```
Function DemoTryCatch2(b)
   Variable b

   Print "DemoTryCatch2 A"
   AbortOnValue b==3,99
   Print "DemoTryCatch2 B"
End
```

User Abort Key Combinations

You can abort procedure execution by clicking the Abort button in the status bar or by pressing the following user abort key combinations:

Command-dot	Macintosh only
Ctrl+Break	Windows only
Shift+Escape	All platforms

Constants

You can define named numeric and string constants in Igor procedure files and use them in the body of user-defined functions.

Constants are defined in procedure files using following syntax:

```
Constant <name1> = <literal number> [, <name2> = <literal number>]
StrConstant <name1> = <literal string> [, <name2> = <literal string>]
```

For example:

```
Constant kIgorStartYear=1989, kIgorEndYear=2099
StrConstant ksPlatformMac="Macintosh", ksPlatformWin="Windows"

Function Test1()
   Variable v1 = kIgorStartYear
   String s1 = ksPlatformMac
   Print v1, s1
End
```

We suggest that you use the “k” prefix for numeric constants and the “ks” prefix for string constants. This makes it immediately clear that a particular keyword is a constant.

Constants declared like this are public and can be used in any function in any procedure file. A typical use would be to define constants in a utility procedure file that could be used from other procedure files as parameters to the utility routines. Be sure to use precise names to avoid conflicts with public constants declared in other procedure files.

If you are defining constants for use in a single procedure file, for example to improve readability or make the procedures more maintainable, you should use the `static` keyword (see **Static** on page V-906 for details) to limit the scope to the given procedure file.

```
static Constant kStart=1989, kEnd=2099
static StrConstant ksMac="Macintosh", ksWin="Windows"
```

Names for numeric and string constants are allowed to conflict with all other names. Duplicate constants of a given type are not allowed, except for static constants in different files and when used with the override keyword. The only true conflict is with variable names and with certain built-in functions that do not take parameters such as pi. Variable names, including local variable names, waves, NVARs, and SVARs, override constants, but constants override functions such as pi.