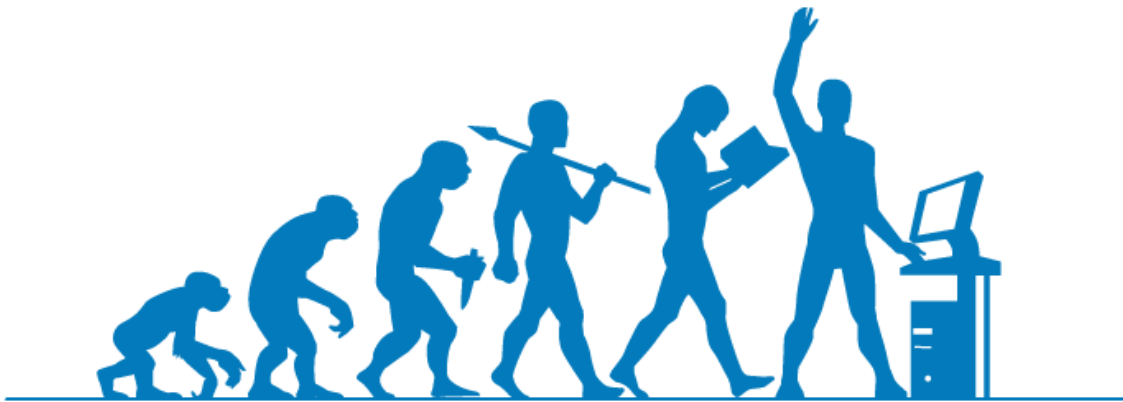




ADMIN SCRIPT EDITOR 2.2

REVIEWERS GUIDE



STEP UP TO THE MORE EVOLVED EDITOR!

Summary

Admin Script Editor is an Integrated Scripting Environment (ISE) which goes far beyond simple script editing. With Admin Script Editor, you can easily generate script code using powerful wizards, access intuitive help at every turn, package scripts for deployment and the list goes on.

In fact, the list of features is so extensive that it may be difficult to identify and see the amazing value that sits before you. To that end, this document aims to familiarize you with the many features that set Admin Script Editor (ASE) apart from the other tools on the market. Most tools are not even comparable to ASE, existing as simple text editors that offer only basic keyword highlighting. You may be aware of at least one other editor out there making weak attempts to duplicate some of these valuable and innovative features (but charging almost three times the price) – don't be tricked- take twenty minutes with ASE and make an informed decision.

If you are looking to develop ActionScript – get Flash. If you are looking to develop in VB.NET - get Visual Studio. If you are looking to write scripts with VBScript, KiXtart, AutoIt or Batch- get ASE!

ASE is an editor designed specifically for Windows Administrators. The interface is not cluttered with functionality you will never use and you will not need to scroll by dozens of development languages to find the one or two you make use of. In fact, those few you probably do use are the sole focus of ASE and it is this focus which allows us to provide unparalleled support for those languages – support which no other product can match.

While we encourage you to start with this basic overview that takes you through some of our key features, the following list of contents is provided to help you jump to the items that interest you most.

[Layout/Organization](#) – take a quick walk through the Integrated Scripting Environment and get a feel for the easily uncovered gems that hide themselves!

[Editing/Scripting Help](#) – help while you type, with ScriptSense, ScriptComplete, and keyword help and sample scripts are just a click away!

[Powerful Code Wizards](#) – even expert scripters will appreciate some of these great tools. ASE provides a growing array of exclusive code wizards designed to make the path to that script you desire as quick and painless as it can be.

[Deploy Scripts as Secure Executable Packages](#) – create executables from your scripts along with any dependant files. Configure packages to show a taskbar icon during execution, log any errors to the event log and even run with alternate credentials!

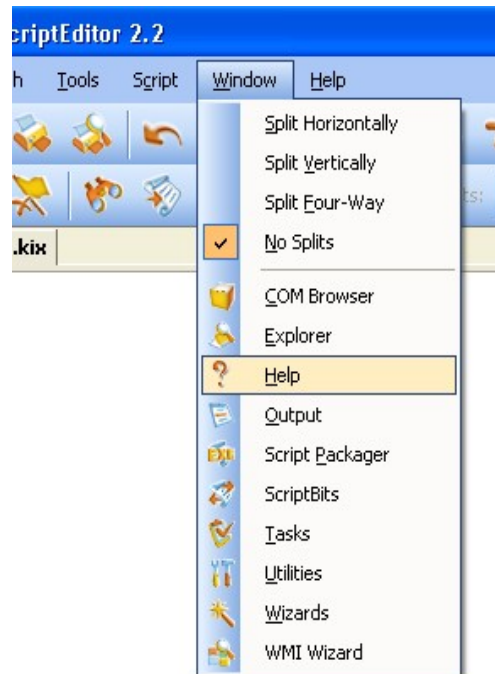
[More Now and More to Come](#) – this last section bullets some of the other cool things to look for when playing with ASE as well as a look at what is making its way off the drawing board and into the next release!

Layout/Organization

ASE provides you maximum work area by auto-hiding the many tools it offers by default. You can access any of these tools from under the "Window" menu at the top of the ISE. If you move your mouse over any of the tabs you see on the right, left and bottom of the window- you will see the feature slide out for your use. Move the mouse away, and the window slides back out of the way!

If you are working with a particular feature and want to keep it in view, just click the pushpin icon to "pin" the feature open. You can drag and arrange these windows from their default configuration to customize your own layout. This layout is automatically remembered, next time you open ASE. Want it back the way it was? A simple Reset Window Layout option is available from the options dialog.

Take a moment to hover over the several feature menus. Moving clockwise from the right, below is a quick description of each to help you along:



Wizards – All wizards provide language appropriate code based on that of the active script tab. A few built in wizards include a Message Box Wizard and a Character Code Wizard. At the top you'll find some included "Custom Wizards". Check out the Registry Wizard which provides an interface which looks like RegEdit, but instead of editing the registry- it generates language appropriate code! Many more wizards are being offered online at <http://www.adminscripteditor.com/editor/wizards>

Utilities – The Utilities feature offers a few handy scripting tools including Custom Tools, which allows you to add shortcuts to your favorite external utilities. The Error Code Lookup tool will provide you the text for any Win32 error code and the Environment Variables tool lets you preview available environment variables and their local computer value. A place for generating GUIDs and converting between hexadecimal and decimal numbers are also provided as useful scripting utilities here.

Script Packager™ – This popular feature (discussed in more detail later in this document) allows you to package your script and any desired support files into secure, encrypted executable packages that may even be configured to run under alternate credentials.

Help – The help window at the bottom provides a very handy way to access keyword documentation for your desired scripting language. From within your script, right click on any keyword and choose the bold help option to have this window slide up with our own maintained documentation. By default, this information is provided via the AdminScriptEditor.com website syntax references. However, you may specify an alternate online resource and we even offer an offline installation of our reference for

the languages you desire so help is always a click away. This help includes details on arguments, return values, examples, notes and more – even a button to perform a search of our online script library so you can see use of the keyword in real-world implementations.

Tasks – Insert place-holders in your script as reminders for code you need to write or update in the future simply by adding a TASK comment in your script. This list represents a list of tasks combined from all open scripts; just click on any one to switch to the target script and jump to the task location.

Output – For languages that support “Standard Out” such as VBScript and Batch, output may be optionally redirected to this window. Run the script from within the editor and see details on the interpreter being used, any output or errors generated by the script and the total script runtime here (it will automatically slide up into view when in use).

ScriptBits™ - Stop writing the same code again and again! This time-saving feature allows you to easily insert code into your scripts. To create a ScriptBit™, just select some code and right click! You can name it whatever you wish and can create new categories just by creating new folders. Organize them anyway you like and build a library of reusable code that even support dynamic replacement of specified elements (see documentation for details).

WMI Wizard – Most are familiar with the simple Script-O-Matic tool offered by Microsoft, a helpful tool for generating simple WMI queries that return all property values. The ASE WMI Wizard takes this concept several steps forward by allow you to select only the properties you wish- even providing sample data for the selected property so you can get just what you need. Options exist to help you customize the output generated by the Wizard in several ways including how the query is written, variable names, conditions and more. The trial version allows limited use of this feature- but you can press the “Show” button to see the code that would be inserted as often as you wish, so click away!


COM Browser – The COM Browser allows you to select any COM objects you wish to investigate from those installed on your system. From there, you may click through to see what properties, methods and other elements are offered. Where supported, a description of each item is displayed along with a view of what parameters the object may require when scripting. While many prefer this ability to browse objects, this same information is also dynamically available when scripting through our *ScriptComplete™* support discussed later in this document.

Explorer – Looking for scripts? This helpful tool lets you browse any local or network location for supported file types. You can even set the root of this view by selecting the location where you keep your scripts and pressing the “Set Root” toolbar button. This will become the root of the Explorer view here and will also serve as the default location for scripts you create in ASE.

Editing/Scripting Help

ASE offers help at every corner. No matter what your scripting experience, help with the unfamiliar is bound to become a need. How much does a good scripting book cost? Well consider that amount a savings when your working with ASE- everything you need is right at your fingertips!

ScriptSense™ goes far to make scripting easier and provides two distinct features: ScriptTips™ and ScriptComplete™. With ScriptTips™, parameters to functions are presented as you type while with ScriptComplete™ items such as COM objects and macro variables are presented as drop-down menus as you need them. Not enough? Right click on a keyword and get help right within ASE – even see sample scripts that make use of specific keywords for ideas and real-world examples! Let's take a look...

1. Create a new script, there are a few ways to do this but for ease – let's just hit the new script toolbar icon at the top left  (note that all icons provide tool tips as to what they do for you when you hover over them)
2. For this example, we will choose VBScript.
3. You may enter an alternate path and file name if you wish. The default path and file name provided is %USERPROFILE%\My Documents\AdminScriptEditor\untitled.vbs
4. Now we'll type the code for a sample VBScript that will help show just how helpful ASE can be in scripting. First we will create a reference to the File System Object...

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

Note that when you hit the first double quote a list of available objects you may create is displayed to you. The list updates as you type, so you don't need to scroll to find what you want. You may also note that there are no duplicate entries or senseless entries as you may see when working with other products. When you find what you want in this clean list of COM libraries, press Enter or Tab to choose it. You'll also notice there are other optional parameters you can provide here. Press the close parenthesis to end the statement.

5. Now let's have our script prompt the user for a path and file name...

```
selectedFile = InputBox("Enter the full path and filename  
to check")
```

Note that when you hit the open parentheses a ScriptTip™ is displayed showing you what parameters are accepted by the InputBox function. Those in brackets ("[...]") are optional arguments. This function is supported for all VBScript, KiXtart and AutoIt functions.

6. Next we create an object reference to the provided file...

```
Set MyFile = fso.GetFile(selectedFile)
```

Note that because the variable *fso* is set as a reference to the File System Object, when you type a period after it, ScriptComplete™ provides you a list of properties and methods available to this object.

7. Now we will generate a message box that displays how old the specified file is in days...


```
Resp = MsgBox(MyFile.Name & " has not been changed in " &  
DateDiff("d", MyFile.DateLastModified, Now) & " days",4420)
```

Note that when typing this line you see ScriptTips™ for MsgBox and DateDiff and you also see ScriptComplete™ selections for the MyFile object reference. Another thing to note here is that when the ScriptTip™ for the DateDiff function closes, it takes you back to the ScriptTip™ for MsgBox again as a *nested ScriptTip™*. Further, when you type a comma in your function arguments the next parameter is displayed in bold to help you keep your place!

8. Finally, we will echo the value returned by the MsgBox function...

```
WScript.Echo "The user answered: " & Resp
```

Note that WScript activates ScriptComplete™ as a natively supported object and you did not need to create an object reference.

9. Wondering about a certain keyword? Right click on any of the keywords (highlighted in blue by default) and choose "<keyword> help" from the context menu. Help on every command is right in the application (save yourself the cost of a reference manual) and click the button displayed for scripts that use the keyword in our online script library, providing you real-world examples!
10. Now let's run that script! Press F5 or press the execute toolbar icon . You'll see that as the script executes the output window slides up from the bottom displaying any information that would normally be returned to a command prompt. Enter a file and path at the script prompt (for example, C:\BOOT.INI) and the number of days will be displayed using the message box function. Depending on if you press, "Yes" or "No" the appropriate value will appear in the output window as we specified (6 or 7 respectively). Your script could then take action on the response, but for demonstration purposes, we'll stop here.

```
Set fso = CreateObject("Scripting.FileSystemObject")  
SelectedFile = InputBox("Enter the full path and filename to check")  
Set MyFile = fso.GetFile(SelectedFile)  
Resp = MsgBox(MyFile.Name & " has not been changed in " & DateDiff("d",  
MyFile.DateLastModified, Now) & " days",4420)  
WScript.Echo "The user answered: " & Resp
```

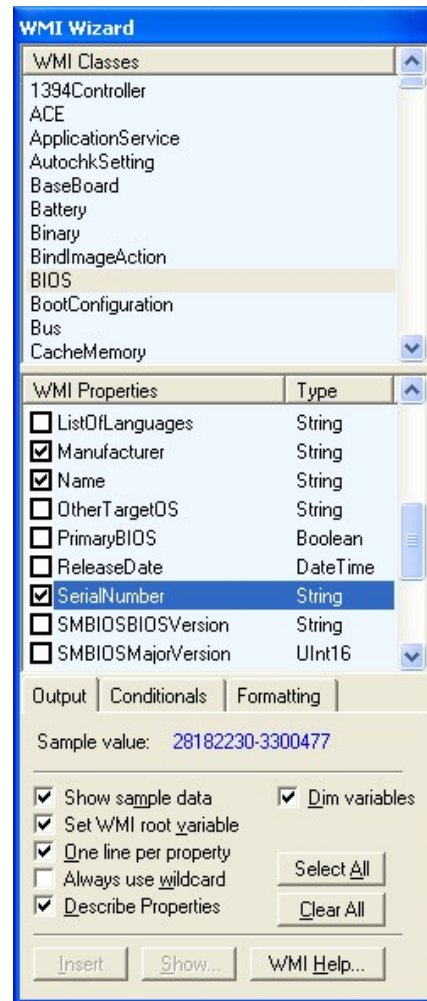
Entire Sample VBS Script

Powerful Code Wizards

Code wizards help you to quickly generate code- particularly code that commonly requires some level of reference due to complex parameters or values that would normally need to be researched. The wizards provided by ASE work appropriately for all supported languages. For example, the Message Box Wizard lets you construct the line of code to create a message box prompt by collecting pertinent information and then inserting the language-appropriate code for the currently open script file. In the case of the message box command (in VBS, KiXtart and AutoIt) there is a single bitwise value that dictates the number of buttons, icon displayed, default button, and modality. Not only would you need a reference to determine the value, but you'd need to add them all up to come to the desired value for this one function argument.

ASE offers a few helpful wizards, but here we will focus on the WMI Wizard. WMI (Windows Management Instrumentation) provides a wealth of information about a given system including its Serial Number, Model, Manufacturer, and status information on such components as the battery, drive, network, display, etc.

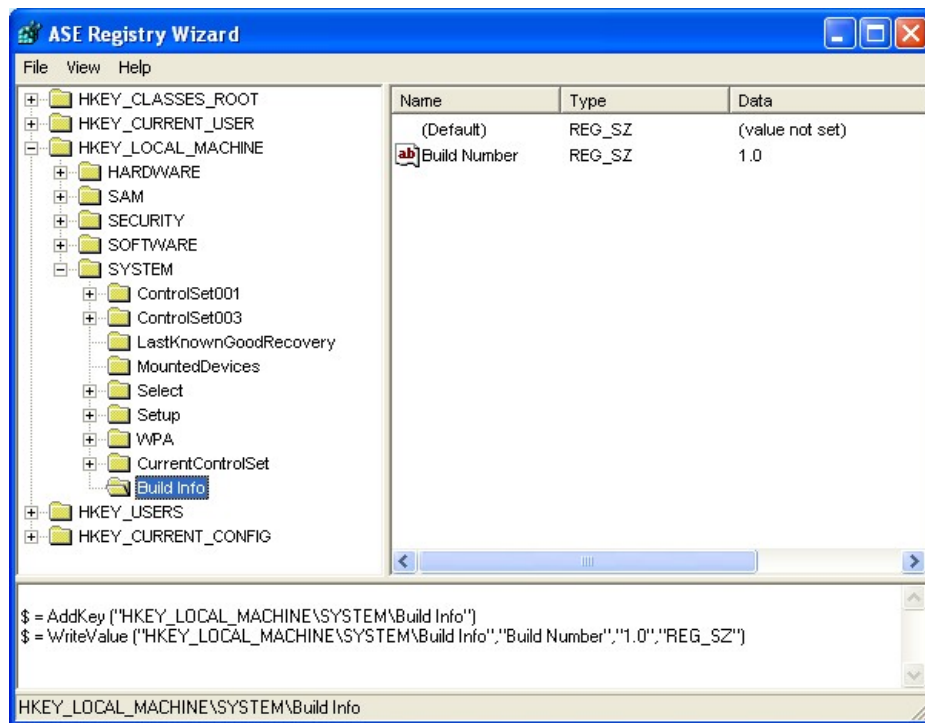
Browsing for the details you may wish to retrieve is simplified by providing a sample value for the selected property based on that of the local system. This helps you to avoid a need to "groom" the generated code to suit your needs. In fact, you have very granular control over how the generated code will appear by manipulating the many options provided. Click around and have a look at the valuable information that can be gleaned from WMI and you are sure to spark your imagination and come up with many potential uses for these details!



Another innovation of the Admin Script Editor is the ability to create your own custom wizards using any language that supports COM (VBScript, KiXtart, or any one of several development languages). This not only allows you to create and share your own wizards, but allows iTripoli to offer new Wizards outside of the normal development cycle of the editor itself. A growing number of custom Wizards are provided at <http://www.adminscripteditor.com/wizards>

A great example is the ASE Registry Wizard. This custom wizard (included with v2.2 of ASE) will allow you to generate language-appropriate code for reading, deleting, adding and editing the Windows registry. Let's check it out:

1. Create a new script, there are a few ways to do this but for ease – let's just hit the new script toolbar icon at the top left (note that all icons provide tool tips as to what they do for you when you hover over them)
2. For this example, we will choose KiXtart.
3. Specify a path and/or filename for your script. Note that if you do not specify a path the default shown below the filename field will be used (configurable by setting the desired root directory in the Explorer window).
4. With you new KiXtart script established, double-click the Registry Wizard listing from the Custom Wizards area of the Wizard window.



5. For this example, we will create a new key and value to store our own information about the system. Double click HKEY_LOCAL_MACHINE and then SYSTEM so that the contents of SYSTEM are displayed.
6. Right-click on SYSTEM and choose "Add Key" from the context menu. Don't worry- this wizard does not manipulate the systems registry, just this temporary view provided in the registry. If you run the script, that is another story- but the wizard itself makes no changes to your system.
7. Type "Build Info" for the new SubKey name and press OK.

8. You should see a new SubKey named "Build Info" appear at the bottom of the list of SubKeys displayed for HKEY_LOCAL_MACHINE\SYSTEM. Double-click on this new SubKey.
9. Right-click on the left pane where the SubKey values would be displayed and choose "Add > String" to create a new string value in this SubKey.
10. Type the value name as "Build Number", the data as "1.0" and press OK.
11. You'll see this new value appear in the registry view of the registry under the specified SubKey.

You can also get values, check for keys, delete keys and the list goes on. The key thing to take note of here is the bottom of the registry editor: as you perform actions in the wizard, language-appropriate code is generated here. When you have what you need, just press "Insert Code" or "Insert Code and Exit" from the File menu at the top of the display!

There are several wizards provided today and more are always on the way, from our own developers at iTripoli or from other smart, helpful scripters like you!

Deploy Scripts as Secure Executable Packages

A ground-breaking feature of the Admin Script Editor is its ability to create executable packages from your scripts with the ASE ScriptPackager™. This innovative feature provides a convenient means of distributing your script, but what else does this buy you?

- ▶ The script and any optional support files (such as your command interpreter) are encrypted in the executable file.
- ▶ Ability to run with alternate credentials – perform operations in a logon script you wouldn't otherwise be able to include. Wrap application installations for locked down users.
- ▶ You can also keep code from prying eyes by protecting the extracted source files during execution with NTFS and EFS security!
- ▶ The ability to include a taskbar icon with custom tooltip during execution provides an option to let users know a script is running without getting in their way.
- ▶ Does your script make use of a command line utility? Does it perform a silent installation that needs access to a setup.exe file? You can include any support files you need right in the executable package!

No project files! While some would complicate your scripting environment to satisfy the needs of developers- There is no need to complicate scripting by having to manage "projects". The idea behind scripting is to keep it simple and to the point. That said, scripts can be extremely complex with hundreds of lines and several external files to be utilized (an excellent case for the ScriptPackager™) but this all stems from your script files, just as you have always managed them.

While this is one of the only features not included in the Trial Edition of ASE, we do provide [full documentation for your review](#). Further, if you would like to see a package in action there are a couple of helpful utilities available which use the ASE ScriptPackager™ at <http://www.appdeploy.com/downloads/local>

In fact, the included ASE Registry Wizard is actually a packaged KiXtart script! Using the freeware KiXforms GUI control and the ScriptPackager™ you too can create scripts that may be easily confused for full-blown applications.

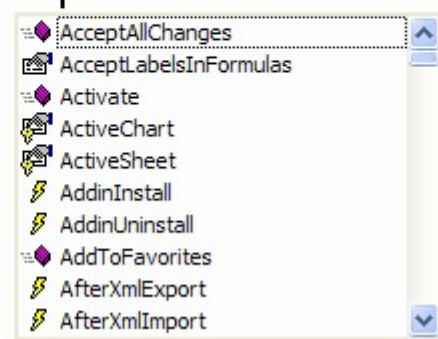
Not enough? Please feel free to [contact us](#) and we will be happy to assist in packaging a sample script for you so that you may see the alternate credentials and file security features in action for yourself!

More Now and More to Come!

Admin Script Editor is always evolving. Right now registered users have access to version 2.3 which offers enhanced ScriptTips™ and support for WSF/WSC, HTA, and HTML/ASP files. ScriptSense™ has also been enhanced to support a more powerful hierarchical ScriptComplete™ than any current script editor on the market. Try typing the following line with any other script editor:

```
Set fso = CreateObject("Scripting.FileSystemObject")
Set objExcel = CreateObject("Excel.Application")

objExcel.workbooks.Open(fso.GetFile("c:\ServerList.txt"))
.OpenAsTextStream().ReadLine().|
```



What is next? Let's just say, you can expect to see ASE do everything you might see in another product and then some. ASE is not done breaking new ground and setting the standard for what you should expect from your Integrated Scripting Environment.