

```

do
    traceName = StringFromList(index, traceList, ";")
    if (strlen(traceName) == 0)
        break
    endif
    Wave w = TraceNameToWaveRef(graphName, traceName)
    list += GetWavesDataFolder(w, 2) + ";"
    index += 1
while(1)
    if (strlen(list) > 0)
        Save/O/J/W/I/B list
    endif
End

```

See Also

[Exporting Data](#) on page II-177.

SaveData

SaveData [flags] fileOrFolderNameStr

The SaveData operation writes data from the current data folder of the current experiment to a packed experiment file on disk or to a file system folder. “Data” means Igor waves, numeric and string variables, and data folders containing them. The data is written as a packed experiment file or as unpacked Igor binary wave files in a file-system folder.

Warning: If you make a mistake using SaveData, it is possible to overwrite critical data, even entire folders containing critical data. It is your responsibility to make sure that any file or folder that you can not afford to lose is backed up. If you provide procedures for use by other people, you should warn them as well.

SaveData provides a way to save data for archival storage or unload data from memory during a lengthy process like data acquisition. The file or files that SaveData writes are disassociated from the current experiment.

Use SaveData to save experiment data using Igor procedures. To save experiment data interactively, use the Save Copy button in the Data Browser (Data menu).

Parameters

fileOrFolderNameStr specifies the packed experiment file (if /D is omitted) or the file system folder (if /D is present) in which the data is to be saved. The documentation below refers to this file or folder as the “target”.

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

If *fileOrFolderNameStr* is omitted or is empty (""), SaveData displays a dialog from which you can select the target. You also get a dialog if the target is not fully specified by *fileOrFolderNameStr* or the /P=*pathName* flag.

If you are saving to a packed experiment file (/D omitted or /D=0), SaveData writes an HDF5 packed experiment file if the file name extension is ".h5xp". Otherwise it writes a standard packed experiment file (.pxp format). The .h5xp format requires Igor Pro 9 or later.

Flags

/COMP={*minWaveElements*, *gzipLevel*, *shuffle*}

Specifies that compression is to be applied to numeric waves saved when saving as an HDF5 packed experiment file. The /COMP flag was added in Igor Pro 9.00.

minWaveElements is the minimum number of elements that a numeric wave must have to be eligible for compression. Waves with fewer than this many total elements are not compressed.

gzipLevel is a value from 0 to 9. 0 means no GZIP compression.

shuffle is 0 to turn shuffle off or 1 to turn shuffle on.

When compression is applied by SaveData, the entire wave is saved in one chunk. See **HDF5 Layout Chunk Size** on page II-214 for background information and **SaveExperiment Compression** on page II-214 for details.

SaveData

/D [=d]	Writes to a file-system folder (a directory). If omitted, SaveData writes to an Igor packed experiment file. <i>d</i> =1: If the target folder already exists, the new data is “mixed-in” with the data already there (same as /D). <i>d</i> =2: If the target folder already exists, it is completely deleted before the writing of data starts.												
	If in doubt, use /D=1. See Details below.												
/I	Presents a dialog in which you can interactively choose the target.												
/J= <i>objectNamesStr</i>	Saves only the objects named in the semicolon-separated list of object names. When saving as an HDF5 packed experiment file (.h5xp), /J is not supported and returns an error if you use it and provide a non-empty string as <i>objectNamesStr</i> . See Saving Specific Objects below for further discussion of /J.												
/L= <i>saveFlags</i>	Controls what kind of data objects are saved with a bit for each data type: <table border="1"><thead><tr><th><i>saveFlags</i></th><th>Bit Number</th><th>Saves this Type of Object</th></tr></thead><tbody><tr><td>1</td><td>0</td><td>Waves</td></tr><tr><td>2</td><td>1</td><td>Numeric variables</td></tr><tr><td>4</td><td>2</td><td>String variables</td></tr></tbody></table> To save multiple data types, sum the values shown in the <i>saveFlags</i> column. For example, /L=1 saves waves only, /L=2 saves numeric variables only and /L=3 saves both waves and numeric variables. If /L is not specified, all of these object types are saved. This is equivalent to /L=7. All other bits are reserved and must be set to zero. See Setting Bit Parameters on page IV-12 for details about bit settings.	<i>saveFlags</i>	Bit Number	Saves this Type of Object	1	0	Waves	2	1	Numeric variables	4	2	String variables
<i>saveFlags</i>	Bit Number	Saves this Type of Object											
1	0	Waves											
2	1	Numeric variables											
4	2	String variables											
/M= <i>modDateTime</i>	Saves waves modified on or after the specified modification date/time. Waves modified before <i>modDateTime</i> will not be saved. Applies to waves only (not variables or strings). <i>modDateTime</i> is in standard Igor time format — seconds since 1/1/1904. If <i>modDateTime</i> is zero, all waves will be saved, as if there were no /M flag at all.												
/O	Overwrites existing files or folders on disk. Warning: If you use the /O flag and if the target already exists, it will be overwritten without any warning. If you use /O with /D=2, you will completely overwrite the target folder and all of its contents, including subfolders. Do not use /O with /D unless you are absolutely sure you know what your doing.												
/P= <i>pathName</i>	Specifies the folder in which to save the specified file or folder. <i>pathName</i> is the name of an Igor symbolic path, created via NewPath . It is not a file system path like "hd:Folder1:" or "C:\\Folder1\\". See Symbolic Paths on page II-22 for details. When used with the /D flag, if /P= <i>pathName</i> is present and <i>fileOrFolderNameStr</i> is ":" , the target is the directory specified by /P= <i>pathName</i> .												
/Q	Suppresses normal messages in the history area.												
/R	Recursively saves subdata folders.												

/T [=topLevelName] Creates an enclosing data folder in the target with the specified name, *topLevelName*, and writes the data to the new data folder.

If just /T is specified, it creates an enclosing data folder in the target using the name of the data folder being saved. However, if the data folder being saved is the root data folder, the name Data is used instead of root. In packed experiment files and unpacked experiment folders, the root data folder is implicit.

If /T is omitted, the contents of the current data folder are saved with no enclosing data folder.

Saving Specific Objects

If */J=objectNamesStr* is used, then only the objects named in *objectNamesStr* are saved. For example, specifying */J="wave0;wave1;"* will save only the two named waves, ignoring any other data in all data folders.

The list of object names used with */J* must be semicolon-separated. A semicolon after the last object name in the list is optional. The object names must not be quoted even if they are liberal. The list is limited to 1000 characters.

Using */J=""* acts like no */J* at all.

When saving as an HDF5 packed experiment file (.h5xp), */J* is not supported and returns an error if you use it and provide a non-empty string as *objectNamesStr*.

Details

The */M=modDateTime* flag can be used in data acquisition projects to save only those waves modified since the previous save. For example, assume that we have a global variable in the root data folder named *gLastWaveSaveDateTime*. Then this function will write out only those waves modified since the previous save:

```
Function SaveModifiedWaves(savePath)
    String savePath      // Symbolic path pointing to output directory
    NVAR lastSave = root:gLastWaveSaveDateTime
    SaveData/O/P=$savePath/D=1/L=1/M=(lastSave) ":""
    lastSave = datetime
End
```

Because the *datetime* function and the wave modification date have a coarse resolution (one second), this function may sometimes save the same wave twice.

The */M* flag makes sense only in conjunction with the */D=1* flag because */D=1* is the only way to mix-in new data with existing data.

Writing to a Packed Experiment File

When writing to a packed file, *SaveData* creates a standard packed Igor experiment file which you can open as an experiment, browse using the Data Browser, or access using the **LoadData** operation.

If you do not use the */O* (overwrite) flag and the packed file already exists on disk, *SaveData* will present a dialog to confirm which file you want to write to. If you use the */O* flag, *SaveData* will overwrite without presenting a dialog. When writing a packed file, *SaveData* always completely overwrites the preexisting packed file.

Appending to a packed experiment file is not supported because dealing with the possibility of name conflicts (e.g., two waves with the same name in the same data folder in the packed experiment file) would be technically difficult, very slow and errors would result in corrupted files.

Writing to a File-System Folder

When saving to a folder on disk, *SaveData* writes wave files, variables files, and subfolders. This resembles the experiment folder of an unpacked experiment, but it does not contain other unpacked experiment files, such as history or procedures. You can browse the folder using the Data Browser or access it using the **LoadData** operation.

If the target directory does not exist, *SaveData* creates it.

If you do not use the */O* (overwrite) flag and the target folder already exists on disk, *SaveData* will present a dialog to confirm that you want to write to it. *SaveData* checks for the existence of the top file system folder only. For example, if you write data to *hd:Data:Run1*, *SaveData* will display a dialog if *hd:Data:Run1* exists. But *SaveData* will not display a dialog for any folders inside *hd:Data:Run1*.

If you use the */O* flag, *SaveData* will write without presenting a dialog.

SaveData

When writing to a directory, SaveData can operate in one of two modes. If you use /D=1 or just /D, SaveData operates in "mix-in" mode. If you use /D=2, SaveData operates in "delete" mode.

Warning: If the target directory exists and delete mode is used, SaveData deletes the target directory and all of its contents. Then SaveData creates the target directory and writes the data to it. This is a complete overwrite operation.

If the target directory exists and mix-in mode is used, SaveData does not do any explicit deletion. It writes data to the target directory and any subdirectories. Conflicting files in any directory are overwritten but other files are left intact.

To prevent you from inadvertently deleting an entire volume, SaveData will not permit you to target the root directory of any volume. You must target a subdirectory.

The /J flag will not work as expected when writing numeric and string variables in mix-in mode. Instead of mixing-in the specified variables, SaveData will overwrite all variables already in the target. This is because all numeric and string variables in a particular data folder are stored in a single file-system folder (named "variables"), so it is not possible to mix-in. Since waves are written one-to-a-file, /J will work as expected for waves.

When SaveData writes a wave to a file-system folder, the file name for the wave is the same as the wave name, with the extension ".ibw" added. This is true even if the wave in the experiment was loaded from a file with a different name.

Limitations of SaveData

SaveData does not save free waves, free data folders, wave waves or DFREF waves all of which are ignored by SaveData.

When saving as HDF5 packed (.h5xp), the entire experiment must use UTF-8 text encoding. If the experiment uses non-UTF8 text encoding, SaveData displays an alert directing you to Misc→Text Encoding→Convert to UTF-8 Text Encoding and returns an error.

When saving as HDF5 packed (.h5xp), the /J flag is not supported and returns an error if you use it and provide a non-empty string as objectNamesStr.

Outputs

SaveData sets the variable V_flag to zero if the operation succeeded or to nonzero if it failed. The main use for this is to determine if the user clicked Cancel during an interactive save. This would occur if you use the /I flag or if you omit /O and the target already exists. V_flag will also be nonzero if an error occurs during the save.

SaveData sets the string variable S_path to the full file system path to the file or folder that was written. S_path uses Macintosh path syntax (e.g., "hd:FolderA:FolderB:"), even on Windows. When saving unpacked, S_path includes a trailing colon.

Examples

Write the contents of the current data folder and all subdata folders to a packed experiment file:

```
Function SaveDataInPackedFile(pathName, fileName)
    String pathName           // Name of symbolic path
    String fileName            // Name of packed file to be written
    SaveData/R/P=$pathName fileName
End
```

Write the contents of the current data folder and all subdata folders to an unpacked file-system folder:

```
Function SaveDataInUnpackedFolder(pathName, folderName)
    String pathName           // Name of symbolic path
    String folderName          // Name of file-system folder
    SaveData/D=1/R/P=$pathName folderName
End
```

Copy the contents of an unpacked file-system folder to a packed experiment file:

```
Function TransferUnpackedToPacked(path1, folderName, path2, fileName)
    String path1               // Points to parent of unpacked folder
    String folderName          // Name of folder containing unpacked data
    String path2               // Points to folder where file is to be written
    String fileName             // Name of packed file to be written
```