

## **#8: 4th Dimension's External Procedures**

Written by: Todd Carper    August 18, 1987  
Modified                      October 14, 1987

---

This note discusses the usage of external procedures with 4D

---

When creating external procedures for 4D it is required that the code be compiled to native 68000 code. Not all compilers will perform this operation. Following is a discussion of tips on creating and installing externals, and directions for using various compilers when writing externals.

### ***TIPS***

Do not use global variables in your External Procedures. Declare all variables to be local.

Some compilers require that you call the Procedure from your main level in order to have that procedure compiled.

When passing variables to externals from within 4D, you can pass a local variable to an external but you cannot return a value from an external thru a local variable.

If you want to install the external directly into your application, instead of thru an EXT file. Hold down the Option key while clicking Open in 4D External Mover. You will then see all existing files. Select the file with the .res extension. If you install your external into the .res file, the external will only be available to that application.

It is possible to use ResEdit to install externals, however, it is recommended that for reliable installation of externals, only use the 4D External Mover.

### ***COMPILERS***

#### **MPW Pascal**

Follow standard procedure for writing, compiling, and installing code.

#### **MPW C**

Follow standard procedure for writing, compiling, and installing code.

Be sure to pass the address of parameters. Passed parameters are pushed on the stack in reverse order to that of Pascal.

## LightSpeed C

The following information was contributed by Gary Killingsworth.

1) All parameters to be passed from 4D to LSC are declared as pointers in main(), and main() is declared as a "pascal" function. Note that you will not be able to run the program from LSC using this format! Example:

```
pascal main(num1, num2)
long    *num1;
short   *num2;
```

2) All functions called from main() may be declared as regular C (i.e., non-Pascal) functions.

3) Use Set Project Type to compile the program as a code resource. Type should be set to 'CODE' and ID to '2'. Unfortunately, there seems to be no way within LSC to set the file attribute to type 'APPL' — which brings us to step 4.

4) Using Fedit, choose "File finder attributes" from the Display menu. Change the type from '????' to 'APPL'. If this step is not included, the 4D External Mover will not display the filename.

5) Per the 4D manual, open your database using the 4D External Mover. Choose transfer, choose SEGMENT 2, offset 0, select your LSC code resource. Parameters should be selected per the manual.

6) It is important that any resources (such as DITL's and DLOG's) to be used by the external function be included alongside the 4DEX, 4BND, and 4DTE resources. In addition to the resources being physically present in the file, the 4BND which bears the same I.D. # as your code resource (typically 15000, if only 1 is present) must bundle them (as described in the manual). An important point to remember is that if you delete a code resource from your database file using the External Mover, other resources may be deleted with it. For this reason, it might be best to use ResEdit to delete the specific 4DEX's and 4BND's which you don't need.

At this point, the external procedure should be executable within 4D. The primary reason for using a code resource is that a regular application resource begins with a lot of unexecutable code. 4D expects to find the "Link A6" assembly language command at the offset which you declare in the External Mover. In the case of LSC, the code resource does not begin directly with this statement, but jumps to it after a few housekeeping instructions.

### **Turbo Pascal**

This section discusses the process required for creating an external procedure for 4th Dimension using Turbo Pascal. This process requires the files 4D DHDR and 4D External Template available on the examples disk.

- 1) Use ResEdit to transfer the resource in the file "4D DHDR" into the Turbo Pascal program. It goes in the DHDR resource of Turbo Pascal.
- 2) Follow the guidelines for structuring your program as presented in the file "4D External Template".
- 3) Compile your program.
- 4) Use 4D External Mover to move you program resource into your 4D application .res file or a Proc.ext file. Install with the following procedure:
  - a) Start 4D External Mover.
  - b) 1) If you want to install into a Proc.ext file then click open.  
2) If you want to install the procedure for use in a single application, hold down the Option key and click Open. Then select the .Res file of your application.
  - c) Now click Transfer...
  - d) Choose segment 1, Offset 0 (zero).
  - e) Hold down the Option key and press Left (or right if your resource file is opened on the right instead of the left). Now select the file you created with Turbo Pascal, (it is a resource file).
  - f) Assign the name you desire for the application, select the parameters in the proper order in which they are passed.