

AFP Implementation Using the AppleTalk Filing Protocol

The **AFPCommand** function passes a command to an AFP server. The first byte of the **AFPCommand** command buffer (the AFP command byte) must contain a valid AFP command code.

Note: Server information should be gotten through an **ASPGetStatus** call. **ASPGetStatus** is equivalent to the **AFPGetSrvrInfo** call. Making an **AFPGetSrvrInfo** call using **AFPCommand** results in an error.

Mapping AFP Commands

Most AFP calls are implemented by XPP through a very simple one-to-one mapping of an AFP call to an ASP call without interpretation or verification of the data.

The .XPP driver maps AFP command codes to ASP commands according to the following conventions:

AFP Command Code	Comment
0x0	Invalid AFP command
0x001--0x0BE (1--190)	Mapped to UserCommand (with the exceptions listed below)
0x0BF (191)	Mapped to UserCommand (Reserved for developers; will never be used by Apple)
0x0C0--0x0FD (192--253)	Mapped to UserWrite
0x0FE (254)	Mapped to UserWrite (will never be used by Apple)
0x0FF (255)	Invalid AFP command

The following AFP calls are exceptions to the above conditions:

AFP Command	Comment
getSrvrInfo (15)	Mapped to ASPGetStatus (You will use the function called ASPGetStatus to make this call)
login (18)	Mapped to appropriate log-in dialog including ASPOpenSession
loginCont (19)	Mapped to appropriate log-in dialog
logout (20)	Mapped to ASPCloseSession
write (33)	Mapped to ASPUserWrite

The following AFP calls can pass or return more data than can fit in quantumSize bytes (eight ATP response packets) and may be broken up by XPP into multiple ASP calls.

AFP Command	Comment
--------------------	----------------

read (27)

Can return up to the number of bytes indicated in reqCount

write (33)

Can pass up to the number of bytes indicated in reqCount