How to Implement a Graphical User Interface Using the SAS® System, Release 6.06 under MVS (without Confusing Your Present Users)

James Hefner, Gulf States Utilities Company, Beaumont, TX

ABSTRACT

IBM®s SAA strategy provided a common graphical interface for all programs on all platforms. Known as CUA, this is the interface that is implemented by SAS® software under MVS/TSO, OS/2® and other platforms, starting with version 6.06.

However, the full CUA implementation under MVS/TSO assumes that the user has a graphics terminal with a mouse. Many of the mainframe terminals in use today do not have a mouse, or even graphics capability. In accommodating users who have such terminals, a programmer using SAS/AF® has to design each SAS application so that the interface does not confuse their non-mouse users, while still being useful for mouse users.

This paper outlines the steps in creating a sample application. The main menu screen and help screens are discussed and built so that they are context-sensitive. A program dialog box is constructed, showing how to design the entry fields so that they guide both mouse and non-mouse users in the entry of data.

APPLICATION CONSTRUCTION

Let's write a SAS application that allows the user to edit the data in a crime database. (For the purpose of this example, the data set SASUSER.CRIME, that is created by SAS when you first call up SAS/ASSIST®, will be used.) The creation of reports and graphs based on this data are needed as part of the application. Context-sensitive help will also be provided. At first, only a few users will be using a mouse, so the transition from ISPF/DMO-style screens to windows will be somewhat transparent. Later on, when all users are using a mouse, the application will be fully mouse-driven.

Main Menu

The first task is to determine what the main menu will look like. A menu could be designed using a PROGRAM entry, similar to SAS/ASSIST, as shown in Figure 1. However, a mainframe veteran encounter this screen, and immediately they may become confused or isolated. That is, the menus on the screen can be clicked on to do something or call something up. That is, the menus become hypertext menus. Also, the PMENU can appear at the top of the window. That way for
example, a person can press PF1, or type a 1 on the command line, or click on the word Help on the menu screen to get help.

Next, the main menu will be set up. First, the entries can be typed in. In trying to keep the application CUA compliant, remember that option 1 should always be Help and option 3 should always be End. Option 2 is the option for opening the database, and option 4 the option for sending a results file to a printer. This makes the application consistent with IBM's Interactive Chart Utility (ICU). Next, the frame indicator lines are typed in. These lines designate what to do when an item is selected, as shown in Figure 3. Notice the frame indicator line above the menu that names the menu screen. Note also the frame indicator lines at the bottom, which indicate where to go when each item is picked, either by clicking on it (the select statement) or by typing in the number (the menu statement). Also, when passing control to another CBT entry (such as primary.cbt in this figure), notice that it is preceded with a '>' instead of '><'. The only downside of using a CBT entry for menu screens instead of a MENU entry is that the links to submenus must be set up manually.

In order to be able to use the function keys to select an item, they must be redefined. Figure 4 shows how to define them for a menu screen. Notice how PF1 and PF3 are not numbered, but remain HELP and END. Since item 1 is already designated to be Help, and item 3 to be End or Exit, it is easier to leave them set up as they are. In this illustration, PF13 to PF24 are left at their default setting; if there are more than 12 menu items, they can be renumbered as well.

Finally, to complete the menu screen, the general attributes for the CBT entry must be defined. Figure 5 shows the general attributes screen for the example menu screen. COMMAND was selected to appear as the banner at the top of the screen. Since no command

![Image](image1.png)

**Figure 2**

![Image](image2.png)

**Figure 4**

![Image](image3.png)

**Figure 5**

![Image](image4.png)

**Figure 6**
Figure 6 shows the main, or index screen for help system. Notice how the main frame or screen is named Index, using the frame indicator line at the top of the screen. By naming the main screen Index, and placing the following line at bottom of each frame, a push button labeled Index can be set up that can always be used to return to the index screen:

```
#0 >> primary.cbt frame=Index menu=index.
```

Notice how the regular text appears as one color, and the text that can be selected (hypertext) appears as another color and underlined. This helps to make the hypertext stand out. The resulting help screens look and feel very much like a Microsoft Windows help screen (see the illustration in Appendix A for a sample Windows help screen).

Always set up a screen describing what the application is used for, and a screen for each option that is available. If the help screens for each option are placed in a separate file, the application can be set up so that if the user asks for help within an option, he or she will instantly receive help for that particular option. This is known as context-sensitive help. By having an index button appear at the bottom of each help screen, the user can click on it to call up the Index. Also, if the current option interacts with another option, provide a field that allows the user to access help information on the interacting option. Figure 7 shows an example. Since the Listing option of this program generates an output file, the user may want to send the file to a printer. Therefore, a field is provided for the user to click on to get more information on sending the output file to a printer. This figure also illustrates adding the above frame indicator line to allow the user to call up the help index. Also notice that the screen is stored under a separate file named LISTING.CBT. By making LISTING.CBT the help screen for LISTING.PROGRAM, the user can get context-sensitive help from within the program by pressing PF1.

Once the help screens have been typed in, their general attributes can be defined. Figure 8 shows the general attributes for all of the help screens. If your help screens are stored in separate files, then set the general attributes the same for all of them. Appendix B lists the source code used to create the Command menu attribute WNDHELP.PMENU for all of the help screens. The main screen for SAS software version 6.06 is listed as the help for the help screens themselves. That way, if a user wants help on the SAS System itself, it is available by pressing PF1 from within the application help. The only problem seems to be that in order to implement this, the SAS System CBT help files must be found, usually under the libname SASHELP, and copied to the program catalog. Note that the Window Type is Help, and that a vertical scroll bar is added for scrolling through the help screens.

### Program Dialog Boxes

At this point, the menu screens and help screens are complete. The next task at hand is to develop the PROGRAM entries to execute the procedures. Very often, the procedures to be incorporated are already written. It may already have a SAS/AF interface, or may have a TSO interface which in turn submits the SAS code for processing. The goal in incorporating an option into the program is to provide an interface that makes it intuitive and easy for your users to make choices and execute that option. Let’s write the interface for the graphing option of this sample application.

Figure 9 shows the Graph module as it originally appeared. This screen may have originally been created in version 5 of SAS software, or was an ISPF/DMS panel. The objective is to take this screen and make it mouse-able without making it confusing for users on non-graphics terminals. The first step is to set up choice groups for your types of crimes and graph destinations. To do this, replace the text for each choice with a field. Figure 10 shows how the screen looks once this is done. Next, set up the field attributes and the CRIME and GRAPH fields.
for each field. The field attributes for the first field are shown in Figure 11. The field is an action field, and is part of a choice group named CRIME1. To allow users to tab to the field, the Protect attribute INITIAL is assigned. Normally, this field would become either a check box or a radio knob, depending on whether it is a single-station or multiple-station choice group. However, this field has a List attribute of 'X'. This causes the field to initially be a blank, allowing non-graphics users to tab to it and place an 'X' in it, as they would with version 5 of the SAS System. By also turning AUTOSSIP off, the cursor will not automatically jump to the next field and cause it to be selected, should the user then press Enter. Figure 12 shows the field attributes for the next field. By naming the Choice Group for this field CRIME1, we have made this field and the previous field a linked action field. If a user places the cursor on either field and presses the mouse button or Enter, an 'X' is placed in the fields and that item is selected. Assign the next pair of fields the Choice Group name of CRIME2, the next pair the Choice Group name of CRIME3, and so on, making each pair a single-station choice group. Also, set up the print destination fields to be a multiple-station choice group named Dest, allowing the user to select either one but not both destinations.

In setting up fields where the user enters either text or numbers, you should set up a HELP entry listing the entries for that field. Figure 13 shows how you enter the name of the HELP entry in the List attribute field so that users can access the list by typing a '?' in the blank and pressing Enter. (If the selection list is small and self-explanatory, list them in the List attribute field instead.) Figure 14 shows the HELP entry for the state field shown in Figure 13.

In the General Attributes window for each program entry, you should type in a meaningful Window Name. A help screen that is part of the HELP system should be entered in the Help: field. A PMENU named HEENDCAL is entered for the Command menu: field. (The source code for producing this PMENU is given in Appendix B.) And finally, the Window Type is set for Dialog Box, with no scroll bars. The general attributes screen is shown in Figure 15. These attributes will display with three buttons labeled 'OK', 'Cancel' and 'Help' along the bottom. Experience has shown that non-graphics users will ignore these buttons, and press PF3 to submit and PF1 for help, while users who have mice will take advantage of them.

Enhancements for Mouse Users

Once program entries are completed using the above guidelines, the finished product is a completed application that is user-friendly to both mouse and non-mouse users. Some of the programs that I have written have not advanced much beyond this point, for the majority of the users do not have a mouse. Let's advance further, and begin adding enhancements that make it easier to use a mouse. In setting up a program to be used with a mouse, the goal is to minimize the amount of typing that is required. That way the user does not have to constantly take their hand off the mouse, type something in, and then click on something else. SAS/ASSIST is a perfect example of optimizing this approach... an absolute minimum amount of typing is required to run SAS/ASSIST.

The first step is to go into the PROGRAM entries, and wherever there is a text field that allows the user to pull up a selection list, enter a '?' as the initial attribute under the field attributes. This causes the selection list to be pulled up automatically when the program is called, allowing the user to click on his/her choice. Another route you can take is to assign a default value to the field, and place a push button next to the field. If the user wants to select something other than the default, they can click on the button and pull up a selection list. In this example, create a push button labeled 'Printers...' that the user can click on to pull up a list of valid printers. It will have an Alias of 'Printers', and have a type attribute of PUSHBTN. The choice group for this button will be 'Printers'. It will have a list attribute of '1' and an initial attribute of 'Printers'. Link this field with the printer name field itself, using the following code:

```
INT:
  CONTROL ASSIS;
  CALL VSGETMAC ( ) : 
  DEST = 1;
  PRINTERS = 0;
```
RETURN.

MAIN:
IF PRINTERS=1 THEN DO:
  DEST=2;
  PRINT=LISTC(PRINTERS,HELP);
  PRINTERS=0;
END;
RETURN;

where Printers is the choice group for the Printers... button and Print is the label of the field containing the printer name. Dest is the choice group for sending the graph either to the terminal (dest = 1) or the printer (dest = 2).

Also, remove the 'X' that is typed in as a List attribute for each ACTION field. This allows either a radio knob or check box to be displayed. You can also assign a default selection for each choice group. This reduces the amount of mouse movements required.

Finally, build a PMENU for the menu CBT screens. Appendix C lists the source code that was used to create MAINMENU.PMENU for the CRIME program main menu. This will replace the 'Command >> ' line at the top with an action bar with pull-down menus. As will be seen shortly, in addition to arranging the present menu choices in a Windows-style menu, additional items or features not listed in the menu itself can be added.

SAMPLE APPLICATION

Let us take a look at the CRIME program, with all of these enhancements. Appendix A defines all the underlined terms in this section.

Figure 16 shows the Main Menu for the CRIME sample application, as it appeared originally. In comparing this screen to the rewritten program in Figure 2, notice that the Main Menu is still the same. The menu choices are the same in the new version, and can still be selected by pressing the appropriate function key, or by typing its number on the command line. Instead of a command line, however, there is now an action bar in the upper left hand corner. Clicking on an item in the action bar reveals a pull-down menu from which an item may be selected. The border of the screen is now a window, with the window name appearing on the banner in the upper left corner. SAS software's new interface allows the user to have multiple windows open on the screen at once, with the active window on top. All windows have a meaningful name to them, so that one can always tell what a window is for by reading the banner. In this example, the main menu screen is displayed, so the window is named for the application - 'Crime Database Program Main Menu.'

Select ANY item on ANY screen, either by tabbing to it with the tab keys and pressing Enter, or by clicking on it with a mouse using the left mouse button. In this SAS application, clicking on the word 'File' reveals a pull-down menu, as shown in Figure 17. This menu has options for updating, browsing, or printing the data file(s) and print file for this application, plus a choice of exiting the program.

In this SAS application, clicking on Options reveals a menu of action options for this application, as shown in Figure 18. Action options include such things as running calculations and creating graphs and reports. Action items under the sub menus can also be accessed from the main menu.

This SAS application has seven Items under Globals, as shown in Figure 19. Menu Off allows the user to turn the pull
down menus, and instead have a command line, just like the previous version of the SAS System. (Pressing the right mouse button in the SAS application also will turn the menus off.) Clock, Calculator, Calendar and Notepad are all like their counterparts in Microsoft Windows. Libname lets the user view a list of the SAS data sets currently allocated to that session for editing or deleting (provided they are allocated in TSO to allow one to do so). Host Command ... issues a TSO command from Within SAS. The user can call up ISPF, ICU, IOF and other programs. The only thing that cannot be ran is another SAS application. These commands could also be issued from the command line. However, the Graphical User interface allows the user to execute these commands without having to remember the syntax.

Clicking on 'Help' reveals a Help Index, Instructions on using Help, SAS help screens, and 'About Crime' as shown in Figure 20. All help screens are context sensitive. Clicking on 'Index' displays the help index, as shown in Figure 21. From this point, select 'Goback' or 'ExitHelp' to return to the Main Menu, 'More' to read more General Help, or 'HELP' to get more help. Help on the SAS System is available by asking for help from the Program Help screens. From anywhere in the general help screens, select 'Index' to return to this screen, and 'Prev' to return to the previous screen.

This application offers help for each module of the program. Selecting 'Index' from the pull-down menu reveals a list of help screens for each module. As will be seen later, these help screens can also be accessed by asking for help within the module itself.

This SAS application, like all Windows applications, has an item labeled 'About CRIME' where CRIME is the name of the application. Selecting 'About CRIME' in this CRIME example application displays the dialog box shown in Figure 22. 'About' screens always tell what the name of the application is and who wrote or published it, with some sort of logo. On several applications (the Microsoft Word for Windows word processor this paper was written on is one example), the 'About' dialog box also provides other information.

Note also in Figure 22 that Clock was selected from the 'Globals' pull-down menu and is now running in the lower right-hand corner.

Let's say that a graph of the murder and auto theft statistics for Texas, Louisiana, and Oklahoma is to be prepared. 'Create Graph of Crime Rate in State' is selected under Options in the pull-down menu, or F7 is pressed. A series of selection lists appears for each blank, allowing the user to select five states. One of these is shown in Figure 23. Texas, Louisiana, and Oklahoma are selected for the first three fields, then OK or Cancel is pressed for the next two fields to leave them blank. Next, the dialog window for the Graph module is displayed. Murder and Auto are selected for the graph. The button labeled 'Printers...' is pressed to pull up a selection list of valid printers, and plotter RPLOT1 is selected. When the Printers... button is pressed, the destination for the graph is automatically changed from the default of Terminal to Printer. The screen then appears as shown in Figure 24. The cursor is now on 'OK,' and the user can exit this screen and produce a report by clicking on the left mouse button, or by pressing Enter. If the user changes his/her mind about producing a report, he/she can select 'CANCEL' and will be taken back to the main menu without producing a report.
Pressing ‘HELP’ from within the graph module dialog window calls up help on the SAS System. These help screens are also available under ‘Help’ in the main menu Action Bar.

Appendix A uses screens from Microsoft Windows to illustrate the terms used in Graphical User Interfaces. However, both SAS and Microsoft Windows are CUA compliant, so the interfaces and the interaction with them are similar. CUA, or Common User Access, is the user-interface component of the SAA software standard endorsed by IBM. PC Tools, the DOS shell under versions 4.0 and 5.0, Presentation Manager under OS/2, X-Window under UNIX®, and CompuServe®’s Information Manager are other examples of CUA-compliant applications and operating systems. Once the user knows how to run one of these applications or operating systems, he/she will feel comfortable running the rest. There are some slight differences in running SAS on the mainframe, and Windows on the PC. These differences are due to the differences in the way mainframe terminals and personal computers handle input. (Mainframe terminals will not refresh the screen, or acknowledge input, until the user presses Enter or a function key.)

OTHER GRAPHICAL USER INTERFACES

The following are additional examples of Graphical User Interfaces. If the system supports SAS software, then a SAS screen is shown. They are all printed at the same scale to give some idea of the resolution of each display. Notice how the basic parts, such as icons, scroll bars, push buttons, banners, check boxes, and so on are similar for these interfaces.

Microsoft Windows 3.x (left), OS/2 Version 2.0, Windows NT® will be similar.

Compare the Windows screen at the left with Figure 24. The same window from our SAS Application is shown. Notice that only the interface components are different. This is true for the SAS Display Manager System running on any platform.

All of the interfaces shown on this page are CUA compliant. Notice how the Action Menus, Push Buttons, Icons, and other interface components are very similar for all of these interfaces. The PC Tools and MS-DOS® Shell Interfaces are examples of character-based interfaces that emulate a graphical user interface. Notice how the Help menu is pulled down on the MS-DOS shell interface. Several of the menu items are the same as our Mainframe SAS Application (see Figure 20).
GEOS™ / Commodore 64, Commodore 128, Apple II (left)

High resolution and large amounts of memory and disk space are useful for running a graphical user interface. However, the GEOS interface shown is running on a Commodore with a 320 x 200 pixel display, 64K of RAM and a single 170K floppy disk drive.

Compare a Database Program running on the Macintosh® (above top) to the same database running under Windows (above bottom). Notice that they are similar in appearance.

OSF™ Motif™/Unix workstations, X-Window terminals, etc.

Open Look is another common Unix and X-Window interface. Motif is CUA compliant. Notice that it is similar in layout to the Windows and OS/2 interface.

DECwindows™/VMS™ and ULTRIX™ operating systems

GEM®/IBM PC Compatible, Atari ST

GEM is owned by Digital Research, and is used by Dr. DOS and some early graphical DOS programs, such as Ventura Publisher® (shown).
CONCLUSION

Using the new windowing interface of the SAS System version 6.06, it is possible to create a graphical user interface that is similar to the interface available on most PC platforms. By using the techniques presented in this paper, you can migrate to such an interface without confusing or isolating your non-graphics terminal users. The resulting interface will be compatible with your existing and future PC platforms, reducing employee confusion and retraining.

APPENDIX A

Illustration of Windowing (CUA) Terms

- Action Bar
- Radio Knobs
- Pull Buttons
- Check Box
- Icons
- Program Manager
- Program Manager Help
- CUA Help

Note the banner on top of each window. (In Microsoft Windows the banner and border of the active window are highlighted. So in this example, the window titled 'Printers' is the active window.)

Selecting an item on the action bar, such as File, opens a pull-down menu. Then select an item on the pull-down menu, such as Open.

The window titled 'Printers' is an example of a dialog box. Dialog boxes cannot be resized (and sometimes cannot be moved).

The window titled 'Program Manager' is an example of a standard (or program) window. It can be moved by clicking on the banner and dragging it to a new location. It also can be resized by clicking on the border and dragging it to a new size.

All windows can be shrunk down into Icons. Icons allow the user to see what windows are available without taking up space or confusing the screen with overlapping windows.

The above illustration shows a Help Screen. All CUA help screens are context sensitive, meaning help is immediately available on the item the user is currently working with. Also, additional help is available by clicking on the underlined words or pictures. This is known as hypertext, and Apple's HypercardTM, shown below, was one of the first hypertext applications. This SAS software version 6.06 program and CUA-compliant programs offer an index as shown, where a topic is selected by tabbing to it and pressing enter, or by clicking on it with the mouse (see Figure 21). Note the push buttons for pulling up the index, browsing back and forth through help, or going back to a previous topic. System help is also available. This SAS software version 6.06 program has these buttons, plus an additional button for exiting help.

Note the banners in the upper left hand corner of SAS software version 6.06 windows. SAS software version 6.06 features standard windows, help windows, dialog boxes and icons.

Also shown above are scroll bars. Scroll bars appear along the right hand and bottom sides of the screen, and are used to scroll through a document using a mouse. SAS software help screens and SAS/FSP screens have scroll bars on them.

APPENDIX B

The following source code was used to generate the PMENU WNDHELP.CBT. Change the catalog: option to create the PMENU in the current application catalog.

```plaintext
proc pmenu catalog=sas606.example desc='Pmenu for Help Screens';
menu wndhelp;
  item 'Goback';
  item 'ExitHelp' selection=exithelp;
  item 'Index';
  item 'Prev' selection=prev;
  item 'More' selection=more;
  item 'Help';
  selection exithelp 'qcan';
  selection prev 'backward';
  selection more 'forward';
run;
```

The following source code was used to generate the PMENU HEENDCAL.CBT. Change the catalog: option to create the PMENU in the current application catalog.

```plaintext
proc pmenu catalog=sas606.example desc='Pmenu displaying OK, Cancel & Help';
  menu heendcal;
    item 'OK' selection=OK;
    item 'Cancel';
    item 'Help';
    selection OK 'end';
run;
```

The following source code was used to generate the PMENU OK.CBT. Change the catalog: option to create the PMENU in the current application catalog.

```plaintext
proc pmenu catalog=sas606.example desc='Pmenu displaying OK';
  menu ok;
    item 'OK';
    item 'Cancel';
    item 'Help';
    selection OK 'end';
run;
```
APPENDIX C

The following source code was used to generate the PMENU MAINMENU.CBT. Change the catalog= option to create the PMENU in the current application catalog.

```
proc pmenu catalog=sas606.example;
  menu mainmenu;
  item File menu=filemenu;
  item Option menu=optionnu;
  item 'Global' menu=globemen;
  item 'Help' menu=helpmenu;
  menu filemenu;
  item 'Update' selection=edit;
  item 'Browse' selection=browse;
  item 'Print' selection=print;
  separator;
  item Exit dialog=leave;
  selection edit '2';
  selection browse '8';
  selection print '4';
  dialog leave 'bye';
  text
    text #1 @4' ;
    text #2 @4 'Enter the Host (!'SO) command: ';
    text #3 @4 len=15;
  text #4 'Are you sure you want to exit the Crime Database Program? :
  menu optionnu;
  item 'Generate Listing of Crimes' selection=listing;
  item 'Generate Crime Rate Report' selection=report;
  item 'Generate Graph Showing Crime Rate in City' selection=graph;
  selection listing '5';
  selection report '6';
  selection graph '7';
  menu globemen;
  item 'Menu Off' selection=meouoff;
  item 'ASSIST' selection=assist;
  item 'Calendar' selection=calendar;
  item 'Notepad';
  item 'Host Command' dialog=host;
  selection assist '99';
  dialog host 'to @1';
  text #1 @4' ;
  text #2 @4 'Enter the Host (TSO) command: ';
  text #3 @4 len=15;
  text #4 'NOTE: If you do not specify a command;'
  text #6 @4 'You will go to the host operating system: '
  menu helpmenu;
  item 'Index' selection=general;
  item 'Using Help' selection=helphelp;
  item 'SAS System Help' selection=sashelp;
  separator;
  item 'About CRIME' selection=about;
  menu helpmenu;
  selection general 'help';
  selection helphelp 'help';
  selection sashelp 'sashelp';
  selection about 'about';
  menu mainmenu;
  item 'Browse' selection=browse;
  item 'Update' selection=edit;
  separator;
  selection edit '2';
  selection browse '8';
  selection listing '5';
  item 'Generate Usting of Crimes' selection=listing;
  item 'Generate Crime Rate Report' selection=report;
  item 'Generate Graph Showing Crime Rate in City' selection=graph;
  separator;
  selection listing '5';
  selection print '4';
  separator;
  selection general 'help';
  selection sashelp 'sashelp';
  selection helpbelp 'helpbelp';
  selection about 'about';
  run;
quit;
```

PICTURE CREDITS

Figures 1 through 24 are of the SAS software running under TSO/MVS. Wall Data's Rumba was the terminal emulator used; the screens were copied from Rumba to Microsoft Word for Windows.

The pictures of Microsoft Windows screens (with the exception noted below) and of the MS-DOS Shell were captured from within Windows, and copied to Microsoft Word for Windows.

The remainder were scanned in from the original documentation, and altered to reproduce the original image as close as was possible. The resulting illustrations were then saved in PCX format, and imported to Word for Windows. Some changes were made in the process.


The illustration of the GEM environment is from page 133 of an article by Tami Peterson. "Desktop Publishing (Ventura Publisher)" PC Magazine, December 27, 1988.


The illustration of the GEOS operating system is a composite of two pictures taken from pages 21 and 34 of GEOS 2.0 Users Manual, published by Berkeley Softworks (now GEOWorks), 1988.

The illustration of the Amiga Intuition operating system is based on an illustration on page 128 of an article by Bob Ryan. "Commodore Sets Course for Multimedia" BYTE, May 1990.

The illustration of Apple Hypercard was taken from an advertisement by Sigma Designs on page 17 of MACWORLD, September, 1990.

ACKNOWLEDGMENTS

The author wishes to thank SAS Institute Inc., for their help. Without the software documentation, technical support, and SAS Communications magazine, this paper would not be possible.

SAS, SAS/AF, and SAS/ASSIST are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. IBM and OS/2 are registered trademarks or trademarks of International Business Machines Corporation. Amiga and Commodore-Amiga are registered trademarks or trademarks of Commodore International Limited. © indicates USA registration.

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

My thanks also to the Lord Jesus Christ for all he has done . . .
By a new and living way, which he hath consecrated for us, ...

Hebrews 10:20.