;
  attribute_anchor     = none;
  attribute_format     = direct %smb;
  attribute_x_off      = 10;
  attribute_y_off      = 0;
}
```

Display labels from the msl pressure field if the value for the FPA attribute FPA_label_type is one of labeled_low, labeled_high or cyclone_at_min. All text will be left justified. The FPA attribute hilo_type will be displayed as text, by matching the attribute value to the values within the look up table `hilo.tab`. The FPA attribute cyclone_type will be displayed as text, by matching the attribute value to the values within the look up table `cyclone.tab`. The FPA attribute EVAL_spval will be displayed as a string in %smb format (1004mb, for example), located 10 columns (characters) to the right of the label location.

3.4.10 @lchain_nodes

Display link chain node information.

Note that the **category_attribute** and **category** keywords can be used to display only selected nodes, and that the **track_category_attribute** and **track_category** keywords can be used to display only nodes from selected tracks.

Note that the **case** and **case_look_up** keywords can only be used for simple (single-attribute) labels.

Note that every node requires a **format** keyword. A **direct** style format will display the value of the label attribute from FPA. A **text** style format also requires a **look_up** keyword to define a look up table, and the value of the label attribute from FPA is cross-referenced in this look up table to identify the text string to display. A **wind_text** format displays winds using the presentation from the directive @wind_presentation.

KEYWORD	TYPE	DESCRIPTION
Note that **element** and **level** can be specified more than once in this directive. Note that if a single element name is matched with a list of level names, or vice versa, the single name must be given first. Multiple element and level name lists should have the same number.		
element	string	FPA element name
element_list	string	List of FPA element names
level	string	FPA level name
level_list	string	List of FPA level names
node_speed_units	string	FPA units for motion of link chain at node.
node_speed_round	real	Precision of node speed.
node_stationary_max	real	Limit node labels based on node speed.
node_stationary_label	string	Label to use if node speed is less than **node_stationary_max**.
units	string	FPA units for node wind speeds or gusts (if required)
start_time	vtime	Nodes before this time will not be included. Default is the start of the track.
end_time	vtime	Nodes after this time will not be included. Default is the end of the track.
or		
times	vtime(s)	List of times to be included.
Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.		
category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)

KEYWORD	TYPE	DESCRIPTION
category	string	FPA category name (or list of names) for selecting which nodes to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)

Note that **track_category_attribute** and **track_category** can be specified more than once in this directive, and that **track_category_cascade** determines how to combine more than one instance.

track_category_cascade	string	How to combine more than one instance of **track_category_attribute** and **track_category** One of: **and** or **or** (default is **and**)
track_category_attribute	string	FPA attribute containing category names for tracks containing the nodes (default is **FPA_category**)
track_category	string	FPA category name (or list of names) for selecting which tracks contain the nodes to display (Note: name(s) are matched with the value sampled from **track_category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)

Note that **case** and **case_look_up** can be specified more than once in this directive!

case	string	Time of day or hemisphere in which to use **case_look_up** look up table (given below) One of:**day night dusk northern_hemisphere southern_hemisphere**
case_look_up	string	File name for the optional look up table (Note: ".tab" suffix is optional)
time_zone	string	Display the time (from a magic attribute using **strftime** format) adjusted to another time zone (see @write_time for more information)
language	string	Display the time (from a magic attribute using **strftime** format) adjusted to another language (see @write_time for more information)

KEYWORD	TYPE	DESCRIPTION
`format`	string	Format of attribute to display One of: **`direct`** ***`<cform>`*** use value **`text`** ***`<cform>`*** use look up **`wind_text`** for wind values (Note: ***`<cform>`*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **`look_up`** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***`<cform>`*** conversion format)
`fit_to_map`	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)
`x_off`	integer	Column (character) offset of node label wrt current anchor position
`y_off`	integer	Row (line) offset of node label wrt current anchor position

For display as a grid ...

`x_shift`	integer	Column (character) increment for display of nodes
`y_shift`	integer	Row (line) increment for display of nodes
`x_wrap`	integer	Apply **`x_shift`** for this many values before resetting **`x_shift`** to 0 and applying **`y_shift`**. (Default is 1) Only one of **`x_wrap`** **`y_wrap`** may be set > 1.

or

`y_wrap`	integer	Apply **`y_shift`** for this many values before resetting **`y_shift`** to 0 and applying **`x_shift`**. (Default is 1) Only one of **`x_wrap`** **`y_wrap`** may be set > 1.

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`justification`	string	Text justification One of: **`left`** **`centre`** **`right`**

Example 3.11

```
@lchain_nodes
{
  element          = tropical_tracks;
  level            = surface;
  times            = -12 -6 0 6 12 18 24;
  x_shift          = 0;
  y_shift          = 1;
  justification    = right;
  attribute        = GPGen_gmt_time;
  attribute_format = direct %H:%MZ;
  attribute_x_off  = 8;
  attribute        = NWquadGales;
  attribute_format = direct %skm;
  attribute_x_off  = 16;
  attribute        = NEquadGales;
  attribute_format = direct %skm;
  attribute_x_off  = 24;
  attribute        = SEquadGales;
  attribute_format = direct %skm;
  attribute_x_off  = 32;
  attribute        = SWquadGales;
  attribute_format = direct %skm;
  attribute_x_off  = 40;
}
```

Display gale quadrant radii from the surface tropical_tracks field for times every 6 hours from -12 to 24. The wind radii will be displayed on successive lines, beginning with a formatted GMT time, and then wind radii distances (in km) for NW, NE, SE and SW quadrants.

Complex (multi-attribute) labels allow the display of several label parameters within the same directive. The entire complex label is displayed with respect to the **`x_off`**, **`y_off`** keywords set above.

Attribute keywords apply only to the attribute that they follow.

The position of an attribute in a complex label can be related to the position of other attributes through use of the keywords **`attribute_anchor`**, **`attribute_ref`**, **`attribute_justification`** and **`attribute_vertical_just`**. The **`attribute_ref`** keyword defines the starting position of the attribute with respect to a location about the symbol or text identified by the **`attribute_anchor`** keyword. The **`attribute_justification`** and **`attribute_vertical_just`** keywords define the horizontal and vertical positioning of the attribute with respect to this starting position.

The display parameters of each attribute default to the values of the **`format`**, **`justification`**, **`look_up`**, **`units`** keywords set above. These default values can be replaced by the appropriate attribute keyword listed below, to create multi-format complex labels.

Note that the **`attribute_show`** keyword can be used to "turn off" the display of attributes, while still maintain positioning with respect to other attributes.

KEYWORD	TYPE	DESCRIPTION
`attribute`	string	FPA attribute containing value to match with values in **`look_up`** table (or a magic attribute)
`attribute_look_up`	string	File name for look up table (Note: ".tab" suffix is optional) (default is **`look_up`** set above)
`attribute_show`	yes/no	Display attribute? (default is **`yes`**)
`attribute_anchor`	string	Reference "anchor" for displaying attribute One of: **`none`** or a previously listed **`attribute`** (default is **`none`**, which corresponds to the starting position of the complex label)
`attribute_ref`	string	Starting position for displaying attribute (a position on the "attribute_anchor" label) One of: **`lower_left`** **`centre_left`** **`upper_left`** **`lower_centre`** **`centre`** **`upper_centre`** **`lower_right`** **`centre_right`** **`upper_right`** (default is **`lower_centre`**)
`attribute_justification`	string	Attribute text justification (horizontal) One of: **`left`**, **`centre`**, **`right`** (default is **`justification`** set above)
`attribute_vertical_just`	string	Attribute text justification (vertical) One of: **`bottom`**, **`centre`**, **`top`** (default is **`bottom`**)
`attribute_units`	string	FPA units for label wind speeds or gusts
`attribute_format`	string	Format of attribute to display One of: **`direct`** ***<cform>*** use value **`text`** ***<cform>*** use look up **`wind_text`** for wind values (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **`attribute_look_up`** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format.)
`attribute_x_off`	integer	Column (character) offset of node attribute wrt current anchor position
`attribute_y_off`	integer	Row (line) offset of node attribute wrt current anchor position

3.4.11 @loop_begin

Begin a loop to repeat all directives from this @loop_begin directive up to the matching @loop_end directive. The keyword values for each group and keyword are reset to the next value in the list for each iteration. If the number of keyword values is less than the number of iterations, then the command reuses the list of keyword values as many times as is necessary. The number of iterations is set either by the keyword **number_of_iterations** or it is determined by the number of features that match the category specified for a particular field (element/level pair).

Note that the **category_attribute** and **category** keywords can be used one or more times to choose only selected features for a particular field (element/level pair).

Note that the keywords and their values are included in another directive by using the format <***group_name*** in the keyword list.

Note that @loop_begin and @loop_end directives may be embedded.

KEYWORD	TYPE	DESCRIPTION
number_of_iterations	string	Number of iterations of directives (up to the matching @loop_end directive)
or		
element	string	Element to loop on
level	string	Level to loop on
field_type	string	Use **Scattered** to loop on field labels. Otherwise not required.
area_type	string	Use **subareas** to loop based on subareas of an area and subarea attributes or **boundary** to loop based on areas and default area attributes.

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which samples to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	value to replace magic term **<iteration_attribute>**.

Note that **group_name**, **keyword_name**, **keyword_value** and **keyword_value_list** can be specified more than once in this directive, and that a **group_name** may not require a **keyword_name** if the directive to be accessed requires only a value.

KEYWORD	TYPE	DESCRIPTION
group_name	string	Group name for keyword list
keyword_name	string	Keyword name
keyword_value	string	Keyword value for one iteration
keyword_value_list	string(s)	Keyword value for a number of iterations

Example 3.12

```
@loop_begin {
  number_of_iterations = 5;  group_name = source_group;
  keyword_name = source;      keyword_value = interp;
  keyword_name = valid_time; keyword_value_list = 00 06 12 18 24; }
```

Set a "macro" called "source_group" containing values for the keywords source and valid_time to be used in subsequent directives. (The value of the source keyword will be interp for each iteration.)

3.4.12 @loop_end

End a loop to repeat directives. The directives between the matching @loop_begin directive and this @loop_end directive will be repeated for the number of iterations specified in the @loop_begin directive.

Note that @loop_begin and @loop_end directives may be embedded.

Note that this directive has no keywords

Example 3.13

```
@loop_end { }
```

End of looping begun by an @loop_begin directive.

3.4.13 @loop_location_look_up

Build an internal location look up table from the current line, point or link chain feature in a loop. Note that the current feature is extracted from the field identified in the @loop_begin directive. Line and link chain features can be used for space type displays. The horizontal extent of the display is determined by the length of the feature, with look up locations set each **`location_interval`** along the length, or at every point in the line or every node on the link chain if the **`location_interval`** is not set. Point features can be used for time type displays, with look up locations at the point location for each time given in the **`times`** keyword, or at the current valid time if the **`times`** keyword is not set. Link chain features can be used for time type displays, with look up locations at every node on the link chain, except when modified by the keywords **`times`** or the keywords **`start_time`** and **`end_time`**.

Note that times in the **`times`** keyword must be listed in increasing order.

Note that when using the **`start_time`** and **`end_time`** keywords, the start time must be before the end time.

KEYWORD	TYPE	DESCRIPTION
`location_look_up`	string	Name for location look up table. (Note: This is an internal name that can be referred to by other directives, and does not require a ".ltab" suffix.)
`location_interval`	real	Distance along feature for each look up location from a line or link chain feature.
`location_units`	string	FPA units for **`location_interval`**.
`times`	vtime(s)	List of times to use in look up table from a point or link chain feature.
`start_time`	vtime	Start time for look up locations from a link chain feature.
`end_time`	vtime	End time for look up locations from a link chain feature.
`labels`	string(s)	Labels to match up with locations or times.

Example 3.14

```
@loop_location_look_up
  {
  location_look_up = time_intervals;
  start_time       = 00/12:00;
  end_time         = 01/12:00;
  }
```

Build an internal location look up table called "time_intervals" from nodes between 12:00 "today" until 12:00 "tomorrow" from link chain features extracted from the field identified in a @loop_begin directive.

3.4.14 @mapdef

Set the map definition for the current map.

A more complete description of map definitions, including acceptable formats for all parameters, can be found in the FPA Graphics Metafile Standard.

Note that this map definition does not need to match the map definition from the FPA that generates the ASCII product, but this map definition should be completely within the map definition from the FPA.

KEYWORD	TYPE	DESCRIPTION
olat	latlon	Latitude of the map "origin" (see Format of Latitudes and Longitudes, (Section 1.3))
olon	latlon	Longitude of the map "origin" (see Format of Latitudes and Longitudes, (Section 1.3))
rlon	latlon	Vertical reference longitude of the map (see Format of Latitudes and Longitudes, (Section 1.3))
xmin	real	Start point of the x-axis (in units of **map_units** keyword below)
ymin	real	Start point of the y-axis (in units of **map_units** keyword below)
xmax	real	End point of the x-axis (in units of **map_units** keyword below)
ymax	real	End point of the y-axis (in units of **map_units** keyword below)
map_units	real	Number of meters (or degrees) per map unit

Example 3.15

```
@mapdef
{
  olat      = 30.0N;
  olon      = 75.0W;
  rlon      = 65.0W;
  xmin      = 0;
  ymin      = 0;
  xmax      = 6500;
  ymax      = 3500;
  map_units = 1000;
}
```

Set a map definition for sampling with an origin of 30.0 degrees North and 75.0 degrees West, oriented with the 65.0 degrees West longitude vertical. All distances in the map definition will be in units of 1000 m, or km. The lower left hand corner of the map will extend from 0 km to 6500 km horizontally from the origin, and from 0 km to 3500 km vertically from the origin. (Note that the origin in this example is at the lower left corner of the map.)

3.4.15 @presentation

Set the appearance of text for the output file.

Note that the @presentation directive with no "name" keyword sets the default appearance of all text for all subsequent directives.

Note that presentation keywords can also be included in other directives, where their effect is to change the appearance of the text for that directive only.

KEYWORD	TYPE	DESCRIPTION
`name`	string	Name for a special presentation
`justification`	string	Text justification One of: **`left`** **`centre`** **`right`**

Example 3.16

```
@presentation { justification = left; }
```

Set a presentation, to be used for all subsequent directives, with text displayed left justified.

3.4.16 @process

Execute one or more UNIX or Linux system commands.

Note that codewords accepted in @file_name directive are also accepted here.

KEYWORD	TYPE	DESCRIPTION
commands	(string)	One or more UNIX or Linux commands (Note: commands can include any of the codewords in Table 3.3 based on the full name of the output file, output directories, "pdf" file names, the **T0** time when the product is run, the current "valid_time" (**Tv**) or the actual clock time (**Tc**).

Table 3.3: Codewords

<file_name>	for full output file name
<file_name_base>	for full output file name without the extension
<iteration>	for a @loop_begin iteration value.
<iteration _attribute>	for the value of the attribute in a @loop_begin iteration.
<SETUP>	for name of local setup
<RTIME>	for **T0** time stamp
<VTIME>	for **Tv** time stamp
<default>	path to ***sub_directory*** in the ***output_directory***
<psmet>	for PSMet directory
<psout>	for PSMet output directory
<svgmet>	for SVGMet directory
<svgout>	for SVGMet output directory
<texmet>	for TexMet directory
<texout>	for TexMet output directory
<home>	for "home" directory defined in local setup file
<pdf>	for the ***product_name*** of the controlling product definition file (without trailing ".fpdf")
<year>	for **T0** year (such as 2008)
<month>	for **T0** month (01-12)
<day>	for **T0** day (01-31)
<julian>	for **T0** day of the year (001-366)
<hour>	for **T0** hour of the day (00-23)
<minute>	for **T0** minute in hour (00-59)
<local>	L if **T0** time is a local time
<v_year>	for **Tv** year (such as 2008)
<v_month>	for **Tv** month (01-12)
<v_day>	for **Tv** day (01-31)
<v_julian>	for **Tv** day of the year (001-366)
<v_hour>	for **Tv** hour of the day (00-23)
<v_minute>	for **Tv** minute in hour (00-59)
<v_local>	L if **Tv** time is a local time

Table 3.3: (continued)

<p_hour>	for **Tv-T0** in hours
<p_minute>	for **Tv-T0** in minutes
<p_hr_min>	for **Tv-T0** in hours and minutes (xx:tt)
<c_year>	for **Tc** year (such as 2008)
<c_month>	for **Tc** month (01-12)
<c_day>	for **Tc** day (01-31)
<c_julian>	for **Tc** day of the year (001-366)
<c_hour>	for **Tc** hour of the day (00-23)
<c_minute>	for **Tc** minute in hour (00-59)

Example 3.17

```
@process
{
  vi <file_name>
  cp <file_name> <home>/transfer
}
```

Display the output ASCII file using the system **vi** command, and then copy the output file to the "transfer" subdirectory of the "home" directory (defined in the local setup file).

3.4.17 @projection

Set the map projection for the current map.

A more complete description of map projections, including acceptable formats for all parameters, can be found in the FPA Graphics Metafile Standard.

KEYWORD	TYPE	DESCRIPTION
type	string	Keyword for projection (predefined) One of: **latitude_longitude** **plate_caree** **polar_stereographic** **lambert_conformal** **mercator_equatorial** **rotated_lat_lon** **oblique_stereographic**
ref1	string	Projection dependent reference value
ref2	string	Projection dependent reference value
ref3	string	Projection dependent reference value
ref4	string	Projection dependent reference value
ref5	string	Projection dependent reference value

Table 3.4: Projection types and associated parameters

Projection	Type	Parameters
Lambert Conformal	**lambert_conformal**	upper reference latitude lower reference latitude
Latitude-Longitude	**latitude_longitude**	
Mercator Equatorial	**mercator_equatorial**	
Oblique Stereographic	**oblique_stereographic**	central latitude central longitude secant angle [optional]
Plate-Caree	**plate_caree**	
Polar Stereographic	**polar_stereographic**	**north** or **south** "true" latitude
Rotated Latitude-Longitude	**rotated_lat_lon**	bottom axis latitude bottom axis longitude rotation angle [optional]

Example 3.18

```
@projection { type = polar_stereographic; ref1 = north; ref2 = 60; }
```

Set a map projection for the current map as north **polar_stereographic** true at 60 degrees North.

3.4.18 @resolution

Set the spline resolution for "continuous" or "vector" fields for sampling.

Note that **map_units** is automatically reset to the value of **map_units** in the @mapdef directive, and **res** is scaled accordingly.

Note that values determined by **xmax** - **xmin** and **ymax** - **ymin** keywords in the @mapdef directive should be multiples of the the value of the **res** keyword, given below.

KEYWORD	TYPE	DESCRIPTION
res	real	Spline resolution (in units of **map_units** keyword below)
map_units	real	Number of meters (or degrees) per map unit

Example 3.19

```
@resolution { res = 200; map_units = 1000; }
```

Set a spline resolution for "continuous" or "vector" fields to 200 distance units, where all distance units are expressed in 1000 m, or km.

3.4.19 @sample_field

Display values (or text) sampled from FPA fields in the ASCII output file.

Complex (multi-attribute) samples are described at the end of this section.

Note that the **category_attribute** and **category** keywords can be used to sample only selected values.

Note that the **case** and **case_look_up** keywords can only be used for simple (single-attribute) samples.

Note that every sample requires a **format** keyword. A **direct** style format will display the value sampled from FPA. A **text** style format also requires a **look_up** keyword to define a look up table, and the value sampled from FPA is cross-referenced in this look up table to identify the text string to display. A **wind_text** format displays winds using the presentation from the directive @wind_presentation. A **vector_text** format displays "vector" field values using the presentation from the directive @vector_presentation.

It is now possible to sample information from ASCII data files. Note that all data files must be arranged row by row, with each data parameter in a separate column. The keywords **data_file**, **data_file_format**, **data_file_units** and **data_file_wind_units** provide control for this action. Note that all parameters in the data file are accessed by magic attributes, as follows:

GPGEN_ident	Identifier string
GPGEN_lat or **GPGEN_lat_ddmm**	Latitude location
GPGEN_lon or **GPGEN_lon_ddmm**	Longitude location
GPGEN_label	Data string
GPGEN_value	Data value
GPGEN_wind	Data wind direction (degrees true)

The keywords **proximity** and **proximity_units** allow sampling attributes from discrete areas, line features, link chains or scattered points within the given proximity of the sample location. Use the magic attribute **GPGEN_proximity** to display the distance from the sample location to the feature in km.

KEYWORD	TYPE	DESCRIPTION
element	string	FPA element name
level	string	FPA level name
equation	string	FPA equation string for calculating a field (refer to Appendix B of Administrator's Guide for format of equations)
field_type	string	Use **Scattered** to sample field labels. Otherwise not required.
units	string	FPA units for "continuous" or "vector" field samples
geo_file	string	FPA Metafile for sampling geography (Note: this must be the name of a file in the "Maps" directory or a full path name)
proximity	real	Only display sample if area, line, link chain or scattered feature is within the given proximity
proximity_units	string	FPA units for proximity

KEYWORD **TYPE** **DESCRIPTION**

Note that **category_attribute** and **category** can be specified more than once in this directive, and that **category_cascade** determines how to combine more than one instance.

KEYWORD	TYPE	DESCRIPTION
category_cascade	string	How to combine more than one instance of **category_attribute** and **category** One of: **and** or **or** (default is **and**)
category_attribute	string	FPA attribute containing category names (default is **FPA_category**)
category	string	FPA category name (or list of names) for selecting which samples to display (Note: name(s) are matched with the value sampled from **category_attribute**) (default is **All** names)
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
look_up	string	File name for the look up table (Note: ".tab" suffix is optional)

Note that **case** and **case_look_up** can be specified more than once in this directive!

KEYWORD	TYPE	DESCRIPTION
case	string	Time of day or hemisphere in which to use **case_look_up** look up table (given below) One of:**day night dusk** **northern_hemisphere** **southern_hemisphere**
case_look_up	string	File name for the optional look up table (Note: ".tab" suffix is optional)
format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **text** ***<cform>*** use look up **wind_text** for wind values **vector_text** for vector values (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)

<table>
<tr><th>KEYWORD</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td><b>time_zone</b></td><td>string</td><td>Display the time (from a magic attribute using <b>strftime</b> format) adjusted to another time zone (see @write_time for more information)</td></tr>
<tr><td><b>language</b></td><td>string</td><td>Display the time (from a magic attribute using <b>strftime</b> format) adjusted to another language (see @write_time for more information)</td></tr>
<tr><td><b>data_file</b></td><td>string</td><td>Identify the name of the ASCII data file to sample. May contain codewords: <default>, <psmet>, <psout>, <svgmet>, <svgout>, <texmet>, <texout> or <home> to refer to PSMet, SVGMEt or TexMet directories in the same FPA database.</td></tr>
<tr><td><b>data_file_format</b></td><td>string</td><td>Identify the format of the data in the ASCII data file with a list of recognized format types separated by the same characters used to separate the values in the ASCII data file.
Recognized types:
<table>
<tr><td><b>identifier</b></td><td>Identifier string</td></tr>
<tr><td><b>latitude</b></td><td>Latitude location</td></tr>
<tr><td><b>longitude</b></td><td>Longitude location</td></tr>
<tr><td><b>timestamp</b></td><td>Valid time</td></tr>
<tr><td><b>label</b></td><td>Data string</td></tr>
<tr><td><b>value</b></td><td>Data value</td></tr>
<tr><td><b>units</b></td><td>FPA units of data value</td></tr>
<tr><td><b>wind_direction</b></td><td>Data wind direction (degrees true)</td></tr>
<tr><td><b>wind_speed</b></td><td>Data wind speed</td></tr>
<tr><td><b>wind_gust</b></td><td>Data wind gust</td></tr>
<tr><td><b>wind_units</b></td><td>FPA units of data wind speed or gusts</td></tr>
<tr><td><b>-</b></td><td>Placeholder in data file</td></tr>
</table>
Note that either the <b>identifier</b> or the <b>latitude</b> and <b>longitude</b> must be present to identify the data locations, but all other format types are optional.
Note that <b>label</b>, <b>value</b> and <b>wind</b> format types can all be present in the same <b>data_file_format</b> keyword.</td></tr>
<tr><td><b>data_file_units</b></td><td>string</td><td>FPA units for data values
Note that all data values are automatically converted to MKS units. Either the <b>units</b> format type or the <b>data_file_units</b> keyword are required for values in the data file that are non-MKS.</td></tr>
</table>

KEYWORD	TYPE	DESCRIPTION
data_file_wind_units	string	FPA units for wind speed or gust values Note that all wind speed and gust values are automatically converted to knots. Either the **wind_units** format type or the **data_file_wind_units** keyword are required for wind speeds or gusts in the data file that are not in knots.
fit_to_map	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)
x_off	integer	Column (character) offset wrt current anchor position
y_off	integer	Row (line) offset wrt current anchor position
source	string	FPA source name for data directory for field to sample (Note: the format for this keyword would be ***source*** if the data directory has no sub sources, or ***source*:*subsource*** if the data directory has sub sources) (default is current **source** from @set_source directive)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (default is current **valid_time** from @set_source directive)
For point samples ...		
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
map_x	real	Position on x-axis of map (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units**)
map_y	real	Position on y-axis of map (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units**)
map_units	real	Number of meters (or degrees) per map unit
location_ident	string	Identifier for location in **location_look_up**
grid_name	string	Grid name (from @define_sample_grid) (Note: one point will be sampled for each location on the grid)
list_name	string	List name (from @define_sample_list) (Note: one point will be sampled for each location in the list)

KEYWORD	TYPE	DESCRIPTION
For point or table samples ...		
location_look_up	string	File name for the location look up table (Note: ".ltab" suffix is optional)
For table samples ...		
table_name	string	Table name (from @define_table) (Note: one point will be sampled for each call to the @table_site directive)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
justification	string	Text justification One of: **left centre right**

Example 3.20

```
@sample_field
{
  element = pressure;
  level   = msl;
  units   = mb;
  lat     = 49N;
  lon     = 65W;
  format  = direct %.1f mb;
  x_off   = 0;
  y_off   = 0;
}
```

Sample the msl pressure field at latitude 49 degrees North and longitude 65 degrees West. The value will be displayed as a real number rounded to the nearest 0.1 mb and followed by " mb" (for example, "1004.6 mb"). The value will be displayed at the current location in the ASCII output file.

Example 3.21

```
@sample_field
{
  table_name    = T1;
  element       = weather_system;
  level         = surface;
  attribute     = clds_and_wx;
  format        = text;
  look_up       = weather_day;
  case          = night;
  case_look_up  = weather_ngt;
  source        = interp;
  valid_time    = 12;
  x_off         = 40;
  y_off         = 0;
  justification = centre;
}
```

Sample the surface weather_system field from the interp source 12 hours after the current time for each site (from the @table_site directive) of a table called "T1" (from the @define_table directive). Display the sampled value for the clds_and_wx attribute as text by matching the attribute value to the values within the look up table `weather_day.tab` (or by matching to the attribute value to the values within the look up table `weather_ngt.tab` at nighttime). The text will be displayed 40 columns (characters) to the right of the current "anchor position" in the table, and centre justified. (Note that "anchor position" is set by @anchor.)

Example 3.22 Example data_file_format

The **`data_file_format`** string is a list of recognized format types separated by the same characters used to separate the values in the ASCII data file.

The following format would indicate that the identifier string is located in the second column of the ASCII data file, the time stamp in the fourth column, data file string in the seventh column, and that the columns are separated by white space (blanks or tabs).

```
data_file_format = - identifier - timestamp - - label
```

The following format would indicate that the latitude and longitude are located in the first and second columns of the ASCII data file, a data file value in the fifth column, the FPA units for the value in the fourth column, and that the columns are separated by commas.

```
data_file_format = latitude,longitude,-,units,value
```

See Example 2.86.

Complex (multi-attribute) samples allow the display of several sampled parameters within the same directive. The entire complex sample is displayed with respect to the **x_off** and **y_off** keywords set above.

Attribute keywords apply only to the attribute that they follow.

The position of an attribute in a complex sample can be related to the position of other attributes through use of the keywords **attribute_anchor**, **attribute_ref**, **attribute_justification**, and **attribute_vertical_just**. The **attribute_ref** keyword defines the starting position of the attribute with respect to a location about the text identified by the **attribute_anchor** keyword. The **attribute_justification** and **attribute_vertical_just** keywords define the horizontal and vertical positioning of the attribute with respect to this starting position.

The display parameters of each attribute default to the values of the **look_up**, **justification**, **units**, and **format** keywords set above. These default values can be replaced by the appropriate attribute keyword listed below, to create multi-format complex labels.

Note that the **attribute_show** keyword can be used to "turn off" the display of attributes, while still maintaining positioning with respect to other attributes.

KEYWORD	TYPE	DESCRIPTION
attribute	string	FPA attribute containing value to match with values in **look_up** table (or a magic attribute)
attribute_look_up	string	File name for look up table (Note: ".tab" suffix is optional) (default is **look_up** set above)
attribute_show	yes/no	Display attribute? (default is **yes**)
attribute_anchor	string	Reference "anchor" for displaying attribute One of: **none** or a previously listed **attribute** (default is **none**, which corresponds to the starting position of the complex label)
attribute_ref	string	Starting position for displaying attribute (a position on the "attribute_anchor" label) One of: **lower_left centre_left upper_left lower_centre centre upper_centre lower_right centre_right upper_right** (default is **lower_centre**)
attribute_justification	string	Attribute text justification (horizontal) One of: **left**, **centre**, **right** (default is **justification** set above)
attribute_vertical_just	string	Attribute text justification (vertical) One of: **bottom**, **centre**, **top** (default is **bottom**)
attribute_units	string	FPA units for "continuous" or "vector" field samples

KEYWORD	TYPE	DESCRIPTION
attribute_format	string	Format of attribute to display One of: **direct** ***<cform>*** use value **text** ***<cform>*** use look up **wind_text** for wind values **vector_text** for vector values (Note: ***<cform>*** is an optional string which contains a **printf** conversion format for the attribute value. This conversion format is applied to the value before accessing the **attribute_look_up** table or displaying the value. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.) (Note: use \ before all control characters in the optional ***<cform>*** conversion format)
attribute_x_off	integer	Column (character) offset of attribute display
attribute_y_off	integer	Row (line) offset of attribute display

Example 3.23

```
@sample_field
{
  element                    = weather_system;
  level                      = surface;
  format                     = direct;
  lat                        = 41.5N;
  lon                        = 67.5W;

  attribute                  = cloud_top;
  attribute_anchor           = none;
  attribute_justification = centre;

  attribute                  = cloud_base;
  attribute_anchor           = cloud_top;
  attribute_ref              = lower_centre;
  attribute_justification = centre;
  attribute_vertical_just = top;
  .....
}
```

Sample the surface weather_system field at latitude 41.5 degrees North and longitude 67.5 degrees West. The attribute cloud_top will be displayed at the sample location, and centre justified. The top edge of the attribute cloud_base will be displayed at the lower_centre of the cloud_top attribute (that is, on the line directly below and at the character at the centre of the cloud_top attribute), and centre justified. (Note that if the value of the cloud_base attribute is blank, then any positions using an attribute_anchor of cloud_base will be positioned at the lower_centre of the cloud_top attribute!) (Note that the remaining keywords from this example have not been included.)

3.4.20 @sample_wind

Display cross-referenced winds from FPA fields in the ASCII output file.

Note that the cross-reference "Fpa_Wind" refers to the FPA wind field.

Note that winds are displayed according to the **wind_text** format, using the presentation defined in the @wind_presentation directive.

KEYWORD	TYPE	DESCRIPTION
wind_crossref	string	FPA wind cross-reference
units	string	FPA units for wind speeds or gusts to display
format	string	Use: **wind_text** for text display
fit_to_map	yes/no	Ensure that label (or complex label) is inside limits of current map? (Default is no.)
x_off	integer	Column (character) offset wrt current anchor position
y_off	integer	Row (line) offset wrt current anchor position
source	string	FPA source name for data directory for field to sample (Note: the format for this keyword would be ***source*** if the data directory has no sub sources, or ***source*:*subsource*** if the data directory has sub sources) (default is current **source** from @set_source directive)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (default is current **valid_time** from @set_source directive)
For point samples ...		
lat	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
lon	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
map_x	real	Position on x-axis of map (Note: position is measured from **xmin** from @mapdef directive, in units of **map_units**)
map_y	real	Position on y-axis of map (Note: position is measured from **ymin** from @mapdef directive, in units of **map_units**)
map_units	real	Number of meters (or degrees) per map unit
location_ident	string	Identifier for location in **location_look_up**
grid_name	string	Grid name (from @define_sample_grid) (Note: one point will be sampled for each location on the grid)

KEYWORD	TYPE	DESCRIPTION
`list_name`	string	List name (from @define_sample_list) (Note: one point will be sampled for each location in the list)
For point or table samples ...		
`location_look_up`	string	File name for the location look up table (Note: ".ltab" suffix is optional)
For table samples ...		
`table_name`	string	Table name (from @define_table) (Note: one point will be sampled for each call to the @table_site directive)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`justification`	string	Text justification One of: **`left centre right`**

Example 3.24

```
@sample_wind { <sample_winds; y_off = 3; }
```

Note that the keywords for the "macro" called "sample_winds" are defined as an example in the @group directive, given above, and are automatically included as part of this directive. Sample the Fpa_Wind wind cross-reference for each site (from the @table_site directive) of a table called "T1" (from the @define_table directive). Display the sampled wind scaled in units of knots, and with a presentation given by the @wind_presentation directive. The sampled wind will be displayed 1 column to the right of and 3 rows below the current "anchor position" in the table.

Example 3.25

```
@sample_wind
{
  wind_crossref = Vg_Msl;
  units         = knots;
  format        = wind_text;
  x_off         = 30;
  y_off         = 0;
  lat           = 44:00;
  lon           = 63:35;
}
```

Sample the Vg_Msl wind cross-reference at latitude 44 degrees and 40 minutes North and longitude 63 degrees and 35 minutes West. Display the sampled wind scaled in units of knots, and with a presentation given by the @wind_presentation directive. The sampled wind will be displayed 30 columns to the right of the current "anchor position". (Note that "anchor position" is set by @anchor.)

3.4.21 @set_source

Set the data directory and valid time for fields to sample.

Note that all times are relative to the **T0** time from FPA.

KEYWORD	TYPE	DESCRIPTION
source	string	FPA source name for data directory for field to sample (Note: the format for this keyword would be ***source*** if the data directory has no sub sources, or ***source*:*subsource*** if the data directory has sub sources)
valid_time	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5))

Example 3.26

```
@set_source { source = depict; valid_time = 00; }
```

Set the data directory for fields to sample to the FPA depict source at the current time.

Example 3.27

```
@set_source { source = GEM; valid_time = 18; }
```

Set the data directory for fields to sample to the FPA GEM source at 18 hours after the current time.

3.4.22 @size

Set the size of the ASCII output file.

Note that @size directive must be included in the product definition files BEFORE any directives which display data!

KEYWORD	TYPE	DESCRIPTION
columns	integer	Number of columns (characters) in output file
rows	integer	Number of rows (lines) in output file

Example 3.28

```
@size { width = 80; height = 120; }
```

Set the size of the ASCII output file to 80 columns (characters) by 120 rows (lines).

3.4.23 @table_site

Set parameters for locations in a table at which to sample data.

Note that the type of table is set in the @define_table directive. A column type table arranges the table data by columns, and each call to the @table_site directive will create another row of data in the table. A row type table arranges the table data by rows, and each call to the @table_site directive will create another column of data in the table.

Note that data in the table is "anchored" to the table location set in the @define_table directive according to the **`x_off`** or **`y_off`** keywords given below. In a column type table, data for each site forms another row in the table, offset vertically from the table location according to **`y_off`**. In a row type table, data for each site forms another column in the table, offset horizontally from the table location according to **`x_off`**.

Note that the data locations for sampling or display can be set directly as latitude-longitude or map grid locations, but can also be set indirectly as a location identifier, which is matched to a **`location_look_up`**" table (set in the sampling or display directives) to determine the location.

KEYWORD	TYPE	DESCRIPTION
`table_name`	string	Table name (from @define_table)
`site_label`	string	Site label for location
`x_label`	integer	Column (character) offset of the site label wrt **`x_off`**
`y_label`	integer	Row (line) offset of the site label wrt **`y_off`**
`x_off`	integer	Column (character) offset of table data wrt table location
`y_off`	integer	Row (line) offset of table data wrt table location

For lat/lon samples ...

`lat`	latlon	Latitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))
`lon`	latlon	Longitude position on current map (see Format of Latitudes and Longitudes, (Section 1.3))

For map samples ...

`map_x`	real	Position on x-axis of map (Note: position is measured from **`xmin`** from @mapdef directive, in units of **`map_units`**)
`map_y`	real	Position on y-axis of map (Note: position is measured from **`ymin`** from @mapdef directive, in units of **`map_units`**)
`map_units`	real	Number of meters (or degrees) per map unit

For location samples ...

KEYWORD	TYPE	DESCRIPTION
location_ident	string	Identifier for location in **location_look_up** (Note that the keyword **location_look_up** in the @sample_field @sample_wind @text directives identifies the location look up table to use.)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
justification	string	Text justification One of: **left centre right**

Example 3.29

```
@table_site {
  table_name = T1; site_label = Halifax;
  lat = 44:40N; lon = 63:35W;
  x_off = 0;     y_off = 8; }
```

Set a location for a site called "Halifax" at latitude 44 degrees and 40 minutes North and longitude 63 degrees and 35 minutes West in a table called "T1" (from the @define_table directive). Since the table is of type "column", all sampled data for this site will be displayed 8 rows (lines) below the table location (from the @define_table directive). The site label will be displayed at the initial data location (which is 8 rows below the table location).

Example 3.30

```
@table_site {
  table_name = T2; site_label = ,;
  map_x = 200; map_y = 400;
  x_off = 14;  y_off = 2;
  justification = left; }
```

Set a location for a site 200 "map units" along the x-axis and 400 "map units" along the y-axis from the lower left corner of a map definition (from the @mapdef directive) in a table called "T2" (from the @define_table directive). For a "column" type table, all sampled data for this site will be displayed 2 rows (lines) below the table location (from the @define_table directive). For a "row" type table, all sampled data for this site will be displayed 14 columns (characters) to the right of the table location (from the @define_table directive). The "," character will be displayed 14 columns (characters) to the right and 2 rows (lines) below the table location, and left justified. (Note that "map_units" defaults to the value set in the @mapdef directive.)

3.4.24 @text

Display text (including text from a file) in the ASCII output file.

KEYWORD	TYPE	DESCRIPTION
`text_file`	string	Full path for file containing text to display
`string`	string	Text to display (Note: use \ before all control characters)
`attribute`	string	"Magic" attribute containing value to display
`format`	string	Format of "magic" attribute to display Must use: **`direct`** ***`<cform>`*** (Note: ***`<cform>`*** is an optional string which contains a **printf** conversion format for the "magic" attribute. See Magic Attributes, (Section 1.10) for conversion formats for each attribute.)
`x_off`	integer	Column (character) offset wrt current anchor position
`y_off`	integer	Row (line) offset wrt current anchor position
For table display ...		
`table_name`	string	Table name (from @define_table) (Note: one point will be sampled for each call to the @table_site directive)
For grid display ...		
`grid_name`	string	Grid name (from @define_sample_grid) (Note: one point will be sampled for each location on the grid)
For list display ...		
`list_name`	string	List name (from @define_sample_list) (Note: one point will be sampled for each location in the list)
For table or list display ...		
`location_look_up`	string	File name for the location look up table (Note: ".ltab" suffix is optional)
`valid_time`	vtime	Valid time to match (see Format of Valid Time Identifiers, (Section 1.5)) (default is current **`valid_time`** from @set_source directive)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`justification`	string	Text justification One of: **`left centre right`**

Example 3.31

```
@text
{
  table_name    = T1;
  string        = |                |;
  x_off         = 18;
  y_off         = 0;
  justification = left;
}
```

Display the text string "| |" for each site (from the @table_site directive) of a table called "T1" (from the @define_table directive). The string will be displayed 18 columns (characters) to the right of the current "anchor position" in the table, and left justified. (Note that "anchor position" is set by @anchor.)

Example 3.32

```
@text
{
  text_file = $FPA/setup/pdf/texmet/examples/example_file;
  x_off    = 0;
  y_off    = 35;
}
```

Display text from the file `$FPA/setup/pdf/texmet/examples/example_file` 35 rows (lines) below the current "anchor position". (Note that "anchor position" is set by @anchor.)

3.4.25 @vector_presentation

Set the appearance of vector direction and speed displayed as text in the ASCII output file.

Note that each vector component requires a "**..._type**" keyword to indicate the appropriate section in the **vector_look_up** file to use in formatting the vector component to be displayed.

KEYWORD	TYPE	DESCRIPTION
vector_look_up	string	File name for the vector look up table (Note: ".vtab" suffix is optional)
calm_type	string	Type of display for "calm" vector speed One of: **none** do not display **value** display as value **text** display matching text
calm_justification	string	Justification for "calm" vector speed One of: **left centre right**
calm_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_calm	integer	Column (character) offset for "calm" vector speed
y_calm	integer	Row (line) offset for "calm" vector speed
direction_type	string	Type of display for vector direction One of: **none** do not display **value** display as value **text** display matching text
direction_justification	string	Justification for vector direction One of: **left centre right**
direction_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_dir	integer	Column (character) offset for vector direction
y_dir	integer	Row (line) offset for vector direction
speed_type	string	Type of display for vector speed One of: **none** do not display **value** display as value **text** display matching text
speed_justification	string	Justification for vector speed One of: **left centre right**
speed_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_spd	integer	Column (character) offset for vector speed
y_spd	integer	Row (line) offset for vector speed

Example 3.33

```
@vector_presentation
{
  vector_look_up          = waves;
  calm_type               = text;
  calm_justification      = centre;
  x_calm                  = 0;
  y_calm                  = 0;
  direction_type          = value;
  direction_justification = right;
  x_dir                   = 0;
  y_dir                   = 0;
  speed_type              = value;
  speed_justification     = centre;
  x_spd                   = 0;
  y_spd                   = 0;
}
```

Display waves according to the formats in the vector look up table `waves.vtab`. Calm waves will be formatted according to the section "calm text", centre justified and not offset. Wave directions will be formatted according to the section "direction value", right justified and not offset. Wave speeds will be formatted according to the section "speed value", left justified and not offset.

3.4.26 @verbose

Set "verbose" mode for TexMet product definition file development.

VALUE	DESCRIPTION
mode	One of: **yes** or **no**

Example 3.34

```
@verbose { yes }
```

Set the verbose mode on for extra diagnostic output. This can be useful if problems are encountered in developing a new TexMet product.

Example 3.35

```
@verbose { no }
```

Set verbose mode off for minimal diagnostic output.

3.4.27 @version

Set the TexMet version number.

Note that this must be the first directive!

VALUE	DESCRIPTION
texmet8.0	Current version number (Note: texmet6.0 is also accepted for backwards compatibility

Example 3.36

```
@version { texmet8.0 }
```

Set TexMet version number. Note that this must be the first directive!

3.4.28 @wind_presentation

Set the appearance of wind direction, wind speed and wind gust components displayed as text in the ASCII output file.

Note that each wind component requires a "**..._type**" keyword to indicate the appropriate section in the **wind_look_up** file to use in formatting the wind component to be displayed.

KEYWORD	TYPE	DESCRIPTION
wind_look_up	string	File name for the wind look up table (Note: ".wtab" suffix is optional
calm_type	string	Type of display for "calm" winds One of: **none** do not display **value** display as value **text** display matching text
calm_justification	string	Justification for "calm" wind One of: **left centre right**
calm_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_calm	integer	Column (character) offset for "calm" wind
y_calm	integer	Row (line) offset for "calm" wind
direction_type	string	Type of display for wind direction One of: **none** do not display **value** display as value **text** display matching text
direction_justification	string	Justification for wind direction One of: **left centre right**
direction_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_dir	real	Column (character) offset for wind direction
y_dir	real	Row (line) offset for wind direction
speed_type	string	Type of display for wind speed One of: **none** do not display **value** display as value **text** display matching text
speed_justification	string	Justification for wind speed One of: **left centre right**
speed_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_spd	integer	Column (character) offset for wind speed
y_spd	integer	Row (line) offset for wind speed
gust_type	string	Type of display for wind gust One of: **none** do not display **value** display as value **text** display matching text

KEYWORD	TYPE	DESCRIPTION
gust_above	real	Gust speed must be larger than wind speed by this amount to be displayed (in units of wind speed)
gust_justification	string	Justification for wind gust One of: **left centre right**
gust_format	string	A **printf** conversion format enclosed in quotes and containing the string "%s"
x_gust	integer	Column (character) offset for wind gust
y_gust	integer	Row (line) offset for wind gust

Example 3.37

```
@wind_presentation
{
  wind_look_up            = winds;
  calm_type               = none;
  gust_type               = none;
  direction_type          = value;
  direction_justification = right;
  x_dir                   = 0;
  y_dir                   = 0;
  speed_type              = value;
  speed_justification     = left;
  x_spd                   = 0;
  y_spd                   = 0;
}
```

Display winds according to the formats in the wind look up table `winds.wtab`. Calm winds and wind gusts will not be displayed. Wind directions will be formatted according to the section "direction value", and displayed at the current location, right justified and not offset. Wind speeds will be formatted according to the section "speed value", and displayed at the current location, left justified and not offset.

3.4.29 @write_time

Display current clock, issue or valid time as formatted text in the ASCII output file.

Note that the issue time is the **T0** time from the FPA, while the valid time is the current valid time from the @set_source directive.

Note that **`type`**, **`zone_type`** and **`format`** are always required.

KEYWORD	TYPE	DESCRIPTION
`x_off`	integer	Column (character) offset wrt current anchor position
`y_off`	integer	Row (line) offset wrt current anchor position

Note that **`type`**, **`zone_type`**, **`time_zone`**, **`language`** and **`format`** can be specified more than once in this directive, and the resulting formatted strings are concatenated before display.

`type`	string	Type of time to display One of: **`creation`** ***`<tadj>`*** creation time **`issue`** ***`<tadj>`*** **T0** time **`valid`** ***`<tadj>`*** valid time (Note: ***`<tadj>`*** is an optional adjustment to the time to display, in ***`hh`*** (hours) or ***`hh:mm`*** (hours:minutes) format.)
`zone_type`	string	Type of time zone to display One of: **`GMT`** or **`UTC`** Coordinated Universal time **`LMT`** or **`LCL`** Local time
`time_zone`	string	Display the equivalent time in another time zone. (Default is set by environment variable **$TZ**) See the directory `/usr/share/zoneinfo` for valid timezones.
`language`	string	Display time parameters in another language. (Default is set by environment variable **$LANG**) See the file `/usr/share/locale/locale.alias` for language options.

KEYWORD	TYPE	DESCRIPTION
`format`	string	Format for time to display Note that replacement characters from the system function **strftime** are allowed as part of the format. These include: `%a` abbreviated weekday name `%A` full weekday name `%b` abbreviated month name `%B` full month name `%c` local date and time `%d` day of the month (01-31) `%H` hour (24 hour clock)(00-23) `%I` hour (12 hour clock)(1-12) `%j` day of the year (001-366) `%m` month (01-12) `%M` minute (00-59) `%n` for new line `%p` local equivalent of AM/PM (Note: midnight is AM, noon is PM) `%P` local equivalent of am/pm `%t` for tab space `%u` weekday (1-7, Monday is 1) `%w` weekday (0-6, Sunday is 0) `%y` year without century (00-99) `%Y` year with century `%Z` time zone name, if any `%%` for "%" (Note: use \ before all control characters)

PRESENTATION KEYWORD	TYPE	DESCRIPTION
`justification`	string	Text justification One of: `left` `centre` `right`

Example 3.38

```
@write_time
{
  type = valid;
  zone_type = LMT;
  time_zone = EST5EDT;
  format = Data Valid %A, %B %d, %Y Time\=%H:%M %Z;
  justification = left;
  x_off = 0;
  y_off = 1;
}
```

Display valid time for the current TexMet product in the Eastern time zone, offset 1 row (line) below the current "anchor position", and left justified. This particular **`format`** might produce, for example:

```
Map Valid Satruday, May 10, 2008 13:00 EDT
```

(Note that "anchor position" is set by @anchor.)

Example 3.39

```
@write_time
{
  type       = valid;
  format     = "DATA VALID %b%d/%Y %H:%M";
  type       = valid +12;
  format     = " - %b%d/%Y %H:%M UTC;
  zone_type = UTC;
  justification = left;
}
```

Display valid time range (from current valid time to 12 hours from current valid time) for the current TexMet product in Coordinated Universal Time at the current "anchor position", and left justified. This particular **`format`** might produce, for example:

```
DATA VALID May10/2008 13:00 - May11/2008 01:00 UTC
```

(Note that "anchor position" is set by @anchor.)

3.4.30 Format of Look Up Tables

Look up tables allow conversion of FPA style attribute values or sampled values into text for display. They are accessed by the **look_up**, **case_look_up**, or **attribute_look_up keywords** in the @label and @sample_field directives. Files containing look up tables have a ".tab" suffix.

Look up tables contain lines with the following format:

```
    key(s)  :  symbol  :  text  :  presentation
    key(s)  :  symbol  :  text  :  presentation
    ...
```

where:

key(s) are one or more "value" strings separated by "white space" (such as blanks or tabs)
symbol is a graphic symbol file (not used by TexMet)
text is text to display (quotes not required)
presentation is a named presentation (not used by TexMet)

Note that one of symbol, text, or presentation must be non-blank, but that the others can be left blank if not required. Note also that lines without the **:** separators are considered comments, as are all parameters that follow the **!** or **#** characters in a line.

When an FPA field is sampled by one of the directives, each of the ***key(s)*** is tested against the resultant FPA "value" string. Each time one ***key*** is found within the FPA "value" string, one "hit" is registered, and the "best" match in the look up table is defined as the first line in the file where the maximum number of "hits" has been found. The text from that line would then used by the directive for displaying the sampled value. Two "special" ***key(s)*** are allowed.

The ***key*** ****missing**** will match against blank FPA "value" strings.
The ***key*** ****default**** will match against FPA "value" strings that register no "hits" in the look up table.

The type of display to use depends on each directive.

The @label @sample_field directives display the values sampled at each location with a text string given by text (if the keyword **format** is set to text).

The order of the lines in the look up table is extremely important. Lines with the most complicated "value" strings should always be included before lines with simpler "value" strings, and all lines with a single ***key*** should be included before lines with two ***key(s)***, and lines with two ***key(s)*** before lines with three ***key(s)***, and so on.

When several lines register the same number of "hits", the line that appears first in the look up table will be used. For example, the ***key(s)*** for rain "R" and light rainshowers "RW-" will both register one "hit" against the FPA "value" string for reduced visibility in light rainshowers "1-3RW-", since each ***key*** is found within the FPA "value" string. However, if the line for rain appears in the look up table before the line for light rainshowers, then the line for rain will ALWAYS be chosen first, and any special presentation for light rainshowers will never be accessed. Therefore, complicated "value" strings should always be listed before simple "value" strings.

The number of ***key(s)*** in a line does not affect the choice of a "best" match. One "hit" on a line with one ***key*** is equivalent to one "hit" on a line with several ***key(s)***. For example, a line containing the ***key(s)*** for rain and snow "R S" and a line containing the ***key*** for rain alone "R" will both register one "hit" against the FPA "value" string for light rain "R-". However, if the line containing the ***key(s)*** for rain and snow appears in the look up table before the line containing the ***key*** for rain alone, then the line containing the ***key(s)*** for rain and snow will ALWAYS be chosen first, and the corresponding presentation for mixed rain and snow will be used. Therefore, lines containing fewer ***key(s)*** should always be listed first.

Care must also be taken to ensure that the combination of look up table ***key(s)*** and FPA "value" strings does not register incorrect "hits". For example, a line containing the ***key*** for snow "S" will register a "hit" against the FPA "value" string for risk of a thunderstorm "RISK TRW". In this case, the "hit" occurs because of the "S" in "RISK"! It may be necessary to constrain the use of some "value" strings within the FPA to avoid generating unwanted matches in the TexMet application.

The following example of a look up table and examples of matches with FPA "value" strings is based on look up tables that are no longer used, but the basic principles are still applicable.

Example 3.40 Look up file format

```
1 TRW           : thunder : Thunderstorm   : trw1
2 RW            : rain    : Showers        : shower1
3 R             : rain    : Rain           : rain1
4 S             : snow    : Snow           : snow1
5 RSF           : rs_mix  : Rain and Snow  : mix1
6 RISK TRW      : thunder : Risk Thunder   : trw2
7 RW RISK TRW+  : rain    : Rain / Thunder : rain2
8 RW RISK TRW   : rain    : Rain / Thunder : rain2
```

Example 3.41 FPA "value" string to match "RW"

One "hit" will be registered by ***key(s)*** in lines 2,3,5,7,8. Therefore, line 2 will be used.
If text is required, the text string "Showers" will be displayed.

Example 3.42 FPA "value" string to match "RW-F"

One "hit" will be registered by ***key(s)*** in lines 2,3,7,8, and two hits by ***key(s)*** in line 5. Therefore, line 5 will be used.
If text is required, the text string "Rain and Snow" will be displayed.
(Note: an additional line with the ***key(s)*** "R F" or "RW F" may be necessary.)

Example 3.43 FPA "value" string to match "RISK TRW+"

One "hit" will be registered by ***key(s)*** in lines 1,2,3,4, and two hits by ***key(s)*** in lines 5,6,7,8. Therefore, line 5 will be used.
If text is required, the text string "Rain and Snow" will be displayed.
(Note: line 5 should be relocated at the end of the look up table to avoid this error!)

3.4.31 Format of Location Look Up Tables

Location look up tables allow for fixed or moving locations to be identified in tabular form, for easy access by the PSMet or SVGMet directives. They are accessed by the **location_look_up** keyword in the @label @sample_field and @sample_wind directives. Files containing location look up tables have a ".ltab" suffix.

Location look up tables contain lines with the following format:

```
    ident    lat lon     [valid_time]     [text_string]
    ident    lat lon     [valid_time]     [text_string]
    ...
```

where:

ident	is the location identifier (in string format)
lat	is the location latitude (in latlon format)
lon	is the location longitude (in latlon format)
valid_time	is an optional valid time (in vtime format)
text_string	is an optional text description (in string format)

Note that the **–** character can be used as a placeholder for any "missing" parameters.

Note that all parameters that follow the **!** or **#** characters in a line are considered comments.

The **location_ident** keyword from the directive is matched with the ***ident*** from a line of the location look up table to set the appropriate latitude and longitude for display or sampling. The optional ***valid_time*** is used to specify a moving location. If ***valid_time*** is included, then the line in the location look up table must also match the current valid time set by the **valid_time** keyword in the directive, or the valid time previously set in the @set_source directive. One "special" ident value is allowed. The ident ***default*** will match against any location identifier not found in the look up table.

3.4.32 Format of Wind Look Up Tables

Wind look up tables control the format for displaying wind direction, speed, and gusts as symbols or text, rather than as wind barbs. They are accessed by the **wind_look_up** keyword in the @wind_presentation directive. Files containing wind look up tables have a ".wtab" suffix.

Wind look up tables contain four sections. One section must specify the "calm" condition, since this may override both the direction and speed displays. The other sections specify the "direction" or "speed" or "gust" formats to use when displaying winds. Note that all wind directions are in degrees true, and all wind speeds gusts are converted to the units given by the "units" keyword of the sampling or display directive.

Note that all parameters that follow the **!** or **#** characters in a line are considered comments.

Note that the examples are taken from wind look up tables in the directories `$FPA/setup/pdf/texmet/examples`.

The "calm" section can include the following two formats:

```
calm    value
  min_spd max_spd round factor sdigits
  end
calm    text
  min_spd max_spd text
  end
```

where:

min_spd is the minimum wind speed for "calm" conditions
max_spd is the maximum wind speed for "calm" conditions
round is the value to round the wind speed to the nearest multiple of
factor is the power of 10 to divide the wind speed by
sdigits is the significant digits of the wind speed to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)

Example 3.44

```
calm value
    0 5 1 0 0
    end
```

Wind speeds from 0 to 5 are rounded to the nearest whole value, and all significant digits are displayed.

Example 3.45

```
calm text
    0 5 "CALM"
    end
```

Wind speeds from 0 to 5 are displayed using the text string "CALM".

The "direction" section can include the following two formats:

```
direction value
    min_spd max_spd round factor sdigits
    end
direction text
    min_dir max_dir text
    min_dir max_dir text
    ...
    end
```

where:

min_spd is the minimum wind speed
max_spd is the maximum wind speed
round is the value to round wind direction to the nearest multiple of
factor is the power of 10 to divide the wind direction by
sdigits is the significant digits of the wind direction to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place
min_dir is the minimum wind direction angle
max_dir is the maximum wind direction angle
text is the text to display (in double quotes)

Example 3.46

```
direction value
    0 999 10 1 2
    end
```

Wind directions for wind speeds from 0 to 999 are rounded to the nearest 10 degrees, divided by 10^1 and displayed as the nearest integer 2 digits long. (That is, a wind direction of "53" degrees would be rounded to "50", divided by 10 to give a value of "5" and displayed as "05".)

Example 3.47

```
 direction text
    337.5  22.5 "N"
     22.5  67.5 "NE"
    end
```

Wind directions from 337.5 to 22.5 degrees are displayed using the text string "N", wind directions from 22.5 to 67.5 degrees are displayed using the text string "NE".
Note that a wind direction of -22.5 is equivalent to a wind direction of 337.5!

The "speed" section can include the following two formats:

```
speed value
    min_spd max_spd round factor sdigits
    end
speed text
    min_spd max_spd text
    min_spd max_spd text
    ...
    end
```

where:

min_spd is the minimum wind speed
max_spd is the maximum wind speed
round is the value to round the wind speed to the nearest multiple of
factor is the power of 10 to divide the wind speed by
sdigits is the significant digits of the wind speed to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)

Example 3.48

```
speed value
    5 999 0 0 -1
    end
```

Wind speeds from 5 to 999 are displayed to 1 decimal place accuracy. (That is, wind speed of "24.783" would be displayed as "24.8".)

Example 3.49

```
speed text
     5 15 "LIGHT"
    15 30 "STRONG"
    end
```

Wind speeds from 5 to 15 are displayed using the text string "LIGHT", and wind speeds from 15 to 30 are displayed using the text string "STRONG".

The "gust" section can include the following two formats:

```
gust value
    min_gst max_gst round factor sdigits
    end
gust text
    min_gst max_gst text
    min_gst max_gst text
    ...
    end
```

where:

min_gst is the minimum wind gust

max_gst is the maximum wind gust

round is the value to round the wind gust to the nearest multiple of

factor is the power of 10 to divide the wind gust by

sdigits is the significant digits of the wind gust to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place

text is the text to display (in double quotes)

Example 3.50

```
gust value
    5 999 5 0 0
    end
```

Wind gusts from 5 to 999 are rounded to the nearest 5, and the nearest integer wind gust is displayed.

Example 3.51

```
gust text
    15  30 "STRONG GUSTS"
    30 999 "VERY STRONG GUSTS"
    end
```

Wind gusts from 15 to 30 are displayed using the text string "STRONG GUSTS", and wind gusts from 30 to 999 are displayed using the text string "VERY STRONG GUSTS".

3.4.33 Format of Vector Look Up Tables

Vector look up tables control the format for displaying vector speed and direction as text. They are set by the **`vector_look_up`** keyword in the @vector_presentation directive. Files containing vector look up tables have a ".vtab" suffix.

Vector look up tables contain three sections. One section must specify the "calm" condition, since this may override both the direction and speed displays. The other sections specify the "direction" or "speed" formats to use when displaying vector parameters.

Note that all parameters that follow the ! or # characters in a line are considered comments.

Note that examples are taken from vector look up tables in the directories `$FPA/setup/pdf/texmet/examples`.

The "calm" section can include the following two formats:

```
calm value
    min_spd max_spd round factor sdigits
    end
calm text
    min_spd max_spd text
    end
```

where:

`min_spd` is the minimum vector speed for "calm" conditions
`max_spd` is the maximum vector speed for "calm" conditions
`round` is the value to round the speed to the nearest multiple of
`factor` is the power of 10 to divide the vector speed by
`sdigits` is the significant digits of the vector speed to display
  0 - for the nearest integer
  >0 - for the nearest integer this long (with leading zeroes)
  <0 - for a real number with digits after the decimal place
`text` is the text to display (in double quotes)

Example 3.52

```
calm value
    0 5 1 0 0
      end
```

Vector speeds from 0 to 5 are rounded to the nearest whole value, and all significant digits are displayed.

Example 3.53

```
calm text
    0 7.5 "CALM"
    end
```

Vector speeds from 0 to 7.5 are displayed using the text string "CALM".

The "direction" section can include the following two formats:

```
direction value
    min_spd max_spd round factor sdigits
    end
direction text
    mid_dir max_dir text
    mid_dir max_dir text
    ...
    end
```

where:

min_spd is the minimum vector speed

max_spd is the maximum vector speed

round is the value to round the speed to the nearest multiple of

factor is the power of 10 to divide the vector direction by

sdigits is the significant digits of the vector direction to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place

text is the text to display (in double quotes)

Example 3.54

```
direction value
    0 999 10 1 2
    end
```

Vector directions for vector speeds from 0 to 999 are rounded to the nearest 10 degrees, divided by 10^1 and displayed as the nearest integer 2 digits long. (That is, a vector direction of "53" degrees would be rounded to "50", divided by 10 to give a value of "5", and displayed as "05".)

Example 3.55

```
direction text
    337.5 22.5 "N"
    22.5  67.5 "NE"
    end
```

Vector directions from 337.5 to 22.5 degrees are displayed using the text string "N", vector directions from 22.5 to 67.5 degrees are displayed using the text string "NE".
Note that a vector direction of -22.5 is equivalent to a vector direction of 337.5!

The "speed" section can include the following two formats:

```
speed value
    min_spd max_spd round factor sdigits
    end
speed text
    min_spd max_spd text
    min_spd max_spd text
    ...
    end
```

where:

min_spd is the minimum vector speed
max_spd is the maximum vector speed
round is the value to round the speed to the nearest multiple of
factor is the power of 10 to divide the vector speed by
sdigits is the significant digits of the vector speed to display
0 - for the nearest integer
>0 - for the nearest integer this long (with leading zeroes)
<0 - for a real number with digits after the decimal place
text is the text to display (in double quotes)

Example 3.56

```
speed value
    5 999 0 0 -1
    end
```

Vector speeds from 5 to 999 are displayed to 1 decimal place accuracy. (That is, a vector of "24.783" would be displayed as 24.8".)

Example 3.57

```
speed text
     5      15     "LIGHT"
    15      30     "STRONG"
    end
```

Vector speeds from 5 to 15 are displayed using the text string "LIGHT", vector speeds from 15 to 30 are displayed using the text string "STRONG".