

```

// Replace \r with \n because GrepString treats \n only as line separator, not \r.
varList = ReplaceString("\r", varList, "\n")
String regExp = "(?m)^\n" + varName + "="
return GrepString(varList, regExp)
End

```

Examples

```

String currentUser = GetEnvironmentVariable("USER")
String varList = GetEnvironmentVariable("")

```

See Also

SetEnvironmentVariable, **UnsetEnvironmentVariable**

GetErrMsg

GetErrMsg(*errorCode* [, *substitutionOption*])

GetErrMsg returns a string containing an explanation of the error associated with *errorCode*. It is most useful for programmers providing custom error handling in user-defined functions.

The **GetRTErrMsg** provides a simpler way to get a description of an error in a user-defined function.

For an overview of error handling, see **Flow Control for Aborts** on page IV-48.

Details

errorCode is an Igor error code. Usually you obtain the error code during execution of a user-defined function via the **GetRTError** function or from the *V_Flag* variable created by many Igor operations.

If multiple errors occur in a user-defined function, this may result in GetErrMsg returning an incomplete error message. See the MultipleErrors example below. You can achieve more reliable error reporting by calling **GetRTErrMsg** from a **try-catch-endtry** block.

Substitution

For a few error codes, the corresponding error message is designed to be combined with "substituted" information available only immediately after the error occurs. An example is the "parameter out of range" error which produces an error message such as "expected number between x and y". The optional *substitutionOption* parameter gives you control over substitution.

To get the correct error message, you must call GetErrMsg immediately after calling the function or operation that generated the error and you must pass the appropriate value for *substitutionOption*.

Igor maintains two contexts which store the substitution information: one for user-defined functions and one for all other contexts (command line execution, macros, and the Execute operation).

Set *substitutionOption* to one of these values:

<i>substitutionOption</i>	GetErrMsg Action
0	Substitution values are filled in with "_Not Available_". This is the default when <i>substitutionOption</i> is not specified.
1	Substitution values are blank.
2	Substitution is performed based on the assumption that the error was received while executing a macro or a command using Igor's command line. This includes a command executed via the Execute operation even from a user-defined function because such commands are executed as if entered in the command line.
3	Substitution is performed based on the assumption that the error was received while executing a user-defined function.

For most purposes you should pass 3 for *substitutionOption* when the error was generated in a user-defined function other than through the Execute operation and pass 2 otherwise.

Examples

```

// Macro, Execute or command line example
Execute/Q/Z "Duplicate/O nonexistentWave, dup"
Print GetErrMsg(V_Flag,2)
// Prints "expected wave name"

```