ABSTRACT
This paper describes the graphical and text user interface of the SAS® System and how the SAS Display Manager takes advantage of the interface. The topics discussed in this paper include:

- how windowing environments improve programmer productivity
- the evolution of the SAS windowing system
- the base windows in display manager
- the software layering that is used to achieve portability
- the future directions for display manager and the windowing system

INTRODUCTION
The SAS System has four different modes of operation. Each mode of operation serves an important purpose for the user. The first two modes are batch and line mode. Batch mode operation provides an environment for executing non-interactive programs. Typically, these are computationally involved programs with no need for user interaction. Line mode operation provides an environment for executing SAS applications interactively and is provided mainly for compatibility with previous versions of SAS.

The third environment is display manager, which provides a highly interactive windowed user interface in which the user can develop and execute SAS applications and control the flow of the SAS session. The display manager does this by having different subtasks of the SAS System to be placed in independent viewable areas named windows. The display manager environment is used by users who prefer to interact with the SAS System through the SAS programming language.

The fourth environment is SAS/ASSIST®, which provides a front end environment to the display manager that allows novice users to use SAS products without having to know the SAS language. The SAS/ASSIST environment displays screens that prompt the user interactively for information. This information is used to generate and submit SAS program statements. The user then may view the submitted code to learn about the SAS syntax, or edit the code to produce a customized program for his site.

The display manager and SAS/ASSIST environments are completely based on the SAS windowing system. The differences in the environments are related to how the environments interact with the user. However, the underlying windowing system they both use are exactly the same. What a user has learned from starting with the SAS/ASSIST environment can be applied to working in the display manager environment.

HOW WINDOWING ENVIRONMENTS IMPROVE PROGRAMMER PRODUCTIVITY
Windowing systems allow the user to interact with the SAS system efficiently. By using display manager, the user can see how the statements he just submitted actually perform. Depending on the results, the user can change the statements and rerun the procedures without having to reinvoke the SAS System (in essence, play “what if” with the system). The user can also invoke pop-up windows to receive help on any of the statements. This gives the user a quick reference to all aspects of the SAS System. The pop-up windows can also be used in place of SAS statements. Options, titles, footnotes, and more can be displayed and set without submitting any SAS statements. This gives the user more control over his environment. If the user is using SAS/ASSIST and pop-up windows, he no longer needs to know the syntax of SAS statements; instead, a user can submit SAS code via data entry screens. By using the windows supplied by the SAS System, the user can control his environment as he prefers.

The features within a window also can improve a programmer’s or user’s productivity. The new widgets allow the user to enter data without having to know the format of the data. The widgets themselves present the current settings and allow the user to change the settings with a simple press of the ENTER or mouse key. The visual layout of widgets allow applications to design screens that are less complex than their field entry counterparts. This allows the designers of applications to create screens that are easy to comprehend, yet allow for the maximum data presentation for a given screen.
Another feature of a windowing system is that command line syntax is no longer required knowledge because of the PMENU interface. Again the user can simply press the mouse or enter key and choose the action that needs to be performed.

The result is that application developers can create professional screen layouts that users find easy and fun to use.

**CURRENT DISPLAY MANAGER FUNCTIONALITY**

Four main windows make up the display manager environment. They are the PROGRAM EDITOR, LOG, OUTPUT and OUTPUT MANAGER windows. These windows are conditionally displayed each time the SAS SYSTEM is invoked and are used to interface into the SAS programming environment.

The PROGRAM EDITOR window implements a basic editor used for developing SAS applications. Unlike previous versions of the SAS System, a common editor is used by all editing windows within the display manager environment. The editor can be customized to allow the user to work in a more familiar environment; or on some hosts, the user can optionally substitute the host’s native editor. Some of the features of the PROGRAM EDITOR include submit, recall, undo, search and replace, and cut and paste. Once a user has finished editing his code, the code may be submitted to the SAS System for execution.

The LOG window displays the SAS statements as they are being executed, along with any notes, warnings or other informative messages. All LOG window data are saved in a scrollable pad that may be browsed at any time.

The OUTPUT window contains all text-oriented output produced by the SAS procedures that have executed. The output is page oriented and may be scrolled to view all portions of the listing. Of course, the data can be saved to an external file at any time.

The OUTPUT MANAGER window contains a directory of the output in the Output window. The name of the procedure which created the output and other statistics are listed. The user may then choose to edit or delete any of the procedure output. Any changes made to the procedure output are reflected in the OUTPUT window. For example, a user may wish to change a title, remove unneeded procedure output, or simply rearrange the listing. To edit the output of a particular procedure, the user simply chooses the edit option for the procedure in the directory list. An edit session containing the output for the procedure chosen is displayed. The user can change any of the data, and when the edit window is terminated, the changes are reflected in the OUTPUT and OUTPUT MANAGER windows. Procedure output can be deleted by choosing the procedure and the delete option. All output produced by that particular procedure run is removed from the output listing.

**POP-UP WINDOWS**

If a user is running in line mode, display manager, or SAS/ASSIST, the SAS System supplies a series of pop-up windows that can be used to manage data, supply information, or control the SAS environment. These windows work the same in each environment. Some of the more important windows are listed below:

HELP
The HELP window can be used to receive help on any portion of the SAS System. The window displays information on all of the procedures, windows, options, and other aspects of the SAS System.

KEYS
The KEYS window allows a user to change the command settings assigned to any of the function keys.

OPTIONS
The OPTIONS window allows the user to view and change many of the set able options in the SAS System.

TITLES / FOOTNOTES
The TITLES and FOOTNOTES windows allow the user to view and change many of the set able options in the SAS System.

FILENAME
The FILENAME window displays a list of all assigned filerefs in the SAS System.

LIBNAME
The LIBNAME window displays a list of all assigned libnames in the SAS System. The user may select any of the libnames; this in turn invokes the DIR window on the selected libname.

DIR
The DIR window displays a list of catalogs and data sets for the given libname. If the user selects one of the catalog entries, the CATALOG window is invoked for the catalog that was selected. If the user selects a data set,
the VAR window is invoked for the data set that was selected. The user is also allowed to delete or rename any of the data sets or catalogs.

**VAR**

The VAR window displays a list of all the variables in a given data set along with format and type information. The user is allowed to change a variable name or the formats associated with the variables.

**CATALOG**

The CATALOG window displays a list of all the objects in the catalog. The user may delete, copy, or rename the objects.

**GRAPH MANAGER**

The GRAPH MANAGER window allows the user to specify up to four graphic output windows that can display the graphs stored in the user's graphic catalogs. Each window can be associated with a different catalog or the same catalog can be viewed in multiple windows.

**SOFTWARE PORTABILITY**

One of the major accomplishments of Version 6 is that the windowing environment is designed to allow SAS to have the "look and feel" of the native windowing operating environment. This is done by layering the code within the windowing system.

```
Application Code
   Core Code
   Window Manager
   Device Driver
```

The application and core code is portable across all host and device environments. This has been accomplished by designing a functional interface that is independent of the machine or environment. This interface supports full window manager functionality and is documented to perform the same feature set and functionality between machines. The applications then call this portable interface since they can be guaranteed that the interface will not change across hosts. For example some of the basic window manager functions are

- Get a window
- Get fields and widgets in the window
- Get the input the user entered

By designing a rich window manager interface, the application does not need to know specifics about the device. Instead the concepts are portable and the application manipulates the portable concepts. This allows all applications supplied by the SAS System, along with applications you as a user may write, to be portable across all host environments.

It is up to the window manager to supply the functionality documented in the windowing interface. Within the SAS System there are three distinct window manager systems (Xwindows, OS/2® Presentation Manager and Windows, and the SAS window manager). Xwindows is currently supported on the UNIX and VAX hosts, but has run under the CMS operating system. Support includes, but is not limited to, the DecWindows and Motif toolkits. OS/2 Presentation Manager and Windows run on the PC systems. The above window managers are vendor supplied and are very graphical, and the applications that are available on these systems fully utilize the graphical features. To be competitive, SAS applications also incorporate these features, which is why we have designed the SAS System internals to take advantage of each windowing system.

The rest of the systems that currently support Version 6 of the SAS System use the SAS window manager, which is written by SAS Institute and provides a full window manager on systems that do not have any vendor supplied windowing software. All functionality that is standard in other windowing systems is available in the SAS window manager, including window resizing and movement, color changes, and iconifying, and the internal window features of widget layout and menus.

The interface between the core and window manager code is designed to take advantage of each of the different systems. In the Xwindows and Presentation Manager systems, calls into the interface are directed straight to the operating system code. Using the vendor supplied interface adds speed and the host look-and-feel to the system. The SAS window manager on the other hand has to manipulate the screen to produce the desired results. In order to do this, the SAS window manager also contains a layer for device drivers that
customizes the windowing system to each device. For example, VT100 devices use character representations to display widget settings, but the 3270 devices use SAS defined program symbols to display the same widget settings. This allows each device to create the best environment possible by using any and all features that a device may support.

One of the biggest surprises in our windowing design is how well the functionality has worked on nonprogrammable terminals. By using everything these devices have to offer, the SAS system has achieved a new look-and-feel to make these devices look like GUI's on newer higher priced hardware. This allows companies to continue to use their current hardware, and still have all the features of a GUI and also be able to use the new interactive procedures being produced by SAS Institute. Who could have imagined a windowing system with full widget support, background colors, interactive graphic support, rotating graphs, and much more on a 3270 terminal?

Even though the environment is portable, functionality has not been sacrificed. Performance is actually improved by taking advantage of hardware and software components native to the environment. In this way, the SAS user gets the best of both worlds by receiving the look and feel of a particular host while also receiving the full feature set of the display manager.

Another feature about the portable system is that each host allows the user to customize his windowing environment. The user is allowed to redefine any of the function keys that are available on his terminal. He is also allowed to resize and reposition any of the windows and choose any color settings within the windows. The user can choose to run using the PMENU system or the command line and choose to have scroll bars for the window set on or off. Within the editor, the user can remove the numbers pane, and within the LOG and OUTPUT windows, the user can specify the amount of data that can be scrolled. On some hosts the user can also substitute the host's native editor for the program editor. These are just a few of the many items that the user can customize for his SAS session. Once a user has customized his session, the settings can be saved in the profile catalog. Then, whenever the user reinvokes a SAS session, his customization will automatically take effect.

**EVOLUTION OF THE DISPLAY MANAGER AND WINDOWING SYSTEMS**

In Version 5 of the SAS System the user was shown a split screen that contained the program editor and log. When output information was displayed, an output screen was shown. These screen areas were predecessors of windows, and this was the beginning of the SAS Display Manager System.

In Releases 6.01 through 6.03 the display manager evolved into a windowing system that placed each of the PROGRAM EDITOR, LOG, and OUTPUT screens into their own windows. Other pop-up windows were added to enable users to interact with the system that formerly could only be done using program statements. These versions of the SAS System were strictly text based and allowed very little mouse interaction or graphics.

**Release 6.06 Windowing Features**

In Release 6.06 of the SAS System, graphics windows have been incorporated into the windowing environment. If the terminal allows text and graphics to be mixed, then all graphic output that is produced by the SAS System can appear in a window. All of the graphics procedures that currently store their images in a graphics catalog can now also display their images in the GRAPH window. This window is similar to the OUTPUT window in that all of the output produced by the graphics procedures is saved and can be viewed by scrolling the images. By incorporating graphics into the windowing environment, the user is capable of controlling his graphics output. He now has the use of function keys, window sizing and positioning, pmenus, scroll bars, and other functions that are available to all windows. Another major benefit of placing the graphics in a window is that the user may choose to size the window so that the graphics image can be viewed while viewing other windows. The other windows could contain text representations of the data or an alternate graphical view of the data.

The system was also improved by implementing common features of standard graphical user interfaces (GUI's). NOTE: these features have been added to many of the applications supplied by SAS Institute, but you as a SAS/AF® developer can also add these features to your applications. Some of the Release 6.06 improvements are detailed below:

**Menus**

Most SAS users are familiar with the standard command line interface. This interface is flexible and allows the user to enter the commands, but there are limitations with this interface. First, the command must be known, and secondly, the command must be typed in and spelled correctly. An optional command interface has been added which is known as
the PMENU or pull down menu interface. On some systems these are referred to as action bars. The PMENU interface replaces the standard command line on the SAS windows with a menu of command groupings. The user may select any one of the groups which in turn drops down a submenu. The submenu can then be scanned and the appropriate action selected. At this point one of several actions may occur: the action could be processed without further user interaction, another submenu may be displayed, or a dialog box may be displayed that prompts the user for more information. The end result of this process is that a command is pushed and processed. The dialog boxes that a user might see are actually small windows that appear and ask him for specific information. The information is requested via input fields, radio buttons, and check boxes.

Cut and Paste
Another major feature that is in Release 6.06 of the SAS System is the ability to copy text from one window to another. Within the SAS System, the user is allowed to copy the contents of any text window and store it in a paste buffer. The buffer contents can then be pasted into any window that is using the SAS editor. The user simply marks the area to copy or cut and executes the appropriate command.

The SAS editor also supports other functions within the marked area. The marked text can be cut or removed, or commands like FIND and CHANGE can be used to limit the area affected by the commands.

Widgets
Widgets are similar to fields except that the user does not need to type to change the information. Instead, the user can move the cursor to the widget, press the mouse button or ENTER, and the state of the widget changes. Each widget has different properties as described below:

Check Boxes
A check box is a single toggle switch that can be in one of two states (on/off). Check boxes are usually grouped together to allow the user to select multiple non-exclusive options. The state of the check box is visually represented, but in a host or windowing system specific manner.

Radio Buttons
Radio buttons are logically grouped in a radio box, and only one button within the radio box can be set at a time. When the user switches to a new setting, the old setting is automatically turned off. This is similar to the buttons on a car radio.

Example:

\begin{itemize}
  \item Car Options:
    \begin{itemize}
      \item Cruise Control
      \item 4 Wheel Drive
      \item Power Windows
      \item AM/FM stereo
      \item Luggage rack
    \end{itemize}
  \end{itemize}

Each option could be included in the list or excluded.

Push Buttons
A push button is an immediate action item. When the button is pressed, the user will usually see the action associated with the button. The push buttons within the SAS System typically allow the user to direct the flow of execution within windows.

Example: Buttons at the bottom of many screens

```
[Run] [End] [Cancel]
```

Pressing the run or end button would cause the system to run or process the information requested. The cancel button informs the system that the request should be canceled.

Scroll Bars
A scroll bar shows the user the relative position of the visible data to all the data.
that can be viewed in the window. Horizontal scroll bars allow the user to scroll the data left and right, while vertical scroll bars allow the user to scroll the data up and down. The scroll bar is also an input tool that allows the user to decide what portion of the data he would like to view without entering commands in the command area. To view other portions of the data, simply move the cursor to the relative position in the scroll bar and click the mouse or press ENTER. The scroll bars also allow a user to scroll one line or one page at a time. Of course, scroll bar usage is optional. Note: scroll bars in Release 6.06 are attached to the window and scroll the data within the window.

Example: A horizontal scroll bar. Note the shaded area in the middle is the thumb and represents the amount and portion of the data that are currently visible in the window.

Icons
The OS/2 version of Release 6.06 contains pushbutton icons. In the SAS/ASSIST product, the block fields have been replaced by graphical icons that represent the selection. The user simply clicks anywhere in the icon to have the action preformed.

Example: Instead of text describing the button, you may see icons like these.

Release 6.07 Windowing Features
Release 6.07 of the SAS System goes a step further in supporting a full GUI interface. Interactive graphics features have been added to the system along with new widgets. Several new products have been added that use the new interactive graphics support such as the Graphics Editor, SAS/INSIGHT®, and Ishikawa. The Graphics Editor allows the user to manipulate the graphs produced by the SAS System via the mouse and function keys. Standard GUI manipulations like dragging and resizing are supported along with graphics editing primitives.

In Release 6.07 icons are supported on 3270 terminals along with background colors and square push buttons. Listed below are some of the new GUI features of Release 6.07 that are available on most hosts.

Listboxes
A listbox is a rectangular area in the window that contains a list of items that can be selected. The listbox is shown with a border around the items and may include scroll bars if the number of items in the list is greater than the area of the listbox. Depending on the application, the listbox may allow single or multiple selections, and the user selects the item by pressing ENTER or clicking on the item. The item can be deselected by reclicking on the item. Applications tend to use the listbox widget when the user has a fixed choice of selections, but the set of choices is large. For example, applications can use the list box to display all the SAS data sets in a directory.

Example: This listbox contains a list of colors along with a vertical scroll bar that can be used to scan through the list. The item Purple has been selected in the list.

Internal Scroll Bars
In Release 6.06, the windowing system allowed scroll bars, but the scroll bars had to be attached to the side of the window. These scroll bars only affected the data within the window. In Release 6.07 the application is allowed to put scroll bars internal to the window and have these scroll bars manipulate application data within a subregion of the window.

Pop-up Menus
A pop-up menu is similar to a pull down menu, yet the menu can be displayed in the internal portion of the window. A pop-up menu is usually triggered to a hot spot in the window, and there may be several hot spots each with different menus. When the user clicks the
mouse pop-up key or presses the pop-up function key in a hot spot area, a menu will appear with a set of selections. The user can then perform actions by selecting an item within the menu. Once an item has been selected the pop-up menu is removed.

Object Linking and Embedding

Release 6.07 of the SAS System under Windows will contain Object Linking and Embedding (OLE). OLE allows one application to give another application real estate within its windows. The secondary application is responsible for displaying the data, and when the user clicks on the data, the secondary application is invoked. This is very useful in the SAS/EIS® product.

The SAS windowing environment is continually undergoing changes. Within the industry as a whole, the user interface area has been the area that has seen continued growth and change. SAS Institute is working hard to incorporate the latest technology into its products. By incorporating the new user interfaces, we are making the products easier to use, without changing significantly the underlying functionality. The next version of the SAS System is currently under development and other releases are being discussed. Currently the next major release is targeting several areas: new development, performance, consistency, and new GUI functionality.

FUTURE DIRECTIONS

One of the items many users have requested is an interactive method of creating popmenus. SAS Institute is currently working on an implementation to allow users to create and modify popmenus interactively. With this tool users will now be able to customize SAS supplied popmenus. Development is also proceeding in several product areas to create highly interactive graphical applications along with upgrading many existing applications to take advantage of the new GUI interfaces.

Performance is a major goal in the next major release especially in the graphics area. This area is being completely redesigned in order to fully utilize the underlying windowing system and device features. This will not only make the product faster, but in many instances require less memory usage. Another change is that graphs produced by the graphics procedures will no longer have to be written to the graphics catalog before being displayed. Instead, the user will be allowed to specify from the graph window which graphs to save in an independent format. Using this option will save time along with disk space.

Consistency between applications is becoming a major concern to our users. Developers from each major product within the Institute have been meeting regularly to discuss and eliminate differences between the products. Many positive changes are being discussed and users will start seeing many of these changes in the next major release. These changes will make the system easier to use especially for sites that license multiple products.

There are also new GUI concepts being added including the following

Combo Boxes

A combo box is a one line edit field that has an associated combo button beside the field. The user can enter data into the field in one of two ways. The user can simply type the data into the combo field, or the user can click on the combo button. When the combo button is pressed, a drop down list appears that contains values for the field. The user can then select one of the values from this list, which will in turn be placed into the combo field. The list may contain many items and can contain scroll bars. Applications will use this type of widget when there are many valid field values, but a common set of values are used a majority of the time. This field can also be used in place of a listbox to save screen real estate.

Example: What the field would look like initially.

Colors: _____ \(\checkmark\)

After the user pressed on the \(\checkmark\), the selections would appear. The selection box can have a scroll bar.

Colors: ___________ \(\checkmark\)

The user selects Green and the selection box is removed and Green is stored in the field.

Colors: Green ___________ \(\checkmark\)

Spinners
A spinner is a one line edit field that has two buttons beside the field and a fixed list of values for the field. The two buttons beside the field sequence through the list of values in the list. One button moves to the next item in the list and the other button moves to the previous item in the list. As the user moves through the list, the current item is displayed in the field. To change the value in the field, the user can simply type the data into the spinner field, or use the buttons to sequence through the items. Spinners can be used in place of a listbox to save screen real estate, but unlike the list and combo boxes, the user can only see one selection choice at a time. Spinners should only be used when the list of valid responses is small.

Example: using the same color list as the combo box

The spinner would have an initial value.
Colors: Red ▼△
If the user pressed the move to next selection(V), the next item Blue would appear.
Colors: Blue ▼△
If the initial value was Red and the user selected move to previous selection(△), White would appear.
Colors: White ▼△

Sliders
A slider is similar to a scroll bar in that it is an input and output widget, but unlike a scroll bar the slider does not scroll data. Instead of scrolling, the slider shows the setting of a value. For example the controls on a stereo system are best described as sliders. A slider usually has a value field associated with it, and as the slider moves the current value shown by the slider is displayed in the field. In many cases, this value is a percentage 0 to 100.

Example: A slider with a 50% setting may look similar to this:

![Slider](image)

Dialog Boxes
In the Releases 6.06 and 6.07 there were several small prompt windows used in the system; these were called requestor windows. The functionality of these windows and the amount of data in the windows was limited. In the next major release a new window type is being created: dialog boxes. Dialog boxes are similar to windows, but dialogs remain on the screen until the user enters the data requested. While the dialog box is displayed, the user can move but not resize the box. Also, the user will not be allowed to activate another window until all requested data have been entered. Dialog boxes are similar to windows in that they can contain any combination of widgets in order to prompt the user for the needed information. Dialogs should be used when information is needed from the user before processing can continue.

Image Support
Image support is being incorporated into the next major version of the SAS System by allowing users to incorporate images they scanned or images from canned libraries into their SAS applications. Several of the standard image formats will be supported, such as TIFF and PostScript. This will allow users to display pictures alongside the text describing the picture. For example, an application could be created that contains information on the housing market in a city. The user would be able to simultaneously see a picture of the house along with the data about the house on the screen.

Font Support
Software font support supplied by Bitstream Inc. is being incorporated into the base product. The Bitstream software fonts are faster than standard SAS/GRAPH® software fonts and produce better quality letters.

Cut and Paste
Cut and Paste is being expanded in order to cut and paste more than simply the text representation of the data. In the next major release the user will be able to move graphics and text objects along with underlying data that are associated with the objects.

The Institute is also discussing other items which may fall into the next major release. Some of the items include adding 3D graphic functionality to SAS products, multimedia support, further enhancements to the GUI, along with more performance and consistency issues.
CONCLUSION

The user interface for the SAS Display Manager System has evolved since Version 5 and Release 6.03. Many new features have been added to make the system easier to use for both experienced and novice users, yet we feel that there are still many more features that can be incorporated. These new features will be added in future releases, and we will continue to remain dedicated to enhancing the interface without sacrificing the flexibility that you are accustomed to in the SAS System.

SAS, SAS/AF, SAS/ASSIST, SAS/GRAPH, SAS/EIS and SAS/INSIGHT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. OS/2 is a registered trademark or trademark of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.