DEVise User Manual

The DEVise Development Group
devise@cs.wisc.edu

March 26, 1997

1 Version

This manual is for version 1.3.4 of DEVise.

2 Introduction

DEVise (Data Exploration and Visualization) is a system that allows users to easily develop, browse, and share visual presentations of large tabular datasets (possibly containing or referencing multimedia objects) from several sources.

3 Installation and Configuration

3.1 Getting the Correct `tar` File

DEVise is available at: ftp://ftp.cs.wisc.edu/pub/devise/. In this directory, there should be several subdirectories containing various versions of DEVise. These subdirectories are named something like Devise-1.3.4, where the number is the version of DEVise that the directory contains. In general, you should get the latest available version of DEVise.

Each subdirectory contains the following gzipped tar files:

- hpux-hppa.tar.gz (HPUX)
- solaris-intel.tar.gz (Solaris on Intel)
- solaris-sparc.tar.gz (Solaris on SPARC)
- sunos-sparc.tar.gz (SunOS on SPARC)

Get the file appropriate for your system architecture.

3.2 Extracting Files from the `tar` Archive

Go to the directory in which you want to install DEVise (this directory will be referred to as `<devisehome>` in the rest of this document). Then enter the following command:

`tar xvfz <tar file>`

where `<tar file>` is replaced by the actual name of the file that you got from the FTP site.

Depending on the version of `tar` that you have, the above command may not work. If this is the case, you will have to do the unzipping and untarring in two separate steps:

`gunzip -c <tar file> | tar xv`

Once you have untarred the archive, you should have subdirectories such as `bin`, `cache`, `dat`, etc., in the `<devisehome>` directory. To make sure that DEVise was installed correctly, cd to the `run` directory and enter the command `devise` (or `./devise` if `.` is not in your path). This should bring up the DEVise Control Panel (see Figure 1).

3.3 Setting up a User to Run DEVise

In setting up a user to run DEVise, there is one main decision that must be made. That is, should the user have their own sessions, schemas, and data, or should they share the global sessions, schemas, and data? If you will have a large number of users running DEVise, you should think carefully about this. If users share the global sessions, etc., it will obviously be easier to share visualizations among various users. On the other hand, if each user has their own sessions, there is less chance of running into problems from one user modifying something that another user depends on (a data source definition, for example).

If a user is to have their own session files, schemas, and data, they need the appropriate directories. The easiest way to create these directories is to recursively copy the `dat`, `schema`, and `session` directories created when you untarred the distribution file. You should do this because some files in these directories must already exist for DEVise to run properly.

3.3.1 Path Names and Environment Variables

In order to run DEVise, the following environment variables must be defined correctly:

- DEVISE_DAT
- DEVISE_CACHE
- DEVISE_SCHEMA
- DEVISE_SESSION
- DEVISE_LIB
- DEVISE_TMP
- DEVISE_WORK
- TK_LIBRARY
- TCL_LIBRARY

There are two ways to set these environment variables: you can run DEVise using the `devise` script in `<devisehome>/run`, or you can set them in the user's environment (`.cshrc` file, etc.). See the `<devisehome>/run/devise` script for examples of how these environment variables should be set. Depending on how you have configured your system and installed DEVise, you may have to set some of the paths differently.

3.3.2 Running with the `devise` Script

If users share the same session files, etc., they can all run DEVise by changing to the directory `<devisehome>/run` and entering the command `devise` or `./devise`. Alternatively, you could modify the devise startup script to define DEVISE_ROOT as an absolute, rather than a relative, path. This allows users to run DEVise from any directory. For this to work properly, the schema and data file names in the data sources must be changed from relative to absolute paths as well.

If users do not share sessions, schemas, and data, each user must have their own copy of the `devise` script that sets the DEVISE_DAT, DEVISE_SCHEMA, and DEVISE_SESSION environment variables appropriately.

3.3.3 DEVise Executables

When you run the `devise` script, the executable that is actually run is `<devisehome>/bin/devise`. The `<devisehome>/bin` directory also contains the executables `deviseb`, `devisec`, and `devised`. `deviseb` is batch-mode DEVise; `devisec` is the DEVise client; and `devised` is the DEVise daemon (server). If you have `devised` running, and then run `devisec`, the client will connect to the server, and will run just like the monolithic DEVise, except that the GUI is a separate process from the processing and display.

Some architectures also have an executable called `Tasvir` (Transform, Arrange, Scale, and View Incremental Rasters). DEVise and Tasvir can be used in combination to display images as symbols in DEVise.

3.3.4 Customization with a DEVise Resource File

When you start DEVise, it looks for a resource file to customize various program settings. It looks for the resource file in the following sequence:

1. toupper($argv0)_RC, if the environment variable toupper($argv0)_RC exists
2. .$argv0_rc in the current directory
3. $argv0.rc in the current directory
4. devise.rc in the current directory
5. $HOME/$argv0.rc
6. $HOME/.devise.rc

For example, if you used a script called "mydev" to run DEVise, DEVise would search for the following resource files:

1. $MYDEV_RC, where MYDEV_RC is an environment variable containing a valid file path
2. .mydev_rc in the current directory
3. mydev.rc in the current directory
4. devise.rc in the current directory
5. mydev.rc in your home directory
6. .devise.rc in your home directory

DEVise uses the first resource file that it finds in the sequence.
Here is a sample DEVise resource file:

```
set viewDefault(int)     "{xlow 0} {xhigh 2300} {ylow 0} {yhigh 100}"
set viewDefault(float)   "{xlow 0} {xhigh 100} {ylow 0} {yhigh 100}"
set viewDefault(double)  "{xlow 0} {xhigh 100} {ylow 0} {yhigh 100}"
set viewDefault(date)    "{xlow {Jan 1 1992 00:00:00}} \
                          {xhigh {Dec 31 1992 23:59:59}} \
                          {ylow {Jan 1 1992 00:00:00}} \
                          {yhigh {Dec 31 1992 23:59:59}}"

# Specify user mode: 1 for user and 0 for superuser
set UserMode 0
```

You don't need a DEVise resource file–it's just a way to customize some of DEVise's settings if you want to.

4 Tutorial

The tutorial is intended to give you a feel for using the basic functionality of DEVise. The tutorial consists of the following steps:

1. Starting DEVise: The Control Panel
2. Importing and Visualizing Data
3. Opening a Session

4. Visualization: Session Windows and Views
5. Links and Cursors
6. View Manipulation
7. View Statistics
8. General Session Operations

4.1 Starting DEVise: The Control Panel

Start DEVise from the command line by typing `devise` from the `run` directory. A window entitled *DEVise* will appear. This is the *Control Panel*, the nerve center of DEVise (Figure 1). The Control Panel contains a six-menu menubar and five buttons, from which most features of DEVise may be accessed.

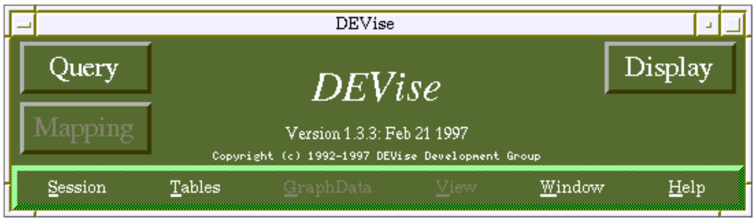

Figure 1: Control Panel

4.2 Importing and Visualizing Data

4.2.1 Creating a Schema

To create a physical schema, select `Physical...` from the `Schema` submenu of the `Tables` menu. Then left click on the New button in the Physical Schemas window. This shows the Schema Description window. Fill in the appropriate information in this window (see section 6.2), and left click the Save button when you are done.

See sections 6.2, 6.3, and 7 for more detailed information on schemas.

4.2.2 Defining a Data Source

Now that you have created a schema, you can tie your file or web URL and the schema together to create a Data Source. Select New Table from the Tables menu, which brings up the Define Data Stream window. First select the type of data source (UNIXFILE or WWW).

If your data source is a UNIXFILE, click the Select button to the right of the Key entry. This brings up a file select box that you can use to select the appropriate file. Once you've done that, the file select box will be shown again to allow you to select a schema file. At this point, you can either change the Display Name of the Data Stream, or simply click OK if the default name is acceptable. You can leave the default values for Evaluation Factor and Cache Priority unchanged, at least for now.

A WWW data source is similar to a UNIXFILE, the main difference being that you enter the web URL in the Info/Command area.

See sections 6.1 and 8 for more detailed information on defining data sources.

4.2.3 Visualizing the Data Source

Now that you have defined your Data Stream, you can visualize it (this is what you really want to do, right?). Select `Visualize/Edit` from the `Tables` menu, and select the data stream you just defined. You can either double click on it, or single click and then click the Visualize button.

Once you've done this, you'll see the Define Visualization window. There are a number of options in this window. The most important of these are:

- Selecting the window in which to visualize this data (no default)
- Selecting the attribute for the X values (defaults to record ID)
- Selecting the attribute(s) for the Y values (no default)
- Selecting the type of view (defaults to bar chart)

When you have made the appropriate selections, click the OK button. At this point, the view(s) needed to visualize the data are created.

See section 9 for more detailed information on visualizing data sources.

4.2.4 Editing the Mapping

Once you have visualized your data, you will probably want to change how it is displayed. You can do this using the Edit Mapping window (see Figure 8). You can access the Edit Mapping window by clicking the `Mapping` button on the Control Panel, or by selecting `Edit Mapping` from the `GraphData` menu (you must select a view for these options to be enabled).

See section 6.4 for more detailed information on editing mappings.

4.2.5 Saving Your Visualization

Saving your visualization will allow you to access it again, without having to repeat all of the steps you needed to do to create it. To save your visualization (as a "session" file), select `Save as` from the `Session` menu. This will bring up the File Select window (see Figure 2). Enter the filename to which you want to save the session (you should enter a name ending in ".tk").

4.3 Opening a Session

The "documents" in DEVise are called *sessions*. For now, think of a session as a collection of windows or views which display data in a variety of formats from one or more perspectives.

To work on an example session, select the Session button in the Control Panel with the mouse (or Alt-S from the keyboard). From the resulting Session menu, select the Open option. A dialog window entitled *File select box* will appear, displaying session files present in the current working directory (Figure 2). (Session files are distinguished by the `tk` extension.)

For this tutorial, double-click the `demo/` directory. The session files in this subdirectory will be displayed in the *File select box*. From the list of files, select `demo.tk` then the OK button. (Note: Under most selection opportunities in DEVise, double-clicking the option performs the same function as clicking the option then selecting the OK button.)

Once opened, the `demo.tk` session is presented in three windows. These windows contain the previously constructed *visualization* of the session.

4.4 Visualization: Session Windows

The three session windows entitled *Global view*, *Trade Price* and *Trade Volume* appear on the screen when the session is opened (Figure 3). Like other X-windows windows, each may be moved, resized, closed, etc. Alteration of a window will cause the corresponding appropriate adjustment(s) to the window's content.

The data for the tutorial session are the trade price and volume of several company stocks over a 30-month period. Inside the three session windows, the nine graphs are the *views* for the session. Each view displays a subset of the session data.

All view-oriented actions initiated from the Control Panel (or elsewhere) operate on the *current* view. The current view is distinguished from other views by the presence of a vertical highlighting bar to the left

Figure 2: File Select Window

of the view or a rectangle surrounding the view. A view can be made *current* by selecting it (clicking the mouse button while the cursor is within the view boundary).

Zooming and scrolling are two operations which may be performed on a selected view. First, make the IBM price view current by selecting the IBM Price view in the Trade Price window. Second, press (or press and hold down) the key corresponding to the desired operation as listed below:

- 4 and 6: scroll left and right
- 8 and 2: scroll up and down
- 7 and 9: zoom in and out (X-axis)
- 1 and 3: zoom in and out (Y-axis)

For example, pressing the 6 key followed by the 1 key will adjust the IBM Price view to display more recent prices valued between 30 and 70. (Note: On some systems, the number lock must be on for the numeric keypad to operate correctly.)

Because the X and Y ranges of data are bounded, a combination of zoom and scroll is often required to focus on a particular region. This is particularly true for some charts and graphs which display only positive-valued data. For example, the Price and Volume bar charts displayed in the current session contain

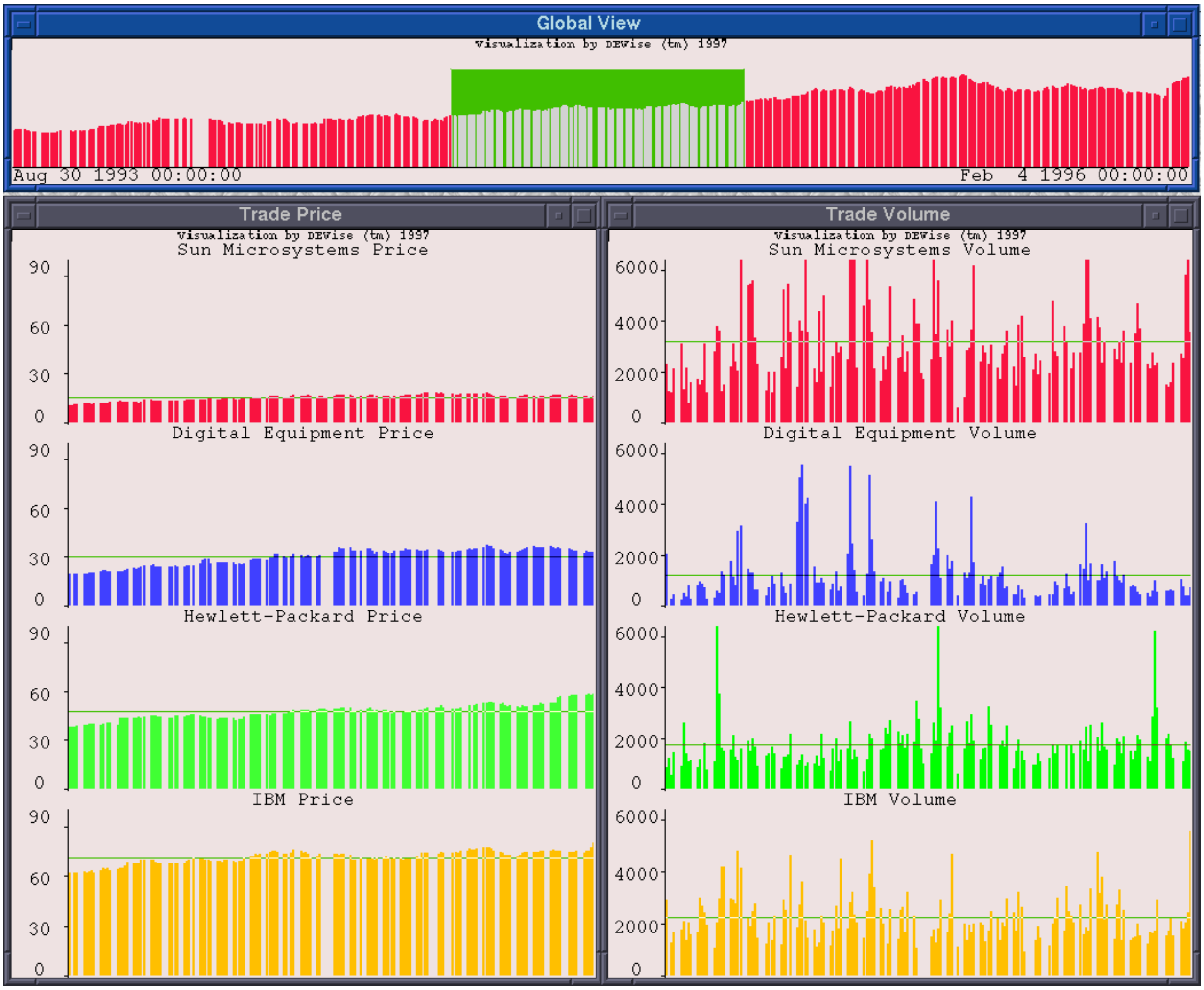

Figure 3: Data Display Windows

no negative-valued information. To zoom in on positive values of Y only, combine scrolling up/down and zooming in/out. For instance, to zoom in in the Y dimension, press the 1 key followed by the 8 key. To zoom out in the Y dimension, press the 2 key followed by the 3 key.

While scrolling and zooming, the other eight views also change—the other three Price and four Volume views scroll, and the colored block in the Global view window moves. Such interdependencies between different views of the same data are called *Links* and *Cursors*.

4.5 Links and Cursors

In the `demo.tk` session, all the views except the global view are *linked* together on their X-axis. What this means is all views display data within the same range of X values. Thus the stock prices and volumes are displayed for all four companies during the same time period (using trading dates as the X values).

Two or more views may be linked along the X axis, the Y axis, or on a color, shape, etc. Linking helps to you to synchronize browsing of multiple views, over a fixed range of an attribute.

The colored rectangle in the Global view (window) is a *cursor*. A cursor visually encompasses the rectangular region of data displayed in another view. The X and Y ranges in the eight Price and Volume views are the same as the X and Y range of the cursor in the Global view. Think of a cursor as a zoom box: the part of a view which has been selected and “zoomed in on” in a different view(s).

The connection between views established with a cursor is maintained as the views are manipulated. For example, when you zoom in on the X axis (using keypad 7) in the IBM Price view, the cursor becomes smaller (as it encompasses a smaller range of X values).

The cursor can also be moved by clicking under the X axis in the Global view window. The cursor is repositioned such that it is centered at the point at which the mouse was clicked. Because the views are connected via the cursor, the displayed range of data changes in the other eight views whenever the cursor is moved. Thus interesting patterns on the global view can be analyzed by moving the cursor to that point and zooming in on the view. Similarly, the cursor is enlarged as the ranges in the Trade Price or Trade Volume is changed using the keypad scroll and zoom operations.

4.6 Manipulating Views

In addition to scroll and zoom, views can be manipulated in many ways from the Control Panel. In the Control Panel, the `Window` menu contains options for moving views and hiding/showing views. The `View` menu contains options for adjusting view appearance.

4.6.1 Moving Views

To swap the position of two views in a single window, select the views to swap, in sequence, then select the `Swap Views` menu option in the `Window` menu.

To move a view to a different window, select the view to move, then select the `Move View` menu option. A dialog box will appear containing a list of possible destination windows. Select the desired destination window—or the new button to create a new window—followed by the OK button to complete the view migration.

4.6.2 Hiding/Showing Views

An unwanted view can be discarded from a window using the `Remove View` menu option. For example, select the IBM Price view in the Trade Price window. Use the `Remove View` menu option to remove the view. The view will disappear from the window; however, DEVise maintains memory of the view and its contents.

To bring back a view, select the `Bring View Back` menu option. A window containing a list of *all* views in the session will appear. Select the "Close vs Date4" entry followed by the OK button. Another dialog box will appear requesting a location (window) for the returned view. Select the "Trade Price" entry to restore the IBM Price view to its former location.

4.6.3 View Appearance

Like typical graphing applications, the features of a view's appearance are adjustable. The X and Y axis label values may be made visible or invisible; the view title may be edited; and the background color may be selected.

To toggle the presence of X labels in a view, select the `Toggle X Axis` menu option in the `View` menu. Likewise, the presence of Y axis labels requires selection of the `Toggle Y Axis` menu option.

4.6.4 View Title and Colors

The `Title` menu option in the `View` menu creates a dialog box for editing the title of the current view. To delete the title, select the delete button. In this special case, the current view indicator is a bar to the left of the view, in contrast to the usual rectangle surrounding the view.

The `Foreground...` and `Background...` menu options in the `View` menu allow customization of foreground (pen) and background (field) colors displayed in the current view.

The other, more advanced `View` menu options are described later.

4.7 View Analysis Tools

Two mechanisms to assist in view analysis may be invoked via options in the `View` menu: Statistics and Kiviat Graph.

4.7.1 View Statistics

You may be curious about the presence of a line across each view. This is the *statistical mean line* for the view over the visible X range. To set or reset the line, select the `Statistics` menu option in the `View` menu. A dialog box will appear with several options for display of common statistical values, including the minimum, maximum, and mean Y values of displayed data (Figure 4). The "line" option fits a linear curve to the displayed data. In addition, the region of 85%, 90%, or 95% confidence in a view may be highlighted. To create the desired statistical indicators in the current view or in all views, select the "Apply Current" or "Apply All" buttons, respectively.

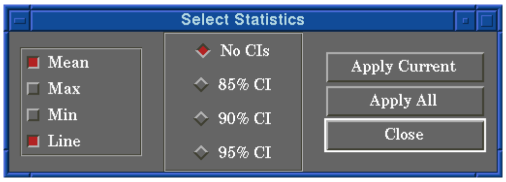

Figure 4: Statistics

4.7.2 Kiviat Graph

Use the Kiviat graph tool to analyze the views in a window. Kiviat graph options are accessible from the `Kiviat Graph` submenu of the `View` menu. (The submenu may be torn off for repeated use.) The graph can display the result for mean, maximum, minimum, or count as selected from the menu. When the `Apply` menu option is selected, the appropriate Kiviat graph is generated and displayed in a new window (Figure 5).

The graph is computed for the displayed X range in the current window. Clicking the middle mouse button inside the Kiviat graph displays a table with the statistical values graphed.

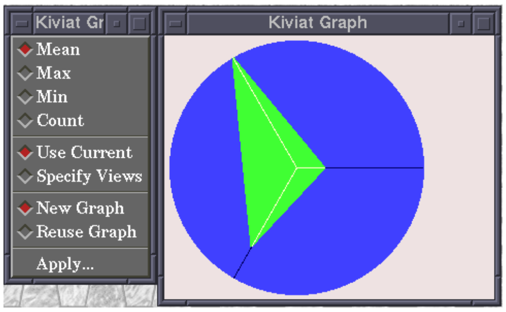

Figure 5: Kiviat Graph

4.8 General Session Operations

DEVise maintains a list of view and window manipulations to facilitate undo operations as needed, such as when a zoom or scroll is inadvertent or exploratory. To access this list and return to a previous data display state, select the `History` menu option in the `Help` menu. A dialog box will appear containing display states viewed since the session was opened (Figure 6), from which the desired previous state may be determined.

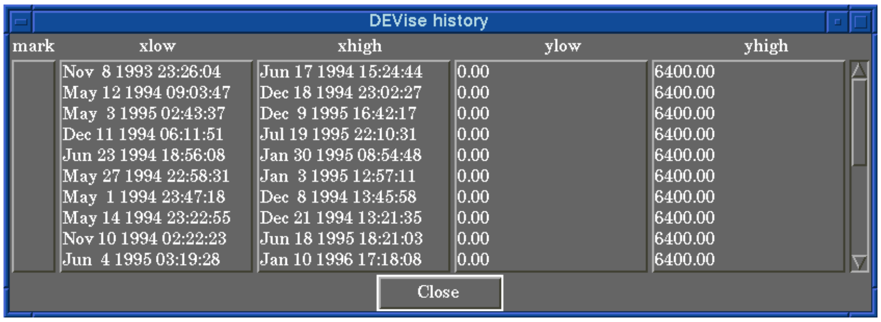

Figure 6: History Window

The button labeled Display in the Control Panel is useful when resizing windows. When a large data set is being visualized, DEVise will update and redisplay window contents while the window is being resized, degrading performance significantly. Toggle the Display button to change from Display mode to Layout mode, in which view contents are not redrawn. Once the resize process in complete, toggle the button back to Display mode to refresh the view contents and continue the session.

To conclude a session use the `Close` menu option in the `Session` menu. To exit from the program use the `Quit` menu option.

This completes the tutorial of DEVise. For more in depth information and details on the operation of other features, consult the later sections of this document.

5 General Operation

DEVise operations are performed on documents called *sessions*. A session is a set of data streams with GData mappings into views. After creation of a session, the user imports data, then explores and edits mappings by manipulation of views, window styles, etc.

When saved, a session file can be used to recreate an entire visualization at a later time, including the data streams visualized. To save a visualization without its associated data streams, a *template* file is created. Templates allow visualization of multiple sets of data streams while eliminating tedious, redundant construction of views and mappings by the user.

The operations on sessions and templates are described next followed by several general operations.

5.1 Session Operations (Open, Save, Save As, Close)

Sessions are the fundamental document type in DEVise, and as such may be opened, saved, and closed. Session operations may be selected in the `Session` menu in the Control Panel. All operations perform as expected: Save overwrites the old version, filenames are prompted as needed, etc.

5.2 Template Operations (Open As Template, Save As Template)

DEVise includes all data streams associated with a session when performing session file operations. A session may be saved without data streams by creating a template. When opened, a template may be joined with data streams of the same schema to create a new visualization. Although the data presented is dependent on the associated streams, the environment—mappings, views, windows. etc.—is that contained in the template.

Templates may be saved and opened using the `Session` menu in the Control Panel. A session file may be opened as a template, in which case DEVise will prompt for data streams to associate with the template. The number and schemas of data streams chosen must match those of the original session.

5.3 Export Template Operations (Export Template, Merge Template)

If you want to send a visualization to someone else ("export") it, you may want to save it as an exported template rather than as a regular session file. There are two forms of exported templates (with and without data).

Which form you want to use for saving a session depends partly on how much your DEVise setup has in common with the person with whom you want to share your visualization. First, some background. A normal session file contains information about the views, windows, and so on, and also references to the data sources and schema files. An exported template without data contains internally everything in a session file, plus the schema and data source definitions. An exported template with data contains all of the above, plus the data itself.

If you save a visualization as a normal session file, you can only share the visualization with someone who is running DEVise with the same data source catalog and schema files that you are. If you use an exported template without data, you can share your visualization with anyone who has the data file(s) mounted on the same path that you do (they can have their own data source catalog and schema files). If you use an exported template with data, anyone running DEVise can view your visualization.

When you have opened an exported template, its internal data source definitions take the place of those in your normal data catalog (meaning that you will not have access to your normal data sources). To add the template's definitions to your catalog, you have to "merge" the template. This is done by selecting `Merge Template` from the `Session` menu. Once you have merged the template, you will once again be able to access the data source definitions in your data catalog.

5.4 Printing

Data and window images may be printed or exported to different print and file formats. Print and export operations are accessible from the `Session` menu in the Control Panel. Upon selecting the print option, a dialog box will present printing options with a choice of whether to print all windows or the current selected window only.

A window image may be saved as a GIF, PostScript, or EPS file. In the print dialog box, select the "To File" button and one of the three file formats from the popup menu.

5.5 Quitting DEVise

To exit DEVise, use the `Quit` menu option in the `Session` menu. DEVise does not confirm whether to save any open, modified sessions. It confirms only that quitting is desired—it's up to you to be sure you don't have any unsaved changes that you want to save.

6 Concepts

This section explains the concepts behind DEVise in depth. DEVise is a "data extraction and visualization" tool, and has the primary purpose in assisting a human operator to discover patterns in data sets by free exploration of the data from varied graphical perspectives. The process of promoting raw file data to a specific graphical presentation is conceptually layered as follows:

- Data Streams
- Physical Schemas
- Logical Schemas
- Graphical Data
- Views and Windows

Additional concepts behind DEVise include:

- View Statistics
- Sessions and Templates

6.1 Data Streams

A *data stream* is a sequence of records to be visualized using a particular schema. For example, the list of daily stock prices for a company is a data stream. Data streams may come from different sources, such as a UNIX file, magnetic tape, or a database query output stream.

Each data stream is associated with a data source and a schema. The schema describes the layout and data fields of records in the stream, as well as a key to uniquely identify the stream. For a database stream, a query can be given as input; the query results are computed and cached as a data stream.

The data streams definition and edit operations in DEVise are accessed from the `Tables` menu, by selecting the `New Table` and `Visualize/Edit` options, respectively. There are numerous data streams which accompany DEVise, descriptions of which follow.

To use a data file, select the `New Table` menu option. A dialog box entitled "Define Data Stream" will appear. Selecting the topmost "Select" button will create a file selection dialog for picking a key.

6.1.1 Unix File

The data fields of a normal UNIX file data stream are separated by a standard delimiter as specified in the schema. An ASCII file containing $sin(t)$ and $cos(t)$ might contain the following records (or tuples):

```
# time   sin      cos
0.000000 1.000000 0.000000
0.017453 0.999848 0.017452
0.034906 0.999391 0.034898
0.052358 0.998630 0.052334
...
```

The output of any program that generates similar sequences of data can be visualized in DEVise as a UNIXFILE stream. The schema defines the delimiter (here a blank space) and comment character (here #), and also assigns the meaning (name) to the various record fields.

6.1.2 WWW Data

Any World Wide Web source can feed a WWW data stream. The URL (Uniform Resource Locator) can use the `ftp` or `http` protocol. Data at the URL is fetched from the WWW and cached locally. This method offers the advantage of directly using the WWW data without additional effort. As for the UNIXFILE type, the stream delimiter and comment characters must be specified in the schema. The key is assigned by the user and must be a unique name within this stream type.

6.2 Physical Schemas

A schema file describes the format/layout of an input data stream. It conveys the name, type, and range of attributes, characters that separate the attributes in the file, characters that define comment lines, and optional attribute range information. More than one file type can be imported into DEVise, each having its own schema file.

The physical schema describes the actual layout of the input stream with all its attributes. Every data stream is associated with a physical schema, and the schema is used by DEVise to extract data values from a data stream. To browse the physical schema, use the `Physical...` menu option in the `Schema` submenu of the `Tables` menu.

For our example file, the schema file looks like:

```
type Sensor ascii
comment #
whitespace ' '
attr time double hi 1000 lo 0
attr sin double hi 1 lo -1
attr cos double hi 1 lo -1
```

The first line names the file type: Sensor. All Sensor files have data stored in the same format. The second line tells DEVise to ignore lines that start with '#'. The third line tells DEVise to treat spaces as whitespace in the input stream and as a separator between field values. The remaining lines describe attribute names, types, and ranges.

There is an important difference between "separators" and "whitespace". If you specify a separator character or characters, each separator delimits a field. In other words, if you have two consecutive separators, the input will be interpreted as having a blank or zero value for a field. However, any number of whitespace characters are considered to be a single delimiter. In general, it is best to use whitespace rather than separators.

6.2.1 Attribute Name

An attribute name identifies a particular field of a record. In the schema defined above, sin is the name of the first attribute. This name is used when defining attributes of X and Y axes in views.

6.2.2 Attribute Type

An attribute may be one of the following five types: double, int, string, float, or date. Each type has its ubiquitous meaning and representation, except date which is the Unix time/data type. In the example above, sin is of type double. Note that length *must* be specified for strings. For ascii data, this length is the maximum length of any string in that field; for binary data, all strings must be exactly the length specified. *The string length must include one extra character for the string terminator.*

6.2.3 Attribute Value Range

The attributes may be defined to have a range bounded by minimum and maximum values. When DEVise displays field values, the range values displayed are calculated using the attribute ranges. For example, for an displayed attribute with value range between 0 and 200, DEVise chooses to display exactly that range. If the attribute's range in undefined, DEVise uses the range 0 to 100 by default.

6.2.4 Sorted vs Unsorted

Records in a stream may be sorted on a field. The specification of a sort key may be specified in the physical schema. DEVise uses this information to optimize data access and display. Moreover, data display in interleaved by value to achieve rapid partial display of data. This allows perusal an outline of the plotted data as the views update, particularly useful for "dense" data.

6.3 Logical Schemas

A logical schema defines a logical view on a physical schema, in so doing restricting the viewed range of an attribute in the schema.

In the COMPUSTAT database there are 350 attributes. A person interested in only the current assets, income tax, and stock price can construct a logical group of these attributes and use that group to define views. In this manner the user need not repeatedly wade through display of the data for the entire physical schema when a restricted subset of data is desired for visualization.

Logical schemas may consist of multiple groups. For example, under a Sales logical schema, there may be groups for sales tax, sales revenue, etc. A group may be nested: the top-level groups are those atop the hierarchy of group attributes. Groups may be nested to any arbitrary level and may consist of any number of attributes. For any physical schema, a default logical schema with the same name is created having all attributes of the physical schema—essentially one group consisting of all attributes.

Like physical schema, logical schema may be browsed under the `Logical...` menu option in the `Schema` submenu of the `Tables` menu. A logical schema may also be browsed using the Logical option under Schema.

6.3.1 Name of Physical Schema

Each logical schema is a view definition based on one physical schema. The physical schema name in the logical schema associates the logical schema with the particular physical schema. A default logical schema is created for every physical schema that has the same name and attributes as the physical schema.

Creation or modification to a logical schema proceeds first by selection of an appropriate physical schema, hence the name of the physical schema is implicitly built in to the logical schema definition.

6.3.2 Group Name

Each logical schema may consist of many groups. A group name identifies a particular group within a logical schema. When visualizing a data stream, the user is automatically prompted to choose the proper group for mapping definition. This group is the *top level group*.

The group name must be unique within the logical schema (and, hopefully, comprehensible to the user). Any number of groups may be defined within this group in a nested fashion.

6.3.3 Attribute Name

The attribute name within a group definition includes that field in that group. The attribute names may appear within any number of sub-groups within a logical group. The attribute name must be the same as defined in the physical schema—it is the name that associates a the logical group attribute to the actual physical one.

6.4 Graphical Data

Graphical data is the second intermediate stage in processing data records before display. Raw data is parsed using the schema to create *TData* (Tabular Data). On the TData, the user selects the required display attributes such as shape, color, etc.

The result is a *mapping* used to form another intermediate representation called *GData* (Graphical Data). GData is the set of values used for actual display of records on the screen, a graphical representation of TData. It consists of the following attributes: X, Y, Z, color, size, pattern, orientation, shape, and shape-specific attributes.

When you visualize a data stream, you specify the initial mapping using the Define Visualization window (Figure 7). (Default mappings are automatically created when a new data stream is imported.) This window lists the various attributes in the logical schema of the relation. You can choose attributes to be displayed as views, as well as the window in which the data will be displayed, background color, and so on. The view(s) can be created as a bar chart, image plot or scatter plot.

Once you have visualized a data stream in a view, you can edit the GData mapping with the Edit Mapping window (Figure 8). You can access this window by clicking the Mapping button on the Control Panel, or by selecting Edit Mapping from the GraphData menu. If you select another view while the Edit Mapping window is shown, it will automatically update to reflect the mapping of the view you select.

The mapping can be edited to change one or all of the GData attributes. Mapping elements may be defined as constant or variable attribute values, or a mathematical expression using the attribute values described next.

The Edit Mapping window gives you more complete control over the mapping that the Define Visualization window does, so sometimes you must temporarily specify a mapping to get the data into a view, and then edit the mapping to visualize the data how you really want to.

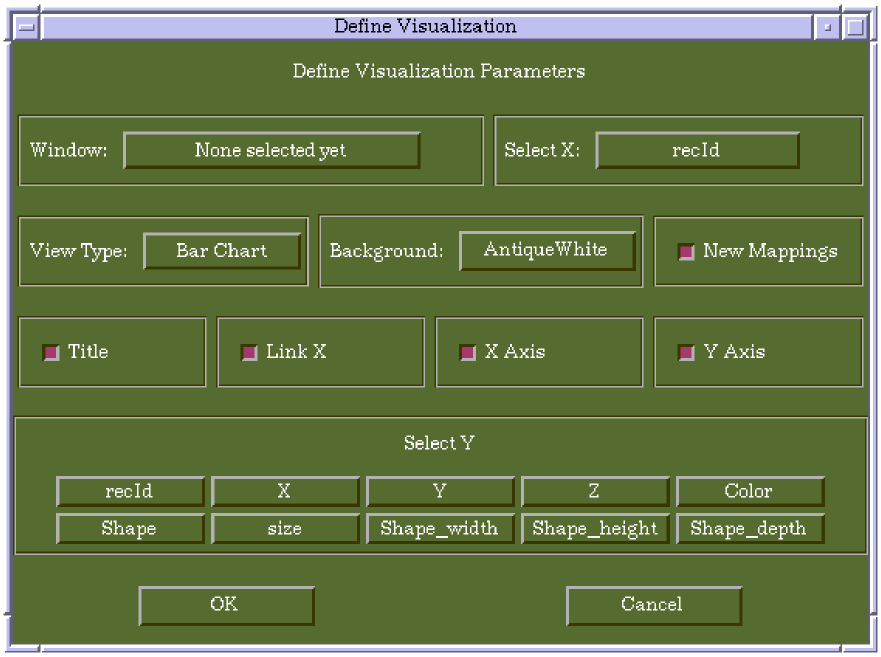

Figure 7: Define Visualization Window

6.4.1 Location: X, Y, and Z

The X, Y and Z parameters specify the location of a record's visual representation with respect to the coordinate axes.

6.4.2 Color

Like the size attribute, color can be a constant, variable, or expression. The use of variable color also yields interesting results. For example, in a scatter plot the color can vary with daily rainfall, using date and temperature on the axes. Currently, a fixed palette is defined such that any expression or variable value computed is mapped to one of 43 colors.

6.4.3 Size

The size of the displayed point/object/glyph in a view is determined by the size attribute. If a constant, the attribute is the enlargement factor of the displayed object. If a variable, the size of the object displayed is dependent of the value of some attribute in the TData.

For example, if a square represents data points and the X and Y axes are date and stock closing prices, respectively, the square size can be proportional to the stock's volume on each date.

Using an expression for the size attribute is even more powerful, as the displayed object size can be a function of several TData attributes.

6.4.4 Pattern

The fill pattern may also be changed for displayed objects. The available patterns are not well-documented at this point, so you will have to try different pattern values. One thing to note is that positive pattern values produce patterns in which the entire shape is filled with the background color before the pattern is drawn (opaque patterns), while a negative value produces the same pattern as the corresponding positive value, except that the pattern is transparent.

Figure 8: Edit Mapping Window

6.4.5 Orientation

Orientation specifies the tilt or rotation of displayed objects, which is useful to display some objects such as diamonds. Like other attributes, orientation can be defined as a constant, variable or as an expression. [This feature is not supported yet.]

6.4.6 Shape

Data points can be plotted as one of the following shapes:

- Rect (rectangle)
- RectX (square–a RectX is square in terms of pixels as long as the relative width and relative height are equal)
- Bar (vertical bar)
- Line (straight, infinite length)
- Line Shade (line, filled to 0 Y value)

- Regular Polygon
- Oval
- Vector (line segment terminated with an arrow)
- HorLine (horizontal line)
- Segment (straight, finite length)
- HighLow (vertical bar, bottom not at 0)
- Polyline (arbitrarily-shaped polygon)
- Image
- Polyline File
- Text Label (string)
- Tcl/Tk Window

The shapes may be defined as constant or variable values (the default is the bar). The square (RectX) gives a visual square shape, block represents a 3 dimensional block, and a polygon represents a dodecagon. The vector is a directional one, with an arrow at the end. The bar is drawn from (X,0) to (X,Y).

6.4.7 Shape Attributes

The shapes have associated with them such as width, height and other values. Shape attributes can be modified in the Edit Mapping window. For an up-to- date list of the attributes available for each shape type, see the Shape Help window (click on Shape Help in the Mapping window). Note that width, height, etc., are generally defined in data units, not pixels.

6.5 Views and Windows

After the creation of a mapping from TData to GData, association of an X attribute to a Y attribute defines a *view*. A view is used to display GData that fall within a specified range of attributes. Views can be manipulated in a variety of ways to achieve visualization of important patterns in the data stream.

Windows provide the screen real estate used by views to draw the GData. A window is responsible for arranging views in its boundaries (the current possible arrangements include tiled/automatic, vertical, and horizontal). Views can be removed from a window to reduce clutter, or be moved to other windows for comparison. Windows can be duplicated.

6.5.1 Manipulating Views in Windows

To arrange views in a window, the automatic mode selects an appropriate layout using the dimensions of the window. For example, a horizontal arrangement is chosen for a long, thin window. Views may also be arranged in a fixed, user-specified way (horizontal, vertical, etc.). The number of views to be laid out either horizontally or vertically may also be defined.

When changing a window, it may be desirable to force DEVise to postpone redraw of complex views until all changes are complete, particularly when working with a large data set. In the Control Panel, toggle the Display button to Layout to block window refreshing. After the changes are complete, toggle the button back to Display to restart continuous window refreshing.

6.5.2 View Title

The view title is independent of the graph displayed and can be modified. The default view title is "<Y attr name> vs. <X attr name>" (e.g., Sales vs. Date). To edit the title of the current view, select the `Title` menu option in the `View` menu. The title may also be deleted, in which case the *current view indicator*, normally a rectangular boundary surrounding the view, becomes a rectangular bar on the left edge. DEVise attaches no significance between the title and data displayed in the view.

6.5.3 View Axes

By default, the X and Y axes of a view's graph are displayed with the minimum and maximum range values shown for each dimension. The visibility of the X axis may be toggled on/off by selecting the `Toggle X axis` menu option in the `View` menu. This toggling may be performed on all views in the current window or the current view only. The same is true of the Y axis.

6.6 Visual Filtering and Display of GData

Graphical data are displayed in many views in many windows. Queries on views take the form of range/region selection using zooming and scrolling mechanisms. For explicit control over view ranges, the Query Tool may be used by selecting the Query button.

A view displays GData that fall within a *visual filter*. Currently there are two types of views: Scatter and SortedX. The Scatter view is used to draw a scattered plot. The SortedX view implements optimizations used to reduce the time used to draw the GData if the X attribute is sorted. The view is basically a graph of the values of the attributes chosen. Views may be manipulated in many ways for effective visualization.

A visual filter defines a query over the graphical data attributes of the GData. Our implementation supports range query over the X and Y GData attributes. The visual filter is used to specify portions of GData to be viewed; the (X,Y) region in a view is the visual filter for that view, since it obscures the rest of the GData.

6.6.1 History

DEVise has a history buffer which maintains a list of previous visual filters for a view. Each history entry records the minimum and maximum X and Y axis values for all views. Using the history buffer, old filters can visited, encouraging exploration with the knowledge that old visualizer states can be restored.

To restore an old visualizer state, select the `History` menu option in the `Help` menu. A history entry can be invoked by clicking on the desired value in the history window. The history mechanism also provides for *marking* of filters, which forces DEVise to retain them in the buffer. If the buffer becomes full, unmarked history entries are discarded.

6.6.2 Links and Cursors

Two features allow establishment of a persistent connection between views: the *visual cursor* and the *visual link*.

Cursors are visual bands that maps and displays the (X,Y) boundary of one view within another, like a zoom window. Cursors provide visualization of a single set of data over multiple, different ranges and scales.

A cursor connects a source view and a destination view. The source view is where the cursor fetches information about the current view (X,Y) axis boundaries, used to display the cursor in the destination view. If the region displayed in the source view changes by scrolling or zooming, the cursor in the destination view moves. Likewise, the cursor can be moved by clicking in the destination view; the cursor will be centered on the click point, and the region displayed in the source view will be updated.

Visual links represent a connection established between two views using one or more of the graphical attributes: (X,Y) axis values, color, size, etc. By linking two views together, changing a linked parameter in one view causes the corresponding changed in the linked view. For example, if two views are linked together by their X axis value, scrolling along the X axis in one view automatically scrolls the other view along the X axis.

Visual links can be named for reference. There can be any number of links at a time between views. However, you're generally best off using the fewest links that you need to achieve a certain relationship between views. Having many links between the same views increases the chance of unexpected behavior when one view changes.

6.6.3 Record Links

Record links are a special kind of link, which operate on the list of records displayed in a view, rather than on the visual filter. A record link has one or more *master* views, and one or more *slave* views.

The definition of a record link is that the slave view or views can display only those records displayed in *all* of the master views. (A slave view may display fewer records, indeed no records at all, depending on its visual filter and mapping.)

To use a record link, first create it as you would any other link, except select "record" as the link parameter. Link views to it as you would to a visual link, except be sure to set one or more views as the master (if you don't set a master view, no records at all will be displayed in any of the views linked to the record link).

Record links can be combined with visual cursors and visual links to allow multi-stage selections to be performed simply by clicking with the mouse.

6.7 Statistics

DEVise provides mean, minimum, maximum, and confidence interval calculation and display as well as Kiviat graphs.

The mean, minimum, and maximum are displayed as lines in the views, calculated over the range of values displayed in the particular view, not over the range of the entire data set.

Confidence intervals are visualized as transparent bands around the mean line. A confidence interval can be any of 85%, 90% or 95%, or none (the default).

Kiviat graphs provide a visual method of analyzing statistical measures over a set of views in a window. A graph can be used to analyze the balance in the values of the various views. Kiviat graphs can display the mean, minimum, maximum, or count (number) of data values. A Kiviat graph changes with changes in the visual filters of the corresponding views.

6.8 Sessions and Templates

A session is a collection of data streams, schemas, mappings, views, and windows. Furthermore, it maintains information including links, cursors, and graphical and statistical preferences for each view. Think of a session as the state of a visual query.

A template is a session without associated data stream(s), consisting of all mappings which can be applied to any data stream conforming to the session's schema. Templates provide a way to view data of different streams under the same mappings and state.

While working in DEVise the current state can be stored as a session or a template. When opening a template, the user must supply a data stream conforming to the schema of that template.

7 Schemas

7.1 User and Superuser Modes

DEVise can be brought up in one of two modes—*user* or *superuser*. The mode used is set by the value of the `UserMode` variable defined in the `.rc` file read by DEVise. A value of 1 indicates user mode, 0 indicates superuser mode. The mode determines the capabilities and presentation of the schema browser. This has nothing to do with UNIX superuser privileges.

To browse schemas, select the `Schema` menu in the `Table` menu, then select the `Browse...` menu option in the former menu.

7.1.1 User Mode

In user mode, the browser allows attribute selection in any predefined schema. Furthermore, views defined on these schemas (through groupings) can be seen and modified. New views can also be defined.

When the schema browser is invoked in the user mode, a list of all schemas defined in the system is presented. By selecting the Attributes button in the browser window, the attributes associated with a schema can be viewed. Furthermore, the set of views defined on these attributes can be seen as a list by selecting the Views button.

New and existing views can be modified within the group browser. The user can either open an existing view for modification or specify a name to create a new view. In either case, a group browser appears. The user can modify/define groups and their contents in terms of base attributes or subgroups. Finally, if a new view is saved, it becomes a part of the list of views on that particular schema.

7.2 Schema Browsing in Superuser Mode

In superuser mode, the browser provides complete control over all schemas. Existing schemas can be opened to display the types and value range of defined attributes, and can be edited by inserting new attributes, modifying range values, etc. Entirely new schemas can also be defined, an operation necessitated by a new data source (and new data layout) to visualized in DEVise.

When the schema browser is invoked in superuser mode, a list of all current system schemas is displayed. One of these schemas can be chosen, or a new one created. In any case, the opened schema is displayed as a list of all possible attributes.

The type and range limits of all attributes are displayed and can be modified. (Range limits are used for automatic display range selection during visualization.)

When a new schema is saved, it becomes an item in the schemas list and a default view is defined for it. The view consists of all schema attributes with no group structure. The default view may be modified and new views created at any time.

8 Data Streams

Data streams are simply sequences of records (or tuples). A data stream can come from almost any source such as the disk file, World Wide Web, tape, etc. Numerous parameters and functions characterize data streams.

8.1 Data Stream Catalog

A data stream catalog contains the name of the data streams defines, their associated physical schema and the caching information.

A data stream catalog can be opened using the `Visualize/Edit` option under the `Tables` menu. The catalog can be browsed and new data streams can be added and existing ones modified.

Data from sources other than UNIXFILEs is read from the source and cached on to the disk. This is indicated by the word Cached in the catalog. A data stream cannot be visualized unless it is cached on to the hard disk.

Note: the data stream catalog is stored in the file `$DEVISE_DAT/sourcedef.tcl`. You shouldn't have to access this file directly, but just in case, you now know where to find it...

8.2 Data Stream Definition

The Define Data Stream dialog box (Figure 9) is used to create or modify a data stream. To create a data stream, select the `New Table` menu option in the `Tables` menu. To edit an existing data stream, select `Visualize/Edit` from the `Tables` menu. Then select the data stream you want to edit, and select `Edit` from the `Streams` menu.

Within the dialog box, select the following: a display name, the type of source (e.g. UNIXFILE), the logical (or physical) schema file associated with that stream, the key (see below), a command (if needed) and cache parameters. These are discussed below.

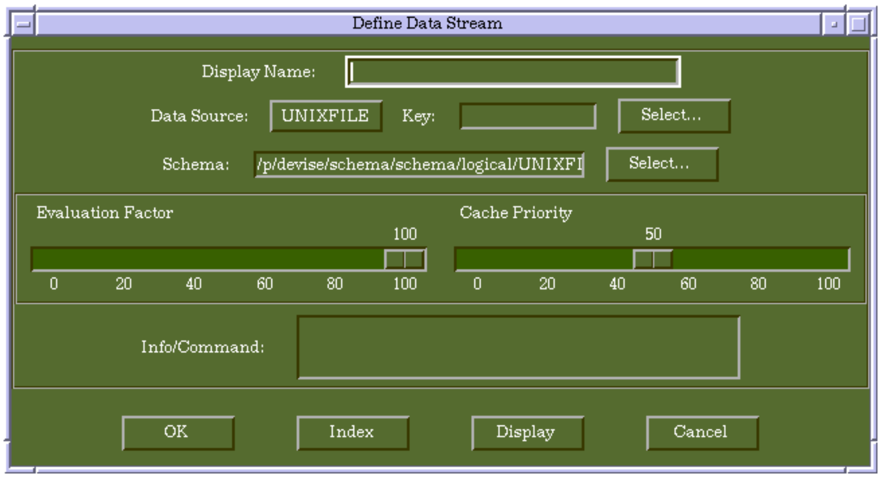

Figure 9: Define Data Stream Window

8.2.1 Display Name

The display name is the name of the stream to be defined. This is the name that will be listed in the catalog as the data stream name. This name must be unique within the catalog. If your data source is a UNIXFILE, the display name will be set to the name of the file when you select the file. Therefore, if you want the name to be something different, you must change the name *after* you select the file.

8.2.2 Source

The source option indicates the source from which data is to be retrieved. There are currently two possible sources namely, UNIXFILE and WWW. The source determines where DEVise has to search for, in order to get the data for caching. The WWW needs a URL to extract the data. The UNIXFILE simply needs the full (relative or absolute) path to a file.

8.2.3 Key

The key is a name that is used to uniquely identify the stream from the rest when storing. Usually the stream definitions are stored as the source name followed by the key name. Thus keys selected should be unique.

1. The keys for the UNIXFILE source must be a valid pathname in UNIX. e.g. /p/devise/dat/UNIX_data1
2. The keys for the WWW source must be a unique and valid name chosen by the user.

8.2.4 Schema

The schema option is used to associate the correct schema for the stream. Remember that DEVise parses and understands the stream with the help of the schema. The schema is not actually used to retrieve the data, but is used when caching it on to the TData form. The schema must be previously defined and stored in a file, whose name is mentioned here. It is important to note that it is by default the logical schema that is stored here. The schema name needs to be a full file pathname. It can also be browsed and selected through the Select option provided.

8.2.5 Evaluation Factor

The evaluation factor specifies the relative time at which the data stream is to be extracted from the specified source and cached on to the local disk. A evaluation factor of 100 would force DEVise to get the data from the source, as soon as the stream is defined and cache it. A low value of the evaluation factor delays this procedure and a value of 0 would postpone this to the time of actual visualization of the stream.

8.2.6 Cache Priority

The cache priority indicates the priority to be given to this stream in the disk cache. When a large number of items are cached, the disk would run out of space. At this point DEVise uses the cache priority specified and removes the least priority item from the disk cache. As a default it is specified as 50%.

8.2.7 Info/Command

For UNIXFILES, the command is the pathname of the file's directory (the name of the file itself is stored as the Key). This should be automatically set when you select a file with the File Select box.

For WWW sources, the command is the URL from which to get the data.

8.3 Manipulating the Catalog

This section deals with manipulating the catalog, part of which has already been discussed above.

8.3.1 Edit

The editing of a stream can be done by using the Stream/Edit option in the Data stream window. The edit is similar to the creation of a new stream and hence is left undiscussed.

8.3.2 Copy

This is used to copy an existing data stream definition on to a different data stream name. The same edit window is displayed, but with the parameters of the data stream to be copied. A unique name is all that is to be given to complete the copy phase (any other parameter can also be changed). This is useful for defining a lot of streams having same characteristic, like schema etc. This option is available under Stream/Copy in the data stream window.

8.3.3 Delete

This is used to delete an existing data stream. The stream is selected first and then deleted. Available under Stream/Delete in the stream window.

9 Visualization

This section gives a detailed account of visualization creation. A tutorial giving step-by-step instructions for using a visualization is below. Before visualization, determine which data streams are available and which attributes you need to visualize. For pedagogical purposes, operation on a demo data stream will be described.

9.1 Visualization Characteristics

Before creating a visualization, open a data stream to be visualized. In the `Tables` menu, select the `Visualize/Edit` menu option. From the list in the resulting dialog box, select the "IBM Stock (MIT Database)" data stream. The "Define Visualization" window will appear. The visualization parameters can be edited from this window.

9.1.1 Window Name

By default, the name of the window in which all views to be defined on this stream will be contained is the name of the stream itself. Once specified, the window name cannot be modified.

9.1.2 X Attribute

The X attribute specifies the schema field to be used as the X axis attribute in views of the visualization. The popup menu lists all attributes available in the schema. When importing a file with many logical groups, the groups need to be selected prior to using the Define visualization window (DEVise automatically prompts for this). Only the attributes present in the top level groups selected are displayed. All views in a visualization have the same X attribute.

9.1.3 Y Attributes

The matrix of buttons at the bottom of the "Define Visualization" window can be toggled on and off to select/deselect Y axis attributes. For each attribute so selected, a view will be created with that attribute as its Y dimension. (For group attributes, buttons are popup menus.)

9.1.4 View Type

You have a choice of viewing your data as a scatter plot, image, or bar chart (the default).

9.1.5 View Background Color

The background color popup menu allows choice of the background color desired for the views of a visualization. A color chart is presented for selection. The background color of a view is retained, even when migrated between windows.

9.1.6 New Mappings

If you have `New Mappings` selected, new mappings will always be generated for the new views. If you do not have `New Mappings` selected, and you already have a mapping of the same attributes of the same TData, the existing mapping will be used for the new view. In general, you should select `New Mappings`. Note that if you have more than one view with the same mapping, changing a mapping parameter, such as color, in one view will change that parameter in all views with the same mapping.

9.1.7 View Title

The view title option indicates whether the title is to be displayed for views or not. The default title is *X attribute Vs Y attribute*. The title can be changed at any time using the `Title` menu option in the `View` menu.

9.1.8 Link X

To link all views along their X axis, select the Link X button. When views are linked in a dimension, scrolling in that dimension changes the visual filter (viewing area) of the views together.

9.1.9 X and Y Axes

The X and Y axis toggle buttons determine whether axis lines and labels are shown or hidden. The X and Y axis display state can be toggled via the View menu after creation of the visualization..

9.2 Visual Filters

Visual filters are regions of X and Y values displayed in a view. A filter extracts visible data from the GData values and displays it on the screen.

9.2.1 Change Visual Filter

Visual filters may be edited in the Query Tool window, accessed using the Query button in the Control Panel. The X and Y ranges of the current view are displayed in the window. Changing the high and low range values changes the region of visible data in the view. To apply a change, select the "Execute" button. To undo the most recent change, select the "Undo Edit" button.

9.2.2 Filter History

DEVise maintains a filter history, a list of X and Y ranges in the reverse order in which they were visited. To view a filter's history, select the "History" button in the Query Tool window.

Places of interest can be distinguished by clicking in the "Mark" column of the history window. Entries are marked with an asterisk. Because a history has finite size, old history entries are replaced whenever the size of the list exceeds its maximum. (Marked entries are preferentially retained.)

The "Back One" option allows you to go back one filter in the history (it toggles the current and previous filters).

9.3 Graphical Data

There are several ways to manipulate mappings from TData to GData. A mapping from TData and GData is partially specified by the options chosen at the time data is imported.

9.3.1 Editing a Mapping

After importing data or opening a session, the mapping can be edited by selecting the "Mapping" button in the Control Panel. The "Edit Mapping" window will appear, in which the following attributes may be edited:

X The attribute mapped to the X axis. A constant value gives the value of the Y axis for a particular X.

Y The attribute mapped on to the Y axis.

Z The attribute mapped on to the Z axis (only valid for 3D views).

Color View foreground color. A constant value paints the graph with the constant value selected, a variable value maps the attribute value to one of the base colors (using modulus *(#colors)* An expression works similarly.

Size Size of the object. The default value is 1.

Pattern The filling style for objects.

Orientation Rotates the objects about their centers. (Giving a proper orientation for a square can make it appear as a diamond.) Note that this feature is not yet implemented.

Shape A shape may be any of Rect, RectX, Oval, Polygon, Bar, Block or Vector. The variable and expression options map the value given to one of these (using modulus *(#shapes)*.

ShapeAttrs Dimensions of the displayed objects, particular to the shape.

9.4 View Properties

Views may be manipulated in a variety of ways.

9.4.1 Swap Views

To swap two views, click them one after the other then select the `Swap Views` menu option in the `Window` menu. The two views must be in the same window.

9.4.2 Move View

To move a view to a between windows, click on the view to make it current, then select the `Move View` menu option in the `Window` menu. In the resulting list of window names, select the desired window. The view may also be moved to a new window by selecting the "New" button beneath the list. In the latter case, another window in created, prompting the user to enter window parameters such as size.

9.4.3 Remove a View

To remove a view, select it then use the `Remove View` menu option in the `Window` menu. The view is removed visually from the window, but not from DEVise's memory.

9.4.4 Bring a View Back

To being back a view to a window, use the `Bring View Back` menu option in the `Window` menu. Select a view from the resulting list of names and a window from the resulting list of window names. Because all views are listed (not just previously removed ones), this option allows copying of views as well.

9.4.5 Destroy View

To permanently destroy a view (remove it from the list of views for the session), select the `Destroy View` option in the `View` menu. A view so destroyed cannot be brought back.

9.4.6 Switch TData

When visualizing a particular data stream, it may be useful to see the data stream in the same format as that of the current session. The `Switch TData` menu option in the `GraphData` menu allows retention of filters and view layouts while changing the data displayed. The switched data stream must conform to the same logical schema as that of the current view.

A view must be selected before switching its TData. To switch, select an open data stream from the resulting table (other data streams may be selected by picking the "Other" button).

9.4.7 Title

The `Title` menu option in the `View` menu allows the view title to be edited or cleared. Clearing the title removes the title from the view completely and changes the current view indicator from a rectangle surrounding the view to a bar to the left of the view.

9.4.8 Axes

The `Toggle X Axis` and `Toggle Y Axis` menu options in the `View` menu show or hide a views axes and high and low axis labels.

9.5 Window Properties

Like views, windows can be manipulated in a variety of ways.

9.5.1 Change View Layout

The layout of views in a window may be changed using the `Layout` menu option in the `View` menu. Three options are presented: automatic, vertical, and horizontal.

Horizontal mode forces DEVise to display the views side by side, vertical mode makes displays the views one on top of each other. The automatic mode adjusts the layout depending on the dimensions of the window. If the window is longer in the horizontal direction than the vertical direction, DEVise uses horizontal layout mode, otherwise vertical layout mode.

The Width and Height parameters are the number of views displayed horizontally or vertically, respectively. For example, two views are oriented vertically when the height is 2. The Height determines the number of views to be displayed atop one another in the horizontal layout mode.

9.5.2 Stack Control

You can use the Stack Control window (accessed by selecting `Stack Control` from the `Window` menu) to create *piled* and *stacked* views.

Both stacks and piles can be thought of as taking all of the views in a window and putting them on top of each other. The difference is that in a pile, the view backgrounds are transparent (as if you made a pile of overhead transparencies) and in a stack, the backgrounds are opaque (like a stack of papers). Therefore, in a pile you can see the data of all of the views, while in a stack you see only the top view. You can use the `Flip` button in the Stack Control window to flip through the views in the stack or pile.

When you pile several views, the X and Y axes of the views are automatically linked, forcing all of the views in the pile to use the same visual filter. When you unpile the views, they are automatically unlinked.

9.5.3 Duplicate Window

Windows can be duplicated, views and all, using the `Duplicate` menu option in the `Window` menu. Duplication is useful for creating cursors and switching TData options. The new window name is the name of the old window with “2”, “3”, “4”, etc. appended as necessary to create a unique name.

9.5.4 Destroy Window

To destroy a window first remove or destroy all views in it, then select the `Destroy` menu option in the `Window` menu. (A window can also be destroyed by using the window manager menu in the window border, although DEVise will complain.)

9.6 Visual Link

Visual links associate two or more views through a common characteristic such as the X axis range, Y axis range, color, etc. Changing the visual filter in one view change the filter in the linked views.

9.6.1 Create Link

To create a link while defining a visualization, select the “Link X” button in the Data Visualization window. Doing so creates a link between all views defined on that data stream.

New links can be created between existing views. Using the `Link` menu option in the `Links` submenu of the `View` menu, open the Select Link window listing all currently available links. (Selecting the “Info” button in the latter window lists all views connection by the chosen link.

To create a new link, select the “New” button then establish the link in the resulting window by picking one of the possible link attributes: X, Y, color, size, pattern, orientation, and shape. A link name must be provided in the field provided to add to the list of existing links.

9.6.2 Unlink

To remove a link, use the `Unlink` menu option in the `Links` submenu of the `View` menu. Select a link name from the resulting list of link names to remove it.

9.7 Visual Cursor

Visual cursors are useful tools for superimposing the visual filter of a source view over a destination view.

To create a cursor, select the `Create` menu option in the `Cursor` submenu in the `View` menu. In the resulting window, enter a name for the cursor and specify whether or not the cursor should be constrained in the X and/or Y dimensions.

To delete a cursor, select the `Delete` menu option in the `Cursor` submenu in the `View` menu.

The source and destination views of a cursor may be changed after creation of the cursor using the `Set Source` and `Set Destination` menu options in the `Cursor` submenu in the `View` menu.

10 Statistics

DEVise provides two statistics facilities for data analysis. First, graphical representation of statistical measures can be added to views, e.g. a line drawn at the across the view at the mean. Second, Kiviat graphs may be generated from data collected across several views.

10.1 Simple Measures

The statistics of displayed data can be visualized by superimposing the mean (average), minimum and maximum on a view of that data. Similarly, confidence intervals can be visualized by coloring a view with transparent bands around the mean line at 85%, 90%, and 95% confidence. All statistical values displayed in a view are calculated for the range of visible X values only, not for the entire data set.

Statistics may be generated and displayed for the current view or all views as desired. To access the statistics options, select the `Statistics` menu option in the `View` menu in the Control Panel. The dialog box which appears contains toggle buttons for showing/hiding the mean, minimum, maximum, and "line" (simple linear fit) in the view(s). A confidence interval around the mean may be displayed by selecting one of the "CI" radio buttons.

The various statistics options may be disabled in the same manner as they are enabled. Each view can have different displayed statistical information by making alterations in the dialog box for successive current views.

10.2 Kiviat Graphs

A Kiviat graph is a powerful method for displaying the collective information of data displayed in several views. For example, a Kiviat graph used in the context of performance data can be used to see how well a system is balanced. The Kiviat graph for the mean, maximum, minimum or count of displayed data can be created.

As the region of displayed data changes (by scrolling or zooming), the Kiviat graph tracks and adjusts. (Informally, a link exists between data views and a Kiviat graph based on them). The actual values along the various axes of the Kiviat graph can be seen by clicking the middle mouse button while in the graph. For successive displays a single Kiviat graph window may be used or new ones generated as needed.

The Kiviat graph options exist as the (tear-off) `Kiviat Graph` submenu of the `View` menu. Either a new graph can be generated or the current one reused (if one exists). The graph is created/updated by selecting one of the four statistical measure then selecting the `Apply...` menu option.

Disable Kiviat graphs by closing the window in which they are displayed (top left corner button).

11 Resources

11.1 DEVise Web Page

The DEVise WWW page at

`http://www.cs.wisc.edu/~devise/`

contains a browseable version of this document. It also contains a number of examples of DEVise visualizations, papers about DEVise, and other information that you may find helpful in understanding and using DEVise.

11.2 Support

If you have questions about DEVise, or wish to report bugs, send email to `devise-sup@cs.wisc.edu`. Please include information on your system architecture and the version of DEVise that you are using.

Copyright

Disclaimer

Agreement

```
-------------------------------------------------------------------------
USER AGREEMENT ----------------------------------------------------------
-------------------------------------------------------------------------

   BY ACCEPTANCE AND USE OF THIS EXPERIMENTAL PROGRAM
   THE USER AGREES TO THE FOLLOWING:

   a.  This program is provided free of charge for the user's personal,
       non-commercial, experimental use.

   b.  All title, ownership and rights to this program and any copies
       remain with the copyright holder, irrespective of the ownership
       of the media on which the program resides.

   c.  The user is permitted to create derivative works to this program.
       However, all copies of the program and its derivative works must
       contain the DEVise COPYRIGHT NOTICE, the RESEARCH SOFTWARE
       DISCLAIMER and this USER AGREEMENT.

   d.  The user understands and agrees that this program and any
       derivative works are to be used solely for experimental purposes
       and are not to be sold or commercially exploited in any manner
       WITHOUT EXPRESS WRITTEN PERMISSION.

   e.  We request that the user supply us with a copy of any changes,
       enhancements, or derivative works which the user may create,
       with the user's permission to redistribute it.
       Copies of such material should be sent to: DEVISE@CS.WISC.EDU
```