

## NVAR\_Exists

When *localName* is the same as the global numeric variable name and you want to reference a global variable in the current data folder, you can omit *pathToVar*.

*pathToVar* can be a full literal path (e.g., root:FolderA:var0), a partial literal path (e.g., :FolderA:var0) or \$ followed by string variable containing a computed path (see [Converting a String into a Reference Using \\$](#) on page IV-62).

You can also use a data folder reference or the /SDFR flag to specify the location of the numeric variable if it is not in the current data folder. See [Data Folder References](#) on page IV-78 and [The /SDFR Flag](#) on page IV-80 for details.

If the global variable may not exist at runtime, use the /Z flag and call **NVAR\_Exists** before accessing the variable. The /Z flag prevents Igor from flagging a missing global variable as an error and dropping into the Igor debugger. For example:

```
NVAR/Z nv=<pathToPossiblyMissingNumericVariable>
if( NVAR_Exists(nv) )
    <do something with nv>
endif
```

Note that to create a global numeric variable, you use the **Variable/G** operation.

### Flags

/C	Variable is complex.
/SDFR=dfr	Specifies the source data folder. See <a href="#">The /SDFR Flag</a> on page IV-80 for details.
/Z	Ignores variable reference checking failures.

### See Also

[NVAR\\_Exists](#) function.

[Accessing Global Variables and Waves](#) on page IV-65.

[Converting a String into a Reference Using \\$](#) on page IV-62.

## NVAR\_Exists

### **NVAR\_Exists (name)**

The NVAR\_Exists function returns one if specified NVAR reference is valid or zero if not. It can be used only in user-defined functions.

For example, in a user function you can test if a global numeric variable exists like this:

```
NVAR /Z var1 = gVar1          // /Z prevents debugger from flagging bad NVAR
if( !NVAR_Exists(var1) )      // No such global numeric variable?
    Variable/G gVar1 = 0      // Create and initialize it
endif
```

### See Also

[WaveExists](#), [SVAR\\_Exists](#), and [Accessing Global Variables and Waves](#) on page IV-65.

## Open

### **Open [flags] refNum [as fileNameStr]**

The Open operation can, depending on the flags passed to it:

- Open an existing file to read data from (/R flag without /D).
- Open a to append results to (/A flag without /D).
- Create a new file or overwrite an existing file to write results to (no /D, /R or /A flags).
- Display an Open File dialog (/D/R or /D/A flags with or without /MULT).
- Display a Save File dialog (/D flag without /R or /A).

### Parameters

*refNum* is the name of a numeric variable to receive the file reference number. *refNum* is set by Open if Open actually opens a file for reading or writing (cases 1, 2 and 3). You use *refNum* with the **FReadLine**, **FStatus**, **FGetPos**, **FSetPos**, **FBinWrite**, **FBinRead**, **fprintf**, and **wfprintf** operations to read from or write to the file. When you're finished, use pass *refNum* to the **Close** operation to close the file.

Open does not set the file reference number when the /D flag is used (cases 4 and 5) but you must still supply a *refNum* parameter.

The following discussion of the *pathName* and *fileNameStr* parameters applies when you are attempting to open a file for reading or writing (cases 2, 3, and 5 above).

The targeted file is specified by a combination of the *pathName* parameter and the *fileNameStr* parameter. There are three ways to specify the targeted file:

Method	How To Use It
Symbolic path and simple file name	Use /P= <i>pathName</i> and <i>fileNameStr</i> , where <i>pathName</i> is the name of an Igor symbolic path (see <b>Symbolic Paths</b> on page II-22) that points to the folder containing the file and <i>fileNameStr</i> is the name of the file.
Symbolic path and partial path	Use /P= <i>pathName</i> and <i>fileNameStrs</i> , where <i>pathName</i> is the name of an Igor symbolic path that points to the folder containing the file and <i>fileNameStr</i> is a partial path starting from the folder and leading to the file.
Full path	Use just <i>fileNameStr</i> , where <i>fileNameStr</i> is a full path to the file.

If you use a full or partial path for *fileNameStr*, see **Path Separators** on page III-451 for details on forming the path.

The targeted file is fully specified if *fileNameStr* is a full path or if both *pathName* and *fileNameStr* are present and not empty strings.

The targeted file is not fully specified in any of these cases:

- as *fileNameStr* is omitted
- *fileNameStr* is an empty string
- *fileNameStr* is not a full path and no symbolic path is specified

### Opening an Existing File For Reading Only

This covers cases 1 (/R without /D).

If the file is fully-specified but does not exist, an error is generated. If you want to detect and handle the error yourself, use the /Z flag.

If the file is not fully-specified, Open displays an Open File dialog.

If a file is opened, *refNum* is set to the file reference number.

### Opening an Existing File For Appending

This covers cases 1 (/R without /D) and 2 (/A without /D).

If the file is fully-specified and exists, it is opened for read/write and the current file position is moved to the end of the file.

If the file is fully-specified but does not exist, the file is created and opened for read/write.

If the file is not fully-specified, Open displays an Open File dialog.

If a file is opened, *refNum* is set to the file reference number.

### Opening a File For Write

This covers case 3 (no /R, /A or /D).

If the targeted file exists, it is overwritten.

If the targeted file does not exist and it is fully-specified and targets a valid path, a new file is created.

If the file is fully-specified and targets an invalid path, an error is generated. If you want to detect and handle the error yourself, use the /Z flag.

If the file is not fully-specified, Open displays a Save File dialog.

If a file is opened, *refNum* is set to the file reference number.

### Displaying an Open File Dialog To Select a Single File

This covers cases 4 (/D with /R or /A).

## Open

Open does not actually open the file but just displays the Open File dialog.

If the user chooses a file in the Open File dialog, the *S\_fileName* output string variable is set to a full path to the file. You can use this in subsequent commands. If the user cancels, *S\_fileName* is set to "".

See the documentation for the /D, /F and /M flags and then read **Displaying an Open File Dialog** on page IV-148 for details.

*refNum* is left unchanged.

### Displaying an Open File Dialog To Select Multiple Files

This covers cases 4 (/D with /R or /A) with the /MULT=1 flag.

Open does not actually open the file but just displays the Open File dialog.

If the user chooses one or more files in the Open File dialog, the *S\_fileName* output string variable is set to a carriage-return-delimited list of full paths to one or more files. You can use this in subsequent commands. If the user cancels, *S\_fileName* is set to "".

See the documentation for the /D, /F, /M and /MULT flags and then read **Displaying a Multi-Selection Open File Dialog** on page IV-149 for details.

*refNum* is left unchanged.

### Displaying a Save File Dialog

This covers cases 5 (/D without /R or /A).

Open does not actually open the file but just displays the Save File dialog.

If the user chooses a file in the Save File dialog, the *S\_fileName* output string variable is set to a full path to the file. You can use this in subsequent commands. If the user cancels, *S\_fileName* is set to "".

See the documentation for the /D, /F and /M flags and then read **Displaying a Save File Dialog** on page IV-150 for details.

*refNum* is left unchanged.

## Flags

/A                    Opens an existing file for appending or, if the file does not exist, creates a new file and opens it for appending.

/C=*creatorStr*    Specifies the file creator code. This is meaningful on Macintosh only and is ignored on Windows. For opening an existing file, creator defaults to "????" which means "any creator". For creating a new file, *creatorStr* defaults to "IGR0" which is Igor's creator code.

/D[=*mode*]      Specifies dialog-only mode.

/D:                 A dialog is always displayed.

/D=1:               Same as /D.

/D=2:               A dialog is displayed only if *pathName* and *fileNameStr* do not specify a valid file.

Use this mode to allow the user to choose a file to be opened by a subsequent operation, such as **LoadWave**.

With /D or /D=1, open presents a dialog from which the user can select a file but does not actually open the file. Instead, Open puts the full path to the file into the string variable *S\_fileName*.

/D=2 does the same thing except that it skips the dialog if *pathName* and *fileNameStr* specify a valid file. In this case, if *pathName* and *fileNameStr* refer to an alias (Macintosh) or shortcut (Windows), the target of the alias or shortcut is returned.

If the user clicks the Cancel button, *S\_fileName* is set to an empty string.

	<p>Use Open/D/R to bring up an Open File dialog. See <b>Displaying an Open File Dialog</b> on page IV-148 for details.</p> <p>Use Open/D/R/MULT=1 to bring up an Open File dialog to select multiple files. See <b>Displaying a Multi-Selection Open File Dialog</b> on page IV-149 for details.</p> <p>Use Open/D to bring up a Save File dialog. See <b>Displaying a Save File Dialog</b> on page IV-150 for details.</p> <p>See <b>Using Open in a Utility Routine</b> on page IV-151 for an example using /D=2.</p> <p>Do not use /Z with /D.</p>
/F= <i>fileFilterStr</i>	/F provides control over the file filter menu in the Open File dialog. See <b>Open File Dialog File Filters</b> on page IV-149 and <b>Save File Dialog File Filters</b> on page IV-151 for details.
/M= <i>messageStr</i>	Prompt message text in the dialog used to select the file, if any.
/MULT=m	Use /D/R/MULT=1 to display a multi-selection Open File dialog. /D/R/MULT=0 or just /D/R displays a single-selection Open File dialog. /MULT=1 is allowed only if /D or /D=1 and /R are specified.
	See <b>Displaying a Multi-Selection Open File Dialog</b> on page IV-149 for details.
/P= <i>pathName</i>	Specifies the folder to look in for the file. <i>pathName</i> is the name of an existing symbolic path.
/R	The file is opened read only.
/T= <i>typeStr</i>	When creating a new file on Macintosh (/A and /R flag omitted), /T sets the Macintosh file type property for the file if it does not already exist. For example, /T="BINA" sets the Macintosh file type to 'BINA'. If /T is omitted the Macintosh file type will be 'TEXT'. Apple has deemphasized Macintosh file types in favor of file name extensions.  For new code, /F is recommended instead of /T.
	When opening an existing file (/A or /R flag used), /T provides control over the file filter menu in the Open File dialog. See <b>Open File Dialog File Filters</b> on page IV-149 for details.
	When creating a new file (/A and /R flag omitted), /T provides control over the file filter menu in the Save File dialog. See <b>Save File Dialog File Filters</b> on page IV-151 for details.
/Z[=z]	Prevents aborting of procedure execution if an error occurs, for example if the procedure tries to open a file that does not exist for reading. Use /Z if you want to handle this case in your procedures rather than having execution abort.  When using /Z, /Z=1, or /Z=2, V_flag is set to 0 if no error occurred or to a nonzero value if an error did occur.
	Do not use /Z with /D.  /Z=0: Same as no /Z. /Z=1: Suppresses normal error reporting. When used with /R, it opens the file if it exists. /Z alone has the same effect as /Z=1. /Z=2: Suppresses normal error reporting. When used with /R, it opens the file if it exists or displays a dialog if it does not exist.

## Details

When Open returns, if a file was actually opened, the *refNum* parameter will contain a file reference number that you can pass to other operations to read or write data. If the file was not opened because of an error or because the user canceled or because /D was used, *refNum* will be unchanged.

If you use /R (open for read), Open opens an existing file for reading only.

## Open

If you use /A, Open opens an existing file for appending. If the file does not exist, it is created and then opened for appending.

If both /R and /A are omitted then Open creates and opens a file. If the specified file does not already exist, Open creates it and opens it for writing. If the file does already exist then Open opens it and sets the current file position to the start of the file. The current file position determines where in the file data will be written. Thus, you will be overwriting existing data in the file.

**Warning:** If you open an existing file for writing (you do not use /R) then you will overwrite or truncate existing data in the file. To avoid this, open for read (use /R) or open for append (use /A).

### Output Variables

The Open operation returns information in the following variables:

V_flag	Set only when the /Z flag is used.  V_flag is set to zero if the file was opened, to -1 if Open displayed a dialog (because the file was not fully-specified) and the user canceled, and to some nonzero value if an error occurred.
S_fileName	Stores the full path to the file that was opened.  If /MULT=1 is used, S_fileName is a carriage-return-separated list of full paths to one or more files.  If an error occurred or if the user canceled, S_fileName is set to an empty string.

When using /D, the value of V\_flag is undefined. Do not use /Z with /D. Use S\_fileName to determine if the user selected a file or canceled.

### Examples

This example function illustrates using Open to open a text file from which data will be read. The function takes two parameters: an Igor symbolic path name and a file name. If either of these parameters is an empty string, the Open operation will display a dialog allowing the user to choose the file. Otherwise, the Open operation will open the file without displaying a dialog.

```
Function DemoOpen(pathName, fileName)
    String pathName      // Name of symbolic path or "" for dialog.
    String fileName       // File name, partial path, full path or "" for dialog.
    Variable refNum
    String str

    // Open file for read.
    Open/R/Z=2/P=$pathName refNum as fileName

    // Store results from Open in a safe place.
    Variable err = V_flag
    String fullPath = S_fileName

    if (err == -1)
        Print "DemoOpen canceled by user."
        return -1
    endif

    if (err != 0)
        DoAlert 0, "Error in DemoOpen"
        return err
    endif

    Printf "Reading from file \"%s\". First line is:\r", fullPath
    FReadLine refNum, str      // Read first line into string variable
    Print str
    Close refNum
    return 0
End
```

### See Also

[Symbolic Paths](#) on page II-22.

[Close](#), [FBinRead](#), [FBinWrite](#), [FReadLine](#)