Apple II
Technical Notes

image

AppleTalk
## #5:	SPCommand Calls and Error $0702

Written by:	Mark Day	July 1989

The system now uses `SPCommand` calls asynchronously. Applications that have AppleShare volumes mounted under System Software 5.0 and also make `SPCommand` calls themselves should now handle the "Too many ASP calls" error, $0702.

AppleShare uses a protocol called AppleTalk Session Protocol (ASP) to maintain a connection (session) with all servers that you are logged on to. All commands and data transfer to the server are sent using ASP.

The implementation of ASP on the Apple IIGS has a limit of one command outstanding (waiting to complete) per session. This means that if one command has been sent, its reply must be received before you can send the next command. Remember, the `SPCommand` call is used to send commands over a session. If you try to issue an `SPCommand` before another (asynchronous) `SPCommand` on the same session has completed, your call will return with a "Too many ASP calls" error, $0702.

Before System Software 5.0 on the Apple IIGS, no system software made asynchronous `SPCommand` calls, and therefore this error would only occur if the developer was making the asynchronous calls. As of System Software 5.0, the AppleShare FST uses asynchronous calls to help prevent the loss of a connection with servers and to assist the Finder in dynamically updating windows when a change is made to a network volume. Therefore, this error may be returned even though the developer is not making asynchronous calls.

The error is easy to handle if you are making synchronous `SPCommand` calls. Simply make the call, and if it completes with error $0702, loop back and make the call again until you can do so without error $0702. This technique forces your program to wait until ASP is free again to make the call.

If you are making asynchronous `SPCommand` calls, and you receive the $0702 error, you might want to install a short (i.e., 1/4 second) timer using the `InstallTimer` call, and make the `SPCommand` call again when the timer completes. Remember, the `InstallTimer` has to be asynchronous, since you are making it from the completion routine of an asynchronous call.

The `SPWrite` call also has a limit of one outstanding call per session. System software does not currently use asynchronous `SPWrite` calls, but looping until ASP returns something other than $0702 would be a good precaution for `SPWrite`, too.

Note: When using the AppleShare FST under GS/OS, there is little reason to make
`SPCommand` calls yourself, since most of the calls you can make are available
through the FST as normal file system calls or as FST-specific calls.

Further Reference

- *AppleShare Programmer's Guide for the Apple IIGS*
- *Inside AppleTalk*
- System Software 5.0 documentation (APDA)