

Call Statement

Description

Executes an external subroutine.

Syntax

```
Call [subroutine | @VARIABLE | @MATRIX (index)]  
    [(argument [ , argument... ])]
```

Parameters

The Call statement has the following parameters.

Parameter	Description
<i>Subroutine</i>	The Call subroutine format transfers program control directly to the specified subroutine. Control will return to the calling program at the line following the Call statement.
@VARIABLE or @MATRIX	Can be used to indirectly call an external subroutine. The subroutine is called indirectly by first assigning the subroutine name as the value of a variable or matrix, and then using the Call @ statement. In the following example, the subroutine FEDTAX is called: ITAX = "FEDTAX" Call @ITAX The number of arguments passed must be less than or equal to the actual number of parameters specified in the Compile Subroutine header in the subroutine.
<i>Argument</i>	A variable, matrix, constant, or expression that is used to pass values to and from the calling program and the subroutine. The number of arguments in the calling program must match the number of arguments defined in the subroutine being called. If there is no value for an argument, include the argument delimiter. An argument can consist of one or more expressions, separated by commas and enclosed in parentheses.

See also

[Declare](#), [Function](#), [Return](#), [Subroutine](#)

Example

```
Call Msg(@window, "Hello world!")  
* OR  
Declare Subroutine Msg  
Msg(@window, "Hello world!")  
* OR  
SUB_NAME = "Msg"  
Call @SUB_NAME(@window, "Hello world!")
```