

## 10. CATALOGUED PROCEDURES

### 10.1 Introduction

A catalogued procedure is a string of control commands held in a special format on disk. Such a procedure may be invoked by simply keying-in the procedure name on the console typewriter or any other device to which file code /E0 is assigned. At the end of the procedure control is automatically handed back to the typewriter.

Predefined control commands may of course be invoked simply by assigning file code /E0 to a library file containing control commands. However, a catalogued procedure can be used more flexibly because the basic control commands in the procedure can be automatically modified according to parameters keyed-in when the procedure is invoked.

### 10.2 Invoking Catalogued Procedures

The general form of a catalogued procedure invocation is:

```
procedure-name [ [formal-parameter = actual-parameter [, formal-parameter =
actual-parameter] ... ]
```

The format of the procedure name is defined in appendix A. It has a maximum length of eight characters. A formal parameter is a character string of one to four characters. An actual parameter is a character string which will be built into a control command before it is obeyed by the CCI. The length and value of an actual parameter depends upon the type of control command to which it applies.

When a catalogued procedure is invoked, the CCI will search for a file named M:PROC in the library of the current userid. If this file does not exist the CCI will locate the standard M:PROC file in the system library. The CCI will then search the M:PROC file for the named catalogued procedure.

Having found the catalogued procedure the CCI will search the procedure for the formal parameters specified in the invocation. It will replace these parameters with the corresponding actual parameters and will create a disk file containing the modified procedure. (File code /EE will be used for this file. The user should avoid using this file code for any other purpose).

This file of control commands is then obeyed by the CCI and control is then handed back to the device from which the procedure was invoked.

### 10.3 The Catalogued Procedure File

Catalogued procedures must be held in a special file called M:PROC. There is an M:PROC file in the system library which contains any standard procedures supplied as part of DOS6800 System Software. An M:PROC file may also exist in each user library to hold any catalogued procedure created by the user.

Each M:PROC file may contain several procedures. Each procedure must start with a procedure name of upto five characters (including the first character which must be \$) and end with the word END followed by :EOF. The following example illustrates these points: \$PROC1

```
    commands of procedure 1
END
$PROC2
    commands of procedure 2
END
:EOF
```

Procedures may be updated or new procedures may be inserted by using the Line Editor (see section 6.11.3 for an example).

#### 10.4 The Use of Parameters

Formal parameters may be used in either the command mnemonics or parameter fields of control commands in catalogued procedures. Neither formal parameters nor comments may be used with the procedure name or END command.

Formal parameters may be used in three formats:

```
@Formal-parameter
@Formal-parameter = character-string
@Formal-parameter =
```

The user may, if it is appropriate, omit the specification of certain parameters in a procedure invocation. In this case the action taken by the CCI on the corresponding formal parameters will depend upon the format used. The rules for each format are described below. However, if the actual parameter is specified in the invocation the action taken by the CCI is the same for each format: the actual parameter simply replaces the formal parameter (including @ and =).

If the format @formal-parameter is used the CCI will simply omit the parameter if an actual parameter is not specified. For example:

```
$PR1
RDS
ASM /S, @NLIST
LKE
END
:EOF
```

If the procedure is invoked by keying-in \$PR1 the ASM control command will become simply ASM /S, and the comma will be ignored as it is not followed by a parameter. If the procedure is invoked by keying-in \$PR1 NLIST=NL the ASM control command will become ASM /S,NL.

If the format @formal-parameter=character-string is used, the character-string will be used as a default value if an actual parameter is not specified in the invocation.

For example:

```
$PR2
LED PROG,@FC=/S
ASM @FC=/S
LKE
END
:EOF
```

If this procedure is invoked by keying-in \$PR2 the LED and ASM control commands will become LED PROG,/S and ASM /S respectively. If the procedure is invoked by keying-in \$PR2 FC=/A the LED and ASM control commands will become LED PROG,/A and ASM /A respectively.

If the format @formal-parameter= is used the complete line will be ignored if an actual parameter is not specified.

For example:

```

$PR3
@COM=
LED @PNAM=
ASM /S
INC @MOD=
LKE
END

```

If this procedure is invoked by keying-in \$PR3 COM=RDS the following sequence of control commands will be obeyed:

```

RDS
ASM /S
LKE

```

If the procedure is invoked by keying-in \$PR3 PNAM=PRGRM,MOD=MAIN the following sequence of control commands will be obeyed:

```

LED PRGRM
ASM /S
INC MAIN
LKE

```

In the last example the Linkage Editor would read input data from the file with file code /EO (standard for console typewriter). It is not possible to include input data in catalogued procedures. Processors or utilities (e.g. LED, SYSGEN) which read input data will always access the device with file code /EO.

Catalogued procedures may be invoked from within other catalogued procedures. However, the procedures will not be "nested". That is, control is not returned to an invoking procedure at the end of an invoked procedure.

If an invoked catalogued procedure cannot be found by the CCI the invocation is ignored and the CCI continues processing the original procedure.

## 10.5 Error Messages

The following error messages may be output by the CCI when processing a catalogued procedure:

```

ERROR IN PROCEDURE DEFINITION
ERROR IN PROCEDURE GENERATION (procedure is not correct)
PROCEDURE NOT CATALOGUED (procedure unknown in M:PROC file)
M:PROC NOT CATALOGUED
I/O ERROR

```

## 10.6 \$PCAS

\$PCAS is a catalogued procedure which is supplied with DOS6800 System Software and is held in the system library. The procedure runs the utilities CPLGEN and JESPER to create a cassette containing a Monitor load module and/or an application load module.

JESPER is used to copy a load module from disk onto cassette in a form suitable for loading into memory during TOSS system start. If the load module is the TOSS Monitor then it must be preceded on the cassette by an IPL. This is written by CPLGEN.

The \$PCAS invoking command has the following format:

\$PCAS followed by one space and then a combination of the following parameters, separated by a comma between each one:

M = mname  
 MU = userid  
 A = aname  
 AU = userid  
 C = cname  
 CU = userid  
 D = dname  
 DU = userid

where mname is the name of the Monitor load module to be copied  
 aname is the name of the application load module to be copied  
 cname and  
 dname are the names of the data files (i.e. configuration  
 data) to be copied, in the sequence C then D.  
 userid is the user identity under which the file to be copied resides;  
 this need not be specified if it is the current user.

The following combinations of parameters are valid (see also below):

A M M+A M+A+C M+A+C+D C C+D

Using \$PCAS the necessary load modules and data can be copied onto one, two or three cassettes. The way in which these files can be distributed among the cassettes and the sequence in which the files must occur is explained below.

For CREDIT applications:

One cassette	Two cassettes	Three cassettes
Monitor	Monitor	Monitor
Application	Application	Application
[Monitor Configuration data]	Monitor Configuration data	CREDIT configuration data
CREDIT configuration data	CREDIT configuration data	[Application data]
or	or	
Monitor	CREDIT configuration data	
Application	[Application data]	
CREDIT configuration data		
[Application data]		

For Assembler applications:

One cassette	Two cassettes
Monitor	Monitor
Application	Application
[Monitor configuration data]	Monitor configuration data
[Application data]	[Application data]

Note that as many runs of \$PCAS are necessary as there are cassettes to be created. In addition, it must be noted that \$PCAS assumes all the files to reside on the same disk volume: it is the users responsibility to ensure that this is the situation prior to running the utility.