

Documentation for SysErr4D - 4D Error External

written by Robert L Jones (CIS 71251,2566)
all documentation and enclosed externals
copyright © Robert L Jones 1990
portions of externals copyright © Symantec Corporation 1989

If you use this external, send \$25.00 to:

**Robert L Jones, MD
1401 Avocado, Suite 505
Newport Beach, CA 92660**

1. The documentation must accompany the external.
2. The Backup external may be used in a database structure or compiled program if a \$25.00 fee is returned to the author at the above address.
3. The SysErr4D, Copyfile, or DeleteFile externals may be used in a database structure or compiled program if a \$25.00 fee is returned to the author at the above address if any or all of these externals are used.
4. The above fee in item #2 and #3 are *only* expected per unique 4th D program/structure (not per workstation).
5. The Backup, SysErr4D, Copyfile, or DeleteFile externals may not be sold separately or as part of a so-called programmer's assistant/helper package/toolkit.

If you should experience any difficulty or would like to suggest improvements or changes, please contact me at the above address.

SysErr4D (Compiler: THINK Pascal v3.0)

This external returns an error string descriptive of the error integer passed to it. The string descriptions are taken from Bill Steinberg's System Errors Table 2.8 DA. Unfortunately, Macintosh error values are not unique. The OSErr system seems to have been built additively by individuals, and not by group planning. That is, the same integer value for an error is generated by the system for different errors depending on the context (eg, the File Manager and the Start Manager could generate an error value of 1 for completely different problems). Certainly a better design would have used value ranges for specific areas, avoiding overlapping error values.

This situation means that a simple scheme as I've employed where an error integer is passed in and a string describing the error is returned is limited in scope. For example, I've eliminated some system errors in favor of file I/O and networking errors which are probably of more interest to a 4th D programmer. Thus, it is possible for the user of ***SysErr4D*** to have an error generate a string which, while correctly indicating that an error occurred, incorrectly describes the error.

Anyhow, moving the return of any error strings to a separate external accomplishes two things: 1) smaller external size for the **Backup** or other externals; 2) ease of updating the **SysErr4D** external without changing the other externals; and 3) allows the user to choose whether to use the **SysErr4D** system (the user could simply act on any non-zero value returned from an external rather than passing it to **SysErr4D**). **SysErr4D** provides almost 500 error strings.

SysErr4D is called in 4th D by:

```
$myErrStr := SysErr4D (errNumber)
```

The variable 'errNumber' can be an error integer returned by **Backup** or any other function or procedure returning a Macintosh toolbox error. (All errors strings are Mac errors except for 9998 and 9999, which are my internal errors indicating for 9998 that the same drive as the source was attempted for copying when 'sameDrive' equalled '1'. 9999 is returned when the user clicks on a 'Cancel' box in any of the standard Mac input/output dialog windows used for file selection, as returned by **Backup**)

I hope you find this external as useful as I do. --Bob