THE BOLOGNA GRAPHICS LIBRARY

REFERENCE INFORMATION FOR THE BGL SUBROUTINES

A.Ficarra and M.Nanni
Istituto di Radioastronomia - CNR

L.Federici
Dipartimento di Astronomia - Bologna University

BGL 3/1985

RAPPORTO VAX 11/780

CONSIGLIO NAZIONALE DELLE RICERCHE

ISTITUTO DI RADIOASTRONOMIA

c/o ISTITUTO DI FISICA « A. RIGHI »
Via Irnerio, 46 - 40126 BOLOGNA (Italy)

INTRODUCTION

1.1 PRESENTATION

This manual describes each of the subroutines in the libraries of the BGL system. These libraries are:

1) Main Graphics Library (MGL)
2) Basic Calcomp Interface (BCI)
3) Terminal Management Library (TML)
4) Miscellaneous Routine Library (MRL)

This manual assumes you know the BGL system, described in two previous documents: the "Presentation" paper (BGL 1/84) and the "BGL User's Guide" (BGL 2/85)(*). We suggest you to read those documents before consulting this manual.

1.2 SUBROUTINE NAMES

The subroutines are listed in alphabetical order by their name, independent of the library to which each subroutine belongs.

With exception of the subroutines of the BCI, that keep their traditional names, all the subroutine names begin with the string "BGL_". This prevents any ambiguity with the names of other user or system routines of your installation.

The remainder of each name is usually either a "mnemonic" word, that attempts to remind what the subroutine does, or, when this is not possible by using only a word, it is an acronym for the sentence describing the function the subroutine performs. The subroutine names of the MGL are all considered as acronyms and the meaning of each acronym is explained below each subroutine name. Moreover, the words in the sentence are usually arranged in such order that the first words describe a general class of functions, whereas the last ones refer to the specific function; in this way, subroutines listed in alphabetical order are automatically listed also by function.

(*) At present (June,1985) the second document is still in press. It is temporarily substituted by a group of seminars given by the authors

1.3 OPTIONAL ARGUMENTS

Several BGL subroutines take "optional arguments"; that is, they assume a value if you do not specify the argument when you call the subroutine. Each of these assumed valued is an "argument default" and coincides with the "current" default defined by the last call to the corresponding default setting routine, or with the initial default if you have not explicitly changed it.

If you want to default an argument when you call a subroutine, you must supply the commas that precede and follow that argument position. However, it is not necessary to supply commas when no specified argument follows them. In particular, if all the arguments are defaulted, neither commas nor parentheses are required. For example, suppose that the subroutine SUB takes three arguments, all optional, named A,B and C; the following list shows some of the possible calling sequences:

`CALL SUB ( A,B,C )`	no argument defaulted
`CALL SUB ( ,B,C )`	argument A defaulted
`CALL SUB ( A,,C )`	argument B defaulted
`CALL SUB ( A )`	arguments B and C defaulted
`CALL SUB`	all arguments defaulted

As the only exception to the above rule, VAX/VMS FORTRAN does not allows you to default string arguments, but there is a way that you can "specify" the default: you can pass a "blank string" (a string containing only a space) for any "optional" argument that is a CHARACTER data type and BGL will assume the default.

1.4 RETURN CODES

Almost always, the last argument of the calling sequence to a subroutine of the MGL is an optional output argument where the subroutine returns information about the status of the operation.

Usually this kind of information concerns some possible user errors that are automatically corrected by BGL in order to avoid conditions that would produce a fatal error on VAX/VMS and cause the program to abort. A typical corrective action consists in resetting a value, that is out of the allowed range, to the closest valid value.

You can handle errors by testing the return code values in your application program, in order to perform the specific error reaction. For example, if you attempts to make a coordinate transformation that, due to some wrong data, would generate too small or too large transformation coefficients, the related routine holds the current coordinate system and notifies you of this event by setting the return code to an appropriate value meaning that the operation was performed unsuccesfully.

Appendix A lists all the possible return codes, their meanings and the error reactions of BGL.

1.5 ORGANIZATION OF THE MANUAL

For each subroutine, information is provided in a predefined format anc is splitted in several sections. Some sections are common to all subroutines, independent of the library to which each subroutine belongs, some other sections are library-dependent. The following list explains what each section describes (if the library name is not specified the section is common to all libraries)

MODULE - This section shows the subroutine name

LIBRARY - This section shows the name of the library to which the subroutine belongs.

MEANING (MGL) - This section shows the sentence which contains the subroutine name as an acronym. The letters of the subroutine name appear in capitals.

DESCRIPTION - This section explains what the subroutines does.

CHARACTERISTICS (MGL) - This sections contains three items:

1) The first item may be :
   - "input" : The subroutine performs a graphic input operation
   - "output" : The subroutine performs a graphic output operation
   - "control" : The subroutine does not perform any graphic operation, but it is called to change the status of the system (for example, to define a new value for a graphic attribute), or to obtain information about the current status (inquire-type function)
2) The second item may be :
   - "interactive" : The subroutine is executable only during an interactive task
   - "interactive and off-line" : The subroutine is always executable (if a physical device is selected)
3) The third item may be :
   - "savable" : The operation can be stored in a "User Graphic File" and executed again later
   - "not savable" : The operation can not be stored in a U.G.F.

FINAL CURRENT POINT LOCATION (MGL and BCI) - This sections explains where the graphic point is located after the subroutine is executed.

EFFECTIVENESS (TML) - This sections tells you on what type of terminal the subroutine is effective. In fact, the Terminal Management Library contains two group of subroutines. The first group consists of routines that allow to operate in "full-screen mode" on video terminals conforming to the ANSI standards (like DEC VT100's), whereas they do not produce any effect if you recall them while you are working at any other kind of terminal. The second group contains routines being always effective and allowing to set or modify some terminal line characteristics. For example, you can use one of these routines if you want to disable the terminal from receiving broadcast messages.

USE - This section shows you the calling statement form. All arguments, both required and optional, are listed in the order that you must place them. Usually, the required arguments precede the optional ones and the input arguments precede the output ones.

ARGUMENTS - This section lists each argument. The first line contains fixed items that inform you whether the argument is REQUIRED or OPTIONAL, whether it is an INPUT argument (user-supplied value) or an OUTPUT one (returned value by the subroutine), and what data type it is (INTEGER, LOGICAL, REAL or CHARACTER). Moreover, if the argument refers to an array structure, the word ARRAY is specified. The remainder of this section contains a brief description of the argument and of its effect on the call. Finally, if the argument is optional, the argument default is given and, if the default itself can be modified, also the initial default is given.

1.6 OVERVIEW OF THE BGL SUBROUTINES

The following table lists all the BGL subroutines. The subroutines are ordered by library and, for each library, are organized by function. The table lists each subroutine name together with a brief description of what the subroutine does.