INTERNAL COMMANDS DLL INTERFACE

1. OVERVIEW

DLL function uses that same data segment. If a DLL function was called reentrantly, static variables would be overwritten. Thus, if a DLL requires more data than will fit on the stack, use dynamic allocation.

2. REQUIREMENTS

Each DLL must provide the following 3 functions for use by Windows Shell:

int FAR PASCAL ModuleProc (HWND hwndDisplay, int argc, LPSTR argv[]);
@ ORDINAL 3

hwndDisplay - Window handle of STDIO Display to use for I/O.
argc - Number of command line arguments.
argv - Array of pointers to command line arguments. The first pointer always points to the name of the DLL.

This function is called to let the DLL do the function which it is providing. For example, if this were a DLL providing a file deletion function, the DLL would perform the deletion at this time.

int FAR PASCAL ShowOptions (HWND hwndParent);
@ ORDINAL 4

This function is called to tell the DLL to show it's options box. The DLL should display a window which allows the user to set options in the DLL.

int FAR PASCAL ShowAbout (HWND hwndParent);
@ ORDINAL 5

This function is called to tell the DLL to show it's about box. The DLL should display an about window at this time.

NOTE - It is essential that the DLL export these functions at the specified ordinal value in it's .DEF file. Otherwise, The Windows Shell will not properly access the DLL.

3. USING THE DISPLAY

The header file 'wstdio.h' has been provided for outputing lines and other function to the display. The most common of these is dputs(), which you can use to output a line to the display. See the header file for the description of the rest of the functions.

4. UTILITY FUNCTIONS

The 'wslib.dll' provides several useful functions for parsing command lines, and yielding to other applications. It is extremely important that you use the YieldToOthers() function your code sits in a tight loop for an extended length of time. See the header file 'wslib.h' for a list of useful functions.