

Subroutine Statement

Description

Identifies the source code that follows as a subroutine. Use in the first line of code.

Note: *The maximum length of a stored procedure executable is 64 Kilobytes. This limit applies only to the compiled object. The subroutine source code may exceed the 64 Kilobyte limit.*

Syntax

```
Begin Condition
  Pre:
  Post:
End Condition
[Compile] Subroutine name(DataType parameter [DataType
parameter, ...])
```

Parameters

The Subroutine statement has the following parameters.

Parameter	Description
<i>name</i>	Any legal variable identifier can be used as the name of the subroutine.
<i>DataType</i>	OpenInsight does not require data typing of subroutine parameters. Specify the data type before the parameter identifier. Mat is available as a data type.
<i>parameter</i>	A parameter is optional, and can contain a variable, matrix, constant, or expression. If used, the parameter specification must be enclosed in parentheses. If no parameter is specified, the parentheses are omitted. Multiple parameters are delimited with commas. Parameters are formal definitions for arguments (actual values) to be passed to and from the calling procedure and the subroutine. The calling procedure must provide the same number of arguments as parameters specified in the called function. If fewer arguments are provided, those not provided will be unassigned. If more are provided, an error will occur.

See also

[Function](#), [Mat](#)

Example

```
/* The following code fragment illustrates the format of a subroutine program. */
```

```
Compile Subroutine TEST_TABLE(CharStr volume, |  
CharStr database,CharStr table_name, CharStr alias_name,| Integer status)
```

```
Begin Condition
```

```
Pre:
```

```
Post:
```

```
End Condition
```

```
Declare Function Set_FSError, Set_Status, valid.file.name, | get_bfs_info
```

```
Declare Subroutine get_volume_rec, rtp49
```

```
Convert @LOWER_CASE to @UPPER_CASE in table_name
```

```
Convert @LOWER_CASE to @UPPER_CASE in alias_name
```

```
Convert @LOWER_CASE to @UPPER_CASE in database
```

```
If table_name Else
```

```
estat = Set_Status(TRUE$, SSP_TABLE_LIST_NULL_ERR$, '')
```

```
status = FALSE$
```

```
Return
```

```
End
```

```
GoSub MainProcess
```

```
Return
```