

Wait For Script Result Action

The Wait for Script Result action is the most generic "Wait For..." action. After each polling interval, the [script event](#) "OnPollWaitCondition" is executed. Set the parameter "KeepWaiting" to false to stop waiting.

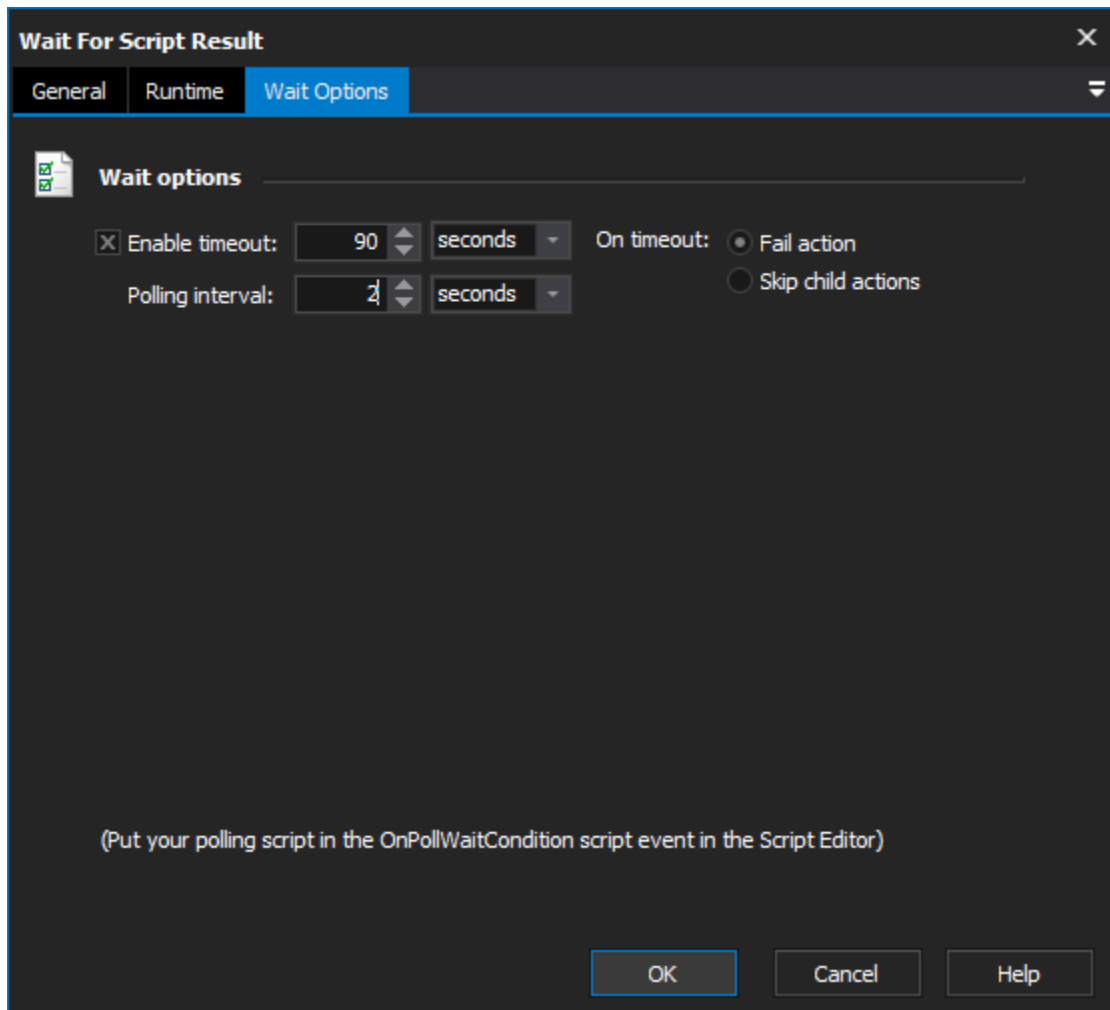
Wait Options

Wait options

Enable timeout

Polling interval

On timeout



The screenshot shows the 'Wait For Script Result' dialog box with the 'Wait Options' tab selected. The dialog has three tabs: 'General', 'Runtime', and 'Wait Options'. The 'Wait Options' tab contains the following settings:

- Wait options** (indicated by a document icon)
- Enable timeout:** A checkbox that is checked, followed by a numeric input field set to '90' and a dropdown menu set to 'seconds'.
- On timeout:** Two radio button options: 'Fail action' (which is selected) and 'Skip child actions'.
- Polling interval:** A numeric input field set to '2' and a dropdown menu set to 'seconds'.

At the bottom of the dialog, there is a note: "(Put your polling script in the OnPollWaitCondition script event in the Script Editor)". At the very bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

In this example the FinalBuilder variable MyCounter is incremented by 2 and then evaluated to see if the value is greater or equal to 10. If the value is less than 10 then the action will poll again after the specified period (set via the Wait Options tab of the action).

If the value is greater or equal to 10 then the action will complete successfully and the next action in the Target will be executed.

Script Editor

Script Language: JavaScript

BeforeActionAfterActionOnStatusMessageOnSucceededOnTimedOutOnPollWaitCondition *

```
1 function OnPollWaitCondition(Action, SecondsPassed, PollCount, KeepWaiting) {
2   // Set KeepWaiting to False to pass through the wait condition
3
4   MyCounter += 2; // Increment my FinalBuilder variable by 2
5
6   // Log Values
7   Action.SendLogMessage("PollCount = " + PollCount);
8   Action.SendLogMessage("MyCounter = " + MyCounter);
9   Action.SendLogMessage("SecondsPassed = " + SecondsPassed);
10  Action.SendLogMessage(" ");
11
12  // If MyCounter is 10 or more then stop waiting.
13  KeepWaiting = MyCounter < 10;
14
15 };
```

Quick HelpValidation MessagesBuild LogBuild HistoryScript EditorWatches

The output would look as follows (assuming variable MyCounter was set to zero before the Wait For Script Result action commenced):

Build Log

Live Log ViewAuto CollapseShow Full LogShow all Error ActionsShow Ignored ErrorsShow In ContextSearch: >>

Message	Date	Start Time	End Time	Run Time	Status
HELP_WaitFor.fbp8	22/05/2015	15:17:30:356	15:17:38:424	00:00:08:068	✓
Default	22/05/2015	15:17:30:355	15:17:38:380	00:00:08:025	✓
Wait For Script Result	22/05/2015	15:17:30:355	15:17:38:379	00:00:08:024	✓
PollCount = 1					
MyCounter = 2					
SecondsPassed = 0					
PollCount = 2					
MyCounter = 4					
SecondsPassed = 2					
PollCount = 3					
MyCounter = 6					
SecondsPassed = 4					
PollCount = 4					
MyCounter = 8					
SecondsPassed = 6					
PollCount = 5					
MyCounter = 10					
SecondsPassed = 8					
Continuing after 8 seconds					