# THE PCMDI VISUALIZATION AND COMPUTATION SYSTEM (VCS): A WORKBENCH FOR CLIMATE DATA DISPLAY AND ANALYSIS

by

Dean N. Williams and Robert L. Mobley

Program for Climate Model Diagnosis and Intercomparison
Lawrence Livermore National Laboratory, Livermore, CA, USA

March 1994

# DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

# Contents

**Abstract** .................... i

**1. Introduction** .................... 1

**2. The Command Line** .................... 4

**3. Interaction** .................... 6

3.1. Message Panel .................... 10
3.2. Graphics Canvas .................... 10
3.3. Page Description Panel .................... 11
3.4. Control Panel .................... 13
3.4.1. Template Panel .................... 14
3.4.1.1. Template Browser Panel .................... 15
3.4.1.1.1. Template Text Editor Panel .................... 17
3.4.1.1.2. Template Format Editor Panel .................... 18
3.4.1.1.3. Template X-Tick Mark Editor Panel .................... 20
3.4.1.1.4. Template Y-Tick Mark Editor Panel .................... 21
3.4.1.1.5. Template X-Label Editor Panel .................... 23
3.4.1.1.6. Template Y-Label Editor Panel .................... 24
3.4.1.1.7. Template Box/Lines Editor Panel .................... 25
3.4.1.1.8. Template Legend Space Editor Panel .................... 27
3.4.1.1.9. Template Data Space Editor Panel .................... 29
3.4.2. Graphics Method Panel .................... 30
3.4.2.1. Continents Editor Panel .................... 32
3.4.2.2. Isoline Editor Panel .................... 33
3.4.2.3. Isofill Editor Panel .................... 35
3.4.2.4. Outline Editor Panel .................... 36
3.4.2.5. Outfill Editor Panel .................... 37
3.4.3. Data Panel .................... 39
3.4.3.1. Data Selection and Data Selection Browser Panels .................... 41
3.4.3.2. Data Editor Panel .................... 43
3.4.3.2.1. Dimension Manipulation Panel .................... 45
3.4.3.2.2. Dimension Assignment Panel .................... 48
3.4.4. List Panel .................... 49
3.4.4.1. List Editor Panel .................... 50
3.4.5. Indices Panel .................... 52
3.4.6. Colormap Panel .................... 54
3.4.7. Basics .................... 57
3.4.7.1. Text (Tt), Orientation (To), and Editor Panels .................... 57
3.4.7.2. Lines (Tl) and Editor Panels .................... 58
3.4.7.3. Fill Area (Tf) and Editor Panel .................... 60
3.4.7.4. Marker (Tm) Panel .................... 61
3.4.7.5. Format (Th) Panel .................... 61
3.4.7.6. Pattern (Tp) Panel .................... 61

3.4.8. Script Panel ........ 63
3.4.9. Animation Panel ........ 63
3.4.10. Output—Postscript and Raster Panels ........ 66
3.4.11. Orientation—Portrait or Landscape ........ 67

**4. Scripts ........ 69**
4.1. Display Attributes—D_name ........ 70
4.2. Primary Element Attribute Sets ........ 71
4.2.1. Picture Template Attributes—P_name ........ 71
4.2.2. Graphics Attributes—G*_name ........ 75
4.2.2.1. Isoline—Gi_name ........ 75
4.2.2.2. Outline—Go_name ........ 77
4.2.2.3. Isofill—Gfi_name ........ 78
4.2.2.4. Outfill—Gfo_name ........ 79
4.2.2.5. Continents—Gcon_name ........ 80
4.2.3. Array Data Attributes—A_name ........ 81
4.2.3.1. Minimal Information Required to Select or Compute a Variable ........ 81
4.2.3.2. "Defined" and "Displayed" Attributes ........ 81
4.2.3.1. Order of Array Dimensions ........ 83
4.3. Basic Element Attributes ........ 87
4.3.1. Text Attributes—Tt_name ........ 87
4.3.2. Text Orientation Attributes—To_name ........ 88
4.3.3. Fill Area—Tf_name ........ 88
4.3.4. Formats—Th_name ........ 89
4.3.5. Line Attributes—Tl_name ........ 90
4.3.6. Marker Attributes—Tm_name ........ 91
4.3.7. Patterns—Pat_name ........ 91
4.3.8. Colormaps—C_name ........ 92
4.3.9. Lists—L_name ........ 92
4.4. Script Commands ........ 93
4.4.1. Canvas ........ 93
4.4.2. Clear ........ 93
4.4.3. Color ........ 93
4.4.4. Copy ........ 93
4.4.5. Dump ........ 94
4.4.6. Index ........ 94
4.4.7. Loop ........ 94
4.4.8. Page ........ 95
4.4.9. Postscript ........ 95
4.4.10. Raster ........ 96
4.4.11. Remove ........ 96
4.4.12. Rename ........ 96
4.4.13. Run ........ 96
4.5. Script Example ........ 97

**5. Conclusions and Future Work ........ 104**

# ABSTRACT

This software was developed by the Program for Climate Model Diagnosis and Intercomparison (PCMDI) at the Lawrence Livermore National Laboratory in Livermore, California. It was designed to provide some of the basic capabilities needed for validating, comparing, and diagnosing climate model behavior. It can be controlled either interactively, or from a script file, or control can alternate between these modes during a session. A script can be saved during an interactive session and merely replayed, or it can be edited and replayed. The state-of-the-system can be dumped, as a script, at any instant, and that script can be used later to restore that instant of the session.

A graphic display page is made up of one or more graphic pictures. Each picture is defined by a picture template, a graphic method, and data. These three elements are termed the "primary elements" in this application. The picture template defines where to display; the graphic method defines how to display; and the data defines what to display. Tables of attribute sets have been created for each of the primary elements, and the application provides the user with the capability for creating a picture by choosing entries from these tables. Once a picture is created, it can be manipulated by changing the choice of primary elements or by altering them. This application also includes tables of basic elements that describe text, lines, markers, fill areas, patterns, colormaps, and lists. These basic elements are used to assign some of the attributes to the primary elements. This fact allows a picture to be manipulated by choosing different basic elements or by directly altering these basic elements.

Attributes for data can describe variables existing in a file or variables to be computed as a function of previously selected variables. The dimensions of variables can be subset, reversed, transposed, wrapped-around, and thinned by selecting either a stride of nodes or by randomly selecting individual nodes. Grid transformations are supported by allowing a different set of dimension vectors to be specified in the dimension descriptors. A display page can be output as either Adobe PostScript for hardcopy, or as a raster image for hardcopy or animation.

# 1. Introduction

The massive amounts of data produced by decadal simulations with climate models makes it imperative that a utility exist for easily visualizing the data with a number of different techniques. It is impossible to gain insight into these models' behavior without graphic representations of the data, and no single display technique is best for recognizing or for explaining all features of their behavior. In comparing decadal simulations from tens of models, it is important to have a computational capability and, since most of these models use different grids, to also have the capability to perform grid transformations. These additional capabilities not only make it easier and quicker to browse through giga-bytes of data in the comparison of models, but it also removes the need for intermediate data files.

Visual displays are used, not only for browsing data, but also for documenting a simulation and presenting results, so it is necessary that the visualization utility allow as much user control over the display as is feasible and provide a method for producing hardcopy visuals without exhaustive human interaction.

The design of this application software was driven by the need to provide a platform upon which an ever growing selection of methods for displaying one- and two-dimensional data could be based with extensive user control over all elements of the process. This platform provides the user with the capability to control the process either interactively or from a script. It provides a scripting capability for saving the instantaneous state-of-the-system for later recovery, and a capability for saving a continuous script of an interactive session that can be edited (or not) and replayed.

The state-of-the-system is defined by a few control variables and by three categories of tables of named attribute sets. There is a single display table, a table for each of the primary elements (picture templates, multiple graphics methods, and array data), and a table for each of the basic elements (text, text orientation, line, format, marker, fill area, pattern, colormap, and list). Support is provided for creating and changing these tables and the control variables.

The tables of named attribute sets define the elements required to create a graphics display page in a workstation drawing area (**Graphics Canvas**). That display page is made up of a picture or pictures, which are made up of multiple graphic segments. All of the entries in the display table define a display page; each individual entry in the display table defines a picture; and segments are defined by the picture template used in that entry. Segments contain displays for: the individual text descriptors for the data, decorative boxes and lines, a legend, and the graphic representation of the data.

A display table entry contains: a unique id name, a priority, an on/off switch, a reference to a graphic method, a name of a picture-template table entry, a name of an entry from the referenced graphics method table, and the correct number (determined by the graphic method) of entry namės from the array data table, and status information for each of the segments of the picture.

A picture-template table entry contains (1) attributes for positioning each graphic segment of a picture, and (2) attributes defining how to create each graphic segment, except for the segments that graphically represent the data and the display legend for these data.

VCS provides five methods for graphically representing data: isoline, outline, isofill, outfill, and continents. There is a graphics method table for each method. An entry in the graphics-method table contains: a name of a method for projecting the graphic representation of data onto a display page, names of list table entries to be used for labeling and tick marking axes, real-world coordinates for scaling the space where the data will be graphically represented, and parameters required for the graphic method.

An entry in an array data table contains the attributes of a variable, including both the information required to define the variable, and the information that will actually be displayed. A portion of the definition is supplied by the user, for instance: the file name and naming strings to define the variable. The displayed information is by default the same as the definition, but substitutes can be supplied by the user.

An interactive procedure is provided for searching files and selecting the variables that will have their descriptors entered into the array data table. The data·acquisition procedure provides the capability to subset, reverse, transpose, and wrap-around, and to select a stride of nodes or randomly select individual nodes of a dimension. This procedure also provides the capability for making grid transformations.

Attributes of the primary elements contain, among other things, references to table entries from the tables of basic elements.

Entries from a table of named colormaps are available for selection as the “active colormap.” The colors in the active colormap can be manipulated interactively, and the resulting changes are visible immediately. There are 240 user definable colors (0 through 239) in a colormap. Color indices 240 through 255 are also usable, but their colors are fixed. The fixed colors are, in order: white, black, red, green, blue, yellow, cyan, magenta, orange, brown, violet, olive green, grey, light green, light red, and light blue.

A table of named lists, each of which consists of pairs of values and strings, can be defined and then can be used in defining some attributes of the graphics methods and array data. Either the value or string, or both, will be used to satisfy the requirement for the attribute. The array data attributes can contain list names or indexed list names to define naming or dimension parameters for the data request, or can contain dimension values for defining a grid transformation. Indexed list names allow the use of a particular list value.

A set of index variables (I, J, K, L, M, and N) is available to define a particular list value selection. If an index variable is used, the selection can be manipulated by altering the value of the index variable.

A procedure is provided to output graphics display pages in Adobe PostScript for later production of hardcopy, and as raster images for later production of hardcopy or animation.

This software was written mostly in ANSI C with some FORTRAN 77, and it uses: GKS for visual displays, the Motif tool kit for its window interface, and the PCMDI Data Retrieval and Storage (DRS) software library for access to data.

## 2. The Command Line

The command for executing this software is: *vcs*

As a part of its initialization, this software attempts to read a script file to load the tables of primary and basic elements with a variety of attribute sets. The contents of this initialization script file can be customized for each user; it is named:

*basedirectory*/PCMDI_GRAPHICS/initial.attributes

The *basedirectory* is defined either by the environment variable, PCMDI_GRAPHICS_DIR, or it is defined to be a subdirectory of the environment variable, HOME, as, $HOME/PCMDI_GRAPHICS. If neither of those environment variables is defined, it is, by default, a subdirectory of the current directory, ./PCMDI_GRAPHICS.

If the *basedirectory* is not readable and writable by the user, this process will terminate with an error.

Another script file can be read immediately after this script file. It is assigned on the command line as:

*vcs -i path/inputscriptfilename*

During the reading of this script file, the process is in a batch-like mode and a **Graphics Canvas** might appear with graphics displayed on it, but the **Graphics Canvas** will disappear after completion. Once this script file has been read, control will become interactive, and any graphics that were displayed will be erased and then redisplayed in the new **Graphics Canvas** for the interactive portion of the session. During the interactive portion of the session, other script files can be selected for input.

A continuous script can be saved during an interactive session by assigning an output file on the command line:

*vcs -o path/outputscriptfilename*

The replay of the output script file will not reproduce the manipulations of buttons and menus that are seen during the interactive session, but it will reproduce the graphic displays and changes to the tables and their attribute sets. It will also save the same postscript file(s) and raster image file(s).

During the processing of the initialization script file, any error messages will be written to the UNIX standard output file. During the processing of the input file defined by the -i option, the error messages will, by default, be written to the file: *basedirectory*/ERROR_GRAPHICS (*basedirectory* is defined above). These error messages can optionally be saved to a file dictated by the user with the command line option:

```
vcs -e path/errorfilename
```

The errors occurring during the interactive portion of a session will be sent to a message panel and to the error file.

# 3. Interaction

The interactive interface provides procedures for: quickly accessing and computing data, producing a picture that visually represents the data values, refining the picture by altering the primary and basic element attributes that describe the data and its visual representation, and saving these attributes so they can be re-used.

When the interactive interface window appears, after the initialization script file and input script file have been processed, it covers the entire workstation screen, either as shown in Figure 1a, or as shown in Figure 1b if either of the script files defines the page orientation to be portrait. See section **2. THE COMMAND LINE** and section **4.4.8. Page.**

The "fixed" panels are always visible in the interactive interface window; they are:

**Message Panel** In the upper left corner of the screen—error, warning, and information messages appear in this panel.

**Graphics Canvas** In the lower left corner of the screen—can be viewed in either landscape (Figure 1a) or portrait (Figure 1b) orientation. This is the panel in which the pictures containing graphic representations of data are drawn.

**Page Description Panel** In the upper right-hand corner of the screen—contains a **Picture Descriptor Form** for describing each picture that is displayed on the **Graphics Canvas.** It appears, in Figures 1a and 1b, with a **Picture Descriptor Form** ready for immediate use.

**Control Panel** In the upper center of the screen—contains buttons for bringing up additional interface panels, and a button at the bottom of the panel for toggling the orientation of the **Graphics Canvas** between portrait and landscape. The additional interface panels will appear on the right side of the interactive interface window, below the **Page Description Panel.**

The **Template Panel** (showing the contents of the table of picture-template attribute set names), the **Isofill Panel** (showing the table of graphic-method isofill attribute set names), and the **Data Panel** (showing the table of array-data attribute set names, if any) are located below the **Page Description Panel.** Below this row of panels is the **Data Selection Panel,** which aids the user in selecting variables for insertion into the table of array data attributes. Below this panel is the **Variable Selection Browser Panel** for viewing attributes of variables as they are marked in the **Data Selection Panel.**

The mouse operations that are useful for manipulating many of the interface panels are described below:

| | |
|---|---|
| Click | Press and release the left mouse button. |
| Click and Hold | Press and hold down the left mouse button. |
| Double Click | Click twice on the same item. |
| Choose | Click and hold to see the choices, then move the cursor to the choice, and release the mouse button. |
| Mark | Click on a markable item, and it will be highlighted or outlined. |
| Copy and Drop | Press and release the middle mouse button on a name in one of the panels, immediately below the **Page Description Panel**, that will allow a copy (such as: **Template, Graphics, Data, List, Tt, To, Tl, Tf, Tm**) to place the name and an indication of the source panel on a clipboard, from which it can, at any time, be dropped into an input text window that will accept a clipboard entry drop from the particular source panel. |
| Remove | Move the mouse cursor to the removable item, then press and release the right mouse button. The selected item in the list will be removed. |

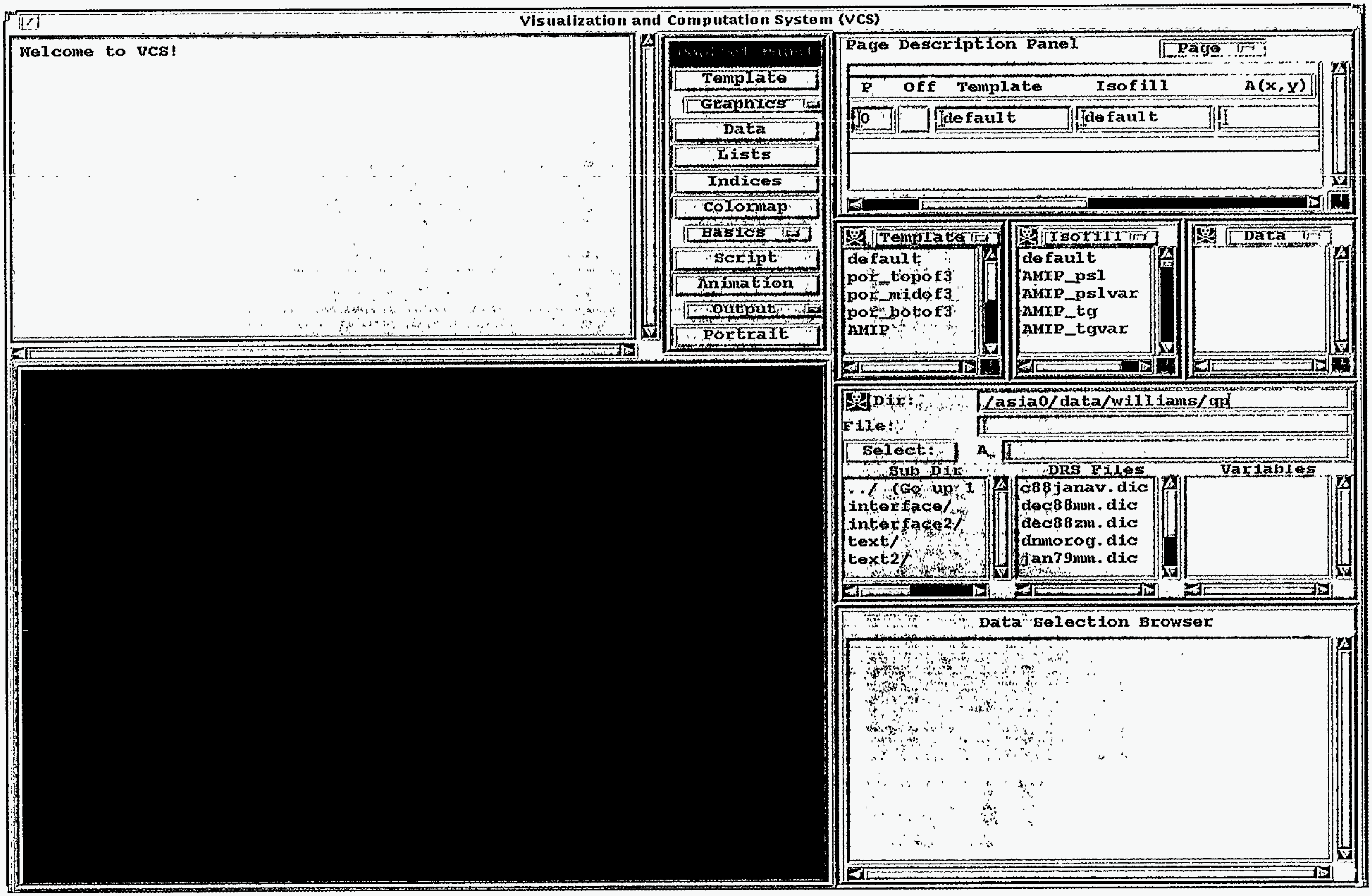
Some interface panels will not allow any mouse or keyboard operations at all. If the user tries to change the interface, the workstation will beep and nothing in the interface will change.

Any panel with a red and yellow skull and cross-bones button, in the upper left corner, can be terminated by clicking on this **Terminate Button**, and all its sub-panels will also be terminated. Terminating a panel with this button will cancel any changes that have not already been inserted into the attribute set.

Any panel with a black down arrow on a green background button in the lower right corner can be extended downward by clicking on this **Extension Button**. This button will extend the interface down to the bottom of the screen. If an interface is extended, it will have a black up arrow on a green background button in the lower right corner. It can be retracted upward to its original position by clicking this **Retraction Button.**

The interactive procedures described in this section are procedures for manipulating attributes that will affect the visual appearance of the **Graphics Canvas.** The effect of these manipulations of attributes can be viewed almost instantaneously if they are used by an active display entry.

Figure 1a. Initial Page with Landscape Orientation for Graphics



Welcome to VCS!

| P | Off | Template | Isofill | A(x,y) |
|---|---|---|---|---|
| 0 | | default | default | |

Template: default, por_topof3, por_midof3, por_botof3, AMIP

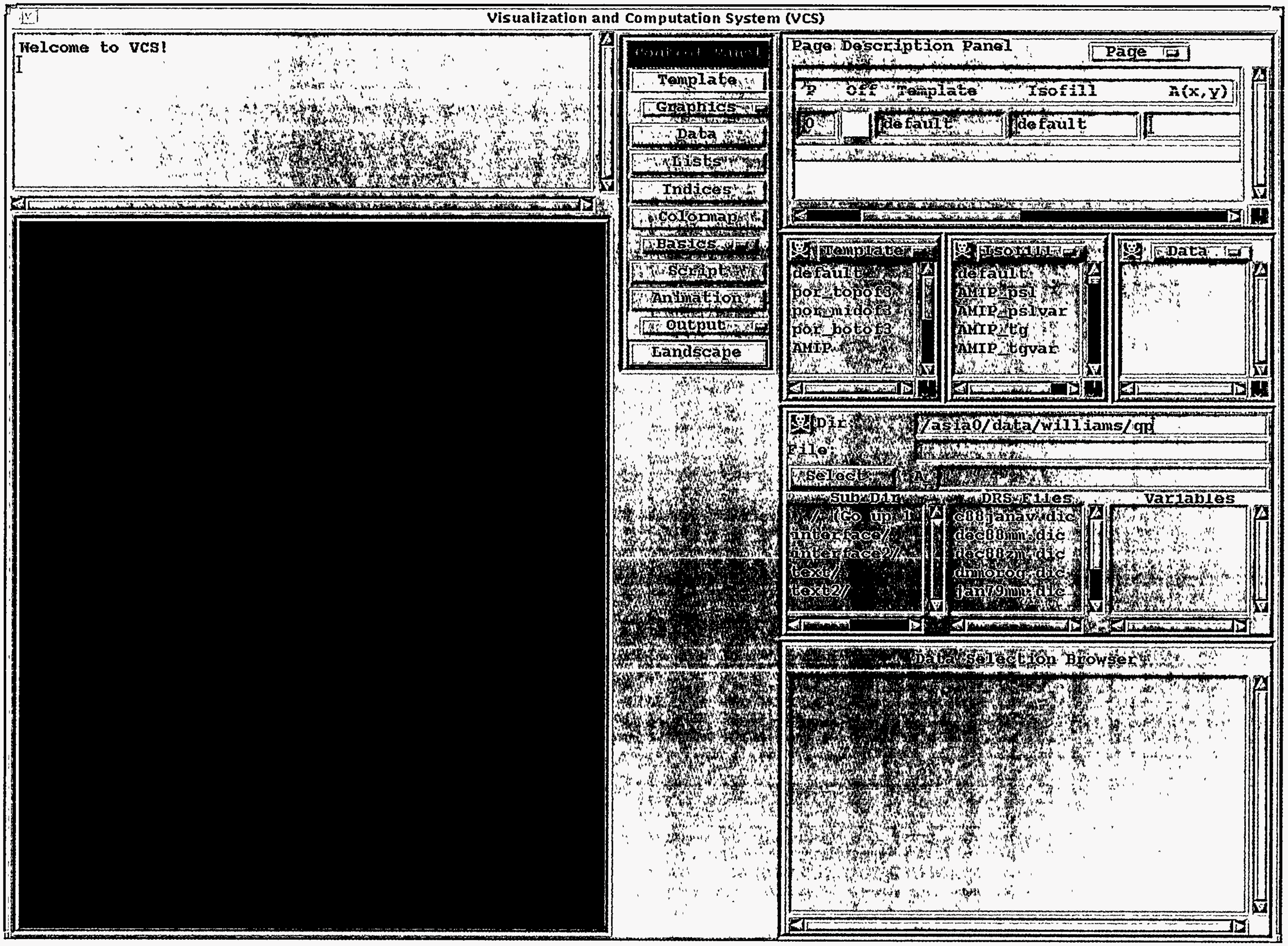
Isofill: default, AMIP_psl, AMIP_pslvar, AMIP_tg, AMIP_tgvar

Dir: /asia0/data/williams/gp

Sub Dir: ../ (Go up 1, interface/, interface2/, text/, text2/

DRS Files: c88janav.dic, dec88mm.dic, dec88zm.dic, dnmorog.dic, jan79mm.dic

Figure 1b. Initial Page with Portrait Orientation for Graphics



Visualization and Computation System (VCS)

Welcome to VCS!

Dir /asia0/data/williams/qp

### 3.1. Message Panel

The Message Panel is used exclusively to relay any error, warning, or information messages to the user. It cannot be edited, and none of the button or keyboard operations work. If editing is tried, the workstation will beep and nothing will happen.

If an error occurs, the error message will appear, and the workstation will beep. Information and warning messages will appear in the interface with no audio notification. Warning messages appear as: "Warning *–warning message.*" Error messages appear as: "Error*–error message.*"

When the **Graphics Canvas** is in portrait orientation, the **Message Panel** height shrinks from the bottom, as shown in Figure 1b.

The right and bottom scroll bars can be used to bring current and previous messages into view.

The **Message Panel** initially contains a welcoming message for the user of the Visualization and Computation System (**VCS**); "Welcome to VCS!" See Figures 1a and 1b.

### 3.2. Graphics Canvas

The **Graphics Canvas**, or workstation drawing area, is where the pictures defined in the **Page Description Forms** of the **Page Description Panel** are drawn.

Raster images of the **Graphics Canvas** can be saved and replayed for animation; thus, it is important that this drawing area be kept uncovered and in full view for the best possible image creation and viewing.

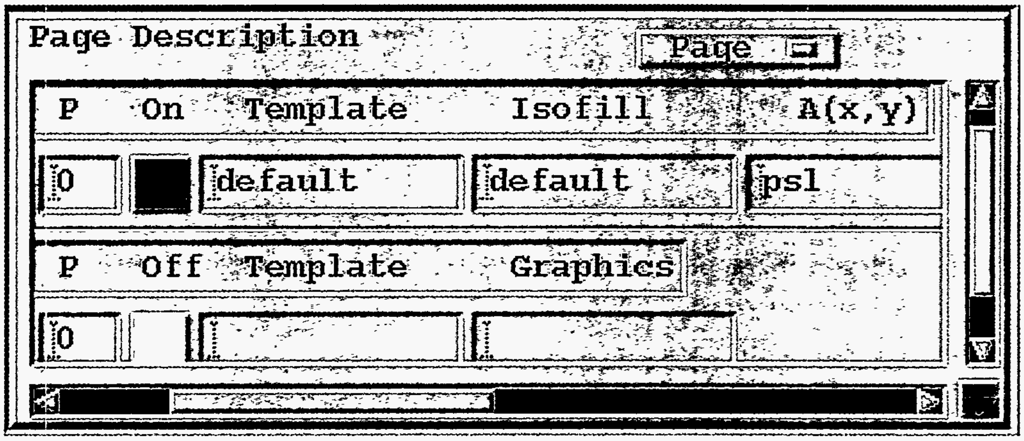
The picture can be partially clipped if a landscape picture is viewed when the **Graphics Canvas** is in portrait orientation, or if a portrait picture is viewed when the **Graphics Canvas** is in landscape orientation. If the picture is clipped, the user can either adjust the picture template, or choose another picture template from the table, or change the orientation.

The last picture(s) showing in the drawing area will be restored if the **VCS** interface is iconified and then deiconified, or if the **Graphics Canvas** is obstructed and then unobstructed. If the **Graphics Canvas** is obstructed or iconified during creation of a picture, it can only be recreated by turning the picture off and then on again. See the next section.

## 3.3. Page Description Panel

The **Page Description Panel** contains: an identifying label, “Page Description”; a **Page** menu button with menu selections for creating or removing **Picture Descriptor Forms**; a scroll window containing the list of **Picture Descriptor Forms**; a vertical and a horizontal scroll bar for viewing hidden **Picture Descriptor Forms**; an **Extension Button,** for extending the scroll window to view more **Picture Descriptor Forms**; and it may or may not contain **Picture Descriptor Forms**. The **Picture Descriptor Form** consists of two lines: the header line, which describes the user input items; and a line of input text windows with one colored button. See Figure 2.

Figure 2. Page Description Panel with two Picture Descriptor Forms



P On Template Isofill A(x,y)

0 ■ default default psl

P Off Template Graphics

0 □

The user can choose “Create” or “Remove” from the “Page” button menu. If “Create” is chosen, a new **Picture Descriptor Form** will be added to the end of the scroll window; it will be scrolled down to show the new **Picture Descriptor Form,** and the scroll window will be scrolled left as shown in Figure 2. A unique picture name is defined at the time a new **Picture Descriptor Form** is created. It is scrolled off the left side of the panel in Figure 2.

To remove one or more **Picture Descriptor Form(s)** from the scroll window, the user must click on the header lines (causing them to become highlighted), then choose “Remove” from the **Page** menu button. If, prior to choosing “Remove,” the user clicks on a highlighted header line, it will lose the highlight, and the **Picture Descriptor Form** will be taken off the removal list.

The mouse button remove operation (described earlier in section **3. Interaction**) will immediately remove a single **Picture Descriptor Form.**

If a **Picture Descriptor Form** describes an active picture on the **Graphics Canvas**, and it is removed from the scroll window, then the picture is also removed.

The full input for a **Picture Descriptor Form** is, from left to right:

ID — Input text window for the picture name.

P — Input text window for a picture priority number.

Status button — For turning the picture on and off and indicating the state of the display.

Template — Input text window for a **Picture Template** attribute set name.

Graphics — Input text window for a **Graphics Method** attribute set name.

Data (none or more) — Input text windows for **Array Data** attribute set names.

The "ID" input text window is not visible initially or when a new **Picture Descriptor Form** is created. The user must slide the horizontal scroll bar to the left to view it. To change a picture name, the user must first click in the "ID" input text window (the text window will become outlined), then use the keyboard to enter a unique picture name, and press the Return or Enter key to cause the rename to take effect.

To change the priority of the picture, the "P" input text window must be changed first by clicking in the text window (the input text window will become outlined), and then by using the keyboard to edit the value. After editing the text window, press the Return or Enter key for the priority change to take affect. The input text window registers only integer input. If the input is other than an integer, then after pressing the Return or Enter key, a zero value is shown and entered into the display attributes. The picture with the highest priority number is displayed on top of all other pictures.

The status button both controls and indicates the status of the picture. It has three states: "Off," the button is yellow; "On," the button is green; and "Err" (error), the button is red. If the **Picture Descriptor Form** is not complete, then the status button is yellow (see the second **Picture Descriptor Form** in Figure 2). If the picture is drawn, then the status button is green (see the first **Picture Descriptor Form** in Figure 2). A picture can be turned on or off by clicking this status button. If the button is red, then there is a problem with either the data or the template or the graphics attributes.

The "Template" input text window is not editable. If the user tries to change it with the keyboard, the workstation will beep and nothing will change. The only way to change the input text window is with a mouse copy and drop from the **Template Panel**. This operation will enter or replace the name in the **Picture Descriptor Form** and in the table of display attributes. This input text window will only accept a drop from the **Template Panel.**

The "Graphics" input text window is not editable. If the user tries to change it with the keyboard, the workstation will beep and nothing will change. The only way to change the input text window is with a mouse copy and drop from a **Graphics Method Panel.** This operation will enter or replace the name in the **Picture Descriptor Form** and in the table of display attributes. This input text window will only accept a drop from one of the **Graphics Method Panels.** After this operation is successfully completed, none or more "Data" input text windows will appear to the right with headers indicating the required number of multi-valued dimensions for each variable (see Figure 2). The label above the "Graphics" input text window will change from "Graphics" to the name of the **Graphics Method** (i.e., Isoline, Isofill, etc.).

The "Data" input text window is not editable. If the user tries to change it with the keyboard, the workstation will beep and nothing will change. The only way to change the input text window is with a mouse copy and drop from the **Array Data Panel.** This operation will enter or replace the name in the **Picture Descriptor Form.** This text window will only accept a drop from the **Array Data Panel.** The user must slide the horizontal scroll bar to the right in order to view any other "Data" input text windows.

Once the **Picture Descriptor Form** is completed and all the entries are satisfactory, the status button will turn green, the status label will indicate "On," and the picture will be drawn in the **Graphics Canvas.**

Script assignment of these attributes is shown in section **4.1. Display Attributes—D_name.**

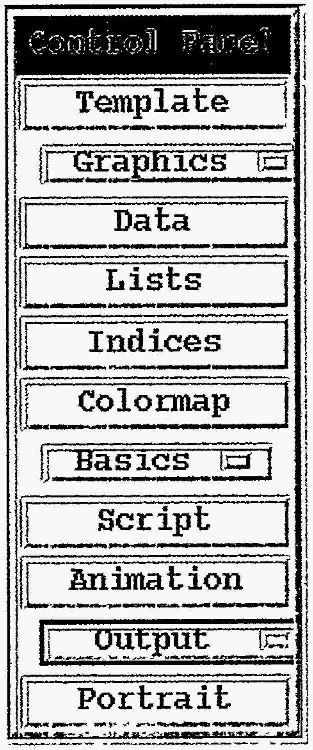
## 3.4. Control Panel

The **Control Panel** (see Figure 3) consists of a label, "Control Panel," identifying the panel, and the following control buttons:

| | |
|---|---|
| Template | Button for controlling the **Picture Template Panel.** |
| Graphics | Menu button for controlling the **Graphics Method Panels.** |
| Data | Button for controlling the **Array Data Panel.** |
| Lists | Button for controlling the **Lists Panel.** |
| Indices | Button for controlling the **Indices Panel.** |
| Colormap | Button for controlling the **Colormap Editor Panel.** |
| Basics | Menu button for controlling the most basic elements. |
| Script | Button for controlling the **Script Panel.** |

| | |
|---|---|
| Animation | Button for controlling the **Animation Panel.** |
| Output | Menu button for controlling the **Postscript or Raster Panel.** |
| Portrait/Landscape | Toggle button to control the orientation of the **Graphics Canvas.** |

Figure 3. Control Panel



The functions, that are controlled by the **Control Panel**, are discussed in detail below.

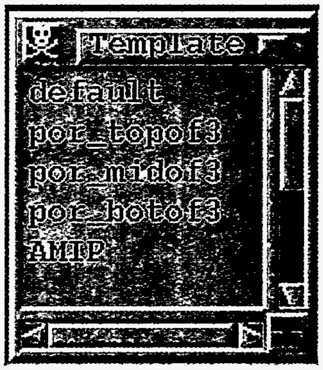
### *3.4.1. Template Panel*

A picture template is a primary element attribute set. Each of its sub-attributes defines the location or space for a segment of a picture, and methods are defined for the display of all segments except the legend and data. The table of attribute set names for picture templates are shown in the **Template Panel**. Picture template attributes describe the placement of segments of a picture, which include: identifying textual information, formatted values, axis tick marks and labels, boxes and lines, a legend, and the graphic representation of the data. They also describe how to display all but the data and legend. See section **4.2.1. Picture Template Attributes—P_name**, for definitions of the attributes of picture templates.

When the "Template" button is clicked in the **Control Panel**, the **Template Panel** appears below the **Page Description Panel**. If the **Template Panel** is visible and the "Template" button is clicked in the **Control Panel**, then the **Template Panel** and its sub-panels disappear. See Figure 4.

This panel contains: a **Terminate Button** (see section **3. INTERACTION**); a “Template” menu button; a scroll window that lists the attribute set names from the picture template table; vertical and horizontal scroll bars to view hidden portions of the list; and an **Extension Button** (see section **3. INTERACTION**).

Figure 4. Template Panel



default
por_topof3
por_midof3
por_botof3
AMIP

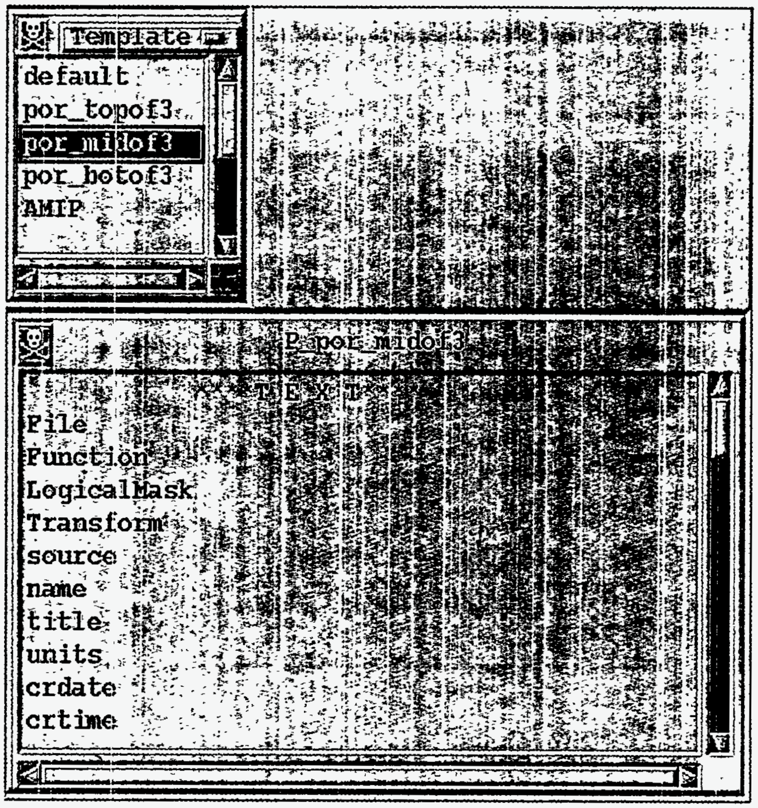
The “Template” menu button brings up a menu of three functions: “Copy,” “Rename,” and “Remove.” The user must first mark one of the picture template names, then bring up the menu and choose a function. If “Copy” is chosen, then a popup window will appear with an input text window and a “Save” and “Cancel” button. Enter the new picture template name in the input text window, and then either click on the “Save” button to copy the picture template, or click on the “Cancel” button to abort the “Copy” operation. If “Rename” is chosen, then another popup window will appear with an input text window, showing the selected picture template name, a “Save” button, and a “Cancel” button. Enter the new picture template name in the input text window, and either click on the “Save” button to change the picture template name, or click on the “Cancel” button to abort the “Rename” operation. If “Remove” is chosen, then the picture template name is removed from the picture template list. The user can also press and release the right mouse button to remove a single picture template immediately.

Click on a name in the **Template Panel**, and the **Template Browser Panel** will appear.

### *3.4.1.1. Template Browser Panel*

This panel consists of: a **Terminate Button** (see section **3. INTERACTION**), a label that identifies the name chosen from the **Template Panel**, a scroll window with a list of names of sub-attributes, and vertical and horizontal scroll bars to view hidden portions of the scroll window. See Figure 5

Figure 5. Template Panel and Template Browser Panel



The label, “P_por_midof3,” indicates the name that was chosen from the **Template Panel**. The “P_” prefix on the label name indicates it is a picture template.

The **Template Browser Panel** shows nine categories of picture template sub-attributes: text, format, x-tick marks; y-tick marks; x-labels, y-labels, box/lines, legend space, and display space. These picture template categories are identified in the browser’s scroll window with four astericks before and four after the category name (i.e., ****TEXT****, see Figure 5). After the category name is the list of sub-attribute names (i.e., File, Function, etc., see Figure 5). To see more picture template subset categories, scroll the window with the vertical scroll bar.

See section **4.2.1. Picture Template Attributes—P_name** for the script representation of picture template assignments.

### 3.4.1.1.1. Template Text Editor Panel

If the user clicks on one of the text category of sub-attribute names in the **Template Browser Panel** (e.g., “File,” “Function,” “LogicalMask,” etc.), then the **Template Text Editor Panel** will appear. See Figure 6.

This panel consists of:

**Terminate Button** See section **3. INTERACTION.**

Label Sub-attribute name chosen from the **Template Browser Panel.**

p Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

x Input line for the horizontal coordinate of the text reference point, with a text input window for keyboard entry.

y Input line for the vertical coordinate of the text reference point, with an input text window for keyboard entry.

Tt Input line for the text attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements’ **Text (Tt) Panel.**

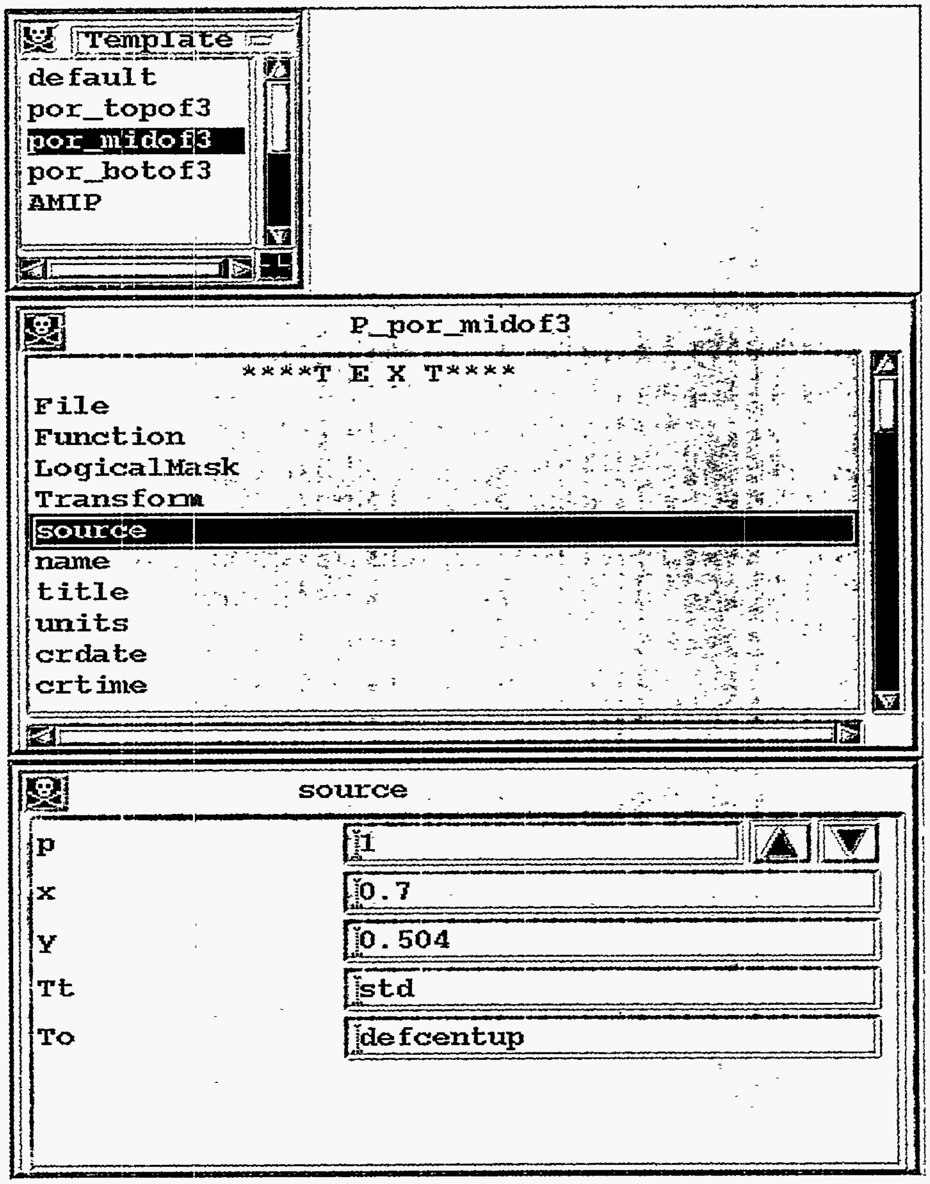
To Input line for the text orientation attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements’ **Text Orientation (To) Panel.**

An “Apply” and a “Reset” button will appear on the top border if any change is made to the input text windows. The “Apply” button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The “Reset” button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the text will not be drawn; otherwise, this priority is added to the picture priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinates (x) and (y) of the text reference point allow the user to change the location of the text segment. The position is given in a normalized device coordinate (ndc) space. If either of the coordinates is outside the range ($0 < x$ or $y < 1$), then the text will not be drawn.

Figure 6. Template, Template Browser, and Template Text Editor Panels



p 1

x 0.7

y 0.504

Tt std

To defcentup

3.4.1.1.2. Template Format Editor Panel

If the user clicks on one of the format category of sub-attribute names in the **Template Browser Panel** (e.g., xvalue, yvalue, etc.), then the **Template Format Editor Panel** will appear. See Figure 7.

This panel consists of the following:

| | |
|---|---|
| **Terminate Button** | See section **3. INTERACTION.** |
| Label | Sub-attribute name chosen from the **Template Browser Panel.** |

p Input line for the priority with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

x Input line for the horizontal coordinate of the text reference point, with an input text window for keyboard entry.

y Input line for the vertical coordinate of the text reference point, with an input text window for keyboard entry.

Th Input line for the format attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Format (Th) Panel.**

Tt Input line for the text attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Text (Tt) Panel.**

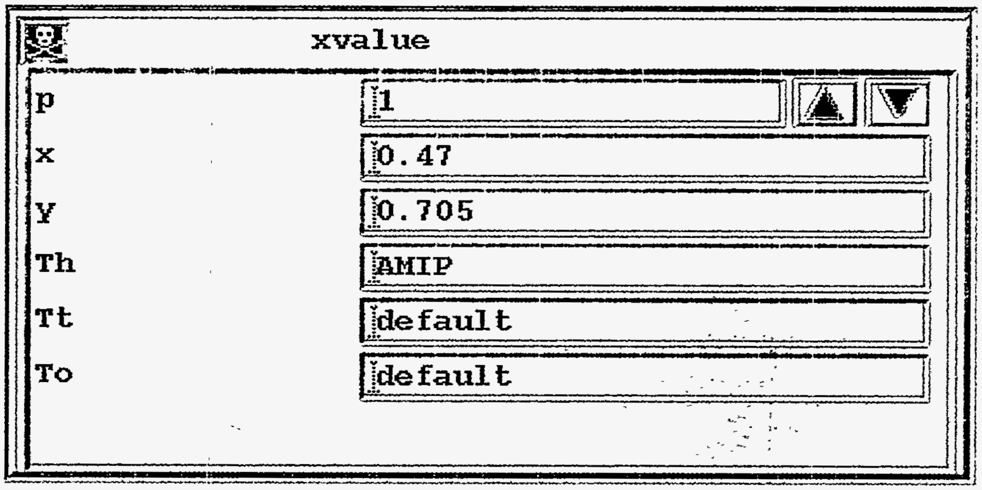
To Input line for the text orientation attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Text Orientation (To) Panel.**

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The "Reset" button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the value will not be formatted and the text will not be drawn; otherwise, this priority is added to the picture priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinates (x) and (y) of the text reference point allow the user to change the location of the text segment. The position is given in a normalized device coordinate (ndc) space. If either of the coordinates is outside the range ($0 < x$ or $y < 1$), then the value will not be formatted and the text will not be drawn.

Figure 7. Template Format Editor Panel



xvalue

p 1

x 0.47

y 0.705

Th AMIP

Tt default

To default

3.4.1.1.3. Template X-Tick Mark Editor Panel

If the user clicks on one of the x-tick marks category of sub-attribute names in the **Template Browser Panel** (e.g., xtic#1, xtic#2, etc.), then the **Template X-Tick Mark Editor Panel** will appear. See Figure 8.

This panel consists of the following:

**Terminate Button** See section **3. INTERACTION.**

Label Sub-attribute name chosen from the **Template Browser Panel.**

p Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

y1 Input line for the vertical coordinate of one end of the tick mark, with an input text window for keyboard entry.

y2 Input line for the vertical coordinate of the other end of the tick mark, with an input text window for keyboard entry.

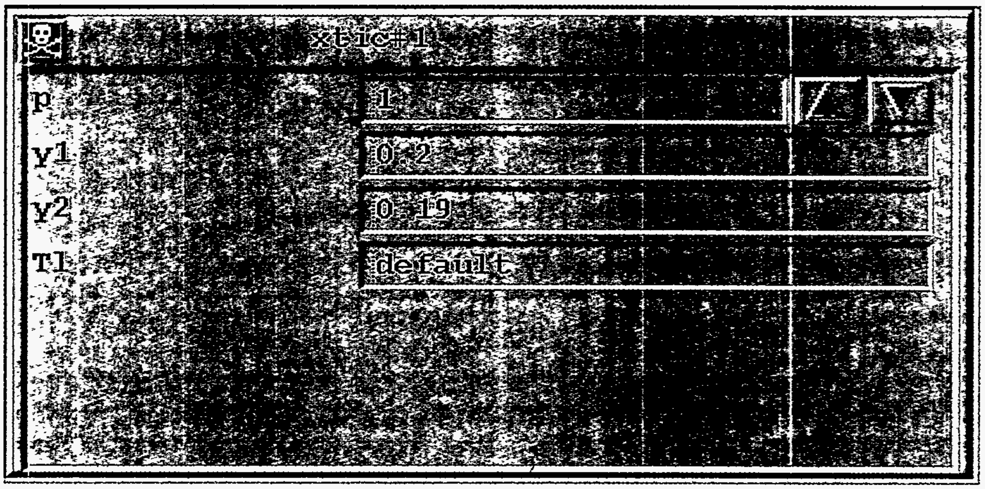
Tl Input line for the line attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Lines (Tl) Panel.**

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas**, then that picture will change appropriately. The "Reset" button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then no tick marks are drawn; otherwise, this priority is added to the picture priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinates (y1 and y2) of tick marks allow the user to change the positions of the ends (top and bottom) of vertical tick marks. The positions are given in a normalized device coordinate (ndc) space. If either of the coordinates is outside the range (0 < y1 or y2 < 1), then the tick marks will not be drawn. The horizontal coordinate for each of the vertical tick marks will be assigned, in real-world coordinates, by the graphics method attribute set that is used to define the graphic representation of data (see section **3.4.2. Graphics Method Panel**).

Figure 8. Template X-Tick Mark Editor Panel



xtic#1

p 1

y1 0.2

y2 0.19

T1 default

3.4.1.1.4. Template Y-Tick Mark Editor Panel

If the user clicks on one of the y-tick marks category of sub-attribute names in the **Template Browser Panel** (e.g., ytic#1, ytic#2, etc.), then the **Template Y-Tick Mark Editor Panel** will appear. See Figure 9.

This panel consists of the following:

| | |
|---|---|
| **Terminate Button** | See section **3. INTERACTION.** |
| Label | Sub-attribute name chosen from the **Template Browser Panel.** |
| p | Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value. |
| x1 | Input line for the horizontal coordinate of one end of the tick mark, with an input text window for keyboard entry. |

x2 Input line for the horizontal coordinate of the other end of the tick mark, with an input text window for keyboard entry.

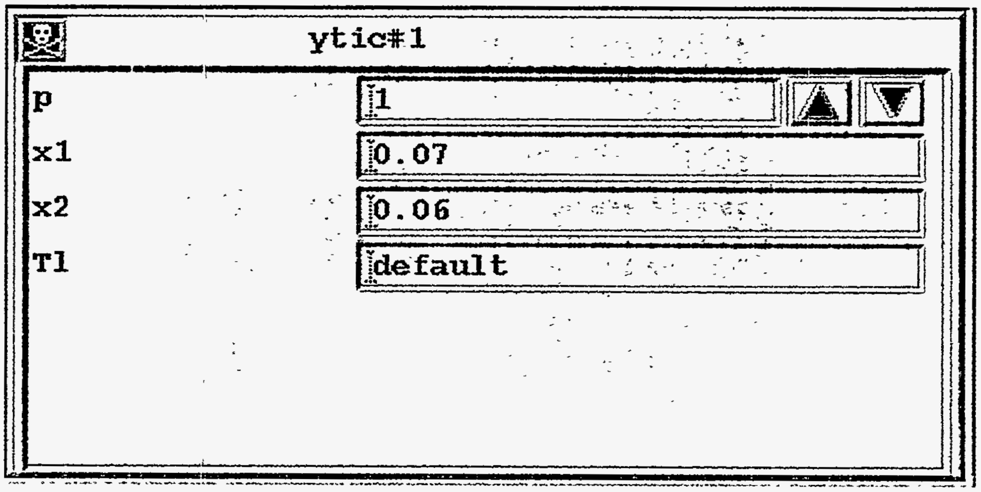
Tl Input line for the line attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Lines (Tl) Panel.**

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The "Reset" button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the tick marks will not be drawn; otherwise, this priority is added to the picture priority, and the sum divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinates (x1 and x2) of tick marks allow the user to change the positions of the ends (left and right) of horizontal tick marks. The positions are given in a normalized device coordinate (ndc) space. If either of the coordinates is outside the range (0 < x1 or x2 < 1), then the tick marks will not be drawn. The vertical coordinate for each of the horizontal tick marks will be assigned, in real-world coordinates, by the graphics method attribute set that is used to define the graphic representation of data (see section **3.4.2. Graphics Method Panel**)

Figure 9. Template Y-Tick Mark Editor Panel



ytic#1

| | |
|---|---|
| p | 1 |
| x1 | 0.07 |
| x2 | 0.06 |
| Tl | default |

### 3.4.1.1.5. Template X-Label Editor Panel

If the user clicks on one of the x-label category of sub-attribute names in the **Template Browser Panel** (e.g., xlabel#1, xlabel#2), then the **Template X-Label Editor Panel** will appear. See Figure 10.

This panel consists of the following:

**Terminate Button** See section **3. INTERACTION.**

Label Sub-attribute name chosen from the **Template Browser Panel.**

p Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

y Input line for the vertical coordinate of the reference point for the horizontal axis labels, with an input text window for keyboard entry.

Tt Input line for the text attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Text (Tt) Panel.**

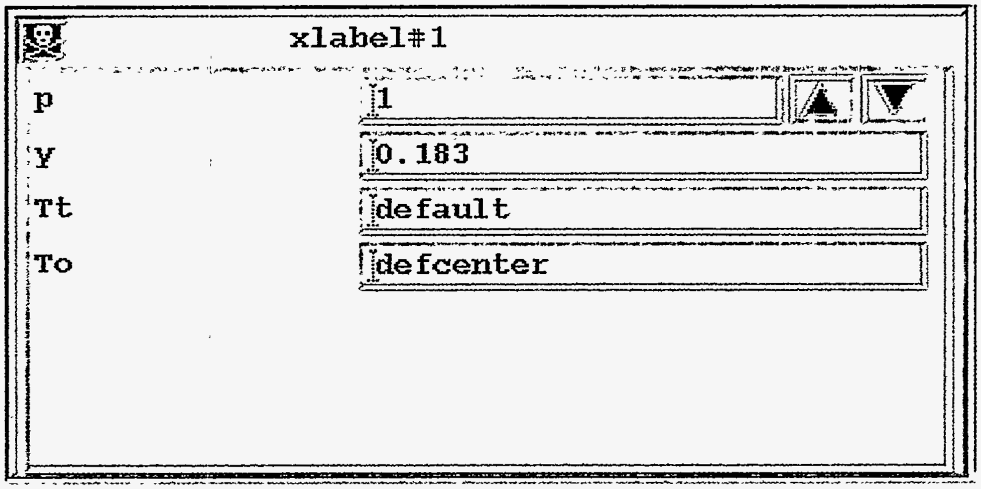
To Input line for the text orientation attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Text Orientation (To) Panel.**

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The "Reset" button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the labels will not be drawn; otherwise, this priority is added to the picture priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinate (y) of horizontal axis labels allows the user to change the vertical positions of labels for the horizontal axis. The position is given in a normalized device coordinate (ndc) space. If the coordinate is outside the range ($0 < y < 1$), then the labels will not be drawn. The horizontal coordinate and label string for each label will be assigned, in real-world coordinates, by the graphics method attribute set that is used to define the graphic representation of data (see section **3.4.2. Graphics Method Panel**).

Figure 10. Template X-Label Editor Panel



xlabel#1

p 1

y 0.183

Tt default

To defcenter

### 3.4.1.1.6. Template Y-Label Editor Panel

If the user clicks on one of the y-label category of sub-attribute names in the **Template Browser Panel** (e.g., ylabel#1, ylabel#2), then the **Template Y-Label Editor Panel** will appear. See Figure 11.

This panel consists of the following:

**Terminate Button** See section **3. INTERACTION.**

Label Sub-attribute name chosen from the **Template Browser Panel.**

p Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

x Input line for the horizontal coordinate of the reference point for the vertical axis labels, with an input text window for keyboard entry.

Tt Input line for the text attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Text (Tt) Panel.**

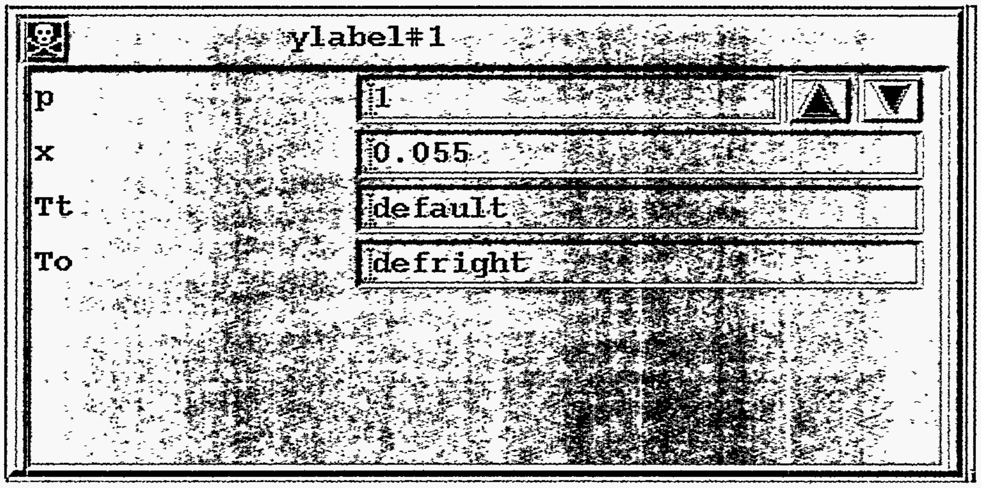
To Input line for the text orientation attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Text Orientation (To) Panel.**

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The “Reset” button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the labels will not be drawn; otherwise, this priority is added to the picture priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinate (x) of vertical axis labels allows the user to change the horizontal positions of labels for the vertical axis. The position is given in a normalized device coordinate (ndc) space. If the coordinate is outside the range (0 < x < 1), then the labels will not be drawn. The vertical coordinate and label string for each label will be assigned, in real-world coordinates, by the graphics method attribute set that is used to define the graphic representation of data (see section **3.4.2. Graphics Method Panel**).

Figure 11. Template Y-Label Editor Panel



p 1 ▲ ▼

x 0.055

Tt default

To defright

3.4.1.1.7. Template Box/Lines Editor Panel

If the user clicks on one of the box/lines category of sub-attribute names in the **Template Browser Panel** (e.g., box#1, line#1, etc.), then the **Template Box/Lines Editor Panel** will appear. See Figure 12.

This panel consists of the following:

**Terminate Button** See section **3. INTERACTION.**

Label Sub-attribute name chosen from the **Template Browser Panel**.

p Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

x1 Input line for the horizontal coordinate of the first point of a line or the first corner of a box, with an input text window for keyboard entry.

x2 Input line for the horizontal coordinate of the second point of a line or the diagonally opposite corner of a box, with an input text window for keyboard entry.

y1 Input line for the vertical coordinate of the first point of a line or the first corner of a box, with an input text window for keyboard entry.

y2 Input line for the vertical coordinate of the second point of a line or the second corner of a box, with an input text window for keyboard entry.

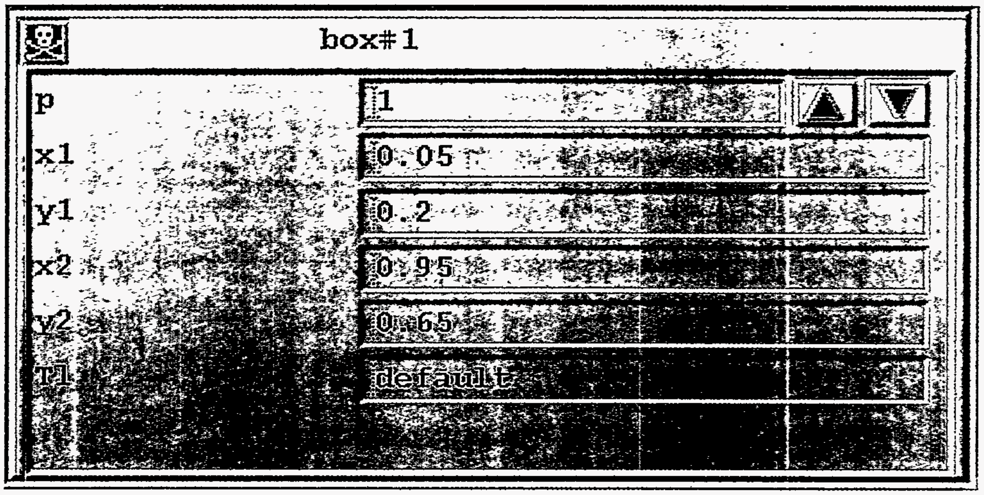
Tl Input line for the line attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Lines (Tl) Panel**.

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The "Reset" button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the box or line will not be drawn; otherwise, this priority is added to the picutre priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinates (x1,y1) and (x2,y2) of a line or box allow the user to change the positions of lines and boxes that can be used to enhance a picture. The positions are given in a normalized device coordinate (ndc) space. If a coordinate is outside the range (0 < x1 or y1 or x2 or y2 < 1), then the line or box will not be drawn.

Figure 12. Template Box/Lines Editor Panel



box#1

p 1

x1 0.05

y1 0.2

x2 0.95

y2 0.65

Tl default

3.4.1.1.8. Template Legend Space Editor Panel

If the user clicks on the name “legend” in the legend space category of sub-attribute names in the **Template Browser Panel**, then the **Template Legend Space Editor Panel** will appear. See Figure 13.

This panel consists of the following:

**Terminate Button** See section **3. INTERACTION.**

Label Sub-attribute name chosen from the **Template Browser Panel.**

p Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

x1 Input line for the horizontal coordinate of the first point of a rectangular area, with an input text window for keyboard entry.

x2 Input line for the horizontal coordinate of a diagonal point of a rectangular area, with an input text window for keyboard entry.

y1 Input line for the vertical coordinate of the first point of a rectangular area, with an input text window for keyboard entry.

y2 Input line for the vertical coordinate of a diagonal point of a rectangular area, with an input text window for keyboard entry.

Tt Input line for the text attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements’ **Text (Tt) Panel.**

To Input line for the text orientation attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements’ **Text Orientation (To) Panel.**

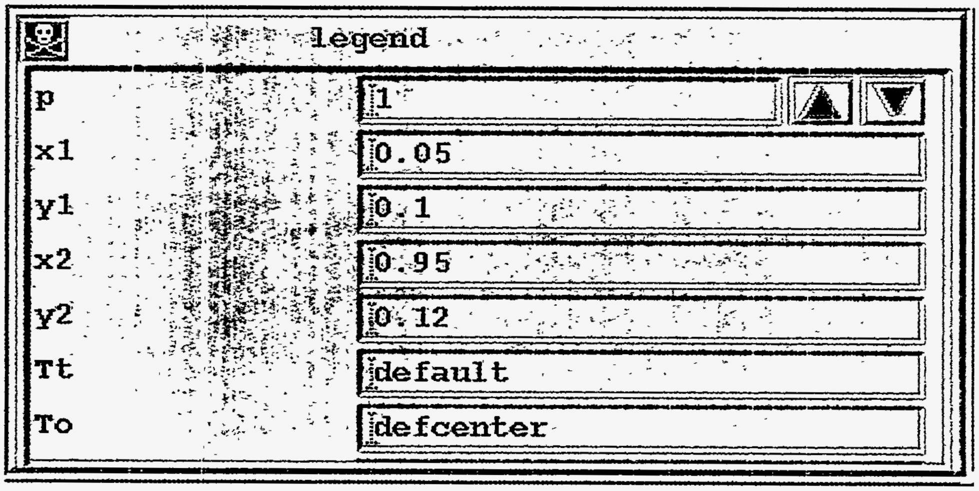
**Tl** Input line for the line attribute set name, with an input text window for keyboard entry or copy and drop insertion from the basic elements' **Lines (Tl) Panel.**

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The "Reset" button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the legend will not be drawn; otherwise, this priority is added to the picture priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinates (x1,y1) and (x2,y2) of a rectangular space allows the user to change the position of the space to be used, by a graphics method, for a legend. The positions are given in a normalized device coordinate (ndc) space. If a coordinate is outside the range (0 < x1 or y1 or x2 or y2 < 1), then the legend will not be drawn.

Figure 13. Template Legend Space Editor Panel



legend

p 1

x1 0.05

y1 0.1

x2 0.95

y2 0.12

Tt default

To defcenter

### 3.4.1.1.9. Template Data Space Editor Panel

If the user clicks on the name “legend” in the legend space category of sub-attribute names in the **Template Browser Panel**, then the **Template Legend Space Editor Panel** will appear. See Figure 14.

This panel consists of the following:

**Terminate Button** See section **3. INTERACTION.**

Label Sub-attribute name chosen from the **Template Browser Panel.**

p Input line for the priority, with an input text window for keyboard entry, an up arrow button to increment the value, and a down arrow button to decrement the value.

x1 Input line for the horizontal coordinate of the first point of a rectangular area, with an input text window for keyboard entry.

x2 Input line for the horizontal coordinate of a diagonal point of a rectangular area, with an input text window for keyboard entry.

y1 Input line for the vertical coordinate of the first point of a rectangular area, with an input text window for keyboard entry.

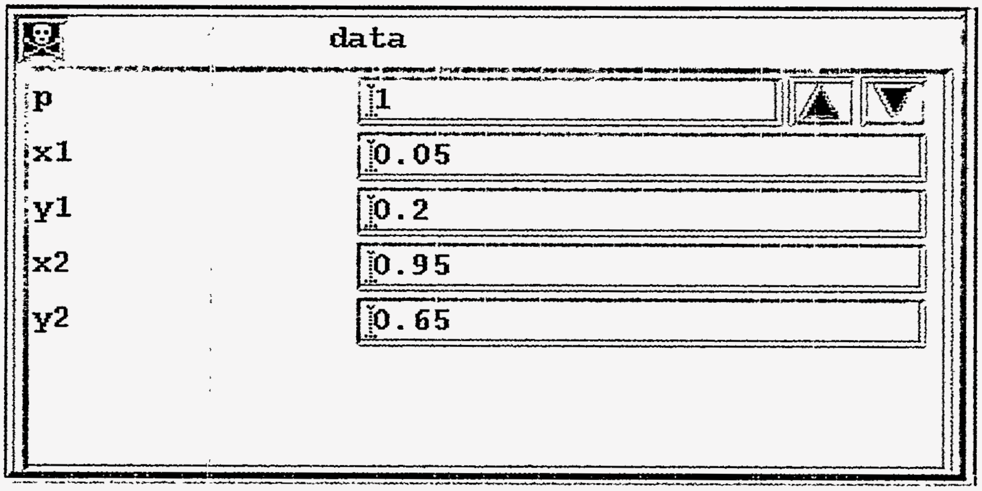
y2 Input line for the vertical coordinate of a diagonal point of a rectangular area, with an input text window for keyboard entry.

An “Apply” and a “Reset” button will appear on the top border if any change is made to the input text windows. The “Apply” button will apply changes that have been made. If the template is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The “Reset” button will reset any changes to their prior value if they have not been applied.

If the priority given here is 0, then the data will not be drawn; otherwise, this priority is added to the picture priority, and the sum is divided by 1000 to determine the total segment priority. The up and down arrow buttons can be used to alter the priority value.

The coordinates (x1,y1) and (x2,y2) of a rectangular space allows the user to change the position of the space to be used, by a graphics method, for a legend. The positions are given in a normalized device coordinate (ndc) space. If a coordinate is outside the range (0 < x1 or y1 or x2 or y2 < 1), then the graphics representation of the data will not be displayed.

Figure 14. Template Data Space Editor Panel



data

p 1

x1 0.05

y1 0.2

x2 0.95

y2 0.65

### *3.4.2. Graphics Method Panel*

A graphics method defines how the array data will be graphically represented on the **Graphics Canvas**. See section **4.2.2. Graphics Attributes—G*_name**, for more details on the graphics attributes.

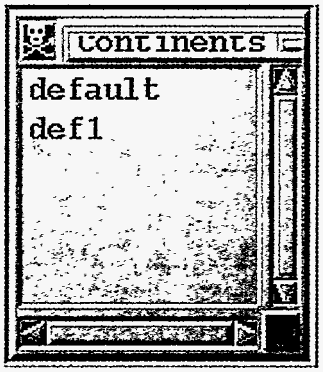
The **Graphics** button in the **Control Panel** is a menu button. The menu includes the abbreviated names for graphics methods that have been implemented in VCS, they are:

| | |
|---|---|
| Continents | Draws a generic set of continental outlines in longitude by latitude space. |
| Isoline | Draws lines of constant value to represent two dimensional arrays of data. |
| Isofill | Fills the area between lines of constant value. |
| Outline | Outlines a set of integer values that occur in a two dimensional array. |
| Outfill | Fills the area where a set of integer values occur in a two dimensional array. |

When one of these methods is chosen from the menu, a **Graphics Method Panel** for that method will appear below the **Page Description Panel**. If a **Graphics Method Panel** already exists, it and all sub-panels will be removed prior to showing the new panel. Selecting **Graphics** in the menu will remove an existing **Graphics Method Panel** and its sub-panels.

This panel contains: a **Terminate Button** (see section **3. INTERACTION**); a **Graphics Method** menu button, with an abbreviated name of the graphics method on it, that will pop-up a menu with "Copy," "Rename," and "Remove" options; a scroll window to view the table of graphics method attribute set names; a vertical and a horizontal scroll bar to view hidden portions of the table; and an **Extension Button** to enlarge the panel and allow the user to view a larger portion of the table. See Figure 15

Figure 15. Graphics Method Panel



To modify the table of attribute set names, mark an attribute set name and then choose the option from the **Graphics Method** menu. If the "Copy" option is chosen, then a popup window will appear with an input text window, a "Save" button, and a "Cancel" button. Enter the new graphics method name, and click on the "Save" button to copy the marked attribute set to the newly named attribute set, or click on the "Cancel" button to abort the operation. If the "Rename" option is chosen, then a similar popup window will appear with an input text window, showing the selected graphics method name, a "Save" button, and a "Cancel" button. Enter or modify the new attribute set name, and click on the "Save" button to change the attribute set name, or click on the "Cancel" button to abort the operation. If "Remove" is chosen, then the attribute set name is removed from the panel and the table. The user can also press and release the right mouse button to remove an attribute set immediately.

When a graphics method attribute set name is chosen from the scroll window, a graphics method editor will appear (see Figure 16). The first 13 attributes are common to all graphics methods:

| | |
|---|---|
| **project** | Name of the graphics projection. |
| **xticlabels#1** | List name for x-axis labels and ticks. |
| **xticlabels#2** | Second list name for x-axis labels and ticks. |
| **xmtics#1** | List name for x-axis minor ticks. |
| **xmtics#2** | Second list name for x-axis minor ticks. |
| **yticlabels#1** | List name for y-axis labels and ticks. |
| **yticlabels#2** | Second list name for y-axis labels and ticks. |
| **ymtics#1** | List name for y-axis minor ticks. |
| **ymtics#2** | Second list name for y-axis minor ticks. |
| **datawc_x1** | Left horizontal coordinate of the data space in real-world coordinates. |
| **datawc_y1** | Bottom vertical coordinate of the data space in real-world coordinates. |

**datawc_x2** Right horizontal coordinate of the data space in real-world coordinates.

**datawc_y2** Top vertical coordinate of the data space in real-world coordinates.

In VCS, only a linear projection is available.

No prefix is needed for the list names because a list name is the only valid entry (i.e., do not use L_name).

The input text windows that require a list name will accept either keyboard input after a click to highlight the window, or a mouse copy and drop operation to copy a name from the **List Panel**.

The real-world coordinates are used to scale the real-world dimensions of the data to the normalized device coordinate space for data, as defined in the picture template.

An "Apply" and a "Reset" button will appear on the top border if any change is made to the input text windows. The "Apply" button will apply changes that have been made. If the graphics method attribute set is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately. The "Reset" button will reset any changes to their prior value if they have not been applied.

### *3.4.2.1. Continents Editor Panel*

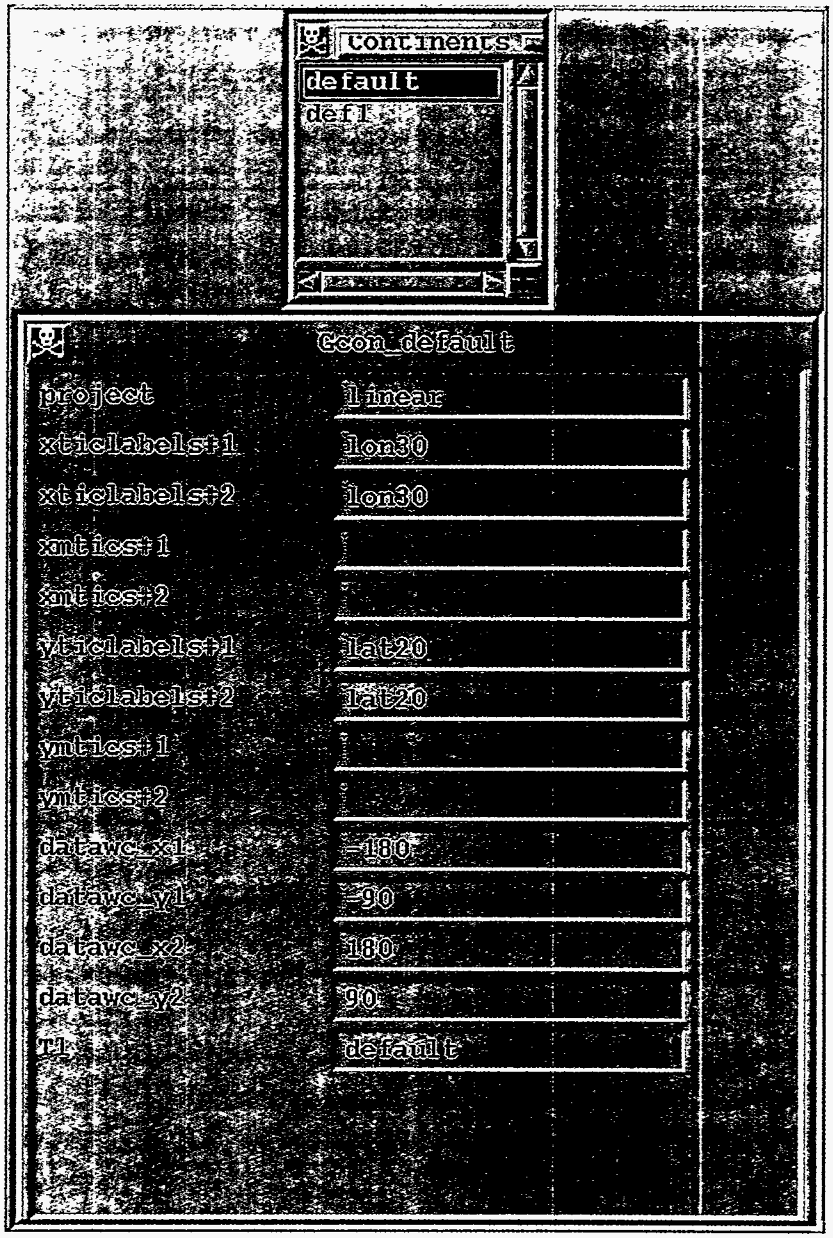
When the **Graphics Method Panel** is showing the Continents methods' table of attribute set names and an attribute set name is chosen, then the **Continents Editor Panel** is brought up (see Figure 16). The purpose of this graphic method is to draw a generic set of continental outlines. See section **4.2.2.5. Continents—Gcon_name** for more details on this graphics method.

This graphics method includes: the 13 input lines for entering attribute values as shown in section **3.4.2.**, and an input line for:

**Tl** Name of a line attribute set.

The (Tl) attribute defines the type of line to use (see section **3.4.7.2. Lines (Tl) and Editor Panels,** and section **4.3.5. Line Attributes—Tl_name**) for drawing the continents. It can be entered into the input text window either with a keyboard or with a mouse copy and drop from the basic elements' **Lines (Tl) Panel.**

Figure 16. Graphics Method and Continents Editor Panels



continents

default
def1

Gcon_default

| | |
|---|---|
| project | linear |
| xticlabels#1 | lon30 |
| xticlabels#2 | lon30 |
| xmtics#1 | |
| xmtics#2 | |
| yticlabels#1 | lat20 |
| yticlabels#2 | lat20 |
| ymtics#1 | |
| ymtics#2 | |
| datawc_x1 | -180 |
| datawc_y1 | -90 |
| datawc_x2 | 180 |
| datawc_y2 | 90 |
| T1 | default |

### *3.4.2.2. Isoline Editor Panel*

When the **Graphics Method Panel** is showing the Isoline methods' table of attribute set names and an attribute set name is chosen, then the **Isoline Editor Panel** is brought up (see Figure 17). The purpose of this graphic method is to draw a set of lines of constant value. See section **4.2.2.1. Isoline—Gi_name** for more details on this graphics method.

This graphics method includes: the 13 input lines for entering attribute values as shown in section **3.4.2.**, and an input line with an input text window for setting a switch (y/n), which determines whether the isolines will be labeled:

**make_labels** (y/n) determines whether labels will be drawn for isolines

It has an additional panel, that is side-wise scrollable from one set of isoline descriptors to another, for viewing and editing sets of isoline descriptors as defined below:

**id** Identification for the isoline descriptor set (to allow changes to each set).<br>
**priority** Determines which isoline takes preference when levels overlap.<br>
**level** Either the reference level or the only level to be drawn.<br>
**increment** Increment between isolines, beginning with the reference.<br>
**hilite_ci** Highlight color index of the background for labels.<br>
**label** Label for all isolines defined here; or if it is an asterisk (*), use the formatted level.<br>
**Tl** Line attribute set name for all levels defined here.<br>
**Tt** Text attribute set name for the labels of all isolines defined here.<br>
**To** Text orientation attribute set name for the labels of all isolines defined here.

The isolines that could possibly be displayed are those defined by I(k) in the equation:

$$I(k) = (\textbf{level}+k*\textbf{increment}) \text{ for all values of } k \geq 0.$$

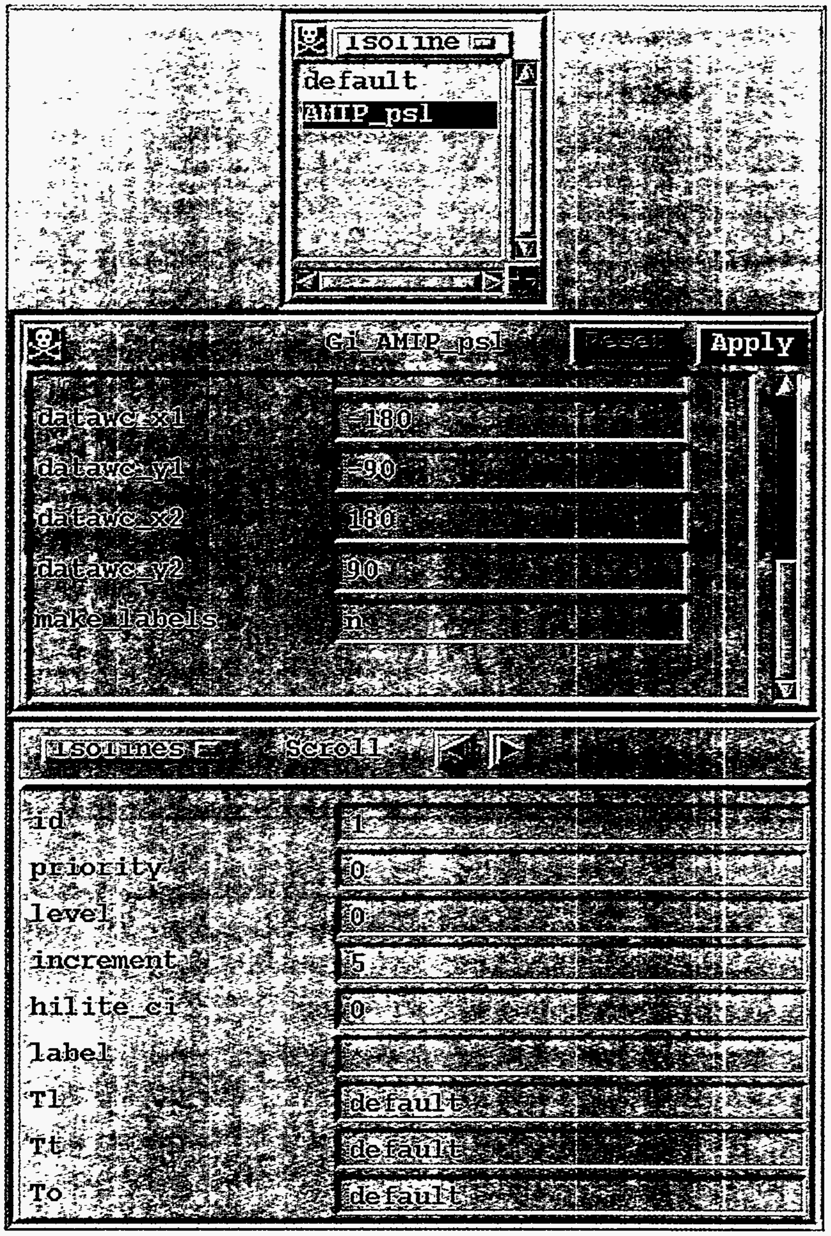
The range of the data values determines which of those are actually drawn.

When multiple isoline descriptors define the same isoline, it is the last defined line that is drawn, unless a higher **priority** is specified in an earlier isoline descriptor.

The (Tl) attribute defines the type of line to use (see section **3.4.7.2. Lines (Tl) and Editor Panels** and section **4.3.5. Line Attributes—Tl_name**) for drawing the isolines. The name can be entered into the input text window either with a keyboard or with a mouse copy and drop from the basic elements' **Lines (Tl) Panel.**

The (Tt) and (To) attributes define the text and text orientation to use (see section **3.4.7.1. Text (Tt), Text Orientation (To), and Editor Panels**; section **4.3.1. Text Attributes—Tt_name**; and section **4.3.2. Text Orientation Attributes—To_name**) for drawing labels on the isolines. The name can be entered into the input text window either with a keyboard or with a mouse copy and drop from the basic elements' **Text (Tt) Panel** and the **Text Orientation (To) Panel.**

Figure 17. Graphics Method and Isoline Editor Panels



default
AMIP_psl

Gi_AMIP_psl

datawc_x1 -180
datawc_y1 -90
datawc_x2 180
datawc_y2 90
make_labels n

id 1
priority 0
level 0
increment 5
hilite_ci 0
label
Tl default
Tt default
To default

*3.4.2.3. Isofill Editor Panel*

When the **Graphics Method Panel** is showing the Isofill methods' table of attribute set names and an attribute set name is chosen, then the **Isofill Editor Panel** is brought up (see Figure 18). The purpose of this graphic method is to fill between lines of constant value. See section **4.2.2.3. Isofill —Gfi_name** for more details on this graphics method.

This graphics method includes: the 13 input lines for entering attribute values as shown in section **3.4.2.**, and an additional panel that is side-wise scrollable from one set of isofill descriptors to another for editing multiple isofill descriptors:

| | |
|---|---|
| **id** | The identification for the isofill descriptor set (to allow changes to each set). |
| **level1** | One isolevel limit for the region to be filled. |
| **level2** | The other isolevel limit for the region to be filled. |
| **Tf** | The fill attribute set name for type of fill between the two levels. |

The (Tf) attribute defines the type of fill area to use (see section **3.4.7.2. Fill Area (Tf) and Editor Panel,** and section **4.3.3. Fill Area—Tf_name**) for filling between isolevels. The name can be entered into the input text window either with a keyboard or with a mouse copy and drop from the basic elements' **Fill Area (Tf) Panel.**

#### *3.4.2.4. Outline Editor Panel*

When the **Graphics Method Panel** is showing the Outline methods' table of attribute set names and an attribute set name is chosen, then the **Outline Editor Panel** is brought up (see Figure 19). The purpose of this graphic method is to draw an outline of atmospheric model continents using the model surface type array that indicates land, ocean, and sea-ice. See section **4.2.2.2. Outline—Go_name** for more details on this graphics method.

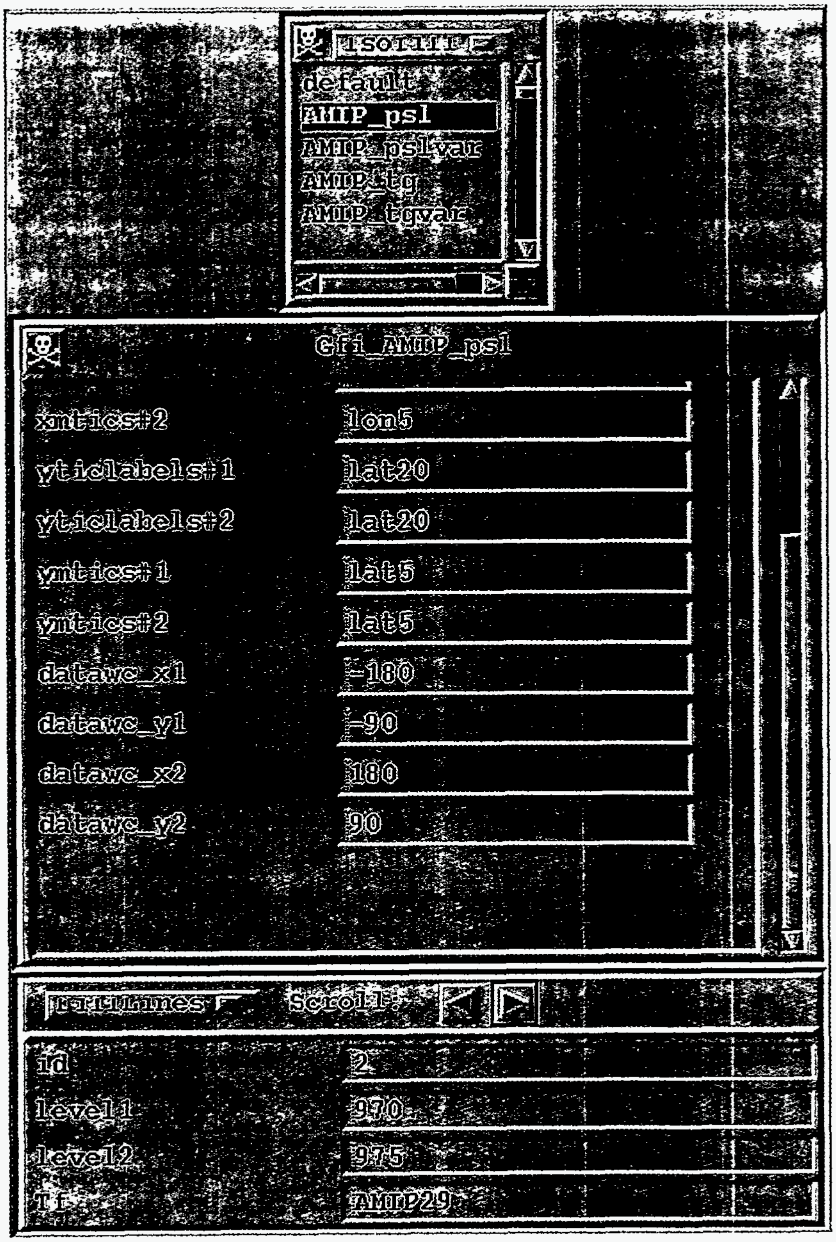
This graphics method includes: the 13 input lines for entering attribute values as shown in section **3.4.2.**, and two additional lines with input text windows:

| | |
|---|---|
| **Tl** | Name of a line attribute set. |
| **outline** | Integer values to outline. |

The (Tl) attribute defines the type of line to use (see section **3.4.7.2. Lines (Tl) and Editor Panels**, and section **4.3.5. Line Attributes—Tl_name**) for drawing the isolines. The name can be entered into the input text window either with a keyboard or with a mouse copy and drop from the basic elements' **Lines (Tl) Panel.**

The outline attribute defines the integer values to outline (up to 10 values).

Figure 18. Graphics Method and Isofill Editor Panels



default
AMIP_psl
AMIP_pslvar
AMIP_tg
AMIP_tgvar

Gfi_AMIP_psl

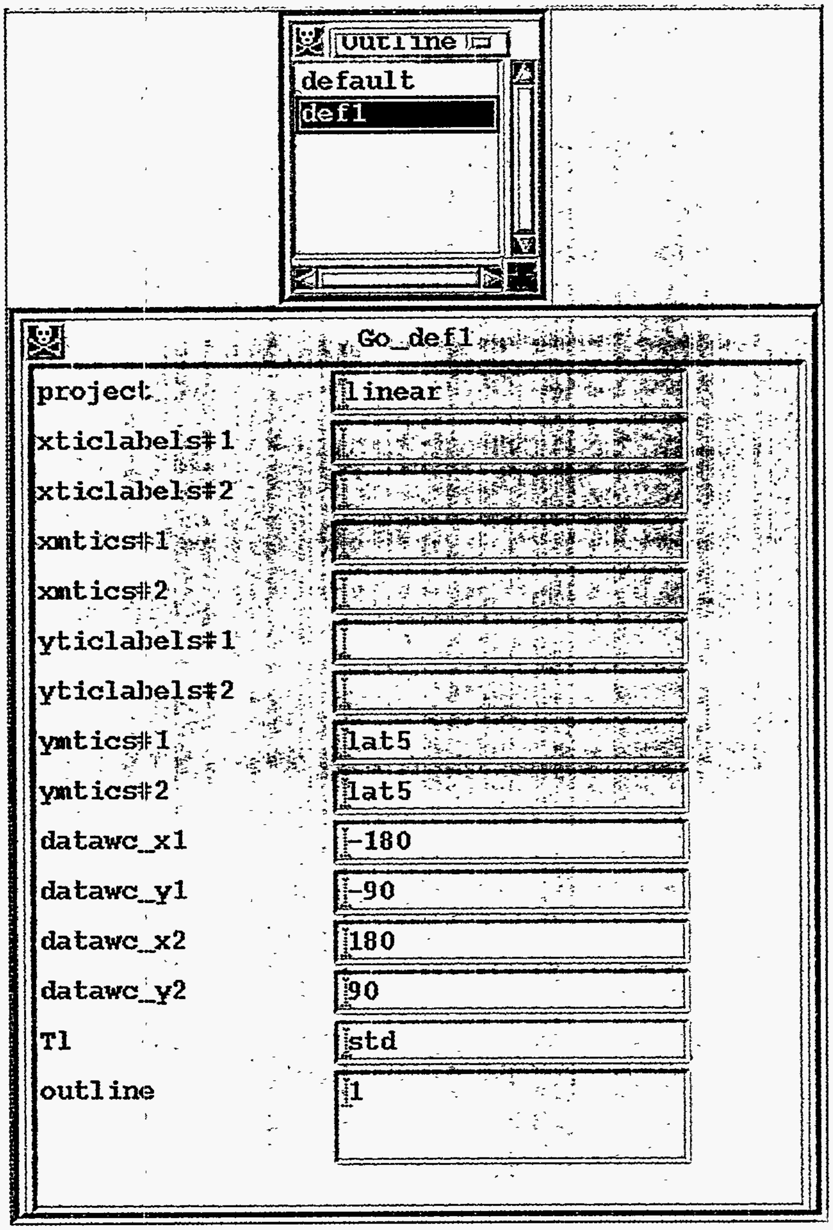
xmtics#2 lon5
yticlabels#1 lat20
yticlabels#2 lat20
ymtics#1 lat5
ymtics#2 lat5
datawc_x1 -180
datawc_y1 -90
datawc_x2 180
datawc_y2 90

id 2
level1 970
level2 975
Tf AMIP29

*3.4.2.5. Outfill Editor Panel*

When the **Graphics Method Panel** is showing the Outfill methods' table of attribute set names and an attribute set name is chosen, then the **Outfill Editor Panel** is brought up (see Figure 20). The purpose of this graphic method is to fill an outline of atmospheric model continents using the model surface type array that indicates land, ocean, and sea-ice. See section **4.2.2.4. Outline—Gfo_name** for more details on this graphics method.

Figure 19. Graphics Method and Outline Editor Panels



project linear
xticlabels#1
xticlabels#2
xmtics#1
xmtics#2
yticlabels#1
yticlabels#2
ymtics#1 lat5
ymtics#2 lat5
datawc_x1 -180
datawc_y1 -90
datawc_x2 180
datawc_y2 90
Tl std
outline 1

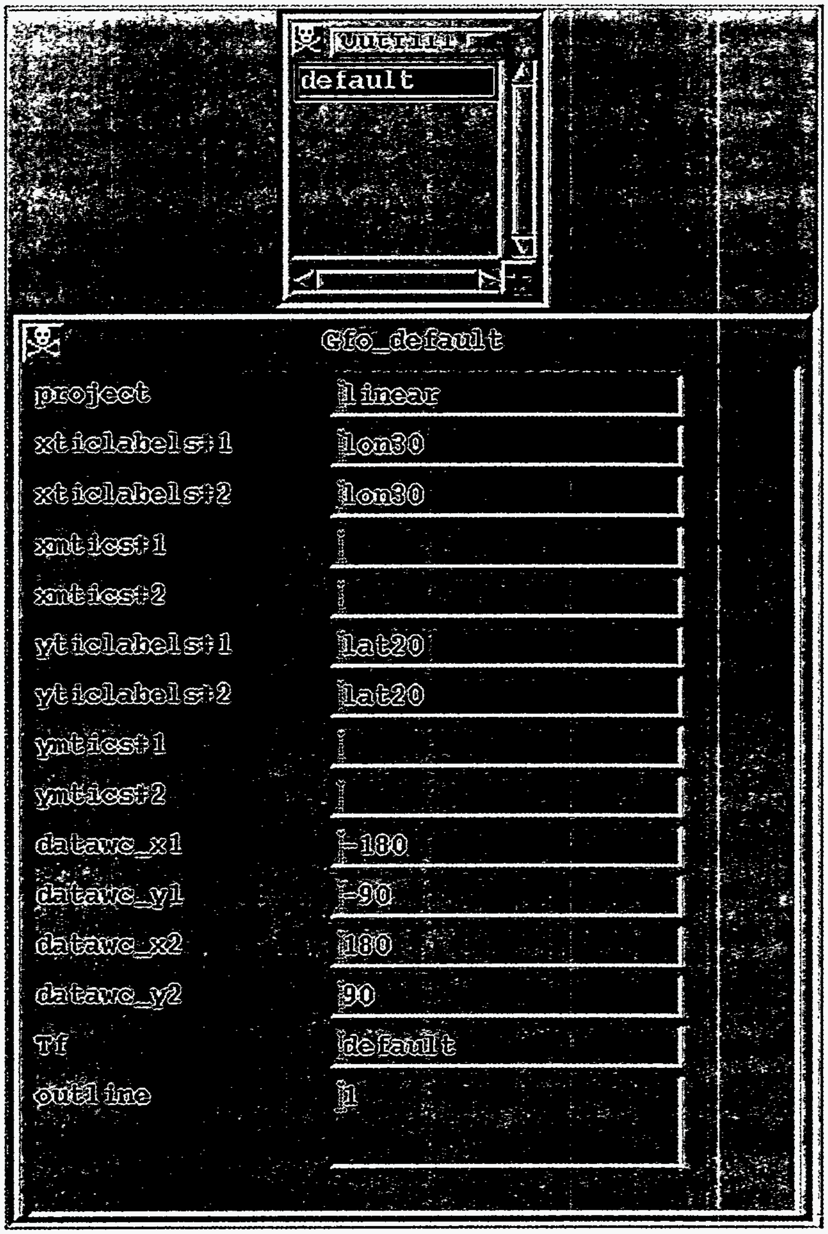
This graphics method includes: the 13 input lines for entering attribute values as shown in section **3.4.2.**, and two additional lines with input text windows:

| | |
|---|---|
| **Tf** | Name of a fill attribute set. |
| **outline** | Integer values to fill. |

The (Tf) attribute defines the type of fill area to use (see section **3.4.7.2. Fill Area (Tf) and Editor Panel** and section **4.3.3. Fill Area—Tf_name**) for filling outlines. The name can be entered into the input text window either with a keyboard or with a mouse copy and drop from the basic elements' **Fill Area (Tf) Panel**.

The outline attribute defines the integer values that determine the area to fill (up to 10 values).

Figure 20. Graphics Method and Outfill Editor Panels



Outfill

default

Gfo_default

project linear

xticlabels#1 lon30

xticlabels#2 lon30

xmtics#1

xmtics#2

yticlabels#1 lat20

yticlabels#2 lat20

ymtics#1

ymtics#2

datawc_x1 -180

datawc_y1 -90

datawc_x2 180

datawc_y2 90

Tf default

outline 1

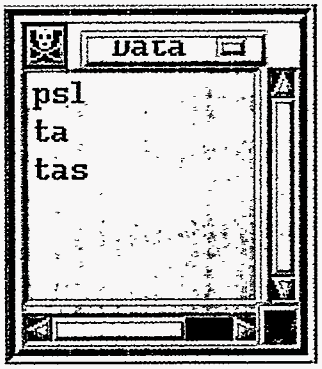
### *3.4.3. Data Panel*

Array data can be either selected from a file or can be computed as a function of one or more previously selected array data sets. See section **4.2.3. Array Data Attributes** for more details.

When the "Data" button is <u>click</u>ed in the **Control Panel**, the **Data Panel** appears below the **Page Description Panel**. See Figure 21. If the **Data Panel** is visible and the "Data" button is <u>click</u>ed in the **Control Panel**, then the **Data Panel** and its sub-panels disappear.

This panel contains: a **Terminate Button** (see section **3. INTERACTION**); a “Data” menu button; a scroll window that lists the attribute set names from the array data attributes table; vertical and horizontal scroll bars to view hidden portions of the list; and an **Extension Button** (see section **3. INTERACTION**).

Figure 21. Data Panel



The “Data” menu button brings up a menu of five functions: “Select,” “Copy,” “Rename,” “Assign,” and “Remove.” The uses of these menu options are described below:

- Choose “Select” to bring up the panels in Figure 22, and select data from a file. See section **3.4.3.1. Data Selection and Data Selection Browser Panels.**
- Mark an entry in the **Data Panel** and choose “Copy” from the “Data” menu to copy an attribute set to another named attribute set. A popup window will appear with an input text window, a “Save” button, and a “Cancel” button. Enter the new array data attribute set name, and click on the “Save” button to copy to a new array data attribute set. Click on the “Cancel” button to abort the “Copy” operation.
- Mark an entry in the **Data Panel**, and choose “Rename” from the “Data” menu to rename an attribute set. A popup window will appear with an input text window, showing the selected array data name, a “Save” button, and a “Cancel” button. Enter or modify the array data attribute set name, and click on the “Save” button to save the name of the array data attribute set. The name will not change if the attribute set is being used in an active picture. Click on the “Cancel” button to abort the “Rename” operation.
- Choose “Assign” from the “Data” menu to assign a variable to an array data attribute set as a function of previously defined variable(s). A popup window will appear with an input text window, a “Save” button, and a “Cancel” button. Enter the equation:

  *name=f(previously defined variables)*

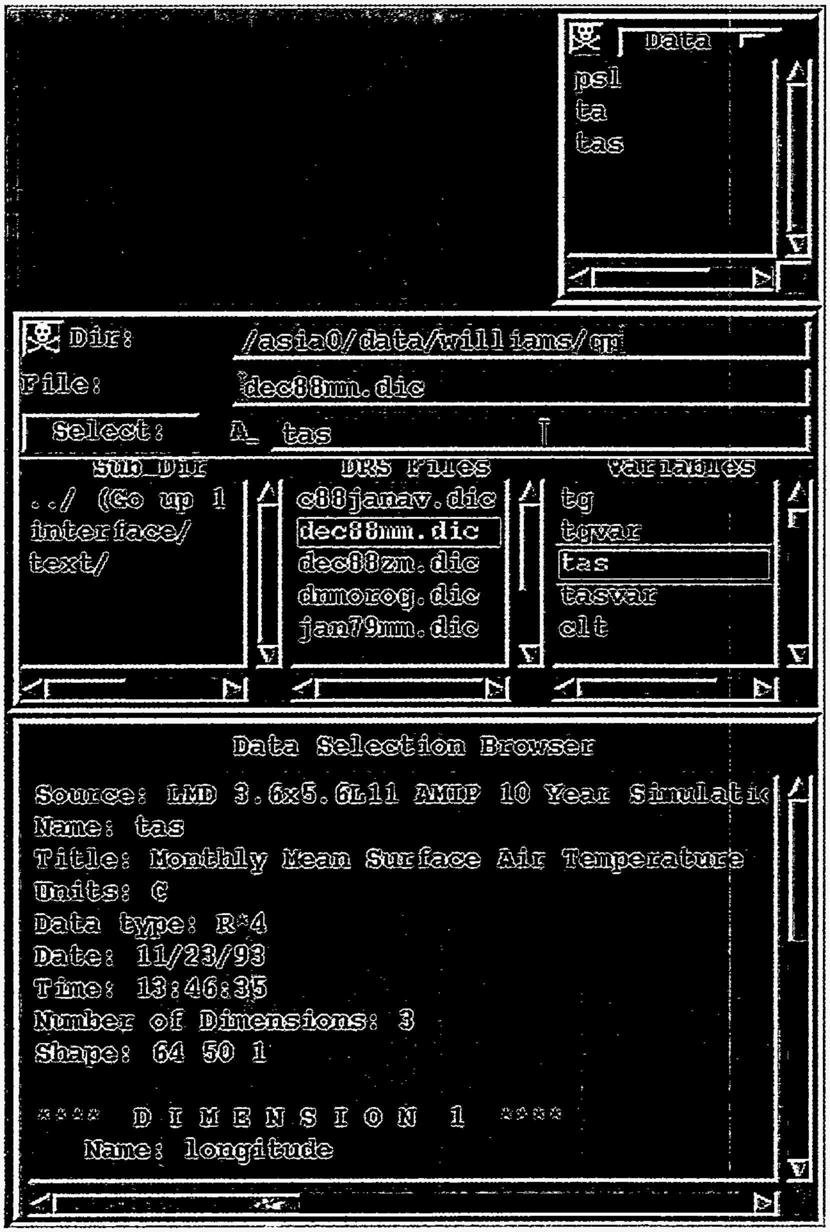
  and click on the “Save” button to assign an array data attribute set name that is a function of previously defined array data (or variables). See Figure 23.
- Mark an entry in the **Data Panel**, and choose “Remove” to remove an array data attribute set. The name cannot be removed if the attribute set is being used in an active picture.

The user can also press and release the right mouse button to immediately remove a single array data attribute set.

### *3.4.3.1. Data Selection and Data Selection Browser Panels*

Unlike the other primary elements, there are no default array data attribute sets; therefore, unless a script file (that defines at least one) is processed, it will be necessary to select data. The panels for selecting data appear only after choosing the "Select" option in the "Data" menu as described in section **3.4.3. Data Panel**. See Figure 22.

Figure 22. Data, Data Selection, and Data Selection Browser Panels



psl
ta
tas

Dir: /asia0/data/williams/cp

File: dec88mm.dic

Select: A_ tas

| Sub_Dir | DRS Files | Variables |
|---|---|---|
| ../ (Go up 1 | c88janav.dic | tg |
| interface/ | dec88mm.dic | tgvar |
| text/ | dec88zm.dic | tas |
| | drnorog.dic | tasvar |
| | jan79mm.dic | clt |

Data Selection Browser

```
Source: LMD 3.6x5.6L11 AMIP 10 Year Simulatic
Name: tas
Title: Monthly Mean Surface Air Temperature
Units: C
Data type: R*4
Date: 11/23/93
Time: 13:46:35
Number of Dimensions: 3
Shape: 64 50 1

****  D I M E N S I O N  1  ****
   Name: longitude
```

The **Data Selection Panel** contains the following buttons and windows:

**Terminate Button** See section **3. INTERACTION.**

Dir Input text window, showing the current search directory, which can be edited to change the current search directory.

File Input text window, showing the current search file, which can be edited to change the file name (it must be one that exists).

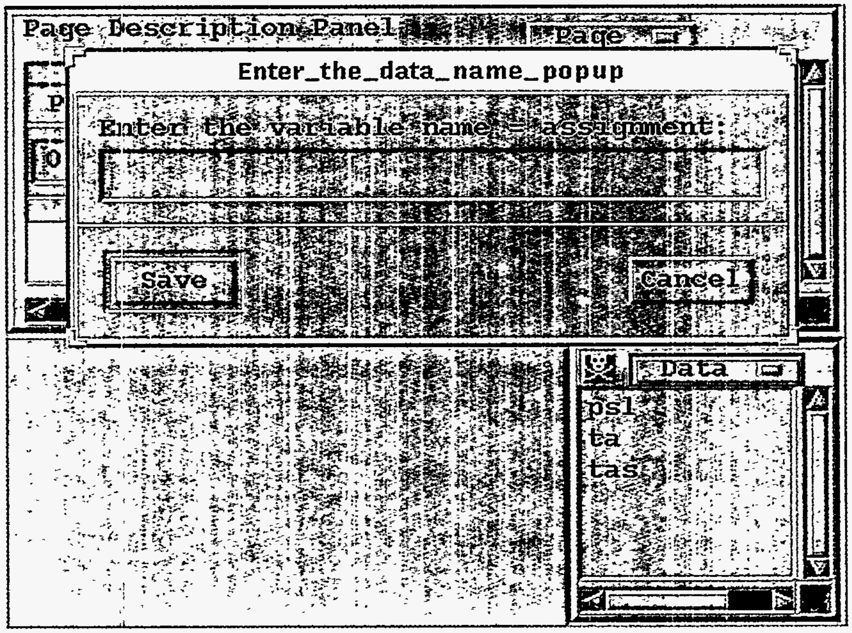
Select Button and an input text window that shows the name of the variable that is currently marked in the "variable" scroll window. The name can be edited prior to clicking on the "Select" button to enter it into the table of array data attribute sets and into the **Data Panel**.

Sub-Dir Scroll window that shows the current sub-directories of the search directory. Click on the name of the sub-directory to view sub-directories of that directory and its DRS dictionary files in the "DRS File" scroll window. Click on "../ (go up 1 directory level)" to move to the parent directory.

DRS File Scroll window that shows the DRS dictionary files located in the current directory. Click on one of these files to see the variables defined in the file.

Variable Scroll window that shows the DRS variables in the currently marked DRS dictionary file. Mark one of these variable names to select it for viewing in the **Data Selection Browser Panel**, and place it in the "Select" input text window.

Figure 23. Function Assignment Popup


Page Description Panel
Page
Enter_the_data_name_popup
Enter the variable name = assignment:
Save
Cancel
Data
psl
ta
tas

Double Click on a variable name in the “Variable” scroll window, and the variable will be placed in the **Data Panel’s** scroll window, provided it is a unique array data attribute set name.

The **Data Selection Browser Panel** is an information-only panel; it has a scroll window with horizontal and vertical scroll bars. To enable a more informed selection of variables, it provides a description of the marked variable in the “Variable” scroll window.

### *3.4.3.2. Data Editor Panel*

Mark one of the array data attribute set names in the **Data Panel**, and the **Data Editor Panel** will appear with its descriptors, as in Figure 24. Section **4.2.3. Array Data Attributes—A_name** describes these attributes in more detail.

The **Data Editor Panel** contains the following buttons and windows:

**Terminate Button** — See section **3. INTERACTION.**

Displayed/Defined — Pair of buttons to toggle between the “Displayed” and the “Defined” set of attributes. When one is on, the other is off.

Dim — Button to bring up the **Dimension Manipulation Panel**. See section **3.4.3.2.1. Dimension Manipulation Panel.**

Apply and a Reset — Button will appear on the top border if any change is made to the input text windows. The “Apply” button will apply changes that have been made. If a representation of the data exists on the **Graphics Canvas,** then that representation will change appropriately. The “Reset” button will reset any changes to their prior value if they have not been applied.

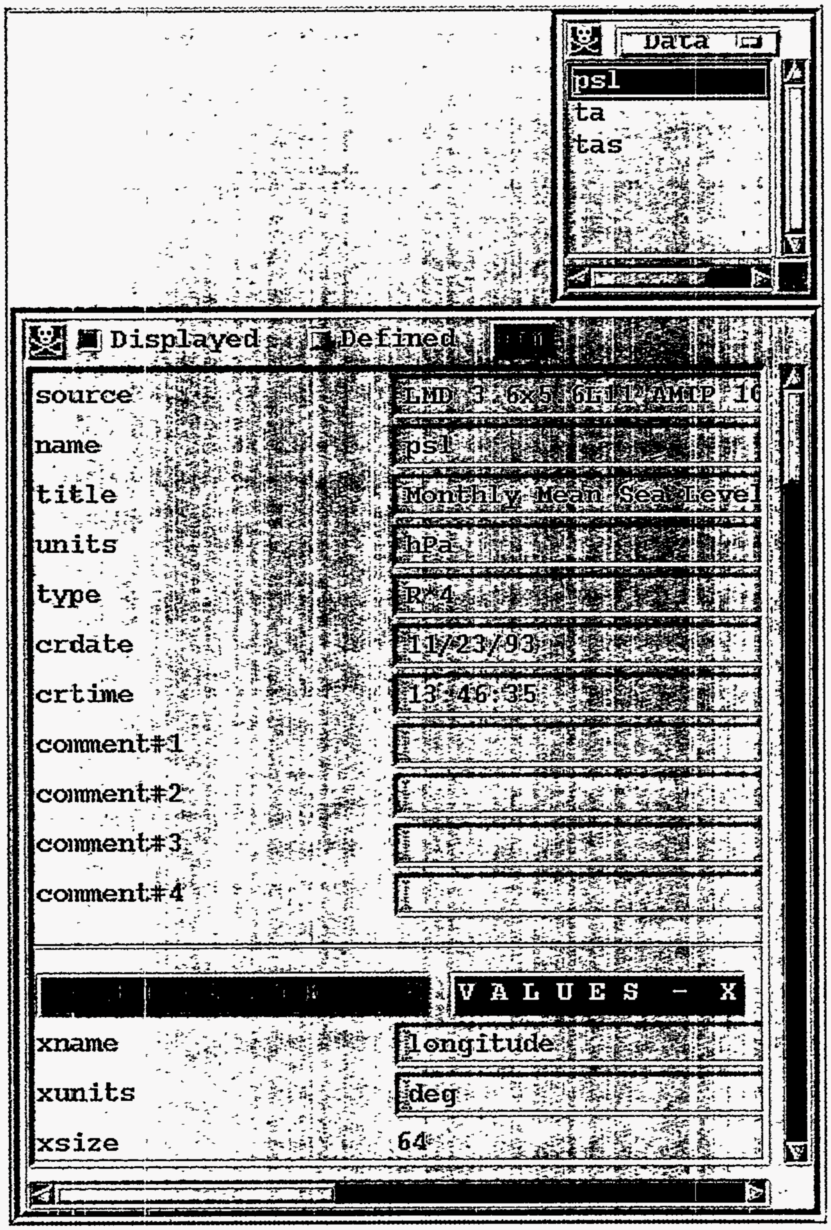
Scroll window — Contains named input text windows that can be edited to change the array data attributes. Horizontal and vertical scroll bars allow scrolling the window to view hidden text.

**DIMENSION** — A button for each dimension exists among the input text windows in the scroll window. Either of these buttons, when clicked, will bring up the **Dimension Manipulation Panel,** and the dimension values for the particular dimension will be visible. See section **3.4.3.2.1. Dimension Manipulation Panel.**

**VALUES** — A button for each dimension exists among the input text windows in the scroll window. Either of these buttons, when clicked, will bring up a **Dimension Assignment Panel**, with the particular dimension attributes shown. See section **3.4.3.2.2. Dimension Assignment Panel.**

Click on the “Defined” toggle button, and the scroll window will show the defined attributes from the array data attribute set, as in Figure 25. Click on the “Displayed” toggle button to return to the displayed attributes, as in Figure 24.

Figure 24. Data and Data Editor Panel (Displayed)

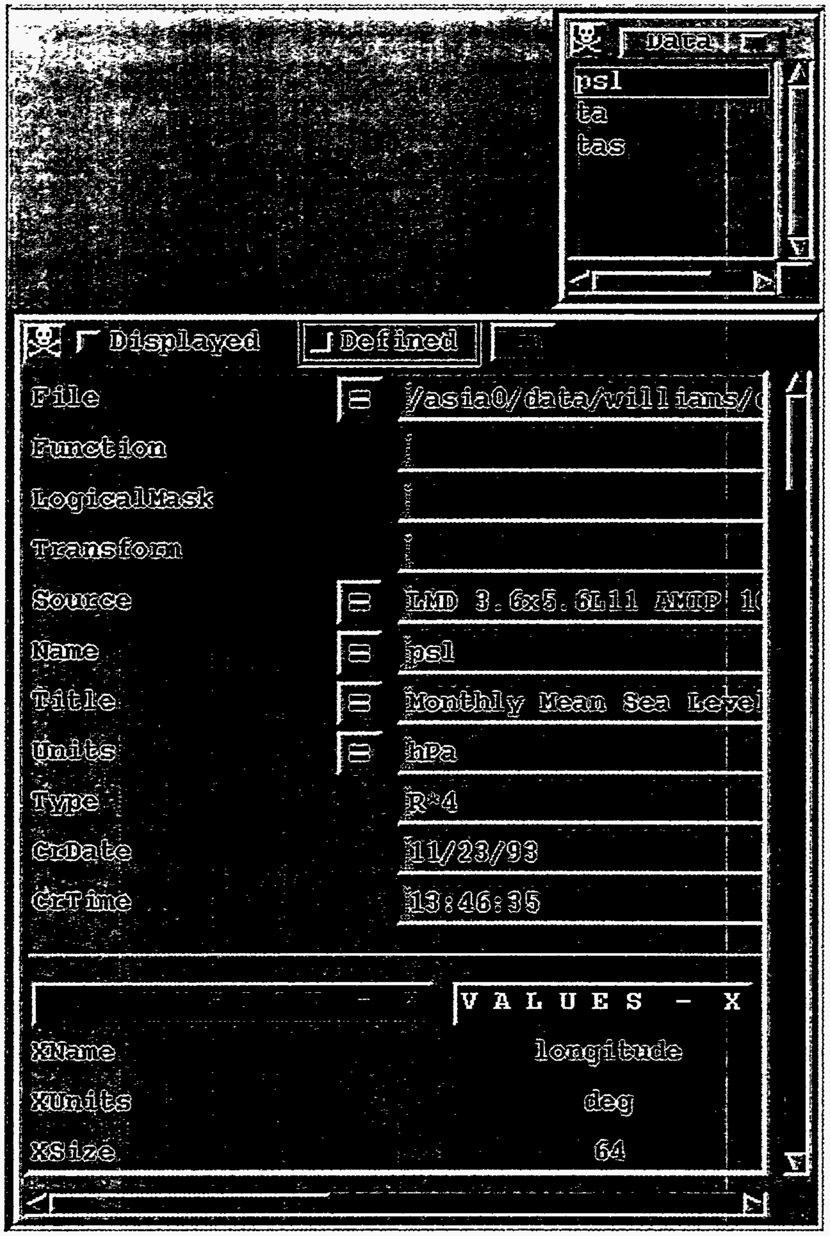


psl
ta
tas

| | |
|---|---|
| source | LMD 3.6x5 6L11 AMIP 1 |
| name | psl |
| title | Monthly Mean Sea Level |
| units | hPa |
| type | R*4 |
| crdate | 11/23/93 |
| crtime | 13:46:35 |
| comment#1 | |
| comment#2 | |
| comment#3 | |
| comment#4 | |
| xname | longitude |
| xunits | deg |
| xsize | 64 |

When a change is made to any of the attributes in the scroll window they become "assigned" attributes. A button with an equal sign (=) will appear prior to the input text window to indicate they are "assigned" attributes and will be retained for re-use of the attribute set. Re-use will occur each time changes are applied with the "Apply" button; all attribute values ("Displayed" and "Defined") are discarded and the "assigned" attribute values are re-inserted and used to redetermine the other array data attributes. To de-assign an attribute, click on its "=" button and the string, *****Apply to see New String*****, appears in the input text window. Click on the "Apply" button, and this string will be replaced with its unassigned value.

Figure 25. Data and Data Editor Panel (Defined)



psl
ta
tas

| | |
|---|---|
| File | /asia0/data/williams/( |
| Function | |
| LogicalMask | |
| Transform | |
| Source | LMD 3.6x5.6L11 AMIP 1( |
| Name | psl |
| Title | Monthly Mean Sea Level |
| Units | hPa |
| Type | R*4 |
| CrDate | 11/23/93 |
| CrTime | 13:46:35 |
| XName | longitude |
| XUnits | deg |
| XSize | 64 |

### 3.4.3.2.1. Dimension Manipulation Panel

Click on the "Dim" button at the top of the **Data Editor Panel** to bring up the **Dimension Manipulation Panel**. The user can also click on one of the "DIMENSION" buttons, in the scroll window to bring up the **Dimension Manipulation Panel**. See Figure 26.

The **Dimension Manipulation Panel** has a column for each dimension that is defined in the array data attribute set.

This panel contains the following:

**Terminate Button** See section **3. INTERACTION.**

Apply Button that will apply changes that have been made. If the array data is used to produce a picture on the **Graphics Canvas,** then that picture will change appropriately.

Help Button that is unattached in VCS.

Dimensions Buttons at the top of each column are labeled with dimension names (i.e., longitude, latitude, etc.). These buttons can be used to transpose dimensions: click on one of the buttons, and a scroll window will appear with a list of the dimension names and their positions (i.e., X -, Y -, Z -, and T -); click on one of these dimensions to transpose it with the present dimension. The user must click on one of the dimensions in this scroll window to escape from it; if it is the present dimension, no transposition will occur.

Reset Button that will, when clicked, reset the dimension values, shown in the scroll window, to their original values.

Reverse Button that will, when clicked, reverse the order of the dimension node values in the scroll window.

Cycle Input text window that allows the user to change or set the cycle value for wrapping a dimension around, for instance: the longitude dimension is cyclic (i.e., 0° and 360° are the same node point), and its cycle is 360°. The values are usually not duplicated in the data, and it is fairly standard practice, in plotting atmospheric data, to include the first node value also at the end.

Wrap Top/

Wrap Bottom Input text window with arrow buttons allows a dimension to be wrapped multiple times in either direction prior to selection of nodes. The values can be input by keyboard or incremented with the up-arrow buttons, or decremented with the down-arrow buttons. These values cannot be less than zero.

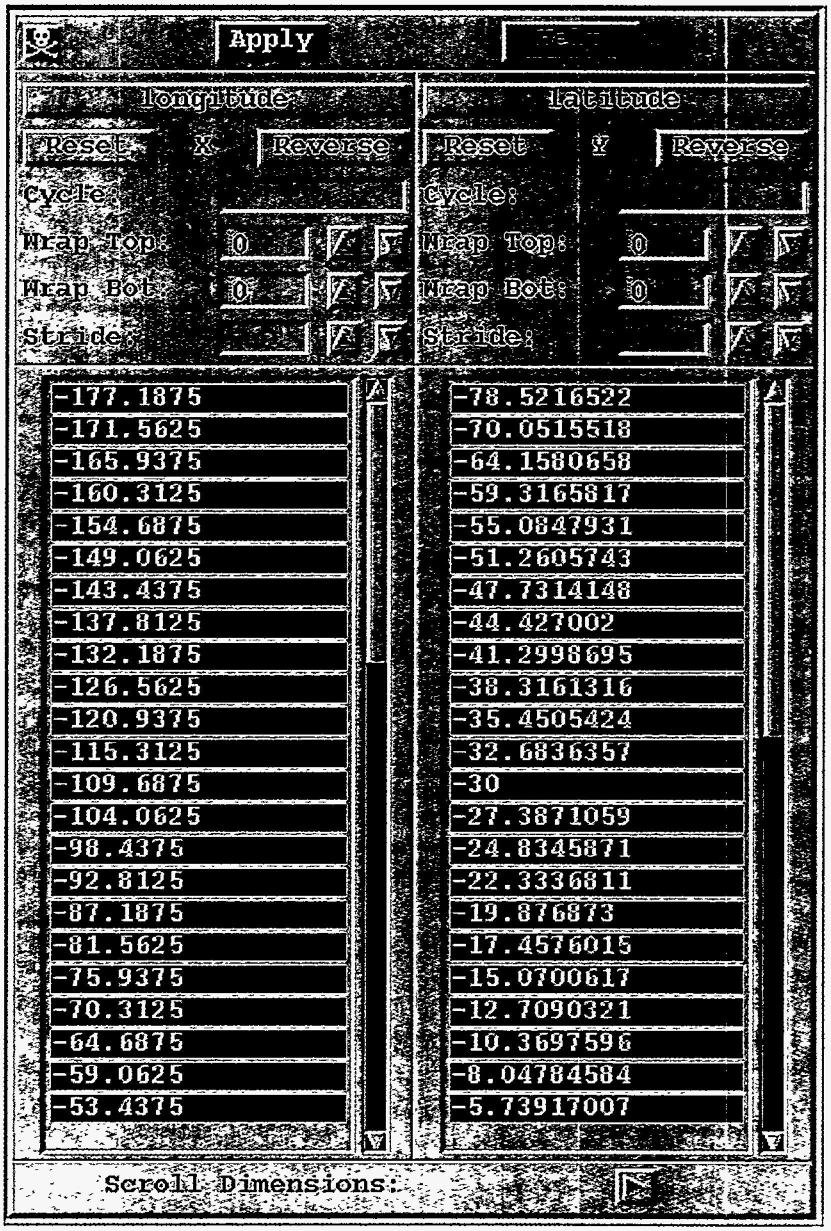
Stride Input text window and arrow buttons allow a stride to be set with keyboard entry, or incremented with the up-arrow button, or decremented with the down-arrow. The stride thins the data by taking every nth value (stride = n) beginning with the first selected value. The stride cannot be less than one.

Scroll window Window showing the dimension node values allows the user to select the desired dimension node values. This scroll window has its own mouse operations: the left mouse button selects the top dimension node value; the middle mouse button selects the bottom dimension node value; and the right mouse button toggles individual dimension node values on or off. A highlighted node value indicates that the value is selected.

Scroll Dimension

Arrow button, or opposing (left and right) buttons, will appear in the bottom border of the panel if there are more than two dimensions. The left arrow will disappear if the first dimension is already shown, and the right arrow will disappear if the last dimension is already shown. When an arrow button exists, click on it and the dimension columns will move over one in that direction. This scrolling dimension operation permits viewing dimensions that are not initially visible to the user. There can be only two dimension columns visible.

Figure 26. Dimension Manipulation Panel



```
Wrap Top: 0                Wrap Top: 0
Wrap Bot: 0                Wrap Bot: 0

-177.1875                  -78.5216522
-171.5625                  -70.0515518
-165.9375                  -64.1580658
-160.3125                  -59.3165817
-154.6875                  -55.0847931
-149.0625                  -51.2605743
-143.4375                  -47.7314148
-137.8125                  -44.427002
-132.1875                  -41.2998695
-126.5625                  -38.3161316
-120.9375                  -35.4505424
-115.3125                  -32.6836357
-109.6875                  -30
-104.0625                  -27.3871059
-98.4375                   -24.8345871
-92.8125                   -22.3336811
-87.1875                   -19.876873
-81.5625                   -17.4576015
-75.9375                   -15.0700617
-70.3125                   -12.7090321
-64.6875                   -10.3697596
-59.0625                   -8.04784584
-53.4375                   -5.73917007
```

#### 3.4.3.2.2. Dimension Assignment Panel

Click on one of the "VALUES" buttons in the **Data Editor Panel** scroll window to view that dimension's node values, bounds, and weights in the **Dimension Assignment Panel**. See Figure 27.

This panel contains the following:

**Terminate Button** See section **3. INTERACTION.**

Reset Button that will reset these attributes to the initial or last applied state.

Apply Button that will change values in the columns to match the new assignment.

Text windows Input text window for each column that shows the **List** name if one is used to assign the dimension attribute, or is blank, if not. Click in one of these input text windows to edit or enter a **List** name to replace the dimension attribute.

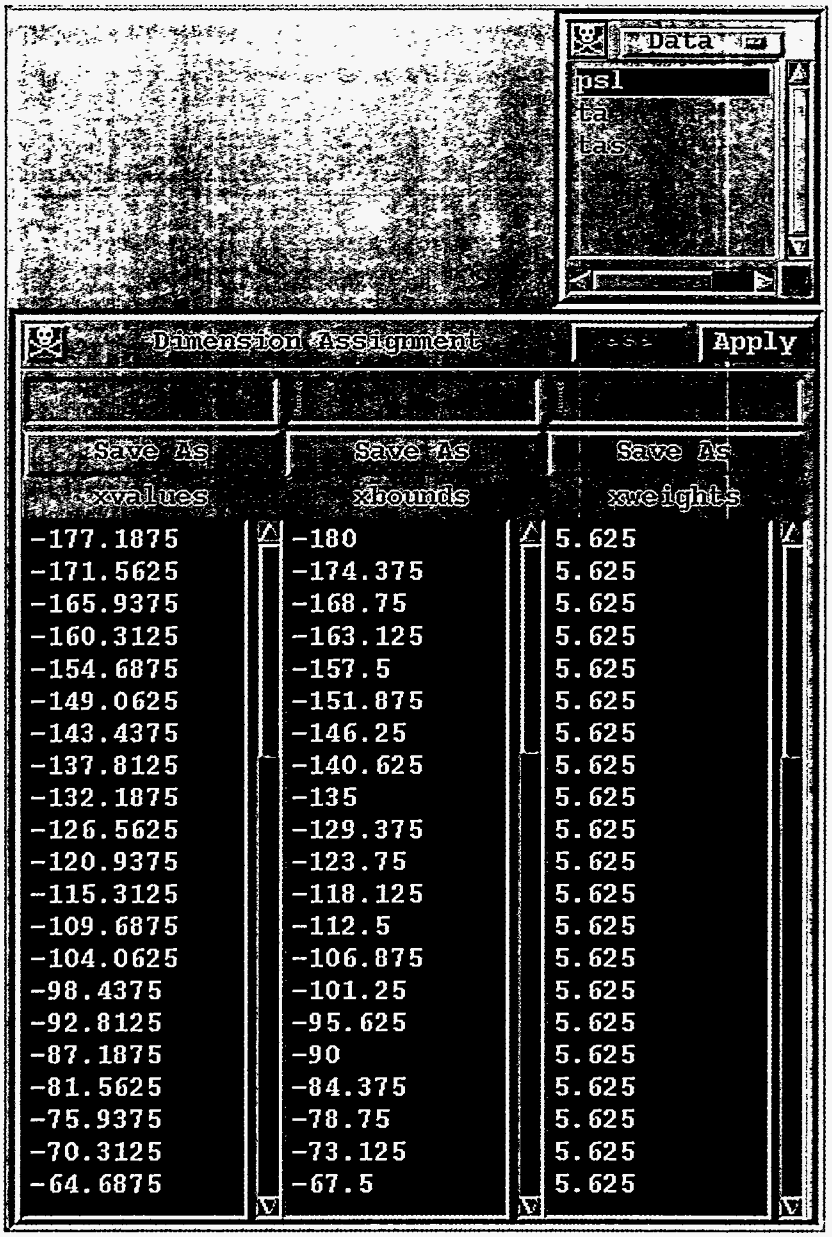
Save As Button for each column that saves a **List** containing the attribute vector. Click on the button to bring up a popup window with an input text window, a "Save" button, and a "Cancel" button. Enter a unique **List** name, and click on "Save" to create a list, or click on "Cancel" to cancel the operation.

Unless the attributes shown in this panel are assigned from a **List**, they are either computed or taken from the file in which the array data attributes are found. They can be assigned a **List** name that remains connected, such that: if the list changes, the attribute changes. They can be assigned from a list, without the connection being retained as: @listname.

The ability to assign the "Displayed" dimension attributes allows the assignment of requested nodes, which, if the nodes are not the "Defined" nodes, will define the target grid for a grid transformation.

The ability to save these attributes into lists allows the user to create **Lists** for use in assigning the nodes of other array data for transformation to that grid.

Figure 27. Data and Dimension Assignment Panels



| xvalues | xbounds | xweights |
|---|---|---|
| -177.1875 | -180 | 5.625 |
| -171.5625 | -174.375 | 5.625 |
| -165.9375 | -168.75 | 5.625 |
| -160.3125 | -163.125 | 5.625 |
| -154.6875 | -157.5 | 5.625 |
| -149.0625 | -151.875 | 5.625 |
| -143.4375 | -146.25 | 5.625 |
| -137.8125 | -140.625 | 5.625 |
| -132.1875 | -135 | 5.625 |
| -126.5625 | -129.375 | 5.625 |
| -120.9375 | -123.75 | 5.625 |
| -115.3125 | -118.125 | 5.625 |
| -109.6875 | -112.5 | 5.625 |
| -104.0625 | -106.875 | 5.625 |
| -98.4375 | -101.25 | 5.625 |
| -92.8125 | -95.625 | 5.625 |
| -87.1875 | -90 | 5.625 |
| -81.5625 | -84.375 | 5.625 |
| -75.9375 | -78.75 | 5.625 |
| -70.3125 | -73.125 | 5.625 |
| -64.6875 | -67.5 | 5.625 |

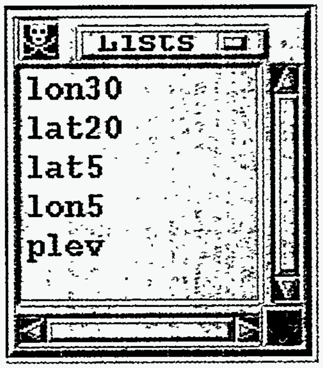
### *3.4.4. List Panel*

**Lists** consist of sequential pairs of (value, string) combinations. The **List** panel shows the **List** names. **List** is one of the basic elements; it is separated from the menu of basic elements in the **Control Panel** because it is expected to be more heavily used. See section **4.3.9 Lists—L_name** for more details on the **Lists.**

When the "Lists" button is clicked in the **Control Panel,** the **List Panel** appears below the **Page Description Panel.** If the **List Panel** is visible and the "List" button is clicked in the **Control Panel,** then the **List Panel** and its sub-panels disappear. See Figure 28.

This panel consists of: a **Terminate Button** (see section **3. INTERACTION**); a "Lists" menu button; a scroll window that lists the attribute set names from the picture template table; vertical and horizontal scroll bars to view hidden portions of the list; and an **Extension Button** (see section **3. INTERACTION**).

Figure 28. List Panel



lon30
lat20
lat5
lon5
plev

### *3.4.4.1. List Editor Panel*

The **List Editor Panel** shows the sequential (value, string) pairs in the **List**, and permits keyboard editing of each in their individual input text windows. See Figure 29. The panel contains the following:

**Terminate Button** See section **3. INTERACTION.**

Delete Button for deleting an entry (a row).

Scroll window Window with numbered lines containing first the value, then the string, in input text windows.

Scroll bars The vertical and horizontal scroll bars allow viewing of hidden portions of the **List.**

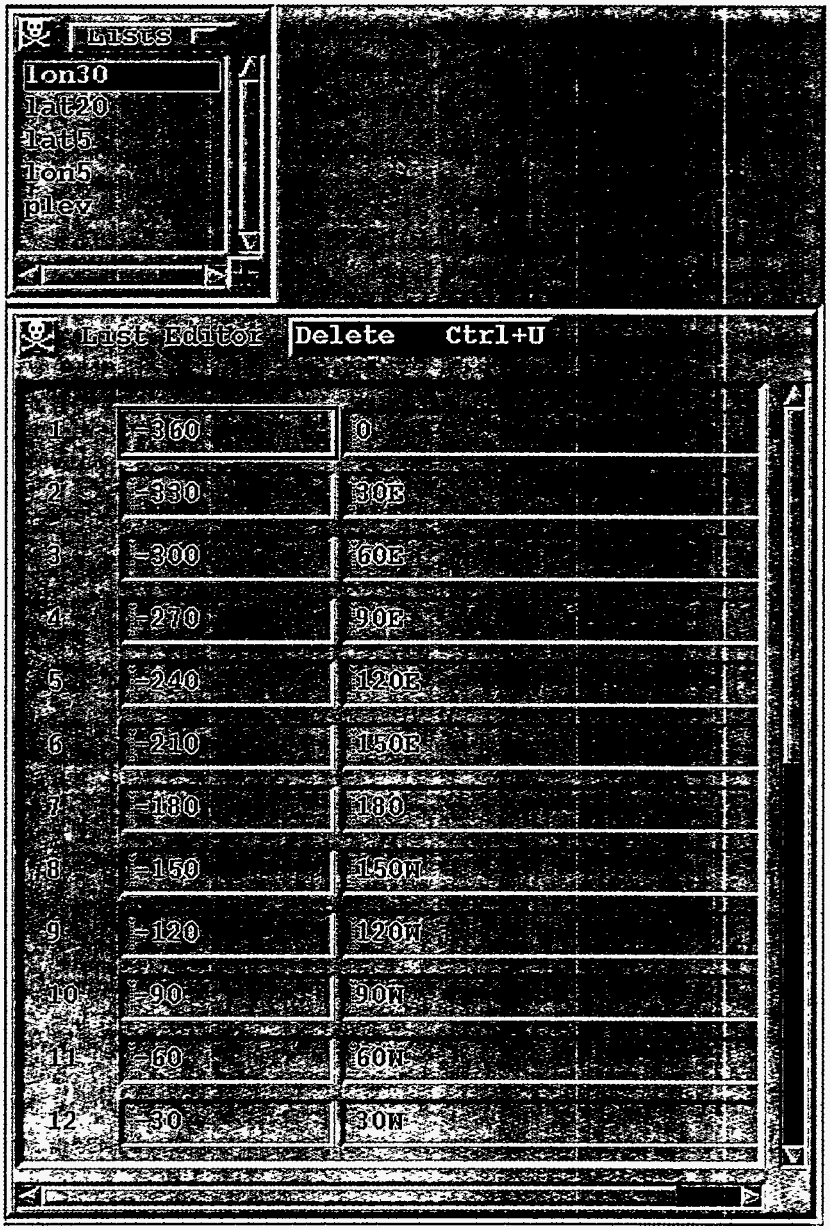
To remove a list entry, mark an input text window in the list and either click the "Delete" button, or use the keyboard "ctrl u." There is always one input text window marked.

The input value must be a number; the input string can be any printing characters.

Click into any input text window to modify the text. The arrow keys move the keyboard cursor from one input text window to another. The tab key moves the keyboard cursor from the value input text window, to the string input text window. The "Return" or "Enter" key will create a new list row.

Any modification of an input text window will cause the "Reset" and "Apply" buttons to appear in the upper right corner of the panel. It will be necessary to click on the "Apply" button prior to terminating this panel to save the list. The "Reset" button will reset all entries to their initial state or to the last applied state.

Figure 29. Lists and List Editor Panels



| | | |
|---|---|---|
| 1 | -360 | 0 |
| 2 | -330 | 30E |
| 3 | -300 | 60E |
| 4 | -270 | 90E |
| 5 | -240 | 120E |
| 6 | -210 | 150E |
| 7 | -180 | 180 |
| 8 | -150 | 150W |
| 9 | -120 | 120W |
| 10 | -90 | 90W |
| 11 | -60 | 60W |
| 12 | -30 | 30W |

### *3.4.5. Indices Panel*

Click on the **Indices** button in the **Control Panel** to bring up the **Indices Panel**. See Figure 30. The top border of the panel contains a **Terminate** button, a **Reset** button, and an **Apply** button. There are six lines, each of which contains an input text window for viewing and editing the value of one of the global indices (I, J, K, L, M, N), an up-arrow button, and a down-arrow button for incrementing and decrementing the index, and input text windows for defining a loop range (first and last) and an increment. A **Run Loop(s)** button, a **Postscript** button, and a **Raster** button are located below the six lines. See sections **4.4.6. Index**, section **4.4.7. Loop**, section **4.4.9. Postscript**, and section **4.4.10. Raster** for the script commands to perform these actions.

Indices can be assigned any positive integer value greater than 0. The index loop range values can also be assigned any positive integer value greater than 0. The loop increment can be left unassigned; by default, it will be assigned unity. It can be assigned any value, but if it is 0, it will be treated as a no-index loop. The sign of the increment value will be determined by the difference between the last and first range values.

The index input text windows have a red background color to match the red background color of the "Apply" button. The changes in these input text windows do not take affect until the "Apply" button is clicked.

The loop input text windows have a green background color to match the green background color of the "Run Loop(s)" button. These values are not used until the "Run Loop(s)" button is clicked.

The user can click on the toggle button for "Postscript" to produce a file of postscript representations of the graphics that exist for each pass through loop, or click on the toggle button for "Raster" to produce a file of the raster images that will exist for each pass through the loop, or both. If the user clicks on these toggle buttons, the **Postscript Panel** (See Figure 39) or the **Raster Panel** (See Figure 40) will cover the **Indices Panel**. Enter the output file name in the input text window, and click on the "Save" button. The panel will disappear from view (after clicking on the "Save" button), and the **Indices Panel** will again be visible. For more complete instructions on the use of these panels, see section **3.4.10. Output—Postscript and Raster Panels.**

When the "Run Loop(s)" button is pressed, the panel will step through each index according to its loop parameters. If the loop parameters for an index are blank, then the currently assigned index value will be retained. The left-most input text window will show the current index values as looping is taking place. At the completion of looping, index values will return to their assigned values.

Values of the indices can only be integer and greater than 0. An index is used as if the statement is written (in FORTRAN) as: L_list2[mod(I-1,n)+1], where n is the number of elements of the list. Therefore, all index values are valid for referring to any **List**. An indexed **List** name can only be used in **Array Data** character strings and single-valued reals, such as: File, Source, Name, strings, and the real values for xfirst and xlast.

If a range is not given for an index, its loop will not be executed and the index value will remain as assigned. If the "Postscript" button or the "Raster" button is not clicked, it/they will not be executed. The sign of the increment will be determined by the relative sizes of the first and last value regardless of any sign given by the user; if no increment is given, then 1 or –1 will be used. The range values must be greater than 0.

This interface and the concept of indices were designed specifically to aid in creating sequences of raster files to be used in an animation. For example, creating an animation of ten years of monthly means of a three-dimensional (lat, lon, month) variable named "psl," when each year's data exists in a different file.

Create a **List** with each of the file names in it, and a **List** with the month numbers in it, as:

> L_files (1, 'file79.dic', 2, 'file80.dic', 3, 'file81.dic', 4, 'file82.dic', 5, 'file83.dic', 6,' file84.dic', 7, 'file85.dic', 8, 'file86.dic', 9, 'file87.dic', 10, 'file88.dic')

> L_months (1, 'Jan', 2, 'Feb', 3 ,'Mar', 4 ,'Apr, 5 ,'May', 6 ,'Jun', 7 ,'Jul', 8 ,'Aug', 9 ,'Sep', 10 ,'Oct',11,'Nov',12,'Dec')

Define an array, as:

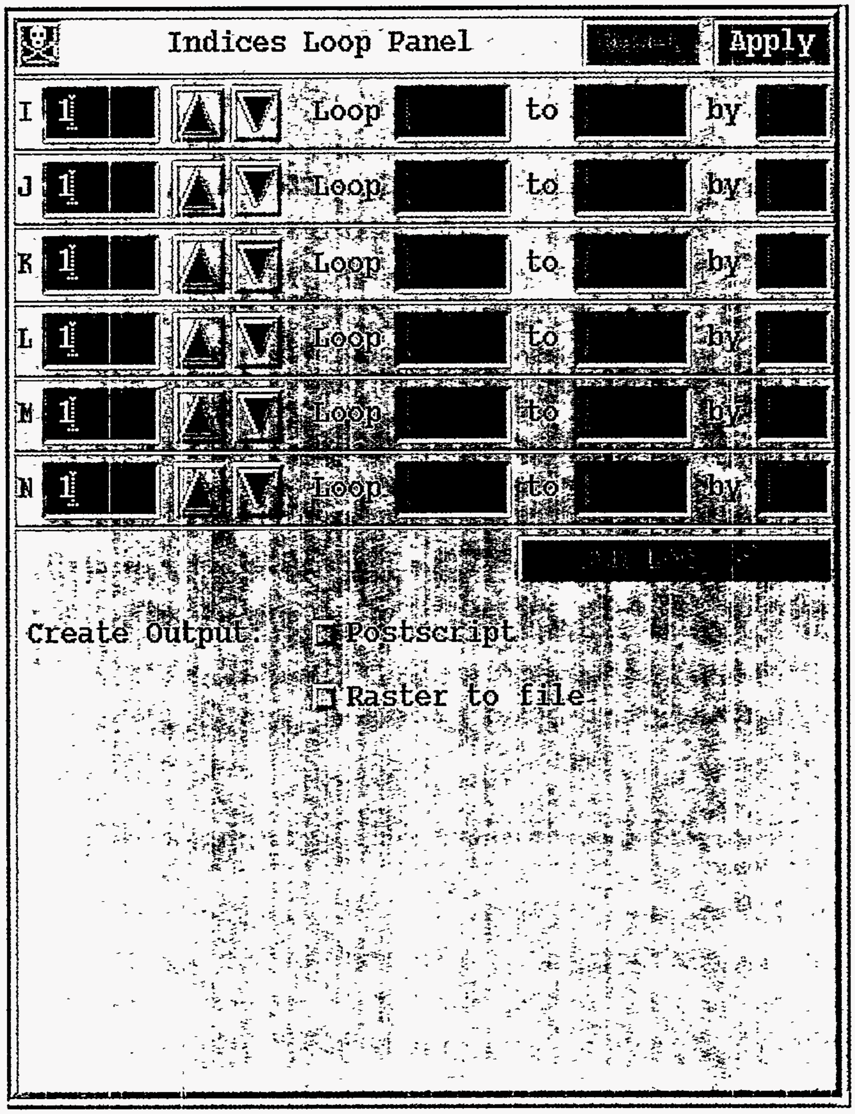
> A_psl (File=L_files[I], Name=psl, XName=lat, YName=lon, Zname=month, zfirst=L_months[J], zlast=L_months[J])

Set up the loops, as:

```
I   Loop from 1 to 10 by 1
J   Loop from 1 to 12 by 1
        Raster()
    End_all_loops
```

Click on the **Run Loop**(s) button.

Figure 30. Indices Panel



### *3.4.6. Colormap Panel*

There can only be one active colormap at any instant, and there is always an active colormap. See section **4.3.8. Colormaps—C_name**, for more details on the Colormap.

The **Colormap Panel** (see Figure 31) becomes visible after clicking the "Colormap" button in the **Control Panel**. If the **Colormap Panel** is visible and the "Colormap" button is clicked, then the **Colormap Panel** disappears. The **Colormap Panel** is located below the **Page Description Panel**. It is one of the basic elements but, since it will be heavily used, it has its own button.

At the top of the **Colormap Panel** is the "Colormap," which is a block of 16 × 16 colored boxes with the index for each printed in its box. Beside the "Colormap" is a vertical "Color Bar" with all these colors displayed in sequence.

Below the "Colormap" is the "Color Mixer" with either **RGB** (red, green, blue) or **CMY** (cyan, magenta, yellow) mixing sliders and a viewing color "Reset" button. Below the "Color Mixer" is a line of buttons: a "Blend" button, a "Color Model" button, a "Delete" button, and a "Save As" button.

A colormap table sub-panel at the bottom of the **Colormap Panel** contains the contents of the colormap table with the active colormap clearly marked.

Select a range of colors by clicking on a color box, and then pressing and releasing the center mouse button on another color box. Notice that the large color box in the "Color Model" is changed to the last selected color, and that the **RGB** or **CMY** colors are those used for the mixture.

Copy colors in the "Colormap" to another position by: selecting a range of colors, and then pressing and releasing the right mouse button on the position where the selection will be copied; the selected colors replace the preceding or following colors depending on whether the end of the selection is before or after the beginning of the selection. If a display in the **Graphics Canvas** uses the colors, then that display will change immediately.

Mix a new color for the last selected color by moving the sliders in the "Color Mixer." The large color box to the right of the sliders, the selected color box, and the display in the **Graphics Canvas**, if it uses the color, will change immediately as the color mixture changes.

When the "Color Mixer" is used, the "Reset" button displays: the modified color, the word "Reset," and the color index. Click on the "Reset" button to reset the color index to its original state.

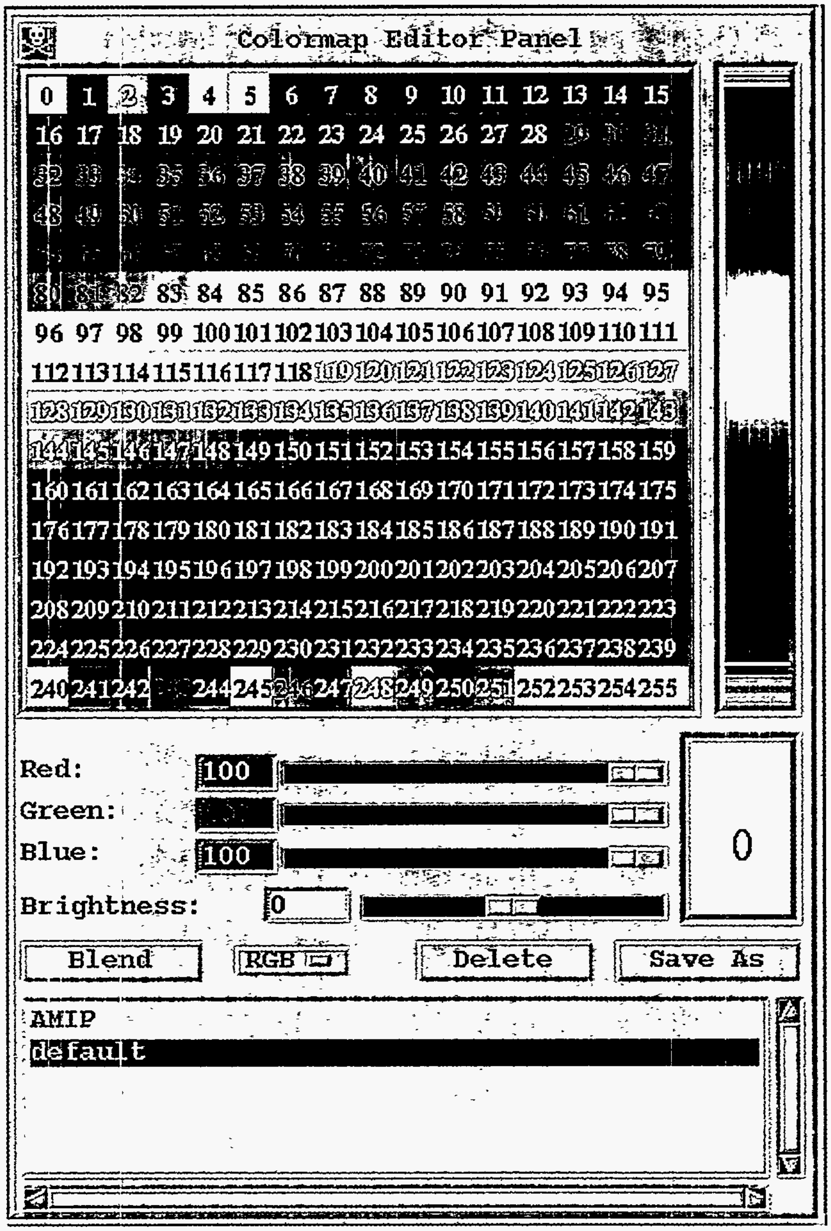
Change basic colors for mixtures by clicking on the "Color Model" button and by clicking either **RGB** (red, green, blue) or **CMY** (cyan, magenta, yellow) colors. The basic colors can be chosen rather than clicked.

Blend a group of colors by selecting a range of color indices and clicking on the "Blend" button. The colors within the range will be changed to an interpolated mixture of the colors at each end.

Store the modified active colormap as a named colormap table entry or replace the current named colormap table entry by clicking the "Save As" button and either entering a new name in the popup window and clicking the menu "Save" button, or by just clicking the "Save" button to replace the active colormap.

Select a different active colormap by marking a colormap name in the colormap table. Selecting the colormap that is already marked will wipe out any changes that may have been made.

Figure 31. Colormap Panel



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| 240 | 241 | 242 | [illegible] | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

Red: 100

Green: [illegible]

Blue: 100

Brightness: 0

0

AMIP

default

### *3.4.7. Basics*

A table of named attribute sets is defined for each of the basic elements, and at least one set of attributes is pre-defined and entered in the table for each. The manipulation of an attribute set that is being used by a primary element, which is, in turn, being used to create a picture, may produce an immediate, visible response.

The basic elements are available from the **Control Panel** by clicking on the "Basics" button, which brings up a menu of buttons for selecting the elements below.

#### *3.4.7.1. Text (Tt), Orientation (To), and Editor Panels*

Choose the "Text" button from the menu of **Basics** buttons, and two panels will appear just below the **Page Description Panel:** one with the text table **(Tt)** listed in it, and one with the text orientation table **(To)** listed in it.

Click on any name in either table and a **Text Edit Panel** and **Text Orientation Edit Panel** will appear, in that order, below the tables, and a **Text Monitor Panel** with an editable text string will appear below these as shown in Figure 32. The text, as described in the edit panels, will be displayed in the **Text Monitor Panel,** and that display will respond to each change in attributes.

These attributes are defined in section **4.3.1 Text Attributes—Tt_name** and in **4.3.2. Text Orientation Attributes—To_name. Tt_name** and **To_name** describe all the attributes required to display text. **Tt_name** and **To_name** are selected from the **Basics** button menu in the **Control Panel** by choosing or clicking the **Text** button.

A **Reset Both** button and an **Apply Both** button appear in a top border for these panels. **Apply Both** will store the changes for both text and orientation in the attribute sets. If they are used in an active display, it will be updated. **Reset** will reset to the initial values or the last applied values.

Click the **Font** button (it has a font name on it), and a menu of fonts will appear; click on one of these to change to that font. Choose can also be used to perform this selection.

Click the **Precision** button (it has a precision name on it), and the list of precisions will appear; click on any of these to change to that precision. Choose can also be used to perform this selection.

The spacing line defines additional spacing between characters in normalized device coordinates (0 to 1), and it has an editable value and a slider bar for changing values. The value is in thousandths, and its range is –50 to 50.

The expansion line defines a character width expansion factor, and it has an editable value and a slider bar for changing values. The value is in hundredths, and its range is 50 to 150.

The color index line defines the index for selecting the text color from the currently active color table. The value range is 0 to 255.

The height line defines the height of the characters in normalized device coordinates (0 to 1), and it has an editable value and a slider bar for changing values. The value is in thousandths, and its range is 1 to 100.

The angle line defines the rotation (positive clockwise) from the horizontal, and it has an editable value and a slider bar for changing values. The values are in degrees, and its range is 0 to 360.

The last line has a path button with a menu for selecting the direction the text will be displayed (up, down, left, right), a horizontal alignment button with a menu for selecting the horizontal reference position with respect to the text string (left, center, right), and a vertical alignment button with a menu for selecting the vertical reference position with respect to the text string (top, cap, half, base, bottom).

#### *3.4.7.2. Lines (Tl) and Editor Panels*

Choose the **Lines** button from the menu of **Basics** buttons, and a **Lines Panel** will appear just below the **Page Description Panel**, with the table of lines attribute sets listed in it.

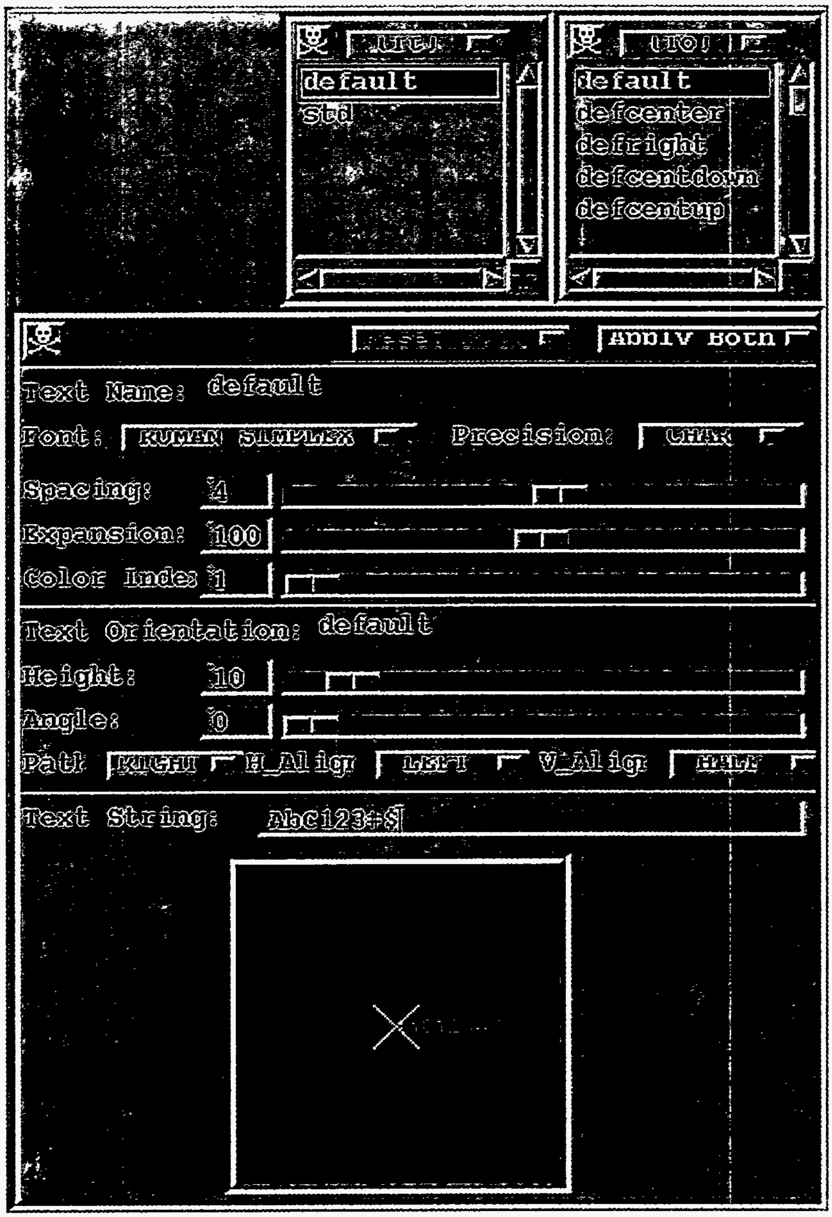
Click on any name in the **Lines Panel**, and a **Lines Editor Panel** will appear below the tables. A **Line Monitor Panel** will appear below these panels as shown in Figure 33. A line is displayed in the **Line Monitor Panel,** and that display will respond to each change in attributes.

These attributes are defined in section **4.3.5. Line Attributes—Tl_name.**

A **Reset** button and an **Apply** button appear in a top border for these panels. **Apply** will store the changes for these in the attribute sets; if they are used in an active display, it will be updated. **Reset** will reset to the initial values or the last applied values.

Click the **Line Type** button (it has a line type on it) and a menu of line types (Solid, Dashed, Dotted, Dashed-Dotted) will appear; click on one of these to change to the line type. Choose can also be used to perform this selection.

Figure 32. Text (Tt), Text Orientation (To), and Editor Panels



Text Name: default

Font: ROMAN SIMPLEX Precision: CHAR

Spacing: 4

Expansion: 100

Color Index: 1

Text Orientation: default

Height: 10

Angle: 0

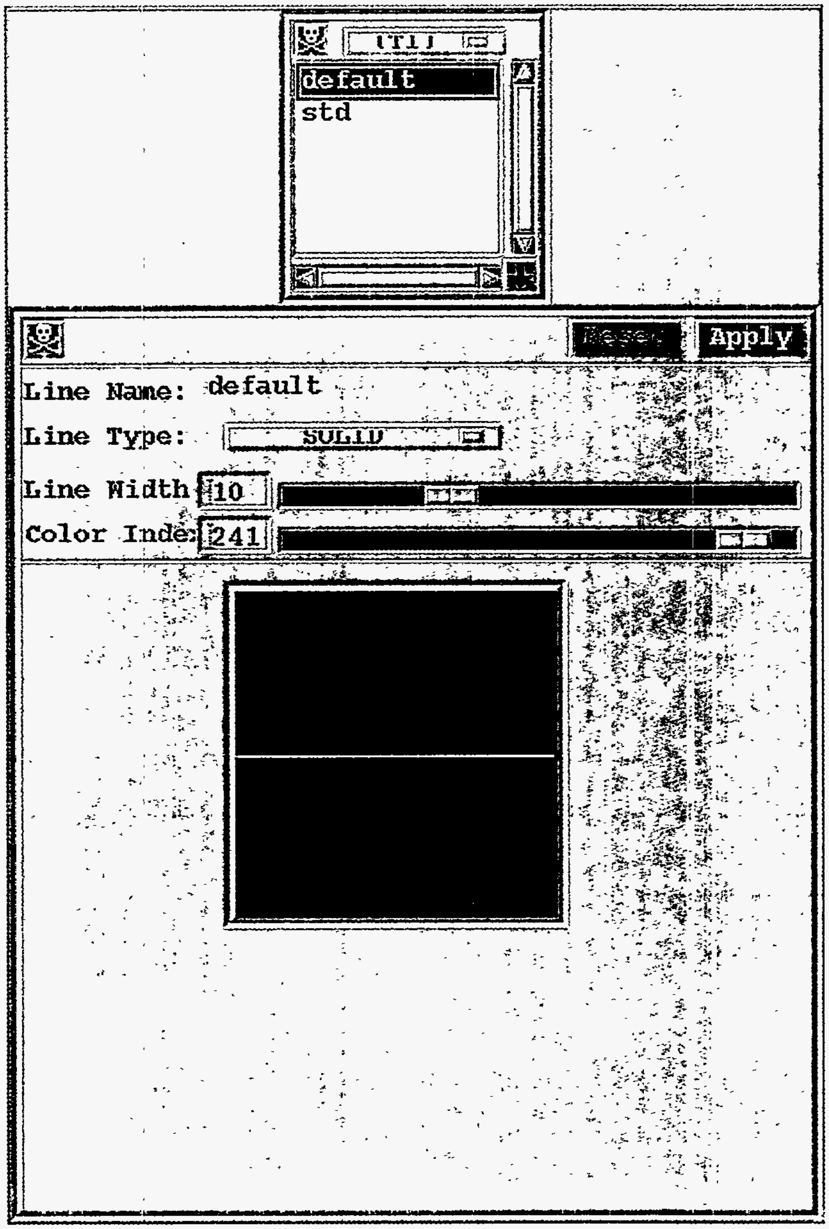
Path: RIGHT H_Align: LEFT V_Align: HALF

Text String: AbC123+$

The line width defines a line-width scale factor, and it has an editable value and a slider bar for changing values. The value is in hundredths, and its range is 1 to 100.

The color index line defines the index for selecting the color, from the currently active color table, that will be used for the lines. The value range is 0 to 255.

Figure 33. Lines (Tl) Panel and Editor Panel



Line Name: default

Line Type: SOLID

Line Width 10

Color Inde: 241

*3.4.7.3. Fill Area (Tf) and Editor Panel*

Choose the **Fill** button from the menu of **Basics** buttons, and a **Fill Panel** will appear centered just below the **Page Description Panel** with the table of fill area attribute sets listed in it.

Click on any name in the table, and a **Fill Area Editor Panel** will appear below the tables. A **Fill Area Monitor Panel** will appear below these panels as shown in Figure 34. A pattern fill area is displayed in the **Fill Area Monitor Panel,** and that display will respond to each change in attributes.

These attributes are defined in section **4.3.3. Fill Area—Tf_name.**

A **Reset** button and an **Apply** button appear in a top border for these panels. **Apply** will store the changes for fill area in the attribute sets. If they are used in an active display, it will be updated. **Reset** will reset to the initial values or the last applied values.

Click the **Interior Style** button (it has a fill area interior style on it), and a menu of fill area interior styles (Solid, Hollow, Pattern, Hatch) will appear; click on one of these to change to that interior style. Choose can also be used to perform this selection.

The style index line defines a style index, and it has an editable value and a slider bar for changing values. It is available only for interior styles: Pattern and Hatch.

The color index line defines the index for selecting the color, from the currently active color table, that will be used for the lines. The value range is 0 to 255.

The X and Y reference point lines define the point of reference, for patterns only, with values in thousandths. The point is in normalized device coordinates 0 to 1.

The Pattern Width and Height lines define the pattern size in hundredths. The size is measured in normalized device coordinates 0 to 1.

### *3.4.7.4. Marker (Tm) Panel*

Choose the **Marker** button from the menu of **Basics** buttons, and a **Marker Panel** will appear centered just below the **Page Description Panel** with the table of marker attribute sets listed in it. See Figure 35.

### *3.4.7.5. Format (Th) Panel*

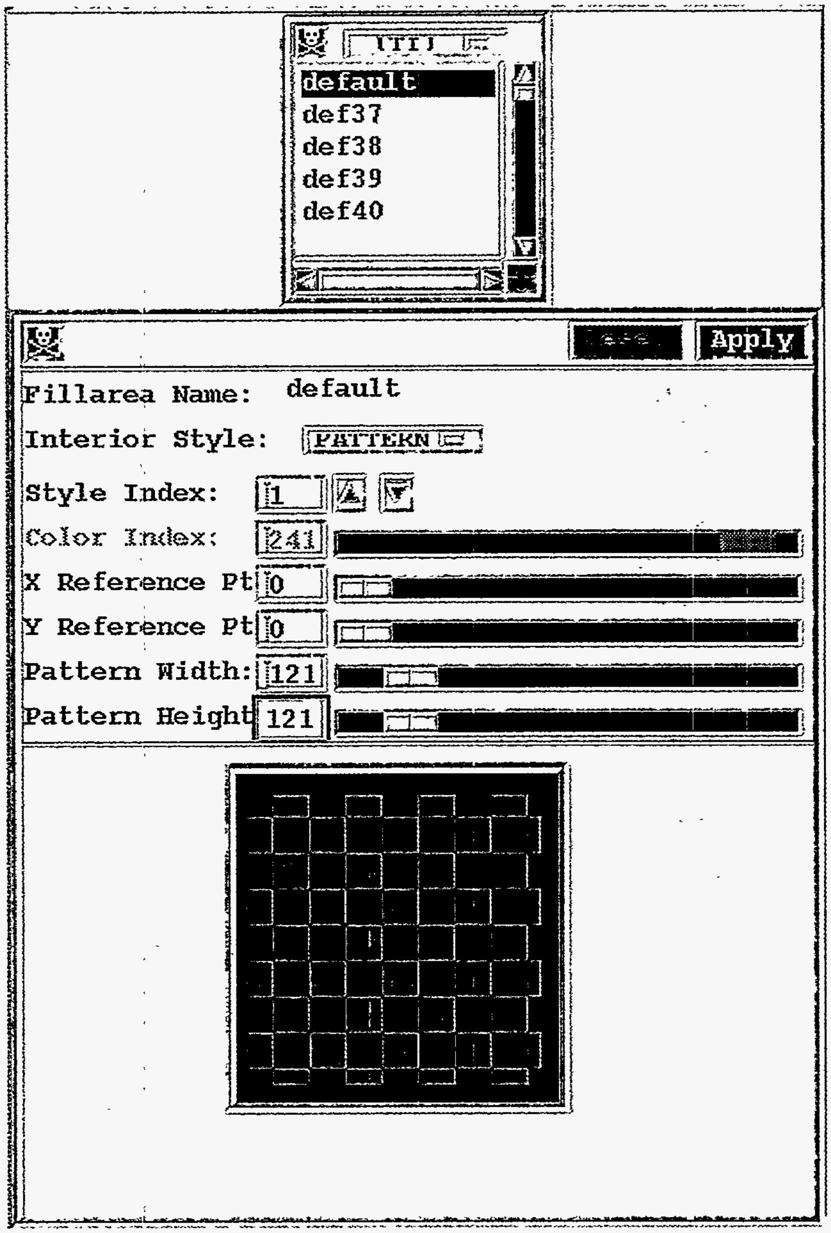
Choose the **Format** button from the menu of **Basics** buttons, and a **Format Panel** will appear centered just below the **Page Description Panel** with the table of format attribute sets listed in it. See Figure 36.

VCS does not have a Format Edit Panel.

### *3.4.7.6. Pattern (Tp) Panel*

VCS does not have an interface panel.

Figure 34. Fill Area (Tf) Panel and Editor Panel



Fillarea Name: default

Interior Style: PATTERN

Style Index: 1

Color Index: 241

X Reference Pt 0

Y Reference Pt 0

Pattern Width: 121

Pattern Height 121

Figure 35. Marker (Tm) Panel

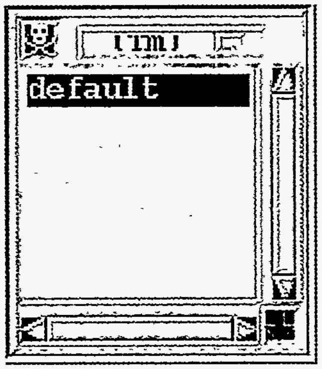
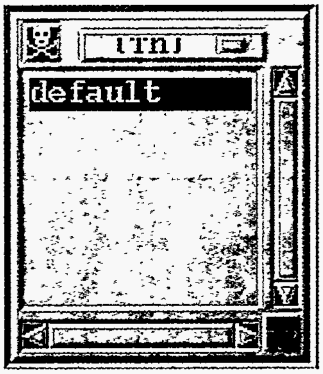


Figure 36. Format (Th) Panel



### *3.4.8. Script Panel*

Click on the **Script** button in the **Control Panel** to bring up a panel with a **Terminate** button a "Run" button, and a "Dump" button in its top border. It will have a **Sub-Directory** sub-panel and a **Files** sub-panel, and below those sub-panels is a **Directory** line that displays the current directory name, and a **File** line that displays the chosen file. See Figure 37.

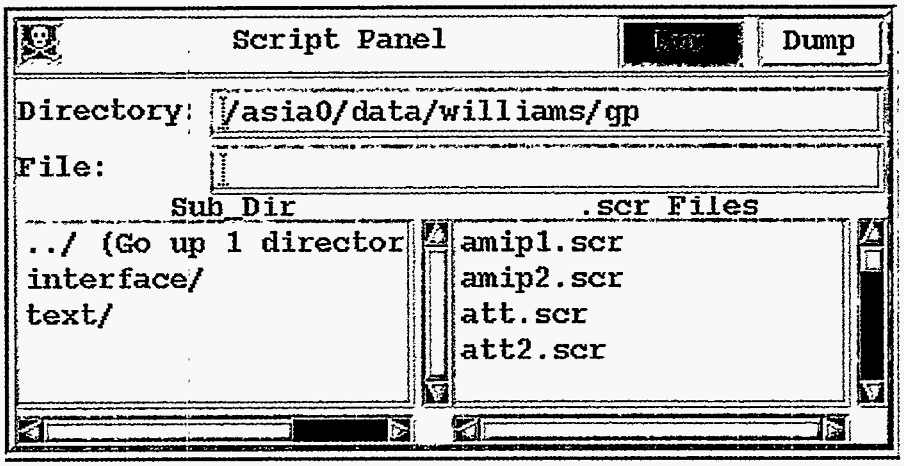
To traverse the directories, click on the "../" directory to move backward up the tree, or click on a directory name to move down the directory tree. The desired directory can also be typed into the directory display line. All the files within the directory that end with a ".scr" extension will be listed in the **Files** sub-panel. When the directory is satisfactory, then mark the desired file. A file name can also be typed directly into the **File** line.

Once a suitable file has been found, click on the "Run" button to run the script file, or click on the "Dump" button to dump the state-of-the-system in the indicated file. If the file exists, it will be overwritten when a dump is performed.

### *3.4.9. Animation Panel*

Click on the **Animation** button in the **Control Panel** to bring up the **Animation Panel** with a **Sub-Dir** sub-panel and a **.ras Files** sub-panel. Above those panels is a **Directory** line that displays the full name of the directory currently being displayed in the **Sub-Dir** sub-panel. Immediately below the sub-panels is: a **Load Images from Disk to Memory** button; and below that a **Read Images from** line with a **Memory** button and a **Disk** button; and below that a **Use Colormap from** line with an **Application** button and an **Image** button; and below that are a **Run Animation** button and a **Stop Animation** button; and below that an **Animation Position** line with a text window that displays the index of the image and a slider bar

Figure 37. Script Panel



Directory: /asia0/data/williams/gp

File:

Sub_Dir

```
../ (Go up 1 director
interface/
text/
```

.scr Files

```
amip1.scr
amip2.scr
att.scr
att2.scr
```

that displays the position of the image in the set of images being animated; and below that an **Animation Delay** line with a text window that displays the delay between frames (in milliseconds) and a slider bar for ease in adjusting the delay; and below that an **Animation Speed** line where the speed of the animation is displayed in frames per second. See Figure 38.

To traverse the directories, <u>click</u> on the "../" directory to move backward up the tree or <u>click</u> on a directory name to move down the directory tree. The desired directory can also be typed into the **Directory** line. All the files within the directory that end with a ".ras" extension will be listed in the **Files** sub-panel.

When the directory is satisfactory, then <u>click</u> on the files in the order they are to be used in an animation. As each file name is <u>click</u>ed, a number will be displayed before the file name to indicate the order.

<u>Click</u> on the **Load Images from Disk to Memory** button if the user wishes a faster animation; this will automatically select the **Memory** button in the next line.

The **Read Images from** line allows the user to select whether the images will be replayed from disk (<u>click</u> on the **Disk** button) or from memory (<u>click</u> on the **Memory** button). These are toggle buttons, when one is <u>click</u>ed on, the other turns off; however, the **Memory** button cannot be <u>click</u>ed on unless images exist in memory (i.e., **Load Images from Disk to Memory** has been <u>click</u>ed).

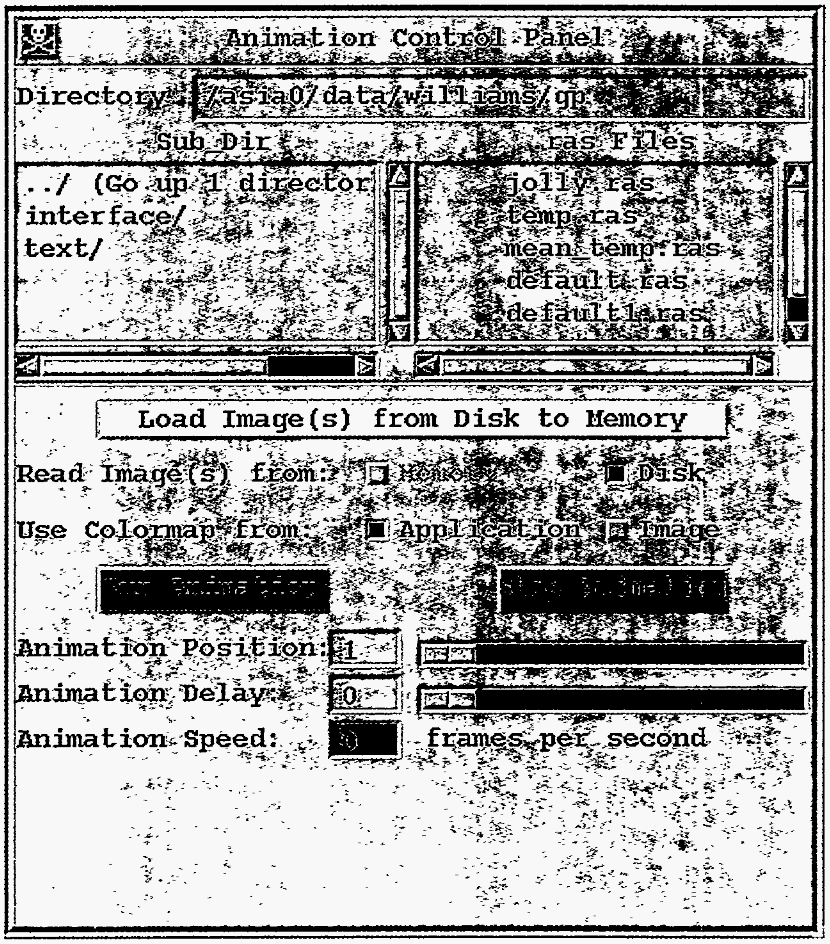
The **Use Colormap from** line allows the user to choose whether the colormap or colormaps that have been saved with the raster images will be used to display the images, or whether the applications' active colormap will be used. The latter option allows the user to manipulate the active colormap during animation and to see the results immediately.

The **Animation Position** line allows the animation position to be manipulated when the animation is in a stopped state. The slot can be edited after clicking in it, or the user can point to the slider and depress the left mouse button to move the slider at will within its bar.

The **Animation Delay** line allows the user to set the delay between frames (in milliseconds) in an animation, in either the stopped or the running state. The slot can be edited after clicking in it, or the user can point to the slider and depress the left mouse button to move the slider at will within its bar.

The **Animation Speed** line displays the speed in frames per second.

Figure 38. Animation Panels



Animation Control Panel

Directory: /asia0/data/williams/gp

Sub_Dir

../ (Go up 1 director
interface/
text/

.ras Files

jolly.ras
temp.ras
mean_temp.ras
default.ras
default1.ras

Load Image(s) from Disk to Memory

Read Image(s) from: □ Memory ■ Disk

Use Colormap from: ■ Application □ Image

Animation Position: 1

Animation Delay: 0

Animation Speed: 0 frames per second

### *3.4.10. Output—Postscript and Raster Panels*

Two output options, postscript and raster image, are available on the menu that pops up after clicking the **Output** button in the **Control Panel**. The menu includes a **Postscript** button and a **Raster** button.

Click on the **Postscript** button to bring up the **Postscript Panel** with a **Sub-Directory** sub-panel and a **.ps Files** sub-panel. See Figure 39. Above those sub-panels is a **Directory** line that displays the previously chosen (or current) directory name and a **File** line that displays either the previously entered file name or no file name.

The first time this panel is entered, there will be no file name listed on the **File** line. Once a previously nonexistent file name, to which the user has write privileges, is entered it will remain in the **File** line and will remain open for use until another file name is entered or the program is terminated. A “.ps” extension will be appended to the entered file name.

To traverse the directories, click on the “../” directory to move backward up the tree, or click on a directory name to move down the directory tree. The desired directory can also be typed into the directory display line. All the files within the directory that end with a “.ps” extension will be listed in the **.ps Files** sub-panel.

When the directory is satisfactory, type a file name into the **File** line and then click on the **Save** button in the top border of the panel to save an Adobe PostScript copy of the **Graphics Canvas** into the file. **Note**: the user *must* have write privileges for the directory, or *no* file will be opened and *no* postscript will be written.

Once a writable file for postscript output has been selected and exists in the **File** line, it is only necessary to click on the **Save** button in the top border of the panel to save the postscript representation of the pictures that are displayed on the **Graphics Canvas** into the open file.

Click on the **Raster** button to bring up the **Raster Panel** with a **Sub-Directory** sub-panel and a **.ras Files** sub-panel. See Figure 40. Above those sub-panels is a **Directory** line that displays the previously chosen (or current) directory name, and a **File** line that displays either the previously chosen or entered file name or no file name.

The first time that this panel is entered, no file name will be listed on the **File** line. Once a file name, to which the user has write privileges, is chosen or entered it will remain in the **File** line and available for use until another file name is either chosen (from the .ras Files sub-panel), or entered (on the **File** line), or the program is terminated. A “.ras” extension will be appended to the entered file name. **Note**: unlike postscript files, a previously existing file can be chosen or entered, and raster dumps will be appended to the end of the file.

To traverse the directories, click on the "../" directory to move backward up the tree, or click on a directory name to move down the directory tree. The desired directory can also be typed into the directory display line. All the files within the directory that end with a ".ras" extension will be listed in the **.ras Files** sub-panel.

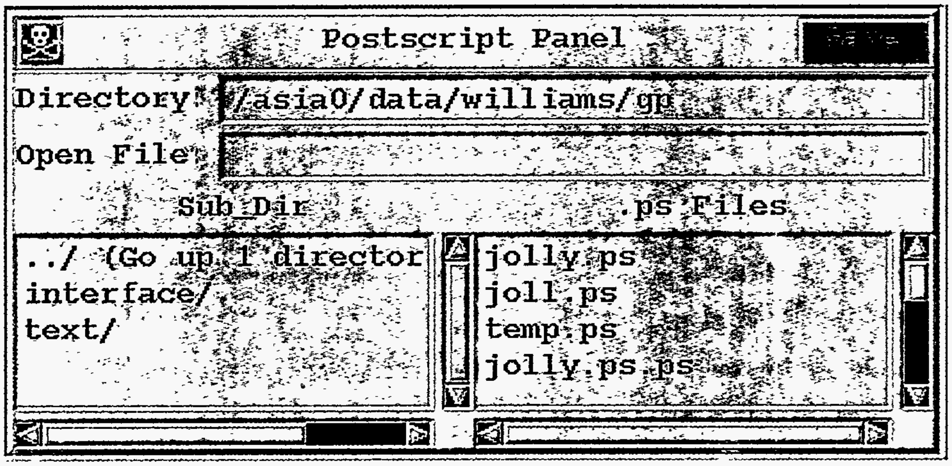
When the directory is satisfactory, either click on a file name or type a file name into the **File** line, and then click on the **Save** button in the top border of the panel to save an Adobe PostScript copy of the **Graphics Canvas** into the file. **Note**: the user *must* have write privileges for the directory, or *no* file will be opened and *no* raster image will be written.

Once a writable file for raster output has been selected and exists in the **File** line, it is only necessary to click on the **Save** button in the top border of the panel to save the raster image of the pictures that are displayed on the **Graphics Canvas** into that file.

### *3.4.11. Orientation—Portrait or Landscape*

The last **Control Panel** button toggles (click on the button) between **Portrait** and **Landscape** orientation of the **Graphics Canvas**. See Figures 1a and 1b.

Figure 39. Postscript Panel



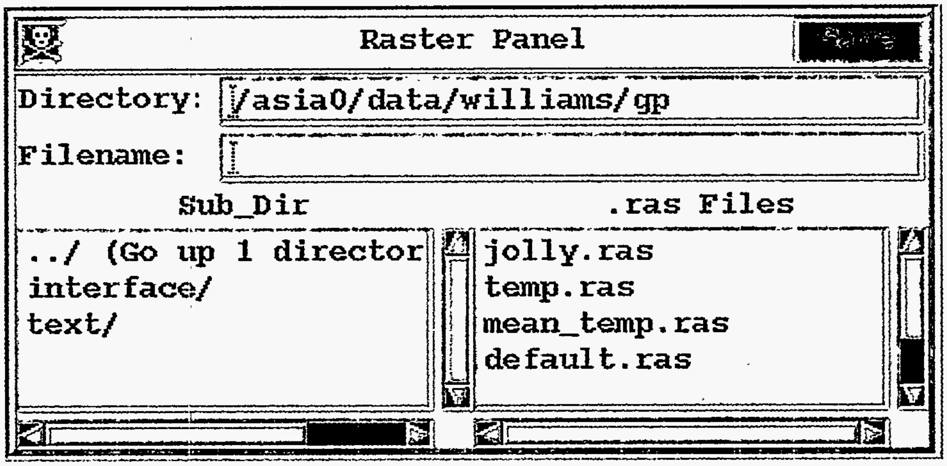
Directory: /asia0/data/williams/gp

Open File:

```
../ (Go up 1 director
interface/
text/
```

```
jolly.ps
joll.ps
temp.ps
jolly.ps.ps
```

Figure 40. Raster Panel



Directory: /asia0/data/williams/gp

Filename:

../ (Go up 1 director
interface/
text/

jolly.ras
temp.ras
mean_temp.ras
default.ras

# 4. Scripts

The scripting capability serves multiple purposes, it allows the user to: (1) predefine and preserve the attributes for a visual presentation of a variable, (2) save the state-of-the-system during a session and return to that state at a later time, and (3) save a sequence of interactive operations for replay.

There are many attributes that are used to create a graphic representation of a variable, including labeling the axes and identifying the variable. This software allows the user to manipulate most of these attributes to create a preferred visual effect and to preserve the attributes for later reuse.

The state-of-the-system can be dumped, in a script form, to an ASCII file. The pertinent contents of all tables, except the entries that are hard coded (default entries), will be included along with the control variables. This file can be used either by including its name in the command line option (-i file), or by including its name in a **Run** command in the script file that is requested on the command line, or by interactively selecting the script file for input.

The command line option (-o file) for saving a sequence of interactive operations will create an ASCII script file that can be edited (perhaps to change the file name(s) or variable names), or not, and replayed.

Assignment statements define or alter entries in the tables of attribute sets, such as: the table of display attribute sets, tables of primary element attribute sets, and tables of basic element attribute sets.

In an assignment statement, the specific table of attribute sets is identified by a prefix, and the table entry is specified by a name for the attribute set; they are separated by an underline. Attributes are enclosed within parentheses. The syntax is:

**tableidentifierprefix _attributesetname(attributeassignments)**
where:

| | |
|---|---|
| **tableidentifierprefix** | identifies the table |
| **attributesetname** | names the attribute set that will appear in the table. |
| **attributeassignments** | can be in either of the forms: |

attribute name=assignment
attribute name(assignment,assignment,...),
attribute name(sub-attribute name = assignment, ...),
attribute name(sub-attribute name = assignment, ...)(another set)...,

where:

assignment=value or 'string' or attribute set name

There are script commands for performing actions, such as: opening a **Graphics Canvas**; capturing the display page that is drawn on the **Graphics Canvas** into a file, in either Adobe PostScript or as a raster image; changing values of the indices (I,J,K,L,M,N); looping indices over ranges of values while saving PostScript or raster images; alternating between portrait (vertical) and landscape (horizontal) orientation of the **Graphics Canvas**; dumping the state of the system into a script file; assigning a colormap table entry to be the active colormap; copying, renaming, and removing entries in the primary and basic elements tables; and running a script file.

The syntax used in this document to describe assignment statements and commands is: **bold** for assignment names, *italic* for description of user-supplied information. Parentheses and commas and equal signs must be supplied as shown. Square brackets [] usage will be described in place.

## 4.1. Display Attributes—D_*name*

Every entry in the display table is used to create a display page. A single display table entry uses the primary elements to create a picture, and each picture is made up of segments.

A display table entry is defined by the assignment statement below. This statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**D_***name*(**priority**=*priority for the picture*,
**type**=*graphics method name*,
**off**=*true or false (1 or 0)*,
**p_name**=*picture template attribute set name*,
**g_name**=*graphics attribute set name in the table defined by* **type**
[,**a**=*first array data attribute set name*
[, **b**=*second array data attribute set name*
[, **c**=*third array data attribute set name*
[, **d**=*fourth array data attribute set name*
[, **e**=*fifth array data attribute set name*
[, **f**=*sixth array data attribute set name*] ] ] ] ] ])

The square brackets [] used above, indicate that the number of array data attribute sets can vary. The number required and their dimensionality are determined by the graphics method name in **type**.

Each picture is assigned a priority (highest value is displayed on top), and each of the segments of a picture has a separate priority. The total priority for each segment is:

(picture priority + segment priority)/1000.

The range of priorities is between 0 and 1. If the priority defined by this equation is greater than 1, the priority is set to 1.

There is a table of graphics attribute sets for each valid graphics method name that can be referenced in **type.** For VCS, the valid graphics method names are: isoline, outline, isofill, outfill, and continents.

The selected picture-template attribute set defines where to display all segments, and it defines how to display all segments, except the legend and data. The selected array data attribute set defines the descriptive information and array data to be displayed. The graphics method defines the legend and determines how the data will be graphically represented.

If the type and primary element attribute sets assigned to **p_name** and **g_name** and those that are required in **a** through **f** are valid, and **off** is false, the picture will be created.

## 4.2. Primary Element Attribute Sets

The primary element attribute sets are defined in three classes of tables: picture templates, graphics methods, and array data. There is one table of attribute sets for picture templates, one for array data, and one table for each of the graphics methods.

### *4.2.1. Picture Template Attributes—P_name*

Picture template attributes describe where segments of a picture will be displayed. The segments are graphic representations of: textual identification of the data; formatted values of single-valued dimensions and mean, maximum, and minimum data values; axes tick marks and labels; boxes and lines; a legend that is graphic method specific; and the data. Picture templates also describe how to display all but the data and legend.

Picture templates allow the user to design templates for either a portrait or landscape orientation of the **Graphics Canvas,** and usually a template designed for one orientation does not work well with the other. If both orientations are commonly used, then it is necessary to devise a technique for naming the templates that will indicate the orientation. In the initialization script file, the prefix “por” is used to indicate that the template is intended for portrait orientation. All others are intended for landscape orientation.

All positions in these attributes are given in a normalized device coordinate (ndc) space, where the largest dimension of the space is 1.0, and the smallest is approximately 0.8.

It is not necessary to provide any picture template attributes for those segments that are not to be displayed; however, the user may find it useful to fill in the attributes, but turn off the segment display. A segment will not be displayed if its priority (**p**) is 0, or if a position coordinate is outside the closed region, 1 < (**x** or **y**) < 0.

The priority of each segment (**p**) is added to the picture priority, and the sum is divided by 1000 to determine the total priority. A priority greater than 1 is set to 1.

The picture template attributes define subsets of attributes that are used for defining the position or space and, for most segments, graphically representing each segment of a picture as described below. The following statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**P_***name*(

The following attribute subsets describe where and how to display the array data attributes that contain textual information. Names of the attribute subsets are the same as those used for the corresponding array data attributes.

| | |
|---|---|
| **File** | (**p**=*priority*,<br>**x**=*x ndc of reference for text*,<br>**y**=*y ndc of reference for text*,<br>**Tt**=*text attributes name*,<br>**To**=*text orientation attribute set name*) |
| **Function** | (same as above) |
| **LogicalMask** | (same as above) |
| **Transform** | (same as above) |
| **source** | (same as above) |
| **name** | (same as above) |
| **title** | (same as above) |
| **units** | (same as above) |
| **crdate** | (same as above) |
| **crtime** | (same as above) |
| **comment#1** | (same as above) |
| **comment#2** | (same as above) |
| **comment#3** | (same as above) |
| **comment#4** | (same as above) |
| **xname** | (same as above) |
| **xunits** | (same as above) |
| **yname** | (same as above) |
| **yunits** | (same as above) |
| **zname** | (same as above) |
| **zunits** | (same as above) |
| **tname** | (same as above) |
| **tunits** | (same as above) |

The following attribute subsets describe how to format floating point values for single-valued dimensions and for the mean, maximum, and minimum of the array data into text, and where and how to display that text. Although attributes can be assigned to all dimensions, only those that are indeed single valued will be displayed in the manner described here.

| | |
|---|---|
| **xvalue** | (**p**=*priority*,<br>**x**=*x ndc of reference for text*,<br>**y**=*y ndc of reference for text*,<br>**Th**=*format attribute set name*,<br>**Tt**=*text attribute set name*,<br>**To**=*text orientation attribute set name*) |
| **yvalue** | (same as above) |
| **zvalue** | (same as above) |
| **tvalue** | (same as above) |
| **mean** | (same as above) |
| **max** | (same as above) |
| **min** | (same as above) |

The following attribute subsets describe where and how to display tick marks. Only the one coordinate is given for the end points of tick marks because the placement of the other coordinate is determined by the graphics method attributes that define tick marks with lists of real-world coordinate values and strings.

| | |
|---|---|
| **xtic#1** | (**p**=*priority*,<br>**y1**=*y ndc for first point on the tick mark*,<br>**y2**=*y ndc for second point on the tick mark*,<br>**Tl**=*line attribute set name*) |
| **xtic#2** | (same as above) |
| **xmintic#a** | (same as above) |
| **xmintic#b** | (same as above) |
| **ytic#1** | (**p**=*priority*,<br>**x1**=*x ndc for first point on the tick mark*,<br>**x2**=*x ndc for second point on the tick mark*,<br>**Tl**=*line attribute set name*) |
| **ytic#2** | (same as above) |
| **ymintic#a** | (same as above) |
| **ymintic#b** | (same as above) |

The following attribute subsets describe where and how to display the axis labels. Only the one coordinate of the reference points is given because the placement of the other coordinate is determined by the graphics method attributes that define labels with lists of real-world coordinate values and strings.

| | |
|---|---|
| **xlabel#1** | (**p**=*priority*,<br>**y**=*y ndc of reference for text*,<br>**Tt**=*text attribute set name*,<br>**To**=*text orientation attribute set name*) |
| **xlabel#2** | (same as above) |
| **ylabel#1** | (**p**=*priority*,<br>**x**=*x ndc of reference for text*,<br>**Tt**=*text attribute set name*,<br>**To**=*text orientation attribute set name*) |
| **ylabel#2** | (same as above) |

The following attribute subsets describe where and how to draw boxes for the picture.

| | |
|---|---|
| **box#1** | (**p**=*priority*,<br>**x1**=*x ndc of first corner of the rectangular box*,<br>**y1**=*y ndc of first corner of the rectangular box*,<br>**x2**=*x ndc of second corner of the rectangular box*,<br>**y2**=*y ndc of second corner of the rectangular box*,<br>**Tl**=*line attribute set name*) |
| **box#2** | (same as above) |
| **box#3** | (same as above) |
| **box#4** | (same as above) |

The following attribute subsets describe where and how to draw lines for the picture.

| | |
|---|---|
| **line#1** | (**p**=*priority*,<br>**x1**=*x ndc of first point of the line*,<br>**y1**=*y ndc of first point of the line*,<br>**x2**=*x ndc of second point of the line*,<br>**y2**=*y ncd of second point of the line*,<br>**Tl**=*line attribute set name*) |
| **line #2** | (same as above) |
| **line #3** | (same as above) |
| **line #4** | (same as above) |

The following attribute subsets describe a rectangular space and text and line attributes to use for drawing a legend for the parameters of the graphics method.

| | |
|---|---|
| **legend** | (**p**=*priority*,<br>**x1**=*x ndc of the first corner of the rectangular space*,<br>**y1**=*y ndc of the first corner of the rectangular space*,<br>**x2**=*x ndc of the next corner of the rectangular space*, |

**y2**=*y ndc of the next corner of the rectangular space*,
**Tt**=*text attribute set name,*
**To**=*text orientation attribute set name,*
**Tl**=*line attribute set name*)

The following attribute subset describes the space for graphically displaying the array data. The graphics method will determine how it is displayed.

**data** (**p**=*priority*,
**x1**=*x ndc of the first corner of the rectangular space*,
**y1**=*y ndc of the first corner of the rectangular space*,
**x2**=*x ndc of the next corner of the rectangular space*,
**y2**=*y ndc of the next corner of the rectangular space*)

). End of the picture template attributes.

### *4.2.2. Graphics Attributes—G*_name*

Graphics attributes define the projection and real-world coordinates of the space for the graphics representation of the data. They define axis labels and their positions and tick mark positions in the real-world coordinate space of the data. They describe the parameters required for the graphics method. The graphics method determines the number of arrays and the number of multiple valued dimensions required for each array.

#### *4.2.2.1. Isoline—Gi_name*

This graphics method draws isolines (lines of constant value) at specified levels to graphically represent the values of a two-dimensional array, and it can display labels on the isolines.

This statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**Gi_***name*(

The default projection is *linear*. In VCS, *linear* is the only projection implemented.

**project**=*projection name*,

The labels and tick marks are defined by lists, which are specified by a list name. Each list value defines a position with respect to the real-world dimension coordinates and a string for display. Each named assignment defines tick positions, and those names that include "ticlabels" also define label strings and positions. For instance: **xticlabels#1=***lat20*, defines a position (in degrees latitude) for each tick and each label as well as a string for each label.

**xticlabels#1=***a prefixed list name for x axis labels and ticks,*
**xticlabels#2=***another prefixed list name for x axis labels and ticks,*
**xmtics#1=***a prefixed list name for x axis minor ticks,*
**xmtics#2=***another prefixed list name for x axis minor ticks,*

**yticlabels#1=***a prefixed list name for y axis labels and ticks,*
**yticlabels#2=***another prefixed list name for y axis labels and ticks,*
**ymtics#1=***a prefixed list name for y axis minor ticks,*
**ymtics#2=***another prefixed list name for y axis minor ticks,*

The data space defined in the picture template, in normalized device coordinates, is mapped to the real-world coordinates given in **datawc.**

**datawc=**(*x1,y1,x2,y2*),

Isolines can be labeled if the **make_labels** switch is set to "y."

**make_labels=***y* or *n*,

If no **lines** attributes are assigned, then an interval will be determined with a starting level at 0.0 and defaults for **Tl**, **Tt**, and **To**. Labels will be the actual level on each. Otherwise, isolines are defined by the subset of attributes shown below. Isolines are drawn at the levels defined by: (**level** + k***increment**), for all k greater than or equal to 0.

When isoline levels are multiply defined, the last definition takes precedence unless a higher priority is specified for an earlier definition.

A background for each isoline label is highlighted with the color indicated in **hilite_ci**.

A single text string can be given for labeling all isolines defined by a set of **lines** attributes. If an asterisk (*) is given instead, it will cause the text representation of each isoline value to be displayed.

The appearance of isolines is determined by the named line attribute set (**Tl**), and the text for labels is determined by the named text (**Tt**) and text orientation attribute sets (**To**).

**lines** (**id=***unique integer identifier,*
**priority=***integer priority value with highest on top,*

**level**=*the starting level,*
**increment**=*increment between successive levels,*
**hilite_ci**=*color index for highlighting labels,*
**label**=*text for a label or * to get the actual level on each,*
**Tl**=*line attribute set name,*
**Tt**=*text attribute set name,*
**To**=*text orientation attribute set name*)
(repeat parenthesized attributes as many times as needed)...

). End of the graphics isoline attributes.

### *4.2.2.2. Outline—Go_name*

The primary purpose of this graphic method is to display an outline of atmospheric model continents using the model's surface type array that indicates land, ocean, and sea-ice. It could also outline sea-ice. It can be used to outline a set of integer values for any array.

This statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**Go**_*name*(

The default projection is *linear.* In VCS, *linear* is the only projection implemented.

**project**=*projection name*,

The labels and tick marks are defined by lists, which are specified by a list name. Each list value defines a position with respect to the real-world dimension coordinates and a string for display. Each named assignment defines tick positions, and those names that include "ticlabels" also define label strings and positions. For instance: **xticlabels#1**=*lat20*, defines a position (in degrees latitude) for each tick and each label as well as a string for each label.

**xticlabels#1**=*a prefixed list name for x axis labels and ticks*,
**xticlabels#2**=*another prefixed list name for x axis labels and ticks*,
**xmtics#1**=*a prefixed list name for x axis minor ticks*,
**xmtics#2**=*another prefixed list name for x axis minor ticks*,

**yticlabels#1**=*a prefixed list name for y axis labels and ticks*,
**yticlabels#2**=*another prefixed list name for y axis labels and ticks*,
**ymtics#1**=*a prefixed list name for y axis minor ticks*,
**ymtics#2**=*another prefixed list name for y axis minor ticks*,

The data space defined in the picture template, in normalized device coordinates, is mapped to the real-world coordinates given in **datawc**.

**datawc=**(*x1,y1,x2,y2*),

The outline is drawn according to the line attributes defined in the basic elements table for the named line attribute set.

**Tl=***line attribute set name*,

Outlines are drawn to enclose the selected values that appear in the data array. There can be as few as one and as many as ten values given.

**outline=**([*n1*,[*n2*,[*n3*,...[*n10*]...]]])

). End of the graphics outline attributes.

*4.2.2.3. Isofill—Gfi_name*

This graphics method fills the area between selected isolevels (levels of constant value) of a two-dimensional array. The manner of filling the area is determined by the named fill area attributes.

This statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**Gfi_***name*(

The default projection is *linear*. In VCS, *linear* is the only projection implemented.

**project=***projection name*,

The labels and tick marks are defined by lists, which are specified by a list name. Each list value defines a position with respect to the real-world dimension coordinates and a string for display. Each named assignment defines tick positions, and those names that include "ticlabels" also define label strings and positions. For instance: **xticlabels#1=***lat20* defines a position (in degrees latitude) for each tick and each label as well as a string for each label.

**xticlabels#1=***a prefixed list name for x axis labels and ticks*,
**xticlabels#2=***another prefixed list name for x axis labels and ticks*,
**xmtics#1=***a prefixed list name for x axis minor ticks*,
**xmtics#2=***another prefixed list name for x axis minor ticks*,

**yticlabels#1=***a prefixed list name for y axis labels and ticks*,
**yticlabels#2=***another prefixed list name for y axis labels and ticks*,
**ymtics#1=***a prefixed list name for y axis minor ticks*,

**ymtics#2**=*another prefixed list name for y axis minor ticks*,

The data space defined in the picture template, in normalized device coordinates, is mapped to the real-world coordinates given in **datawc**.

**datawc**=(*x1*,*y1*,*x2*,*y2*),

If no **ranges** attributes are specified, then the levels will be determined, and the fill area attributes sets will begin with the name "def 37" or "default" and will use as many as needed, in sequence, or will repeatedly use those that exist until all ranges have a fill area name. Otherwise, the bounding isolevels and the type of fill area to be used are defined by the sets of **range** attributes shown below.

**range** (**id**=*unique integer identifier*,
**level1**=*the first bounding level*,
**level2**=*the second bounding level*,
**Tf**=*fill attribute set name*)
(repeat attributes defined in the parenthesis above as needed)...)

). End of the graphics isofill attributes.

### *4.2.2.4. Outfill—Gfo_name*

The primary purpose of this graphic method is to display model continents by filling their area as defined by the model's surface type array that indicates land, ocean, and sea-ice. It will fill a set of integral values in any data array.

This statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**Gfo_***name*(

The default projection is *linear*. In VCS, *linear* is the only projection implemented.

project=*projection name*,

The labels and tick marks are defined by lists, which are specified by a list name. Each list value defines a position with respect to the real-world dimension coordinates and a string for display. Each named assignment defines tick positions, and those names that include "ticlabels" also define label strings and positions. For instance: **xticlabels#1**=*lat20*, defines a position (in degrees latitude) for each tick and each label as well as a string for each label.

**xticlabels#1**=*a prefixed list name for x axis labels and ticks*,
**xticlabels#2**=*another prefixed list name for x axis labels and ticks*,

**xmtics#1**=*a prefixed list name for x axis minor ticks*,
**xmtics#2**=*another prefixed list name for x axis minor ticks*,

**yticlabels#1**=*a prefixed list name for y axis labels and ticks*,
**yticlabels#2**=*another prefixed list name for y axis labels and ticks*,
**ymtics#1**=*a prefixed list name for y axis minor ticks*,
**ymtics#2**=*another prefixed list name for y axis minor ticks*,

The data space defined in the picture template, in normalized device coordinates, is mapped to the real-world coordinates given in **datawc**.

**datawc**=(*x1*,*y1*,*x2*,*y2*),

The area is filled according to the fill area attributes defined in the basic elements table for the named fill area attribute set.

**Tf**=*fill area attribute set name*,

Outlines are filled to enclose the selected values that appear in the data array. There can be as few as one and as many as ten values given.

**outline**=([*n1*,[*n2*,[*n3*,...[*n10*]...]]])

). End of the graphics outfill attributes.

### *4.2.2.5. Continents—Gcon_name*

The primary purpose of this graphic method is to display a predefined, generic, set of continental outlines in a longitude x latitude space. No check is made to verify that the space dimensions are, in fact, longitude and latitude.

This statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**Gcon_***name*(

The default projection is *linear*. In VCS, *linear* is the only projection implemented.

**project**=*projection name*,

The labels and tick marks are defined by lists, which are specified by a list name. Each list value defines a position with respect to the real-world dimension coordinates and a string for display. Each named assignment defines tick positions, and those names that include "ticlabels" also define label strings and positions. For instance: **xticlabels#1**=*lat20,* defines a position (in degrees latitude) for each tick and each label as well as a string for each label.

**xticlabels#1**=*a prefixed list name for x axis labels and ticks,*
**xticlabels#2**=*another prefixed list name for x axis labels and ticks,*
**xmtics#1**=*a prefixed list name for x axis minor ticks,*
**xmtics#2**=*another prefixed list name for x axis minor ticks,*

**yticlabels#1**=*a prefixed list name for y axis labels and ticks,*
**yticlabels#2**=*another prefixed list name for y axis labels and ticks,*
**ymtics#1**=*a prefixed list name for y axis minor ticks,*
**ymtics#2**=*another prefixed list name for y axis minor ticks,*

The data space defined in the picture template in normalized device coordinates is mapped to the real-world coordinates given in **datawc.**

**datawc**=(*x1,y1,x2,y2*),

The area is outlined with a line whose attributes are defined in the named line attribute set.

**Tl**=*line attribute set name,*

). End of the graphics continents attributes.

### *4.2.3. Array Data Attributes—A_name*

#### *4.2.3.1. Minimal Information Required to Select or Compute a Variable*

Variables can be either selected from a file, or computed as a function of previously selected variables. To select a variable from DRS (DRS has a dictionary file for each data file that describes the variables it contains), it is only necessary to assign the dictionary file name (**File**) and sufficient naming attributes (**Source, Name, Title, Units**) to uniquely identify the variable. All other attributes are determined from the description of the variable that exists in the DRS dictionary. To compute a variable, it is only necessary to assign a valid **Function** attribute (rhs of an equation). All other information is determined from the attributes of the variables used in the equation.

#### *4.2.3.2. "Defined" and "Displayed" Attributes*

The attributes described above are termed the "defined" array data attributes; they define the variable. Another, similar, set of attributes is available for refining the attributes for display; these are termed the "displayed" attributes of the variable. The "displayed" attributes are, by default, determined from the "defined" attributes, but they can be assigned. Many of the "displayed" attributes are named the same as the "defined" attributes, but the "defined" attribute names are capitalized and the "displayed" are all lower case (see below).

The "defined" dimension attributes should never be assigned, unless: (1) dimensions are to be transposed, in which case the affected and previous dimension names (**XName, YName, Zname**, **Tname**) need to be supplied in the transposed order; (2) a dimension is to be wrapped around, which makes it necessary to assign wrap indices (**XKwrap, YKwrap, ZKwrap, TKwrap**) and the cycle value (**XCycle, YCycle, ZCycle, TCycle**). The only instance when the "defined" range (**XFirst, Xlast**) or (**YFirst, Ylast**) or (**ZFirst, Zlast**) or (**TFirst, Tlast**) must be specified is when the dimension is to be wrapped but the entire set of dimension values is not wrapped.

The "displayed" dimension attributes need not be assigned unless a subset, reversal, stride, or random selection of dimensions is needed, or unless a grid transformation is needed. Subset and reversal require defining the range (**xfirst, xlast**) or (**yfirst, ylast**) or (**zfirst, zlast**) or (**tfirst, tlast**). Stride selection requires specifying the jump interval (**xjump, yjump, zjump**, or **tjump**) and possibly a range. Random selection requires specifying the node values of the randomly selected nodes (**xvalue, yvalue, zvalue**, or **tvalue**). Grid transformation requires that the node values (**xvalue, yvalue, zvalue**, or **tvalue**) be specified, and it is best to also specify the weights (**xweight, yweight, zweight**, or **tweight**) and bounds (**xbound, ybound, zbound**, or **tbound**).

If any dimension attribute in either the "defined" or the "displayed" attributes is assigned, then that dimension name and all prior dimension names (**XName, YName, Zname, Tname**) must appear in the desired order, even if it is the same as that defined in the dictionary.

Another example for which the duplication of attributes is needed is when a variable is computed; its attributes are taken from the first variable in the equation, and others as needed. It is desirable to assign new naming strings (**source, name, title, units**) to describe the computed variable.

All array data attributes that are assigned by the user are preserved, so that an array data attribute set can be reassigned by altering only the previous assignments made by the user. If any change is made to an array-data attribute set, its defined and displayed attributes are discarded, and only the previously assigned values are retained. The changes are applied to the previously assigned attributes, and the unassigned attributes are refilled, as before.

This software knows about the dimensions named "latitude" and "longitude," and, if no other assignments have been made, sets the range for latitude, to (**x, y, z,** or **t)first**=–90 and (**x, y, z,** or **t)last**=90, and the range for longitude, to (**x, y, z,** or **t)first**=–180 and (**x, y, z,** or **t)last**=180, and sets wrap indices (**X, Y, Z,** or **T)Kwrap** as needed and (**X, Y, Z,** or **T)Cycle**=360.

### *4.2.3.1. Order of Array Dimensions*

Array data that has been stored with DRS can have as many as four dimensions, and any or all of these dimensions can be single valued. Single-valued dimensions can, of course, be in any position without affecting the order of the data. *However, to visualize the data it is necessary to request the data with the dimensions (even single valued) in the desired order*. In the two-dimensional graphics methods, the X dimension is defined on the horizontal axis, and the Y dimension is defined on the vertical axis. Single values must be chosen for the dimensions that are not used by the graphic display method. The single-valued dimensions can have their values formatted to text and displayed in segments, and the order can be important (when there is more than one) because of their placement.

This statement can be used to define a table entry or to change some, or all, of the attributes in an existing table entry.

**A_***name*(

The file containing the selected data is named here.

**File**='*/path/filename*',

If a functional assignment statement exists, it overrides any **File** assignment.

The functional assignment statement consists of operands and operators. The operators are: add (+), subtract (–), multiply (*), divide (/), power (**). The operands can be either numbers or names of array-data attribute sets or function references with arguments. In VCS, only one function is available (sqrt), but other functions will be added to the list as the need arises, and time is available.

**Function**='*assignment statement*',

Each selected variable has a logical mask assigned as it is retrieved. If any data value is greater than or equal in absolute value to 10**20, that node point is masked out (mask value is false). The logical mask assigned here is and'ed with that mask.

When a variable is used in a **Function** assignment statement, its mask is and'ed with the masks of the other variables to form a mask for the computed variable.

The operators that can be used in this logical assignment statement are: greater than (>), less than (<), equal (=), greater than or equal (>=), less than or equal (<=). The operands are variables, which can include the variable defined by this attribute set. Operands must have the same nonsingular dimensions as the one defined by these array attributes. The dimensions need not be in the same order.

**LogicalMask=**'*logical assignment statement for masking out data* ',

The default grid transformations can be changed for particular dimensions with the transform assignment. Default transform functions are defined for each dimension: “latitude” and “longitude” are area weighted (*awt*), and all other dimensions are transformed with linear interpolation (*linear*). In VCS, only the two methods exist.

**Transform=**'*transformation function (dimension name,...)* ',

The following four character strings define the variable. It is necessary to provide enough of these strings to unambiguously determine the variable with respect to other variables in the **File.** If the variable is defined as a **Function** of other variables, these strings need not be supplied; they will become the same as the displayed attributes for the first variable in the assignment statement.

**Source=**'*description of data source* ', (120 char limit)
**Name=**'*name of the variable*', (16 char limit)
**Title=**'*description of the variable*', (80 char limit)
**Units=**'*units description for the variable*', (40 char limit)

The following four character strings can be used to substitute naming strings that will be displayed. If these strings are not supplied, they will be the same as the defined set of strings immediately above.

**source=**'*substitute description of data source*', (120 char limit)
**name=**'*substitute name of the variable*', (16 char limit)
**title=**'*substitute description of the variable*', (80 char limit)
**units=**'*substitute units description for the variable*', (40 char limit)

The variable type and creation date and time are determined either from the DRS dictionary or from the description of the variables on the right-hand side of the **Function** assignment. User assignment of these strings will be overridden.

| | |
|---|---|
| **Type** | is the data type: character (C*n), integer (I*n), or real (R*n) |
| **CrDate** | is the date the data were created |
| **CrTime** | is the time the data were created |

The type can be entered but it will not be displayed. The creation date and time can be displayed.

**type=**'*the data type*', Character (C*n), integer (I*n), or real (R*n)
**crdate=**'*substitute data creation date*', (8 char limit)
**crtime=**'*substitute data creation time*', (8 char limit)

The following four comment strings can be supplied and used for the display.

**comment#1=**'*comment line for display only*', (120 char limit)
**comment#2=**'*comment line for display only*', (120 char limit)
**comment#3=**'*comment line for display only*', (120 char limit)
**comment#4=**'*comment line for display only*', (120 char limit)

Those dimensions for which attributes are to be assigned must have their names also assigned, and all prior dimensions must also be named.

The order of dimensions can be assigned even when the variable is to be computed. If they are not assigned, the order is determined from the other variables by left to right examination of their dimensions.

**XName=**'*name of the 1st dimension*', (16 char limit)
**XUnits=**'*units of the 1st dimension*', (40 char limit)

The data can be wrapped around by specifying wrap indices, **XKwrap**(*k1*,*k2*), and a cycle value, **XCycle**(*C*), so that a dimension then has values,

$$X = x(i) + k*C \text{ (for all k=}k1\text{ to }k2\text{ and i = i1 to i2 for each k)}$$

where x(i1)=**XFirst,** and x(i2)=**XLast**. For example, longitude can be wrapped negatively by 360 degrees and positively by 360 degrees with:

$k1$=-1 and $k2$=1 and $C$=360.

The range of values actually requested can then be defined within this larger range. The requested range is specified by **xfirst** and **xlast** below.

**XKwrap=***(first wrap index value,last wrap index value)*,
**XCycle=***wrap-around cycle*,
**XFirst=***first value of wrap range*,
**XLast=***last value of wrap range*,

The "defined" values, bounds, and weights can be assigned but will be overridden by the actual values either taken from DRS or computed. If a file name is specified, the bounds and weights will be sought from the same file and, if not found, will be computed. After the bounds are determined, the weights are computed as the difference between bounds for each node or difference between the sine of the bounds if the dimension is named, "latitude." If a **Function** is assigned, these values are taken from the first variable in the equation and as many afterward as necessary to fill all dimensions.

| | |
|---|---|
| **XValue** | values of the first dimension nodes |
| **XBound** | values of the first dimension node boundaries |
| **XWeight** | weights to be used in computing weighted averages |

A substitute for the dimension name and units can be defined for use on the display; by default, they are the same as the "defined" strings.

**xname**='*substitute name of the 1st dimension*', (16 char limit)
**xunits**='*substitute units description of the 1st dimension*', (40 char limit)

Every nth value of a dimension, beginning with **xfirst**, can be selected by specifying **xjump**=n. By default, **xjump** is 1.

**xjump**=*jump index,*

The actual range of values chosen from a dimension (wrapped or not) is specified by **xfirst** and **xlast,** and the direction of the dimension is determined by the direction given. However, if grid transformation is specified by assigning a list of values for **xvalue,** then **xfirst** and **xlast** are given the first and last values of the list.

**xfirst**=*first value of the selected range,*
**xlast**= *last value of the selected range,*

Values, bounds, and weights of the actual dimensions are determined either from the defined values or computed, or they can be assigned as named lists or as values. If assigned values do not coincide with actual nodes, then a grid transformation for the dimension is performed.

**xvalue**=*list of values or actual values of the dimension nodes,*
**xbound**=*list of values or actual values of the node bounds,*
**xweight**=*list of values or actual values of the weights for each node,*

The **Y**, **Z**, and **T** dimensions each have the same attributes as **X,** and the descriptions are identical with **X** or **x** changed to the proper dimension specifier; therefore, the descriptions from **Xname** to **xweight** above are not reiterated for each dimension.

). End of the array data attributes.

## 4.3. Basic Element Attributes

The basic elements are used to define attributes of the primary elements; they are assigned in the same manner as the primary elements, except that there are no attribute names, which means the order is necessarily fixed and all values must be given.

### *4.3.1. Text Attributes—Tt_name*

**Tt***_name(font,pr,cexp,csp,ci)*

The text attributes are:

| | |
|---|---|
| *font* | character font index |
| *pr* | character precision index |
| *cexp* | horizontal character expansion factor |
| *csp* | character spacing |
| *ci* | color index |

Font numbers are GKS implementation dependent; if the requested font number does not exist for the implementation, then font number 1 is used.

Characters are drawn with increasing precision from (0) character, to (1) string, to (2) stroke. The precision is both GKS and font implementation specific and has an effect on the manipulation and clipping of text. ***Suggestion***: either read the GKS manual and experiment with fonts and precisions, or use stroke precision with each font.

Characters can be expanded in the horizontal direction by a character expansion factor that multiplies the width. It is, by default, the number 1.

Character spacing defines the space between characters, in normalized device coordinates. It can be changed from the default, 0, to negative spacing, which overlaps characters, and positive spacing, which increases the space between characters.

The color index is an index into the color table, which has colors numbered 0 through 255. Colors numbered 240 through 255 cannot be changed, and 0 through 4 are used for the interactive windows and panels.

### *4.3.2. Text Orientation Attributes—To_name*

**To**_*name*(*chh*,*chua*,*chp*,*chhal*,*chval*)

The text orientation attributes are:

| | |
|---|---|
| *chh* | character height |
| *chua* | character up angle |
| *chp* | character path |
| *chhal* | character horizontal alignment |
| *chval* | character vertical alignment |

The character height is given in normalized device coordinates ($0 < \text{chh} \leq 1$). A value of 0.01 is easily visible.

The character up angle is measured positive clockwise from the vertical in units of degrees.

The character path is either “r” for right, or “l” for left, or “u” for up, or “d” for down.

The horizontal alignment refers to the character string, and it is either “l” for the left end, or “c” for center, or “r” for the right end.

The vertical alignment refers to the character space, and it is either “t” for top of the character space, or “c” for cap (or top) of the physical character, or “h” for half (or middle) of the character, or “b” for bottom of the physical character, or “B” for the base (or bottom) of the character space.

These attributes and the text attributes (above) are both necessary for each description of text, but, by separating the two, it is possible to significantly reduce the number of predefined attribute sets required to cover a large variety of uses.

### *4.3.3. Fill Area—Tf_name*

**Tf**_*name*(*fais*,*fasi*,*fa*ci,*rx*,*ry*,*w*,*h*)

The fill area attributes are:

| | |
|---|---|
| *fais* | fill area interior style |
| *fasi* | fill area style index |
| *faci* | fill area color index |
| *rx* | x coordinate of the pattern reference point |
| *ry* | y coordinate of the pattern reference point |
| *w* | pattern width |
| *h* | pattern height |

The area is filled according to the fill area interior style (*fais*), which must be either:

| | |
|---|---|
| 0 | hollow |
| 1 | solid |
| 2 | pattern |
| 3 | hatch |

If hatch or pattern is used, then the fill area style index (*fais*) determines which of those that have been defined will be used. If the index number does not exist, then number 1 is used.

The color index applies only if hollow, solid, or hatch is chosen. The color index is an index into the color table, which has colors numbered 0 through 255. Colors numbered 240 through 255 cannot be changed, and 0 through 4 are used for the interactive windows and panels.

The reference point (*rx*,*ry*), width (*w*), and height (*h*) in normalized device coordinates (0 to 1) are used only if pattern is chosen. By default, *rx*=*ry*=0 and *w*=*h*=1.

### *4.3.4. Formats—Th_name*

**Th**_*name*((*name*,*units*,*format*)(*name*,*units*,*format*)...)

A different format is assigned for each of a set of variables that are defined by name and units. The format attributes are:

*name* names the variable that will use this format
*units* defines the units of the variable that will use this format
*format* describes how to format the variable

The variables that can use this formatting capability are the single-valued dimensions of the array data, which use the dimension name and units descriptors, and the mean, maximum, and minimum of the array data variable, which are named “mean,” “max,” and “min,” respectively, and the units are the units of the variable.

The units attribute can be specified with an asterisk (*) to indicate that they do not have an effect on the format; that is, any variable with the specified name (*) will use the format.

A single-format attribute set can contain multiple formats that are selected by comparing the name and units of the variable to be formatted with those defined by the attributes. The formats are used to convert floating point numbers to a text form. They are especially rich in capabilities for displaying time, with a different format specification for each of the different units of time that can be encountered.

The units attribute can be specified with an asterisk (*) to indicate that they do not have an effect on the format; that is, any variable with the (*) specified will use the format.

Formats are character strings that can be as much as 120 characters long. They can contain printing characters and blanks and:

%n to display the name of the variable.
%u to display the units of the variable.
%g to format the floating point value to a string of digits.
%t to format the floating point value using the bracketed description that must follow as: "%t[h:m:s]." The name and units of the variable determine which of the formats to use. The available format descriptors that can be used within the brackets are designed for time:

| | |
|---|---|
| s | second |
| m | minute |
| h | hour |
| M | month number |
| y | last two digits of the year (e.g., 93) |
| c | full century and year designation (e.g., 1993) |
| mon | lower case, three-character, month designation (e.g., jan) |
| Mon | capitalized, three-character, month designation (e.g., Jan) |
| MON | upper case, three-character, month designation (e.g., JAN) |
| S | upper case, three-character, season designator (e.g., DJF) |
| Sea | capitalized, three-character/month, season designator (e.g., Dec-Jan-Feb) |
| SEA | uppercase, three-character/month, season designator (e.g., DEC-JAN-FEB) |
| season | lowercase, full-month, season designator (e.g., december-january-february) |
| Season | capitalized, full-month, season designator (e.g., December-January-February) |
| SEASON | uppercase, full month, season designator (e.g., DECEMBER-JANUARY-FEBRUARY) |

### *4.3.5. Line Attributes—Tl_name*

**Tl**_*name*(*type*,*lwsf*,*ci*)

Line attributes are:

| | |
|---|---|
| *type* | type of line |
| *lwsf* | line width scale factor |
| *ci* | color index |

Line types defined in the existing GKS are: 1—solid, 2—dashed, 3—dotted, 4—dot dashed.

The line width scale factor is, by default, 1.0. Some experimentation may be necessary if other than the default is to be used.

The color index is an index into the color table, which has colors numbered 0 through 255. Colors numbered 240 through 255 are not changeable, and 0 through 4 are used for the interactive windows.

### *4.3.6. Marker Attributes—Tm_name*

**Tm**_*name*(*type*,*size*,*ci*)

Marker attributes are:

| | |
|---|---|
| *type* | type of marker |
| *size* | marker size scale factor |
| *ci* | color index |

Marker types defined in the existing GKS are: 1—dot, 2—plus, 3—asterisk, 4—circle, 5—cross.

The marker size scale factor is, by default, 1.0. Some experimentation may be necessary if other than the default is to be used.

The color index is an index into the color table, which has colors numbered 0 through 255. Colors numbered 240 through 255 are not changeable, and 0 through 4 are used for the interactive windows.

### *4.3.7. Patterns—Pat_name*

**Pat**_*name*(*Ix*,*Jy*,*ci*(*Ix*,*Jy*) )

Patterns are defined by:

| | |
|---|---|
| *Ix*,*Jy* | dimensions of pattern array |
| *ci*(*Ix*,*Jy*) | two dimensional array of color indices defining the pattern |

A pattern index number will be established at the time the pattern is defined. Pattern index number 1 (the default pattern) is predefined as a black and white checkerboard pattern. Other higher indices are established in order as additional patterns are defined. These pattern indices are the fill area style index (*fasi*) that can be chosen when fill area interior style (*fais*) number 2 (pattern) is chosen.

There are predefined patterns in GKS, and the capability to add or replace patterns is a part of the GKS standard. The predefined patterns in GKS are patterns numbered 1 through 4, which are replaced first. This basic element allows that capability.

The color index is an index into the color table, which has colors numbered 0 through 255. Colors numbered 240 through 255 are not changeable, and 0 through 4 are used for the interactive windows.

### *4.3.8. Colormaps—C_name*

**C**_*name*(*red*,*green*,*blue*, ...)

A colormap contains 240 user definable colors. The color mixtures are defined in terms of percentages of red, green, and blue colors (0 to 100% for each). There are 16 additional colors that are usable, but they are fixed and cannot be changed by the user. The fixed colors are, in order: 240-white, 241-black, 242-red, 243-green, 244-blue, 245-yellow, 246-cyan, 247-magenta, 248-orange, 249-brown, 250-violet, 251-olive green, 252-grey, 253-light green, 254-light red, and 255-light blue.

### *4.3.9. Lists—L_name*

**L**_*name*(*value*,'*string*',...)

Lists consist of sequential pairs of (value, string) combinations, which can be referred to in their entirety or as a single, indexed pair. The ability to refer to predefined lists in this manner offers a shorthand method for assigning sequences of values, or strings, or coincident pairs of both, and overcomes the necessity to enter them at execution time.

Lists are useful for defining tick mark positions and for defining dimension labels and their positions.

Lists provide a manner of defining the node points for the dimensions of a new grid, which is necessary to request grid transformations.

Lists are useful for defining many of the assignable attributes of array data; indexed list assignments can be used to step through a dimension (such as time) by incrementing the index value. This provides a simple means for producing time-sequential raster-image files to be used in an animation.

## 4.4. Script Commands

The script commands define the actions that are necessary to preserve an interactive session as a script and to mimic that session in a noninteractive replay of the script. The button, menu, and panel manipulation visuals do not exist during the replay, but the graphics and postscript and raster images are reproduced.

None of the commands described here are case sensitive.

### *4.4.1. Canvas*

**Canvas**(*open* or *close*)

A **Graphics Canvas** is either opened or closed on the attached graphics workstation. If none is logically attached to the process, nothing will happen. If the interactive mode has been previously entered, this command will do nothing, because a **Graphics Canvas** is already open, and it cannot be closed by this command.

### *4.4.2. Clear*

**Clear**(*display name* or *all*)

If a display name is given, the display table entry with the identical name will be removed, and its graphic representation will be cleared from the **Graphics Canvas**, if it is open.

If "all" is specified, then all display table entries will be cleared, and the entire **Graphics Canvas** will be cleared, if it is open.

### *4.4.3. Color*

**Color**(*color table name*)

The table of color attribute sets can contain many entries, but only one can be active at any given instant. There is always an active color table. This command assigns one of the table entries as the active color table.

### *4.4.4. Copy*

**Copy**(*attribute set prefix _ attribute set name*, *attribute set name*)

The first is copied to the second, and the new table entry is established.

Since the copy command will copy most attribute sets, it requires the identifying prefix for the table, followed by an underline (_), followed by the name. The display, pattern, and color attribute sets cannot be copied.

### *4.4.5. Dump*

**Dump**(*filename*)

Dumps the state-of-the-system in a script form to the named file.

### *4.4.6. Index*

**Index**([*I=i*] [,*J=j*] [,*K=k*] [,*L=l*] [,*M=m*] [,*N=n*] or
[*I+*] [,*J+*] [,*K+*] [,*L+*] [,*M+*] [,*N+*] or
[*I-*] [,J-] [,*K-*] [,*L-*] [,M-] [,*N-*] or
[,*I=i+*] [,*J=j+*] [,*K=k+*] [,*L=l+*] [,*M=m+*] [,*N=n+*])

The internally defined indices (I, J, K, L, M, and N) can be set to a positive definite value (I = i > 0), or incremented by one (I++), or decremented by one (I--), or incremented by a specified amount (I=i+), or decremented by a specified amount (I=i-).

The index value cannot be set to anything less than 1.

An indexed list name can be used to specify a particular element of the list. The actual index value used is the index value, less one, modulo the number of values in the list, which removes any possibility of referring to nonexistent elements.

### *4.4.7. Loop*

**Loop**([*I(first,last*[*,increment*]*)*]
[,][*J(first,last*[*,increment*]*)*]
[,][*K(first,last*[*,increment*]*)*]
[,][*L(first,last*[*,increment*]*)*]
[,][*M(first,last*[*,increment*]*)*]
[,][*N(first,last*[*,increment*]*)*]
[,][*Raster(*[*file name*]*)*]
[,][*Postscript(*[*file name*]*)*])

Include only those indices that are needed to create the desired effect and only the output command that is desired, if any. The increment need not be supplied if it is unity or negative unity. The sign of the increment will be set according to the relative sizes of first and last; only the size of the increment is retained from the user input. The commas only need be included between successive arguments.

The loop command was created to aid in making raster images of temporal sequences of variables. The effect is as if the following pseudo code were executed:

```
Loop I from i1 to i2 by di
Loop J from j1 to j2 by dj
Loop K from k1 to k2 by dk
Loop L from l1 to k2 by dk
Loop M from m1 to m2 by dm
Loop N from n1 to n2 by dn
          Raster(filename)
          Postscript(filename)
End_all_loops
```

It is, of course, necessary that the single index or indices that are looped be used in an array data attribute that will have an effect on an active display table entry; otherwise, the loop will have no effect.

### *4.4.8. Page*

**Page**(*orientation*)

The **Graphics Canvas**, postscript, and raster image orientation can be controlled by this command. Only "portrait" (y > x) or "landscape" (x > y) orientations are permitted.

### *4.4.9. Postscript*

**Postscript**([*path/file name*])

If a file name is given with no path, then the path defined for the base directory (see section **2.0. The Command Line**) will be prefixed to the file name. If neither is given, or if the path and file name matches a previously opened postscript file, then it will be used. If neither is given, and if no previous postscript file is open, then a file named "*basedirectory*/defaultxx" (xx is a two digit number) will be used unless all 99 defaults exist, in which case no postscript file will be written.

If the path and file name does not match the open postscript file name, then the open postscript file will be closed, and an attempt will be made to open the new file. If the new file already exists, then a file named "*basedirectory*/defaultxx" will be attempted as above. If the new file does not exist, but cannot be opened for write, then no postscript will be written. Otherwise, the new file is opened.

If possible, the current display page will be written to the open file in postscript. The file will remain open, but no more graphics are written to it until subsequent "**Postscript()**" or "**Postscript(***same file name***)**" commands are given.

### *4.4.10. Raster*

**Raster**([*path/file name*])

If no path/file name is given and no previous raster file has been given, then basedirectory/ default will be used; but, if a previous raster file has been given, then it will be used.

If a file cannot be opened for append, then no raster image will be written.

If possible, the file will be opened for append, and the current display will be appended to it in a raster image form. The file will then be closed, and no more will be written to it until subsequent "**Raster**()" or "**Raster**(*same file name*)" commands are given. A subsequent "**Raster**(*different filename*)" command will append a raster image to the different file name.

### *4.4.11. Remove*

**Remove**(*attribute set prefix _attribute set name*)

The named attribute set will be removed from its table.

Since the remove command will remove most attribute sets, the full identifier is required for the attribute set name that is to be removed. The display and pattern attribute sets cannot be removed.

### *4.4.12. Rename*

**Rename**(*attribute set prefix _attribute set name*, *attrubute set name*)

The first attribute set is renamed to the second. This rename cannot happen if the attribute set is used in a picture.

Since the rename command will rename most attribute sets, the full identifier is required for the attribute set name that is to be renamed. The display, pattern, and color table attribute sets cannot be renamed.

### *4.4.13. Run*

**Run**(*script file*)

The script file will be read and executed, and then control will return to the current script file

## 4.5. Script Example

The simple script example in Figure 41 defines the page with two pictures that are shown in Figure 42. The initial.attributes file did not exist, so all attributes that were used are defined, either in Figure 41 or in the set of default attributes.

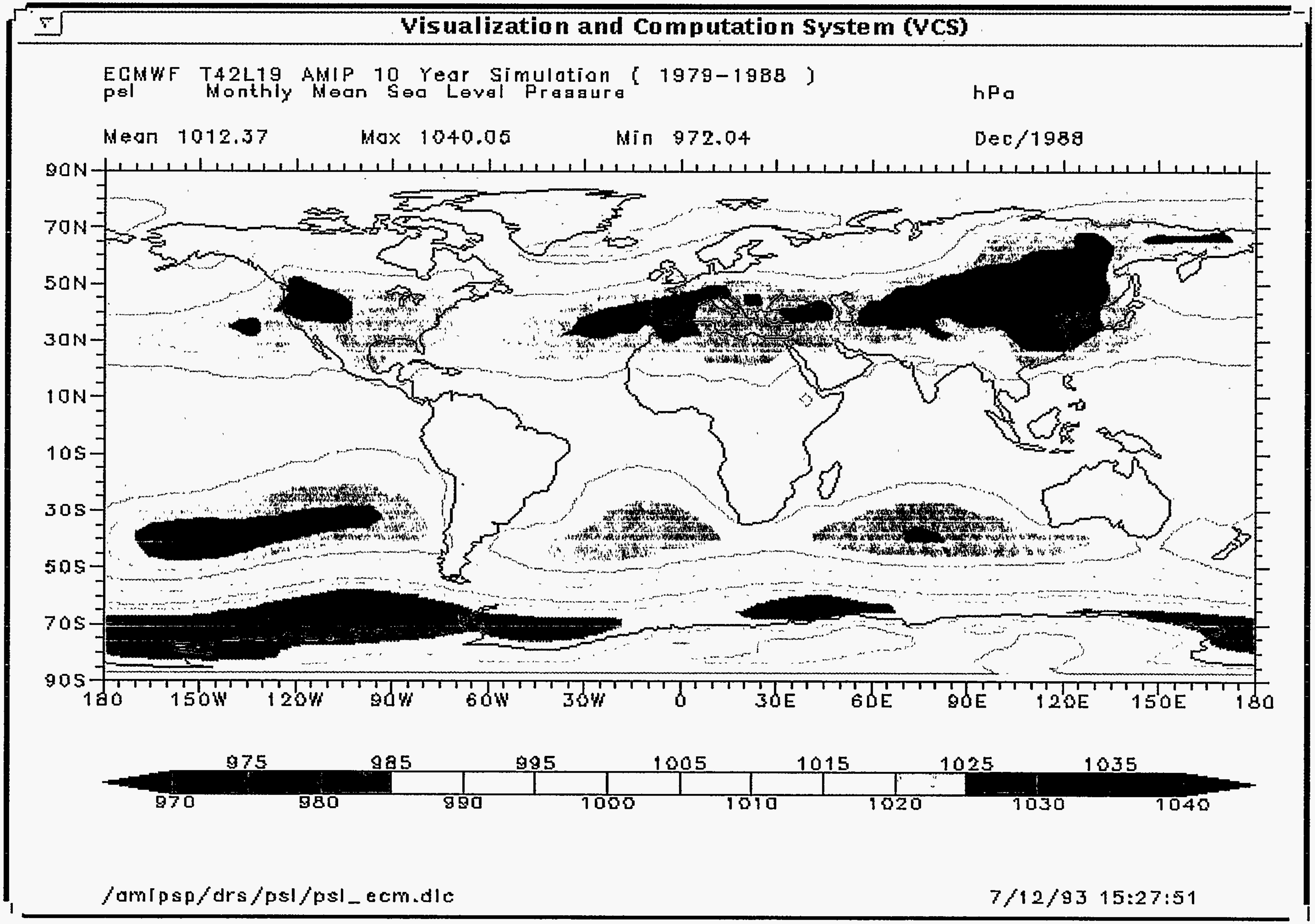
Figure 41. Example Script

```
Th_AMIP((‘time’,’month’,’%t[Mon/c+1979]’)(‘mean’,’*’,’%n %g’)
(‘max’,’*’,’%n %g’)(‘min’,’*’,’%n %g’)(‘month’,’*’,’%t[Mon/c+1979]’))

L_lat5(-90,’90S’,-85,’85S’,-80,’80S’,-75,’75S’,-70,’70S’,-65,’65S’,-60,’60S’,
-55,’55S’,-50,’50S’,-45,’45S’,-40,’40S’,-35,’35S’-30,’30S’,-25,’25S’,-20,’20S’,
-15,’15S’,-10,’10S’,-5,’5S’,0,’EQ’,5,’5N’,10,’10N’,15,’15N’,20,’20N’,25,’25N’
,30,’30N’,35,’35N’,40,’40N’,45,’45N’,50,’50N’,55,’55N’,60,’60N’,
65,’65N’,70,’70N’,75,’75N’,80,’80N’,85,’85N’,90,’90N’)

L_lon5(
-360,’0’  ,
-355,’5E’  ,-350,’10E’ ,-345,’15E’ ,-340,’20E’ ,-335,’25E’ ,-330,’30E’ ,
-325,’35E’ ,-320,’40E’ ,-315,’45E’ ,-310,’50E’ ,-305,’55E’ ,-300,’60E’ ,
-295,’65E’ ,-290,’70E’ ,-285,’75E’ ,-280,’80E’ ,-275,’85E’ ,-270,’90E’ ,
-265,’95E’ ,-260,’100E’,-255,’105E’,-250,’110E’,-245,’115E’,-240,’120E’,
-235,’125E’,-230,’130E’,-225,’135E’,-220,’140E’,-215,’145E’,-210,’150E’,
-205,’155E’,-200,’160E’,-195,’165E’,-190,’170E’,-185,’175E’,-180,’180’ ,
-175,’175W’,-170,’170W’,-165,’165W’,-160,’160W’,-155,’155W’,-150,’150W’,
-145,’145W’,-140,’140W’,-135,’135W’,-130,’130W’,-125,’125W’,-120,’120W’,
-115,’115W’,-110,’110W’,-105,’105W’,-100,’100W’,-95 ,’95W’ ,-90 ,’90W’ ,
-85 ,’85W’ ,-80 ,’80W’ ,-75 ,’75W’ ,-70 ,’70W’ ,-65 ,’65W’ ,-60 ,’60W’ ,
-55 ,’55W’ ,-50 ,’50W’ ,-45 ,’45W’ ,-40 ,’40W’ ,-35 ,’35W’ ,-30 ,’30W’ ,
-25 ,’25W’ ,-20 ,’20W’ ,-15 ,’15W’ ,-10 ,’10W’ ,-5  ,’5W’  , 0  ,’0’   ,
 5  ,’5E’  , 10 ,’10E’ , 15 ,’15E’ , 20 ,’20E’ , 25 ,’25E’ , 30 ,’30E’ ,
 35 ,’35E’ , 40 ,’40E’ , 45 ,’45E’ , 50 ,’50E’ , 55 ,’55E’ , 60 ,’60E’ ,
 65 ,’65E’ , 70 ,’70E’ , 75 ,’75E’ , 80 ,’80E’ , 85 ,’85E’ , 90 ,’90E’ ,
 95 ,’95E’ , 100,’100E’, 105,’105E’, 110,’110E’, 115,’115E’, 120,’120E’,
 125,’125E’, 130,’130E’, 135,’135E’, 140,’140E’, 145,’145E’, 150,’150E’,
 155,’155E’, 160,’160E’, 165,’165E’, 170,’170E’, 175,’175E’, 180,’180’ ,
 185,’175W’, 190,’170W’, 195,’165W’, 200,’160W’, 205,’155W’, 210,’150W’,
 215,’145W’, 220,’140W’, 225,’135W’, 230,’130W’, 235,’125W’, 240,’120W’,
 245,’115W’, 250,’110W’, 255,’105W’, 260,’100W’, 265,’95W’ , 270,’90W’ ,
 275,’85W’ , 280,’80W’ , 285,’75W’ , 290,’70W’ , 295,’65W’ , 300,’60W’ ,
 305,’55W’ , 310,’50W’ , 315,’45W’ , 320,’40W’ , 325,’35W’ , 330,’30W’ ,
 335,’25W’ , 340,’20W’ , 345,’15W’ , 350,’10W’ , 355,’5W’  , 360,’0’  )

C_AMIP(100,100,100,0  ,0  ,0  ,    44.7,62.4,100,  29.8,44.3,62.0,
 76.9,84.3,100,  100,100,0,
 0  ,0  ,0  ,  0  ,0  ,0  ,100,100,100,
 0 ,0 ,0 , 100,0 ,0 ,0 ,100,0 ,
 0 ,0 ,100, 100,100,0 ,0 ,100,100,
 100,0 ,100,

  98.4314, 98.4314, 100,
  78.4314, 12.5490,3.1373,
  88.6274, 20,5.4902,
  94.5098, 33.3333, 12.5490,
 100, 55.6863, 16.4706,
  99.6078, 80, 24.7059,
  97.6471, 87.8431, 24.7059,
  95.6863, 100, 24.3137,
  79.2157, 100, 83.5294,
  52.5490, 100, 94.5098,
  36.4706, 100, 94.5098,
7.0588, 78.4314, 100,
  23.5294, 52.9412, 100,
  57.6471, 20.7843, 99.6078,
  84.7059,6.6667, 99.6078,
 100,0, 100,
  80.7843, 100, 15.6863,
  44.3137, 100, 14.1176,
  23.1373, 85.0980, 56.8627,
  65.8824,0,0,
  47.4510,8.2353, 11.3725,

  0, 50,100,0, 40,100,0, 30,100,0, 20,100,0, 10,100,

  0, 0,100, 10, 0,100, 20, 0,100, 30, 0,100, 40, 0,100,
 50, 0,100, 60, 0,100, 70, 0,100, 80, 0,100, 90, 0,100,
100, 0,100, 100, 0, 90, 100, 0, 80, 100, 0, 70, 100, 0, 60,
100, 0, 50, 100, 0, 40, 100, 0, 30, 100, 0, 20, 100, 0, 10,

100, 0, 0, 95, 10, 10, 90, 20, 20, 85, 30, 30, 80, 40, 40,
 75, 50, 50, 70, 60, 60, 65, 70, 70, 60, 80, 80, 55, 90, 90,
 50,100,100, 45,100, 90, 40,100, 80, 35,100, 70, 30,100, 60,
 25,100, 50, 20,100, 40, 15,100, 30, 10,100, 20, 05,100, 10,

  0,100, 0, 10, 95, 10, 20, 90, 20, 30, 85, 30, 40, 80, 40,
 50, 75, 50, 60, 70, 60, 70, 65, 70, 80, 60, 80, 90, 55, 90,
100, 50,100, 90, 45,100, 80, 40,100, 70, 35,100, 60, 30,100,
 50, 25,100, 40, 20,100, 30, 15,100, 20, 10,100, 10, 5,100,

  0, 0,100, 10, 10, 95, 20, 20, 90, 30, 30, 85, 40, 40, 80,
 50, 50, 75, 60, 60, 70, 70, 70, 65, 80, 80, 60, 90, 90, 55,
100,100, 50, 100, 90, 45, 100, 80, 40, 100, 70, 35, 100, 60, 30,
100, 50, 25, 100, 40, 20, 100, 30, 15, 100, 20, 10, 100, 10, 5,
100,  0,  0,  95,  0,  0,  90,  0,  0,  85,  0,  0,  80,  0,  0,
 75,  0,  0,  70,  0,  0,  65,  0,  0,  60,  0,  0,  55,  0,  0,
 50,  0,  0,  45,  0,  0,  40,  0,  0,  35,  0,  0,  30,  0,  0,
 25,  0,  0,  20,  0,  0,  15,  0,  0,  10,  0,  0,  05,  0,  0,

  0,  0,  0,0,  5,  0,0, 10,  0,0, 15,  0,0, 20,  0,
  0, 25,  0,0, 30,  0,0, 35,  0,0, 40,  0,0, 45,  0,
  0, 50,  0,0, 55,  0,0, 60,  0,0, 65,  0,0, 70,  0,
  0, 75,  0,0, 80,  0,0, 85,  0,0, 90,  0,0, 95,  0,

  0,100,  0,0, 95,  5,0, 90, 10,0, 85, 15,0, 80, 20,
  0, 75, 25,0, 70, 30,0, 65, 35,0, 60, 40,0, 55, 45,
  0, 50, 50,0, 45, 55,0, 40, 60,0, 35, 65,0, 30, 70,
  0, 25, 75,0, 20, 80,0, 15, 85,0, 10, 90,0,  5, 95,

  0,  0,100,0,  0, 95,0,  0, 90,0,  0, 85,0,  0, 80,
  0,  0, 75,0,  0, 70,0,  0, 65,0,  0, 60,0,  0, 55,
  0,  0, 50,0,  0, 45,0,  0, 40,0,  0, 35,0,  0, 30,
  0,  0, 25,0,  0, 20,0,  0, 15,0,  0, 10,0,  0,  5,

  0,  0,  0,  05,  5,  5,  10, 10, 10,  15, 15, 15,  20, 20, 20,
 25, 25, 25,  30, 30, 30,  35, 35, 35,  40, 40, 40,  45, 45, 45,
 50, 50, 50,  55, 55, 55,  60, 60, 60,  65, 65, 65,  70, 70, 70,
 75, 75, 75,  80, 80, 80,  85, 85, 85,  90, 90, 90,  95, 95, 95,

100,100,100, 100, 95, 95, 100, 90, 90, 100, 85, 85, 100, 80, 80,
100, 75, 75, 100, 70, 70, 100, 65, 65, 100, 60, 60, 100, 55, 55,
100, 50, 50, 100, 45, 45, 100, 40, 40, 100, 35, 35, 100, 30, 30,
100, 25, 25, 100, 20, 20, 100, 15, 15)


Tf_AMIP16(1,1,16,0,0,0.1,0.1)
Tf_AMIP17(1,1,17,0,0,0.1,0.1)
Tf_AMIP18(1,1,18,0,0,0.1,0.1)
Tf_AMIP19(1,1,19,0,0,0.1,0.1)
Tf_AMIP20(1,1,20,0,0,0.1,0.1)
Tf_AMIP21(1,1,21,0,0,0.1,0.1)
Tf_AMIP22(1,1,22,0,0,0.1,0.1)
Tf_AMIP23(1,1,23,0,0,0.1,0.1)
Tf_AMIP24(1,1,24,0,0,0.1,0.1)
Tf_AMIP25(1,1,25,0,0,0.1,0.1)
Tf_AMIP26(1,1,26,0,0,0.1,0.1)
Tf_AMIP27(1,1,27,0,0,0.1,0.1)
Tf_AMIP28(1,1,28,0,0,0.1,0.1)
Tf_AMIP29(1,1,29,0,0,0.1,0.1)
Tf_AMIP30(1,1,30,0,0,0.1,0.1)
Tf_AMIP31(1,1,31,0,0,0.1,0.1)
```

```
Tf_AMIP32(1,1,32,0,0,0.1,0.1)
Tf_AMIP33(1,1,33,0,0,0.1,0.1)
Tf_AMIP34(1,1,34,0,0,0.1,0.1)
Tf_AMIP35(1,1,35,0,0,0.1,0.1)
Tf_AMIP36(1,1,36,0,0,0.1,0.1)
Tf_AMIP241(1,1,241,0,0,0.1,0.1)

P_AMIP(File(p=1,x=0.07,y=0.01,Tt=default,To=default),
source(p=1,x=0.07,y=0.735,Tt=default,To=default),
name(p=1,x=0.07,y=0.72,Tt=default,To=default),
title(p=1,x=0.15,y=0.72,Tt=default,To=default),
units(p=1,x=0.75,y=0.72,Tt=default,To=default),
crdate(p=1,x=0.75,y=0.01,Tt=default,To=default),
crtime(p=1,x=0.85,y=0.01,Tt=default,To=default),
comment#1(p=1,x=0.1 ,y=0.72,Tt=default,To=default),
comment#2(p=1,x=0.1 ,y=0.73,Tt=default,To=default),
comment#3(p=1,x=0.1 ,y=0.74,Tt=default,To=default),
comment#4(p=1,x=0.1 ,y=0.75,Tt=default,To=default),

xvalue(p=1,x=0.47,y=0.705,Th=AMIP,Tt=default,To=default),
yvalue(p=1,x=0.67,y=0.705,Th=AMIP,Tt=default,To=default),
zvalue(p=1,x=0.75,y=0.68,Th=AMIP,Tt=default,To=default),
tvalue(p=1,x=0.27,y=0.705,Th=AMIP,Tt=default,To=default),
mean(p=1,x=0.07 ,y=0.68,Th=AMIP,Tt=default,To=default),
max(p=1,x=0.27 ,y=0.68,Th=AMIP,Tt=default,To=default),
min(p=1,x=0.47 ,y=0.68,Th=AMIP,Tt=default,To=default),

xtic#1(p=1,y1=0.2,y2=0.19,Tl=default),
xtic#2(p=1,y1=0.65,y2=0.66,Tl=default),
xmintic#a(p=1,y1=0.2,y2=.195,Tl=default),
xmintic#b(p=1,y1=0.65,y2=.655,Tl=default),
ytic#1(p=1,x1=0.07,x2=0.06,Tl=default),
ytic#2(p=1,x1=0.97 ,x2=0.98,Tl=default),
ymintic#a(p=1,x1=0.07,x2=0.065,Tl=default),
ymintic#b(p=1,x1=0.97,x2=0.975,Tl=default),
xlabel#1(p=1,y=0.183,Tt=default,To=defcenter),
ylabel#1(p=1,x=0.055,Tt=default,To=defright),

box#1(p=1,x1=0.07,y1=0.2 ,x2=0.97,y2=0.65,Tl=default),

legend(p=1,x1=0.07,y1=0.1,x2=0.97,y2=0.12,Tt=default,
To=defcenter,Tl=default),
data(p=1,x1=0.07,y1=0.2,x2=0.97 ,y2=0.65)
)

P_AMIPDUD(data(p=1,x1=0.07,y1=0.2,x2=0.97 ,y2=0.65))

Gfi_AMIP_psl(project=linear,xticlabels#1=lon30,xticlabels#2=lon30,
  xmtics#1=lon5,xmtics#2=lon5,yticlabels#1=lat20,yticlabels#2=lat20,
  ymtics#1=lat5,ymtics#2=lat5,datawc=(-180,-90,180,90),
range=
(id=1,level1=-1E20,level2=970,Tf=AMIP30)
(id=2,level1=970,level2=975,Tf=AMIP29)
(id=3,level1=975,level2=980,Tf=AMIP28)
(id=4,level1=980,level2=985,Tf=AMIP27)
(id=5,level1=985,level2=990,Tf=AMIP26)
(id=6,level1=990,level2=995,Tf=AMIP25)
(id=7,level1=995,level2=1000,Tf=AMIP24)
(id=8,level1=1000,level2=1005,Tf=AMIP23)
(id=9,level1=1005,level2=1010,Tf=AMIP22)
(id=10,level1=1010,level2=1015,Tf=AMIP21)
(id=11,level1=1015,level2=1020,Tf=AMIP20)
(id=12,level1=1020,level2=1025,Tf=AMIP19)
(id=13,level1=1025,level2=1030,Tf=AMIP18)
(id=14,level1=1030,level2=1035,Tf=AMIP17)
(id=15,level1=1035,level2=1040,Tf=AMIP35)
(id=16,level1=1040,level2=1E20,Tf=AMIP36)
)

color(AMIP)

canvas(open)

A_psl(Name='psl',XName='longitude',YName='latitude',ZName='time',
File='/amipsp/drs/psl/psl_ecm.dic',zfirst=120,zlast=120)

D_d1(type=isofill,priority=2,graph=AMIP_psl,template=AMIP,a=psl)
D_d2(type=continents,priority=4,graph=default,template=AMIPDUD)
```

The 120th month of a three-dimensional variable defined by an array data attribute set named, "A_psl," is displayed with overlaid continents in Figure 42. Two pictures (defined by: "D_d1" and "D_d2") make up the page that is displayed. If the initial.attributes file exists, then the only part of the script in Figure 41 that is necessary is that script shown in Figure 43.

Figure 42. Example Display



Visualization and Computation System (VCS)

ECMWF T42L19 AMIP 10 Year Simulation ( 1979-1988 )
psl Monthly Mean Sea Level Pressure hPa

Mean 1012.37 Max 1040.05 Min 972.04 Dec/1988

/amipsp/drs/psl/psl_ecm.dic 7/12/93 15:27:51

Figure 43. Reduced Example Script

```
color(AMIP)

canvas(open)

A_psl(Name=’psl’,XName=’longitude’,YName=’latitude’,ZName=’time’,
File=’/amipsp/drs/psl/psl_ecm.dic’,zfirst=120,zlast=120)

D_d1(type=isofill,priority=2,graph=AMIP_psl,template=AMIP,a=psl)
D_d2(type=continents,priority=4,graph=default,template=AMIPDUD)
```

## 5. Conclusions and Future Work

This software provides a workbench, with an ever-growing menu of methods for both visualization and computation, to support those capabilities that are necessary for analyzing and comparing climate model behavior. It provides both an interactive interface and a script-driven interface, as well as a method for saving a script of an interactive session for replay, and a method for saving a script of the instantaneous state-of-the-system for recovery to that instant. It provides the user with procedures for modifying the most basic elements of a picture, and seeing those changes take effect almost immediately. It also provides procedures for producing animation and hardcopy.

At this point, it is appropriate to release the software for extensive testing and for evaluation of the computer–human interface. Development and inclusion of new visualization methods and computations will continue.

In the near future, some refinements and developments are obviously needed. For example, more interactive interfaces are required, and some of the current keyboard-intensive interfaces will either be replaced with mouse-driven interfaces, or both keyboard and mouse-driven interfaces will be provided. Two-dimensional visualization of data vectors of U(x,y) vs. V(x,y), color-filled grid box images of A(x,y), and scatter plots for A(x,y) vs. B(x,y) are also needed. Graphic projections will be implemented for a Robinson projection, polar stereographic, log-log, and semi-log. One-dimensional plots of A(x) vs. x, A(x) vs. B(x), and histograms are needed. Finally, this software will be interfaced to the Climate Data Management System (CDMS), which is currently under development in PCMDI.

Other possible developments for the future include possibly changing the underlying graphics library to PEX or Open GL, or to an accepted standard, to aid in the development of three-dimensional graphics. As more support for object-oriented development becomes available, this software may be converted to an object-oriented language and structure. We will take advantage of developments in distributed computing capabilities; and we will update the interactive interfaces as X changes or is replaced.

# PCMDI REPORTS

| Number | Title | Author(s) | Date |
|---|---|---|---|
| 1 | The Validation of Atmospheric Model | W. L. Gates | March 1992 |
| 2 | Analysis of the Temporal Behavior of Tropical Convection in the ECMWF Model | J. M. Slingo<br>K. R. Sperber<br>J.-J. Morcrette<br>G. L. Potter | April 1992 |
| 3 | The Effect of Horizontal Resolution on Ocean Surface Heat Fluxes in the ECMWF Model | P. J. Gleckler<br>K. E. Taylor | July 1992 |
| 4 | Behavior of an Ocean General Circulation Model at Four Different Horizontal Resolutions | C. Covey | August 1992 |
| 5 | The Effects of Sampling Frequency on the Climate Statistics of the ECMWF General Circulation Model | T. J. Phillips<br>W. L. Gates<br>K. Arpe | September 1992 |
| 6 | Sensitivity of Dynamical Quantities to Horizontal Resolution in a Climate Simulation with the ECMWF Atmospheric General Circulation Model (Cycle 33) | J. S. Boyle | October 1992 |
| 7 | AMIP: The Atmospheric Model Intercomparison Project | W. L. Gates | December 1992 |
| 8 | The Impact of Horizontal Resolution on Moist Processes in the ECMWF Model | T. J. Phillips<br>L. C. Corsetti<br>S. L. Grotch | January 1993 |

| Number | Title | Author(s) | Date |
|---|---|---|---|
| 9 | A Modeling Perspective on Cloud Radiative Forcing | G. L. Potter<br>J. M. Slingo<br>J.-J. Morcrette<br>L. Corsetti | February 1993 |
| 10 | The Use of General Circulation Models in Detecting Climate Change Induced By Greenhouse Gases | B. D. Santer<br>U. Cubasch<br>U. Mikolajewicz<br>G. Hegerl | March 1993 |
| 11 | Preliminary Validation of the Low Frequency Variability of Tropospheric Temperature and Circulation Simulated for the AMIP by the ECMWF Model | J. S. Boyle | April 1993 |
| 12 | Simulation of the Indian and East-Asian Summer Monsoon in the ECMWF Model: Sensitivity to Horizontal Resolution | K. R. Sperber<br>S. Hameed<br>G. L. Potter<br>J. S. Boyle | November 1993 |
| 13 | Statistical Intercomparison of Global Climate Models: A Common Principal Component Approach | S. K. Sengupta<br>J. S. Boyle | November 1993 |
| 14 | Ocean Variability and Its Influence on the Detectability of Greenhouse Warming Signals | B. D. Santer<br>U. Mikolajewicz<br>W. Brüggemann<br>U. Cubasch<br>K. Hasselmann<br>H. Höck<br>E. Maier-Reimer<br>T. M. L. Wigley | January 1994 |

| Number | Title | Author(s) | Date |
|---|---|---|---|
| 15 | Cloud-Radiative Effects on Implied Oceanic Energy Transports as Simulated by Atmospheric General Circulation Models | P. J. Gleckler<br>D. A. Randall<br>G. Boer<br>R. Colman<br>M. Dix<br>V. Galin<br>M. Helfand<br>J. Kiehl<br>A. Kitoh<br>W. Lau<br>X.-Z. Liang<br>V. Lykossov<br>B. McAvaney<br>K. Miyakoda<br>S. Planton | March 1994 |
| 16 | DRS User's Guide | R. Drach<br>R. Mobley | March 1994 |