

# **Forecast Production Assistant Version 8**

## **Interactive Graphics Editor Application Procedure Interface Specification**



Copyright © 1995-2016 Environment Canada

All Rights Reserved.

Forecast Production Assistant© Environment Canada

FPA© Environment Canada

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the "GNU Free Documentation License" is included as a preface in this Reference Manual.

# Contents

<b>1</b>	<b>Term Definitions</b>	<b>1</b>
<b>2</b>	<b>Control Functions</b>	<b>3</b>
2.1	GEAction . . . . .	4
2.2	GEAnimate . . . . .	8
2.3	GEConnect . . . . .	9
2.4	GEDepiction . . . . .	10
2.5	GEDisconnect . . . . .	22
2.6	GEEedit . . . . .	23
2.7	GEGuidance . . . . .	25
2.8	GEImagery . . . . .	28
2.9	GESequence . . . . .	30
2.10	GEStatus . . . . .	32
2.11	GEScratchpad . . . . .	35
2.12	GETimelink . . . . .	36
2.13	GEZoom . . . . .	38
<b>3</b>	<b>Function Prototypes</b>	<b>39</b>
3.1	cursor_proc . . . . .	39
3.2	message_proc . . . . .	41
3.3	status_proc . . . . .	42
<b>4</b>	<b>Attribute List (CAL) Functions</b>	<b>47</b>
4.1	Function Descriptions . . . . .	47



# GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright 2000, 2001, 2002, 2007, 2008 [Free Software Foundation, Inc.](#)

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## 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. Secondly, 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 world-wide, 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.

---



# Introduction

This manual is intended to help programmers integrate Ingrid into application programs and thus gain access to its meteorological field editing power. It is not intended to act as a programmer's manual as no details are given as to the internal workings of the program itself.

Ingrid is a series of library functions and has no user interface within itself. In order to maintain flexibility Ingrid is configured through the use of configuration and set up files. Configuration files contain information on all possible states and set up files contain information on the configuration specific to the particular application and is a subset of the configuration files.

---



# Chapter 1

## Term Definitions

This section discusses the organization of data within Ingrid and defines the terms used to identify the data sections.

### **Element**

The basic definition of data is the element. Elements are well known things such as pressure, temperature and so on.

### **Field**

Fields are defined as elements at a specific level, such as msl pressure and 850mb heights. Fields can be surfaces, or collections of areas, lines, link chains or points. Discrete areas can be fields such as weather, lines can be collections of fronts, link chains can be collections of storm tracks, and so on.

### **Group**

Fields are arranged into Groups. The definition of the groups is entirely arbitrary and is defined by the users. Most of the time the groups will have meaning so that there will be a 850mb group, a surface group and so on.

### **Chart**

A group at a specific time is called a Chart. Note that not every group needs to exist at any particular time so that there may or may not be a chart at any particular time.

### **Depiction**

The collection of charts for any specific time is called a Depiction.

### **Sequence**

The collection of depictions for every time is called the depiction Sequence.

---





## Chapter 2

# Control Functions

This section contains alphabetically-organized reference pages of Inged control functions. Control over Inged is exercised through the use of library function calls which are to be found in the **ingred.a** library (/usr/workstation/fpa/lib). It is the responsibility of the driving interface to ensure that commands are sent in a meaningful order. If the currently active field is a spline field and a discrete field editing command is sent it may be ignored and an invalid status returned, but there is no guarantee of this.

The routines are:

GEAction()	Miscellaneous graphics requests
GEAnimate()	Animation commands.
GEConnect()	Initialize the Inged library.
GEDepiction()	Commands applicable to individual depictions.
GEDisconnect()	Disconnect from the display and close all files.
GEEedit()	Edit the currently active element.
GEGuidance()	Controls guidance display.
GEImagery()	Controls the display of imagery.
GESequence()	Handles depiction sequence control requests.
GEStatus()	Request status information from Inged.
GEScratchpad()	Scratch pad control functions.
GETimelink()	Time linking control requests.
GEZoom()	Zoom control.

Three functions are required by Inged which must be supplied by the calling application during the call to GEConnect(). The prototype functions are to be found in section 2.

In the description of the control functions all keywords are given in **bold** and all parameters are given in *italic*. Items which are separated by a "|" are exclusive selections and those enclosed within square brackets are optional depending on the specifics of the command. Any parameter given as time has a format of "yyyy:jjj:hh" where yyyy is the year, jjj the julian day of the year, and hh the hour, all with leading zeros if required.

---

## 2.1 GEAction

Handles miscellaneous graphics requests.

### 2.1.1 SYNOPSIS

```
#include <ingred.h>
```

```
GEREPLY GEAction( command )  
char *command
```

### 2.1.2 DESCRIPTION

The **command** parameter consists of a string of key words. These are:

#### **FONT ADD**    *key description*

Add a font to the list of fonts that Ingrid will recognize.

<b>key</b>	The key name to use to identify the font in Ingrid commands.
<b>description</b>	The font in standard X font descriptor format.

#### **LINK PALETTE**    *chains labels guess*

Set the link chain colour coding scheme (use '-' for the default colour).

<b>chains</b>	Colour for link chains and nodes.
<b>labels</b>	Colour for labels.
<b>guess</b>	Colour of link node that is not attached to a feature.

#### **STATE DRAW\_MODE** **CONT** | **PPS** *smoothing*

Specifies the way in which lines will be drawn by the input device. The options are:

1. Continuous (**CONT**) in which lines are drawn by pressing and holding down the first mouse button, drawing the line, then releasing it when the line is to be terminated. The smoothing parameter is not used. This is the default mode on start-up.
2. Point-to-point smoothed (**PPS**) in which lines are drawn as smoothed line segments by clicking a point with the first mouse button, then moving to the next point and clicking again. This process continues until terminated by clicking with the second button. The amount of smoothing is given by the smoothing parameter as a percentage (0 - 100).

#### **STATE LINK\_MODE** **NORMAL** | **INTERMEDIATE**

Set the current linking mode.

#### **STATE MOVE\_MODE** **FIELD** | **FIELD\_AND\_LABELS**

Sets what is to be moved when the move edit command is issued. For any given object in the field either just the object itself or the object and any associated labels will be moved.

**STATE MERGE\_MODE FIELD | FIELD\_AND\_LABELS**

Sets what is to be merged into a field when the merge edit command is issued. Either the field by itself will be merged in or the field and its associated labels.

**STATE MODIFY\_MODE CONT | PPS | PUCK [size] smoothing**

Specifies the way in which lines will be modified when the editor is in modify mode for both line and area edit. The options are:

1. Continuous (**CONT**) in which lines are modified by pressing and holding down the first mouse button close to the existing line, drawing the new line position, then releasing it when the line is to be terminated. The **size** and **smoothing** parameters are not used.
2. Point-to-point smoothed (**PPS**) in which lines are drawn as smoothed line segments by clicking a point with the first mouse button, then moving to the next point and clicking again. This process continues until terminated by clicking with the second button. Otherwise it is the same as option 1. The amount of smoothing is given by the **smoothing** parameter as a percentage (0 - 100). The **size** parameter is not used.
3. **PUCK** modify. Lines are modified by being "pushed" by a puck shaped tool. The size of the puck is specified as a percentage (1 - 100) given by the **size** parameter and the amount of smoothing by the **smoothing** parameter as a percentage (0 - 100).

**STATE SMOOTHING\_AMOUNT amount**

Specifies the amount of smoothing, in number of patches, to apply to a field when the edit command SMOOTH is in effect. The **amount** must have a minimum of 1.0.

**STATE SPREAD amount**

Specifies the radius of influence of an edit on a continuous field. This is given as a percentage of the maximum radius of influence (**amount**) which is about one quarter of the visible map area. The minimum radius of influence depends on the resolution of the underlying grid. If a zoom operation is performed these rules are still in operation and the maximum radius of influence is still about one quarter of the visible map area (what the user sees on the screen). Note that if the zoom in is near the resolution of the grid then the maximum and minimum radius of influence may be the same.

**STATE STACK\_ORDER TOP | BOTTOM**

Specifies the stacking order of any new area drawn on a discrete field. If the order is given as **TOP** then all new areas will be drawn on top of any existing areas. If the order is given as **BOTTOM** then all new areas will be drawn underneath any existing areas.

**MAP BASE map noverlays**

Specifies the base **map** and the number of map overlays, **noverlays**, which are defined. Note that this command must come before any **MAP OVERLAY** commands.

**MAP OVERLAY overlay map | OFF [REREAD]**

Display map overlay found in the file **map** in overlay number **overlay** or turn off overlay number **overlay**. **overlay** must have a value between 1 and **noverlays** inclusive (see above). The optional parameter **REREAD** will force Ingrid to reread the overlay file and redisplay the overlay. This is necessary if the overlay file changes.

**MAP PALETTE land water coast border lat-lon forecast-area forecast-border**

Change the colours of the map background (use "-" to leave a colour alone).

**MODE NORMAL | SUSPEND**

Change the mode of operation. By default Ingred starts up in **START\_UP** mode. In this mode all interactive commands are blocked until the mode is changed by this command. The modes are:

<b>NORMAL</b>	Start full interactive mode.
<b>SUSPEND</b>	Clean up all ongoing processes and suspend all operations.

**DEFAULTS WINDS *windmin windmax units***

Sets the minimum and maximum wind speeds (in the given *units*) for display of winds in the interface. Winds outside this range may be displayed as circles.

**DEFAULTS TIMELINK *units label***

Sets the *units* (and a label for the units) for display of motion at timelink nodes.

**RASTER\_DUMP *format mode vtime* | **ACTIVE** *width height dfile* **PAD** | **NOPAD** [*overlay*]**

Print the given depiction to an output file.

- *format* sets the image encoding scheme and must be one of

<b>gif</b>	Graphical Interchange Format
<b>xgl</b>	Native graphics library format
<b>xwd</b>	X windows dump format

- *mode* determines the type of image required and must be one of

<b>bw</b>	Depiction is converted to black and white
<b>greyscale</b>	Depiction is converted greyscale
<b>colour</b>	Depiction output in original colours

- *vtime* is the valid time of the depiction wanted. This must be either

<b>vtime</b>	the valid time of the depiction or
<b>ACTIVE</b>	whatever is displaying on the screen will be output.

- *width* and *height* of the image are given in pixels.
- *dfile* is the name of the output file
- **PAD** | **NOPAD** Normally Ingred will create an output image with the same aspect ratio as the requested depiction and maximize the image size based on the given *width* and *height* (**NOPAD**). If **PAD** is specified the output image will be exactly *width* x
- *height* with padding added to the depiction image if the aspect ratio differs from that of the depiction.
- **overlay** (optional parameter) specifies the full path-name of a meta-file which will be overlaid on the output image. This can be used to provide labels for graphical output.

**REDISPLAY**

This command forces Ingred to redisplay its graphics. This command is most useful at the end of the mainline after all of the initialization has been done. If the user has resized the window during the initialization procedure, Ingred will not receive the proper resize event and the display may not appear correctly forcing the user to resize the window again to get the correct appearance. Using this command just before entering the main loop will avoid this problem.

**2.1.3 RETURN**

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

---

## 2.2 GEDanimate

Controls the animation mode of Ingrid.

### 2.2.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GEDanimate( **command** )

char \*command

### 2.2.2 DESCRIPTION

The **command** parameter consists of a string of key words. These are:

#### **DELAY *ms***

Set the delay interval in milliseconds between depictions in the animation.

#### **ENTER**

Enter animation mode. All of the other commands in this function are only recognized after the ENTER command is issued. The initial visibility state of the fields will be that of those in the depiction sequence.

#### **EXIT**

Exit animation mode.

#### **ZOOM**

Exit animation mode to perform a zoom or pan, and then re-enter animation mode preserving the current animation state.

#### **FIELD\_VISIBILITY *element level* ON | OFF**

Turn the visibility of the given field in the animating sequence on or off. Any field which is off will not be seen by the user.

#### **MODE DEPICT | INTERP**

Set the animation to be of either the depiction sequence currently being edited or of the sequence of interpolated depictions.

#### **SHOW NEXT | PREV | *valid\_time***

Show the next (**NEXT**) or previous (**PREV**) depiction/interpolation in the sequence or the depiction/interpolation valid at the specified time (*valid\_time*).

#### **START**

Start the animation using the selected **DELAY** and **MODE** settings.

#### **STOP**

Stop the animation.

---

### 2.2.3 RETURN

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

## 2.3 GEConnect

Initializes Ingred.

### 2.3.1 SYNOPSIS

```
#include <ingred.h>
```

```
GEREPLY GEConnect( app_context, w, wait_proc, msg_proc, status_proc)
XtAppContext app_context,
Widget w,
char *options,
void (*cursor_proc)();,
void (*msg_proc)();,
void (*status_proc)();
```

### 2.3.2 DESCRIPTION

This function initializes Ingred and passes the required start-up parameters. Note that by default Ingred starts up in a non-interactive mode. After the controlling interface has issued all required initialization commands, Ingred must be put into its interactive mode (see GEAction("MODE INTERACTIVE") command). The parameters are:

<b><i>app_context</i></b>	The application context of the interface.
<b><i>w</i></b>	The widget to use for display and input (Normally the drawing window).
<b><i>cursor_proc</i></b>	Function to call to request that the interface display a specified cursor.
<b><i>msg_proc</i></b>	Function to call when Ingred wants the interface to display a message.
<b><i>status_proc</i></b>	Function to call when Ingred needs to pass status information.

The three functions are supplied by the calling interface and must have specific parameter calling lists. For a detailed description see the function prototype descriptions at the end of this document.

### 2.3.3 RETURN

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

---

## 2.4 GEDepiction

Handles individual depiction commands.

### 2.4.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GEDepiction( **command**, **cal** )

char \***command**,

CAL **cal**

### 2.4.2 DESCRIPTION

The **cal** variable is an opaque structure of type CAL which contains the object attribute values for the given command. Values are set into and retrieved from the structure through the use of CAL\_...() functions. See the Complete Attribute List (CAL) section at the end of this document.

The **command** parameter consists of a string of key words. These are:

#### BACKGROUND **element level**

Set the background for all depictions for the field defined by **element** and **level**. The background is that as specified in the given CAL structure.

#### EDIT **command\_list**

Edit the currently active field. See below for details on the actual commands.

#### FIELD **element level**

Make the field, as given by **element** and **level**, editable and the group containing the field and the field itself active. If **element** and **level** are given as **NONE** then no field is active. Note that this is done automatically by the **HIDE** command.

#### FIELD\_VISIBILITY **element level ON | ON\_WHEN\_GROUP\_VISIBLE | OFF**

Change the visibility mode of the named field. If the mode is **ON** then the field will always be shown. If **ON\_WHEN\_GROUP\_VISIBLE** the field will only be visible when the group is visible. If the mode of a given field is **OFF** then the element will only be displayed if it is the element being edited. That is, an **ELEMENT** command will cause it to be shown.

#### GROUP **group**

Make the given **group** the active group without an active field.

#### GROUP\_VISIBILITY **group ON | OFF**

Change the visibility mode of the given group. If mode is **ON** then the group fields are always visible if the **FIELD\_VISIBILITY** is **ON\_WHEN\_GROUP\_VISIBLE**. If **OFF** then the group fields are only visible when the group is active.



**LABEL *process type* [ SET ]**

Label the currently active field. Insertions, deletions, modification and moves are done with the left mouse button. The commands are:

***process***

The labeling procedure. This must be one of:

**ADD**

Add a label to the depiction at the selected location. If there is an entry menu file associated with the given *type*, then ask the interface to put up an entry menu to allow modification of the label attributes. If *type* is followed by **SET** the attributes of a previously selected label are being set.

**DELETE**

Delete the label closest to the selected location.

**LIST *latitude longitude* | GO**

labeling is to be done for a list of points. If *latitude* and *longitude* are given Ingrid will cache the points and will not do the labeling until a **GO** command is received. The **GO** command will clear the internal cache. Note that *type* is not required with the **LIST *latitude longitude*** command.

**MODIFY**

Modify the attributes of the label of type *type* closest to the location selected on the depiction only if there is an entry menu file associated with the given *type*. If so, ask the interface to put up an entry menu to allow modification of the label attributes. If *type* is followed by **SET**, then the attributes of a previously selected label are being set.

**MOVE**

Move the label closest to the selected location.

**SHOW**

Send the interface the attributes of the label closest to the location selected on the depiction (See function prototype `status_proc()` **LABEL** command).

***type***

The keyword for the type of label as specified in the configuration files. See the configuration file documentation for details.

**SAMPLE *process type name font size colour***

Sample the currently active field. The commands are:

***process***

The sampling procedure. This must be one of:

**NORMAL**

The sampling is done on a point by point basis through use of the mouse by selecting the sample point on the map area.

**GRID *nx ny***

The sampling is to be done on a grid with the number of points in the x and y axis specified by *nx* and *ny*.

**LIST *latitude longitude* | GO**

Sampling is to be done for a list of points. If *latitude* and *longitude* are given, Ingrid will cache the points and will not do the sampling until a **GO** command is received. The

**GO** command will cause Ingrid to sample at the given points and then clear the internal cache. Note that none of the other parameters after **process** are required when the **LIST latitude longitude** command is issued.

**CANCEL**

Only the type and name are needed for this command. This command needs to be sent to Ingrid when changing the sample type, closing the graphical display of attributes, or leaving the sample state.

**type**

One of **AttribType**, **ValueType**, **FieldLabels**, **LinkNodes**, **WindType** or **WindCrossRef** followed by a valid **name**.

**name**

Each **type** has a list of valid item names which can be sampled (depending on the particular field).

For **AttribType** one of: **All** or an **attribute\_name** in the Config files.

For **ValueType** one of the Values in the Samples block in the Config files.

For **FieldLabels** use **FPA\_field\_labels**.

For **LinkNodes** use **FPA\_link\_nodes**.

For **WindType** one of the Winds in the Samples block in the Config files.

For **WindCrossRef** one of the Winds in the CrossRefs block in the Config files.

**font**

The font to use in displaying the sample.

**size**

The size of the font. If the numerical size is followed by a percent sign (%), this is the size as a percent of map size. If followed by a P, this is the size in points.

**colour**

Colour of the sample as a standard X colour name.

**SHOW**

Show the active depiction in the map window.

**HIDE**

Hide the active depiction. The active element is set to **NONE**.

## 2.4.3 EDIT COMMAND NOTES

Editing is performed on six types of objects; b-spline fields (such as pressure), vector fields (2-D b-spline fields, such as u/v components of wind), discrete areas (such as cloud areas), line objects (such as fronts), link chain objects (such as storm tracks), and point objects (such as station plots).

These editing commands are persistent, that is they remain in effect until suspended by an incoming request. Thus in order to maintain a given edit state, the edit command must be re-issued after any other request.

The recognized **font** names are **Helvetica**, **HelveticaBold**, **TimesRoman** and **TimesRomanBold**.

Line styles are specified as width.style where the allowable values of width are: **thin**, **medium** and **thick**, and the allowable values of style are: **solid**, **dashed**, **dotted**, **dash\_dot**, **dash\_dot\_dot**, **long\_dash**, **centre\_dash** and **centre\_dash\_dash**.

### 2.4.3.1 SPLINE FIELD EDIT COMMANDS

The area of influence of spline edits are determined by the current SPREAD setting. See GEAction for details.

#### **EDIT AREA [ PRESET\_OUTLINE <metafile> ]**

#### **EDIT AREA [ TRANSLATE | ROTATE | COPY ]**

The command given without parameters puts Ingred into move mode, which allows the user to relocate a portion of the spline surface.

Once in move mode, Ingred will request that the user draw an outline which defines the area on the spline surface that will be moved. Optionally the command **PRESET\_OUTLINE** can be issued using the special programming macros FpaEditorLastDrawnBoundary, FpaEditorLastMovedBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once an outline has been drawn, issuing the command with **TRANSLATE**, **ROTATE** or **COPY** will cause Ingred to go into a state that allows the area on the spline surface to be translated, rotated or copied. (Note that the last moved boundary can be used to translate and then rotate the same area of the spline surface.)

#### **EDIT DRAG**

Allows the user to perform a drag edit operation.

#### **EDIT MERGE [ FETCH source sub-source run-time valid-time element level ]**

The command given without parameters puts Ingred into merge mode, which allows the user to merge a portion of a specified spline field with the current spline field.

The command with **FETCH** (and parameters) specifies the source field to use for the merge. Any parameter not relevant for a particular source must be place held with a dash (-). Once completed, Ingred will return to the state assumed upon receiving the **MERGE** command without parameters.

#### **EDIT MERGE [ DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]**

#### **EDIT MERGE [ MERGE | TRANSLATE | ROTATE ]**

After a **MERGE FETCH** command has been issued, Ingred will request that the user draw an outline which defines the area from the fetched spline surface to be merged with the existing spline surface. Drawing an outline will generate a **DRAW\_OUTLINE** command. Optionally the user can choose a preset outline which will generate a **PRESET\_OUTLINE** command using the special programming macros FpaEditorLastDrawnBoundary, FpaEditorLastMovedBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once an outline has been drawn, issuing the command with **MERGE**, **TRANSLATE** or **ROTATE** will cause Ingred to go into a state that allows the area on the spline surface to be merged in place, or translated/rotated and then merged.

#### **EDIT POKE delta**

Allows the user to perform a poke edit operation. The *delta* defines the outer limit of a modification region centred on the poke location.

**EDIT SMOOTH**

Allows the user to perform a smooth edit operation. Note that the amount of smoothing is given by the current value of the "GAction STATE SMOOTHING\_AMOUNT *<value>*" command.

**EDIT STOMP *delta interior-delta***

Allows the user to perform a stomp edit operation. The stomp is like the poke but over an area instead of centred at a point. The *delta* defines the outer limit of a modification region around the area, and the *interior-delta* allows control of modification within the area.

**2.4.3.2 VECTOR FIELD EDIT COMMANDS**

The area of influence of vector field (two component spline field) edits are determined by the current **SPREAD** setting. See GAction for details.

**EDIT AREA [ PRESET\_OUTLINE *<metafile>* ]****EDIT AREA [ TRANSLATE | ROTATE | COPY ]**

The command given without parameters puts Ingred into move mode, which allows the user to relocate a portion of the two component spline surface.

Once in move mode, Ingred will request that the user draw an outline which defines the area on the two component spline surface that will be moved. Optionally the command **PRESET\_OUTLINE** can be issued using the special programming macros `FpaEditorLastDrawnBoundary`, `FpaEditorLastMovedBoundary` or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once an outline has been drawn, issuing the command with **TRANSLATE**, **ROTATE** or **COPY** will cause Ingred to go into a state that allows the area on the two component spline surface to be translated, rotated or copied. (Note that the last moved boundary can be used to translate and then rotate the same area of the two component spline surface.)

**EDIT DRAG**

Allows the user to perform a drag edit operation.

**EDIT MERGE [ FETCH *source sub-source run-time valid-time element level* ]**

The command given without parameters puts Ingred into merge mode, which allows the user to merge a portion of a specified two component spline field with the current two component spline field.

The command with **FETCH** (and parameters) specifies the source field to use for the merge. Any parameter not relevant for a particular source must be place held with a dash (-). Once completed, Ingred will return to the state assumed upon receiving the **MERGE** command without parameters.

**EDIT MERGE [ DRAW\_OUTLINE | PRESET\_OUTLINE *<metafile>* ]****EDIT MERGE [ MERGE | TRANSLATE | ROTATE ]**

After a **MERGE FETCH** command has been issued, Ingred will request that the user draw an outline which defines the area from the fetched two component spline surface to be merged with the existing two component spline surface. Drawing an outline will generate a **DRAW\_OUTLINE** command. Optionally the user can choose a preset outline which will generate a **PRESET\_OUTLINE** command

using the special programming macros `FpaEditorLastDrawnBoundary`, `FpaEditorLastMovedBoundary` or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once an outline has been drawn, issuing the command with **MERGE**, **TRANSLATE** or **ROTATE** will cause Ingred to go into a state that allows the area on the two component spline surface to be merged in place, or translated/rotated and then merged.

#### **EDIT POKE *delta***

Allows the user to perform a poke edit operation. The *delta* defines the outer limit of a modification region centred on the poke location. (The poke edit can be applied to either or both of the magnitude and direction components of the spline surface.)

#### **EDIT SMOOTH**

Allows the user to perform a smooth edit operation. Note that the amount of smoothing is given by the current value of the "GEAction STATE SMOOTHING\_AMOUNT <value>" command.

#### **EDIT STOMP *delta interior-delta***

Allows the user to perform a stomp edit operation. The stomp is like the poke but over an area instead of centred at a point. The *delta* defines the outer limit of a modification region around the area, and the *interior-delta* allows control of modification within the area. (The stomp edit can be applied to either or both of the magnitude and direction components of the spline surface.)

### **2.4.3.3 AREA AND WIND FIELD EDIT COMMANDS**

#### **EDIT DRAW [ SET ]**

Allows the user to draw a discrete area. The area is either drawn on top of or underneath of any existing areas according to the setting **STACK\_ORDER** state (see GEAction). The line attributes are set with the current *cal*.

Once in draw mode, issuing this command with the optional parameter **SET** sets the attributes of the last area drawn, but not posted, to the returned *cal*. The editor will automatically return to draw mode after this command.

#### **EDIT DRAW\_HOLE [ PRESET\_OUTLINE <metafile> ]**

Allows the user to draw holes in an area. Ingred will send a message asking the user to select the area in which to draw the holes. Once selected Ingred then prompts the user to draw the holes.

Optionally the command **PRESET\_OUTLINE** can be issued using the programming macro `FpaEditorLastDrawnHole` or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

#### **EDIT DIVIDE [ SET | REJOIN ]**

Allows the user to divide an existing discrete area. Ingred will send a message asking the user to select the area to divide. Once selected Ingred then prompts the user to draw the dividing line.

Once the dividing line is drawn, the command **SET** will set new attributes for the first or second portions of the divided area. The area attributes are set using the returned *cal*. If the **REJOIN** command is given, then the selected portion will be joined to the adjacent portion (that is the previous divide operation will be undone).

**EDIT MERGE [ FETCH *source sub-source run-time valid-time element level* ]**

The command given without parameters puts Ingred into merge mode, which allows the user to merge areas from a specified field with the current field.

The command with **FETCH** (and parameters) specifies the source field to use for the merge. Any parameter not relevant for a particular source must be place held with a dash (-). Once completed, Ingred will return to the state assumed upon receiving the **MERGE** command without parameters.

**EDIT MERGE [ SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]****EDIT MERGE [ MERGE | TRANSLATE | ROTATE ]**

After a **MERGE FETCH** command has been issued, Ingred will request that the user choose areas to merge with the existing field. Areas can be selected or deselected with the mouse, or all the areas in the merge field can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed areas will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro FpaEditorLast-DrawnBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once the desired areas have been chosen, issuing the command with **MERGE**, **TRANSLATE** or **ROTATE** will cause Ingred to go into a state that allows the chosen areas to be merged in place, or translated/rotated and then merged.

**EDIT MODIFY****EDIT MODIFY [ DELETE | DELETE HOLE ]****EDIT MODIFY [ SET | STACK ] [ TOP | UP | DOWN | BOTTOM ]**

The command given without parameters puts Ingred into modify mode, which allows the user to modify area boundaries, dividing lines or holes for areas in the current field. Ingred will send a message asking the user to select the area feature to modify.

Once an area feature has been chosen, the user can modify the shape of the feature.

Once an area feature has been chosen, the command **DELETE** will delete the area containing the currently selected feature.

Once an area hole has been chosen, the command **DELETE HOLE** will delete the area hole.

Once the interior of the area boundary has been chosen, the command **SET** will set the attributes for the chosen area. The attributes are provided in the returned *cal* structure.

Once an area feature has been chosen, the command **STACK** will change the stacking order of the chosen area. The area may be moved to the top of the stacking order (**TOP**), moved up one position (**UP**), moved down one position (**DOWN**) or moved to the bottom of the stacking order (**BOTTOM**).

**EDIT MOVE [ SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]****EDIT MOVE [ TRANSLATE | ROTATE | COPY | CUT ]**

The command given without parameters puts Ingred into move mode, which allows the user to relocate areas in the current field.

Once in the move mode Ingred will request that the user choose areas to move. Areas can be selected or deselected with the mouse, or all the areas can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed areas will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro `FpaEditorLast-DrawnBoundary` or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once the desired areas have been chosen, issuing the command with **TRANSLATE**, **ROTATE**, **COPY** or **CUT** will cause Ingred to go into a state that allows the chosen areas to be translated and/or rotated, or Ingred will copy or cut the areas.

#### 2.4.3.4 LINE FIELD EDIT COMMANDS

In any given line field there may be more than one type of line defined. We could, for example, have cold and warm fronts. For those commands which set the line type through the *cal* structure, the specific items that must be set are: *CALcategory*, *CALuserlabel* and *CALautolabel*. See the chapters on the *cal* structure at the end of this document.

##### **EDIT DRAW [ SET ]**

Allows the user to draw a line feature. The line attributes are set with the current *cal*.

Once in draw mode, issuing this command with the optional parameter **SET** sets the attributes of the last line drawn, but not posted, to the returned *cal*. The editor will automatically return to draw mode after this command.

##### **EDIT FLIP [ FLIP | REVERSE | FLIP-REVERSE ]**

Allows the user to flip or reverse a line feature. The line to flip is selected with the mouse. Once the line is selected, issuing the command with the optional parameters will result in the currently selected line being flipped, reversed end to end or flipped and reversed.

##### **EDIT JOIN**

Allows the user to join two line features, or break a line feature into two parts.

The two lines to join are chosen by the mouse, and must be of the same type. The end of the first selected line will be joined to the nearest point of the second selected line. The portion of the second line joined will be that part which continues in the same direction as the first line.

The user can also choose a line to break into two sections at a location defined by the mouse.

##### **EDIT MERGE [ FETCH *source sub-source run-time valid-time element level* ]**

The command given without parameters puts Ingred into merge mode, which allows the user to merge lines from a specified field with the current field.

The command with **FETCH** (and parameters) specifies the source field to use for the merge. Any parameter not relevant for a particular source must be place held with a dash (-). Once completed, Ingred will return to the state assumed upon receiving the **MERGE** command without parameters.



---

**EDIT MERGE [ SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]**

**EDIT MERGE [ MERGE | TRANSLATE | ROTATE ]**

After a **MERGE FETCH** command has been issued, Ingred will request that the user choose lines to merge with the existing field. Lines can be selected or deselected with the mouse, or all the lines in the merge field can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed lines will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro FpaEditorLast-DrawnBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once the desired lines have been chosen, issuing the command with **MERGE**, **TRANSLATE** or **ROTATE** will cause Ingred to go into a state that allows the chosen lines to be merged in place, or translated/rotated and then merged.

**EDIT MODIFY [ DELETE | SET ]**

The command given without parameters puts Ingred into modify mode, which allows the user to modify line features in the current field. Ingred will send a message asking the user to select the line feature to modify.

Once a line feature has been chosen, the user can modify the shape of the feature.

Once a line feature has been chosen, the command **DELETE** will delete the currently selected feature.

Once a line feature has been chosen, the command **SET** will set the attributes for the chosen line feature. The attributes are provided in the returned *cal* structure.

**EDIT MOVE [ SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]**

**EDIT MOVE [ TRANSLATE | ROTATE | COPY | CUT ]**

The command given without parameters puts Ingred into move mode, which allows the user to relocate line features in the current field.

Once in the move mode Ingred will request that the user choose line features to move. Line features can be selected or deselected with the mouse, or all the line features can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed line features will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro FpaEditorLast-DrawnBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once the desired line features have been chosen, issuing the command with **TRANSLATE**, **ROTATE**, **COPY** or **CUT** will cause Ingred to go into a state that allows the chosen line features to be translated and/or rotated, or Ingred will copy or cut the line features.

---



### 2.4.3.5 LINK CHAIN FIELD EDIT COMMANDS

Link chains are defined by a set of nodes at a sequence of times separated by *interp\_delta* minutes, which is set in the **GESequence INITIALIZE EDIT *interp\_delta*** command.

Each link node has a type which determines whether it has attributes and/or a location, or whether the attributes and/or location are automatically interpolated. The following types of nodes are allowed:

<b>normal</b>	has both attributes and a location
<b>control</b>	has a location but attributes are interpolated
<b>floating</b>	has attributes but a location is interpolated
<b>interp</b>	both attributes and location are interpolated

**EDIT ADD *reference-time time-delta normal|control***

**EDIT ADD END\_CHAIN**

Allows the user to create a link chain node by node, beginning with a **normal** or **control** node at the *reference-time*, and with the next node occurring *time-delta* minutes thereafter.

The command with **END\_CHAIN** ends and saves the link chain.

**EDIT MERGE [ **FETCH *source sub-source run-time valid-time element level*** ]**

The command given without parameters puts Ingred into merge mode, which allows the user to merge link chains from a specified field with the current field.

The command with **FETCH** (and parameters) specifies the source field to use for the merge. Any parameter not relevant for a particular source must be place held with a dash (-). Once completed, Ingred will return to the state assumed upon receiving the **MERGE** command without parameters.

**EDIT MERGE [ **SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE *<metafile>*** ]**

**EDIT MERGE [ **MERGE | TRANSLATE | ROTATE** ]**

After a **MERGE FETCH** command has been issued, Ingred will request that the user choose link chains to merge with the existing field. Link chains can be selected or deselected with the mouse, or all the link chains in the merge field can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed link chains will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro FpaEditorLast-DrawnBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once the desired link chains have been chosen, issuing the command with **MERGE**, **TRANSLATE** or **ROTATE** will cause Ingred to go into a state that allows the chosen link chains to be merged in place, or translated/rotated and then merged.

**EDIT MODIFY [ **DELETE | SET** ]**

The command given without parameters puts Ingred into modify mode, which allows the user to modify link chain in the current field. Ingred will send a message asking the user to select the link chain to modify.

Once a link chain has been chosen, the command **DELETE** will delete the currently selected link chain.

Once a link chain has been chosen, the command **SET** will set the attributes for the chosen link chain. The attributes are provided in the returned **cal** structure.

**EDIT MOVE [ SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]**

**EDIT MOVE [ TRANSLATE | ROTATE | COPY | CUT ]**

The command given without parameters puts Ingred into move mode, which allows the user to relocate link chains in the current field.

Once in the move mode Ingred will request that the user choose link chains to move. link chains can be selected or deselected with the mouse, or all the link chains can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed link chains will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro FpaEditorLast-DrawnBoundary or with a **metafile** reference. With the metafile reference the first outline in the **metafile** will be used.

Once the desired link chains have been chosen, issuing the command with **TRANSLATE**, **ROTATE**, **COPY** or **CUT** will cause Ingred to go into a state that allows the chosen link chains to be translated and/or rotated, or Ingred will copy or cut the link chains.

**EDIT NODES MOVE | DELETE | SHOW | MODIFY normal|control|floating|interp [ SET ]**

Allows the user to choose a link chain and a node on the link chain, and then to move, delete or show the attributes of the node, or to modify the type and optionally set the attributes of the node.

#### 2.4.3.6 POINT FIELD EDIT COMMANDS

In any given point field there may be more than one type of point defined. We could, for example, have maximum and minimum temperature points. For those commands which set the point type through the **cal** structure, the specific items that must be set are: CALcategory, CALuserlabel and CALautolabel. See the chapters on the **cal** structure at the end of this document.

**EDIT DRAW type [ SET ]**

Allows the user to add new points. The point types are defined in the configuration files, and the attribute values of the point are set with the current **cal**.

Once in draw mode, issuing this command with the optional parameter **SET** sets the attributes of the last point added, but not posted, to the returned **cal**. The editor will automatically return to draw mode after this command.

**EDIT MERGE [ FETCH source sub-source run-time valid-time element level ]**

The command given without parameters puts Ingred into merge mode, which allows the user to merge points from a specified field with the current field.

The command with **FETCH** (and parameters) specifies the source field to use for the merge. Any parameter not relevant for a particular source must be placed with a dash (-). Once completed, Ingred will return to the state assumed upon receiving the **MERGE** command without parameters.

**EDIT MERGE [ SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]**

**EDIT MERGE [ MERGE | TRANSLATE | ROTATE ]**

After a **MERGE FETCH** command has been issued, Ingred will request that the user choose points to merge with the existing field. Points can be selected or deselected with the mouse, or all the points in the merge field can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed points will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro FpaEditorLast-DrawnBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once the desired points have been chosen, issuing the command with **MERGE**, **TRANSLATE** or **ROTATE** will cause Ingred to go into a state that allows the chosen points to be merged in place, or translated/rotated and then merged.

**EDIT MODIFY type [ DELETE | SET ]**

The command given without parameters puts Ingred into modify mode, which allows the user to modify a point of *type* (as defined in the configuration files) in the current field. Ingred will send a message asking the user to select the point to modify.

Once a point has been chosen, the command **DELETE** will delete the currently selected point.

Once a point has been chosen, the command **SET** will set the attributes for the chosen point. The attributes are provided in the returned *cal* structure.

**EDIT MOVE [ SELECT\_ALL | DRAW\_OUTLINE | PRESET\_OUTLINE <metafile> ]**

**EDIT MOVE [ TRANSLATE | COPY | CUT ]**

The command given without parameters puts Ingred into move mode, which allows the user to relocate points in the current field.

Once in the move mode Ingred will request that the user choose points to move. points can be selected or deselected with the mouse, or all the points can be chosen with the **SELECT\_ALL** option.

Optionally the user can choose to use an outline within which all fully enclosed points will be chosen. Drawing an outline will generate a **DRAW\_OUTLINE** command. Choosing a preset outline will generate a **PRESET\_OUTLINE** command using the special programming macro FpaEditorLast-DrawnBoundary or with a *metafile* reference. With the metafile reference the first outline in the *metafile* will be used.

Once the desired points have been chosen, issuing the command with **TRANSLATE**, **COPY** or **CUT** will cause Ingred to go into a state that allows the chosen points to be translated, or Ingred will copy or cut the points.

## 2.4.4 RETURN

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

## 2.5 GEDisconnect

Ingred does a cleanup and disconnects from the display window.

### 2.5.1 SYNOPSIS

```
#include <ingred.h>
```

```
GEREPLY GEDisconnect();
```

### 2.5.2 DESCRIPTION

This function causes Ingred to clean up the state of the graphics, save all relevant data and disconnect from all graphics devices.

### 2.5.3 RETURN

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

---

## 2.6 GEdit

Handles commands for certain editing procedures.

### 2.6.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GEdit( **command** )

char \*command

### 2.6.2 DESCRIPTION

Issues edit commands valid for the currently active depiction element, guidance annotation field or scratch pad. These commands do an auto-setback of the editing state and do not require the edit command to be re-issued after they are used.

The **command** parameter consists of the following key words:

#### **BREAK**

Break a line in the currently active editing field into two sections.

#### **CANCEL**

Cancel the pending edit. This applies to those edits which require multiple inputs before being acted upon.

#### **CLEAR**

Clear the currently active editing field. Note that if the current state for a given field is **LABEL**, then all labels will be cleared from the field.

#### **DRAW\_DONE**

When Ingrid is in the point-by-point draw state, this command will terminate the draw and accept the line defined by the points.

#### **END\_CHAIN**

Terminate the input of a timelink chain.

#### **JOIN**

Join two lines in the currently active editing field.

#### **MODIFY\_CONFIRM**

When Ingrid is in the point-by-point modify state, this command will confirm the final modified display of the feature.

#### **PASTE**

Paste a copied feature into the currently active editing field.

---

**REJOIN**

Rejoin two sections of a divided area. The attributes will be taken from the first area chosen.

**RESUME**

Forces the editor to pick up where it left off. This is the command to issue if there is unrelated windowing activity going on.

**UNDO**

Undo the previous edit.

**UPDATE**

Post all pending edits. Note that after this command an undo is no longer possible.

**2.6.3 RETURN**

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

---

## 2.7 GEGuidance

Controls the display and annotation of guidance.

### 2.7.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GEGuidance( **command** )

char \*command

### 2.7.2 DESCRIPTION

This function controls the display of guidance fields. The **command** parameter consists of a string of key words. These are:

**EDIT FIELD** *field\_id* [**key** [**value**]]

Modify the appearance of a guidance field with field identifier *field\_id* to that given by **key**. The **key** values are: **ERASE**, **RESTORE**, **COLOR** and **STYLE**. **ERASE** removes the field from the display. **RESTORE** displays the field in the default display state. **COLOR** changes the colour of the contours. **STYLE** changes the line style and thickness of the contours (see Note 3 above for the format). The colour, style and thickness values are given by **value**. If the **FIELD** command is given without any specifiers, then the current valid date and time are displayed by Ingrid.

**LABEL process** *field\_id* **cmd**

Label the field identified by *field\_id*. Insertions are done with the left mouse button. The commands are:

**process**

The labeling procedure. This must be one of:

**ADD** - Add the label to the field at the location selected with the cursor.

**field\_id**

The guidance field identifier (See GEGuidance).

**cmd**

One of **VALUE**, which labels the field with the field value, or **WIND** which labels the field with a wind barb appropriate to the field, as set in the Fields config file.

**FIELD\_REGISTER** *field\_id* **element** **level** **source** **sub-source**[- **run\_time**]-

Associate the field as defined by **element**, **level**, **source** (and **sub-source**) and **run\_time** with the field identifier *field\_id*. Depending on the **source**, **sub-source** and **run\_time** may not be applicable and are then replaced by a dash ("-"). The field identifier can be any arbitrary string. If a field has already been associated with the given identifier when this command is issued, then all references to the existing field are eliminated from memory and replaced with the new association. Note that the same field can be registered more than once (this would allow showing the same field at more than one valid time).

---

**FIELD\_DEREGISTER *field\_id***

Disassociate the field associated with the given *field\_id*. All references to the field are eliminated from memory.

**FIELD\_VISIBILITY *field\_id valid\_time* ON|OFF**

Set the field visibility, as identified by the given *field\_id*, at the specified *valid\_time* to be either **ON** (showing) or **OFF** (not showing). The **FIELD** command above must have been issued previously in order to obtain the *field\_id*. Note that this command only sets the visibility state and a **SHOW** command is required.

**SAMPLE *process field\_id valid\_time type name font size colour***

Sample the field as given by field identifier *field\_id* at valid time *valid\_time*. The remaining commands are:

***process***

The sampling procedure. This must be one of:

**NORMAL**

The sampling is done on a point by point basis through use of the mouse by selecting the sample point on the map area.

**GRID *nx ny***

The sampling is to be done on a grid with the number of points in the x and y axis specified by *nx* and *ny*.

**LIST *latitude longitude* | GO**

Sampling is to be done for a list of points. If *latitude* and *longitude* are given, Ingrid will cache the points and will not do the sampling until a **GO** command is received. The **GO** command will cause Ingrid to sample at the given points and then clear the internal cache. Note that none of the other parameters after *process* are required when the **LIST *latitude longitude*** command is issued.

**CANCEL**

Only the first four terms are needed for this command. This command needs to be sent to Ingrid when changing the sample type, closing the graphical display of attributes, or leaving the sample state.

***type***

One of **AttribType**, **ValueType**, **FieldLabels**, **LinkNodes**, **WindType** or **WindCrossRef** followed by a valid *name*.

***name***

Each *type* has a list of valid item names which can be sampled (depending on the particular field).

For **ValueType** one of the Values in the Samples block in the Config files.

For **FieldLabels** use **FPA\_field\_labels**.

For **LinkNodes** use **FPA\_link\_nodes**.

For **WindType** one of the Winds in the Samples block in the Config files.

For **WindCrossRef** one of the Winds in the CrossRefs block in the Config files.

***font***

The font to use in displaying the sample.



***size***

The size of the font. If size is followed by a percent sign (%) the size is a percent of map size. If followed by a **P** the size is in points.

***colour***

Colour of the sample as a standard X colour name.

**SHOW**

Show all guidance fields for which the field visibility is set to **ON** and hide all those set to **OFF**.

**HIDE**

Hide all guidance fields irrespective of the state of their field visibility.

**2.7.3 RETURN**

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

---

## 2.8 GEImagery

Handles commands for displaying imagery.

### 2.8.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GEImagery( **command** )

char \*command

### 2.8.2 DESCRIPTION

Issues commands to display imagery at a given valid time. There are four image planes into which images can be displayed. There are string macros defined for these planes: UNDERLAY\_DISPLAY\_PLANE, SATELLITE\_DISPLAY\_PLANE, RADAR\_DISPLAY\_PLANE and OVERLAY\_DISPLAY\_PLANE. The stacking order of the planes are in the given order of the macros with the underlay on the bottom of the stack. The satellite and radar display planes can be blended and are affected by the blending command discussed elsewhere. Images are identified by an image tag. This is the tag provided by the image access library.

The **command** parameter consists of a string of key words. These are:

**ACTIVE tag time**

Set the given image as identified by the given **tag** and **time** as the active image to which actions (like sampling) will be applied.

**ANIMATE ON | OFF speed**

Turn animation on or off and set the animation speed with the on command. The speed is given as a percentage of between 0 and 100 percent.

**BLEND ON | OFF blend\_percentage**

Turn the blending of images on or off and set the percentage blend. The blending ratio has a value of 0 to 100 and is the percentage of the RADAR\_DISPLAY\_PLANE to be blended with the SATELLITE\_DISPLAY\_PLANE. A value of 100 will result in the radar image completely covering the satellite image with no image blending.

**DISPLAY plane tag time**

Display the image identified by **tag** at time **time** in the given **plane**. Note that **plane** must be the string defined in one of the macros specified above.

**REMOVE tag**

Remove the image identified by **tag** from whatever image plane it is in.

**SETLUT tag lut**

Set the look up table for the image identified by **tag** to **lut**. Note that **lut** of **DEFAULT** will use the first look up table in the list.

**SAMPLE** *process type item font size colour*

Sample the field as given by field identifier *field\_id* at valid time *valid\_time*. The remaining commands are:

**process**

The sampling procedure. This must be one of:

**NORMAL**

The sampling is done on a point by point basis through use of the mouse by selecting the sample point on the map area.

**GRID** *nx ny*

The sampling is to be done on a grid with the number of points in the x and y axis specified by *nx* and *ny*.

**LIST** *latitude longitude* | **GO**

Sampling is to be done for a list of points. If *latitude* and *longitude* are given, Ingrid will cache the points and will not do the sampling until a **GO** command is received. The **GO** command will cause Ingrid to sample at the given points and then clear the internal cache. Note that none of the other parameters after **process** are required when the **LIST** *latitude longitude* command is issued.

**type**

For now only **Default** is recognized.

**item**

For images which are associated with several different items (for example URP data files where a given array element can be associated with ZR, rainfall rate or snowfall rate) this specifies which item to sample. For images with no defined *item*, **Default** must be used.

**font**

The font to use in displaying the sample.

**size**

The size of the font. If size is followed by a percent sign (%) the size is a percent of map size. If followed by a **P** the size is in points.

**colour**

Colour of the sample as a standard X colour name.

**SHOW**

Any **BLEND**, **DISPLAY** or **REMOVE** commands will not be executed until a **SHOW** command is received.

**HIDE**

Hide all image planes.

## 2.8.3 RETURN

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

## 2.9 GESequence

Processes requests dealing with the depiction sequence.

### 2.9.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GESequence( **command** )

char \*command

### 2.9.2 DESCRIPTION

This group handles requests which deals with the depiction sequence. Note that Ingrid does not assume anything about what is required upon start-up. It is the responsibility of the interface control to issue the required command sequence.

The **command** parameter consists of a string of key words. These are:

#### **ACTIVE** *depiction\_time* [*active\_field\_time*]

Display the given depiction from the sequence and make it the active depiction to which all further commands will apply. If a field is selected then make the field editable at the given time. If the selected field is to be edited at a time different from **depiction\_time** then this time is specified by **active\_field\_time**. This will usually apply to static and daily fields. Automatically turned off by **DELETE**.

#### **CREATE\_DEPICTION** *source*- *sub\_source*- *issue\_time*- *valid\_time*- [*target\_time*]

Create a depiction using model fields from the given **source** and **sub\_source** with the issue (run) time **issue\_time** as the base for the depiction. If the model does not have a **sub\_source** then use the dash (-) place holder. Use the fields from the given **valid\_time** and create a depiction at **target\_time**. If **target\_time** is not given, use **valid\_time** as the target time. The fields to use from the model for depiction creation are specified in the configuration files associated with Ingrid. If **source**, **sub\_source**, **issue\_time** and **valid\_time** are all dashed (-) then an empty depiction is created at **target\_time** which must be specified.

#### **CREATE\_FIELD** *element level valid\_time*

Create an empty field for the given **element** and **level** at the given **valid\_time**. For daily fields the time will consist of a year and a day only (yyyy:jjj).

#### **DELETE\_DEPICTION** *group*|**ALL** *time*|**ALL**

Delete the depiction for the given **group** and **time**. Note that if both group and time are **ALL**, then all groups for all times will be deleted. (Only the **ALL** option for group is implemented at this time).

#### **DELETE\_FIELD** *element level time*|**ALL**

Delete the field, as defined by the given **element** and **level**, from the depiction specified by **time**. If **time** is **ALL** remove the field from every depiction in the sequence.

**GET\_DEPICTION *source valid\_time*|ALL [*target\_time*]**

Get a depiction for the given *valid\_time* from the given *source* and insert it into the sequence at *target\_time*. If *target\_time* is not given, use *valid\_time* as the target time. *Source* is the directory key, from the associated setup file, of any directory containing depictions. If *valid\_time* is **ALL** then *target\_time* is ignored and all of the depictions from the given *source* are read in.

**GET\_FIELD *source sub\_source*|- *issue\_time*|- *element level valid\_time* [*target\_time*]**

Get a field from the given *source* and *sub\_source* with the given *issue\_time* and element and level at *valid\_time* and insert into the *target\_time* depiction. If *target\_time* is not given assume that it is the same as *valid\_time*. *Source* can be either the directory key, from the associated setup file, of any directory containing depictions or a named *source* as defined in the Fields configuration file. If the given *source* does not have a *sub\_source* and/or an *issue\_time* then the appropriate parameter is replaced with a dash(-) as a place holder.

**INITIALIZE VIEW|EDIT *interp\_delta* ALL|NONE|RANGE *start end* SAVE|NOSAVE**

Initialize the depiction sequence. This command must only be issued once. The action taken by Ingrid depends on the specific initialization command. There are three groups of commands.

The first group specifies the mode. **VIEW** puts Ingrid into view only mode. **EDIT** puts Ingrid into edit mode and the timestep between the interpolated depictions generated when the depiction sequence is interpolated must be specified by *interp\_delta*. The *interp\_delta* is given in the format **hh** for a timestep in hours or **0:mm** for a timestep in minutes. The *interp\_delta* value must not be given for **VIEW** mode.

The next group specifies the number of depictions to be read in. If **ALL**, Ingrid reads in all of the depictions from the existing sequence on disc. If **NONE**, Ingrid deletes any existing depictions from disc and presents the user with an empty sequence. If **RANGE**, Ingrid will read only those depictions which are between *start* and *end* and deletes all others from disc (start time and end time are not used with **ALL** or **NONE**). If *start* is a dash ("-") then the oldest depiction found is used as the start time and if *end* is a dash the newest depiction is used as the end time.

The last group specifies if when depictions are deleted they can be saved to the backup directory. **SAVE** will backup the depictions while **NOSAVE** will not.

Notes:

1. This must be done after the background map has been defined with the **GEAction()** command **MAP\_BASE**.
2. If Ingrid is in viewer mode the depictions will be read but not deleted from the sequence stored on disc.

**SAVE\_DEPICTION *group*|ALL *time*|ALL**

Save the depiction for the given *group* and *time* to the archive store. (Only the **ALL** option is implemented for group at this time)

**TZERO *T0\_timestamp***

Set the timestamp to be used as the "T0" reference time.

## 2.9.3 RETURN

Upon successful completion **GE\_VALID** will be returned, a failure returns **GE\_INVALID**.

## 2.10 GStatus

Returns status information on the current depiction sequence.

### 2.10.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GStatus(request, nlist, list1, list2, cal)

char \*request,

int \*nlist,

char \*\*\*list1,

char \*\*\*list2,

CAL \*cal

### 2.10.2 DESCRIPTION

The **request** parameter defines the information required and **nlist** and **list1**, **list2** and **cal** return the information as requested. If a list is not required the corresponding parameter (**list1** and/or **list2**) may be set to NULL. Note that **list1**, **list2** and **cal** are internally static to this function and so will be overwritten by the next call to GStatus(). If values are required to be kept they must be copied. Do not free or modify any of these parameters.

The **request** parameter consists of a string of key words. These are:

#### DEPICT MODIFIED

Return a list of those depictions which have been modified since the last save. The number of depictions is given by **nlist** and the list of depictions is returned in **list1** (**list2** is undefined).

#### DEPICT TIMES

Returns a list of depiction times. The number of depictions is given by **nlist** and the list of depictions is returned in **list1** (**list2** is undefined).

#### INTERP TIMES

Returns a list of interpolated depiction times. The number of interpolated depictions is given by **nlist** and the list of interpolated depictions is returned in **list1** (**list2** is undefined).

#### FIELD BACKGROUND *element level*

Returns the background value for the given field in **cal**.

#### FIELD LINK\_INFO *element level*

Returns information on the time link status of the given field. Variable **list1[0]** will contain the link status as **NOT\_LINKABLE**, **NONE**, **PARTIAL**, **LINKED**, **FIELD** or **INTERPOLATED**. The linked to status is given in **list2[0]** as **MASTER\_LINK**, **FIELD** or **SELF**. If the linked to status is **FIELD** then the field to which it is linked is in **list1[1]** (element) and **list2[1]** (level).

A link status of **LINKED** indicates that the field is linked but not yet interpolated. A link status of **FIELD** indicates that the field is linked and interpolated but the labels are not. A link status of **INTERPOLATED** indicates that both the field and labels are linked and interpolated.

**FIELD MODIFIED *element level***

Return a list of times at which the given field has been modified since the last save. The number of times in the list is given by *nlist* and the list of times is returned in *list1* (*list2* is undefined).

**FIELD TIMES *element level***

Returns a list of times at which the specified field, as given by *element* and *level*, exists. The times are returned in *list1*. The time in list 1 corresponding to the current active time in the depiction sequence is returned in *list2*[0]. Note that for a field in the depiction sequence this will be the same as the current active time. For static and daily fields, however, it will normally be different.

**FIELDS [ *time* ]**

Returns a list of fields found in depiction time. If *time* is not specified then the return will be a list of those fields found one or more times in any depiction in the sequence. The number of fields is given by *nlist* and the list of fields is returned in *list1* and *list2*, where *list1* contains the element and *list2* the corresponding level.

**FIELDS EDIT\_POSTED**

Returns a list of fields for which an edit has been posted. The number of fields is given by *nlist* and the list of fields is returned in *list1* and *list2* where *list1* is the element and *list2* the corresponding level.

**FIELDS INTERPOLATED [ALL|FIELD]**

By default, or if the optional parameter **ALL** is given, returns a list of those fields which have been interpolated and which have not been modified or had any time links modified since they were interpolated. The number of fields is given by *nlist* and the list of fields is returned in *list1* and *list2* where *list1* is the element and *list2* the corresponding level.

If the parameter **FIELD** is specified, returns the list of fields which have the field interpolated but not the labels.

**FIELDS LINKABLE**

Return a list fields which are found one or more times in any depiction sequence and which are capable of being linked. The number of fields is given by *nlist* and the list of fields is returned in *list1* and *list2* where *list1* is the element and *list2* the corresponding level.

**FIELDS LINKED**

Return a list fields which are linked and which have not been modified since being linked. The number of fields is given by *nlist* and the list of fields is returned in *list1* and *list2* where *list1* is the element and *list2* the corresponding level.

**GROUP MASTER\_LINK\_STATUS *group*|GLOBAL**

Returns the link status of the master link for the given *group* in *list1*[0]. If *group* is **GLOBAL** then the status of the global master link is returned. The returned status will be one of **NONE**, **PARTIAL** or **LINKED**.

**GROUPS [ *time* ]**

Returns a list of groups found in depiction *time*. If *time* is not specified then the return will be a list of those groups found one or more times in any depiction in the depiction sequence. The number of groups is given by *nlist* and the list of groups is returned in *list1* (*list2* is undefined).

**2.10.3 RETURN**

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.



## 2.11 GEScratchpad

Handles all of the non-editing commands for the scratch pad.

### 2.11.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GEScratchpad( *command* )

char \**command*

### 2.11.2 DESCRIPTION

The *command* parameter consists of a string of key words. These are:

**EDIT DISTANCE** *colour style d\_colour d\_font d\_font\_size*

Show the distance and direction between two points. The parameters *d\_colour*, *d\_font* and *d\_font\_size* are for the display of the distance and direction and the *colour* and *style* are for the displayed joining line.

**EDIT DRAW** *colour style*

Draw a line on the scratch pad using the named X *colour*, with the given *style*. For allowable *style* values see Note 3 above.

**EDIT PRESET\_DISTANCE** *colour style d\_colour d\_font d\_font\_size distance*

Display a line of a pre-determined length as given by the distance parameter. The remaining parameters are as for the **DISTANCE** command.

**EDIT SELECT** [**ALL**]

Enter a mode that will allow the selection of lines on the scratchpad. The optional parameter **ALL** will select all of the lines.

**EDIT TEXT** *colour font size label*

Put a text string label on the scratch pad using the named X *colour* using font type *font* of size *size*. For recognized font values see Note 2 above. The size can be either in percent of map size (indicated by a "%" after the size) or in pixels.

**SHOW**

Show the scratch pad. The scratch pad will appear to be displayed "under" the currently active depictions.

**HIDE**

Hide the scratch pad.

### 2.11.3 RETURN

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

## 2.12 GETimelink

Handles all of the non-editing commands for the time linking.

### 2.12.1 SYNOPSIS

```
#include <ingred.h>
```

GEREPLY GETimelink( **command** )

char \*command

### 2.12.2 DESCRIPTION

Time linking is the process of linking one point or area or line on one chart with the same point or area or line on another thus identifying them as the same object to the graphics editor. Most links are specific to a given field but there is a special class of link called Master Link which is tied to a field group and not to any specific field. There is also a Master Link which is not tied to any particular group or field but is a "Master" Master Link.

The **command** parameter consists of a string of key words. These are:

#### **EDIT BACKWARD | DELINK | FORWARD | MOVE**

Set time link editor to either link in a time forward direction or a time backwards direction. The type of link to edit is set by the STATE LINK\_MODE command.

#### **EDIT DELINK [ NODE | CHAIN | CLEAR ]**

Delete the closest link node or link chain. If the optional **CLEAR** parameter is given, then delete all the link chains.

#### **EDIT MERGE [ FETCH *source sub-source element level* ]**

The command given without parameters puts the time link editor into merge mode, which allows the user to merge time links from another field into the currently active field.

The command with **FETCH** (and parameters) specifies the source field to use for the merge. Any parameter not relevant for a particular source must be place held with a dash (-). Once completed, the time link editor will return to the state assumed upon receiving the **MERGE** command without parameters.

#### **EDIT MERGE [ SELECT\_ALL ]**

#### **EDIT MERGE [ MERGE | TRANSLATE | ROTATE ]**

After a **MERGE FETCH** command has been issued, the time link editor will request that the user choose link chains to merge with the existing field. Link chains can be selected or deselected with the mouse, or all the link chains in the merge field can be chosen with the **SELECT\_ALL** option.

Once the desired link chains have been chosen, issuing the command with **MERGE**, **TRANSLATE** or **ROTATE** will cause the time link editor to go into a state that allows the chosen link chains to be merged in place, or translated/rotated and then merged.

**EDIT MOVE**

Move a link node. The type of link to move is set by the STATE LINK\_MODE command.

**ENTER**

Start the time link procedure. Note that the procedure commands are entered under the edit group.

**EXIT**

Exit from the time link editor.

**INTERPOLATE [ CANCEL | FIELD *element level* ]**

Interpolate any completely linked fields. If **CANCEL** is given then the interpolation in progress is to be stopped. If **FIELD** followed by the *element* and *level* names of a field is given, then interpolate just the specified field.

**LINK\_TO *element level***

The currently active field will use the links of the field defined by *element* and *level* for time linking. If the link is to be to the group master link then the *element* must be **MASTER\_LINK** with no associated *level*. If the link is to be to the active field itself then both *element* and *level* must not be present. If the field is to be linked to itself (it will have its own link) then both *element* and *level* must be absent.

**MASTER\_LINK *group* | GLOBAL**

Create a Master Link. If *group* is given create the link for the specified group. If *group* is **GLOBAL** create the global master link for all groups. Note: If the given field has links of its own they will be destroyed.

**SHOW *option* ON | OFF**

Turn the given display mode option on or off. The option can be one of:

**CONTROL**

Show time of control nodes.

**EARLY\_LATE**

Show early/late start/end times for link chains that do not start/end at depiction times.

**SPEED**

Show speed above link nodes.

**TIME**

Show time below the link nodes.

### 2.12.3 RETURN

Upon successful completion GE\_VALID will be returned, a failure returns GE\_INVALID.

## 2.13 GEZoom

Handles all zoom commands.

### 2.13.1 SYNOPSIS

```
#include <ingred.h>
```

```
GEREPLY GEZoom( command )  
char *command
```

### 2.13.2 DESCRIPTION

The **command** parameter consists of a string of key words. These are:

#### IN

Set Ingred into zoom define mode. Ingred will request a zoom cursor and wait for the user to create a rubber band box on the map which defines the area to which to zoom. Once this is done Ingred will send the zoom information back to the controlling interface (see `status_proc` Function prototype).

#### OUT

Ingred will zoom out to the full original map.

#### PAN

Set Ingred into pan mode. The user can pan around the base map by "clicking" at the location at which the zoomed area is to be centered. When the pan is completed Ingred sends the zoom information back to the controlling interface. The right button is used to terminate the pan and the interface is notified of the termination of the zoom by Ingred (see `status_proc` Function prototype).

#### PAN\_DONE

Exit pan mode.

#### AREA **x y width height**

Zoom into the area defined by **x**, **y**, **width** and **height** the upper-left corner of the zoomed area to the specified location on the base map. The x and y co-ordinates are as a percent of the base map x-axis and y-axis length respectively.

### 2.13.3 RETURN

Upon successful completion `GE_VALID` will be returned, a failure returns `GE_INVALID`.

---

## Chapter 3

# Function Prototypes

This section contains alphabetically-organized reference pages for each function type that is required by the GEConnect() function.

Each page contains a synopsis of the function's calling sequence, its arguments, a description of its purpose and features.

### 3.1 cursor\_proc

Prototype procedure invoked when a processing delay is expected.

#### 3.1.1 SYNOPSIS

```
typedef void (*cursor_proc)(char *type, Boolean state)
char *type,
Boolean state
```

#### 3.1.2 ARGUMENTS

***type***

Specifies the type of cursor requested. The types send by ingred are:

**"busy"**

A cursor which indicates that the program is busy.

**"finger"**

A pointing finger.

**"obscured"**

A cursor which indicates that the drawing window is obscured.

---

**"pen"**

The figure of a pen or pencil.

**"stop"**

A stop sign.

**state**

Specifies the display state of the cursor. May be either on (**True**) or off (**False**).

### 3.1.3 DESCRIPTION

This function is invoked by Ingrid when a specific style of cursor is wanted. It is up to the calling interface to provide the function and determine the look of the requested cursor type.

---

## 3.2 message\_proc

Prototype procedure invoked when Ingred wants the interface to post a message.

### 3.2.1 SYNOPSIS

```
typedef void (*message_proc)(char *type, char *text)
```

### 3.2.2 ARGUMENTS

***type***

The type of the message. Normally the type is used to index into the resource file to obtain information on the colour and font that the message should be displayed in. There is a special case where type is the keyword "ErrorDialog". In this case the interface is expected to display the message in a separate dialog.

***text***

Normally the text of the message. In the special case where type is "ErrorDialog", text contains the keyword into the CommonMdb file which contains the actual message.

### 3.2.3 DESCRIPTION

This function is invoked by Ingred when messages are to be displayed to the user. It is the responsibility of the calling interface to provide the display mechanism. With the exception of "ErrorDialog", type is used as a reference in the resource database as:

```
xfpa.ingredMessage.type: colour font
```

Where ***colour*** and ***font*** give the colour and font specification for the particular ***type*** of message. If ***font*** is not specified, then the normal font will be used.

---

### 3.3 status\_proc

Prototype procedure invoked when Ingred needs to pass status information.

#### 3.3.1 SYNOPSIS

```
typedef void (*status_proc)(char *status, CAL cal)
char *status,
CAL cal
```

#### 3.3.2 ARGUMENTS

**char \*status**

A null terminated string containing the status information.

**CAL cal**

If not NULL, a CAL data structure object.

#### 3.3.3 DESCRIPTION

The status information is sent as a series of key words and will depend on the current operating mode of Ingred.

**ANIMATION SHOWING *valid\_time***

The valid time of the depiction/interpolation currently being displayed in the animation sequence.

**BACKGROUND *element level***

The background attribute values of the given field. If no background attribute values exist **cal** will be null. This is an implied request for the controlling program to send Ingred valid background attribute values.

**EDIT CONTOUR|VECTOR|AREA|WIND|LINE|LCHAIN|POINT SELECT|DESELECT [ SET ]**

An area, line, link chain, point or an area of influence of a spline or vector field has been selected or deselected. If selected and **cal** is not NULL, then the attribute values will be contained in this structure. If the **SET** parameter is part of the command, then the appropriate object attributes can be given values. If not present then the object attribute values may not be modified. (See **EDIT** for commands to sent to Ingred).

**EDIT AREA|WIND SELECT [ HOLE ]**

An area hole has been selected, and can be deleted.

**EDIT LCHAIN SELECT|DESELECT [ SET ]**

A link chain node has been selected or deselected. If selected and **cal** is not NULL, then the attribute values will be contained in this structure. If the **SET** parameter is part of the command, then the link chain node attributes can be given values. If not present then the link chain node attribute values may not be modified.

---



**EDIT ADDING *time*|NEW**

This status return gives the time of a link chain node that is about to be added or **NEW**. The latter return indicates that the chosen link chain node location is at the start or end of another link chain, and Ingrid must be informed whether the next node will be a new link chain or a continuation of the other link chain.

**EDIT BUTTON *command* ON|OFF**

The button corresponding to the given Ingrid edit command should be set active (**ON**) or inactive (**OFF**). The permitted values of *command* are **BREAK, CANCEL, CLEAR, COPY, CREATE, DELETE, JOIN, PASTE, PRESET\_OUTLINE, PROCEED, REJOIN, UNDO** or **UPDATE**. (See GEEedit for the associated command that is to be sent to Ingrid when the button has been made visible and the user has selected it. For **PRESET\_OUTLINE** see GEDepiction. Note also that in this case there is a series of commands associated with the requested button).

**EDIT DRAWING ON|OFF**

This command is issued at the start of a point-by-point draw. The interface is expected to make available buttons that will issue a **DRAW\_DONE** or **CANCEL GEEedit** command.

**EDIT MODIFYING ON|OFF**

This command is issued when an object has been modified (such as the border of an area having been redrawn).

**EDIT OUTLINE DRAWN|HOLE|MOVED|STOMP**

This is status information. It lets the interface know that outlines of the appropriate type have been created and are now available for use in the **PRESET\_OUTLINE** command. Normally the interface would make the corresponding action buttons available for use. Once created these outlines are always available.

**FIELD CREATE *element level valid\_time* [ ASK ]**

The controlling interface is requested to send the command to create an empty field at *valid\_time*. If the last parameter is **ASK**, the the interface must put up a requestor asking the user for permission to create the field.

**FIELD STATUS *element level time status***

The given field as defined by *element* and *level* has had a status change at the time given by *time*. This can be used by the controlling interface to set visual indications of what times the field exists at and what is the link state of the field at each depiction time. Status can have a value of - (dash, meaning not defined), **DELETED** (meaning the field does not exist anymore), **NONE, PARTIAL, LINKED, FIELD** or **INTERP** (See **TIMELINK STATUS\_UPDATE** command below for a description of the last five keywords).

The *element* can be returned as "**MASTER\_LINK**" with the *level* undefined. This is a special case and gives the status of the Master Link.

**FIELD TIMES *element level ntimes time1 status1 time2 status2 ...***

The given field as defined by *element* and *level* exists as the times as given in the list *time1, time2, ...* with a link status for each time given by *status1, status2, ...*. The number of times is given by *ntimes*. The content of the status returns is the same as those defined for the **FIELD STATUS** command above.

**GUIDANCE LEGEND *field\_id colour\_index***

Information about any active guidance fields are usually displayed in a legend dialog. This status information line provides the colour that the information should be displayed in. The guidance field (as identified by *field\_id*) should have its information displayed in the colour as given by *colour\_index*.

**INTERRUPT ON|OFF**

The editor is in a state such that GUI elements (such as dialogs) can be popped up for the users consideration (**ON**). If **OFF**, the editor is in a state where interruptions cannot be tolerated.

**INTERPOLATE *element level %done %total***

The field as given by *element* and *level* is being interpolated. The percentage complete of the interpolation procedure for the field is given by *%done* and the percentage complete of the interpolation for all of the fields is given by *%total*.

**LABEL**

The CAL structure for the label is given in *cal*. The contents of the structure should be put into an attribute editor for possible label addition or modification depending on the current state of the interface.

**MODE BUTTON *command* ON|OFF**

The button corresponding to the given Ingrid mode command should be set active at this time (**ON**) or inactive at this time (**OFF**). The permitted value of command is DRAW. (See the GE\_ACTION command).

**SAMPLE DISPLAY ON|OFF**

If the amount of information to be displayed for a sample operation is too great to look reasonable on the screen, Ingrid will pass off the display of this information to the calling interface. If sample information is to be shown, then the **ON** command is given and the information is contained in the passed CAL structure. If the sample display is to be turned off, then the **OFF** command is sent.

**SCRATCHPAD BUTTON DELETE ON|OFF**

A button allowing scratchpad lines to be deleted should be turned on or off.

**TIMELINK LINKING *time|NEW|EXTRAP [ state ]***

This status return gives the time of the field that is about to be linked or **NEW** or **EXTRAP**. The latter returns indicate that Ingrid has put up the requestor for the start time of the timelink chain or the end time of the chain. The optional *state* parameter can have the values:

**NO\_FIELD**

No field is available for linking.

**NO\_DATA**

No data is available.

**TIMELINK STATUS\_UPDATE *element level* NONE|PARTIAL|LINKED|FIELD|INTERP**

The link status of the given *element* at *level* is:

**NONE**

the element is not linked or interpolated,

**PARTIAL**

the element is partially linked but not enough for interpolation,

**LINKED**

the element is linked but not interpolated,

**FIELD**

the element is both linked and interpolated by the labels are not

**INTERP**

the element and labels are both linked and interpolated

**TIMELINK MASTER\_LINK\_STATUS\_UPDATE group|GLOBAL NONE|PARTIAL|LINKED**

The status of the master link for the given group is:

**NONE**

not linked or interpolated,

**PARTIAL**

partially linked but not enough for interpolation,

**LINKED**

there are sufficient links for interpolation

If group is GLOBAL then the link status of the global master link is given.

**TIMELINK ACTION type**

The action of type has been completed. The types are:

**INTERMEDIATE\_DONE**

The selection and move of an intermediate link point as requested by a timelink edit move command has been completed ( see the GETimelink "TIMELINK EDIT MOVE" command).

**SELECT**

One or more of the links that were fetched during the **MERGE** operation have been selected (See the GETimelink "EDIT MERGE FETCH ..." command).

**DESELECT**

Any links selected during the **MERGE** operation have been deselected.

**ZOOM AREA x y width height**

The specifications for the zoom area defined interactively by the user within Ingrid. Once Ingrid has accepted the area this message will be sent to the interface. The values are the position of the zoom area within the main map (**x, y**) and the size of the area (**width, height**) in decimal percent of the map axis length. (For example: 20.0 34.3 34.0 35.3 would define a zoomed area starting 20% along the x-axis, 34.3% along the y-axis and with a **width** of 34% of the x-axis length and a **height** of 35.3% of the y-axis length.)

**ZOOM PAN END**

This message is sent when the user indicates to Ingrid that they are finished the current panning operation.

**ZOOM START**

This message is sent as soon as the user has defined the zoom area on the map and has released the mouse button. This should be used to put up a "zooming in progress" cursor indicator.



## Chapter 4

# Attribute List (CAL) Functions

The Complete Attribute List (CAL) functions allow the user to input and output data from the opaque data type CAL. The CAL data structure contains all of those attributes defined in the configuration files for a particular field, plus some pre-defined attributes that are present by default and not normally specified in the configuration files.

All attributes and their values are specified as strings. It is up to the calling functions to interpret the specific meaning of any value associated with an attribute. There are predefined attributes that are recognized by the functions and these are defined in Table 4.1.

Table 4.1: Predefined Attributes

Attribute	Definition
CALautolabel	Automatically generated label
CALcategory	FPA category
CALlabeltype	FPA label type
CALlatitude	Latitude at point on feature
CALlongitude	Longitude at point on feature
CALlinetype	FPA line field type
CALscatteredtype	FPA scattered field type
CALuserlabel	FPA user defined label

### 4.1 Function Descriptions

**void CAL\_add\_attribute (CAL, STRING name, STRING value)**

Add an attribute identified by name with an associated value to the CAL structure.

**void CAL\_add\_location (CAL, MAP\_PROJ \*mproj, POINT point)**

Adds a location point to the CAL structure by converting the location specified by point to a latitude and longitude using the map projection mproj and storing these in CAL using the predefined attributes CALlongitude and CALlatitude.

**void CAL\_clean (CAL)**

Remove all attributes from the CAL structure that do not have a value associated with them.

**CAL CAL\_create\_by\_name (STRING element , STRING level)**

Create a CAL structure from the element and level names of a field that is populated with the attributes specific to that field.

**CAL CAL\_create\_by\_edef (FpaConfigElementStruct \*cfg)**

Create a CAL structure from the field element definition structure that is populated with the attributes specific to that field.

**CAL CAL\_create\_default (void)**

Create a CAL structure that contains the predefined attributes CALautolabel, CALcategory and CALuser-label.

**CAL CAL\_create\_empty (void)**

Create a CAL structure with no attributes.

**CAL CAL\_destroy (CAL)**

Destroy the given CAL structure and return a value of NULL.

**CAL CAL\_duplicate (const CAL)**

Duplicate the given CAL structure.

**void CAL\_empty (CAL)**

Remove all attributes from the CAL structure.

**STRING CAL\_get\_attribute (CAL, STRING name)**

Return the value associated with the given attribute.

**void CAL\_get\_attribute\_names (CAL, STRING \*\*name, int \*number)**

Return a list of all attributes in the CAL structure. The name list is allocated and it is the responsibility of the calling function to free the returned list. Note that the elements the list points to are internal static and must not be freed. (That is, free name but not name[i]).

**void CAL\_get\_defaults (CAL, STRING \*category, STRING \*auto\_label, STRING \*user\_label)**

Returns the values associated with the predefined attributes CALcategory, CALautolabel and CALuser-label. The returned values point to internal static values and must not be freed.

**LOGICAL CAL\_has\_attribute (CAL, STRING name)**

Returns true if the given CAL structure contains the given attribute name.

**LOGICAL CAL\_is\_preset (STRING name)**

Returns True if the attribute name is one of the preset magic attributes.

**LOGICAL CAL\_is\_value ( STRING value )**

Returns True if the attribute value has a value.

**void CAL\_merge (CAL dest, CAL source, LOGICAL overwrite)**

Merges the given source CAL into the destination CAL. If overwrite is True, then attributes in source will overwrite those in dest. If False, then source attributes will only be written into dest if the attribute does not exist in dest or if the attribute in dest does not have a value.

---

**LOGICAL CAL\_no\_value ( STRING value )**

Returns True if the attribute value does not have a value.

**LOGICAL CAL\_same (CAL a, CAL b)**

Returns True if CAL a and CAL b are the same.

**void CAL\_set\_attribute (CAL, STRING name, STRING value)**

Set the value of the attribute name to value. If the attribute does not exist in CAL then no action is taken.

**void CAL\_set\_defaults (CAL, STRING cval, STRING alval, STRING ulval)**

Set the value of the predefined attributes CALcategory to cval, CALautolabel to alval and CALuser-label to ulval.

---