Do not use numpnts to test if a wave reference is null as this causes a runtime error. Use **WaveExists**.

numtype

numtype(*num*)

The numtype function returns a number which indicates what kind of value *num* contains.

Details

If *num* is a real number, numtype returns a real number whose value is:

0: If *num* contains a normal number.

1: If *num* contains +/-INF.

2: If *num* contains NaN.

If *num* is a complex number, numtype returns a complex number in which the real part is the number type of the real part of *num* and the imaginary part is the number type of the imaginary part of *num*.

NumVarOrDefault

NumVarOrDefault(*pathStr*, *defVal*)

The NumVarOrDefault function checks to see if the *pathStr* points to a numeric variable. If the numeric variable exists, NumVarOrDefault returns its value. If the numeric variable does not exist, it returns *defVal* instead.

Details

NumVarOrDefault initializes input values of macros so they can remember their state without needing global variables to be defined first. String variables use the corresponding numeric function, **StrVarOrDefault**.

Examples

```
Function DemoNumVarOrDefault()
    Variable nVal = NumVarOrDefault("root:Packages:MyPackage:gNVal",2)
    String sVal = StrVarOrDefault("root:Packages:MyPackage:gSVal","Hello")

    Print nval, sval

    // Store values in package data folder for next time

    // Create package data folder if it does not yet exist
    NewDataFolder/O root:Packages
    NewDataFolder/O root:Packages:MyPackage

    DFREF dfr = root:Packages:MyPackage     // Get reference to package data folder

    // Create or overwrite globals in package data folder
    Variable/G dfr:gNVal = nVal
    String/G dfr:gSVal = sVal

    NVAR gNVal = dfr:gNVal
    gNVal += 1

    SVAR gSVal = dfr:gSVal
    gSVal += "!"
End
```

NVAR

NVAR [**/C**][**/Z**][**/SDFR**=*dfr*] ***localName*** [**= *pathToVar***][**, *localName1* [**= *pathToVar1***]]…

NVAR is a declaration that creates a local reference to a global numeric variable accessed in a user-defined function.

The NVAR declaration is required when you access a global numeric variable in a function. At compile time, the NVAR statement specifies the local name referencing a global numeric variable. At runtime, it makes the connection between the local name and the actual global variable. For this connection to be made, the global numeric variable must exist when the NVAR statement is executed.