

## #undef

Igor ignores unknown pragmas such as pragmas introduced in later versions of the program.

Currently Igor supports the following pragmas:

```
#pragma rtGlobals = value
#pragma version = versionNumber
#pragma IgorVersion = versionNumber
#pragma hide = value
#pragma ModuleName = name
#pragma IndependentModule = name
#pragma rtFunctionErrors = value
#pragma TextEncoding = "textEncodingName" // Igor Pro 7.00
#pragma DefaultTab = {<mode>,<width in points>,<width in spaces>} // Igor Pro 9.00
```

### See Also

[Pragmas](#) on page IV-52

[The rtGlobals Pragma](#) on page IV-52

[The version Pragma](#) on page IV-54

[The IgorVersion Pragma](#) on page IV-54

[The hide Pragma](#) on page IV-54

[The ModuleName Pragma](#) on page IV-54

[The IndependentModule Pragma](#) on page IV-55

[The rtFunctionErrors Pragma](#) on page IV-55

[The TextEncoding Pragma](#) on page IV-55

[The Default Tab Pragma For Procedure Files](#) on page III-406

## #undef

### **#undef symbol**

A #undef statement removes a nonglobal *symbol* created previously by #define. See [Conditional Compilation](#) on page IV-108 for information on undefining a global *symbol*.

### See Also

The [#define](#) statement and [Conditional Compilation](#) on page IV-108 for more usage details.

## Abort

### **Abort [errorMessageStr]**

The Abort operation aborts procedure execution.

#### Parameters

The optional *errorMessageStr* is a string expression, which, if present, specifies the message to be displayed in the error alert.

#### Details

Abort provides a way for a procedure to abort execution when it runs into an error condition.

### See Also

[Aborting Functions](#) on page IV-112 , [Aborting Macros](#) on page IV-124, and [Flow Control for Aborts](#) on page IV-48. The [DoAlert](#) operation.

## AbortOnRTE

### **AbortOnRTE**

The AbortOnRTE flow control keyword raises an abort when a runtime error has occurred in a user-defined function.

AbortOnRTE should be used after a command that might give rise to a runtime error.

You can place AbortOnRTE immediately after a command that might give rise to a runtime error that you want to handle instead of allowing Igor to handle it by halting procedure execution. Use a **try-catch-endtry** block to catch the abort, if it occurs.