

File Dependency Action

[Automise Professional Edition]

The file dependency action allows you to control the flow of your build based on whether or not a file (or set of files) has changed relative to a time stamp.

For example, you can choose to run a set of compiler actions only if the source files are newer than the compiled application.

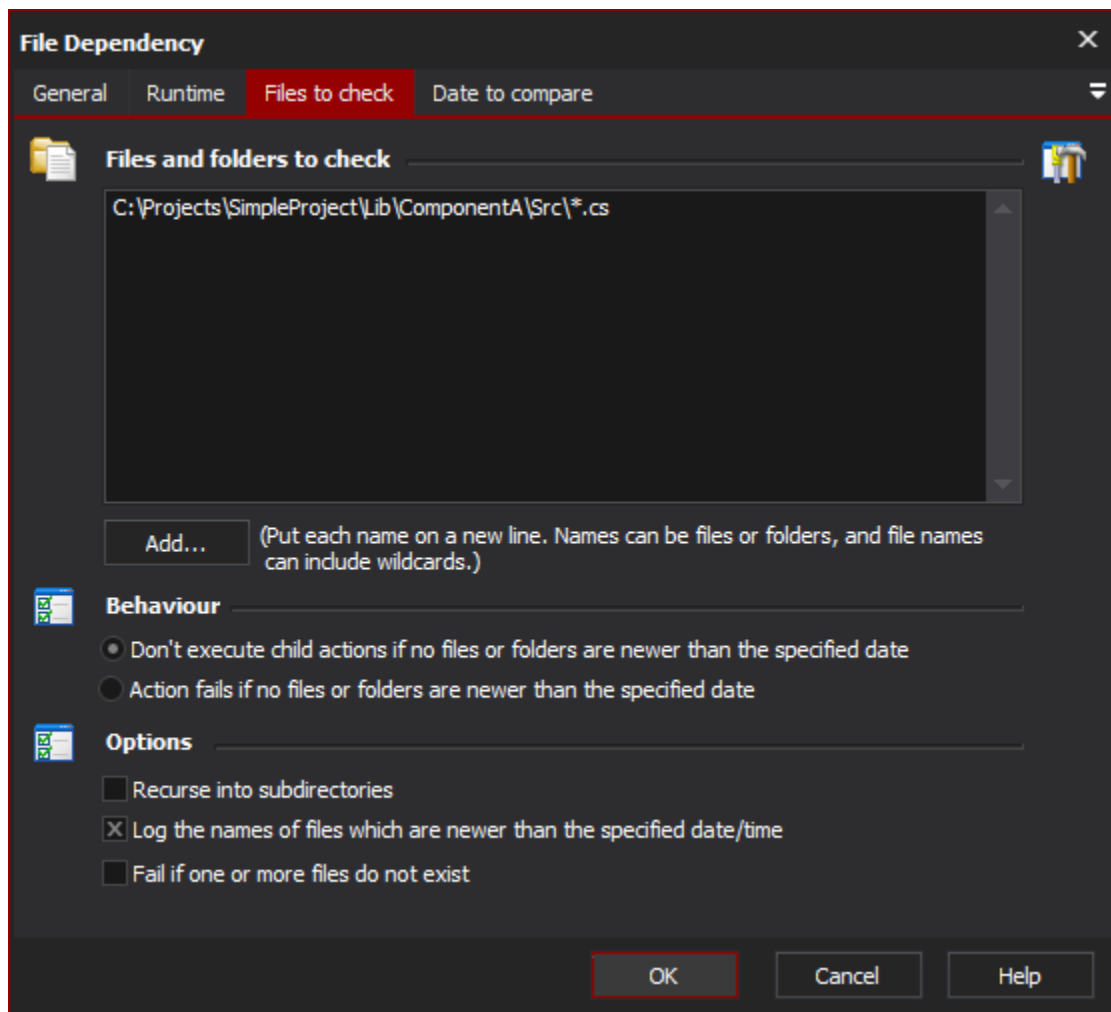
In order to use a file dependency action, you have to choose

(a) the files and folders to check (the "source files") and.

(b) the date timestamp to compare them to.

On this page:

Files To Check



Files and Folders to Check

These are the files which will be compared against the time stamp (the source files, in the example above.) The action only executes its children if at least one of the files in this list is newer than the time stamp.

Files can be specified in the following ways:

Format Example

Full Pathname D:\ComponentA\MainClass.cs

Wildcard Filespec D:\ComponentA*.cs

Directory Pathname D:\ComponentA\

Behaviour

Don't execute child actions if no files or folders are newer than the specified date

The child actions of the File Dependency actions will only be executed if one or more files is newer than the timestamp. Otherwise, Automise will skip to the next sibling action of the File Dependency.

Action fails if no files or folders are newer than the specified date.

Then the action will fail outright if none of the files is newer than the timestamp. This can lead to the build failing completely, or you can use a [Try...Catch](#) block to catch the error.

Options

Recurse into subdirectories

If you choose this option, then Automise will expand subdirectories of directory & wildcard pathnames that you specify. In the example from the screenshot, "Recurse into subdirectories" would return all ".cs" files from all subdirectories of the "ComponentA" directory.

Recurse into subdirectories does not recursively search for plain pathname file names.

Log the names of files which are newer than the specified date/time

If you choose this option, then Automise will log the names of files which are newer than the timestamp.

Fail if one or more files does not exist

This option will cause the action to fail if any of the full pathnames specified do not exist.

The action will not fail if a wildcard filespec fails to match any files, or if a directory is empty. However, the action will fail (regardless of how this option is set) if none of the specified files can be found.

Date to compare

The screenshot shows the 'File Dependency' dialog box with the 'Date to compare' tab selected. The dialog has a dark theme and a red border. It contains three radio button options for selecting the date source. The first option, 'Use specific date:', is currently selected, with a date field showing '12/02/2015' and a time field showing '2:15 PM'. The second option, 'Use date and time that action was last executed', is unselected, but shows the current date and time '6/01/2016 10:39:55 AM' and includes 'Reset' and 'Set to current date and time' buttons. The third option, 'Use date from file, filespec or directory:', is unselected and includes a text field with a file explorer icon. Below this, there are three sub-options: 'Use date of oldest file from selection' (selected), 'Use date of newest file from selection' (unselected), and 'Recurse into subdirectories' (unselected). There is also an 'Always execute if file(s) not found' checkbox. The 'Options' section at the bottom has an 'Ignore files/folders with dates less than' spinner set to '1' minute, and an 'Override and always execute if variable is true:' checkbox with an empty text field. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

File Dependency

General Runtime Files to check **Date to compare**

Date to compare

☐ Use specific date:

12/02/2015 2:15 PM

☒ Use date and time that action was last executed

(Currently 6/01/2016 10:39:55 AM) Reset Set to current date and time

☐ Use date from file, filespec or directory:

(Use wildcards - ie *.exe - to select multiple files)

☒ Use date of oldest file from selection

☐ Use date of newest file from selection

☐ Recurse into subdirectories

☐ Always execute if file(s) not found

Options

☐ Ignore files/folders with dates less than 1 minute after the specified date

☐ Override and always execute if variable is true:

OK Cancel Help

Date to Compare

Specify the kind of timestamp you wish to use. This is the date of the "built files".

Use Specific Date

Enter a specific (static) date and time combination.

Use date and time that action was last executed

Automise keeps track of the last time that each File Dependency action has run.

Click the "Reset" button to set the last run time back to January 1, 2005.

Click the "Set to current date and time" to bring the last run time value up to the present moment.

Use date from file, filespec, or directory

As indicated, you can use a file (or group of files) to determine the timestamp.

Enter the full pathname to a file, a wildcard pathname (as in the screenshot), or a directory name. You can choose to have either the oldest or the newest file used for the timestamp.

Use date to oldest file from selection

Specify this option to use the oldest date of the files listed for the date comparison.

Use date to newest file from selection

Specify this option to use the newest date of the files listed for the date comparison.

Recurse into subdirectories

Specify this option to move down the sub-directories when performing the compare.

Always execute if files not found

Specify this option to always execute the action even if files are not located.

Options

Ignore files/folders with dates less than X minutes after the specified date

If you select this option then the timestamp will be brought forward the specified number of minutes to account for "jitter" in the timestamps of object files, etc.

Override and always execute if variable is true

If this option is selected, then the chosen Automise variable (which needs to be a boolean variable) is evaluated before file dependencies are calculated. If the variable evaluates to true then the dependency evaluation is skipped and the child actions are run automatically.

(This option is useful if you want to sometimes force your build to go ahead, regardless of timestamps.)