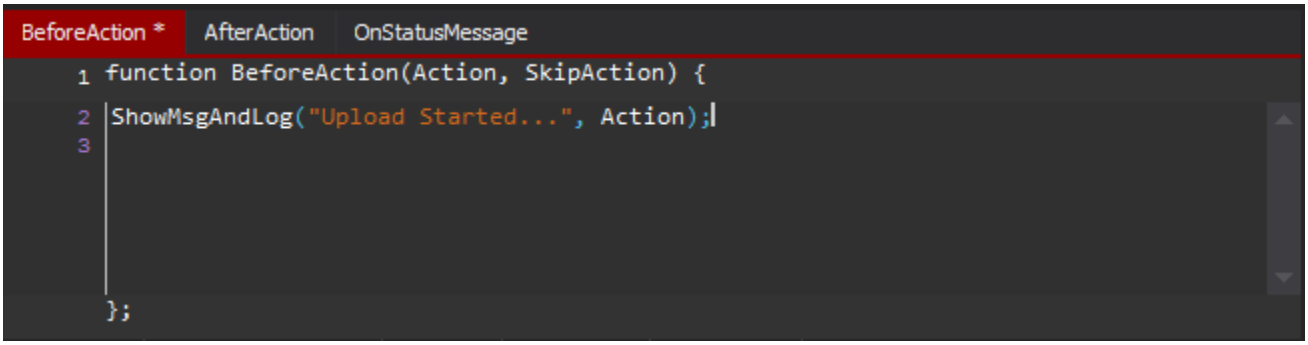
Action Script Events

Each Action in Automise has **BeforeAction**, **AfterAction,** and **OnStatusMessage** script events. Some Actions (such as Iterators) define more script events.

The action Script Events are shown at the bottom of the Automise IDE in the Script Editor tab, see screenshot.



Scripting Events

Script Event	Parameters		Description
BeforeAction	`Action : <TAction>`	The instance of the current action. Allows access to the action properties and methods. All properties are set to the values provided by the action editor or their defaults.	Called before the action is executed. When called all properties on the action have been initialised to those provided in the action editor dialog. Properties which have not been provided are set to their defaults. Use this event to change anything about the action or perform operations which need to occur before the action is run. Also this event can be used to skip an action entirely. If the action is skipped it will report as such in the log, and no more processing of the action or its scripts will occur. Run-time errors in this script will stop the action from running and the action will report as failed.
	`SkipAction : Boolean`	An out parameter which allows for the action to be skipped during a build. Default is ***false***.	
After Action	`Action : <TAction>`	The instance of the current action. Allows access to the action properties and methods. All properties are set to the values used during the run of the action.	Called after the action has executed. All properties on the action will be the same as when the action was run. Any properties which change during the run will be available at this point (e.g. Exit codes properties). This event allows for the handling of action error states that where are not directly handled by the action itself. For example if an certain error code is acceptable it can be logged and ignored through this event. Conversely a certain successful run condition that is not desirable could be reported through this event. To achieve these outcomes set the ActionResult and Continue parameters to values which reflect the true outcome of the actions run. Run-time errors produced by this script will cause the action to report as failed. Setting of either the ActionResult or Continue parameters will not override this.
	`ActionResult : boolean`	Indicates if the action succeeded or failed. Allows for the handling or overriding of the actions status during a build.	
	`Continue : boolean`	Indicates if the build should continue after this action has completed or not. Return false to stop the build, return true to ignore the build result and continue the build. Default is ***unset***.	

OnStatusMessage	`Action : <TAction>` The instance of the current action. Allows access to the action properties and methods. All properties are set to values used during the run of the action.	Called whenever the action generates a log message. Actions will generate this event when the action has received output from the underlying tool, or when the action itself has something to report. The StatusMessage may contain zero-to-many lines of text and is in the format directly seen in the log. As formatting depends on the tool being used we advise using the RegExp function to parse the output if required. Also please review the ***TString*** object type for more information on how to access the contents of the messages. Run-time errors produced by this script will cause the action to report as failed.
	`StatusMessage : TStatusMessage` The status message object contains all information relating to the message being logged (Lines, MessageText, MessageTitle, and Progress).	