Error_Services

Application errors should be logged using the Error Services module. There are a few methodological assumptions built into way errors are managed which are important to understand in order to properly work with Error Services:

- The term 'top' refers to the originating procedure of a call stack and the term 'bottom' refers to the last routine (or the current routine) within a call stack. Within the OpenInsight Debugger this will appear backwards since the originating procedure always appears at the bottom of the list and the current routine appears at the top of the list. We are using this orientation because it is common to refer to the process of calling other procedures as 'drilling down'.
- The reason for defining the orientation of the call stack is because Error_Services allows for multiple error conditions to be appended to an original error. In most cases this will happen when a procedure at the bottom of the stack generates an error condition and then returns to its calling procedure. This higher level procedure can optionally add more information relevant to itself. This continues as the call stack 'bubbles' its way back to the top to where the originating procedure is waiting.
- Native OpenInsight commands that handle errors (e.g., Set_Status, Set_FSError, Set_EventStatus) preserve their error state until explicitly cleared. This can hinder the normal execution of code since subsequent procedures (usually SSPs) will fail if a pre-existing error condition exists. Our philosophy is that error conditions should automatically be cleared before a new procedure is executed to avoid this problem. However, the nature of Basic+ does not make this easy to automate for any given stored procedure. Therefore, if a stored procedure wants to conform to our philosophy then it should include a call into the 'Clear' service request at the top of the program. Alternatively this can be done through a common insert (see SERVICE_SETUP for example.)
- Service modules will use the SERVICE_SETUP insert and therefore automatically clear out any error conditions that were set before.

A	
Add	Adds an error to the stack.
C	
Clear	Clears all error conditions and related information.
D	
DisplayError	Displays the current error message to the end user.
G	
GetMessage	Returns the most current error message.
GetMessages	Returns the stack of error messages.
H	
HasError	Returns True if there is an error condition, False if there is no error condition.
N	
NoError	Returns True if there are no error conditions, False if there is an error condition.
S	
Set	Sets an error to the stack.