

LIBJCP

GENERAL DESCRIPTION

LIBJCP is used as a FLEXtm command within a procedure to load and execute another procedure. At the conclusion of the called procedure, control will return to the calling procedure and execution will resume at the line following the LIBJCP command.

The general syntax of the LIBJCP command call is:

```
LIBJCP,<filespec>[+]['<parameter list>']
```

where <filespec> is the procedure file that will be loaded and then executed by JCP. The default extension is TXT, and the default drive is the current working drive. With one exception, the rules for the use of the optional <parameter list> are the same as those for the JCP command call. The difference is in the assignment of a null value. Any position that is not assigned a value will retain the value generated by the calling procedure. To assign a null value to a parameter, use the %n=(CR) statement in the calling procedure prior to using LIBJCP. If the "+" option is entered, the echo of JCP lines to the terminal will be turned off during the execution of the called procedure.

Upon conclusion of the called procedure, control will return to the mainline procedure. Execution will resume at the line following the LIBJCP command, and all parameters will retain the values generated by the called procedure.

Example:

(Mainline procedure)

```
ONERROR BREAK
LIBJCP,FORMAT
COPY,Ø,%1
END
```

The LIBJCP command in this mainline procedure will load and execute the procedure FORMAT.TXT. At the conclusion of the called procedure FORMAT, execution will continue at the line containing the COPY command.

A called procedure can also use the LIBJCP command to call another procedure; however, upon conclusion of the procedure last called, control will return to the mainline procedure. In other words, the procedure that invoked the first LIBJCP command will regain control.

* FLEX is a trademark of Technical Systems Consultants, Inc.

SYSTEM REQUIREMENTS

LIBJCP.CMD is assembled to run with the supplied version of JCP.CMD. Additional memory is required for saving the calling procedure's variable storage area and loading the called procedure. LIBJCP will try to use the remainder of the JCP storage area used for holding the calling procedure; if there is not enough room, then an auxiliary storage area is used. The size of this storage area is 1.5K bytes and it begins at \$6A00.

If JCP has been modified, or if a change in the size or location of the auxiliary storage area is desired, then refer to the next section for information on reassembly of LIBJCP.

MAKING CHANGES TO LIBJCP

Reassembly of LIBJCP will be necessary if any changes have been made to the supplied version of JCP, or if a change in the location or size of the auxiliary storage area is desired. The necessary source code files LIBEQU.TXT and LIBJCP.TXT are supplied and this section will describe the various assembler directives that affect program organization.

The LIB file LIBEQU.TXT contains some of the same EQU directives that are in the file JCPEQU.TXT. These directives are described in APPENDIX D of the JCP User's Manual and must be set to the same value as their corresponding directives in the file JCPEQU.TXT. For example, if the EQU directive PART2 is set to a value of \$9000 in JCPEQU, then the EQU directive PART2 in LIBEQU must also be set to \$9000.

The following two directives are also contained in the file LIBEQU.TXT and can be changed to suit a particular system configuration:

BUFF2 EQU \$6A00

The beginning of the auxiliary storage area is determined by the value this directive sets. If BUFF2 is set below the current value of MEMEND and it is necessary to use the auxiliary storage area, then LIBJCP will adjust MEMEND to point just below BUFF2 and JCP will restore MEMEND upon conclusion of the called procedure.

BFSIZE EQU \$05FF

This directive will determine the size of the auxiliary storage area. BFSIZE is currently set to a value of 1.5K bytes.

ASSEMBLY PROCEDURE

There are two procedures supplied that will aid in reconfiguring JCP and LIBJCP. The procedure GENLIB.TXT will automate the process of editing and assembling a new version of LIBJCP. This procedure will require TSC's editor and assembler, and its output will be the command file LIBJCP1.CMD, which can be renamed and copied to the system diskette.

To use this procedure, format a diskette and copy the files LIBEQU.TXT, LIBJCP.TXT, and GENLIB.TXT. Now assign the work drive and enter the following JCP command call:

```
JCP,GENLIB[+]['P#1'P#2'P#3'P#4'P#5]
```

where parameters 1-5 are as follows:

```
P#1 = PART2  
P#2 = PLIMIT  
P#3 = PSIZE  
P#4 = BUFF2  
P#5 = BFSIZE
```

The following JCP command call will assemble a new version of LIBJCP and produce the command file LIBJCP1.CMD. The auxiliary storage area will be relocated to begin at \$E400 and decreased to a size of 1K bytes:

```
+++JCP,GENLIB'''E400'03FF
```

If you are not able to use GENLIB or have to make changes manually to one of the source files, then the following FLEX command will assemble a new version of LIBJCP as a command:

```
ASMB,LIBJCP,LIBJCP.CMD,LS
```

The file LIBEQU.TXT must retain this name because it is used as a LIB file in the assembly process. Also, the edited files LIBEQU.TXT and LIBJCP.TXT must be located on the working drive.

The second procedure, GENJCP2.TXT, will edit and assemble a new version of both JCP and LIBJCP. This procedure uses the LIBJCP command; therefore, some form of LIBJCP must already be running. GENJCP2 will also require TSC's editor and assembler, and its output will be the command files JCP1.CMD and LIBJCP1.CMD.

To use this procedure, format a diskette and copy the files JCPEQU.TXT, JCP.TXT, EDIT.TXT, LIBEQU.TXT, LIBJCP.TXT, and GENJCP2.TXT. Now assign the work drive and enter the following JCP command:

```
JCP,GENJCP2[+]['P#1'P#2'P#3'P#4'P#5]
```

where parameters 1-5 have the same definition as in the procedure GENLIB. The following example will demonstrate:

+++JCP,GENJCP2'9000''E400'07FF

This JCP command call will assemble new versions of JCP and LIBJCP, and produce the command files JCPl.CMD and LIBJCPl.CMD. PART2 of JCP will be located to begin at \$9000, and the auxiliary storage area will be relocated to begin at \$E400 and increased to a size of 2K bytes.