Action Properties And Methods

Action Properties and Methods

The Action object is passed into Script Events. The methods and properties available on the Action object are as follows:

Procedure/Methods

Method Call	Example	Description
`procedure SetLogTitle (value As String)`	`Action. SetLogTitle ("Building...");`	Allows you to set the title of the Automise log.
`procedure SendLogMessage (MessageText As String)`	`Action. SendLogMessage ("Built!");`	Sends a message to the output window.
`procedure Echo (MessageText As String)`	`Action.Echo ("Built!");`	Same as SendLogMessage (see above).
`procedure SendProgress (Value As String, progress As Integer)`	`Action. SendProgress ("Help Built.", 60);`	Sends a progress message to the Run Status window. This enables the action to report it's progress as it's executing. Progress is a percentage and should therefore range between 0 and 100.

Properties

Method Call	Example	Description
`function ExpandExpression (expr As String) As String`	`var exePath = Action. ExpandExpression("%ProjectDir% \\MyExecutable.exe");`	This expands the string passed in by substituting Automise variables when %<variable>% is encountered.

`function ChildActions (index As Integer) As OleVariant`	`var child = Action. ChildActions(1);`
	or
	`for (var i = 0; i < Action. ChildActionCount - 1; i++){`
	`var child = Action. ChildActions(i);`
	`if (child != null){`
	`if (child.ActionName === "Compile Delphi Win32 Project"){`
	`child.CompilerOpt. DebugInfo = IncludeDebugInfo;`
	`}`
	`}`
	`}`
		Allows access to child actions, index is zero based. Use ChildActionCount property to get a count of the child actions.
		
		In the second example the script turns off debug info on all child Delphi compiler actions. Add this code to the BeforeAction event handler of the parent action typically a group action.

`function Parent As OleVariant`	`var parent = Action.Parent();`	Returns the Parent action of the current action. This will return null for root level actions.

Functions

Method Call	Example	Description
`property ActionComment As String`	`var curActionComment = Action.ActionComment;`	Provides access to the action comment.
`property ActionLogTitle As String`	`Action.LogTitle  = "Full Build - VB6";`	Allows you to change the title of the action's entry in the output window. This has no effect if set from the AfterAction event (since the log entry has already been made.)
`property ActionName As String`	`var curActionName = Action.Name;`	Provides access to the Action's name
`property ChildActionCount As Integer`	`var curChildCount = Action.ChildCount;`	Returns the number of Child Actions the action has.
`property Description As String`	`var description = Action.Description;`	Provides access to the Action's description
`property IgnoreFailure As Boolean`	`Action.IgnoreFailure = true;`	Provides access to the Action's IgnoreFailure flag
`property LogToVariable As String`	`Action.LogToVariable = "ActionLog";`	Provides access to the Action's LogToVariable flag
`property PauseInterval As Cardinal`	`Action.PauseInterval = 100;`	Provides access to the Action's PauseInterval property
`property SuppressStatusMessages As Boolean`	`Action.SuppressStatusMessages = false;`	Provides access to the Action's SuppressStatusMessages property