

Execute Program Action

The Execute Program Action allows you to execute nearly any program from a Automise build.

Automise can capture the output of console applications (such as command line compilers) and display the captured output in the Automise output window.



If you are executing a native DOS command (ie 'dir'), or a batch file, or you wish to redirect console output to a file, then use the [Run DOS Command action](#) instead.



Unknown Attachment

Wait For Completion

By default, Automise will wait for the program to complete before continuing. However you can turn this option off. When Wait For Completion is turned off, the action will complete as soon as the program begins executing. When Wait For Completion is turned off, program output is not available for capture.

Hide Window

If this option is set, console programs will not run in a visible window.

Log Output

If this option is set, Automise will send all console output to the Automise log. This option is not available if WaitForCompletion is set.

You may need to disable **Log Output** on some programs (very few) which do not flush their console output buffer correctly. These programs may appear to hang when executed inside Automise. Unfortunately, this is a problem within the third party programs themselves and it is not possible for Automise to correct the issue externally. If output needs to be captured, you can sometimes use the [Run DOS Command action](#) to execute the program and redirect the output to a file.

"Program exit code must be..."

If you enable this option, Automise will require the program's exit code to be within a certain range:



If the exit code is outside the specified range, the action will fail.

Advanced Exit Code Checking

If you need more fine-grained control over the action's behaviour, you can read the Action.ReturnCode property from script. The following script fails the action if the return code is any value apart from 2 or 4:



Parsing The Output of a Program

Some programs do not provide accurate enough return codes to determine if an error has occurred. In these cases you may need to parse the actual program output for errors. You can do this by using script in the OnStatusMessage event handler (assuming you have enabled **Log Output**.)

Scripting Info

The Action properties available are :

property ProgramName : WideString;

property Params : WideString;

property StartInDir : WideString;

property LogOutput : WordBool;

property WaitForCompletion : WordBool;

property ReturnCode : integer; // Read only

property HideWindow : WordBool;

property EnableReturnCodeCheck : Boolean;

property ReturnCodeComparator : TFBRunReturnCodeComparator;

property ReturnCodeToCheck : Boolean;

These properties may be set in the BeforeAction and AfterAction Script events.