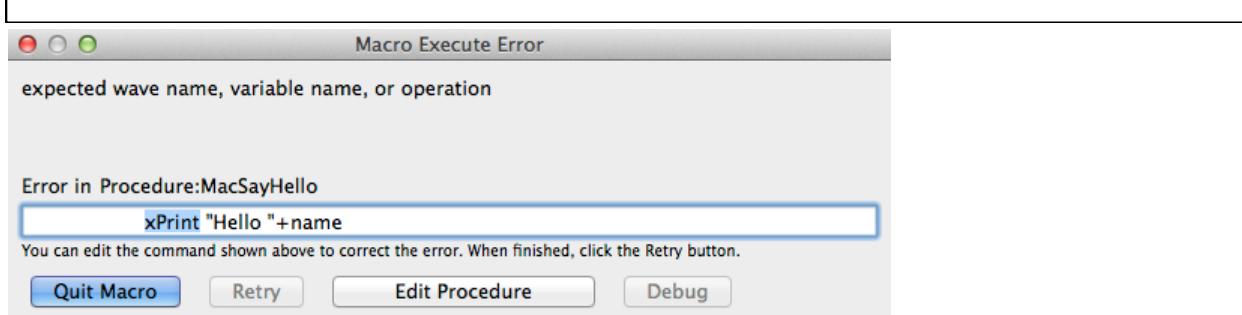


## Chapter IV-4 — Macros



Notice the outline around the line containing the error. This outline means you can edit the erroneous command. If you change “xPrint” to “Print” in this dialog, the Retry button becomes enabled. If you click Retry, Igor continues execution of the macro. When the macro finishes, take a look at the Procedure window. You will notice that the correction you made in the dialog was put in the procedure window and your “broken” macro is now fixed.

## Macro Syntax

Here is the basic syntax for macros.

```
<Defining keyword> <Name> ( <Input parameter list> ) [ :<Subtype>]
    <Input parameter declarations>

    <Local variable declarations>

    <Body code>
End
```

### The Defining Keyword

<Defining keyword> is one of the following:

#### Defining Keyword   Creates Macro In

Window	Windows menu
Macro	Macros menu
Proc	—

The Window keyword is used by Igor when it automatically creates a window recreation macro. Except in rare cases, you will not write window recreation macros but instead will let Igor create them automatically.

### The Procedure Name

The names of macros must follow the standard Igor naming conventions. Names can consist of up to 255 characters. Only ASCII characters are allowed. The first character must be alphabetic while the remaining characters can include alphabetic and numeric characters and the underscore character. Names must not conflict with the names of other Igor objects, functions or operations. Names in Igor are case insensitive.

Prior to Igor Pro 8.00, macro names were limited to 31 bytes. If you use long macro names, your procedures will require Igor Pro 8.00 or later.

### The Procedure Subtype

You can identify procedures designed for specific purposes by using a subtype. Here is an example:

```
Proc ButtonProc(ctrlName) : ButtonControl
    String ctrlName

    Beep
End
```