

CONVERTING FROM THE FORTRAN-66 VERSION OF
THE UNIFIED GRAPHICS SYSTEM
TO THE FORTRAN-77 VERSION

ROBERT C. BEACH
COMPUTATION RESEARCH GROUP
STANFORD LINEAR ACCELERATOR CENTER
STANFORD, CALIFORNIA 94063

INTRODUCTION: FORTRAN-77 is a substantial improvement over FORTRAN-66. Any FORTRAN-66 program that manipulated character strings either had to use machine-dependent extensions to FORTRAN-66 or do things in a very inefficient manner; FORTRAN-77 has a CHARACTER data type, and these manipulations can be done in a completely transportable manner. In addition, the IF-THEN-ELSE construction of FORTRAN-77 will finally allow FORTRAN programs to be written which are not an unreadable tangle of GO-TO's. To take advantage of these new features of FORTRAN, and for other reasons, a new version of the Unified Graphics System was prepared.

This document is a quick guide for converting from the old Unified Graphics System (described in CGTM 170) to the new FORTRAN-77 system (described in CGTMs 203 and 204). It will also supply some of the reasons why these changes were made.

THE OPTIONS LIST: The options list, the first argument in almost all Unified Graphics System subroutines, must no longer be terminated by an asterisk. FORTRAN-77 passes the length of a character string along with the string itself, so the asterisk is not necessary, and its presence can cause trouble. A null options list should be passed as a character string consisting of a single blank; the FORTRAN-77 standard does not allow character strings of zero length.

GRAPHIC ELEMENT/SEGMENT GENERATION: There are many changes that were made in this section, but most of them are relatively simple to identify. The first change is that the name "graphic element" has been changed to "graphic segment" to bring this system into conformity with standard naming conventions. Since the "E" in such subroutines as UGEINT stood for "element", the following name changes were made:

UGEINT becomes UGINIT	UGEPTS becomes UGPMRK
UGEPTN becomes UGMARK	UGELNS becomes UGPLIN
UGELIN becomes UGLINE	UGETXT becomes UGTEXT or UGXTXT

Many of the options themselves have been changed. Among these changes are:

VDIM, DIMM, BRIT, VBRT becomes VDIM, DIM, MEDIUM, BRIGHT, VBRIGHT
COL1, COL2, COL3, COL4 becomes RED, GREEN, BLUE, YELLOW, MAGENTA, CYAN
SOLD, DASH, DOTS, DDSH becomes SOLID, DASHED, DOTTED, DOTDASH
WINK, STDY becomes BLINK, STEADY
SPACING=<value>, XSPACING=<value> becomes SIZE=<value>

VSML,SMAL,LARG,VLRG becomes DSIZE=<value>

There are some subtle reasons why the interpretation of options of differing lengths is faster than the case when most of the options are of the same length. This is important because the scanning of the options list uses a surprisingly large amount of time.

The difference between UGTEXT and UGXTXT is that the first subroutine will plot characters using the hardware character generator or the basic character set while the second always uses one of the extended character sets (SIMPLEX or DUPLEX). UGXTXT is essentially a combination of the old subroutines UGCHAR and UGETXT. Because of the way FORTRAN-77 passes character strings, the length of the strings in subroutine UGTEXT and UGXTXT is not needed.

The new subroutines UGMARK and UGPMRK can do more than just plot points; they can also create any of the ten plotting symbols. The positioning of character strings within a picture is also much more flexible in the new system; it is, for example, very easy to center the titles.

GRAPHIC DATA SET CONTROL: Subroutines UGOPEN, UGCLOS, and UGSLCT have not changed. The only difference is in some of the options for UGOPEN. For example, SEQ4013 becomes SEQ4010 and the FULSCR option is never needed. Any use of subroutines UGDINF, UGRINF, or UGODEV will require changes.

DISPLAY DEVICE CONTROL: The principal change here is that UGEPUT has had its name changed to UGWRIT. Some of the seldom used options in subroutine UGWRIT and UGPICT are different than in the old system; however, the CLEAR option in UGPICT has not changed. The BEEP function in subroutine UGCTRL is handled by the new subroutine UGMCTL.

ATTENTION CONTROL: Any use of subroutines UGEATN, UGDATN, UGRATN, UGKPUT, or UGKGET will have to be examined and modified. The principal use of these subroutines was with the IBM 2250, and since that device is no longer available, this should not affect very many existing programs. These extensive changes were made to bring the Unified Graphics System into conformity with a nomenclature and practice that has become almost standard since the Unified Graphics System was first conceived. The new subroutines that replace the old subroutines are UGENAB, UGDSAB, UGEVNT, and UGECTL.

SCALING AND SCISSORING: This section was the source of serious difficulties in the old Unified Graphics System. Subroutine UGSCAL could be used in a device-independent or device-dependent manner. I believe anything could have been done in a device-independent manner but the device-dependent scheme was usually easier. As a result, many programs have the device limits for a specific graphic device built into them. This device-dependency is not allowed in the new system. Instead a much more flexible scheme is available. This new scheme is very similar to schemes in other systems similar to the Unified Graphics System.

When subroutine UGRSCL is used to establish the plotting area at the beginning of a picture, a simple translation is possible. If the code in the old system looked similar to:

```
REAL PLIM(2,2)
DATA PLIM/XLO,YLO,XHI,YHI/
CALL UGRSCL('* ',PLIM)
```

Then, if XLO and YLO are both zero, this simply becomes:

```
CALL UGDSPC('PUT',XHI,YHI,1.0)
```

If one or more of XLO and YLO is not zero, then code with the effect of:

```
REAL WDOWN(2,2),VPRT(2,2)
DATA WDOWN/XLO,YLO,XHI,YHI/
DATA VPRT/0.0,0.0,XHI-XLO,YHI-YLO/
CALL UGDSPC('PUT',XHI-XLO,YHI-YLO,1.0)
CALL UGWDOW('PUT',VPRT,WDOWN)
```

will be needed. The call to UGWDOW must be repeated for each picture because a call to UGPICT with the CLEAR option deletes the current window.

Any use of UGSCAL will have to be examined to determine what is necessary in the new system. A new addition is the ability to shield rectangular areas from line drawing.

THE EXTERNAL CHARACTER SET: Instead of UGCSET, the new subroutine UGFONT may be used to select between the SIMPLEX and DUPLEX fonts. UGCHAR is not needed because its function has been included in UGXTXT. Finally, subroutine UGCTOL does the same thing but its calling sequence has been extensively modified.

The primary and secondary characters have been changed for some graphic symbols. The old system only allowed 256 graphic symbols while the new system has over 300. The addition of these new symbols required that some of the primary and secondary characters be reassigned. One likely source of trouble is the new secondary character for the plotting symbols, but you would be better off using subroutines UGMARK or UGPMRK to plot these symbols.

ERROR PROCESSING: The error processor has not changed, except that the subroutine name is now contained in a character string. However, the error indices have been changed. When writing an error processing subroutine, UGXERR, an important change is that the overflow indicator in the graphic segment generating subroutines was changed from 9 to 11. In the new Unified Graphics System, two digit error indices are used for errors that can be identified by more than one subroutine.

GRAPHIC ALGORITHMS: This section refers to the subroutines UGAXIS, UGXHCH, UG3DMS, UGCNTR, UGDXY, UGPROJ, UGORTH, UG3TO2, and UGCNVF. These subroutines have been changed extensively. Some subroutines, UGAXIS, UGXHCH, UG3DMS, and UGCNTR, used to call Unified Graphics Systems subroutines directly to add data to a graphic segment but now call a user supplied subroutine which may then call the proper Unified Graphics System subroutine. This new scheme is much more flexible but requires that each call be modified and then some new subroutines be written. Note that

the subroutine names, which are passed as arguments to the call, must be declared in an EXTERNAL statement.

NEW FEATURES IN THE FORTRAN-77 VERSION: Besides the additions described above, there are a few other significant additions.

One of these additions is a pseudo-device, PDEVUGS, that may be used to save pictures in a file in a device-independent form. There was a somewhat similar feature, PDSPDEV, in the old system but that picture data was incomplete and it never worked under VM/CMS. Some of the problems with the PDSPDEV device were related to the scaling and scissoring problems. In the new system, a file containing PDEVUGS pictures is complete and these pictures may be sent to any graphic device supported by the Unified Graphics System. To make this easy, two programs PDEVUGSI and PDEVUGSN have been written. PDEVUGSI is an interactive program that allows anyone to select any pictures from any PDEVUGS file and view them on any interactive graphic device supported by the Unified Graphics System. PDEVUGSN is a non-interactive program that allows pictures to be selected from any PDEVUGS file and sent to any non-interactive device supported by the system.

As new graphic devices become available, they will probably only be supported on the new Unified Graphics System. One of the reasons for this is that it is much easier to add new graphic devices to the new system. At present the IMAGEN printer/plotter is supported on the IBM computers only by the new Unified Graphics System. On the VAX-11 computers, the PRINTRONIX Model MVP, is only supported by the new system.

WARNINGS ABOUT POSSIBLE PROBLEMS: On the IBM computers, the old system is obtained by the statement "GIME UGS66" while the new system is obtained by "GIME UGS77". It is preferable that you never have both mini-disks linked to you at once. However, if you must be linked to both at once, it is imperative that the "GLOBAL TXTLIB" statement in effect only lists one of the Unified Graphics System text libraries. If a mixture of old and new Unified Graphics System subroutines are included in your load module, the result will be an unpredictable disaster. This problem is unlikely to occur on the VAX-11 computers.

With the old Unified Graphics System on the IBM computers, you had to be linked to the UGS66 mini-disk when you created a load module and when you executed the load module. In the new system, the UGS77 mini-disk is only needed when you are creating a load module.

When creating a load module using the new Unified Graphics System, you must bring together more modules than were required in the old system. It is absolutely vital that the **NUCLEUS** module be included. In addition, you must include any device-dependent code and extended character sets that you intend to use.