

**Flags**

<i>/AC=createFlags</i>	Enables auto-creation of globals. Bit 0 enables creation of numeric and string global variables which are set to 0 and "" respectively. Variables are auto-created only if they do not already exist. Bit 1 enables creation of waves which are created with zero points. Waves are auto-created only if they do not already exist.
<i>/SDFR=dfr</i>	Specifies a data folder. If you omit <i>/SDFR</i> , the current data folder used. See <b>The <i>/SDFR</i> Flag</b> on page IV-80 for details.

**Output Variables**

StructFill sets the following output variables:

V_Flag	The number of NVAR, SVAR and WAVE fields that were successfully initialized.
V_Error	The number of NVAR, SVAR and WAVE fields that could not be initialized.

If auto-creation is off, a field can not be initialized if the corresponding global variable, string variable, or wave does not exist. If auto-creation is on, a field can not be initialized if there was an error creating the global variable or wave.

When using auto-creation, errors are not reported other than via V\_Error.

If you unexpectedly get non-zero for V\_Error, you can print the structure to see which fields were left null.

**See Also**

**Structures in Functions** on page IV-99

**StructGet**

**StructGet** [*/B=b*] *structVar*, *waveStruct* [*[colNum]*]

**StructGet** */S* [*/B=b*] *structVar*, *strStruct*

The StructGet operation reads binary numeric data from a specified column of a wave or from a string variable and copies the data into the designated structure variable. The source wave or string will have been filled beforehand by **StructPut**.

**Parameters**

*structVar* is the name of an existing structure that is to be filled with new data values.

*waveStruct* is the name of a wave containing binary numeric data that will be used to fill *structVar*. Use the optional *colNum* parameter to specify a column from the structure wave. The contents of *waveStruct* are created beforehand using StructPut.

*strStruct* is the name of a string variable containing binary numeric data. The contents of *strStruct* are created beforehand using StructPut.

**Flags**

<i>/B=b</i>	Sets the byte ordering for reading of structure data. <i>b=0</i> : Reads in native byte order. <i>b=1</i> : Reads bytes in reversed order. <i>b=2</i> : Default; reads data in big-endian, high-byte-first order (Motorola). <i>b=3</i> : Reads data in little-endian, low-byte-first order (Intel).
<i>/S</i>	Reads binary data from a string variable, which was set previously with StructPut.

**Details**

The data that are stored in *waveStruct* and *strStruct* are in binary format so you can not directly view a meaningful representation of their contents by printing them or viewing the wave in a table. To view the contents of *waveStruct* or *strStruct* you must use StructGet to export them back into a structure and then retrieve the members.