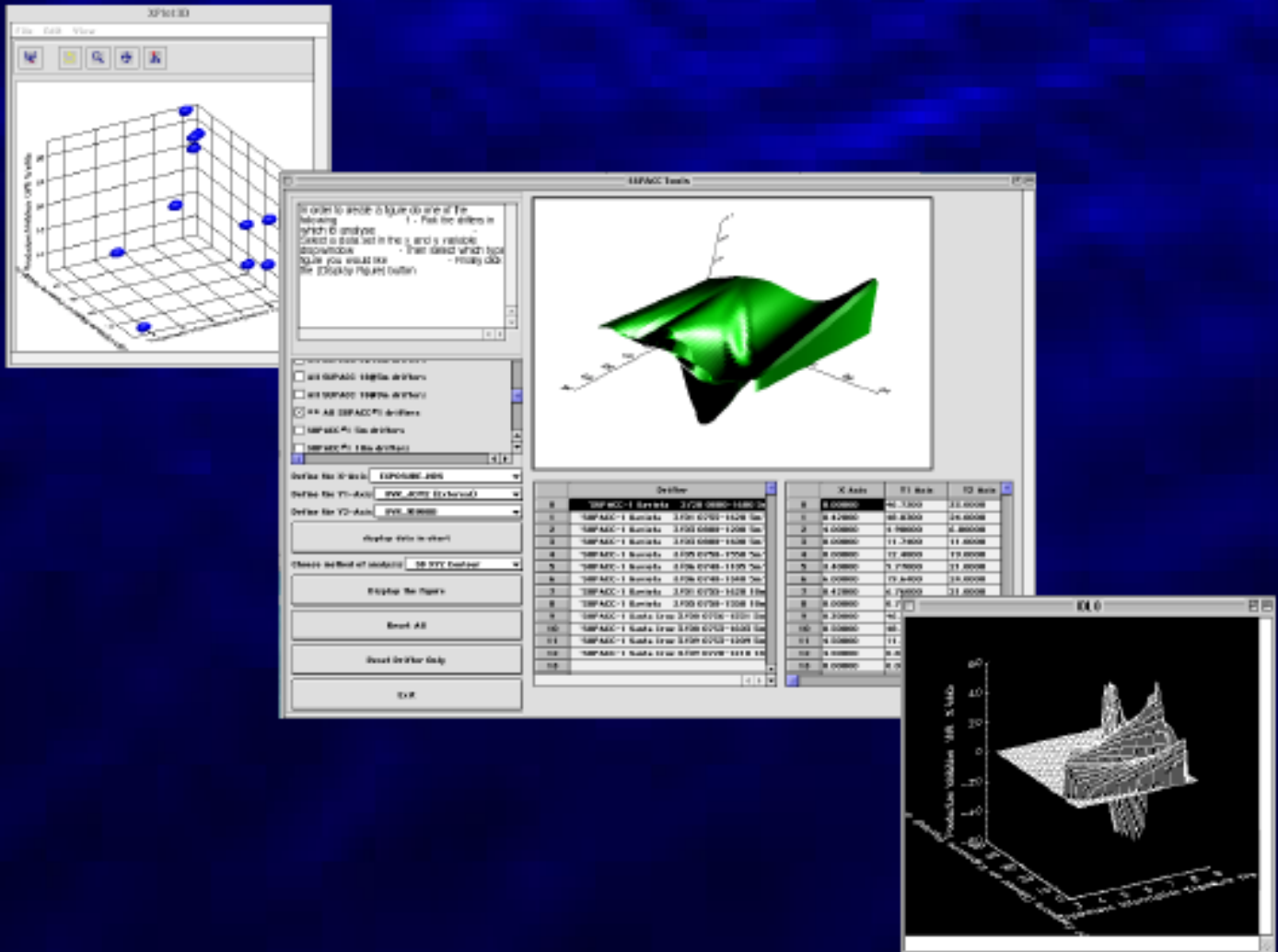


"REAP"

Ongoing Development of IDL-Based Relationship Evaluator and Plotter (REAP) software

Incorporating a suite of data management, computational and visualization tools to recognize scientific patterns in the data inputs and outputs of SLIC, BWF ARCHITECT, and BEAR



University of California, Santa Barbara
Marine Primary Production Group
Technical Report 4 (June 28, 2002)

Steve McKagan and Barbara Prézelin

Foreword and Acknowledgements

This technical report is one in a series of four companion reports that document computer software code development between 1999-2002. The complete set of computer software codes was required for advanced computational and visualization analyses of oceanographic field data collected by UCSB's Marine Primary Production Group (MPPG). **Figure 1** illustrates how the four codes were used in tandem to process raw *in situ* light and primary production data to numerous mid and final output products required to understand and predict the influence of solar ultraviolet radiation on marine primary production and ecosystem dynamics. Each report includes a CD containing a working code, relevant input data from field studies, and a copy of the report. *The research and code development is supported by federal and university grants to B. B. Prézelin (NSF OCE96-18390, NSF OCE00-74422, UCSB Chancellor Research Award).*

The field campaigns that prompted the development of the four codes were the MPPG's SUPACC (Solar Ultraviolet Production Algorithms for Coastal California) cruises conducted within the Santa Barbara Channel and at the continental shelf break on the seaward side of Santa Cruz Island during February to March, June, August to September, and December 1998. Specially designed experimental drifters and modified light instrumentation were deployed during each cruise in order to quantify the spectral sensitivity of phytoplankton primary production, community assemblages and biogeochemistry to solar ultraviolet radiation (295 nm to 400 nm in nature) penetrating to depth in the ocean. The principal scientific objective of SUPACC is to determine, for the first time, spectral algorithms (also known as BWFs - biological weighting functions) quantifying the spectral sensitivity of marine primary productivity along coastal California to natural fluctuations in solar ultraviolet radiation. A second objective is to determine the patterns and sources of variability in these BWFs for highly diverse phytoplankton communities within these coastal waters. A third objective is to develop reasonably accurate, predictive, and broadly applicable approaches that allow for the bio-optical mechanistic modeling of solar UV effects on marine primary production.

Such accomplishments require a level of rigorous and thorough analyses not attempted before. The resulting environmental data set from the four cruises is large, the quality assurance requirements have been met and data is presently archived in a format suitable for advance computational analyses and multidimensional visualization. The four computer software codes are original and were developed as no existing software was available to meet our specialized requirements.

MPPG Technical Report 1 documents the development of **SLIC**, a Visual Basic code for generating fully spectral UV (290-400 nm) underwater light (UWL) fields from *in situ* PUV (Profiling UltraViolet - BioSpherical Inc.) profiling at four UV wavebands. The acronym originally came from SUPACC Light In-water Calculations; it is now interpreted as Solar Light In-water Calculations. The current research version of the software, **SLIC(uv) v1**, is documented here; a production version of the code, **SLIC (uv) v2**, with reduced flexibility but increased stability, is in preparation.

SLIC (uv) v.1 (from hereon referred to as **SLIC**) is an original research code designed specifically for the determination of *in situ* spectral UV radiation and integrated Photosynthetically Available Radiation (PAR, 400-700 nm) exposures received by phytoplankton samples incubated on SUPACC drifters where spectral light treatments are further modified by the use of various UV cutoff filters. Additional **SLIC** features allow for comparison of UWL fields for a variety of conditions and/or relevant hydrodynamic water column characteristics. Such comparisons are helpful in estimating the likely UV and PAR light field paths a phytoplankton might experience in the water column over a day due to vertical mixing. **SLIC** may be of general use to a wide range of biological and chemical oceanographers requiring an accurate but relatively straightforward field approach to determining *in situ* UV spectral irradiances. *The concept, development and testing of this code benefited from ongoing consultation with Dr. Sasha Madronich, Atmospheric Science Division, and National Center for Atmospheric Research.*

*MPPG Technical Report 2 documents the development of SUPACC's **BWF Architect**, an IDL-based program that combines field measurements of the spectral UVR inhibition of *in situ* primary production with **SLIC** determinations of in-water spectral UVR exposures in order to derive biological weighting functions (BWFs) that quantify and can be used to predict UVR effects over a range of environmental circumstances. The most important of these circumstances is the impact of changing UV-B (290-320nm in nature) climatology where UV-B radiation reaching the earth surface increases due to depletion of stratospheric ozone. *The conceptual and technical development of BWF Architect was exceptionally challenging, and the final product would not have been possible without advice generously and patiently provided by Dr. Sasha Madronich, Atmospheric Science Division, and National Center for Atmospheric Research.**

*MPPG Technical Report 3 documents the development of SUPACC's **BWF Error and RAF (BEAR)** code, an IDL-based program that recalls replicate BWF determinations from **BWF Architect** and allows for the computation and viewing of an averaged BWF and its error estimates as well as for comparison with outcomes from different field experiments. For each BWF, **BEAR** calculates the apparent Radiation Amplification Factor (RAF) that is the commonly used measure of the absolute dependency of a biological or chemical process on changes in atmospheric ozone concentrations (Madronich *et al.*, 1995). In addition, **BEAR** calculates a variation of the RAF so that it is "ozone-weighted", which appears especially useful for calculating the ozone sensitivity of a process when the usual fitting functions for a BWF fail due to additional but independent processes being induced at longer wavelengths of UV radiation (Prézelin *et al.*, 1997). The RAF outputted by BEAR is our most highly derived and climatologically valued parameter. It is the end product of the initial input and subsequent processing of field light and productivity data through **SLIC**, **BWF Architect**, and **BEAR**. *We acknowledge gratefully the guidance of Dr. Sasha Madronich, Atmospheric Science Division, and National Center for Atmospheric Research in aspects of the development of BEAR.**

MPPG Technical Report 4 documents the ongoing development of SUPACC's Relationship Evaluator and Plotter (**REAP**) software which incorporates a suite of data management, computational and visualization tools to recognize scientific patterns in the data inputs and outputs of **SLIC**, **BWF ARCHITECT**, and **BEAR**. For these purposes, appropriate data sets and derived parameters are converted to an IDL-format on a master data display for the quick retrieval and processing of IDL-based assessments of interdependent and possibly multidimensional relationships.

References:

Madronich, S., McKenzie, R. L., Caldwell, M.M. and L. O. Bjorn (1995) Changes in ultraviolet radiation reaching the Earth's surface. *AMBIO*, 24, 143-152.

Prezelin, B. B., Madronich, S., Booth, R. and A. Matlick (1997) Natural variability in biological weighting functions.

Referral

For questions, comments and requests for additional information please contact the corresponding technical report authors who are part of the Marine Primary Production Group within the Marine Science Institute at the University of California, Santa Barbara.

Jordan Hastings (jordan@geog.ucsb.edu) - Tech Report 1.

Steve McKagan (smckagan@hotmail.com) - Tech Reports 1,2,3, and 4.

Barbara Prézelin (prezelin@lifesci.ucsb.edu) - Tech Reports 1,2,3, and 4.

(805) 893-4319 (Lab)

(805) 893-4724 (Fax)

http://www.lifesci.ucsb.edu/eemb/labs/prezelin/Sites/MPPG/MPPG_Home.html

Figure 1

SUPACC DATA DEVELOPMENT FLOWCHART

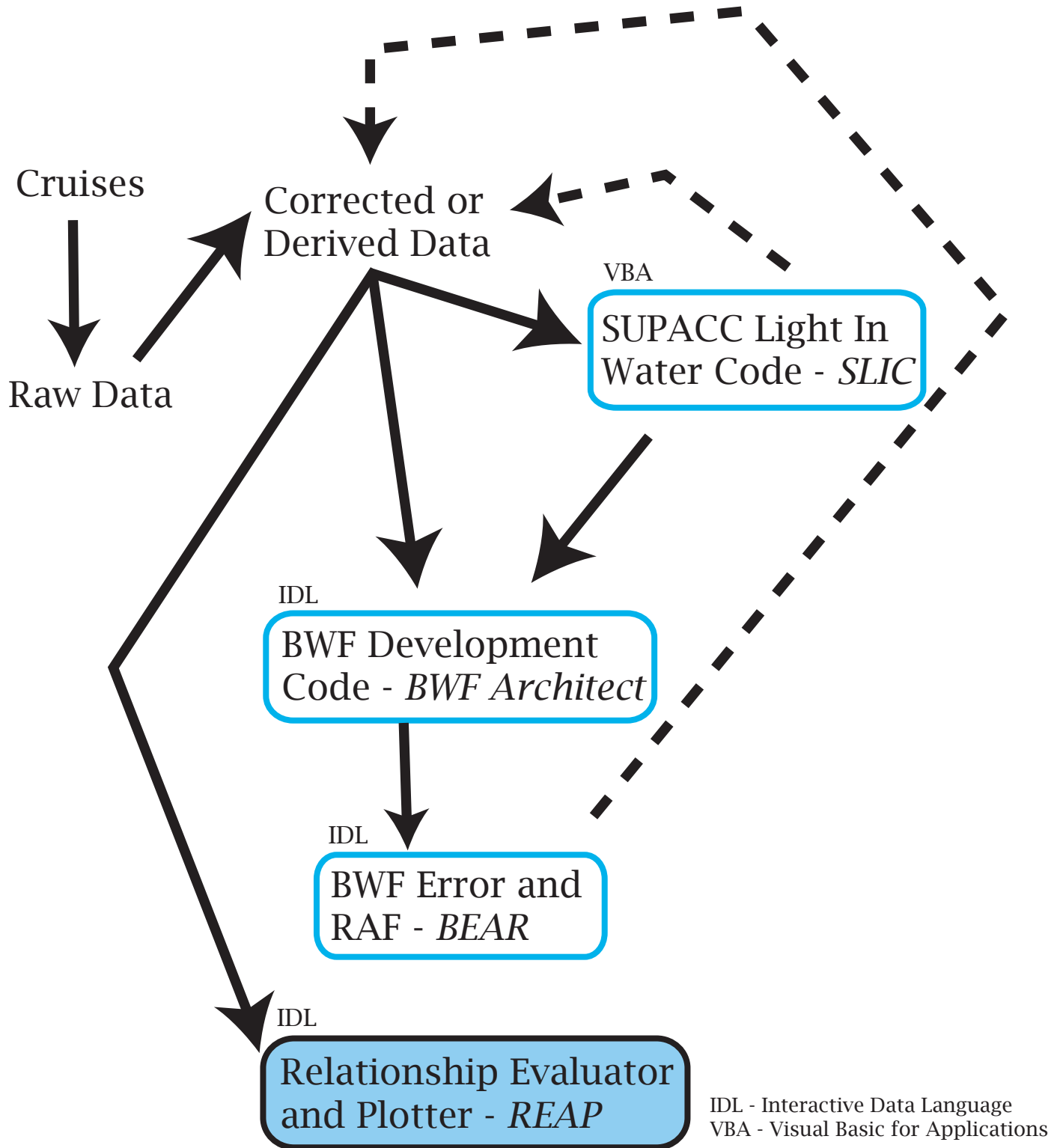


Table of Contents

<i>Foreword and Acknowledgments</i>	<i>i-iv</i>
I. Introduction	1
Introduction Figures	3
II. Computational and Visualization Goals	4
III. Data Inputs	5
IV. Understanding the Program	8
Understanding the Program Figures	15
V. User Manual	17
VI. Example Outputs	22
Example Outputs Figures	23
VII. Citations where BEAR was employed	25
VIII. References	26
IX. BEAR software Code	27

I. Introduction

The **SUPACC Relationship Evaluator and Plotter (REAP)** is a robust data analysis program designed specifically for the exploration and organization of data collected on the **SUPACC** cruises. This set of tools allows the user to cross reference any of the parameters collected on the **SUPACC** cruises or any parameters developed from cruise data. The program was developed as a replacement for traditional spreadsheet based analysis tools, which have limited graphics capabilities and are difficult to work with when using large data sets. The **REAP** program is set up so that the user may choose any three data products and derived parameters to analyze for interrelationships. The user may choose to do the analysis for all or distinct subsets of the drifter experiments completed for all four **SUPACC** cruises. There is also a choice of graphical display in 2D and 3D, and new choices are being added as they are identified. The data is also displayed in a table that can be manipulated by the user so that the new values will feed back into the graphics.

The ability to choose from a wide array of parameters and to quickly evaluate their relationship enables us to use visualization as an important part of efforts to identify key influences and driving forces behind seasonal, hydrographic, community structure, spectral UV and PAR, and other variations on the patterns and sources of variation in the patterns of solar UV inhibition of *in situ* marine primary production. Unlike, **BWF Architect**, **BEAR** and **SLIC** (Hastings et.al., Technical Report 1, 2002, McKagan et.al., Technical Reports 2 and 3, 2002), **REAP** is not designed with a specific result in mind, instead it is aimed at providing an open ended and flexible suite of analysis options.

The ability to quickly and aesthetically cross reference all of the available light information, production information, pigment data, water body characteristics and BWF/RAF findings will prove invaluable for multi dimensional analysis. What would otherwise have taken hours of data formatting and graphical parameterization using other tools will now be performed with only a few clicks of the mouse. Once the graphic displays are created it is up to the user to note any relationships and/or save any of the findings via screenshots. At present **REAP** does not perform any additional calculations nor provide any specific output files. Instead, all of the code is devoted to organization of the **SUPACC** data and creating the graphics interfaces.

II. Computational and Visualization Goals

The goal of the **SUPACC** Relationship Evaluator and Plotter (**REAP**) software is to provide a user friendly data management tool in which the user can graphically cross reference any combination of data from the SUPACC cruises.

More specifically, the **REAP** program is designed to:

- A. Act as an easily accessible storage format for ALL of the **SUPACC** data, thereby turning IDL into the official data analysis tool for the **SUPACC** cruises. This is especially important because the **SUPACC** data set marks the first complete implementation of IDL as the primary research and analysis program for the Marine Primary Production Group.
- B. Provide a means by which the physical, chemical and biological properties of the ocean, of the test organisms and of the light field can be investigated both separately and together. The program is designed to allow the user to explore the relationship between any of these parameters for parts of a single cruise, for the entire suite of cruises or any combination therein.
- C. Generate a 3-Dimensional graphics platform so that historically 2-Dimensional data sets can not only be compared over the top of one another, but actually as a 3-Dimensional surface with potentially meaningful peaks and valleys.

III. Data Inputs

The entire **REAP** program is built around having a huge array of data inputs. The inputs are taken from a large Excel™ worksheet, which contains all of the **SUPACC** data compiled to date. The data is not called directly from the Excel spreadsheet (though this may change in the future), but rather each parameter is assigned its own file where the data is saved in a SimpleText™ format. Thus, each of the parameters that can be chosen for analysis within the **REAP** program come from a separate data file and since there are already over 100 parameters available in the **SUPACC** data compilations spreadsheet that means there are over 100 separate files awaiting **REAPs** call. The following is a list of major categories which are currently available for analysis within **REAP**, most of these categories contain several elements:

External Light Exposure	Enhancement Effects
295 Treatment Exposure	PAR Error Estimates
310 Treatment Exposure	T-Corrected Production
318 Treatment Exposure	T-NON-Corrected Production
328 Treatment Exposure	T-Corrected Chlorophyll
342 Treatment Exposure	T-NON-Corrected Chlorophyll
381 Treatment Exposure	T-Corrected Production per Chlorophyll
410 Treatment Exposure	T-NON-Corrected Production per Chlorophyll

Each of the SimpleText input files contains a string of data (or vector), in which there is one value for each of the drifters cast during all four of the **SUPACC** cruises. The data sets are organized by date such that the first number in every data set correlates to the first data collection for the initial drifter cast in **SUPACC** #1 and the last data point correlates to the final drifter cast in **SUPACC** #4. Each of the data files starts off with a title that explains which variable is listed in the vector, and what units they are listed in (**Figure 2**).

For more information about how each of the parameters called in the **REAP** program were developed please refer to the appropriate Technical Report or contact the Marine Primary Production Group as listed in the foreword.

Figure 2

An example Data output file containing 1 column where each value or lack thereof (i.e. NAN) correlates to the Initial measured Chlorophyll in ng/L . This data set shows only data for SUPACC 1 and 2. SUPACC 3 and 4 data would follow directly below without any spaces, but have been removed from this figure in the interest of space.

Chlorophyll Summary

Initial

ng/L

1044.7

713.9

1503.3

1579.1

2766.3

2088.9

2077.1

785.4

2714.6

702.2

757.5

533.9

691.2

1132.3

948.7

274.6

566.2

255.5

307.4

1012.0

211.1

211.1

NAN

NAN

56.1

50.1

IV. Understanding the Program

In this section we will explain more specifically what the **REAP** code does and how it makes certain analysis possible.

1. Core Structure: Graphical User Interface (GUI)

The foundation of the code is the Graphical User Interface (GUI). When the program is started, a window pops up (GUI) and provides the user with a way to make decisions and changes to a variety of parameters and operational settings within the program (**Figure 3**). These changes are made by pointing and clicking on various options with the mouse. The portion of code, which drives the user interface, acts as a central hub. Whenever the user changes a setting in the GUI the main program reaches out to a sub program which explains what the change means and passes the new information back to the central hub of the program. Information that is passed to and from the hub can be identified in the code by the (*pstate) prefix, which is used as a marker for all information stored in the central hub also used in the sub routines.

The GUI portion of the code is the central hub of the software and drives the user interface activities. Whenever the user changes a setting in the GUI and event occurs within the code, and the main program reaches out to a subroutine, which explains what change was made and how it will affect the curve fitting procedure. There is a subroutine within the program for each parameter in the GUI as well as a large subroutine, which drives the output graphics.

The first sub routine listed is the make grid function which defines the procedure for converting X, Y, and Z data points into a 3-D image. This function is used in several of the sub programs that follow and thus is listed first so it can be called with just its title in the programs below. The next sub routine defines the graphic options and below that is a long series of routines which tell the hub which parameters and which drifters the user chose for this particular analysis. Thus, the program gathers all of the information from the subroutines in the middle of the program and feeds them into the HUB at the very bottom. The HUB then organizes all of the parameters and sends them to the make grid function and plotting routine at the top of the code where the results are graphed (**Figure 4**).

2. The Make Grid Function

The first code listed in the SUPACC tools program is the make grid function. This function will be recalled in various places throughout the code and therefore is defined at the top of the program. This section of code takes advantage of two built in IDL routines, *triangulate* and *trigrd*. These are the two routines that make it possible to convert regular X, Y, and Z data into a 3-D image. A detailed description of how these routines work can be found within the IDL online help files and in the Adobe Acrobat™ reference files which were installed in folders within IDL itself.

3. The Plot It (Graphing) Event

The Plot It event is the most complicated and most important section of the entire code. In this section all of the choices made by the user about which drifters and which parameters to explore are fed into the desired graphing option. This portion of the code can be broken down into several parts including data definition, determination of axis and range, and five different plotting options.

I. Data Definition

In this phase the data requested by the user is brought into the Plot It event and identified as X, Y or Z data. All three data sets then undergo a slight manipulation that replaces all zero values which will not show up in 3-D with a .00001 value that plots as near zero as is necessary given our typical data ranges. If the X, Y and Z data are all zero or NAN for a specific drifter then that point is removed from the graph completely. The data that remains after these initial corrections is called 'realdata' within the code.

II. Define Axis and Ranges

The second phase of the Plot It event defines the labels which match up to the X, Y and Z data sets defined by the user. The data ranges are also developed here for each of the axis and then converted into ranges which will be used for the view plane in which the figure will be rendered. The symbols and tick lengths to be used on the axis are also defined in this portion of the code.

III. Standard X Y Data Plot

When the X Y graphic option is chosen by the user the Plot It sub routine will take all of the information supplied above and then move to this section of the code for information of graph specifics. If any other graphic option is chosen the program will skip this section of code and move on to the appropriate graphics option.

The X Y data plot code begins with a further definition of the X and Y data ranges. This may seem redundant from the step that took place second in this list, but it is necessary because each of the graphics types look better under slightly different ranges, especially when we move between 2-D and 3-D. Once the ranges are established the built in IDL object graphics routines are employed. Here the curve or curves, titles and axis are all added to the window that will render the image. Once all of the pieces have been added the image is drawn.

IV. X Y Data plot with Two Y's

As above, this section of code will not be addressed unless the user has chosen the option to graph multiple Y data sets against a single X axis. The code starts with a set of conversions, which adjust the second set of Y data so that it can fit in the same viewing area as the initial Y data set. The second Y-axis is also defined here, though only represented by the minimum, middle and maximum of the data set. Once the data is converted and the axis defined the object graphics routines are employed and the figures, titles and axis are all added to the viewing window and rendered.

V. X Y Data plot with Staggered Y's

This section of code is much like the section above with one fundamental difference. The converted data, secondary Y-axis and title are all shifted up above the initial Y data axis and title such that they don't overlap.

VI. Plot 3-D Scatter

When the 3-D scatter plot option is chosen the Plot It routine jumps to this portion of the code and begins by defining the size of the spheres which will be plotted. The size will appear the same on most graphs because it is set as a ratio of the data ranges. The code then calls upon the XPLOT3D sub routine, which is built into IDL. This routine pulls open a separate window and plots the X, Y and Z data. The new window has several options which can be used to examine the scattered data more closely by changing the reference angle and observing the points in rotation.

VII. Plot 3-D Contour and Scatter

When the 3-D contour plot option is chosen the program automatically creates the 3-D scatter plot as defined above. The scatter plot and contour plot are both utilized so that the user can better understand which data points are driving the shape observed in the contour plot.

The code driving the scatter plot creation is listed after the much larger portion of code which creates the contour plot. The contour plot creation starts with the conversion of the X, Y and Z data to the normalized coordinate system. Then the new normalized data is converted into a complete 3-D shape with the built in sub routines trigrid and triangulate, this shape will replace the sample image that has been sitting in the large view window in the GUI. Next the structure for the X and Y-axis is established and the data and axis are plotted using the built in IDL SURFACE sub routine. Once the SURFACE is rendered the scatter plot is created using XPLOT3D.

More Graphics Options To Come

The graphics options listed above are nearly complete. Certain portions of the 3-D contour plotting are still being tested and may lead to new and more sophisticated graphics options. The **SUPACC** tools program is designed with flexibility so that it can

grow and change to match the users needs. Therefore, it would be inappropriate to think of it as anything less than a work in progress.

4. User Choice Sub Routines

The next several pieces of the **REAP** program are used in order to update optional parameters that the user can change. Each time a parameter is changed by the user it triggers an update within that program which is called an event. The relationship between the user choice sub routines and other elements of the **REAP** program can be explored visually in the **REAP** program flow chart (**Figure 4**). Each of the potential events is listed below along with a brief explanation of what influences the change.

i. Plot Tools Event

The Plot tools event feeds back information to the central hub about which figure type the user would like to see graphed.

ii. X-Display Event

The X-Display event pulls the relevant information which the user has chosen to represent the X data set, and feeds it back to the central hub.

iii. Y-Display Event

The Y-Display event pulls the relevant information which the user has chosen to represent the Y data set, and feeds it back to the central hub.

iv. Z-Display Event

The Z-Display event pulls the relevant information which the user has chosen to represent the Z data set, and feeds it back to the central hub.

v. Table Edit Event

The table edit event takes the information about the X, Y and Z data series that has been updated by the user through changes in the GUI table, and feeds the new values back into the central hub.

vi. Test Trackball Event

The Test Trackball event incorporates several built in IDL subroutines in order to create a freely rotating 3-D image. The 'transform' option is employed in order to update the program about changes in the location of the object so that it can be rendered again in its new location. Without this section of the code the image in the main window would not be visible and could not rotate.

vii. Chart It Event

The chart it event, like the X, Y and Z-Display events, is designed to pull out the relevant information chosen by the user pertaining to which drifters were chosen, and feed that information back to the main GUI. This information is then outputted as text into the main GUI table allowing the user to observe the X, Y and Z data sets in light of which drifter they correspond to.

viii. Drifter Events

There are a large variety of potential drifter combinations available for the user to choose. These combinations are listed within the code in the same order that they are shown in the drop down window of the GUI. Because the drifters are sorted in a variety of ways each of the drifters was given a set of defining parameters within the codes representing the depth, time and location of each. The drifter events portion of the code converts the set of drifters requested by the user into these defining parameters and sorts out the drifters which match. Once the desired drifter series is chosen, that information is then

passed to the X, Y and Z data calling section so that only the appropriate numbers from each data vector or incorporated into the graphics phase.

5. Clearing Events

i. Reset Treatment Event

The Reset Treatment event occurs whenever the 'Reset' button is pushed. This option destroys the current GUI and all of its contents and restarts the **REAP** program allowing the user to have a fresh slate.

ii. Reset Drifter Event

The Reset Drifter event occurs whenever the 'Reset Drifter' button is pushed. This option clears the set of drifters that were chosen for the analysis from memory but maintains the X, Y and Z details. This way the user can look at the same data for a variety of drifter combinations without having to restart the program.

iii. Exit Event

The Exit Event is tied to the 'Exit' button on the GUI. When this button is clicked the program is terminated.

FIGURE 3

Diagram of REAP GUI with instructions and user information.

A. This is the window where all contour 3-D images are plotted. Note-use mouse to rotate images.

B. This box explains how to run this GUI in several basic steps.

C. In this window the user must select the box which correlates with the series of drifters that he/she would like to use for this analysis.

D. This graphic appears when the 3-D option is selected, showing the points in space without the contouring

E. These drop down windows allow the user to change the X, Y and Z variable and the button will display them in the table.

F. In this drop down window the user must select which type of graph to use.

G. Push this button when ready to see the images.

H. Push this button in order to clear all of the entered information and try a new analysis.

I. Push this button in order to store the selected X, Y, Z data and graph type, in order to look at different drifter combinations.

J. Push this button to exit the program.

K. The drifter information and the X, Y and Z data will be displayed in this table.

L. This graphic appears when the 3-D option is selected, showing the latticed image. Note- this image will not rotate.

M. This is one of three 2-D graphics which are also available in option (F). This Image, like (D) and (L) will appear as its own window.

Table K: Drifter Information

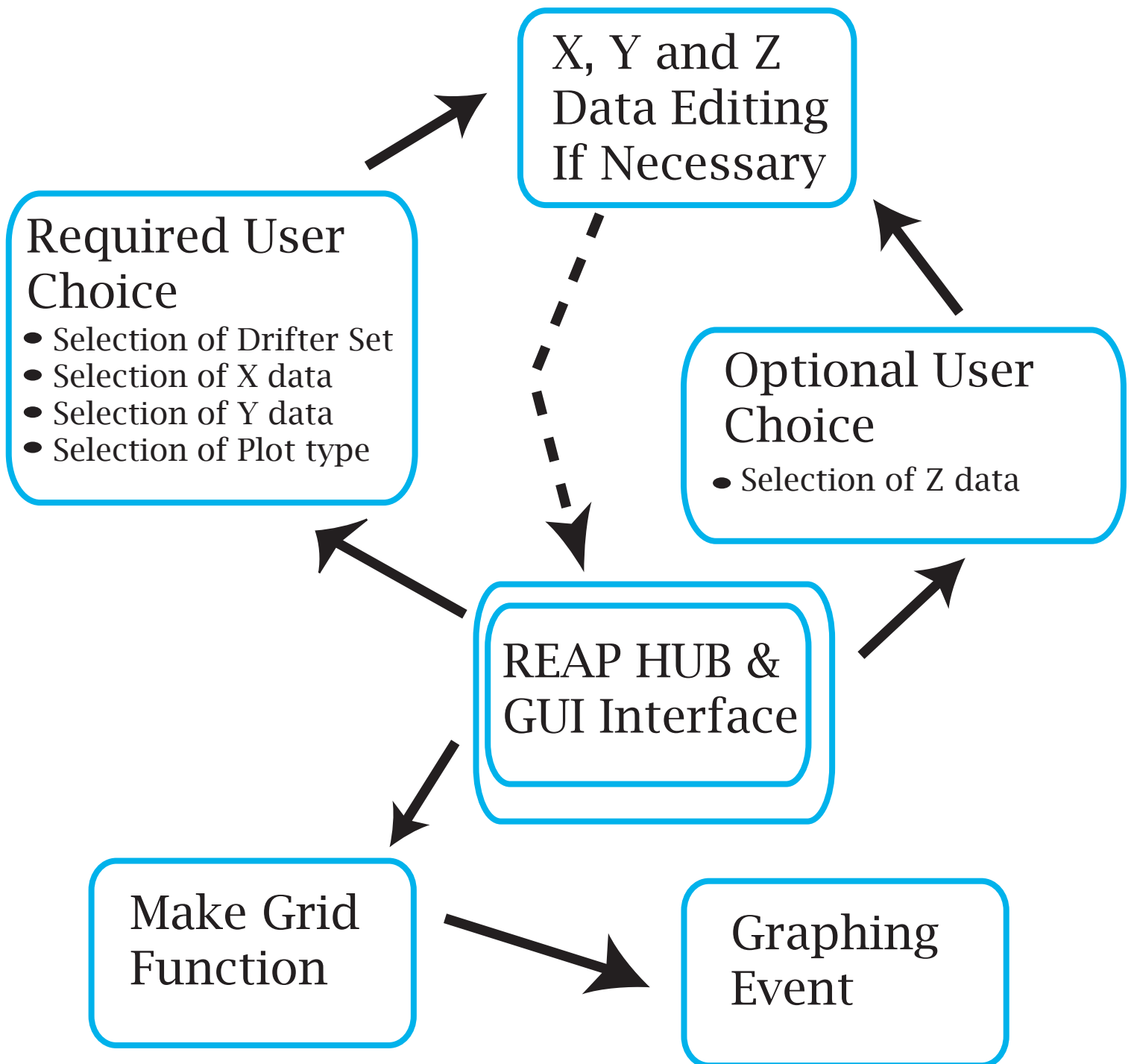
Drifter	X Axis	Y1 Axis	Y2 Axis
0	0.00000	0.00000	0.00000
1	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000
12	0.00000	0.00000	0.00000
13	0.00000	0.00000	0.00000

Table M: 2-D Scatter Plot

External Light Exposure (Based on Exposure Period) UVB vs Treatment Information Exposure hrs

Figure 4

REAP Program Flow Chart



V. Users Manual for REAP

A. System Requirements

The **REAP** program is built using the Interactive Data Language (IDL) program developed by Research Systems Incorporated (RSI). In order to run the program a version of IDL 5.4 or higher must be installed on the Macintosh or PC. IDL can be downloaded and used for free in 7-minute demonstration periods. The **REAP** program can be used swiftly because there are no major calculations taking place, and though perhaps a bit clumsy, the 7-minute period may prove useful at least for preliminary analysis.

B. Installing the Program and all Components

Once IDL has been installed on one of your computers the next step is to pull over a copy of the program. A hard copy of the program can be found at the back of this technical report and electronic versions can be found in the attached CD and, in the future, be downloaded from the Marine Primary Production Group Website at:

http://www.lifesci.ucsb.edu/eemb/labs/prezelin/Sites/MPPG/MPPG_Home.html.

In order to successfully run the program the user must create or duplicate a folder containing a copy of the program and all of the relevant data files called by the program. It is important to note that all of the relevant files must be kept together with the main program or the program will fail to run.

C. Pulling Up the GUI

In order to pull up the Graphical User Interface (GUI) the user needs to open up the **REAP** program in IDL, then compile and run the program. The program can be compiled and run using the drop down windows at the top of the IDL prompt screen, by using the hotkeys, or by typing the name of the program into the command prompt. If the program was imported correctly and all of the relevant data files are present then the GUI should appear providing the user with a series of options for use during graphics

development. **Figure 3** provides a picture of the GUI, which will appear if the program is compiled and run properly.

D. Reading the GUI

The GUI is divided into several sections starting with a small text screen explaining some basics about how to create images, a window containing check boxes which the user can use to define which drifters to run, a series of drop down menus defining which data to use for X, Y and Z, another drop down menu for defining which graph to use, a series of buttons which trigger displays, an editable table where the user can make adjustments and a large window for the graphics outputs. The program is set up so that the user need only point and click in each window or menu or button creating a very simple working environment for the user. The following is a checklist of steps which the user must undergo in order to begin running the **REAP** program.

E. In the First Window below the text box on the left

- Select the drifter combination you would like to examine. This can be a cross reference of all drifters that were placed at a certain depth for any combination of cruises, or for all drifters at a particular location for any combination of cruises or of all drifters that were exposed during certain hours or certain lengths of time for any combination of cruises. This section can also be updated to include any new combination of drifters the user might like to explore. Once the user has selected a set of drifters it is time to define the dependant and independent variables.

F. Using the drop down menus and buttons below the drifter selection window

- Each of the next three drop down menus provides options for the X, Y and Z axis. The potential data options are the same for all three variables, currently consisting of over 200 potential data sets. The X and Y parameters must be defined for any of the graphics analysis and the Z parameter must be selected if the user is planning on creation of 3-D graphics. Once all of the desired variables have been selected the 'Display Data' button should be pushed.

- The ‘Display Data’ button will prompt the program to display all of the data selected in the drop down menus above, into the large table on the right side of the GUI. Once the data has been imported into the table it can also be edited. That is, the user can now double click any cell containing X, Y or Z data and enter a new value or a NAN value if required. **BE SURE TO HIT RETURN** after changing data in a cell or it will not be incorporated into the graphics phase.
- The next drop down window provides a list of graphics options the user can select. There are three different ways in which the user can view the data in 2-D and a 3-D graphics option which provides three different views, see Data Outputs section for more details here. Once the user has selected the graphic option, the X, Y and Z variables, modified any data and double-checked that the right drifters are selected it is time to view the graph.
- The ‘View Figure’ button will display the graph. Most of the graphs will appear as new windows, because the main window in the GUI is only designed for the 3-D contour figures. Note that the initial image in that main window is just a place holder and does not represent any of the experimental data until both the 3-D plotting option is selected and the ‘View Figure’ button is pushed. Also, notice that the 3-D options will not run successfully if the X, Y and Z parameters are not all defined.

G. The Resetting and Exit Option buttons just below the ‘View Figure’ button

- There are three different buttons for resetting and exiting the program and they each do something different. The first button is the ‘Reset All’ button. This button will clear all information from the GUI, essentially by ending the program and restarting it. This allows the user to begin a totally new examination.
- The second option is the ‘Reset Drifters’ button. This button will only clear the set of drifters that has been chosen by the user and will leave the X, Y, and Z data and the graphics options as they are, thereby allowing the user to see what different days or depths looked like under the very same circumstances.

- The 'Exit' button terminates the program completely.

H. Saving and Organizing

It is important to note that figures generated with **REAP** are not saved in any output files. Once the user is satisfied with a graphic set of graphics the data displayed on the screen should be copied over onto a spreadsheet by hand or a screenshot should be taken. This information can then be reviewed later for additional analysis or so that a more complete figure can be created for presentation or publication purposes.

VI. Output Files

There are no saved output files associated with the **REAP** code. However, the program does provide a variety of graphics displays that can be stored in the form of screenshots. Screenshots should be taken in such a way as to also display information about which drifters were selected and what data was selected for the X, Y and Z axis. **Figure 5**, provides an example of a screen shot taken of the first graphics option, the generic 2-D display. **Figure 6**, illustrates the 2-D graphics with two Y data sets graphed together. **Figure 7**, staggers the two Y data sets. **Figure 8**, shows an example of the 3-D data display. This display is created with the x-plot 3-D built in IDL graphics tool, but only shows the points floating in space. **Figure 9**, contains the same plot as **Figure 8**, but also contains two different 3-D contoured plots including the rotating green surface in the main window of the GUI.

The entire purpose of the **REAP** program is the generation of graphics for multidimensional data arrays. **Figure 9** achieves this goal, but the **REAP** program is still a work in progress and more variations and a greater variety of graphics options will eventually be included.

Figure 5

Screenshot of generic 2-D graph created using the **REAP** program.

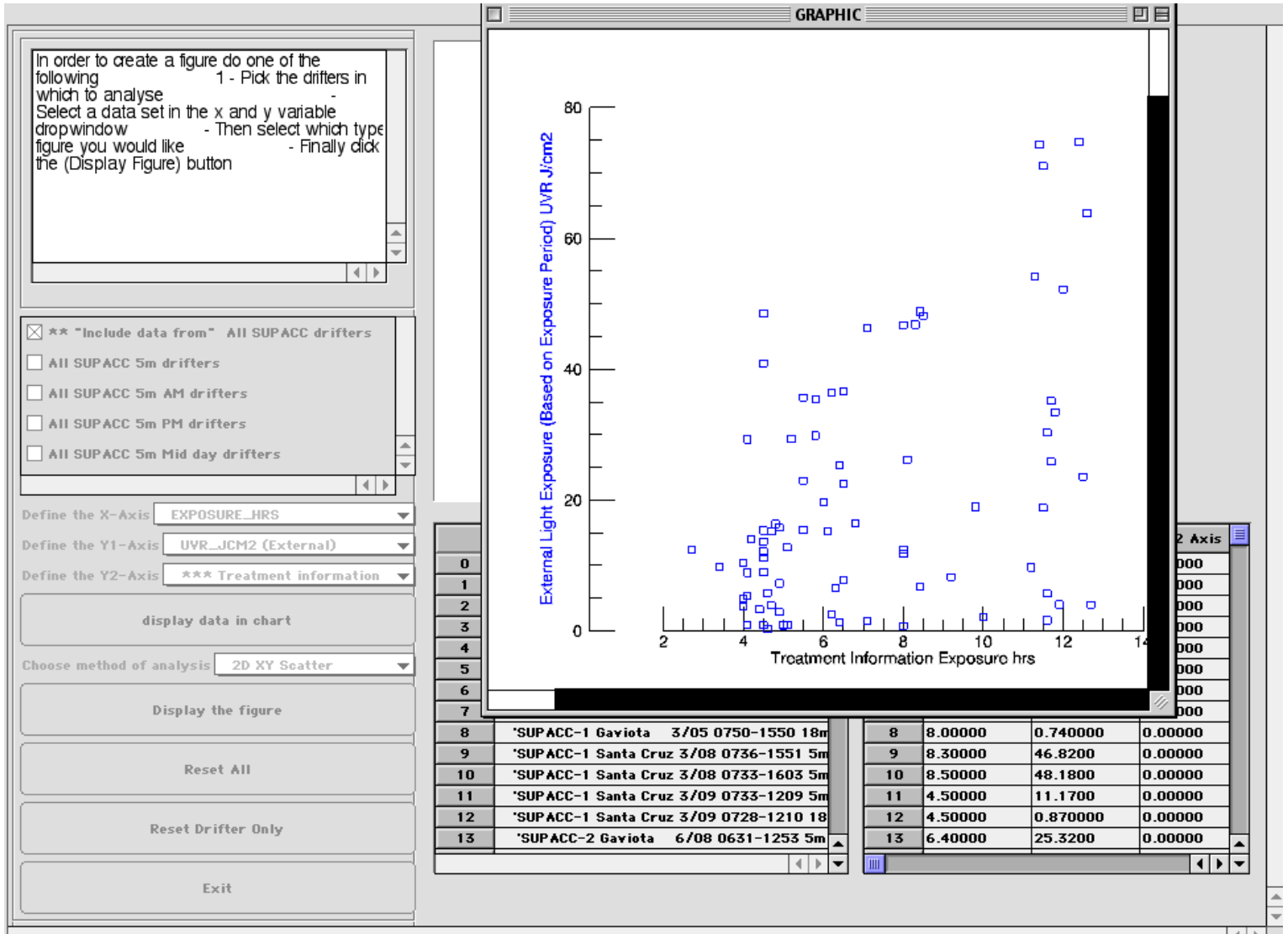


Figure 6

Screenshot of 2-D graph with multiple Y-Axis, created using the **REAP** program.

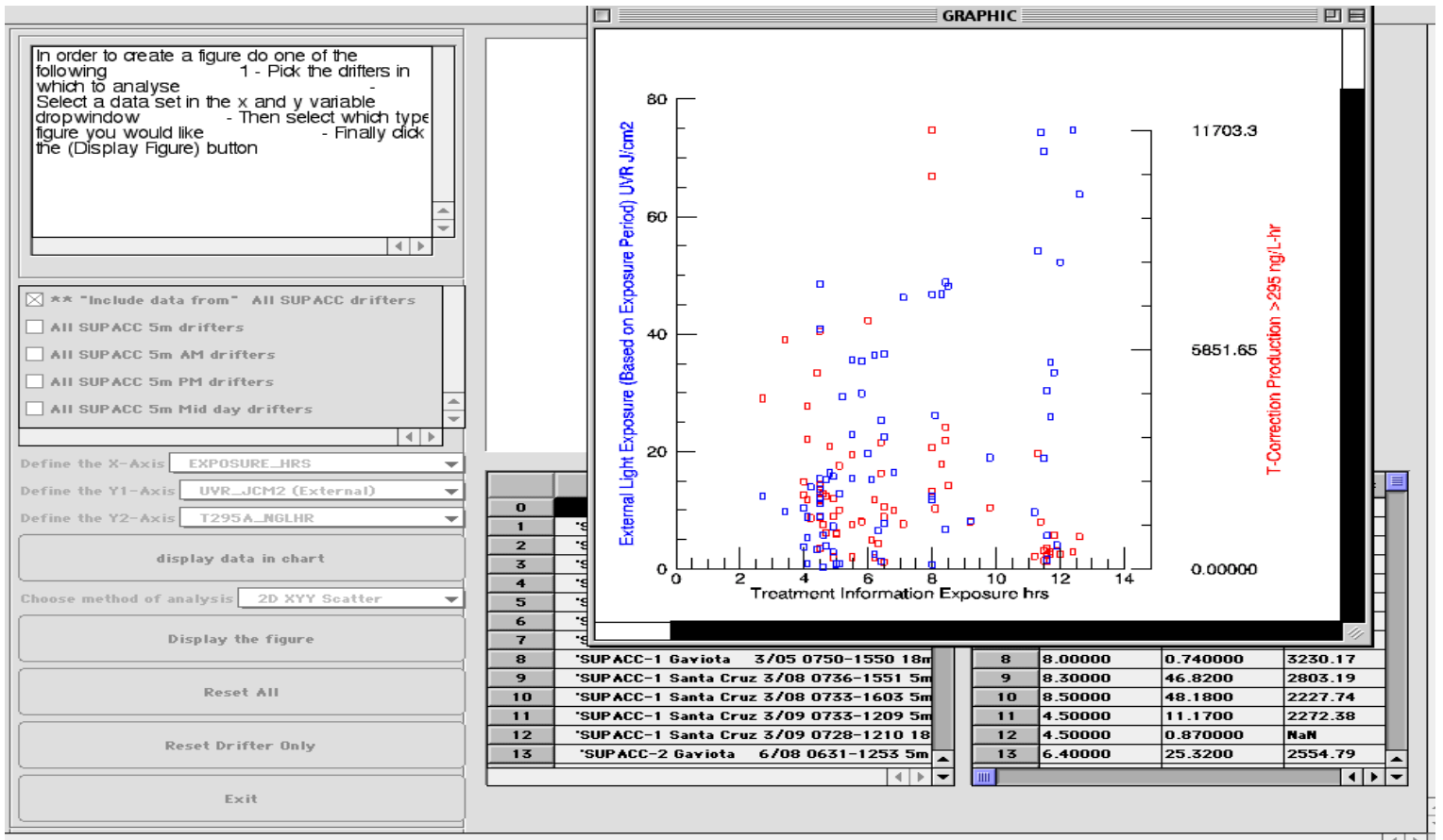


Figure 7

Screenshot of 2-D graph with staggered multiple Y-Axis, created using the **REAP** program.

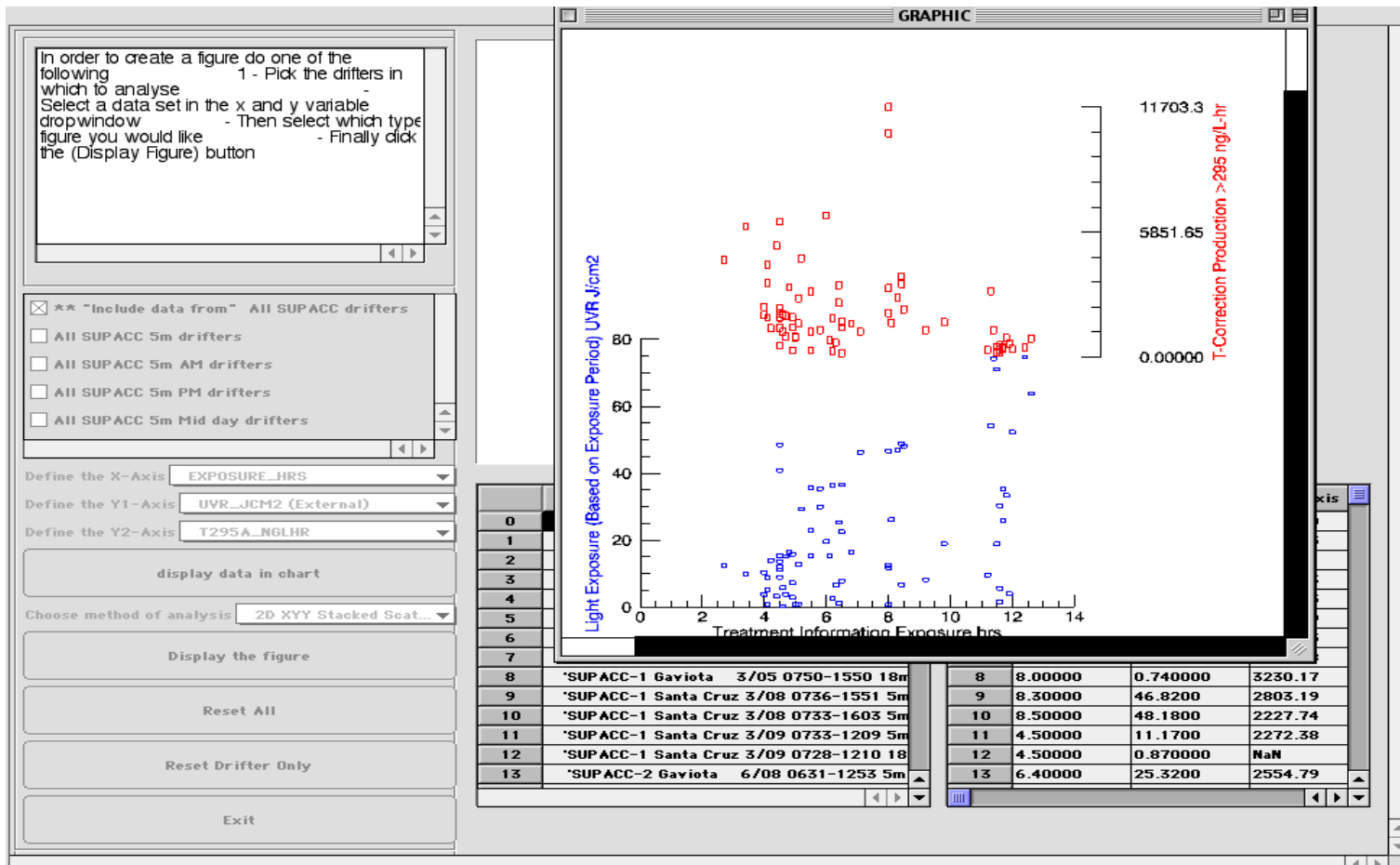
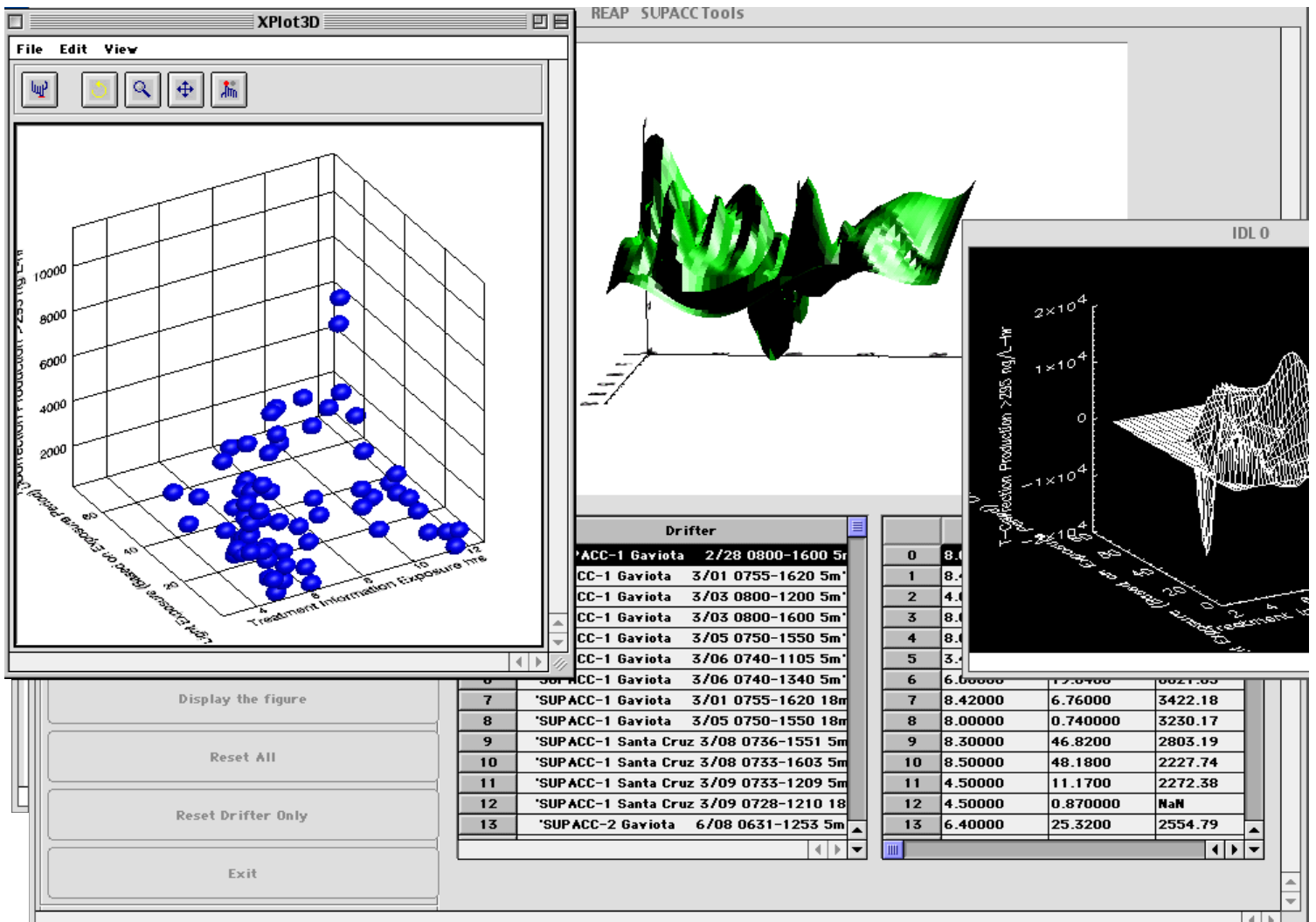


Figure 9

Screenshot of 3-D contour and data points graph, created using the **REAP** program.



VII. References

Hastings, J.T., S.C.McKagan, J. Robidart, B. B. Prezelin (2002) Technical Report 1 - SLIC. Marine Primary Production Group. University of California Santa Barbara.

McKagan, S. C., B.B. Prezelin (2002) Technical Report 2 - BWF Architect. Marine Primary Production Group. University of California Santa Barbara.

McKagan, S. C., B.B. Prezelin (2002) Technical Report 3 - BEAR . Marine Primary Production Group. University of California Santa Barbara.

McKagan, S. C., B.B. Prezelin (2002) Technical Report 4 - REAP. Marine Primary Production Group. University of California Santa Barbara.

VIII. REAP software Code

```
function make_grid, x, y, z
  triangulate, x, y, tr, b ;obtain triangulation.
  z_grid = trigrid(x, y, z, tr, xgrid = xg, ygrid = yg, $
    extrapolate = b)
; Rerturn the x and y vectors along with the z values.
  return, {z:z_grid, x:xg, y:yg}
end

pro plotit, event

;the compile_opt avoids the ()=[] ambiguity from old IDL programs
compile_opt idl2

;the @catch calls the error catch procedure defined in catch_procedure.pro
;This is more of an include file than a procedure
@catch_procedure

widget_control, event.top, get_uvalue=pstate

rawxdata = (*pstate).chosendatax
rawyldata = (*pstate).chosendatay1
rawy2data = (*pstate).chosendatay2

;this adjusts any values set at 000 to actually plot at .00001 because it is set to ignore zeros
for i = 0,82 do begin
if rawxdata[i] gt 0 or rawxdata[i] lt 0 and rawyldata[i] gt 0 or rawyldata[i] lt 0 and rawy2data[i]
eq 0 then begin
rawy2data[i] = .00001
endif
if rawxdata[i] gt 0 or rawxdata[i] lt 0 and rawy2data[i] gt 0 or rawy2data[i] lt 0 and rawyldata[i]
eq 0 then begin
rawyldata[i] = .00001
endif
if rawyldata[i] gt 0 or rawyldata[i] lt 0 and rawy2data[i] gt 0 or rawy2data[i] lt 0 and rawxdata[i]
eq 0 then begin
rawxdata[i] = .00001
endif
endifor

;print, rawxdata

realdata = where(rawxdata gt 0 or rawxdata lt 0 and rawyldata gt 0 or rawyldata lt 0 and rawy2data
gt 0 or rawy2data lt 0, count)

;print, realdata

xdata = rawxdata[realdata]
yldata = rawyldata[realdata]
y2data = rawy2data[realdata]

;print, xdata, yldata, y2data

xaxis = (*pstate).xword1 + ' ' + (*pstate).xword2 + ' ' + (*pstate).xword3
ylaxis = (*pstate).ylword1 + ' ' + (*pstate).ylword2 + ' ' + (*pstate).ylword3
y2axis = (*pstate).y2word1 + ' ' + (*pstate).y2word2 + ' ' + (*pstate).y2word3

;Now we need to define the ranges for all the graphic variables
xdatamin = min(xdata)
xdatamax = max(xdata)
xdatarange = (xdatamax - xdatamin)

yldatamin = min(yldata)
```

```

yldatamax = max(yldata)
yldatarange = (yldatamax - yldatamin)

y2datamin = min(y2data)
y2datamax = max(y2data)
y2datarange = (y2datamax - y2datamin)

;now we need to define the points in the viewplane rectangle, these will change
;based on which data set is being used
xmin = (0.0 - .2 * xdatamax)
xmax = (xdatamax + .85 * xdatamax)
y1min = (0.0 - .12 * yldatamax)
y1max = (yldatamax + .35 * yldatamax)
y2min = (0.0 - 1.2 * y2datamax)
y2max = (y2datamax + .35 * y2datamax)

symblesizeX = (xdatamax * .007)
symblesizeY = (yldatamax * .007)

xtick = yldatamax * .05
ytick = xdatamax * .05

sym1 = obj_new('idlgrsymbol',6, color=[0,0,250], $
size=[symblesizeX,symblesizeY])

;This will only graph one set of data, typical x and y fashion
if (*pstate).figtype eq 0 then begin

;now we need to define the points in the viewplane rectangle, these will change
;based on which data set is being used
xmin = (xdatamin - .2 * xdatamax)
xmax = (xdatamax + .3 * xdatamax)
y1min = (0 - .12 * yldatamax)
y1max = (yldatamax + .35 * yldatamax)
y2min = (y2datamin - 1.2 * y2datamax)
y2max = (y2datamax + .35 * y2datamax)

oview = obj_new('idlgrview', viewplane_rect = [(xmin-.2*xdatamax),y1min,xmax,(y1max)])
omodel = obj_new('idlgrmodel')
fig1 = obj_new('idlgrplot', xdata, yldata, symbol = sym1, linestyle=6, $
xrange=[xdatamin,xdatamax])
oXTitle = OBJ_NEW('IDLgrText', xaxis, font=font1)
oXAxis = OBJ_NEW('IDLgrAxis', 0, RANGE=[xdatamin, xdatamax], $
TITLE=oXTitle, ticklen = (xtick), location=[0, 0, 0])
oYTitle = OBJ_NEW('IDLgrText', y1axis, color=[0,0,250])
oYAxis = OBJ_NEW('IDLgrAxis', 1, RANGE=[0, yldatamax], $
location=[xmin,yldatamin], title = oYTitle, ticklen = ytick, /use_text_color)

oview-> add, omodel
omodel -> add, fig1
omodel -> add, oxaxis
omodel -> add, oyaxis

owindow = obj_new('idlgrwindow', title = 'GRAPHIC', $
dimensions = [500,500], location=[50,50])
;widget_control, draw, get_value=owindow
owindow->draw, oview
endif

;This will graph 2 sets of data, where 2 ys are plotted against the same x
if (*pstate).figtype eq 1 then begin

;This conversion stuff is necessary for the axis to line up
convfactor = yldatamax/y2datamax
y2dataconv = (y2data * convfactor)
y2dataminconv = min(y2dataconv)
y2datamaxconv = max(y2dataconv)
y2datarangeconv = (y2datamaxconv - y2dataminconv)

```

```

symblesizey2 = (y2datamaxconv * .007)
sym2 = obj_new('idlgrsymbol',6, color=[250,0,0], $
size=[symblesizey2,symblesizey2])
y2local = (xdatamax + .18 * xdatamax)

;makes it so the 2nd axis only has 3 values to account for
y2tickinterval = (y2datamaxconv /2)

;The first axis value is zero
y2axisvalue = fltarr(3)
y2axisvalue[0] = 0s
y2axisvalue[1] = (.5 * y2datamax)
y2axisvalue[2] = (y2datamax)

y2axisvaluestring = string(y2axisvalue)
axisvalue = obj_new('idlgrtext', y2axisvaluestring)

oview = obj_new('idlgrview', viewplane_rect = [xmin,ylmin,xmax,(ylmax)])
omodel = obj_new('idlgrmodel')
fig1 = obj_new('idlgrplot', xdata, yldata, symbol = sym1, linestyle=6)
oXTitle = OBJ_NEW('IDLgrText', xaxis, font=font1)
oXAxis = OBJ_NEW('IDLgrAxis', 0, RANGE=[0, xdatamax], $
TITLE=oXTitle, ticklen = (xtick), location=[0, 0, 0])
oYTitle = OBJ_NEW('IDLgrText', y1axis, color=[0,0,250])
oYAxis = OBJ_NEW('IDLgrAxis', 1, RANGE=[0, yldatamax], $
location=[0,0], title = oyttitle, ticklen = ytick, /use_text_color)

fig2 = obj_new('idlgrplot', xdata, y2dataconv, symbol = sym2, linestyle=6)
yltitle =obj_new('idlgrtext', y2axis, color=[250,0,0])
ylaxis = obj_new('idlgraxis', 1, RANGE=[0, y2datamaxconv], tickdir = 1, $
location=[y2local, 0], title = yltitle, ticklen = (ytick), textpos=1, $
tickinterval = y2tickinterval, ticktext = axisvalue, /use_text_color)

oview-> add, omodel
omodel -> add, fig1
omodel -> add, oxaxis
omodel -> add, oyaxis
omodel -> add, fig2
omodel -> add, ylaxis

owindow = obj_new('idlgrwindow', title = 'GRAPHIC', $
dimensions = [500,500], location=[50,50])
;widget_control, draw, get_value=owindow
owindow->draw, oview
endif

;This will graph 2 sets of data, where 2 ys are plotted against the same x but staggered
if (*pstate).figtype eq 2 then begin

;This conversion stuff is necessary for the axis to line up
convfactor = yldatamax/y2datamax
y2dataconv = (y2data * convfactor + yldatamax)
y2dataminconv = min(y2dataconv)
y2datamaxconv = max(y2dataconv)
y2datarangeconv = (y2datamaxconv - y2dataminconv)
symblesizey2 = (y2datamaxconv * .007)
sym2 = obj_new('idlgrsymbol',6, color=[250,0,0], $
size=[symblesizey2,symblesizey2])
y2local = (xdatamax + .18 * xdatamax)

;makes it so the 2nd axis only has 3 values to account for
y2tickinterval = (y2datamaxconv /4)

;The first axis value is zero
y2axisvalue = fltarr(3)
y2axisvalue[0] = 0s
y2axisvalue[1] = (.5 * y2datamax)

```

```

y2axisvalue[2] = (y2datamax)

y2axisvaluestring = string(y2axisvalue)
axisvalue = obj_new('idlgrtext', y2axisvaluestring)

oview = obj_new('idlgrview', viewplane_rect = [xmin,ylmin,xmax,(1.8 * ylmax)])
omodel = obj_new('idlgrmodel')
fig1 = obj_new('idlgrplot', xdata, yldata, symbol = sym1, linestyle=6)
oXTitle = OBJ_NEW('IDLgrText', xaxis, font=font1)
oXAxis = OBJ_NEW('IDLgrAxis', 0, RANGE=[0, xdatamax], $
TITLE=oXTitle, ticklen = (xtick), location=[0, 0, 0])
oYTitle = OBJ_NEW('IDLgrText', yldata, color=[0,0,250])
oYAxis = OBJ_NEW('IDLgrAxis', 1, RANGE=[0, yldatamax], $
location=[0,0], title = oyttitle, ticklen = ytick, /use_text_color)

fig2 = obj_new('idlgrplot', xdata, y2dataconv, symbol = sym2, linestyle=6)
y2title =obj_new('idlgrtext', y2axis, color=[250,0,0])
y2axis = obj_new('idlgraxis', 1, RANGE=[y2dataminconv, y2datamaxconv], tickdir = 1, $
location = [y2local, yldatamax], title = y2title, ticklen = (ytick), textpos=1, $
tickinterval = y2tickinterval, ticktext = axisvalue, /use_text_color)

oview-> add, omodel
omodel -> add, fig1
omodel -> add, oxaxis
omodel -> add, oyaxis
omodel -> add, fig2
omodel -> add, y2axis

owindow = obj_new('idlgrwindow', title = 'GRAPHIC', $
dimensions = [500,500], location=[50,50])
;widget_control, draw, get_value=owindow
owindow->draw, oview
endif

;This will set up the 3D scatter graph
if (*pstate).figtype eq 3 then begin

;Define custom plotting symbols:
oOrb = OBJ_NEW('orb', COLOR=[0, 0, 255])
oOrb->Scale, (xdatamax/35), (yldatamax/35), (y2datamax/35)
oSymbol = OBJ_NEW('IDLgrSymbol', oOrb)

;Overplot curve 2 in XPLOT3D:
xplot3d, Xdata, Yldata, y2data, COLOR=[0,255,0], NAME='3D Figure',$; zrange = [0,70],$
SYMBOL=oSymbol, linestyle = 6, XTITLE = xaxis , YTITLE = yldata, ZTITLE = y2axis
endif

;This will set up the 3D scatter graph
if (*pstate).figtype eq 4 then begin

;This if for the figure in the draw window

x = xdata
y = yldata
z = y2data

new_data = make_grid(x,y,z)

(*pstate).osurface -> setproperty, datax = new_data.x, $
datay = new_data.y, $
dataz = new_data.z

; Scaling the new data
; Getting the [x,y,z]-range.
(*pstate).osurface -> getproperty, xrange = xr, yrange = yr, zrange = zr

```

```

print, 'XRANGE :', xr
print, 'YRANGE :', yr
print, 'ZRANGE :', zr

; Convert to normalized coordinates
xc = norm_coord(xr)
yc = norm_coord(yr)
zc = norm_coord(zr)

; Center the surface in the view object.
xc[0] = xc[0] - 0.5
yc[0] = yc[0] - 0.5
zc[0] = zc[0] - 0.5

(*pstate).osurface -> setproperty, xcoord = xc, ycoord = yc, zcoord = zc

(*pstate).owindow -> draw, (*pstate).oview
widget_control, event.top, set_uvalue = pstate, /no_copy

;This is for the lattice style pop up figure

triangulate, xdata, yldata, tr, b
data = trigrd(xdata, yldata, y2data, tr, xgrid = xg, ygrid = yg, /quintic); or /quintic can be
replaced with extra=b

xdataAXIS = findgen(51) * ((xdatamax - xdatamin)/50) +(xdatamin)
yldataAXIS = findgen(51) * ((yldatamax - yldatamin)/50) +(yldatamin)

surface, data, xdataaxis, yldataaxis, $
charsize = 2, XTITLE = xaxis , YTITLE = ylaxis, ZTITLE = y2axis, $
max_value= (2 * y2datamax), min_value= (y2datamin - 2 * y2datamax)

;Define custom plotting symbols:
oOrb = OBJ_NEW('orb', COLOR=[0, 0, 255])
oOrb->Scale, (xdatamax/35), (yldatamax/35), (y2datamax/35)
oSymbol = OBJ_NEW('IDLgrSymbol', oOrb)

;Overplot curve 2 in XPLOT3D:

;This is the 3D points only figure
xplot3d, Xdata, Yldata, y2data, COLOR=[0,255,0], NAME='3D Figure', $; zrange = [0,70], $
SYMBOL=oSymbol, linestyle = 6, XTITLE = xaxis , YTITLE = ylaxis, ZTITLE = y2axis
endif

return
end

pro plottools_event, event

;the compile_opt avoids the ()=[] ambiguity from old IDL programs
compile_opt idl2

;the @catch calls the error catch procedure defined in catch_procedure.pro
;This is more of an include file than a procedure
@catch_procedure
help, event, /structure

widget_control, event.top, get_uvalue=pstate

case event.index of

0: begin
(*pstate).figtype = 0
end

1:begin

```

```

(*pstate).figtype = 1
end

2:begin
(*pstate).figtype = 2
end

3:begin
(*pstate).figtype = 3
end

4:begin
(*pstate).figtype = 4
end

endcase

return
end

;This section defines which data will be pulled, it is based on the boxes checked in the main GUI
;could try to develop state matrices that the drifters access in order to fullfill certain questions

pro xdisplay_event, event

;the compile_opt avoids the ()=[] ambiguity from old IDL programs
compile_opt idl2

;the @catch calls the error catch procedure defined in catch_procedure.pro
;This is more of an include file than a procedure
@catch_procedure
help, event, /structure

widget_control, event.top, get_uvalue=pstate

case event.index of

0:begin
end

1:begin
openr, lun, 'SUPACC_EXPOSURE_HRS', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

2:begin
openr, lun, 'SUPACC_FULLLDAY_HRS', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

3:begin
openr, lun, 'SUPACC_EXP_VS_FULLDAY', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

4:begin
end

```

```

5:begin
openr, lun, 'SUPACC_EXT_UVB_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

6:begin
openr, lun, 'SUPACC_EXT_UVB_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

7:begin
openr, lun, 'SUPACC_EXT_UVA_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

8:begin
openr, lun, 'SUPACC_EXT_UVA_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

9:begin

```

```

openr, lun, 'SUPACC_EXT_UVR_JCM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

10:begin
openr, lun, 'SUPACC_EXT_UVR_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

11:begin
openr, lun, 'SUPACC_EXT_PAR_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

12:begin
openr, lun, 'SUPACC_EXT_UVB_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

13:begin
openr, lun, 'SUPACC_EXT_UVA_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

14:begin
openr, lun, 'SUPACC_EXT_UVR_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''

```

```

        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

15:begin
openr, lun, 'SUPACC_EXT_UVB_VS_UVA', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

16:begin
openr, lun, 'SUPACC_EXT_UVB_VS_PAR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

17:begin
openr, lun, 'SUPACC_EXT_UVBPAR_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

18:begin
openr, lun, 'SUPACC_EXT_TOT_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

19:begin
end

20:begin
openr, lun, 'SUPACC_t295_UVB_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)

```

```

        readf, lun, datam
        free_lun, lun
end

21:begin
openr, lun, 'SUPACC_t295_UVB_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

22:begin
openr, lun, 'SUPACC_t295_UVA_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

23:begin
openr, lun, 'SUPACC_t295_UVA_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

24:begin
openr, lun, 'SUPACC_t295_UVR_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

25:begin
openr, lun, 'SUPACC_t295_UVR_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

26:begin

```

```

openr, lun, 'SUPACC_t295_PAR_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

27:begin
openr, lun, 'SUPACC_t295_UVB_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

28:begin
openr, lun, 'SUPACC_t295_UVA_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

29:begin
openr, lun, 'SUPACC_t295_UVR_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

30:begin
openr, lun, 'SUPACC_t295_UVB_VS_UVA', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

31:begin
openr, lun, 'SUPACC_t295_UVB_VS_PAR', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''

```

```

        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

32:begin
openr, lun, 'SUPACC_t295_UVBPAR_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

33:begin
openr, lun, 'SUPACC_t295_TOT_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

34:begin
end

35:begin
openr, lun, 'SUPACC_t310_UVB_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

36:begin
openr, lun, 'SUPACC_t310_UVB_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

37:begin
openr, lun, 'SUPACC_t310_UVA_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3

```

```

        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

38:begin
openr, lun, 'SUPACC_t310_UVA_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

39:begin
openr, lun, 'SUPACC_t310_UVR_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

40:begin
openr, lun, 'SUPACC_t310_UVR_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

41:begin
openr, lun, 'SUPACC_t310_PAR_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

42:begin
openr, lun, 'SUPACC_t310_UVB_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

43:begin
openr, lun, 'SUPACC_t310_UVA_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

44:begin
openr, lun, 'SUPACC_t310_UVR_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

45:begin
openr, lun, 'SUPACC_t310_UVB_VS_UVA', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

46:begin
openr, lun, 'SUPACC_t310_UVB_VS_PAR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

47:begin
openr, lun, 'SUPACC_t310_UVBPAR_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

48:begin
openr, lun, 'SUPACC_t310_TOT_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''

```

```

        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

49:begin
end

50:begin
openr, lun, 'SUPACC_t318_UVB_JCM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

51:begin
openr, lun, 'SUPACC_t318_UVB_UECM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

52:begin
openr, lun, 'SUPACC_t318_UVA_JCM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

53:begin
openr, lun, 'SUPACC_t318_UVA_UECM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

54:begin
openr, lun, 'SUPACC_t318_UVR_JCM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''

```

```

        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

55:begin
openr, lun, 'SUPACC_t318_UVR_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

56:begin
openr, lun, 'SUPACC_t318_PAR_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

57:begin
openr, lun, 'SUPACC_t318_UVB_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

58:begin
openr, lun, 'SUPACC_t318_UVA_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

59:begin
openr, lun, 'SUPACC_t318_UVR_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

60:begin
openr, lun, 'SUPACC_t318_UVB_VS_UVA', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

61:begin
openr, lun, 'SUPACC_t318_UVB_VS_PAR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

62:begin
openr, lun, 'SUPACC_t318_UVBPAR_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

63:begin
openr, lun, 'SUPACC_t318_TOT_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

64:begin
end

65:begin
openr, lun, 'SUPACC_t328_UVB_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

66:begin
openr, lun, 'SUPACC_t328_UVB_UECM2', /get_lun

```

```

        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

67:begin
openr, lun, 'SUPACC_t328_UVA_JCM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

68:begin
openr, lun, 'SUPACC_t328_UVA_UECM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

69:begin
openr, lun, 'SUPACC_t328_UVR_JCM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

70:begin
openr, lun, 'SUPACC_t328_UVR_UECM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

71:begin
openr, lun, 'SUPACC_t328_PAR_UECM2', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''

```

```

        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

72:begin
openr, lun, 'SUPACC_t328_UVB_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

73:begin
openr, lun, 'SUPACC_t328_UVA_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

74:begin
openr, lun, 'SUPACC_t328_UVR_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

75:begin
openr, lun, 'SUPACC_t328_UVB_VS_UVA', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

76:begin
openr, lun, 'SUPACC_t328_UVB_VS_PAR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```
77:begin
openr, lun, 'SUPACC_t328_UVBPAR_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end
```

```
78:begin
openr, lun, 'SUPACC_t328_TOT_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end
```

```
79:begin
end
```

```
80:begin
openr, lun, 'SUPACC_t341_UVB_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end
```

```
81:begin
openr, lun, 'SUPACC_t341_UVB_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end
```

```
82:begin
openr, lun, 'SUPACC_t341_UVA_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end
```

```
83:begin
```

```

openr, lun, 'SUPACC_t341_UVA_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

84:begin
openr, lun, 'SUPACC_t341_UVR_JCM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

85:begin
openr, lun, 'SUPACC_t341_UVR_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

86:begin
openr, lun, 'SUPACC_t341_PAR_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

87:begin
openr, lun, 'SUPACC_t341_UVB_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

88:begin
openr, lun, 'SUPACC_t341_UVA_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''

```

```

        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

89:begin
openr, lun, 'SUPACC_t341_UVR_VS_TOT', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

90:begin
openr, lun, 'SUPACC_t341_UVB_VS_UVA', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

91:begin
openr, lun, 'SUPACC_t341_UVB_VS_PAR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

92:begin
openr, lun, 'SUPACC_t341_UVBPAR_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

93:begin
openr, lun, 'SUPACC_t341_TOT_X_INC', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

```

```

94:begin
end

95:begin
openr, lun, 'SUPACC_t382_UVB_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

96:begin
openr, lun, 'SUPACC_t382_UVB_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

97:begin
openr, lun, 'SUPACC_t382_UVA_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

98:begin
openr, lun, 'SUPACC_t382_UVA_UECM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

99:begin
openr, lun, 'SUPACC_t382_UVR_JCM2', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

100:begin

```

```

openr, lun, 'SUPACC_t382_UVR_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

101:begin
openr, lun, 'SUPACC_t382_PAR_UECM2', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

102:begin
openr, lun, 'SUPACC_t382_UVB_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

103:begin
openr, lun, 'SUPACC_t382_UVA_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

104:begin
openr, lun, 'SUPACC_t382_UVR_VS_TOT', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

105:begin
openr, lun, 'SUPACC_t382_UVB_VS_UVA', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''

```

```

        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

106:begin
openr, lun, 'SUPACC_t382_UVB_VS_PAR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

107:begin
openr, lun, 'SUPACC_t382_UVBPAR_X_INC', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

108:begin
openr, lun, 'SUPACC_t382_TOT_X_INC', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

109:begin
end

110:begin
openr, lun, 'SUPACC_t410_UVB_JCM2', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

111:begin
openr, lun, 'SUPACC_t410_UVB_UECM2', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3

```

```

        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

112:begin
openr, lun, 'SUPACC_t410_UVA_JCM2', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

113:begin
openr, lun, 'SUPACC_t410_UVA_UECM2', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

114:begin
openr, lun, 'SUPACC_t410_UVR_JCM2', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

115:begin
openr, lun, 'SUPACC_t410_UVR_UECM2', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

116:begin
openr, lun, 'SUPACC_t410_PAR_UECM2', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

117:begin

```

```

openr, lun, 'SUPACC_t410_UVB_VS_TOT', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

118:begin
openr, lun, 'SUPACC_t410_UVA_VS_TOT', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

119:begin
openr, lun, 'SUPACC_t410_UVR_VS_TOT', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

120:begin
openr, lun, 'SUPACC_t410_UVB_VS_UVA', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

121:begin
openr, lun, 'SUPACC_t410_UVB_VS_PAR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

122:begin
openr, lun, 'SUPACC_t410_UVBPAR_X_INC', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''

```

```

        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

123:begin
openr, lun, 'SUPACC_t410_TOT_X_INC', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

124:begin
end

125: begin
openr, lun, 'SUPACC_ENH_UVA_%ENH', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

126: begin
openr, lun, 'SUPACC_ENH_MAXPS_TRT', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

127:begin
end

128:begin
openr, lun, 'SUPACC_PINH_UVR_%INHIB', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

129:begin
openr, lun, 'SUPACC_PINH_UVB_%INHIB', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2

```

```

        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

130:begin
openr, lun, 'SUPACC_PINH_UVA_%INHIB', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

131:begin
openr, lun, 'SUPACC_PINH_UVB_VS_UVR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

132:begin
openr, lun, 'SUPACC_PINH_UVA_VS_UVR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

133:begin
end

134:begin
openr, lun, 'SUPACC_PERR_VPS_%CORR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

135:begin
openr, lun, 'SUPACC_PERR_CLOSS_CCHLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3

```

```

        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

136:begin
end

137:begin
openr, lun, 'SUPACC_CCOR_CHLPS_%CORR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

138:begin
openr, lun, 'SUPACC_CCOR_CLOSS_CCHLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

139:begin
end

140:begin
openr, lun, 'SUPACC_TCP_T295A_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

141:begin
openr, lun, 'SUPACC_TCP_T295A_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

142:begin
openr, lun, 'SUPACC_TCP_T295A_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''

```

```

        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

143:begin
openr, lun, 'SUPACC_TCP_T295B_NGLHR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

144:begin
openr, lun, 'SUPACC_TCP_T295B_STDEV', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

145:begin
openr, lun, 'SUPACC_TCP_T295B_N', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

146:begin
openr, lun, 'SUPACC_TCP_T310_NGLHR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

147:begin
openr, lun, 'SUPACC_TCP_T310_STDEV', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun

```

```

end

148:begin
openr, lun, 'SUPACC_TCP_T310_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

149:begin
openr, lun, 'SUPACC_TCP_T318_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

150:begin
openr, lun, 'SUPACC_TCP_T318_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

151:begin
openr, lun, 'SUPACC_TCP_T318_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

152:begin
openr, lun, 'SUPACC_TCP_T328_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

153:begin
openr, lun, 'SUPACC_TCP_T328_STDEV', /get_lun
    word1 = ''
    readf, lun, word1

```

```

        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

154:begin
openr, lun, 'SUPACC_TCP_T328_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

155:begin
openr, lun, 'SUPACC_TCP_T341_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

156:begin
openr, lun, 'SUPACC_TCP_T341_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

157:begin
openr, lun, 'SUPACC_TCP_T341_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

158:begin
openr, lun, 'SUPACC_TCP_T382_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam

```

```

        free_lun, lun
end

159:begin
openr, lun, 'SUPACC_TCP_T382_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

160:begin
openr, lun, 'SUPACC_TCP_T382_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

161:begin
openr, lun, 'SUPACC_TCP_T410_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

162:begin
openr, lun, 'SUPACC_TCP_T410_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

163:begin
openr, lun, 'SUPACC_TCP_T410_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

164:begin
end

```

```
165:begin
openr, lun, 'SUPACC_TNCP_T295A_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
166:begin
openr, lun, 'SUPACC_TNCP_T295A_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
167:begin
openr, lun, 'SUPACC_TNCP_T295A_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
168:begin
openr, lun, 'SUPACC_TNCP_T295B_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
169:begin
openr, lun, 'SUPACC_TNCP_T295B_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
170:begin
openr, lun, 'SUPACC_TNCP_T295B_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
```

```

        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

171:begin
openr, lun, 'SUPACC_TNCP_T310_NGLHR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

172:begin
openr, lun, 'SUPACC_TNCP_T310_STDEV', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

173:begin
openr, lun, 'SUPACC_TNCP_T310_N', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

174:begin
openr, lun, 'SUPACC_TNCP_T318_NGLHR', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

175:begin
openr, lun, 'SUPACC_TNCP_T318_STDEV', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun

```

```

end

176:begin
openr, lun, 'SUPACC_TNCP_T318_N', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

177:begin
openr, lun, 'SUPACC_TNCP_T328_NGLHR', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

178:begin
openr, lun, 'SUPACC_TNCP_T328_STDEV', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

179:begin
openr, lun, 'SUPACC_TNCP_T328_N', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

180:begin
openr, lun, 'SUPACC_TNCP_T341_NGLHR', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
  readf, lun, word2
  word3 = ''
  readf, lun, word3
  Datam = strarr(83)
  readf, lun, datam
  free_lun, lun
end

181:begin
openr, lun, 'SUPACC_TNCP_T341_STDEV', /get_lun
  word1 = ''
  readf, lun, word1

```

```

        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

182:begin
openr, lun, 'SUPACC_TNCP_T341_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

183:begin
openr, lun, 'SUPACC_TNCP_T382_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

184:begin
openr, lun, 'SUPACC_TNCP_T382_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

185:begin
openr, lun, 'SUPACC_TNCP_T382_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

186:begin
openr, lun, 'SUPACC_TNCP_T410_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam

```

```

        free_lun, lun
end

187:begin
openr, lun, 'SUPACC_TNCP_T410_STDEV', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

188:begin
openr, lun, 'SUPACC_TNCP_T410_N', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

189:begin
end

190:begin
openr, lun, 'SUPACC_CHL_T295A_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

191:begin
openr, lun, 'SUPACC_CHL_T295B_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

192:begin
openr, lun, 'SUPACC_CHL_T310_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

```

```
193:begin
openr, lun, 'SUPACC_CHL_T318_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
194:begin
openr, lun, 'SUPACC_CHL_T328_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
195:begin
openr, lun, 'SUPACC_CHL_T341_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
196:begin
openr, lun, 'SUPACC_CHL_T382_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
197:begin
openr, lun, 'SUPACC_CHL_T410_NGL', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end
```

```
198:begin
```

```

end

199:begin
openr, lun, 'SUPACC_CCHL_T295A_NGL', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

200:begin
openr, lun, 'SUPACC_CCHL_T295B_NGL', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

201:begin
openr, lun, 'SUPACC_CCHL_T310_NGL', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

202:begin
openr, lun, 'SUPACC_CCHL_T318_NGL', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

203:begin
openr, lun, 'SUPACC_CCHL_T328_NGL', /get_lun
word1 = ''
readf, lun, word1
word2 = ''
readf, lun, word2
word3 = ''
readf, lun, word3
Datam = strarr(83)
readf, lun, datam
free_lun, lun
end

204:begin
openr, lun, 'SUPACC_CCHL_T341_NGL', /get_lun

```

```

        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

205:begin
openr, lun, 'SUPACC_CCHL_T382_NGL', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

206:begin
openr, lun, 'SUPACC_CCHL_T410_NGL', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

207:begin
end

208:begin
openr, lun, 'SUPACC_TCPCHL_T295A_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

209:begin
openr, lun, 'SUPACC_TCPCHL_T295B_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

210:begin
openr, lun, 'SUPACC_TCPCHL_T310_NGLHR', /get_lun

```

```

        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

211:begin
openr, lun, 'SUPACC_TCPCHL_T318_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

212:begin
openr, lun, 'SUPACC_TCPCHL_T328_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

213:begin
openr, lun, 'SUPACC_TCPCHL_T341_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

214:begin
openr, lun, 'SUPACC_TCPCHL_T382_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

215:begin
openr, lun, 'SUPACC_TCPCHL_T410_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1

```

```

        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

216:begin
end

217:begin
openr, lun, 'SUPACC_TNCPCHL_T295A_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

218:begin
openr, lun, 'SUPACC_TNCPCHL_T295B_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

219:begin
openr, lun, 'SUPACC_TNCPCHL_T310_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

220:begin
openr, lun, 'SUPACC_TNCPCHL_T318_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''
    readf, lun, word2
    word3 = ''
    readf, lun, word3
    Datam = strarr(83)
    readf, lun, datam
    free_lun, lun
end

221:begin
openr, lun, 'SUPACC_TNCPCHL_T328_NGLHR', /get_lun
    word1 = ''
    readf, lun, word1
    word2 = ''

```

```

        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

222:begin
openr, lun, 'SUPACC_TNCPCHL_T341_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

223:begin
openr, lun, 'SUPACC_TNCPCHL_T382_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end

224:begin
openr, lun, 'SUPACC_TNCPCHL_T410_NGLHR', /get_lun
        word1 = ''
        readf, lun, word1
        word2 = ''
        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun
end
endcase

(*pstate).datamx = datam

totalchosen = 0
chosendata = fltarr(83)
for i = 0,82 do begin
if (*pstate).drifter[i] eq 1 then begin
chosendata[i] = datam[i]
totalchosen = totalchosen + 1
endif
endifor

(*pstate).chosendatax= chosendata
(*pstate).xword1 = word1
(*pstate).xword2 = word2
(*pstate).xword3 = word3

;widget_control, widget_info(event.top, find_by_undef='table_vals'), $
;        set_value=chosendata

return

```

end

```
*****
THE DATA CALLING ROUTINE PERFORMED FOR X ABOVE IS REPEATED FOR Y AND FOR Z
BUT HAS BEEN REMOED HERE IN THE INTEREST OF PRESERVING SPACE.  FOR THE ENTIRITY
OF THE CODE PLEASE SEE THE ATTACHED CD AND OR CONTACT THE MPPG AS LISTED IN THE FOREWARD
*****
```

```
pro table_edit, event
```

```
;the compile_opt avoids the ()=[] ambiguity from old IDL programs
compile_opt idl2
```

```
;the @catch calls the error catch procedure defined in catch_procedure.pro
;This is more of an include file than a procedure
@catch_procedure
help, event, /structure
```

```
widget_control, event.top, get_uvalue=pstate
widget_control, event.id, get_value=tblval
```

```
takeval = fltarr(3,83)
takeval = tblval
```

```
;print, takeval[0,*]
```

```
(*pstate).chosendatax = takeval[0,*]
(*pstate).chosendatay1 = takeval[1,*]
(*pstate).chosendatay2 = takeval[2,*]
```

```
return
end
```

```
pro test_trackball_draw3, event
  widget_control, event.top, get_uvalue = pstate, /no_copy
  if event.type eq 4 then (*pstate).owindow -> draw, (*pstate).oview
  h = (*pstate).otrack -> update(event, transform=new)
  if h then begin
    (*pstate).omodel -> getproperty, transform = old
    (*pstate).omodel -> setproperty, transform = old#new
    (*pstate).owindow -> draw, (*pstate).oview
  endif
  widget_control, event.top, set_uvalue = pstate, /no_copy
end
```

```
pro chartit, event
widget_control, event.top, get_uvalue=pstate
```

```
;print, (*pstate).drifter
```

```
;Array = total((*pstate).drifter)
```

```
drifter_name = strarr(1,83)
```

```
openr, lun, 'SUPACC_DRIFTER_NAMES', /get_lun
  word1 = ''
  readf, lun, word1
  word2 = ''
```

```

        readf, lun, word2
        word3 = ''
        readf, lun, word3
        Datam = strarr(83)
        readf, lun, datam
        free_lun, lun

totalchosen = 0
drifter_name = strarr(1,83)
for i= 0,82 do begin
if (*pstate).drifter[i] eq 1 then begin
totalchosen = totalchosen + 1
drifter_name[i] = datam[i]
endif
endifor

widget_control, widget_info(event.top, find_by_uname='drifter_names'), $
set_value= drifter_name

chartdata = fltarr(3,83)

chartdata[0,*] = (*pstate).chosendatax
chartdata[1,*] = (*pstate).chosendatay1
chartdata[2,*] = (*pstate).chosendatay2

;print, 'ChartData', (*pstate).chosendatax
(*pstate).chartdata = chartdata

widget_control, widget_info(event.top, find_by_uname='table_vals'), $
        set_value=chartdata

return
end

pro drifter1, event
widget_control, event.top, get_uvalue=pstate
for i=0,82 do begin
(*pstate).drifter[i] = 1
(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endifor

return
end

pro drifter2, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter3, event

```

```

widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).time[i] eq 1 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif

return
end

pro drifter4, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).time[i] eq 2 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif

return
end

pro drifter5, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).time[i] eq 3 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif

return
end

pro drifter5a, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).expvsday[i] lt .4 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif

return
end

pro drifter5b, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin

```

```

if (*pstate).expvsday[i] gt .4 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter5c, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).expvsday[i] lt .5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter5d, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).expvsday[i] gt .5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter5g, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).expvsday[i] lt .6 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter5h, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).expvsday[i] gt .6 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

```

```

end

pro drifter5e, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).expvsday[i] lt .7 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter5f, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).expvsday[i] gt .7 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter6, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 1 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endifor

return
end

pro drifter7, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] ge 2 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endifor

return
end

```

```

pro drifter8, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 1.5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif
endfor

return
end

pro drifter9, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 9 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endfor

return
end

pro drifter10, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 18 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endfor

return
end

pro drifter11, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 59 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endfor

return
end

pro drifter12, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 518 then begin

```

```

(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

pro drifter13, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 95 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

pro drifter14, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 918 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

pro drifter15, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 185 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

pro drifter16, event
widget_control, event.top, get_uvalue=pstate
for i =0,82 do begin
if (*pstate).depth[i] eq 189 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

```

```

end

pro drifter17, event
widget_control, event.top, get_uvalue=pstate
for i=0,12 do begin
(*pstate).drifter[i] = 1
(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endfor

return
end

pro drifter18, event
widget_control, event.top, get_uvalue=pstate
for i =0,12 do begin
if (*pstate).depth[i] eq 5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

pro drifter19, event
widget_control, event.top, get_uvalue=pstate
for i =0,12 do begin
if (*pstate).depth[i] eq 9 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

pro drifter20, event
widget_control, event.top, get_uvalue=pstate
for i =0,12 do begin
if (*pstate).depth[i] eq 18 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

return
end

pro drifter21, event
widget_control, event.top, get_uvalue=pstate
for i =0,12 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 1 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]

```

```

endif
endif
endif

return
end

pro drifter22, event
widget_control, event.top, get_uvalue=pstate
for i =0,12 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 2 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif

return
end

pro drifter23, event
widget_control, event.top, get_uvalue=pstate
for i=13,37 do begin
(*pstate).drifter[i] = 1
(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif

return
end

pro drifter24, event
widget_control, event.top, get_uvalue=pstate
for i =13,37 do begin
if (*pstate).depth[i] eq 5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif

return
end

pro drifter25, event
widget_control, event.top, get_uvalue=pstate
for i =13,37 do begin
if (*pstate).depth[i] eq 9 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif

return
end

pro drifter26, event
widget_control, event.top, get_uvalue=pstate
for i =13,37 do begin

```

```

if (*pstate).depth[i] eq 18 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

return
end

pro drifter27, event
widget_control, event.top, get_uvalue=pstate
for i =13,37 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 1 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endifor

return
end

pro drifter28, event
widget_control, event.top, get_uvalue=pstate
for i =13,37 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 2 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endifor

return
end

pro drifter29, event
widget_control, event.top, get_uvalue=pstate
for i=38,64 do begin
(*pstate).drifter[i] = 1
(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endifor

return
end

pro drifter30, event
widget_control, event.top, get_uvalue=pstate
for i =38,64 do begin
if (*pstate).depth[i] eq 5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endifor

```

```

return
end

pro drifter31, event
widget_control, event.top, get_uvalue=pstate
for i =38,64 do begin
if (*pstate).depth[i] eq 9 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endfor

return
end

pro drifter32, event
widget_control, event.top, get_uvalue=pstate
for i =38,64 do begin
if (*pstate).depth[i] eq 18 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endfor

return
end

pro drifter33, event
widget_control, event.top, get_uvalue=pstate
for i =38,64 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 1.5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif
endfor

return
end

pro drifter34, event
widget_control, event.top, get_uvalue=pstate
for i =38,64 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 2 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif
endfor

return
end

```

```

pro drifter35, event
widget_control, event.top, get_uvalue=pstate
for i=65,82 do begin
(*pstate).drifter[i] = 1
(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endfor

```

```

return
end

```

```

pro drifter36, event
widget_control, event.top, get_uvalue=pstate
for i =65,82 do begin
if (*pstate).depth[i] eq 5 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

```

```

return
end

```

```

pro drifter37, event
widget_control, event.top, get_uvalue=pstate
for i =65,82 do begin
if (*pstate).depth[i] eq 9 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

```

```

return
end

```

```

pro drifter38, event
widget_control, event.top, get_uvalue=pstate
for i =65,82 do begin
if (*pstate).depth[i] eq 10 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endfor

```

```

return
end

```

```

pro drifter39, event
widget_control, event.top, get_uvalue=pstate
for i =65,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 3 then begin
(*pstate).drifter[i] = 1

```

```

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif

```

```

endfor

return
end

pro drifter40, event
widget_control, event.top, get_uvalue=pstate
for i =65,82 do begin
if (*pstate).depth[i] eq 5 then begin
if (*pstate).location[i] eq 4 then begin
(*pstate).drifter[i] = 1

(*pstate).chosendatax[i]=(*pstate).datamx[i]
(*pstate).chosendatay1[i]=(*pstate).datamy1[i]
(*pstate).chosendatay2[i]=(*pstate).datamy2[i]
endif
endif
endif
endif

return
end

pro reset_treatment_event, event
;the compile_opt avoids the ()=[] ambiguity from old IDL programs
compile_opt idl2

;This will allow us to destroy the TLB, killing widget and exiting Xmanager
widget_control, event.top, /destroy

reap

return
end

pro reset_drifter_event, event
;the compile_opt avoids the ()=[] ambiguity from old IDL programs
compile_opt idl2

;This will allow us to destroy the TLB, killing widget and exiting Xmanager
widget_control, event.top, get_uvalue = pstate

for i =0,82 do begin
(*pstate).drifter[i] = 0
(*pstate).chosendatax[i]=0
(*pstate).chosendatay1[i]=0
(*pstate).chosendatay2[i]=0
endif
endif
return
end

pro exit, event
widget_control, event.top, /destroy
return
end

;This sets the widget base to avoid errors
pro reap_event, event

```

```

return
end

pro reap
compile_opt idl2
@catch_procedure

;This is the main gui in which all the other guis are found
Base = widget_base(title = 'REAP SUPACC Tools', /row, xoffset=30, yoffset = 30, $
ysize=700, xsize=960)

;This is the set of GUI's that drives teh left half of the interface
ABase = widget_base(base, /column, /frame, xsize = 300)

;This is the gui which provides instructions on how to use these tools
IBase = widget_base(abase, /column, /frame, space=5, ysize = 200, xsize = 290)

;These are the instructions that will expand as the program expands
text = 'In order to create a figure do one of the following           ' $
      +'1 - Pick the drifters in which to analyse                   '$
      +'    - Select a data set in the x and y variable dropdown   ' $
      +'    - Then select which type of figure you would like      ' $
      +'    - Finally click the (Display Figure) button'

wtext = widget_text(ibase, value=text, /wrap, /edit, /scroll, ysize = 12, font='helvetica*13')

;Here I need to set up a widget base inside of abase which is nonexclusive so that I can have the
button array
nonexbase = widget_base(abase, /column, /nonexclusive, /scroll, scr_ysize = 120)

;This is a list of the possibel drifter options that can be mixed and matched
drifter1 = widget_button(nonexbase, value = '** "Include data from" All SUPACC drifters', event_pro
= 'drifter1')
drifter2 = widget_button(nonexbase, value = 'All SUPACC 5m drifters', event_pro = 'drifter2')
drifter3 = widget_button(nonexbase, value = 'All SUPACC 5m AM drifters', event_pro = 'drifter3')
drifter4 = widget_button(nonexbase, value = 'All SUPACC 5m PM drifters', event_pro = 'drifter4')
drifter5 = widget_button(nonexbase, value = 'All SUPACC 5m Mid day drifters', event_pro = 'drifter5')

drifter5a = widget_button(nonexbase, value = 'drifters where exp(t)/DL is < 0.4', event_pro =
'drifter5a')
drifter5b = widget_button(nonexbase, value = 'drifters where exp(t)/DL is > 0.4', event_pro =
'drifter5b')
drifter5c = widget_button(nonexbase, value = 'drifters where exp(t)/DL is < 0.5', event_pro =
'drifter5c')
drifter5d = widget_button(nonexbase, value = 'drifters where exp(t)/DL is > 0.5', event_pro =
'drifter5d')
drifter5g = widget_button(nonexbase, value = 'drifters where exp(t)/DL is < 0.6', event_pro =
'drifter5g')
drifter5h = widget_button(nonexbase, value = 'drifters where exp(t)/DL is > 0.6', event_pro =
'drifter5h')
drifter5e = widget_button(nonexbase, value = 'drifters where exp(t)/DL is < 0.7', event_pro =
'drifter5e')
drifter5f = widget_button(nonexbase, value = 'drifters where exp(t)/DL is > 0.7', event_pro =
'drifter5f')

drifter6 = widget_button(nonexbase, value = 'All Gaviota 5m drifters', event_pro = 'drifter6')
drifter7 = widget_button(nonexbase, value = 'All Santa Cruz 5m drifters', event_pro = 'drifter7')
drifter8 = widget_button(nonexbase, value = 'All Santa Rosa 5m drifters', event_pro = 'drifter8')
drifter9 = widget_button(nonexbase, value = 'All SUPACC 9m drifters', event_pro = 'drifter9')
drifter10 = widget_button(nonexbase, value = 'All SUPACC 18m drifters', event_pro = 'drifter10')
drifter11 = widget_button(nonexbase, value = 'All SUPACC 5@9m drifters', event_pro = 'drifter11')
drifter12 = widget_button(nonexbase, value = 'All SUPACC 5@18m drifters', event_pro = 'drifter12')
drifter13 = widget_button(nonexbase, value = 'All SUPACC 9@5m drifters', event_pro = 'drifter13')
drifter14 = widget_button(nonexbase, value = 'All SUPACC 9@18m drifters', event_pro = 'drifter14')

```

```

drifter15 = widget_button(nonexbase, value = 'All SUPACC 18@5m drifters', event_pro = 'drifter15')
drifter16 = widget_button(nonexbase, value = 'All SUPACC 18@9m drifters', event_pro = 'drifter16')

drifter17 = widget_button(nonexbase, value = '** All SUPACC#1 drifters', event_pro = 'drifter17')
drifter18 = widget_button(nonexbase, value = 'SUPACC#1 5m drifters', event_pro = 'drifter18')
;drifter19 = widget_button(nonexbase, value = 'SUPACC#1 9m drifters', event_pro = 'drifter19')
drifter20 = widget_button(nonexbase, value = 'SUPACC#1 18m drifters', event_pro = 'drifter20')
drifter21 = widget_button(nonexbase, value = 'SUPACC#1 Gaviota 5m drifters', event_pro = 'drifter21')
drifter22 = widget_button(nonexbase, value = 'SUPACC#1 Santa Cruz 5m drifters', event_pro =
'drifter22')

drifter23 = widget_button(nonexbase, value = '** All SUPACC#2 drifters', event_pro = 'drifter23')
drifter24 = widget_button(nonexbase, value = 'SUPACC#2 5m drifters', event_pro = 'drifter24')
drifter25 = widget_button(nonexbase, value = 'SUPACC#2 9m drifters', event_pro = 'drifter25')
drifter26 = widget_button(nonexbase, value = 'SUPACC#2 18m drifters', event_pro = 'drifter26')
drifter27 = widget_button(nonexbase, value = 'SUPACC#2 Gaviota 5m drifters', event_pro = 'drifter27')
drifter28 = widget_button(nonexbase, value = 'SUPACC#2 Santa Cruz 5m drifters', event_pro =
'drifter28')

drifter29 = widget_button(nonexbase, value = '** All SUPACC#3 drifters', event_pro = 'drifter29')
drifter30 = widget_button(nonexbase, value = 'SUPACC#3 5m drifters', event_pro = 'drifter30')
drifter31 = widget_button(nonexbase, value = 'SUPACC#3 9m drifters', event_pro = 'drifter31')
drifter32 = widget_button(nonexbase, value = 'SUPACC#3 18m drifters', event_pro = 'drifter32')
drifter33 = widget_button(nonexbase, value = 'SUPACC#3 Santa Rosa 5m drifters', event_pro =
'drifter33')
drifter34 = widget_button(nonexbase, value = 'SUPACC#3 Santa Cruz 5m drifters', event_pro =
'drifter34')

drifter35 = widget_button(nonexbase, value = '** All SUPACC#4 drifters', event_pro = 'drifter35')
drifter36 = widget_button(nonexbase, value = 'SUPACC#4 5m drifters', event_pro = 'drifter36')
drifter37 = widget_button(nonexbase, value = 'SUPACC#4 9m drifters', event_pro = 'drifter37')
drifter38 = widget_button(nonexbase, value = 'SUPACC#4 18m drifters', event_pro = 'drifter38')
drifter39 = widget_button(nonexbase, value = 'SUPACC#4 Santa Cruz (15.5) 5m drifters', event_pro =
'drifter39')
drifter40 = widget_button(nonexbase, value = 'SUPACC#4 Santa Cruz (13.8) 5m drifters', event_pro =
'drifter40')

;The following will create the gui that has the large drop windows in which you define the x and y
axis
;This defines the x-axis though the data calling will be in the xdisplay event, based on what is
chosen here
xvariables = ['*** Treatment information', $
'EXPOSURE_HRS', 'FULLDAY_HRS', 'EXP_VS_FULLDAY', $
'*** External Light Exposure', $
'UVB_JCM2 (External)', 'UVB_UECM2 (External)', 'UVA_JCM2 (External)', 'UVA_UECM2 (External)', $
'UVR_JCM2 (External)', 'UVR_UECM2 (External)', 'PAR_UECM2 (External)', 'UVB_VS_TOT (External)', $
'UVA_VS_TOT (External)', 'UVR_VS_TOT (External)', 'UVB_VS_UVA (External)', 'UVB_VS_PAR (External)', $
'UVBPAR_X_INC (External)', 'TOT_X_INC (External)', $
'*** >295 Treatment Exposure', $
'UVB_JCM2', 'UVB_UECM2 (>295)', 'UVA_JCM2 (>295)', 'UVA_UECM2 (>295)', 'UVR_JCM2 (>295)', $
'UVR_UECM2 (>295)', 'PAR_UECM2 (>295)', 'UVB_VS_TOT (>295)', 'UVA_VS_TOT (>295)', $
'UVR_VS_TOT (>295)', 'UVB_VS_UVA (>295)', 'UVB_VS_PAR (>295)', 'UVBPAR_X_INC (>295)', 'TOT_X_INC
(>295)', $
'*** >310 Treatment Exposure', $
'UVB_JCM2 (>310)', 'UVB_UECM2 (>310)', 'UVA_JCM2 (>310)', 'UVA_UECM2 (>310)', 'UVR_JCM2 (>310)', $
'UVR_UECM2 (>310)', 'PAR_UECM2 (>310)', 'UVB_VS_TOT (>310)', 'UVA_VS_TOT (>310)', 'UVR_VS_TOT
(>310)', $
'UVB_VS_UVA (>310)', 'UVB_VS_PAR (>310)', 'UVBPAR_X_INC (>310)', 'TOT_X_INC (>310)', $
'*** >318 Treatment Exposure', $
'UVB_JCM2 (>318)', 'UVB_UECM2 (>318)', 'UVA_JCM2 (>318)', 'UVA_UECM2 (>318)', 'UVR_JCM2 (>318)', $
'UVR_UECM2 (>318)', 'PAR_UECM2 (>318)', 'UVB_VS_TOT (>318)', 'UVA_VS_TOT (>318)', 'UVR_VS_TOT
(>318)', $
'UVB_VS_UVA (>318)', 'UVB_VS_PAR (>318)', 'UVBPAR_X_INC (>318)', 'TOT_X_INC (>318)', $
'*** >328 Treatment Exposure', $
'UVB_JCM2 (>328)', 'UVB_UECM2 (>328)', 'UVA_JCM2 (>328)', 'UVA_UECM2 (>328)', 'UVR_JCM2 (>328)', $
'UVR_UECM2 (>328)', 'PAR_UECM2 (>328)', 'UVB_VS_TOT (>328)', 'UVA_VS_TOT (>328)', 'UVR_VS_TOT
(>328)', $
'UVB_VS_UVA (>328)', 'UVB_VS_PAR (>328)', 'UVBPAR_X_INC (>328)', 'TOT_X_INC (>328)', $

```

```

'*** >342 Treatment Exposure', $
'UVB_JCM2 (>342)', 'UVB_U ECM2 (>342)', 'UVA_JCM2 (>342)', 'UVA_U ECM2 (>342)', 'UVR_JCM2 (>342)', $
'UVR_U ECM2 (>342)', 'PAR_U ECM2 (>342)', 'UVB_VS_TOT (>342)', 'UVA_VS_TOT (>342)', 'UVR_VS_TOT
(>342)', $
'UVB_VS_UVA (>342)', 'UVB_VS_PAR (>342)', 'UVBPAR_X_INC (>342)', 'TOT_X_INC (>342)', $
'*** >381 Treatment Exposure', $
'UVB_JCM2 (>381)', 'UVB_U ECM2 (>381)', 'UVA_JCM2 (>381)', 'UVA_U ECM2 (>381)', 'UVR_JCM2 (>381)', $
'UVR_U ECM2 (>381)', 'PAR_U ECM2 (>381)', 'UVB_VS_TOT (>381)', 'UVA_VS_TOT (>381)', 'UVR_VS_TOT
(>381)', $
'UVB_VS_UVA (>381)', 'UVB_VS_PAR (>381)', 'UVBPAR_X_INC (>381)', 'TOT_X_INC (>381)', $
'*** >410 Treatment Exposure', $
'UVB_JCM2 (>402)', 'UVB_U ECM2 (>402)', 'UVA_JCM2 (>402)', 'UVA_U ECM2 (>402)', 'UVR_JCM2 (>402)', $
'UVR_U ECM2 (>402)', 'PAR_U ECM2 (>402)', 'UVB_VS_TOT (>402)', 'UVA_VS_TOT (>402)', 'UVR_VS_TOT
(>402)', $
'UVB_VS_UVA (>402)', 'UVB_VS_PAR (>402)', 'UVBPAR_X_INC (>402)', 'TOT_X_INC (>402)', $
'*** Enhancement Effects', $
'UVA_%ENH', 'MAXIMUMPS_TRT', $
'*** Inhibition Effects', $
'UVR_%INHIB', 'UVB_%INHIB', 'UVA_%INHIB', 'UVB_VS_UVR (14-C)', 'UVA_VS_UVR (14-C)', $
'*** PAR Error Estimates', $
'VPS_%CORR', 'CLOSS_CCHLHR', $
'*** CHL Corrections', $
'CHLPS_%CORR', 'CLOSS_CCHLHR', $
'*** T-Corrected Production', $
'T295A_NGLHR', 'T295A_STDEV', 'T295A_N', 'T295B_NGLHR', 'T295B_STDEV', $
'T295B_N', 'T310_NGLHR', 'T310_STDEV', 'T310_N', 'T318_NGLHR', 'T318_STDEV', $
'T318_N', 'T328_NGLHR', 'T328_STDEV', 'T328_N', 'T341_NGLHR', 'T341_STDEV', $
'T341_N', 'T382_NGLHR', 'T382_STDEV', 'T382_N', 'T410_NGLHR', 'T410_STDEV', $
'T410_N', $
'*** T-NON-Corrected Production', $
'T295A_NGLHR', 'T295A_STDEV', 'T295A_N', 'T295B_NGLHR', 'T295B_STDEV', $
'T295B_N', 'T310_NGLHR', 'T310_STDEV', 'T310_N', 'T318_NGLHR', 'T318_STDEV', $
'T318_N', 'T328_NGLHR', 'T328_STDEV', 'T328_N', 'T341_NGLHR', 'T341_STDEV', $
'T341_N', 'T382_NGLHR', 'T382_STDEV', 'T382_N', 'T410_NGLHR', 'T410_STDEV', $
'T410_N', $
'*** T-Corrected Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-NON-Corrected Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-Corrected Production per Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-NON-Corrected Production per Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR']

```

```

xdisplay = widget_droplist(Abase, value = xvariables, title = 'Define the X-Axis', $
event_pro = 'xdisplay_event')

```

;This defines the y1-axis though the data calling will be in the xdisplay event, based on what is chosen here

```

ylvariables = ['*** Treatment information', $
'EXPOSURE_HRS', 'FULLDAY_HRS', 'EXP_VS_FULLDAY', $
'*** External Light Exposure', $
'UVB_JCM2 (External)', 'UVB_U ECM2 (External)', 'UVA_JCM2 (External)', 'UVA_U ECM2 (External)', $
'UVR_JCM2 (External)', 'UVR_U ECM2 (External)', 'PAR_U ECM2 (External)', 'UVB_VS_TOT (External)', $
'UVA_VS_TOT (External)', 'UVR_VS_TOT (External)', 'UVB_VS_UVA (External)', 'UVB_VS_PAR (External)', $
'UVBPAR_X_INC (External)', 'TOT_X_INC (External)', $
'*** >295 Treatment Exposure', $
'UVB_JCM2 (>295)', 'UVB_U ECM2 (>295)', 'UVA_JCM2 (>295)', 'UVA_U ECM2 (>295)', 'UVR_JCM2 (>295)', $
'UVR_U ECM2 (>295)', 'PAR_U ECM2 (>295)', 'UVB_VS_TOT (>295)', 'UVA_VS_TOT (>295)', $
'UVR_VS_TOT (>295)', 'UVB_VS_UVA (>295)', 'UVB_VS_PAR (>295)', 'UVBPAR_X_INC (>295)', 'TOT_X_INC
(>295)', $
'*** >310 Treatment Exposure', $
'UVB_JCM2 (>310)', 'UVB_U ECM2 (>310)', 'UVA_JCM2 (>310)', 'UVA_U ECM2 (>310)', 'UVR_JCM2 (>310)', $
'UVR_U ECM2 (>310)', 'PAR_U ECM2 (>310)', 'UVB_VS_TOT (>310)', 'UVA_VS_TOT (>310)', 'UVR_VS_TOT
(>310)', $

```

```

'UVB_VS_UVA (>310)', 'UVB_VS_PAR (>310)', 'UVBPAR_X_INC (>310)', 'TOT_X_INC (>310)', $
'*** >318 Treatment Exposure', $
'UVB_JCM2 (>318)', 'UVB_UECM2 (>318)', 'UVA_JCM2 (>318)', 'UVA_UECM2 (>318)', 'UVR_JCM2 (>318)', $
'UVR_UECM2 (>318)', 'PAR_UECM2 (>318)', 'UVB_VS_TOT (>318)', 'UVA_VS_TOT (>318)', 'UVR_VS_TOT
(>318)', $
'UVB_VS_UVA (>318)', 'UVB_VS_PAR (>318)', 'UVBPAR_X_INC (>318)', 'TOT_X_INC (>318)', $
'*** >328 Treatment Exposure', $
'UVB_JCM2 (>328)', 'UVB_UECM2 (>328)', 'UVA_JCM2 (>328)', 'UVA_UECM2 (>328)', 'UVR_JCM2 (>328)', $
'UVR_UECM2 (>328)', 'PAR_UECM2 (>328)', 'UVB_VS_TOT (>328)', 'UVA_VS_TOT (>328)', 'UVR_VS_TOT
(>328)', $
'UVB_VS_UVA (>328)', 'UVB_VS_PAR (>328)', 'UVBPAR_X_INC (>328)', 'TOT_X_INC (>328)', $
'*** >342 Treatment Exposure', $
'UVB_JCM2 (>342)', 'UVB_UECM2 (>342)', 'UVA_JCM2 (>342)', 'UVA_UECM2 (>342)', 'UVR_JCM2 (>342)', $
'UVR_UECM2 (>342)', 'PAR_UECM2 (>342)', 'UVB_VS_TOT (>342)', 'UVA_VS_TOT (>342)', 'UVR_VS_TOT
(>342)', $
'UVB_VS_UVA (>342)', 'UVB_VS_PAR (>342)', 'UVBPAR_X_INC (>342)', 'TOT_X_INC (>342)', $
'*** >381 Treatment Exposure', $
'UVB_JCM2 (>381)', 'UVB_UECM2 (>381)', 'UVA_JCM2 (>381)', 'UVA_UECM2 (>381)', 'UVR_JCM2 (>381)', $
'UVR_UECM2 (>381)', 'PAR_UECM2 (>381)', 'UVB_VS_TOT (>381)', 'UVA_VS_TOT (>381)', 'UVR_VS_TOT
(>381)', $
'UVB_VS_UVA (>381)', 'UVB_VS_PAR (>381)', 'UVBPAR_X_INC (>381)', 'TOT_X_INC (>381)', $
'*** >410 Treatment Exposure', $
'UVB_JCM2 (>402)', 'UVB_UECM2 (>402)', 'UVA_JCM2 (>402)', 'UVA_UECM2 (>402)', 'UVR_JCM2 (>402)', $
'UVR_UECM2 (>402)', 'PAR_UECM2 (>402)', 'UVB_VS_TOT (>402)', 'UVA_VS_TOT (>402)', 'UVR_VS_TOT
(>402)', $
'UVB_VS_UVA (>402)', 'UVB_VS_PAR (>402)', 'UVBPAR_X_INC (>402)', 'TOT_X_INC (>402)', $
'*** Enhancement Effects', $
'UVA_%ENH', 'MAXIMUMPS_TRT', $
'*** Inhibition Effects', $
'UVR_%INHIB', 'UVB_%INHIB', 'UVA_%INHIB', 'UVB_VS_UVR (14-C)', 'UVA_VS_UVR (14-C)', $
'*** PAR Error Estimates', $
'VPS_%CORR', 'CLOSS_CCHLHR', $
'*** CHL Corrections', $
'CHLPS_%CORR', 'CLOSS_CCHLHR', $
'*** T-Corrected Production', $
'T295A_NGLHR', 'T295A_STDEV', 'T295A_N', 'T295B_NGLHR', 'T295B_STDEV', $
'T295B_N', 'T310_NGLHR', 'T310_STDEV', 'T310_N', 'T318_NGLHR', 'T318_STDEV', $
'T318_N', 'T328_NGLHR', 'T328_STDEV', 'T328_N', 'T341_NGLHR', 'T341_STDEV', $
'T341_N', 'T382_NGLHR', 'T382_STDEV', 'T382_N', 'T410_NGLHR', 'T410_STDEV', $
'T410_N', $
'*** T-NON-Corrected Production', $
'T295A_NGLHR', 'T295A_STDEV', 'T295A_N', 'T295B_NGLHR', 'T295B_STDEV', $
'T295B_N', 'T310_NGLHR', 'T310_STDEV', 'T310_N', 'T318_NGLHR', 'T318_STDEV', $
'T318_N', 'T328_NGLHR', 'T328_STDEV', 'T328_N', 'T341_NGLHR', 'T341_STDEV', $
'T341_N', 'T382_NGLHR', 'T382_STDEV', 'T382_N', 'T410_NGLHR', 'T410_STDEV', $
'T410_N', $
'*** T-Corrected Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-NON-Corrected Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-Corrected Production per Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-NON-Corrected Production per Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR' ]

```

```

yldisplay = widget_droplist(Abase, value = ylvariables, title = 'Define the Y1-Axis', $
event_pro = 'yldisplay_event')

```

;This defines the y2-axis though the data calling will be in the xdisplay event, based on what is chosen here

```

y2variables = [ '*** Treatment information', $
'EXPOSURE_HRS', 'FULLDAY_HRS', 'EXP_VS_FULLDAY', $
'*** External Light Exposure', $
'UVB_JCM2 (External)', 'UVB_UECM2 (External)', 'UVA_JCM2 (External)', 'UVA_UECM2 (External)', $

```

```

'UVR_JCM2 (External)', 'UVR_UECM2 (External)', 'PAR_UECM2 (External)', 'UVB_VS_TOT (External)', $
'UVA_VS_TOT (External)', 'UVR_VS_TOT (External)', 'UVB_VS_UVA (External)', 'UVB_VS_PAR (External)', $
'UVBPAR_X_INC (External)', 'TOT_X_INC (External)', $
'*** >295 Treatment Exposure', $
'UVB_JCM2', 'UVB_UECM2 (>295)', 'UVA_JCM2 (>295)', 'UVA_UECM2 (>295)', 'UVR_JCM2 (>295)', $
'UVR_UECM2 (>295)', 'PAR_UECM2 (>295)', 'UVB_VS_TOT (>295)', 'UVA_VS_TOT (>295)', $
'UVR_VS_TOT (>295)', 'UVB_VS_UVA (>295)', 'UVB_VS_PAR (>295)', 'UVBPAR_X_INC (>295)', 'TOT_X_INC
(>295)', $
'*** >310 Treatment Exposure', $
'UVB_JCM2 (>310)', 'UVB_UECM2 (>310)', 'UVA_JCM2 (>310)', 'UVA_UECM2 (>310)', 'UVR_JCM2 (>310)', $
'UVR_UECM2 (>310)', 'PAR_UECM2 (>310)', 'UVB_VS_TOT (>310)', 'UVA_VS_TOT (>310)', 'UVR_VS_TOT
(>310)', $
'UVB_VS_UVA (>310)', 'UVB_VS_PAR (>310)', 'UVBPAR_X_INC (>310)', 'TOT_X_INC (>310)', $
'*** >318 Treatment Exposure', $
'UVB_JCM2 (>318)', 'UVB_UECM2 (>318)', 'UVA_JCM2 (>318)', 'UVA_UECM2 (>318)', 'UVR_JCM2 (>318)', $
'UVR_UECM2 (>318)', 'PAR_UECM2 (>318)', 'UVB_VS_TOT (>318)', 'UVA_VS_TOT (>318)', 'UVR_VS_TOT
(>318)', $
'UVB_VS_UVA (>318)', 'UVB_VS_PAR (>318)', 'UVBPAR_X_INC (>318)', 'TOT_X_INC (>318)', $
'*** >328 Treatment Exposure', $
'UVB_JCM2 (>328)', 'UVB_UECM2 (>328)', 'UVA_JCM2 (>328)', 'UVA_UECM2 (>328)', 'UVR_JCM2 (>328)', $
'UVR_UECM2 (>328)', 'PAR_UECM2 (>328)', 'UVB_VS_TOT (>328)', 'UVA_VS_TOT (>328)', 'UVR_VS_TOT
(>328)', $
'UVB_VS_UVA (>328)', 'UVB_VS_PAR (>328)', 'UVBPAR_X_INC (>328)', 'TOT_X_INC (>328)', $
'*** >342 Treatment Exposure', $
'UVB_JCM2 (>342)', 'UVB_UECM2 (>342)', 'UVA_JCM2 (>342)', 'UVA_UECM2 (>342)', 'UVR_JCM2 (>342)', $
'UVR_UECM2 (>342)', 'PAR_UECM2 (>342)', 'UVB_VS_TOT (>342)', 'UVA_VS_TOT (>342)', 'UVR_VS_TOT
(>342)', $
'UVB_VS_UVA (>342)', 'UVB_VS_PAR (>342)', 'UVBPAR_X_INC (>342)', 'TOT_X_INC (>342)', $
'*** >381 Treatment Exposure', $
'UVB_JCM2 (>381)', 'UVB_UECM2 (>381)', 'UVA_JCM2 (>381)', 'UVA_UECM2 (>381)', 'UVR_JCM2 (>381)', $
'UVR_UECM2 (>381)', 'PAR_UECM2 (>381)', 'UVB_VS_TOT (>381)', 'UVA_VS_TOT (>381)', 'UVR_VS_TOT
(>381)', $
'UVB_VS_UVA (>381)', 'UVB_VS_PAR (>381)', 'UVBPAR_X_INC (>381)', 'TOT_X_INC (>381)', $
'*** >410 Treatment Exposure', $
'UVB_JCM2 (>402)', 'UVB_UECM2 (>402)', 'UVA_JCM2 (>402)', 'UVA_UECM2 (>402)', 'UVR_JCM2 (>402)', $
'UVR_UECM2 (>402)', 'PAR_UECM2 (>402)', 'UVB_VS_TOT (>402)', 'UVA_VS_TOT (>402)', 'UVR_VS_TOT
(>402)', $
'UVB_VS_UVA (>402)', 'UVB_VS_PAR (>402)', 'UVBPAR_X_INC (>402)', 'TOT_X_INC (>402)', $
'*** Enhancement Effects', $
'UVA_%ENH', 'MAXIMUMPS_TRT', $
'*** Inhibition Effects', $
'UVR_%INHIB', 'UVB_%INHIB', 'UVA_%INHIB', 'UVB_VS_UVR (14-C)', 'UVA_VS_UVR (14-C)', $
'*** PAR Error Estimates', $
'VPS_%CORR', 'CLOSS_CCHLHR', $
'*** CHL Corrections', $
'CHLPS_%CORR', 'CLOSS_CCHLHR', $
'*** T-Corrected Production', $
'T295A_NGLHR', 'T295A_STDEV', 'T295A_N', 'T295B_NGLHR', 'T295B_STDEV', $
'T295B_N', 'T310_NGLHR', 'T310_STDEV', 'T310_N', 'T318_NGLHR', 'T318_STDEV', $
'T318_N', 'T328_NGLHR', 'T328_STDEV', 'T328_N', 'T341_NGLHR', 'T341_STDEV', $
'T341_N', 'T382_NGLHR', 'T382_STDEV', 'T382_N', 'T410_NGLHR', 'T410_STDEV', $
'T410_N', $
'*** T-NON-Corrected Production', $
'T295A_NGLHR', 'T295A_STDEV', 'T295A_N', 'T295B_NGLHR', 'T295B_STDEV', $
'T295B_N', 'T310_NGLHR', 'T310_STDEV', 'T310_N', 'T318_NGLHR', 'T318_STDEV', $
'T318_N', 'T328_NGLHR', 'T328_STDEV', 'T328_N', 'T341_NGLHR', 'T341_STDEV', $
'T341_N', 'T382_NGLHR', 'T382_STDEV', 'T382_N', 'T410_NGLHR', 'T410_STDEV', $
'T410_N', $
'*** T-Corrected Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-NON-Corrected Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-Corrected Production per Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR', $
'*** T-NON-Corrected Production per Chlorophyll', $
'T295A_NGLHR', 'T295B_NGLHR', 'T310_NGLHR', 'T318_NGLHR', 'T328_NGLHR', 'T341_NGLHR', $
'T382_NGLHR', 'T410_NGLHR' ]

```

```

y2display = widget_droplist(Abase, value = y2variables, title = 'Define the Y2-Axis', $
event_pro = 'y2display_event')

;rbase = widget_base(base, /column)

;rbase2 = widget_base(rbase, /column)

;rbasel = widget_base(rbase, /row, scr_ysize=600)
;zero_vals1 = strarr(83)
;table1 = widget_table(rbasel, alignment=1, $
; value = zero_vals1, column_labels = ['Drifter'], $ /scroll, $;
; xsize=1, ysize=83, scr_xsize=320, ;$
; column_widths = 265, uname = 'drifter_names')

;zero_vals2 = fltarr(3,83)
;table1 = widget_table(rbasel, alignment=1, /editable, /scroll, $
; value = zero_vals2, column_labels = ['X Axis', 'Y1 Axis', 'Y2 Axis'], $
; xsize=3, ysize=83, scr_xsize=299,$
; column_widths = 80, uname = 'table_vals', event_pro='table_edit')

;Displays the days and data chosen into the table in the startup GUI
chartit = widget_button(abase, ysize=40, value = 'display data in chart', event_pro = 'chartit')

;This section lets you define the type of analysis you want to do
plottools =['2D XY Scatter', '2D XYY Scatter', '2D XYY Stacked Scatter', '3D XYZ Scatter', $
'3D XYZ Contour']

plotwidget = widget_droplist(Abase, value = plottools, title = 'Choose method of analysis', $
event_pro = 'plottools_event')

plotit = widget_button(abase, ysize=40, value = 'Display the figure', event_pro = 'plotit')

resetDrifterbutton = widget_button(abase, value='Reset All', $
ysize = 40, event_pro='reset_treatment_event')

resetTreatbutton = widget_button(abase, value='Reset Drifter Only', $
ysize = 40, event_pro='reset_drifter_event')

Exitbutton = widget_button(abase, value = 'Exit', ysize=40, $
event_pro='exit')

rbase = widget_base(base, /column)

;*****
rbase2 = widget_base(rbase, /column)

device, get_screen_size = ss
print, ss
draw = widget_draw(rbase2, xsize=ss[0]*0.5, ysize=ss[1]*0.5, $
graphics_level=2, /expose_events, /motion_events, $
/button_events, event_pro= 'test_trackball_draw3')

widget_control, base, /realize
widget_control, draw, get_value = owindow

```

```

x = RANDOMN(seed, 83) ;Make 50 normal x, y points.
y = RANDOMN(seed, 83)
z = EXP(-(x^2 + y^2)) ;Make the Gaussian.

; Making a 2D array from the x,y,z data.
triangulate, x, y, tr, b ;obtain triangulation.
data = trigrid(x, y, z, tr, [.2,.2], $
xgrid = xg, ygrid = yg);, extrapolate = b, /quintic)

;Create a model for the graphics; add to the view.
oView = OBJ_NEW('IDLgrView')
oModel = OBJ_NEW('IDLgrModel')
osurface = obj_new('IDLgrSurface', data, color = [0,255,0], style = 2)
olight = obj_new('IDLgrLight', location = [-1,1,1], type = 1)
omodel -> add, osurface
omodel -> add, olight
oview -> add, omodel

osurface -> getproperty, xrange = xr, yrange = yr, zrange = zr
  xc = norm_coord(xr)
  yc = norm_coord(yr)
  zc = norm_coord(zr)
  xc[0] = xc[0] - 0.5
  yc[0] = yc[0] - 0.5
  zc[0] = zc[0] - 0.5
  osurface -> setproperty, xcoord = xc, ycoord = yc, zcoord = zc

otrack = obj_new('trackball', 0.2*ss, 0.2*ss[0])

xmax = xr[1]
ymax = yr[1]
zmax = zr[1]

xaxis = obj_new('idlgraxis',0, range = [0, xmax])
yaxis = obj_new('idlgraxis',1, range = [0, ymax])
zaxis = obj_new('idlgraxis',2, range = [0, zmax])

xaxis -> setproperty, xcoord = xc, ycoord = yc, zcoord = zc
yaxis -> setproperty, xcoord = xc, ycoord = yc, zcoord = zc
zaxis -> setproperty, xcoord = xc, ycoord = yc, zcoord = zc

omodel -> add, xaxis
omodel -> add, yaxis
omodel -> add, zaxis

owindow -> draw, oview

rbase1 = widget_base(rbase, /row, scr_ysize=280)
zero_vals1 = strarr(83)
table1 = widget_table(rbase1, alignment=1, $
  value = zero_vals1, column_labels = ['Drifter'], $ /scroll, $
  xsize=1, ysize=83, scr_xsize=320, $
  column_widths = 265, uname = 'drifter_names')

zero_vals2 = fltarr(3,83)
table1 = widget_table(rbase1, alignment=1, /editable, /scroll, $
  value = zero_vals2, column_labels = ['X Axis', 'Y1 Axis', 'Y2 Axis'], $
  xsize=3, ysize=83, scr_xsize=299,$

```

```

column_widths = 80, uname = 'table_vals', event_pro='table_edit')

widget_control, base, /realize

;Variable definitions column

;All of these variables are defined for data calling purposes
drifter = fltarr(83)
drifter_name = strarr(83)
drifter_names = strarr(83)
table = fltarr(3,83)
chartdata = fltarr(3,83)
chosendata = fltarr(83)
chosendatax = fltarr(83)
chosendatay1 = fltarr(83)
chosendatay2 = fltarr(83)
figtype = bytarr(1)
xword1 = strarr(1)
xword2 = strarr(1)
xword3 = strarr(1)
ylword1 = strarr(1)
ylword2 = strarr(1)
ylword3 = strarr(1)
y2word1 = strarr(1)
y2word2 = strarr(1)
y2word3 = strarr(1)
datamx = fltarr(83)
datamy1 = fltarr(83)
datamy2 = fltarr(83)
osurface1 = ('idlgrsurface')

depth = [5,5,5,5,5,5,5,18,18,5,5,5,18, $ ;this is SUPACC1 depths (13)
        5,5,5,5,59,518,5,5,9,18,5,5,5,5,518,5,9,5,5,18,18,185,185,18, $ ;This is the SUPACC2 depths
(24)
        5,5,9,18,5,5,5,18,5,9,18,5,5,5,5,9,18,5,9,18,55,59,518,9,5,18, $ ;This is the SUPACC3
depths (27)
        5,5,9,5,5,5,5,9,18,5,9,18,5,185,189,18,95,9,918] ;THIS is the SUPACC4 depths (18)

Time = [4,4,1,4,4,1,3,4,4,4,4,1,1, $ ;SUPACC 1 time 1=AM, 2=PM 3=Midday 4=FULL DAY
        1,4,4,3,4,4,3,4,4,4,1,2,4,4,4,3,3,3,3,3,3,3,3,3, $ ;SUPACC2 time
        3,4,4,4,1,2,4,1,3,3,3,1,2,4,3,4,4,4,3,3,3,3,3,3,3,3, $ ;SUPACC3 time
        3,3,3,1,2,4,4,4,4,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3] ;SUPACC4 time

location = [1,1,1,1,1,1,1,1,1,2,2,2,2, $ ;SUPACC1 Local 1 = gav, 2,3 &4 = SC 5 = SR
           1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2, $ ;SUPACC2 Local
           1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2, $ ;SUPACC3
Local
           3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4] ;SUPACC4 Local

ExpVsDay = [0.73, 0.83, 0.38, 0.76, 0.81, 0.32, 0.57, 0.83, 0.81, 0.79, 0.81, 0.43, 0.43, $
           0.49, 0.86, 0.94, 0.34, 0.90, 0.93, 0.34, 0.87, 0.90, 0.88, 0.44, 0.44, 0.95, 0.96, 0.96, $
           0.35, 0.35, 0.32, 0.56, 0.32, 0.56, 0.21, 0.39, 0.49, 0.48, 0.89, 0.88, 0.91, 0.55, 0.50, $
           1.05, 0.56, 0.51, 0.49, 0.52, 0.55, 0.46, 0.96, 0.38, 1.05, 1.01, 1.06, 0.44, 0.41, 0.45, $
           0.57, 0.60, 0.58, 0.57, 0.57, 0.57, 0.46, 0.80, 0.76, 0.50, 0.52, 0.89, 1.11, 1.04, 1.13, 0.59, $
           0.48, 0.59, 0.52, 0.57, 0.53, 0.58, 0.51, 0.50, 0.53]

;The state function defines all variables that will be called in other programs within this tool set
state = {drifter:drifter, depth:depth, time:time, location:location, drifter_name:drifter_name, $
drifter_names:drifter_names, table:table, ylvariables:0L, y2variables:0L, xvariables:0L, $
chosendatax:chosendatax, chosendatay1:chosendatay1, chosendatay2:chosendatay2, chartdata:chartdata,
$

```

```
figtype:figtype, xword1:xword1, xword2:xword2, xword3:xword3, y1word1:y1word1, y1word2:y1word2, $
ylword3:y1word3, y2word1:y2word1, y2word2:y2word2, y2word3:y2word3, expvsday:expvsday,
datamx:datamx, $
datamy1:datamy1, datamy2:datamy2, otrack:otrack, $
osurface1:osurface1, omodel:omodel, oview:oview, olight:olight, owindow:owindow, $
osurface:osurface, yc:yc, xc:xc, zc:zc}

pstate = ptr_new(state, /no_copy)
widget_control, base, set_uvalue=pstate

xmanager, 'reap', abase

return
end
```