



**FinalBuilder
ActionStudio
Manual**

Copyright © 2006 VSoft Technologies Pty Ltd

Table of Contents

	0
Part I ActionStudio Introduction	1
1 Getting Started With FinalBuilder	1
2 What are FinalBuilder Plug-Ins	1
3 Action Implementation Options	2
4 ActionStudio IDE Overview	2
Action Package Tree	3
ActionStudio IDE Help	4
Action Types	5
Standard Action.....	5
Execute Program Action.....	5
Iterator Action.....	6
Loading Custom Actions In FinalBuilder	7
Protected Action Packages	7
Part II Examples	7
1 Example Actions	7
2 Creating Script Based Actions	8
Creating a HelloWorld Example Action	8
3 Creating .Net Based Actions	18
Define a .Net Action In ActionStudio	18
Creating A .Net Action Class	19
Referencing the Action Class	23
Calling .NET Methods at Design Time	24
.NET Custom Action API Reference	25
4 Creating PowerShell Based Actions	25
Part III Reference	26
1 Available Objects	26
Action Object	27
ActionEvents Object	28
Execute Context Object	28
Validation Context Object	29
CustomActionEvent Object	29
Page Object	30
Properties Object	30
FBProperty Object	31
CommandLine Object	31
FBStrings Objects	32
2 Available Global Funtions/Objects	33
VBScript	33
JScript	45
JScript Object: Object.....	46
JScript Object: Number.....	47

JScript Object: Boolean.....	49
JScript Object: Error.....	49
JScript Object: String.....	49
JScript Object: Array.....	55
JScript Object: Date.....	57
JScript Object: Enumerator.....	65
JScript Object: Arguments.....	65
JScript Object: Function.....	66
JScript Object: ActiveXObject.....	67
JScript Object: Debug.....	68
JScript Object: Math.....	68
JScript Object: RegExp.....	72
JScript Object: RegExp.....	74
3 Upgrading Older Action Packages	75
Upgrading .NET Actions	75
COM-Based Actions	76
 Index	 0

1 ActionStudio Introduction

FinalBuilder ActionStudio is a companion program to FinalBuilder. Use this manual as a reference for working with FinalBuilder ActionStudio.

ActionStudio is an IDE (Integrated Development Environment) for creating plugins for FinalBuilder. Using ActionStudio, an action package is created containing one or more Actions. Each action is defined by specifying its properties, events, and property pages. The implementation of the action can be either ActiveScript based (JScript or VBScript), PowerShell based or .NET based (with an implementation written in any .NET language.)

ActionStudio features include:

- Intuitive and modern UI to easily develop plugins
- Embedded form designer to create and modify property pages
- Code Completion to help write script code
- Integrated help

1.1 Getting Started With FinalBuilder

This document provides a quick introductory section and reference to common ActionStudio types, properties and methods. It is highly recommended that you follow the [creating a hello world example](#) action as a first step once you have read the Introductory sections.

For a hands-on introduction, look at the [Example Actions shipped with FinalBuilder](#)

If you are planning to develop actions in .NET, the .NET Custom Action API includes a [comprehensive API reference](#) and IntelliSense help for Visual Studio.

1.2 What are FinalBuilder Plug-Ins

FinalBuilder Plugins allow customers and third parties to add extra functionality to FinalBuilder. These plugins are typically referred to as "Custom Action Packages."

Action packages are not compiled, but instead are XML file definitions. ActionStudio can load and save these xml files. The action packages (with file extension .FBAP) are stored in the directory FinalBuilder\ActionDefs.

FinalBuilder will automatically load any Action Package files in the ActionDefs folder when it starts up.

For Script and PowerShell based actions, the action package contains all of the action definitions and the implementation script code.

For .NET based custom action packages, the action and property page definitions are inside the Action Package file, but the action implementation is a .NET assembly. This assembly must be stored in the relevant subdirectory of the Assemblies directory inside the ActionDefs directory. For .NET 2.0 and 3.5, the correct directory is ActionDefs\Assemblies\2.0.

Some built-in FinalBuilder actions are not located in .FBAP files, but are instead loaded from .BPL files in the FinalBuilder program directory. End users cannot create this type of action package, they are used for built-in actions only.

1.3 Action Implementation Options

Actions created in ActionStudio can be implemented in one of three different ways: [ActiveScript \(JScript/VBScript\)](#), [PowerShell](#), and [.Net](#).

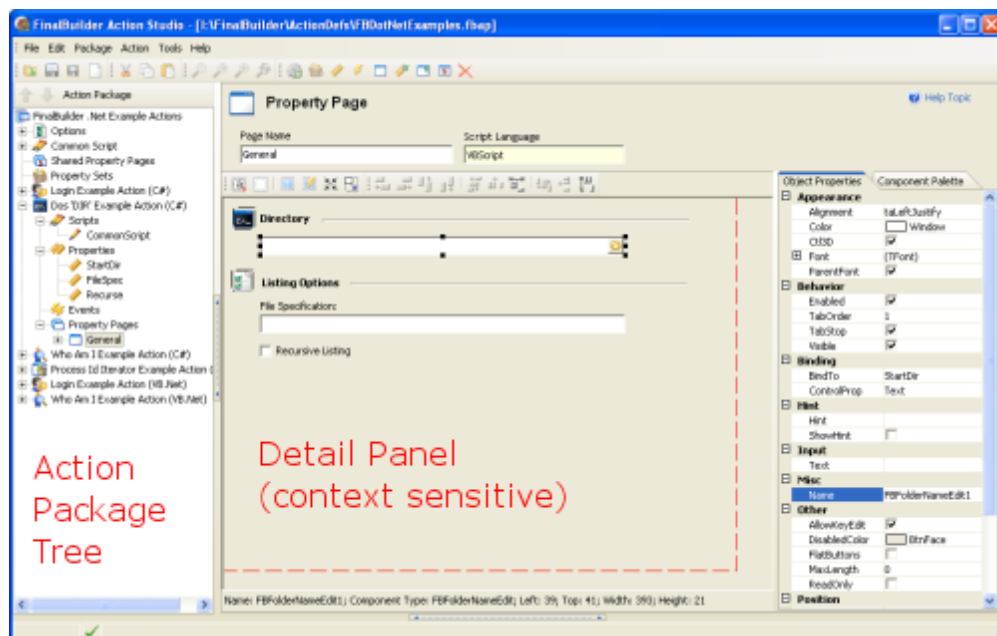
For simple actions, it is recommended to use script. Script based actions have the advantage that they can be developed completely in ActionStudio, whereas .NET based actions must be developed and debugged in a separate IDE (eg. Microsoft Visual Studio.)

Creating a .NET based action in a .NET IDE has the advantage of higher-level debugging and improved familiarity for developers. It is also possible to leverage the .NET Framework and .NET Custom Action API to quickly and easily create more powerful actions.

1.4 ActionStudio IDE Overview

ActionStudio contains the following key user interface elements:

- [Action Package Tree](#)
- Detail Panel



The Detail Panel changes depending on the active node in the Action Package Tree. In the example above, we are editing a property page of an action.

The Detail Panel can show form editors with property pages (as above) or options pages, script editors (for script events) and property editors (for action properties and option properties.)

1.4.1 Action Package Tree

The Action Package Tree displays the contents of the action package. Each action package contains nodes for Options, Common Script, Shared Property Pages and Property Sets. They also contain an Action node for each action in the package file.

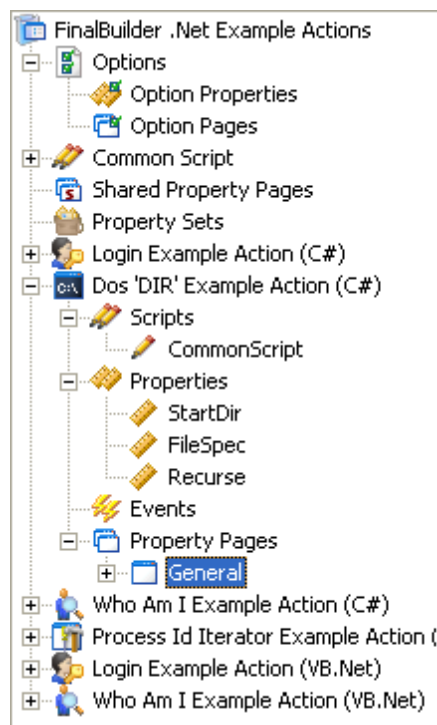
The **Options** node defines global properties that are available to all instances of the action types in the Action Package.

The **Common Script** node is used to define script functions which can be accessed from any script event in the Action Package.

The **Shared Property Pages** node is used to define property pages which can be added to any action in the Action Package.

The **Property Sets** node is used to define custom Property Set types, which are used by actions in this Action Package.

Each **action node** defines a single action which can be added to FinalBuilder projects by the end user.



The Options Node contains the following child nodes:

- **Option Properties**

Like Action Properties, this node contains any option properties that are saved by the FinalBuilder IDE. Clicking on a property node will bring up the Property Editor. To add a new Option Property, choose Add New Option Property from the Package menu.

- **Option Pages**

Option Pages appear in the FinalBuilder IDE Options dialog (under Tools -> Options) and allow the user to edit the Option Properties. Clicking on an option page will bring up the Form Editor. To add a new Option Page, choose Add New Option Page from the Package menu.

Each action contains the following child nodes:

- **Scripts node**

Depending on the type of action (eg. Script based, .NET based) the Scripts node will have different child script nodes relevant to the action type. Clicking on a script node will bring up the Script Editor.

- **Properties node**

The properties node contains the properties that can be set on an instance of the action. To add a new property, choose Add New Property from the Action menu. Clicking on a property node will bring up the Property Editor.

- **Events node**

The events node contains custom script events that can be called as part of the action execution. When a user adds the action a project, they can place code to handle these events into the Script Editor in the FinalBuilder IDE. Click on the Events node will bring up the Events Editor.

- **Property Pages node**

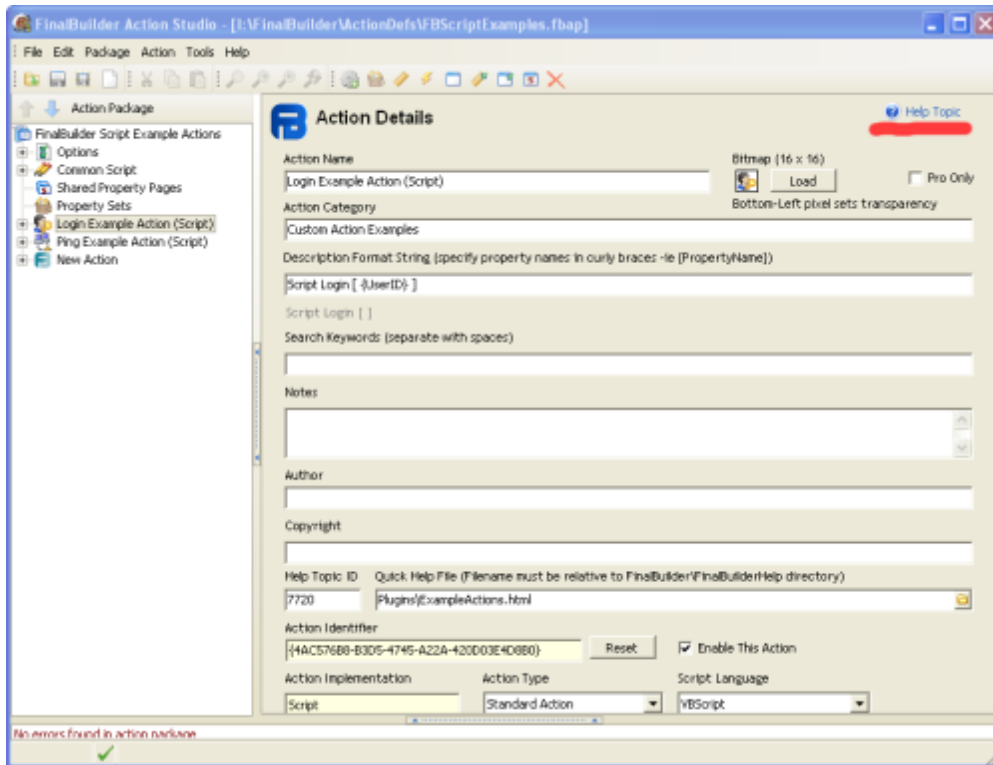
The property pages node contains property pages for the FinalBuilder user to customise the action properties. These are the pages which appear when you double-click an action in the FinalBuilder IDE. To add a new property page, choose Add New Property Page from the Action menu.

An Action Package can contain any number of Actions, and there is no restriction on what type of actions can be added to an action package.

1.4.2 ActionStudio IDE Help

The ActionStudio IDE contains integrated context-sensitive help for all of its major features.

To see the context-sensitive help for a section, click the Help Topic button at the top-right of the window. See below:



1.4.3 Action Types

ActionStudio allows you to create three different kinds of actions, [Standard Actions](#), [Execute Program Actions](#) and [Iterator Actions](#).

Note: Once an action is created, its language and implementation type cannot be changed.

1.4.3.1 Standard Action

Standard actions are the default type of action. These can be used to automate anything (ActiveX automation objects, script logic, .NET Assemblies, etc.)

When a standard action executes in FinalBuilder a single method call to the action is performed (OnExecute.) The action does all of its work within that call, and finally returns success or failure.

1.4.3.2 Execute Program Action

The Execute Program action type has been designed to help create an action which calls an executable program. An Execute Program action supplies the executable filename and the command line, and then FinalBuilder handles calling the external program automatically.

By default, all console output from the external program is sent to the FinalBuilder log.

Script-Based Execute Program actions define the following script events:

- **GetExecutablePath** – this must return the fully qualified path to the executable.
- **GetStartDir** – returns the starting directory. If it is blank, then the directory of the executable is used (optional).
- **GetCommandLine** – returns the command line parameters to pass to the executable file. Use the "CommandLine" object to build the command line.
- **BeforeExecute** – allows the setting of how the output will be handled (optional).
- **OnInterpretResult** – the return code of the executable is passed in, and the function return is Boolean to signify success or failure (optional).
- **OnSuccess** – this function is called if the action succeeds (optional).
- **OnFailure** – this function is called if the action fails (optional).
- **OnStdOut** – called for each line of text sent to StdOut for custom processing (optional).
- **OnStdErr** – called for each line of text sent to StdErr for custom processing (optional).

.NET-based Execute Program actions define a similar set of methods which can be overridden by the user. See the [.NET Custom ActionApi reference](#) for details.

Note that you don't have to use Execute Program actions to run external programs. In particular, you could use the .NET System.Diagnostics.Process class to run an external program from a Standard Action type. However, Execute Program makes it easy to do this without custom code.

1.4.3.3 Iterator Action

The Iterator Action Type allows you to create an action which runs it child actions more than one. This is the same as Iterator actions in FinalBuilder

Iterator Actions automatically come with two properties:

- **Variable** - The variable to be set on each iteration. Setting the variable is done automatically, you don't need to do it inside the custom action.
- **FailIfNoIterations** - If this property is set to True and the iterator does not perform any iterations, the action will fail.

For script-based Iterator Actions, the following script events are available:

- **OnInitialiseIterator** - This event is called before the iterator starts running. The parameter "IteratorList" is a [FBStrings](#) object which can be automatically populated with the items to iterate over.
- **OnGetCurrentValue** - This event is called before each iteration, to get the value for that iteration. IteratorList is passed in as a parameter, as well as the number of the current iteration, and the default behaviour is to return the relevant item from IteratorList. You only need to modify this event in order to provide custom processing or output.

- **IsVariableCompulsory** - This event is called before the iterator runs to find out if the **Variable** property must be set to a valid variable. If the event returns True and **Variable** is not valid, the action will fail.

1.4.4 Loading Custom Actions In FinalBuilder

FinalBuilder requires that any custom action package file be located in the **FinalBuilder\ActionDefs** directory. The action package must be saved with a file extension of .fbap in the ActionDefs directory.

Any action implemented using .Net requires the assembly to be saved inside **FinalBuilder\ActionDefs\Assemblies** directory. For .NET Framework 2.0, 3.0 and 3.5, the directory to use is **FinalBuilder\ActionDefs\Assemblies\2.0**

When FinalBuilder loads, it scans the ActionDefs directory for any new fbap files and loads the action packages. FinalBuilder will automatically detect when ActionStudio saves changes to a custom action package, and will reload them (this option can be disabled in the ActionStudio options.)

Custom action package files are standard XML files which can be loaded in any XML viewer or editor.

1.4.5 Protected Action Packages

FinalBuilder Professional edition includes a license for ActionStudio Professional. ActionStudio Professional has the extra capability to import and export protected action packages. A protected action package is protected with a password so that others cannot view the action package source. Both FinalBuilder Standard and FinalBuilder Professional can automatically load up protected action packages (read-only) at runtime. To open a protected action package to edit in ActionStudio, you must enter the same password that was used to protect it.

2 Examples

2.1 Example Actions

FinalBuilder ships with many example actions, including full source code. It is recommended that you look at these action packages and source code to familiarize yourself with ActionStudio and custom actions.

Script and PowerShell Based

Package :
(%ProductName%> Directory)\ActionDefs\FBScriptExamples.fbap

.Net Based

Package :
(FinalBuilder Directory)\ActionDefs\FBDotNetExamples.fbap

Source code :
(%ProductName%> Directory)\ActionDefs\Examples\
DotNet\Dot Net Examples.sln

2.2 Creating Script Based Actions

Script based actions are entirely defined and coded within ActionStudio. They can be implemented in JavaScript, VBScript or [PowerShell](#).

ActionStudio includes an advanced code editor for editing scripts, which features syntax highlighting and code completion.

2.2.1 Creating a HelloWorld Example Action

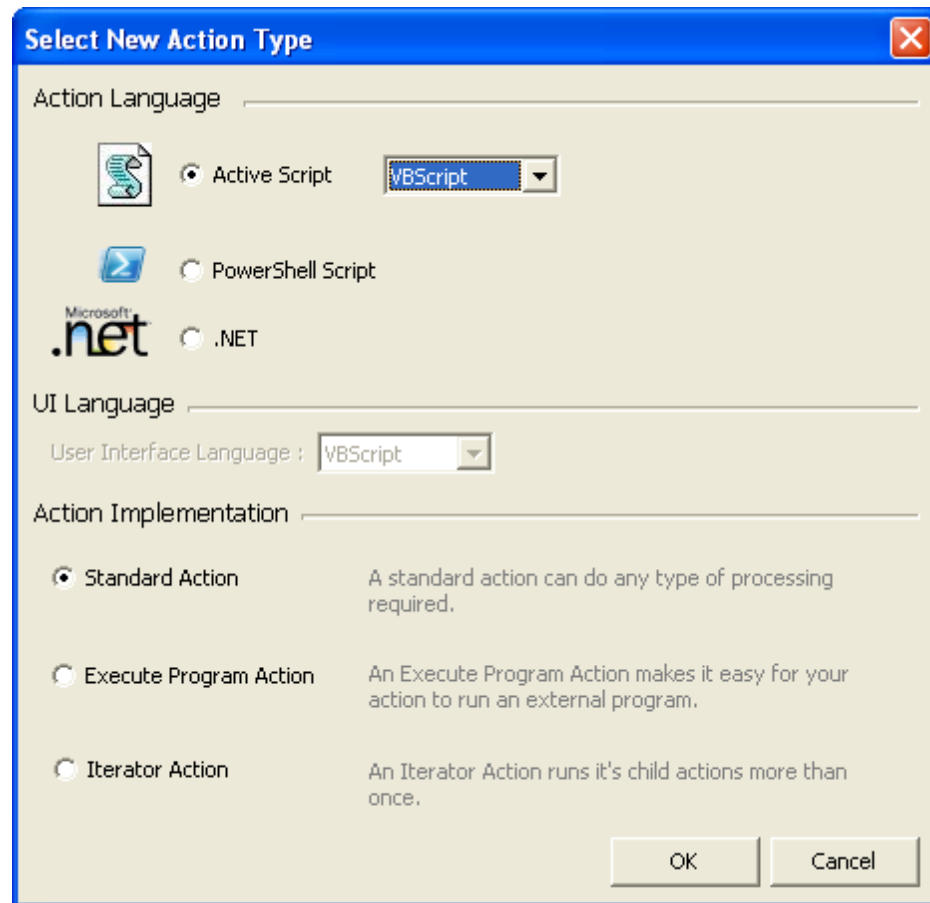
1. Launch ActionStudio

You can start ActionStudio either from the FinalBuilder program group (ie. Start->Programs->FinalBuilder->ActionStudio), or by selecting FinalBuilder ActionStudio from the Tools menu in FinalBuilder.

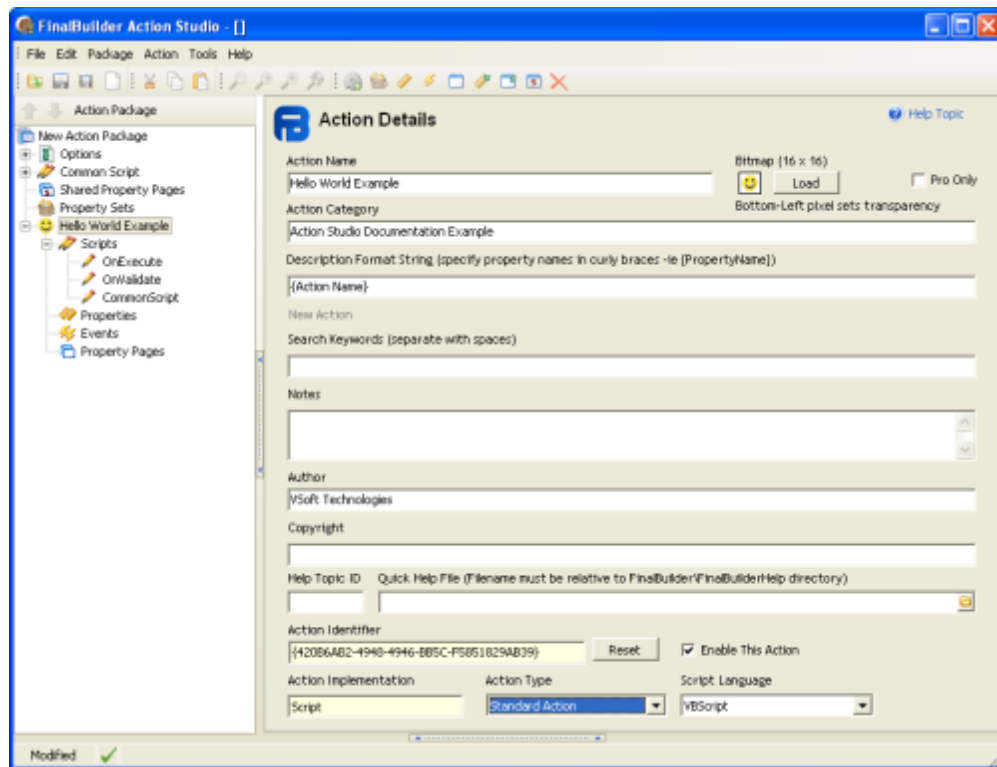
2. Select File -> New to start a new Action Package.

3. Choose "Add New Action" from the Package menu

The following dialog will appear, which allows you to choose the type of action to create. For this example, we will create an Active Script action using JScript as the script language.



3. Set the name of the action to "Hello World Example" (as shown), and optionally fill out any other fields.



Action Name – a unique name for the action. The name will appear in the Action Types in FinalBuilder

Action Category – the category in which the action will be grouped in.

Bitmap – Click "Load" to choose a 16x16 bitmap for the action. The bottom left pixel is used to set the transparency.

Notes – documentation for the action.

Author – the action author

Copyright – any copyright you want to add to the action

Action Identifier – this is a unique identifier so that FinalBuilder can match up action instances to the action implementations. You should only reset the action identifier if you have copied an action package and want to create new actions using the copied ones as a base.

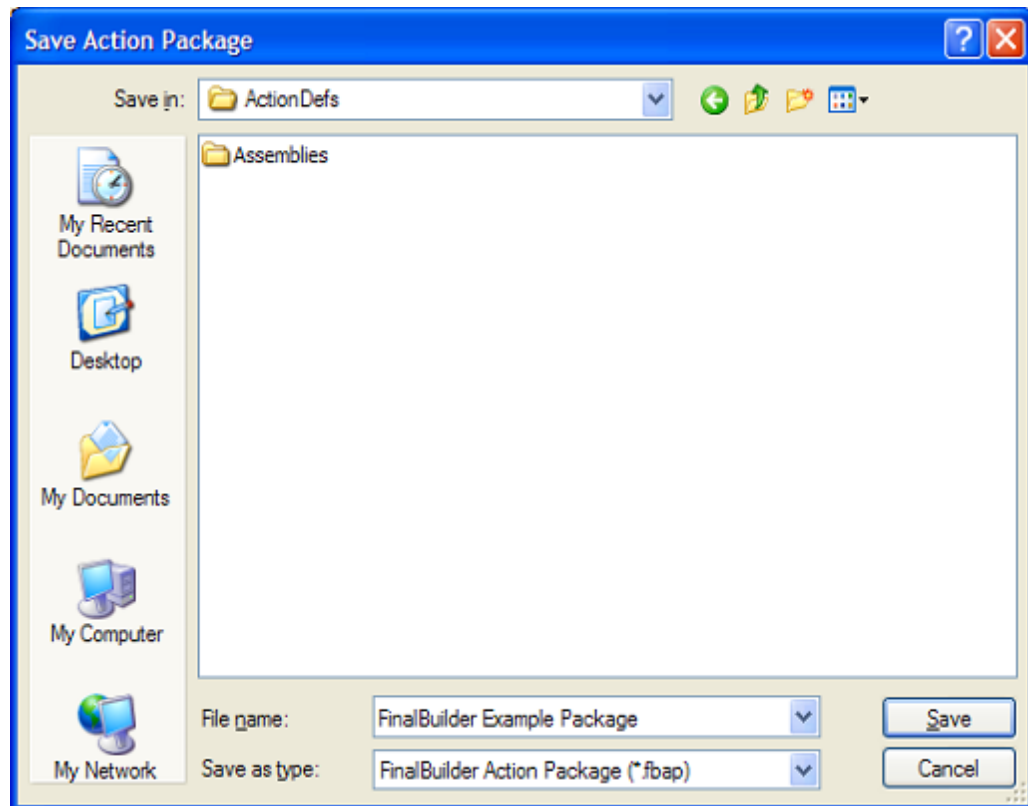
Action Implementation – shows that this action is being implemented in ActiveScript

Action Type – Standard action (see [Action Types](#) for more information)

Script Language – VBScript (all script for the action must be implemented in the one language)

4. Save the Action Package

It is very important that the action package is saved in the FinalBuilder\ActionDefs directory with the file extension .fbap, otherwise FinalBuilder won't load up the action package.



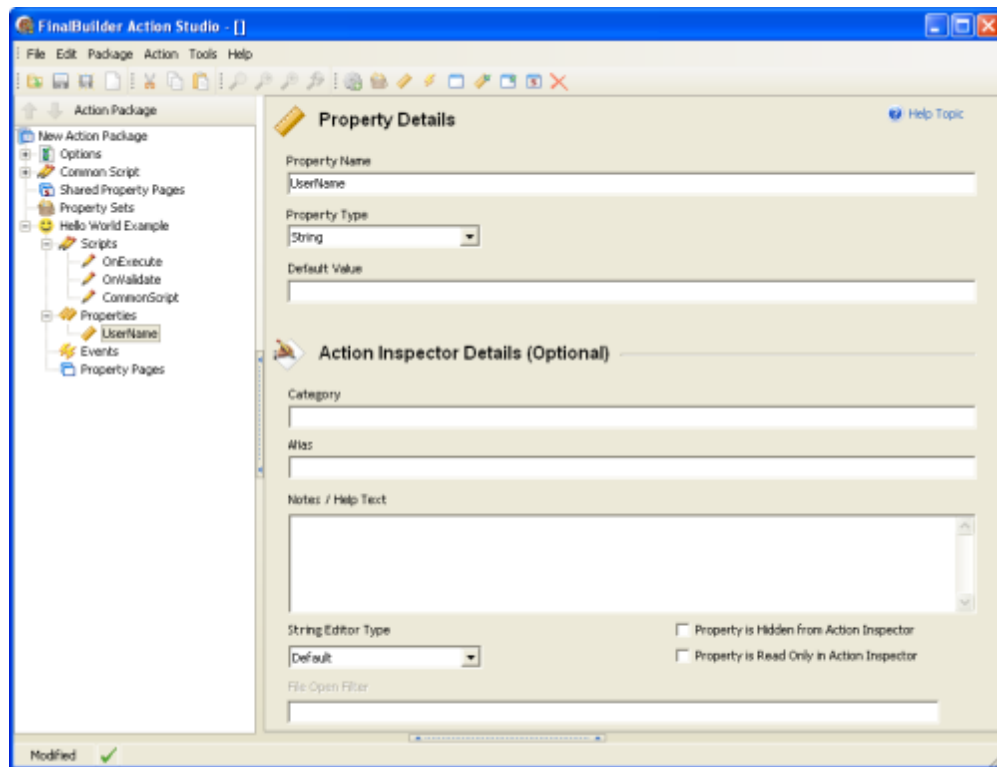
The package you just saved will load in FinalBuilder, but it won't do anything useful. For the next step, we will add a property called "User Name", add a property page which allows the FinalBuilder user to edit the "UserName" property, and when the action executes it will log a message that says "Hello <user name>".

5. Add a new property to the Hello World action

Right-Click on the Properties node and select "Add New Property"

You could also click the "Add New Property" toolbar button, or select it from the Action Menu.

6. Change the property name to "UserName"



Property Name – Enter the property name "UserName". Property names cannot contain spaces or non-alphanumeric characters

Property Type – Select the String type of property. Various other types are available.

The following fields can also be used to configure the property. You do not need to set them now, but feel free to do so if you want:

Default Value – This can be used to set the property to a default when the action is first added to a project.

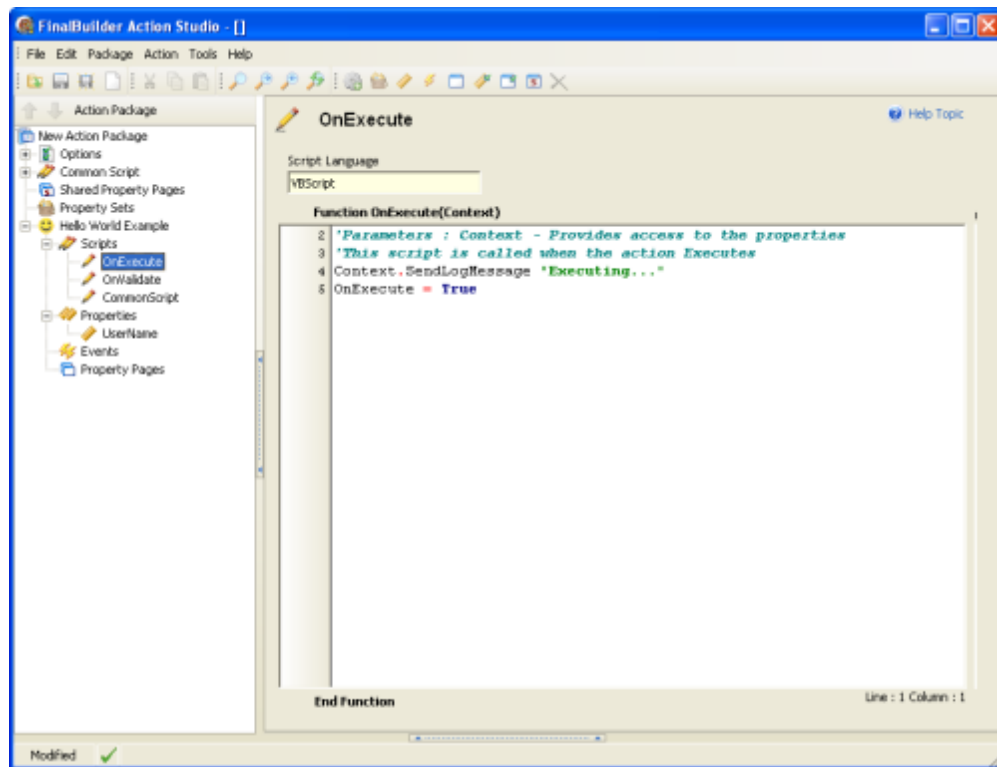
Category - The property will turn up under this category in the Action Inspector list in FinalBuilder. If left blank, this defaults to "Other".

Alias - The property will have this name in the Action Inspector list in FinalBuilder. The alias is also used when .NET-based custom actions need to show a custom error message.

Notes – write any documentation that will help you remember what the property is for or what it does. This information shows up in the Action Inspector as the "help" text.

The other properties all relate to how the property will display in the Action Inspector. You can ignore them for now.

7. Select the "OnExecute" script node from the package tree



When the action is executed in FinalBuilder, the OnExecute function is called. This is where you implement the functionality of your action. Return True or False depending on the success or failure of the action.

The other script event, OnValidate, can be used to add validation to the action. FinalBuilder automatically validates an action after it has been edited, and before the project starts running.

8. Change the OnExecute event to the following code

```
Context.SendLogMessage "Hello " & Context.Properties.PropertyAsString("UserName"),
stInformation
OnExecute = True
```

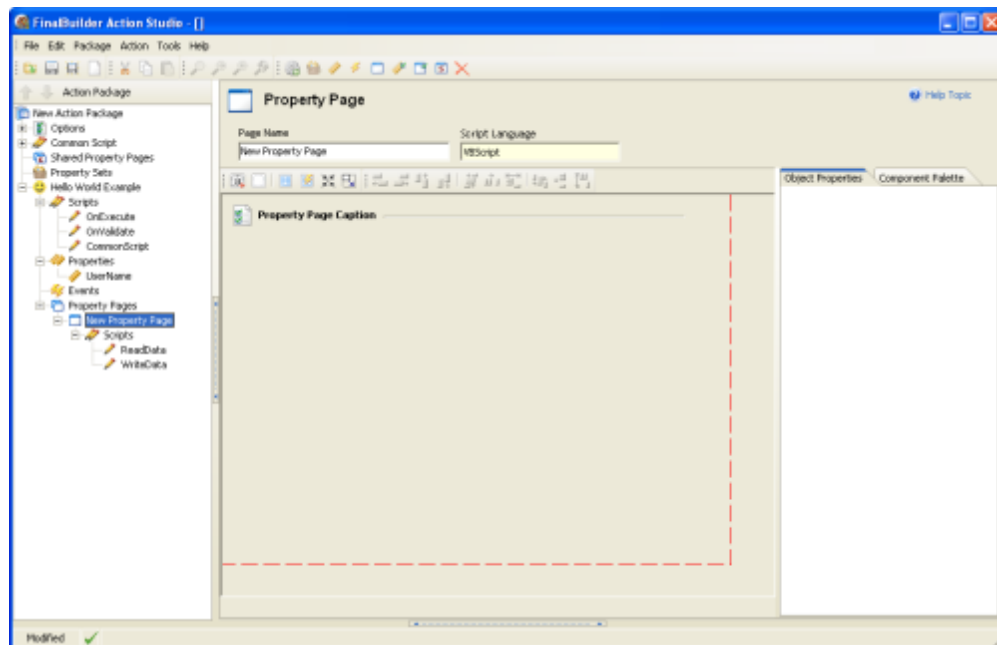
We send a Hello message with the user's name, as specified from the Property.

Notice that there is Code Completion to help you write Script by showing you the available types, function, procedures and parameters. Code completion is activated automatically when a dot is typed, or you can manually activate it by pressing Ctrl-Space.

Parameter help is also available, to activate it you need to press Ctrl-Shift-Space

9. Now add a property page

Select "Add New Property Page" by right clicking on the Property Pages node, or choose it from the Action menu.



Page Name – The name of the property page, this will display as the tab caption in the properties dialog in FinalBuilder.

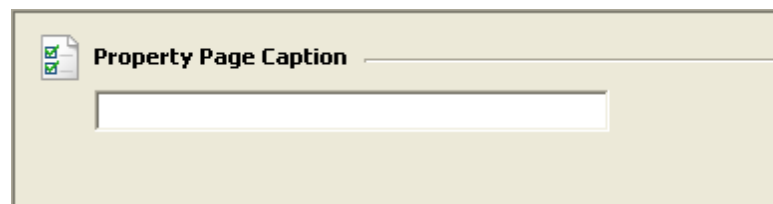
Design Surface – the design surface is where you design the property page. You need to place visual controls from the component palette, and customise them using the object inspector

Object Properties – this is an object inspector. When a visual control is selected in the design surface, the object inspector displays all the properties that you can customise

Component Palette – the component palette contains the available visual controls to design the property page. To add a component, drag it to the desired location.

10. Add a Label and a Text Edit to the design surface

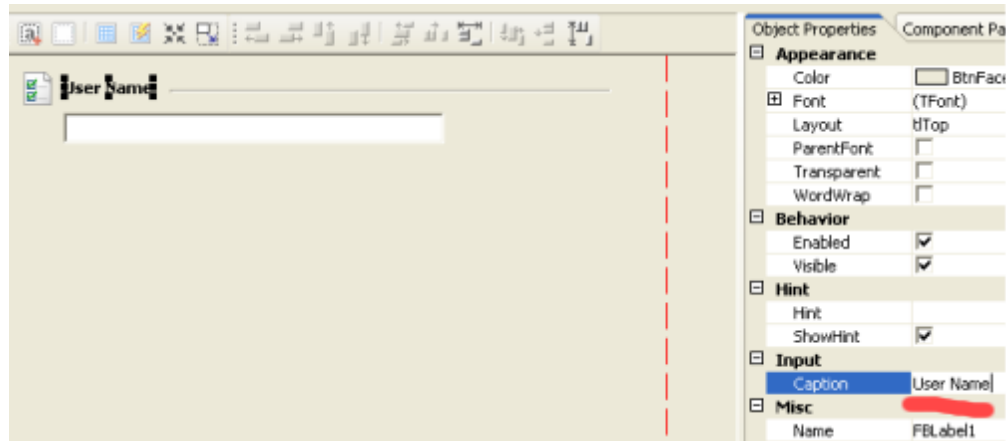
Select the Component Palette in the right-hand frame, and drag a "Text Edit" control onto the property page. Resize the text edit to make it a bit wider than the default. When you're done, it should look like this:



11. Modify the Label Caption

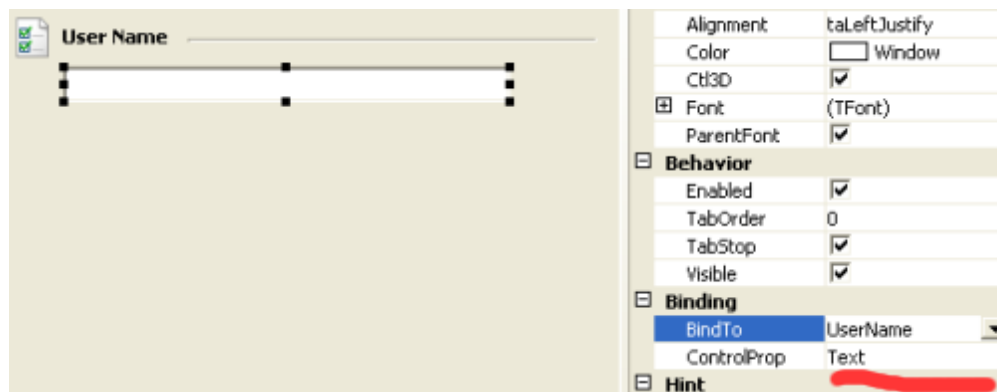
Click on "Property Page Caption" to select it, and then click on the "Object Inspector" tab to bring up the object properties. Change the caption property to "User Name".

If you're a perfectionist (like me), then you'll also drag the left-hand corner of the bevel so to it fits properly (as shown.)



12. Bind the TextEdit control to the UserName property

Select the TextEdit control in the design surface, and change the BindTo property (in the object properties) to the UserName property. The BindTo will only display the available properties of the action.



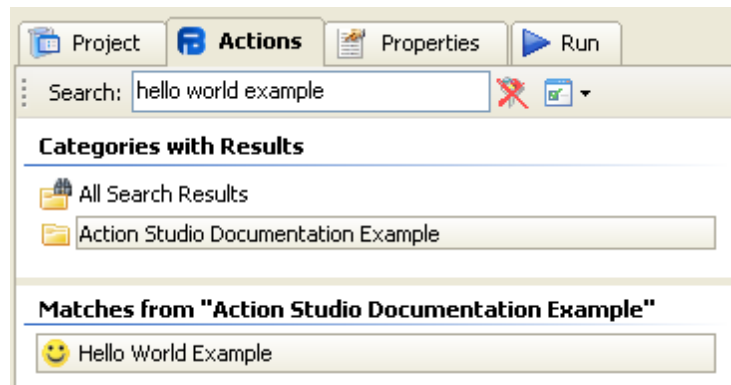
With a binding set, FinalBuilder will automatically load and save the selected property to and from the control when the property page is edited. The property specified by ControlProp (in this case, Text) is used for the binding value.

If you want to use script to load and save properties, then you can use the ReadData and WriteData script events of the property page.

13. Save the action package and start FinalBuilder

Save the action package, then launch FinalBuilder (or restart it if it was running.) Find your new action in the Actions list (the search will help.)

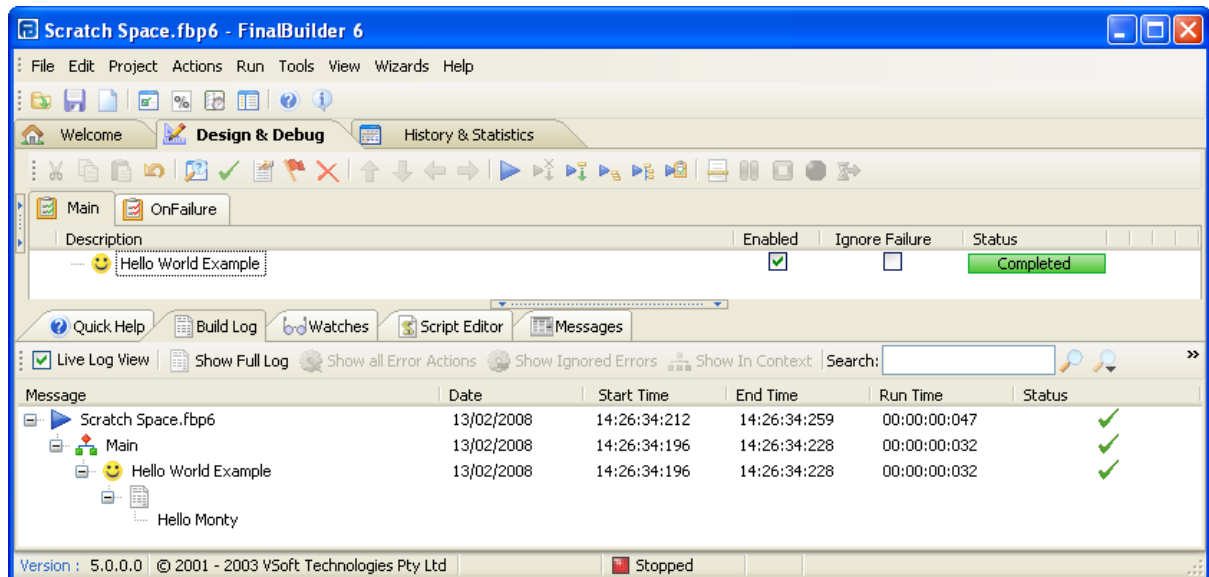
You should see the new action as part of the category that you specified back in Step 3 (or (No Category) if no category was supplied.)



If you can't see your action in the list, make sure that the package is saved in the ActionDefs folder inside the FinalBuilder program directory and that you restarted FinalBuilder so it could automatically pick up the new package.

14. Add the Hello World action to your Project

Click on the action to add it to your project. Edit the property page to set the username. Then run the action to see it do its stuff!

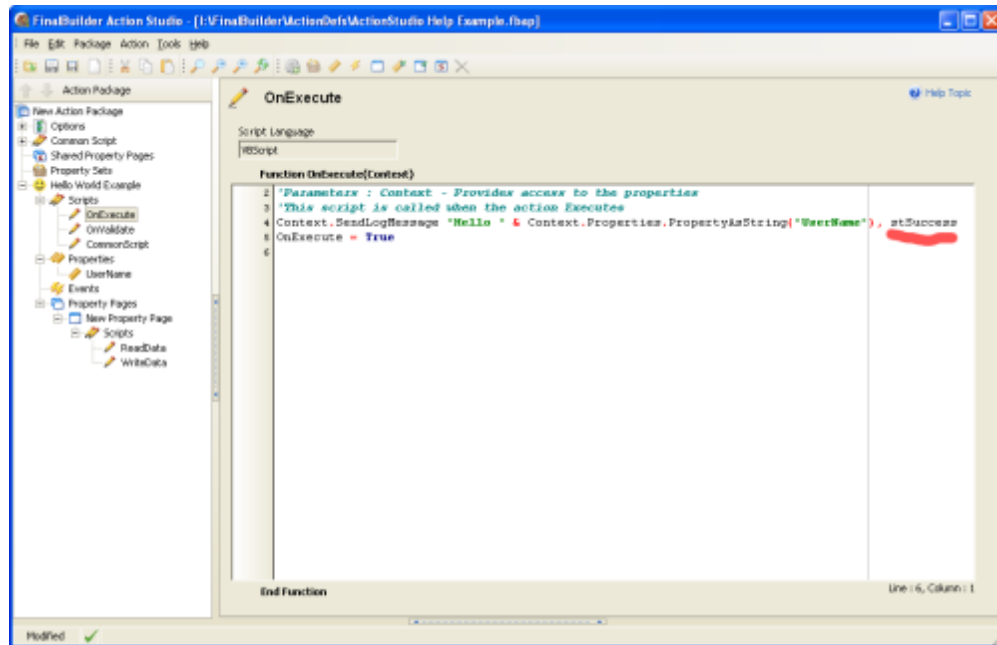


Congratulations - you have a working custom action. But the "Hello <Username>" text is pretty bland, let's fix it up.

15. Change the log message tag

Don't bother to close FinalBuilder, just switch back to your running copy of ActionStudio.

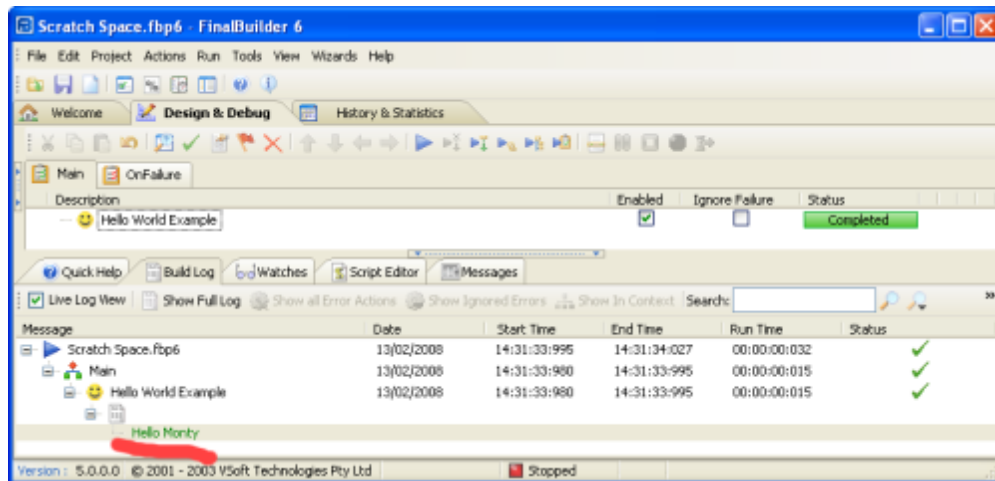
Click on the OnExecute event, and change the "stInformation" tag to "stSuccess".



Save the action package and switch back immediately to FinalBuilder.

16. Run the Project Again

The change you made will be automatically picked up, without needing to reload anything:



You're done. For more information, see the [Example Actions which are included with FinalBuilder](#)

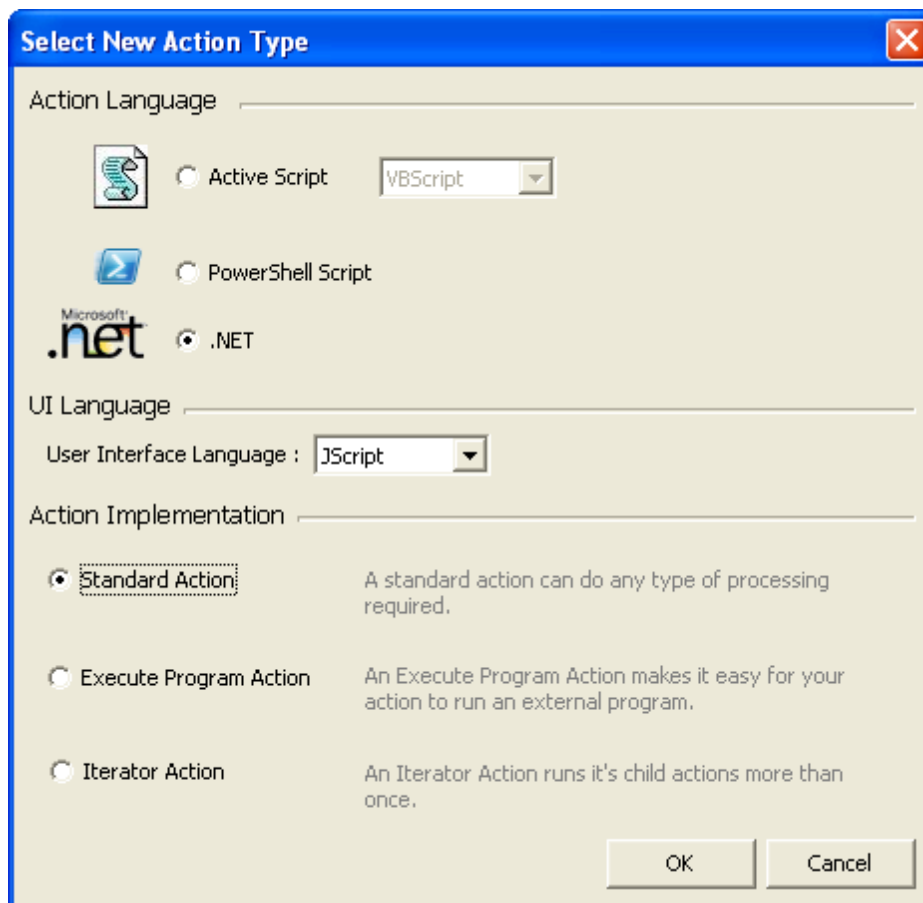
2.3 Creating .Net Based Actions

Microsoft .NET based actions are developed in three stages.

1. The actions, properties, events, and property pages are created in ActionStudio. This is exactly the same as creating a script-based action, as we saw in our [HelloWorld example](#).
2. The action's execution code is authored in a .NET Class, contained in a .NET Assembly. You can create the Assembly using any .NET authoring tool. The assembly must be placed inside the ActionDefs\Assemblies\<<version> directory.
3. The action definition is updated in Action Studio to refer to the assembly and the class name.

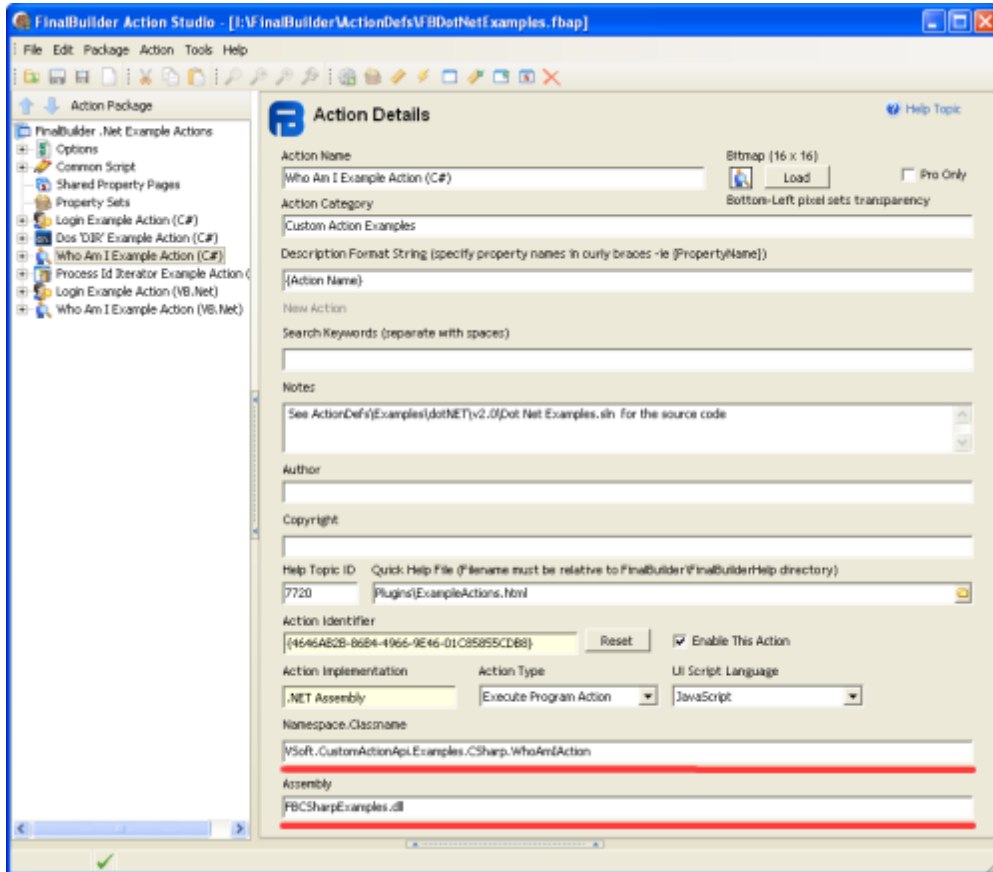
2.3.1 Define a .Net Action In ActionStudio

1. Create a new action (Package menu, Add New action).
2. Select the .NET option when the New Action Type dialog appears



Note that we also need to choose a User Interface script language for the action. This will be used for scripts which are attached to the property pages, created in ActionStudio.

3. Define the properties, property pages, etc. for your action



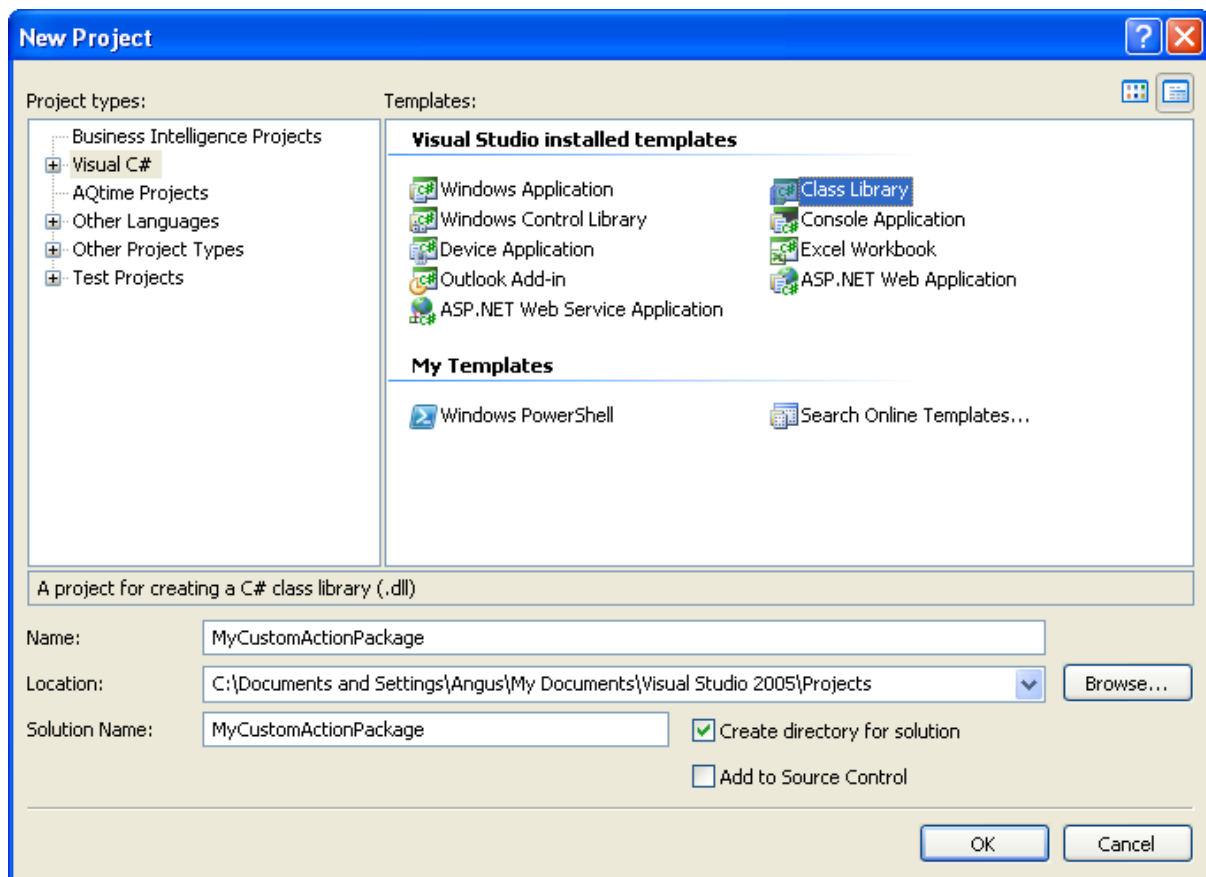
This is the same process as for script-based actions, as shown in the [HelloWorld](#) Example.

The only difference is two new properties: "Namespace.Classname" and "Assembly". You may not know the values for these properties until after you have created the assembly (see [Referencing the Action Class.](#))

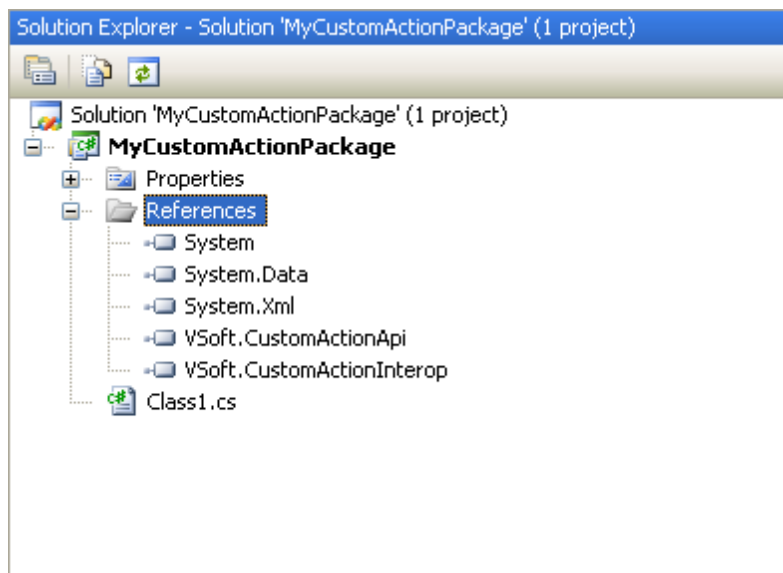
2.3.2 Creating A .Net Action Class

This example shows how to create the action as a C# Class in Microsoft Visual Studio. However, the action implementation can be created in any .NET language using any .NET tool.

1. Open Visual Studio and Create a new C# class library project.



2. In the new project, right-click on References and choose "Add Reference..." to add references to two assemblies, both located in the <FinalBuilder>\ActionDefs\Assemblies\2.0 directory. The assemblies are named VSoft.CustomActionApi.dll and VSoft.CustomActionInterop.dll.



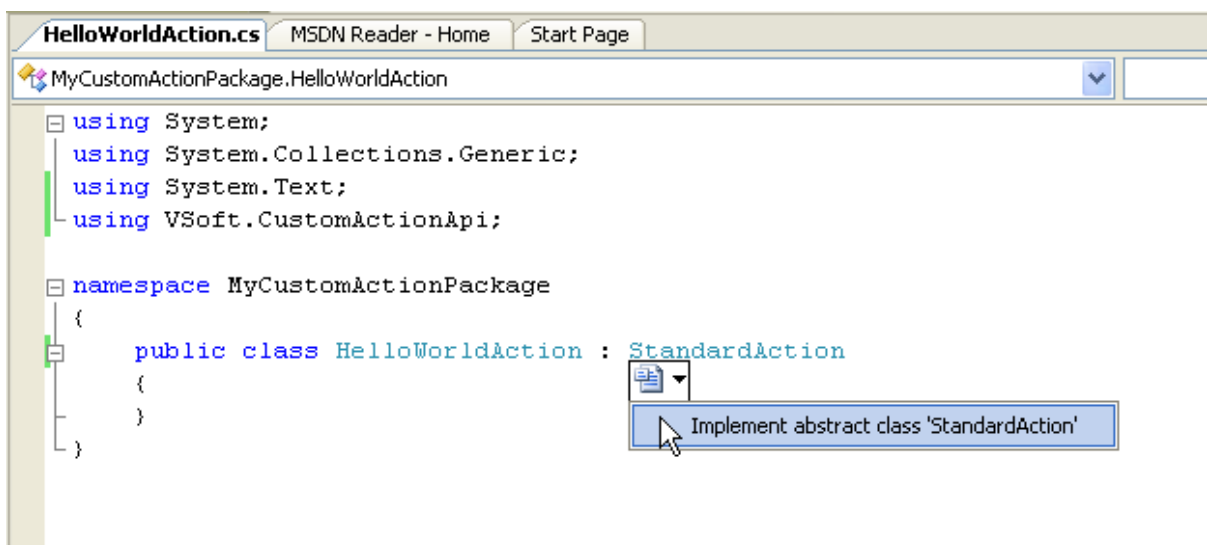
3. Rename Class1.cs to a more informative name (say, HelloWorldAction.cs)

4. Open HelloWorldAction.cs, and add the following "using" clause:

```
using VSoft.CustomActionApi;
```

5. Change the new action class so it inherits from StandardAction.

When you're done typing StandardAction, press Control-Period and select "Implement abstract class..."



6. The class now looks like this:

```
namespace MyCustomActionPackage
{
    public class HelloWorldAction : StandardAction
    {
        public override bool Execute()
        {
            throw new Exception("The method or operation is not implemented.");
        }

        public override void Validate()
        {
            throw new Exception("The method or operation is not implemented.");
        }
    }
}
```

Implementing these two methods is all you need to do to create a new action class. The Execute() method is called when the action is executing, and the Validate() method is

called when the action is validating before running.

The Execute method returns True or False to indicate success or failure. The Validate method will fail if the method SendValidationError() is called.

For this example, we will recreate the Hello World example from the Script example, but add validation that the "UserName" property is not empty before running.

7. Modify the class so it looks like this:

```
using System;
using System.Collections.Generic;
using System.Text;
using VSoft.CustomActionApi;

namespace MyCustomActionPackage
{
    public class HelloWorldAction : StandardAction
    {
        public override bool Execute()
        {
            SendMessage("Hello {0}", GetString("Username"));
            return true;
        }

        public override void Validate()
        {
            ValidateNonEmptyProperty("Username");
        }
    }
}
```

There are several things to notice here:

- The code makes heavy use of "helper" functions inherited from the base *StandardAction* class:

 GetString() returns the current value of a property, as a string. The property must be defined in ActionStudio first.

 SendMessage() sends a message to the log, and optionally supports the String.Format() syntax of the .NET Framework.

 ValidateNonEmptyProperty() checks if a named property's value is empty, and sends the appropriate validation failure if it is.

 All of the helper functions, and much more, are documented in the [Custom Action API Reference](#).

- It is still possible to use the same method syntax as used in script actions (ie Context.

Properties.get_PropertyAsString()), although we recommend that you use the helper methods (they are much simpler.)

- Helpful information is displayed whenever Visual Studio IntelliSense appears for the helper methods and parameters.

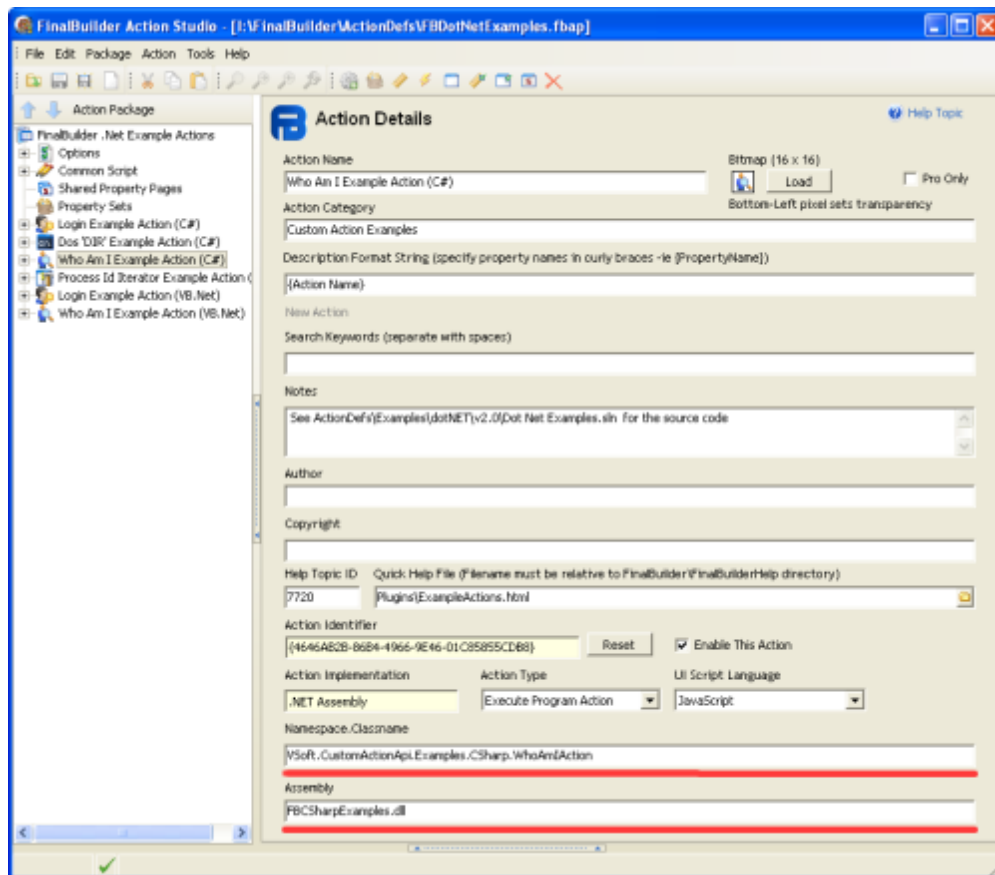
8. Build the assembly, and copy it to the ActionDefs\Assemblies\<>ver> folder, inside the FinalBuilder program folder.

You can copy the assembly by hand, or by changing the output directory for the Visual Studio project.

For more examples of actions coded in .NET (C# and VB.Net), see the [Dot Net Examples solution file](#).

2.3.3 Referencing the Action Class

After you have created a .NET Assembly with the action class then you will need to set the class name and the assembly name in ActionStudio.



"Namespace.ClassName" is the fully qualified name of the action class (ie MyCompany.MyNamespace.HelloWorldAction)

"Assembly" is the name of the assembly file, which must be located inside the ActionDefs\Assemblies\<>ver> directory.

2.3.4 Calling .NET Methods at Design Time

Custom actions implemented in .NET still require all of their design time support (action property pages) to be implemented in ActionStudio with JavaScript or VBScript.

However, it is possible to call .NET methods at design time. The .NET method to call must be a public instance method defined on the class for the action. The method must take exactly one parameter, and return a result. Both parameter and result need to be a type which is supported via COM Interop. This can include COM visible .NET object, which can be accessed via late binding from script.

The syntax for calling such a method is

Action.InvokeAssemblyMethod(methodName, methodArg)

Action is a parameter passed to property page events.
methodName is a string parameter with the name of the method to call.

Example

The example action named *Process Id Iterator Example Action (C#)* (defined in the [.NET Example Action Package](#)) contains an example of this technique.

2.3.5 .NET Custom Action API Reference

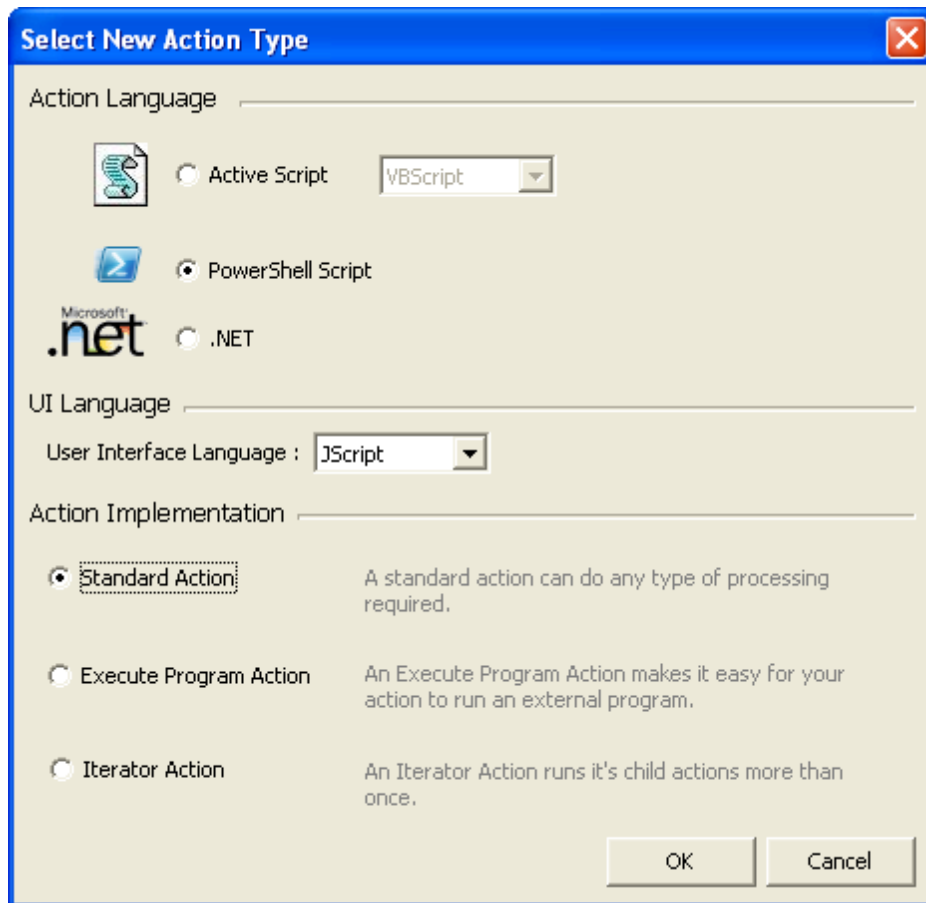
The VSoft.CustomActionApi namespace contains classes and helper methods designed to make it very easy to create .NET-based custom actions. You can find information about these classes in various ways:

- In Visual Studio, code completion will automatically provide detailed information about parameters, methods, and types.
- The Help file "[VSoft.CustomActionApi.chm](#)" provides a complete reference to the VSoft.CustomActionApi namespace.
- The [Dot NET Example Action solution file](#) contains working demonstration classes written in C# and VB.Net.

2.4 Creating PowerShell Based Actions

PowerShell actions are similar to other script-based actions, except that the action's runtime behaviour is implemented in [Microsoft PowerShell](#).

To create a new PowerShell action, choose Add New Action from the Package menu and select PowerShell for the action language:



Choose "PowerShell Script" for the Action Language.

PowerShell can only be used as the action language at runtime, so you should also choose a script language (VBScript or JScript) for any user interface design-time code.

Implementing PowerShell based actions is functionally identical to the example shown in [Creating a HelloWorld Example Action](#), except using the PowerShell script syntax.

Indicate success or failure by setting the `$Result` parameter to `$True` or `$False`.

For fully-fledged examples of PowerShell-based custom actions, look in the `FBScriptExamples.fbap` custom action package.

3 Reference

3.1 Available Objects

The following objects are available from the various methods of actions developed using ActionStudio:

[Action Object \(ActionInfo\)](#)

[ActionEvents Object \(CustomActionEvents\)](#)

[Context Object \(ActionExecuteContext\)](#)

[Context Object \(ActionValidateContext\)](#)

[CustomActionEvent Object](#)

[Page Object](#)

[Properties Object \(PropertyStore\)](#)

[FBProperty Object](#)

3.1.1 Action Object

Object: Action (type ActionInfo)

Available from:

Property Page, ReadData Script
Property Page, WriteData Script
Execute Context Object
Validate Context Object

Description:

The ActionInfo object represents the Action. It allows you to set or query the basic properties of an action.

Properties:

Enabled : Boolean
IgnoreFailure : Boolean
Description : String

Methods:

None

Example:

```
Action.Description = "MyAction [" & Properties.PropertyAsString("Database") & "]"
```

3.1.2 ActionEvents Object

Object: ActionEvents (type CustomActionEvents)

Available from:

Context Object (ActionExecuteContext)

Description:

This object holds the collection of Events for the Action.

Properties:

Count : Integer

ActionEvents(index : Integer) : CustomActionEvent

ActionEventsByName(Name : string) : CustomActionEvent

Methods:

None

Example:

```
if Context.ActionEvents.Count > 0 then
  ' do something
end if
```

3.1.3 Execute Context Object

Object: ActionExecuteContext

Available from:

OnExecute Method

GetExecutablePath Method

GetStartDir Method

GetCommandLine Method

OnInterpretResult Method

OnSuccess Method

OnFailure Method

OnStdOut Method

Description:

This object is the link between the FinalBuilder execution engine and the Action when it's executing. It provides a way for the Action to interface with logging engine, variables, etc. The object is not available during validation or when reading/writing property pages.

Properties:

ActionEvents : CustomActionEvents

Action : ActionInfo

Properties : PropertyStore

Methods:

SendLogMessage(msg : String)

SendProgressMessage(msg : String; progress : Integer)

ExecuteEvent(Name : String) : boolean

ExpandExpression(Expression : String; RaiseExceptions : boolean) : string

TryExpandExpression(PropertyName : string; Expression : string; FailIfBlank : boolean;

NewValue : string) : boolean
SetVariable(Name : string; Variable : OleVariant) : boolean
GetVariablesList(UpdateableOnly : boolean) : FBVariablesList
GetOptionsObject(Category : string; Name : string) : FBOptionsObject

Example:

Context.SendLogMessage "Executing Action..."

3.1.4 Validation Context Object

Object: ActionValidateContext

Available from:

OnValidate Method

Description:

This object is the link between the FinalBuilder execution engine and the Action when it's validating. It provides a way for the Action to interface with logging engine, variables, etc. The object is not available during execution or when reading/writing property pages.

Properties:

Action : ActionInfo
Properties : PropertyStore

Methods:

SendValidationErrorMessage(PropertyName : String; msg : String)
GetVariablesList(UpdateableOnly : boolean) : FBVariablesList

Example:

Context.SendValidationErrorMessage "MyProperty", "MyProperty has not been set"

3.1.5 CustomActionEvent Object

Object: CustomActionEvent

Available from:

ActionEvents Object

Description:

This object represents a single Event of the Action.
To execute the event, use the **ExecuteEvent** method of the [ActionExecuteContext](#) object

Properties:

EventName : String
ParameterCount : Integer

Methods:

ParameterNames(index : Integer) : String
ParameterValues(Name : String) : OleVariant

3.1.6 Page Object

Object: Page

Available from:

Option Page, ReadData Script
Option Page, WriteData Script
Property Page, ReadData Script
Property Page, WriteData Script

The Page object can also be accessed via the **GetPage** method. The GetPage method can be called from the ReadData, WriteData scripts as well as from control events (eg. The OnClick method of a button). The syntax for GetPage is: `GetPage(<pageindex>)`.

Description:

This object represents the option or property page and allows you to access the UI objects which you have placed on the page.

Properties:

Each UI element can be accessed as a separate property

Methods:

None

Example:

```
Page.FBLabel1.Caption = "New Text"
```

```
GetPage(1).MyLabel.Caption = "Label set from another page"
```

3.1.7 Properties Object

Object: Properties (type PropertyStore)

Available from:

Option Page, ReadData Script
Option Page, WriteData Script
Property Page, ReadData Script
Property Page, WriteData Script

Description:

This object holds the properties for the Action, allowing you to read and write the property values to/from the UI controls on the option and property pages

Properties:

```
PropertyAsString(Name : String) : String  
PropertyAsInteger(Name : String) : Integer  
PropertyAsBoolean(Name : String) : Boolean  
PropertyAsDouble(Name : String) : Double  
PropertyAsVariant(Name : String) : Variant  
PropertyAsStrings(Name : String) : FBStrings  
Properties(index : Integer) : FBProperty  
PropertyByName(Name : String) : FBProperty
```

Methods:

None

Example:

Properties.PropertyAsString("MyProp") = "New Value"

3.1.8 FBProperty Object

Object: FBProperty (type FBProperty)

Available from:

PropertyStore Object

Description:

This object represents a single property of an action

Properties:

Name : String

PropertyType : Integer

AsString : String

AsInteger : Integer

AsBoolean : Boolean

AsVariant : OleVariant

AsStringsObject : FBStrings

AsDouble : Double

Methods:

None

Example:

Properties.PropertiesByName("MyProperty").AsString = "New Value"

3.1.9 CommandLine Object

Object: CommandLine

Available from:

GetCommandLine Method

BeforeExecute Method

Description:

The CommandLine object enables you to build up a command line which is used to call an executable with.

Properties:

AsString : string

DefNameValueSeparator : string

DefArgSeparator : string

Count : integer

Arguments(index : Integer) : FBArgument

ArgumentsByName(index : String) : FBArgument

Methods:

AddArgument(Name : string; Value : String; ValueQuoted : TQuoteType) : FBArgument
(*)

```
RemoveArgument(Arg : FBArgument)
RemoveArgumentByName(Arg : String)
Clear
InsertArgumentBefore(newArg : FBArgument, beforeArg : FBArgument) : boolean
InsertArgumentBeforeName(newArg : FBArgument, beforeArg : string) : boolean
InsertArgumentAfter(newArg : FBArgument, afterArg : FBArgument) : boolean
InsertArgumentAfterName(newArg : FBArgument, afterArg : string) : boolean
```

Example:

```
CommandLine.AddArgument "-p", "c:\temp", qtAutoDouble
```

Notes

(*) In FinalBuilder versions prior to 5, adding an argument with the same (non-empty) "Name" parameter as a previous argument would result in only the second argument being kept, ie:

```
CommandLine.AddArgument("-f", "ArgumentA", qtAutoDouble);
CommandLine.AddArgument("-f", "ArgumentB", qtAutoDouble);
```

Would have resulted in (CommandLine.AsString == "-f ArgumentB"). The workaround was to add all of the information in the Value parameter, and leave Name blank.

In FinalBuilder 5, this behaviour has been changed and all arguments are kept. The above code now results in "-f ArgumentA -f ArgumentB".

3.1.10 FBStrings Objects

Description:

The FBStrings type provides access to a list of strings, similar to the TStrings type in the Borland VCL. This type is used for the return value of Properties.PropertyAsStrings(), and is also used by [Iterator Actions](#) to set the list of values.

You can also use the "Names" and "Values" methods to add name/value pairs in the form Name=Value.

Properties:

Count : Integer - The number of strings in the list.

Methods:

Add(value : String)	- Add a string to the list
IndexOf(value : String)	- Returns the index of a string in the list, or -1 if not found
Delete(index : Integer)	- Deletes the string at the specified index.
Strings(index : Integer) : String	- Returns the string at the specified index.
Names(Index : Integer) : String	- Returns the name for the specified index.

Values(Name : String) : String - Returns the value for the specified name.
ValueFromIndex(Index : String) : String - Returns the value at the specified index.

3.2 Available Global Functions/Objects

3.2.1 VBScript

Method: Abs

Method Declaration: Abs(Value : OleVariant) : integer

Description: Returns the absolute value of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfcAbs.asp>

Method: Array

Method Declaration: Array(argList : OleVariant) : OleVariant

Description: Returns a variant containing an array.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfcArray.asp>

Method: Asc

Method Declaration: Asc(Value : String) : integer

Description: Returns the ANSI character code corresponding to the first letter in a string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfcAsc.asp>

Method: Atn

Method Declaration: Atn(number : Double) : double

Description: Returns the arctangent of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfcAtn.asp>

Method: CBool

Method Declaration: CBool(expression : OleVariant) : Boolean

Description: Returns an expression that has been converted to a Variant of subtype Boolean.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfcCBool.asp>

Method: CByte

Method Declaration: CByte(expression : OleVariant) : Byte

Description: Returns an expression that has been converted to a Variant of subtype Byte.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfcCByte.asp>

Method: CCur**Method Declaration:** CCur(expression : OleVariant) : Currency**Description:** Returns an expression that has been converted to a Variant of subtype Currency.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCCur.asp>**Method:** CDate**Method Declaration:** CDate(expression : OleVariant) : Date**Description:** Returns an expression that has been converted to a Variant of subtype Date.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCDate.asp>**Method:** CDb1**Method Declaration:** CDb1(expression : OleVariant) : Double**Description:** Returns an expression that has been converted to a Variant of subtype Double.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCDbl.asp>**Method:** Chr**Method Declaration:** Chr(charcode : Integer) : String**Description:** Returns the character associated with the specified ANSI character code.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctChr.asp>**Method:** CInt**Method Declaration:** CInt(expression : OleVariant) : Integer**Description:** Returns an expression that has been converted to a Variant of subtype Integer.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCInt.asp>**Method:** CLng**Method Declaration:** CLng(expression : OleVariant) : Integer**Description:** Returns an expression that has been converted to a Variant of subtype Long.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCLng.asp>**Method:** Cos**Method Declaration:** Cos(expression : OleVariant) : Double**Description:** Returns the cosine of an angle.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCos.asp>**Method:** CreateObject**Method Declaration:** CreateObject(progid : OleVariant) : VBActiveXObject**Description:** Creates and returns a reference to an Automation object.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/>

[vsfctCreateObject.asp](#)

Method: CSng

Method Declaration: CSng(expression : OleVariant) : Integer

Description: Returns an expression that has been converted to a Variant of subtype Single.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCSng.asp>

Method: CStr

Method Declaration: CStr(expression : OleVariant) : String

Description: Returns an expression that has been converted to a Variant of subtype String.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctCStr.asp>

Method: Date

Method Declaration: Date() : Date

Description: Returns the current system date.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctDate.asp>

Method: DateAdd

Method Declaration: DateAdd(interval : String, number : Integer, date : Date) : Date

Description: Returns a date to which a specified time interval has been added.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctDateAdd.asp>

Method: DateDiff

Method Declaration: DateDiff(interval : String, date1 : Date, date2 : Date, firstdayofweek : Integer, firstweekofyear : Integer) : Integer

Description: Returns the number of intervals between two dates.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctDateDiff.asp>

Method: DatePart

Method Declaration: DatePart(interval : String, date : Date, firstdayofweek : Integer, firstweekofyear : Integer) : Integer

Description: Returns the specified part of a given date.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctDatePart.asp>

Method: DateSerial

Method Declaration: DateSerial(year : Integer, month : Integer, day : Integer) : Date

Description: Returns a Variant of subtype Date for a specified year, month, and day.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctDateSerial.asp>

Method: DateValue

Method Declaration: DateValue(date : string) : Date

Description: Returns a Variant of subtype Date.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsftDateValue.asp>

Method: Day

Method Declaration: Day(date : OleVariant) : Integer

Description: Returns a whole number between 1 and 31, inclusive, representing the day of the month.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsftDay.asp>

Method: Escape

Method Declaration: Escape(charString : String) : String

Description: Encodes a string so it contains only ASCII characters.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsftEscape.asp>

Method: Eval

Method Declaration: Eval(expression : String) : Boolean

Description: Evaluates an expression and returns the result.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsftEval.asp>

Method: Exp

Method Declaration: Exp(Number : OleVariant) : Double

Description: Returns e (the base of natural logarithms) raised to a power.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsftExp.asp>

Method: Filter

Method Declaration: Filter(inputStrings : OleVariant, Value : String) : OleVariant

Description: Returns a zero-based array containing a subset of a string array based on a specified filter criteria.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsftFilter.asp>

Method: FormatCurrency

Method Declaration: FormatCurrency(expression : OleVariant, NumDigitsAfterDecimal : integer, IncludeLeadingDigit : integer, UseParensForNegativeNumbers : integer, GroupDigits : integer) : String

Description: Returns an expression formatted as a currency value using the currency symbol defined in the system control panel.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/>

[vsfctFormatCurrency.asp](#)

Method: FormatDateTime

Method Declaration: FormatDateTime(expression : OleVariant, NamedFormat : Integer) : String

Description: Returns an expression formatted as a date or time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctFormatDateTime.asp>

Method: FormatNumber

Method Declaration: FormatNumber(expression : OleVariant, NumDigitsAfterDecimal : integer, IncludeLeadingDigit : integer, UseParensForNegativeNumbers : integer, GroupDigits : integer) : String

Description: Returns an expression formatted as a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctFormatNumber.asp>

Method: FormatPercent

Method Declaration: FormatPercent(expression : OleVariant, NumDigitsAfterDecimal : integer, IncludeLeadingDigit : integer, UseParensForNegativeNumbers : integer, GroupDigits : integer) : String

Description: Returns an expression formatted as a percentage (multiplied by 100) with a trailing % character.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctFormatPercent.asp>

Method: GetLocale

Method Declaration: GetLocale() : Integer

Description: Returns the current locale ID value.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctGetLocale.asp>

Method: Hex

Method Declaration: Hex(number : OleVariant) : String

Description: Returns a string representing the hexadecimal value of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctHex.asp>

Method: Hour

Method Declaration: Hour(time : OleVariant) : Integer

Description: Returns a whole number between 0 and 23, inclusive, representing the hour of the day.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctHour.asp>

Method: InputBox

Method Declaration: InputBox(prompt : String, title : String, default : String, xpos : Integer, ypos :

Integer, helpfile : String, context : Integer) : String

Description: Displays a prompt in a dialog box, waits for the user to input text or click a button, and returns the contents of the text box.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctInputDialog.asp>

Method: InStr

Method Declaration: InStr(start : Integer, string1 : String, string2 : String, compare : Integer) : Integer

Description: Returns the position of the first occurrence of one string within another.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctInStr.asp>

Method: InStrRev

Method Declaration: InStrRev(string1 : String, string2 : String, start : Integer, compare : Integer) : Integer

Description: Returns the position of an occurrence of one string within another, from the end of string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctInStrRev.asp>

Method: Int

Method Declaration: Int(number : OleVariant) : Integer

Description: Returns the integer portion of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctInt.asp>

Method: Fix

Method Declaration: Fix(number : OleVariant) : Integer

Description: Returns the integer portion of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctInt.asp>

Method: IsArray

Method Declaration: IsArray(varname : OleVariant) : Boolean

Description: Returns a Boolean value indicating whether a variable is an array.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctIsArray.asp>

Method: IsDate

Method Declaration: IsDate(varname : OleVariant) : Boolean

Description: Returns a Boolean value indicating whether an expression can be converted to a date.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctIsDate.asp>

Method: IsEmpty

Method Declaration: IsEmpty(varname : OleVariant) : Boolean

Description: Returns a Boolean value indicating whether a variable has been initialized.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctIsEmpty.asp>

Method: IsNull

Method Declaration: IsNull(varname : OleVariant) : Boolean

Description: Returns a Boolean value that indicates whether an expression contains no valid data (Null).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctIsNull.asp>

Method: IsNumeric

Method Declaration: IsNumeric(varname : OleVariant) : Boolean

Description: Returns a Boolean value indicating whether an expression can be evaluated as a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctIsNumeric.asp>

Method: IsObject

Method Declaration: IsObject(varname : OleVariant) : Boolean

Description: Returns a Boolean value indicating whether an expression references a valid Automation object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctIsObject.asp>

Method: Join

Method Declaration: Join(list : OleVariant) : String

Description: Returns a string created by joining a number of substrings contained in an array.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctJoin.asp>

Method: LBound

Method Declaration: LBound(arrayname : OleVariant, dimension : Integer) : Integer

Description: Returns the smallest available subscript for the indicated dimension of an array.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLBound.asp>

Method: LCase

Method Declaration: LCase(string : string) : String

Description: Returns a string that has been converted to lowercase.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLCase.asp>

Method: Left

Method Declaration: Left(string : string, length : Integer) : String

Description: Returns a specified number of characters from the left side of a string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLeft.asp>

Method: Len

Method Declaration: Len(string : string) : Integer

Description: Returns the number of characters in a string or the number of bytes required to store a variable.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLen.asp>

Method: Log

Method Declaration: Log(number : OleVariant) : Double

Description: Returns the natural logarithm of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLog.asp>

Method: LTrim

Method Declaration: LTrim(string : string) : String

Description: Returns a copy of a string without leading spaces (LTrim), trailing spaces (RTrim), or both leading and trailing spaces (Trim).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLTrim.asp>

Method: Mid

Method Declaration: Mid(string : string, start : Integer, length : Integer) : String

Description: Returns a specified number of characters from a string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctMid.asp>

Method: Minute

Method Declaration: Minute(time : OleVariant) : Integer

Description: Returns a whole number between 0 and 59, inclusive, representing the minute of the hour.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctMinute.asp>

Method: Month

Method Declaration: Month(date : OleVariant) : Integer

Description: Returns a whole number between 1 and 12, inclusive, representing the month of the year.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctMonth.asp>

Method: MonthName

Method Declaration: MonthName(month : Integer, abbreviate : Boolean) : String

Description: Returns a string indicating the specified month.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctMonthName.asp>

Method: MsgBox

Method Declaration: MsgBox(prompt : String, buttons : Integer, title : String, helpfile : String, context : Integer) : Integer

Description: Displays a message in a dialog box, waits for the user to click a button, and returns a value indicating which button the user clicked.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctMsgBox.asp>

Method: Now

Method Declaration: Now() : Date

Description: Returns the current date and time according to the setting of your computer's system date and time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctNow.asp>

Method: Oct

Method Declaration: Oct(number : OleVariant) : String

Description: Returns a string representing the octal value of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctOct.asp>

Method: Replace

Method Declaration: Replace(expression : String, find : String, replacewith : String, start : Integer, count : Integer, compare : Integer) : String

Description: Returns a string in which a specified substring has been replaced with another substring a specified number of times.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctReplace.asp>

Method: RGB

Method Declaration: RGB(red : Integer, blue : Integer, green : Integer) : Integer

Description: Returns a whole number representing an RGB color value.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctRGB.asp>

Method: Right

Method Declaration: Right(string : string, length : Integer) : String

Description: Returns a specified number of characters from the right side of a string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctRight.asp>

Method: Rnd

Method Declaration: Rnd(number : OleVariant) : Integer

Description: Returns a random number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctRnd.asp>

Method: Round

Method Declaration: Round(expression : OleVariant, numdecimalplaces : Integer) : Double

Description: Returns a number rounded to a specified number of decimal places.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctRound.asp>

Method: RTrim

Method Declaration: RTrim(string : string) : String

Description: Returns a copy of a string without leading spaces (LTrim), trailing spaces (RTrim), or both leading and trailing spaces (Trim).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLTrim.asp>

Method: Second

Method Declaration: Second(time : OleVariant) : Integer

Description: Returns a whole number between 0 and 59, inclusive, representing the second of the minute.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctSecond.asp>

Method: SetLocale

Method Declaration: SetLocale(lcid : OleVariant) : Integer

Description: Sets the global locale and returns the previous locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctSetLocale.asp>

Method: Sgn

Method Declaration: Sgn(number : OleVariant) : Integer

Description: Returns an integer indicating the sign of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctSgn.asp>

Method: Sin

Method Declaration: Sin(number : OleVariant) : Double

Description: Returns the sine of an angle.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctSin.asp>

Method: Space

Method Declaration: Space(number : Integer) : String

Description: Returns a string consisting of the specified number of spaces.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctSpace.asp>

Method: Split

Method Declaration: Split(expression : String, delimiter : String, count : Integer, compare : Integer) : OleVariant

Description: Returns a zero-based, one-dimensional array containing a specified number of substrings.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctSplit.asp>

Method: Sqr

Method Declaration: Sqr(number : OleVariant) : Double

Description: Returns the square root of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctSqr.asp>

Method: StrComp

Method Declaration: StrComp(string1 : String, string2 : String, compare : Integer) : Integer

Description: Returns a value indicating the result of a string comparison.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctStrComp.asp>

Method: String

Method Declaration: String(number : Integer, character : OleVariant) : String

Description: Returns a repeating character string of the length specified.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctString.asp>

Method: StrReverse

Method Declaration: StrReverse(string1 : String) : String

Description: Returns a string in which the character order of a specified string is reversed.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctStrReverse.asp>

Method: Tan

Method Declaration: Tan(number : OleVariant) : Double

Description: Returns the tangent of an angle.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctTan.asp>

Method: Time

Method Declaration: Time() : Date

Description: Returns a Variant of subtype Date indicating the current system time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctTime.asp>

Method: Timer**Method Declaration:** Timer() : Integer**Description:** Returns the number of seconds that have elapsed since 12:00 AM (midnight).**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctTimer.asp>**Method:** TimeSerial**Method Declaration:** TimeSerial() : Date**Description:** Returns a Variant of subtype Date containing the time for a specific hour, minute, and second.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctTimeSerial.asp>**Method:** TimeValue**Method Declaration:** TimeValue() : Date**Description:** Returns a Variant of subtype Date containing the time.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctTimeValue.asp>**Method:** Trim**Method Declaration:** Trim(string : string) : String**Description:** Returns a copy of a string without leading spaces (LTrim), trailing spaces (RTrim), or both leading and trailing spaces (Trim).**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctLTrim.asp>**Method:** TypeName**Method Declaration:** TypeName(varname : OleVariant) : String**Description:** Returns a string that provides Variant subtype information about a variable.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctTypeName.asp>**Method:** UBound**Method Declaration:** UBound(arrayname : OleVariant, dimension : Integer) : Integer**Description:** Returns the largest available subscript for the indicated dimension of an array.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctUBound.asp>**Method:** UCase**Method Declaration:** UCase(string : string) : String**Description:** Returns a string that has been converted to uppercase.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctUCase.asp>**Method:** UnEscape

Method Declaration: UnEscape(charString : String) : String

Description: Decodes a string encoded with the Escape function.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctUnescape.asp>

Method: VarType

Method Declaration: VarType(varname : OleVariant) : Integer

Description: Returns a value indicating the subtype of a variable.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctVarType.asp>

Method: WeekDay

Method Declaration: WeekDay(date : OleVariant, firstdayofweek : Integer) : integer

Description: Returns a whole number representing the day of the week.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctWeekday.asp>

Method: WeekDayName

Method Declaration: WeekDayName(weekday : Integer, abbreviate : Boolean, firstdayofweek : Integer) : String

Description: Returns a string indicating the specified day of the week.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctWeekdayName.asp>

Method: Year

Method Declaration: Year(date : OleVariant) : integer

Description: Returns a whole number representing the year.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vsfctYear.asp>

3.2.2 JScript

The following objects are available for use with JScript:

- [Object](#)
- [Number](#)
- [Boolean](#)
- [Error](#)
- [String](#)
- [Array](#)
- [Date](#)
- [Enumerator](#)
- [Arguments](#)
- [Function](#)

- [ActiveXObject](#)
- [Debug](#)
- [Number](#)
- [Math](#)
- [RegExp](#)
- [RegExp](#)

3.2.2.1 JScript Object: Object

Object: Object

Description: Provides functionality common to all JScript objects.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobject.asp>

Object Properties

- `propertyIsEnumerable`

Object Methods

- `isPrototypeOf`
- `hasOwnProperty`
- `toLocaleString`

Property Name: `propertyIsEnumerable`

Property Type: String

Description: Returns a Boolean value indicating whether a specified property is part of an object and if it is enumerable.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jspropertyisenumerable.asp>

Method: `isPrototypeOf`

Method Declaration: `isPrototypeOf(object2 : JSObject) : boolean`

Description: Returns a Boolean value indicating whether an object exists in another object's prototype chain.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthisprototypeof.asp>

Method: `hasOwnProperty`

Method Declaration: `hasOwnProperty(propertyName : String) : boolean`

Description: Returns a Boolean value indicating whether an object has a property with the specified name.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthasownproperty.asp>

Method: `toLocaleString`

Method Declaration: `toLocaleString() : JSString`

Description: Returns a date converted to a string using the current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtolocalestring.asp>

3.2.2.2 JScript Object: Number

Object: Number

Description: An object representation of the number data type and placeholder for numeric constants.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjnumber.asp>

Object Properties

- MAX_VALUE
- MIN_VALUE
- NaN
- NEGATIVE_INFINITY
- POSITIVE_INFINITY

Object Methods

- toExponential
- toFixed
- toLocale
- toString
- valueOf

Property Name: MAX_VALUE

Property Type: Number

Description: Returns the largest number representable in JScript. Equal to approximately 1.79E+308.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jspromaxvalue.asp>

Property Name: MIN_VALUE

Property Type: Number

Description: Returns the number closest to zero representable in JScript. Equal to approximately 5.00E-324.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprominvalue.asp>

Property Name: NaN

Property Type: Number

Description: A special value that indicates an arithmetic expression returned a value that was not a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jspronannumber.asp>

Property Name: NEGATIVE_INFINITY

Property Type: Number

Description: Returns a value more negative than the largest negative number (-Number.MAX_VALUE) representable in JavaScript.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jspronegativeinf.asp>

Property Name: POSITIVE_INFINITY

Property Type: Number

Description: Returns a value larger than the largest number (Number.MAX_VALUE) that can be represented in JavaScript.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jspropositiveinf.asp>

Method: toExponential

Method Declaration: toExponential(fractionDigits : integer) : string

Description: Returns a string containing a number represented in exponential notation.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtoexponential.asp>

Method: toFixed

Method Declaration: toFixed(fractionDigits : integer) : string

Description: Returns a string representing a number in fixed-point notation.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtofixed.asp>

Method: toLocale

Method Declaration: toLocale() : string

Description: Returns a date converted to a string using the current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtolocalestring.asp>

Method: toString

Method Declaration: toString(radix : integer) : string

Description: Returns a string representation of an object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtostring.asp>

Method: valueOf

Method Declaration: valueOf() : Object

Description: Returns the primitive value of the specified object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthvalueof.asp>

3.2.2.3 JScript Object: Boolean

Object: Boolean

Description: Creates a new Boolean value.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjboolean.asp>

Object Methods

- toString
- valueOf

Method: toString

Method Declaration: toString() : String

Description: Returns a string representation of an object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtostring.asp>

Method: valueOf

Method Declaration: valueOf() : String

Description: Returns the primitive value of the specified object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthvalueof.asp>

3.2.2.4 JScript Object: Error

Object: Error

Description: Contains information about errors.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjerror.asp>

3.2.2.5 JScript Object: String

Object: String

Description: Allows manipulation and formatting of text strings and determination and location of substrings within strings.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjstring.asp>

Object Properties

- length

Object Methods

- anchor
- big
- blink
- bold
- charAt

- charCodeAt
- concat
- fixed
- fontcolor
- fontsize
- fromCharCode
- indexOf
- italics
- lastIndexOf
- link
- localeCompare
- match
- replace
- search
- slice
- small
- split
- strike
- sub
- split
- sup
- toLocaleLowerCase
- toLocaleUpperCase
- toLowerCase
- toUpperCase
- toString
- valueOf

Property Name: length

Property Type: Number

Description: Returns the length of a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolengthstring.asp>

Method: anchor

Method Declaration: anchor(anchorString : JSSString) : JSSString

Description: Places an HTML anchor with a NAME attribute around specified text in the object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthanchor.asp>

Method: big

Method Declaration: big() : JSSString

Description: Places HTML <BIG> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/>

js56jsmthbig.asp

Method: blink

Method Declaration: blink() : JSString

Description: Places HTML <BLINK> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthblink.asp>

Method: bold

Method Declaration: bold() : JSString

Description: Places HTML tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthbold.asp>

Method: charAt

Method Declaration: charAt(index : Number) : String

Description: Returns the character at the specified index.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthcharat.asp>

Method: charCodeAt

Method Declaration: charCodeAt(index : Number) : Number

Description: Returns an integer representing the Unicode encoding of the character at the specified location.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthcharCodeAt.asp>

Method: concat

Method Declaration: concat(string2 : JSString, stringn : JSString) : String

Description: Returns a string value containing the concatenation of two or more supplied strings.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthconcatstring.asp>

Method: fixed

Method Declaration: fixed() : JSString

Description: Places HTML <TT> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthfixed.asp>

Method: fontcolor

Method Declaration: fontcolor(colorVal : JSObject) : JSString

Description:

URL for more info:

Method: fontsize

Method Declaration: fontsize(intSize : Number) : JSString

Description: Places an HTML tag with the SIZE attribute around the text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthfontsize.asp>

Method: fromCharCode

Method Declaration: fromCharCode(Code1 : Number, CodeN : Number) : JSString

Description: Returns a string from a number of Unicode character values.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthfromCharCode.asp>

Method: indexOf

Method Declaration: indexOf(subString : String, startIndex : Number) : Number

Description: Returns the character position where the first occurrence of a substring occurs within a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthindexOf.asp>

Method: italics

Method Declaration: italics() : JSString

Description: Places HTML <I> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthitalics.asp>

Method: lastIndexOf

Method Declaration: lastIndexOf(subString : String, startIndex : Number) : Number

Description:

URL for more info:

Method: link

Method Declaration: link(linkString : String) : JSString

Description: Places an HTML anchor with an HREF attribute around the text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthlink.asp>

Method: localeCompare

Method Declaration: localeCompare(stringExp : String) : Number

Description: Returns a value indicating whether two strings are equivalent in the current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthLocaleCompare.asp>

Method: match

Method Declaration: match(rgExp : RegExp) : JSArray

Description: Executes a search on a string using a regular expression pattern, and returns an array containing the results of that search.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/>

js56jsmthmatch.asp

Method: replace

Method Declaration: replace(rgExp : RegExp, replaceText : String) : JSString

Description: Returns a copy of a string with text replaced using a regular expression or search string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthreplace.asp>

Method: search

Method Declaration: search(rgExp : RegExp) : Number

Description: Returns the position of the first substring match in a regular expression search.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsearch.asp>

Method: slice

Method Declaration: slice(start : Number, end : Number) : JSString

Description: Returns a section of a string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthslicestring.asp>

Method: small

Method Declaration: small() : JSString

Description: Places HTML <SMALL> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsmall.asp>

Method: split

Method Declaration: split(separator : String, limit : Number) : JSArray

Description: Returns the array of strings that results when a string is separated into substrings.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsplit.asp>

Method: strike

Method Declaration: strike() : JSString

Description: Places HTML <STRIKE> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthstrike.asp>

Method: sub

Method Declaration: sub() : JSString

Description: Places HTML <SUB> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsub.asp>

Method: split

Method Declaration: split(start : Number, length : Number) : JSString

Description: Returns the array of strings that results when a string is separated into substrings.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsplit.asp>

Method: sup

Method Declaration: sup() : JSString

Description: Places HTML <SUP> tags around text in a String object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsup.asp>

Method: toLocaleLowerCase

Method Declaration: toLocaleLowerCase() : JSString

Description: Returns a string where all alphabetic characters have been converted to lowercase, taking into account the host environment's current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthToLocaleLowerCase.asp>

Method: toLocaleUpperCase

Method Declaration: toLocaleUpperCase() : JSString

Description: Returns a string where all alphabetic characters have been converted to uppercase, taking into account the host environment's current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthToLocaleUpperCase.asp>

Method: toLowerCase

Method Declaration: toLowerCase() : JSString

Description: Returns a string where all alphabetic characters have been converted to lowercase.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtolowercase.asp>

Method: toUpperCase

Method Declaration: toUpperCase() : JSString

Description: Returns a string where all alphabetic characters have been converted to uppercase.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtouppercase.asp>

Method: toString

Method Declaration: toString() : String

Description: Returns a string representation of an object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtoString.asp>

Method: valueOf

Method Declaration: valueOf() : String

Description: Returns the primitive value of the specified object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthvalueOf.asp>

3.2.2.6 JScript Object: Array

Object: Array

Description: Provides support for creation of arrays of any data type.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjarray.asp>

Object Properties

- length

Object Methods

- concat
- join
- pop
- push
- reverse
- shift
- slice
- sort
- splice
- toString
- valueOf

Property Name: length

Property Type: Number

Description: Returns an integer value one higher than the highest element defined in an array.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolengtharray.asp>

Method: concat

Method Declaration: concat(item1 : JScriptObject, itemn : JScriptObject) : JScriptString

Description: Returns a new array consisting of a combination of two or more arrays.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthconcatarray.asp>

Method: join

Method Declaration: join(separator : JScriptObject) : JScriptString

Description: Returns a string value consisting of all the elements of an array concatenated together and separated by the specified separator character.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthjoin.asp>

Method: pop**Method Declaration:** pop() : JSObject**Description:** Removes the last element from an array and returns it.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthpop.asp>**Method:** push**Method Declaration:** push(item1 : JSObject, itemn : JSObject) : Number**Description:** Appends new elements to an array, and returns the new length of the array.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthpush.asp>**Method:** reverse**Method Declaration:** reverse() : JSArray**Description:** Returns an Array object with the elements reversed.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthreverse.asp>**Method:** shift**Method Declaration:** shift() : JSObject**Description:** Removes the first element from an array and returns it.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthshift.asp>**Method:** slice**Method Declaration:** slice(start : Number, end : Number) : JSArray**Description:** Returns a section of an array.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthslicearray.asp>**Method:** sort**Method Declaration:** sort(sortfunction : String) : JSArray**Description:** Returns an Array object with the elements sorted.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsort.asp>**Method:** splice**Method Declaration:** splice(start : Number, deletecount : Number, item1 : JSObject, itemn : JSObject) : JSArray**Description:** Removes elements from an array and, if necessary, inserts new elements in their place, returning the deleted elements.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsplice.asp>**Method:** toString**Method Declaration:** toString() : String

Description: Returns a string representation of an object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtostring.asp>

Method: valueOf

Method Declaration: valueOf() : String

Description: Returns the primitive value of the specified object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthvalueof.asp>

3.2.2.7 JScript Object: Date

Object: Date

Description: Enables basic storage and retrieval of dates and times.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjdate.asp>

Object Methods

- getDate
- getDay
- getFullYear
- getHours
- getMilliseconds
- getMinutes
- getMonth
- getSeconds
- getTime
- getTimezoneOffset
- getUTCDate
- getUTCDay
- getUTCFullYear
- getUTCHours
- getUTCMilliseconds
- getUTCMinutes
- getUTCMonth
- getUTCSeconds
- getVarDate
- getYear
- parse
- setDate
- setFullYear
- setHours
- setMilliseconds
- setMinutes

- setMonth
- setSeconds
- setTime
- setUTCDate
- setUTCFullYear
- setUTCHours
- setUTCMilliseconds
- setUTCMinutes
- setUTCMonth
- setUTCSeconds
- setYear
- toDateString
- toGMTString
- toLocaleDateString
- toLocaleString
- toLocaleTimeString
- toDateString
- toString
- toTimeString
- toUTCString
- valueOf
- UTC

Method: getDate

Method Declaration: getDate() : Number

Description: Returns the day of the month value in a Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetdate.asp>

Method: getDay

Method Declaration: getDay() : Number

Description: Returns the day of the week value in a Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetday.asp>

Method: getFullYear

Method Declaration: getFullYear() : Number

Description: Returns the year value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetfullyear.asp>

Method: getHours

Method Declaration: getHours() : Number

Description: Returns the hours value in a Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgethours.asp>

Method: getMilliseconds

Method Declaration: getMilliseconds() : Number

Description: Returns the milliseconds value in a Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetmilliseconds.asp>

Method: getMinutes

Method Declaration: getMinutes() : Number

Description: Returns the minutes value in a Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetminutes.asp>

Method: getMonth

Method Declaration: getMonth() : Number

Description: Returns the month value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetmonth.asp>

Method: getSeconds

Method Declaration: getSeconds() : Number

Description: Returns the seconds value in a Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetseconds.asp>

Method: getTime

Method Declaration: getTime() : Number

Description: Returns the time value in a Date object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgettime.asp>

Method: getTimezoneOffset

Method Declaration: getTimezoneOffset() : Number

Description: Returns the difference in minutes between the time on the host computer and Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgettimezone.asp>

Method: getUTCDate

Method Declaration: getUTCDate() : Number

Description: Returns the date in a Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutcdate.asp>

Method: getUTCDay

Method Declaration: getUTCDay() : Number

Description: Returns the day of the week value in a Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutcday.asp>

Method: getUTCFullYear

Method Declaration: getUTCFullYear() : Number

Description: Returns the year value in a Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutcfullyear.asp>

Method: getUTCHours

Method Declaration: getUTCHours() : Number

Description: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutchores.asp>

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutchores.asp>

Method: getUTCMilliseconds

Method Declaration: getUTCMilliseconds() : Number

Description: Returns the milliseconds value in a Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutcmilliseconds.asp>

Method: getUTCMinutes

Method Declaration: getUTCMinutes() : Number

Description: Returns the minutes value in a Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutcminutes.asp>

Method: getUTCMonth

Method Declaration: getUTCMonth() : Number

Description: Returns the month value in a Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutcmonth.asp>

Method: getUTCSeconds

Method Declaration: getUTCSeconds() : Number

Description: Returns the seconds value in a Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetutcseconds.asp>

Method: getVarDate

Method Declaration: getVarDate() : OleVariant

Description: Returns the VT_DATE value in a Date object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetvardate.asp>

Method: getYear

Method Declaration: getYear() : Number

Description: Returns the year value in a Date object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthgetyear.asp>

Method: parse

Method Declaration: parse(dateval : JSString) : Number

Description: Parses a string containing a date, and returns the number of milliseconds between that date and midnight, January 1, 1970.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthparse.asp>

Method: setDate

Method Declaration: setDate(numDate : Number) :

Description: Sets the numeric date of the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetdate.asp>

Method: setFullYear

Method Declaration: setFullYear(numYear : Number, numMonth : Number, numDate : Number) :

Description: Sets the year value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetfullyear.asp>

Method: setHours

Method Declaration: setHours(numHours : Number, numMin : Number, numSec : Number, numMilli : Number) :

Description: Sets the hour value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsethours.asp>

Method: setMilliseconds

Method Declaration: setMilliseconds(numMilli : Number) :

Description: Sets the milliseconds value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetmilliseconds.asp>

Method: setMinutes

Method Declaration: setMinutes(numMin : Number, numSec : Number, numMilli : Number) :

Description: Sets the minutes value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetminutes.asp>

Method: setMonth

Method Declaration: setMonth(numMonth : Number, dateVal : Number) :

Description: Sets the month value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetmonth.asp>

Method: setSeconds

Method Declaration: setSeconds(numSec : Number, numMilli : Number) :

Description: Sets the seconds value in the Date object using local time.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetseconds.asp>

Method: setTime

Method Declaration: setTime(numMilli : Number) :

Description: Sets the date and time value in the Date object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsettime.asp>

Method: setUTCDate

Method Declaration: setUTCDate(numDate : Number) :

Description: Sets the numeric date in the Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetutcdate.asp>

Method: setUTCFullYear

Method Declaration: setUTCFullYear(numYear : Number, numMonth : Number, numDate : Number) :

Description: Sets the year value in the Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetutcfullyear.asp>

Method: setUTCHours

Method Declaration: setUTCHours(numHours : Number, numMin : Number, numSec : Number, numMilli : Number) :

Description: Sets the hours value in the Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetutcheours.asp>

Method: setUTCMilliseconds

Method Declaration: setUTCMilliseconds(numMilli : Number) :

Description: Sets the milliseconds value in the Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetutcmilliseconds.asp>

Method: setUTCMinutes

Method Declaration: setUTCMinutes(numMin : Number, numSec : Number, numMilli : Number) :

Description: Sets the minutes value in the Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetutcminutes.asp>

Method: setUTCMonth

Method Declaration: setUTCMonth(numMonth : Number, dateVal : Number) :

Description: Sets the month value in the Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetutcmonth.asp>

Method: setUTCSeconds

Method Declaration: setUTCSeconds(numSec : Number, numMilli : Number) :

Description: Sets the seconds value in the Date object using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetutcseconds.asp>

Method: setYear

Method Declaration: setYear(numYear : Number) :

Description: Sets the year value in the Date object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsetyear.asp>

Method: toDateString

Method Declaration: toDateString() : String

Description: Returns a date as a string value.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtodatestring.asp>

Method: toGMTString

Method Declaration: toGMTString() : String

Description: Returns a date converted to a string using Greenwich Mean Time(GMT).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtogmtstring.asp>

Method: toLocaleDateString

Method Declaration: toLocaleDateString() : String

Description: Returns a date as a string value appropriate to the host environment's current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtolocaledatestring.asp>

Method: toLocaleString

Method Declaration: toLocaleString() : String

Description: Returns a date converted to a string using the current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtoLocalestring.asp>

Method: toLocaleTimeString

Method Declaration: toLocaleTimeString() : String

Description: Returns a time as a string value appropriate to the host environment's current locale.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtolealetimestring.asp>

Method: toDateString

Method Declaration: toDateString() : String

Description: Returns a date as a string value.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtodatestring.asp>

Method: toString

Method Declaration: toString(radix : integer) : string

Description: Returns a string representation of an object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtoString.asp>

Method: toTimeString

Method Declaration: toTimeString() : String

Description: Returns a time as a string value.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtotimestring.asp>

Method: toUTCString

Method Declaration: toUTCString() : String

Description: Returns a date converted to a string using Universal Coordinated Time (UTC).

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtoutcstring.asp>

Method: valueOf

Method Declaration: valueOf() : String

Description: Returns the primitive value of the specified object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthvalueof.asp>

Method: UTC

Method Declaration: UTC(numYear : Number, numMonth : Number, numDate : Number, numHours : Number, numMin : Number, numSec : Number, numMilli : Number) : Number

Description: Returns the number of milliseconds between midnight, January 1, 1970 Universal

Coordinated Time (UTC) (or GMT) and the supplied date.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthutc.asp>

3.2.2.8 JScript Object: Enumerator

Object: Enumerator

Description: Enables enumeration of items in a collection.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjenumerator.asp>

Object Methods

- atEnd
- item
- moveFirst
- moveNext

Method: atEnd

Method Declaration: atEnd() : Boolean

Description: Returns a Boolean value indicating if the enumerator is at the end of the collection.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthatend.asp>

Method: item

Method Declaration: item() : JSObject

Description: Returns the current item in the collection.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthitem.asp>

Method: moveFirst

Method Declaration: moveFirst() :

Description: Resets the current item in the collection to the first item.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthmovefirst.asp>

Method: moveNext

Method Declaration: moveNext() :

Description: Moves the current item to the next item in the collection.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthmovenext.asp>

3.2.2.9 JScript Object: Arguments

Object: Arguments

Description: An object representing the arguments to the currently executing function, and the functions that called it.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjarguments.asp>

Object Properties

- length

Property Name: length

Property Type: Number

Description: Returns the actual number of arguments passed to a function by the caller.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolengtharguments.asp>

3.2.2.10 JScript Object: Function

Object: Function

Description: Creates a new function.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjfunction.asp>

Object Properties

- arguments
- callee
- caller
- length

Object Methods

- apply
- call
- toString
- valueOf

Property Name: arguments

Property Type: JSArguments

Description: Returns the arguments object for the currently executing Function object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproarguments.asp>

Property Name: callee

Property Type: JSFunction

Description: Returns the Function object being executed, that is, the body text of the specified Function object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprocallee.asp>

Property Name: caller

Property Type: JSFunction

Description: Returns a reference to the function that invoked the current function.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprocaller.asp>

Property Name: length

Property Type: Number

Description: Returns the number of arguments defined for a function.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolengthfunction.asp>

Method: apply

Method Declaration: apply(thisObj : JSObject, argArray : JSArray) :

Description: Applies a method of an object, substituting another object for the current object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthapply.asp>

Method: call

Method Declaration: call(thisObj : JSObject, arg1 : JSObject, argN : JSObject) :

Description: Calls a method of an object, substituting another object for the current object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthcall.asp>

Method: toString

Method Declaration: toString(radix : integer) : string

Description: Returns a string representation of an object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtostring.asp>

Method: valueOf

Method Declaration: valueOf() : JSFunction

Description: Returns the primitive value of the specified object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthvalueof.asp>

3.2.2.11 JScript Object: ActiveXObject

Object: ActiveXObject

Description: Enables and returns a reference to an Automation object.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjactivexobject.asp>

3.2.2.12 JScript Object: Debug

Object: Debug

Description: An intrinsic global object that can send output to a script debugger, such as the Microsoft Script Debugger.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjdebug.asp>

Object Methods

- write
- writeln

Method: write

Method Declaration: write(value : String) :

Description: Sends strings to the script debugger.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthwritedebug.asp>

Method: writeln

Method Declaration: writeln(value : String) :

Description: Sends strings to the script debugger, followed by a newline character.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthwritelndebug.asp>

3.2.2.13 JScript Object: Math

Object: Math

Description: An intrinsic object that provides basic mathematics functionality and constants.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjmath.asp>

Object Properties

- E
- LN2
- LN10
- LOG2E
- LOG10E
- PI
- SQRT1_2
- SQRT2

Object Methods

- abs
- acos
- asin

- atan
- atan2
- ceil
- cos
- exp
- floor
- log
- max
- min
- pow
- random
- round
- sin
- sqrt
- tan

Property Name: E

Property Type: Number

Description: Returns the mathematical constant e, the base of natural logarithms. The E property is approximately equal to 2.718.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproe.asp>

Property Name: LN2

Property Type: Number

Description: Returns the natural logarithm of 2.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproln2.asp>

Property Name: LN10

Property Type: Number

Description: Returns the natural logarithm of 10.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproln10.asp>

Property Name: LOG2E

Property Type: Number

Description: Returns the base-2 logarithm of e, Euler's constant.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolog2e.asp>

Property Name: LOG10E

Property Type: Number

Description: Returns the base-10 logarithm of e, Euler's constant.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolog10e.asp>

Property Name: PI

Property Type: Number

Description: Returns the ratio of the circumference of a circle to its diameter, approximately 3.141592653589793.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprop1.asp>

Property Name: SQRT1_2

Property Type: Number

Description: Returns the square root of 0.5, or one divided by the square root of 2.

URL for more info: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprosqrt1_2.asp

Property Name: SQRT2

Property Type: Number

Description: Returns the square root of 2.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprosqrt2.asp>

Method: abs

Method Declaration: abs(Value : OleVariant) : integer

Description: Returns the absolute value of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthabs.asp>

Method: acos

Method Declaration: acos(Value : Number) : Number

Description: Returns the arccosine of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthacos.asp>

Method: asin

Method Declaration: asin(Value : Number) : Number

Description: Returns the arcsine of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthasin.asp>

Method: atan

Method Declaration: atan(Value : Number) : Number

Description: Returns the arctangent of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthatan.asp>

Method: atan2**Method Declaration:** atan2(x : Number, y : Number) : Number**Description:** Returns the angle (in radians) from the X axis to a point (y,x).**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthatan2.asp>**Method:** ceil**Method Declaration:** ceil(Value : Number) : integer**Description:** Returns the smallest integer greater than or equal to its numeric argument.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthceil.asp>**Method:** cos**Method Declaration:** cos(Value : Number) : Number**Description:** Returns the cosine of a number.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthcos.asp>**Method:** exp**Method Declaration:** exp(Value : Number) : Number**Description:** Returns e (the base of natural logarithms) raised to a power.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthexp.asp>**Method:** floor**Method Declaration:** floor(Value : Number) : integer**Description:** Returns the greatest integer less than or equal to its numeric argument.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthfloor.asp>**Method:** log**Method Declaration:** log(Value : Number) : Number**Description:** Returns the natural logarithm of a number.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthlog.asp>**Method:** max**Method Declaration:** max(number1 : Number, number2 : Number) : Number**Description:** Returns the greater of zero or more supplied numeric expressions.**URL for more info:** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthmax.asp>**Method:** min**Method Declaration:** min(number1 : Number, number2 : Number) : Number**Description:** Returns the lesser of zero or more supplied numeric expressions.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthmin.asp>

Method: pow

Method Declaration: pow(base : Number, exponent : Number) : Number

Description: Returns the value of a base expression taken to a specified power.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthpow.asp>

Method: random

Method Declaration: random() : Number

Description: Returns a pseudorandom number between 0 and 1.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthrandom.asp>

Method: round

Method Declaration: round(Value : Number) : integer

Description: Returns a supplied numeric expression rounded to the nearest integer.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthround.asp>

Method: sin

Method Declaration: sin(Value : Number) : Number

Description: Returns the sine of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsin.asp>

Method: sqrt

Method Declaration: sqrt(Value : Number) : Number

Description: Returns the square root of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthsqrt.asp>

Method: tan

Method Declaration: tan(Value : Number) : Number

Description: Returns the tangent of a number.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmhtan.asp>

3.2.2.14 JScript Object: RegExp

Object: RegExp

Description: An intrinsic global object that stores information about the results of regular expression pattern matches.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjregexp.asp>

Object Properties

- index
- input
- lastIndex
- lastMatch
- lastParen
- leftContext
- rightContext

Property Name: index

Property Type: Number

Description: Returns the character position where the first successful match begins in a searched string. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproindex.asp>

Property Name: input

Property Type: String

Description: Returns the string against which a regular expression search was performed. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproinput.asp>

Property Name: lastIndex

Property Type: Number

Description: Returns the character position where the next match begins in a searched string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolastindex.asp>

Property Name: lastMatch

Property Type: String

Description: Returns the last matched characters from any regular expression search. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolastmatch.asp>

Property Name: lastParen

Property Type: String

Description: Returns the last parenthesized submatch from any regular expression search, if any. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprolastparen.asp>

Property Name: leftContext

Property Type: String

Description: Returns the characters from the beginning of a searched string up to the position

before the beginning of the last match. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproleftcontext.asp>

Property Name: rightContext

Property Type: String

Description: Returns the characters from the position following the last match to the end of the searched string. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprorightcontext.asp>

3.2.2.15 JScript Object: RegExp

Object: RegExp

Description: An object that contains a regular expression pattern along with flags that identify how to apply the pattern.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsobjregexexpression.asp>

Object Properties

- ignoreCase
- multiline
- source

Object Methods

- compile
- exec
- test

Property Name: ignoreCase

Property Type: Boolean

Description: Returns a Boolean value indicating the state of the ignoreCase flag (i) used with a regular expression. Default is false. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsproignorecase.asp>

Property Name: multiline

Property Type: Boolean

Description: Returns a Boolean value indicating the state of the multiline flag (m) used with a regular expression. Default is false. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jspromultiline.asp>

Property Name: source

Property Type: String

Description: Returns a copy of the text of the regular expression pattern. Read-only.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsprosource.asp>

Method: compile

Method Declaration: compile(pattern : String, flags : String) :

Description: Compiles a regular expression into an internal format for faster execution.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthcompile.asp>

Method: exec

Method Declaration: exec(str : String) : JSArray

Description: Executes a search on a string using a regular expression pattern, and returns an array containing the results of that search.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthexec.asp>

Method: test

Method Declaration: test(str : String) : Boolean

Description: Returns a Boolean value that indicates whether or not a pattern exists in a searched string.

URL for more info: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56jsmthtest.asp>

3.3 Upgrading Older Action Packages

3.3.1 Upgrading .NET Actions

If you created action packages using older versions of FinalBuilder, then you will have noticed some major changes in the way .NET based actions are created.

The new API for creating .NET actions is located inside the VSoft.CustomActionApi.dll assembly. This is the recommended API to use for creating new action packages.

Backwards Compatibility

It is still possible to reference the old Interop interfaces, instead of the new CustomActionApi classes. It is also possible to upgrade older packages which use the *Interop.FBxxCustomActionInterfaces* namespace.

The Interop assembly is now called VSoft.CustomActionInterop.dll. Note that this assembly also needs to be referenced when using the new API.

The *Interop.FBxxCustomActionInterfaces* namespace has been renamed *VSoft.CustomActionApi.Interop*. This namespace contains all of the interfaces and enumerated types from the old Interop namespace. Interop interface names have also changed. The "IFB" prefix is now shortened to "I".

For example, the interface formerly known as *Interop.FB50CustomActionInterfaces.IFBCustomAction* is now known as *VSoft.CustomActionApi.Interop.ICustomAction*.

3.3.2 COM-Based Actions

FinalBuilder versions older than 6 included support for developing custom actions using COM Interfaces. This type of action is no longer supported.

It is recommended that COM-based custom actions are rewritten using .NET or script.

If access to COM components is mandatory, it is recommended to either create and register scriptable automation objects to be accessed from a script-based actions, or to create and register a type library to be accessed via .NET Interop from a .NET-based action.