

## Exec and Include

Executing a program is the process of running all the commands in a program file. There are several ways to execute an EViews program: pushing the *Run button* on an open program window, using the *exec* command, using the *run* command, or using the *include* statement. The **exec** and **run** commands allow you to write general programs that execute other programs. Alternatively, the **include** keyword allows you to include the contents of a program file in another program file.

The **run** command has several shortcomings when used inside a program, the most notable of which is that all subsequent program execution will halt after the **run** command has executed its program. Therefore, it is recommended to never use the **run** command inside a program.

The choice of whether to use **exec** or **include** is based upon the different behavior of each. With each there are differences in behavior of program logs, variables, subroutines and path.

In the discussion that follows we define the program that uses **exec** or **include** to execute another program as the *parent* program, and the program that is being executed as the *child* program.

## Log windows

When executing a program in EViews, it may be useful to keep track of what is happening during execution. Log windows allow you to follow program progression. Log windows may be switched on using the *logmode* command. The *logmode* and *logmsg* commands give the ability to specify the name of the log window and direct the messages to the window with the specified name. The following tables describe the behavior of log windows when *exec* or *include* are used inside a program to execute another program:

### Include Statement

Program commands		Log output: EViews 11 and earlier
<b>Parent</b>	<b>Child</b>	<b>Log:Parent</b>
<i>logmode 1</i> <i>include child</i> <i>logmsg Msg from parent</i>	<i>logmsg Msg from child</i>	<i>Msg from child</i> <i>Msg from parent</i>

Program commands		Log output: EViews 11 and earlier
<b>Parent</b>	<b>Child</b>	<b>Log:Parent</b>
<i>logmode 1</i> <i>include child</i> <i>logmsg Msg from parent</i>	<i>logmode 1</i> <i>logmsg Msg from child</i>	<i>Msg from child</i> <i>Msg from parent</i>

Program commands		Log output: EViews 11 and earlier
<b>Parent</b>	<b>Child</b>	<b>Log:Parent</b>
<i>include child</i> <i>logmsg Msg from parent</i>	<i>logmode 1</i> <i>logmsg Msg from child</i>	<i>Msg from child</i> <i>Msg from parent</i>

where

- *Log output is the is the contents of the log window after running Parent*

## Exec command

Program commands		Log output: EViews 11	Log output: EViews 10 and earlier
Parent	Child	Log:Parent	Log:Parent
<i>logmode 1</i> <i>exec child</i> <i>logmsg Msg from parent</i>	<i>logmsg Msg from child</i>	<i>Msg from parent</i>	<i>Msg from Child</i> <i>Msg from parent</i>

Program commands		Log output: EViews 11	Log output: EViews 10 and earlier
Parent	Child	Log:Parent	Log:Parent
<i>logmode 1</i> <i>exec child</i> <i>logmsg Msg from parent</i>	<i>logmode 1</i> <i>logmsg Msg from child</i>	<i>Msg from parent</i>	<i>Msg from child</i> <i>Msg from parent</i>
		Log:Child	
		<i>Msg from child</i>	

Program commands		Log output: EViews 11	Log output: EViews 10 and earlier
Parent	Child	Log:Child	Log:Parent
<i>exec child</i> <i>logmsg Msg from parent</i>	<i>logmode 1</i> <i>logmsg Msg from child</i>	<i>Msg from child</i>	<i>Msg from child</i>

where

- *Log output is the is the contents of the log window(s) after running Parent*

## Program variables

Program variables are variables that can be used in place of numeric or string values in EViews programs. The following table describes whether variables can be accessed when **exec** or **include** are used inside a program to execute another program:

### Include Statement

Parent	Child	
<code>%varstr = "somestring" include Child</code>	<code>%newstr = %varstr</code>	Child has access to %varstr from Parent
<code>%varstr = "somestring" include Child</code>	<code>%varstr = "new string"</code>	Child will overwrite the contents %varstr in Parent

### Exec Command

Parent	Child	
<code>%varstr = "somestring" exec Prog2</code>	<code>%newstr = %varstr</code>	Error: Child has no access to %varstr from Parent

## Program subroutines

A subroutine is a collection of commands that allow you to perform a given task without duplicating the commands. Subroutines can be used from one program to perform the same task in other programs. The following table describes if subroutines can be called when *exec* or *include* are used inside a program to execute another program:

### Include Statement

Parent	Child	
<pre>wfcreate a 2010 2019 include Child call z_square</pre>	<pre>subroutine z_square series x = 2^2 endsub</pre>	Parent can call subroutine z_square from Child.

### Exec Command

Parent	Child	
<pre>wfcreate a 2010 2019 exec Child call z_square</pre>	<pre>subroutine z_square series x = 2^2 endsub</pre>	Error: Parent can not call subroutine z_square from child.

### Path information

EViews offers functionality in dealing with file path specifications, including expanded support for relative paths using the ".\" and "..\" prefixes. The interpretation of the prefix varies depending on the context in which it is employed. The following table describe the behavior in the various contexts in which paths could be specified.

Context	(no prefix)	.\"	..\"	..\..\"
<b>Include</b>	<i>runpath</i>	<i>linepath</i>	above <i>runpath</i>	above <i>linepath</i>
<b>Exec</b>	<i>addins</i>	<i>linepath</i>	above <i>addins</i>	above <i>linepath</i>
<b>Run/Open/CD/Other</b>	<i>evdata</i>	<i>linepath</i>	above <i>evdata</i>	above <i>linepath</i>

where

- *linepath* is the location of the program file containing the currently executed line. Thus in an included file, it will contain the location of the "child" program, not the location of the "parent" program.
- *runpath* is the location of the program file that is "running", where execution via exec or include does not count as "running". Thus if you run a program that contains included or execs, *runpath* in those included/exec'd files will refer to the location of the parent file, not the subprogram files. If a program was exec'd from the command line, then *runpath* will refer to the EViews data directory.
- *addins* is the location of the add-ins directory
- *evdata* is the current EViews default directory

It is worth noting that the following string functions may be used to obtain the locations of these directories:

- @addinspath - the location of the add-ins directory
- @datapath – the location of the current EViews data directory
- @linepath – the location of the currently executed line • @runpath – the location of the program that is "running", where execs and includes do not count as "running".

Note also that the special "<addins>" keyword may be used as part of a path to indicate the *addins* directory.