

Service Bulletin: PrimalScript 2007 Build 567

This latest service build of PrimalScript 2007 features a number of improvements, large and small. Some of these are based on user suggestions; others are the result of our continuous work to improve PowerShell support in our products. This document will provide you with an overview of these changes.

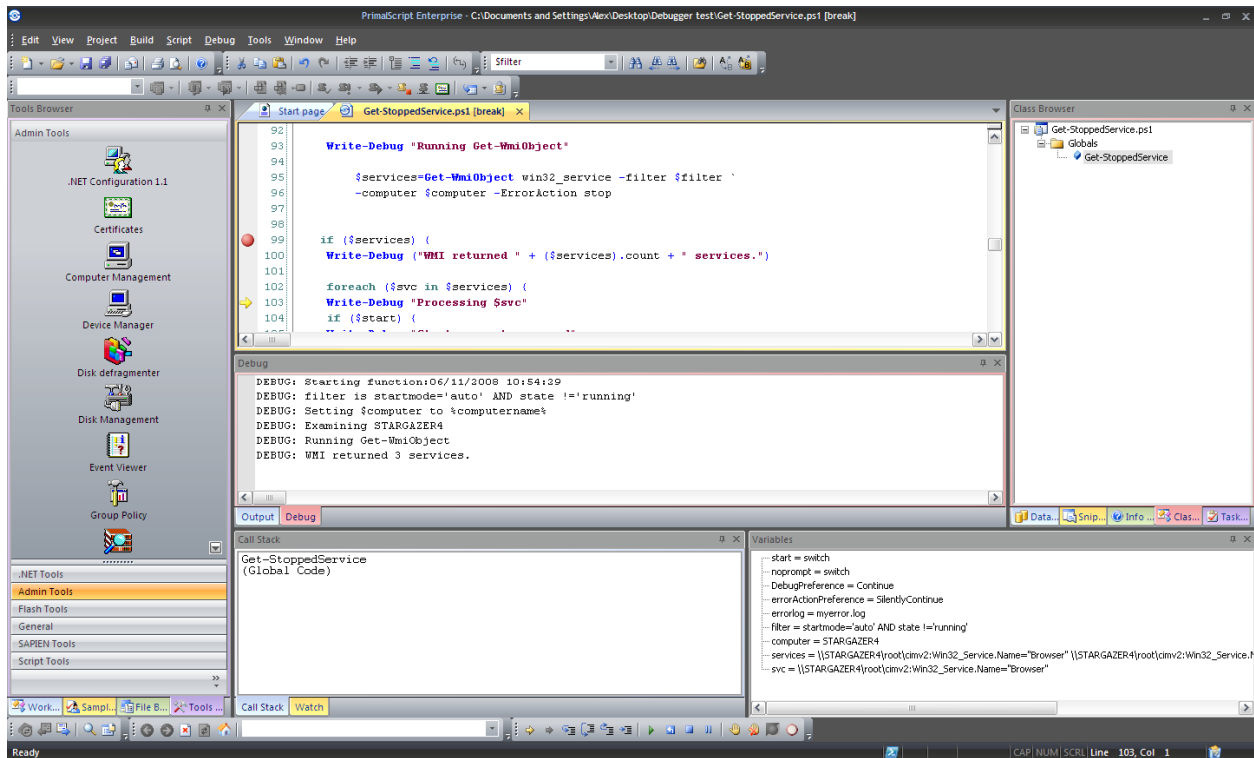
Please note that not all improvements and additions mentioned here apply to every edition of PrimalScript 2007.

Summary of changes

PowerShell Debugger now integrated into Professional and Enterprise Editions	2
PrimalSense for PowerShell operators	4
Find feature adds "Find All" and multi-line input	5
Code signing for VBScript files now can use PFX files	8
Parameter prompting for PowerShell scripts	9
Registry files are now a supported file type	10

PowerShell Debugger now integrated into Professional and Enterprise Editions

PrimalScript 2007 now includes a PowerShell debugger. Despite the fact that PowerShell 1.0 has no debugger API we managed to coax it into compliance. The debugger, as shown below, features the usual stepping functions, automatic variable display and call stack display.



The call stack actually displays the value of the parameters for a particular call.

Using the following code:

```
13
14 function write-something($argument)
15 {
16     Write-Host $argument
17 }
18
19 function do-something($action)
20 {
21     if($action -eq 1)
22     {
23         write-something("Doing one thing")
24     }
25     else {
26         write-something("Doing something else")
27     }
28 }
29
30 do-something(1)
31 do-something(2)
32
```

We can demonstrate what the call stack looks like each time we hit that breakpoint:

```
Call Stack
write-something (Doing one thing)
do-something (1)
(Global Code)
```

And the second time:

```
Call Stack
write-something (Doing something else)
do-something (2)
(Global Code)
```

As you can see, the real live parameter values show up in the call stack.

Of course this is only a 1.0 release of our PowerShell debugger, so some features are missing that you may look for. Please make sure that you visit our wish list forum at support@sapien.com and let us know what you consider the most important improvements for the next versions.

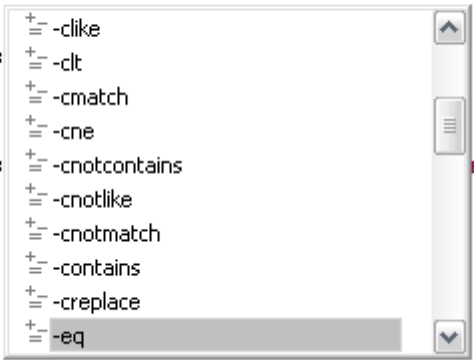
Please note that this PowerShell debugger does work with PowerShell 1.0 and does not require a PowerShell 2.0 CTP to be installed.

Although we do not anticipate any major problems, individual 2.0 scripts may not run or be supported by the debugger. In such a case please let us know and send your script inside a zip file to support@sapien.com.

PrimalSense for PowerShell operators

Occasionally we really mess up. Completely missing PrimalSense for PowerShell operators is one of those cases. Many of our customers informed us about this shortcoming and we fixed this omission in this service release.

```
19 function do-something($action)
20 {
21     if($action -e 1)
22     {
23         write-s
24     }
25     else {
26         write-s
27     }
28 }
29
30 do-something(1)
31 do-something(2)
32
```



The screenshot shows a PowerShell console window with a list of operators. The list is displayed in a scrollable area with a search box and a list of operators. The operators listed are: -clike, -clt, -cmatch, -cne, -cnotcontains, -cnotlike, -cnotmatch, -contains, -creplace, and -eq. The -eq operator is highlighted in the list.

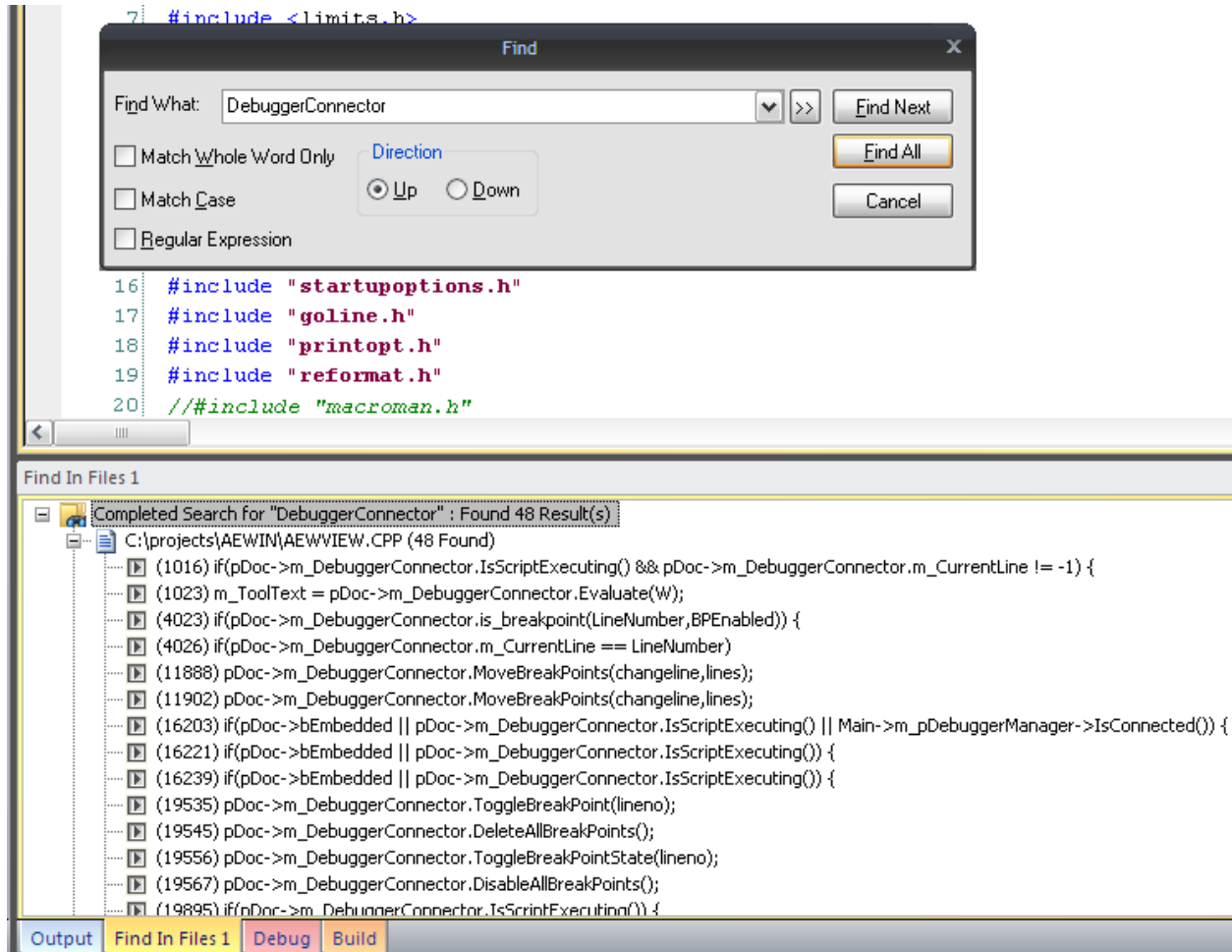
In case you wonder where PrimalScript takes these operators, look into the “languages” folder where PrimalScript is installed. You’ll find a file called “psh.operators” which contains the list.

With the frequency of new CTPs coming out of the PowerShell team you may have to add a new operator or two manually before we get to it.

Find feature adds “Find All” and multi-line input

Ever since we introduced the new “Find In Files” output format, customers have been asking for the same output from a “Find All” feature.

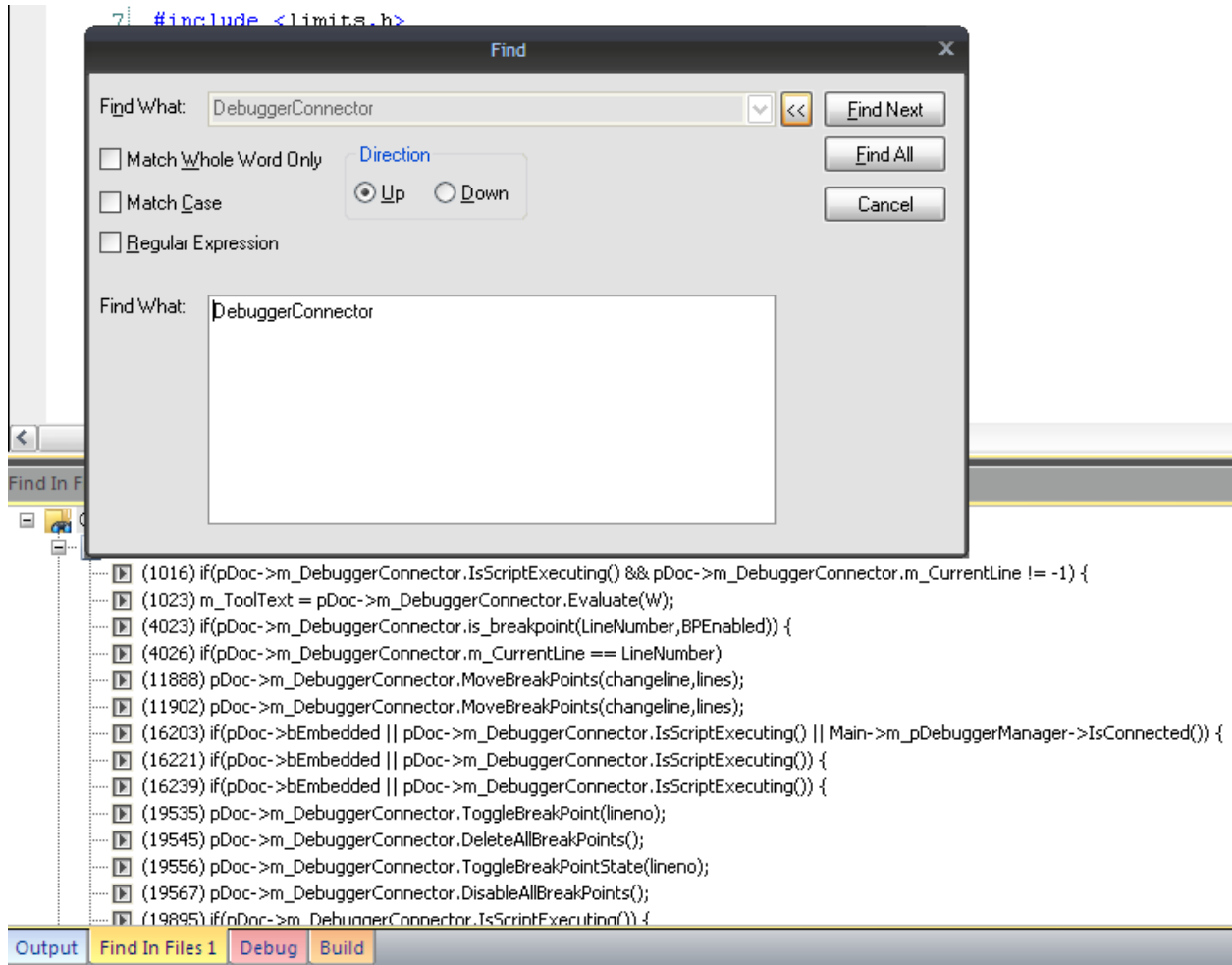
I am pleased to announce that this is now available with this service release.



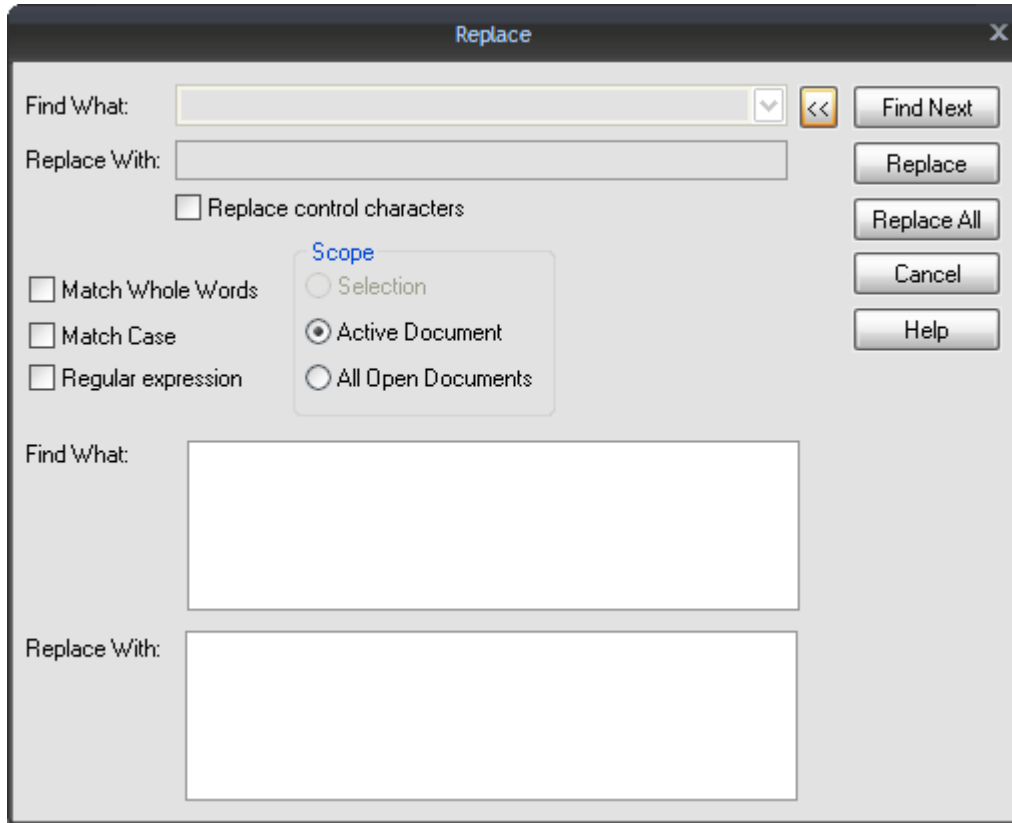
As with “Find In Files” the icon in front of a hit changes once you have clicked on it, so it is easy to keep track of what locations you already reviewed.

Another popular feature request was the ability to search and replace multi-line expressions. With this service release we have added an expand button next to the search expression combo box.

Pressing it will expand the dialog to reveal a multi-line input box, in case you are searching for larger expressions than just a single word or two.



Naturally the Replace dialog has received the same treatment and the Expand/Hide button reveals the multi-line equivalent fields for your expressions.



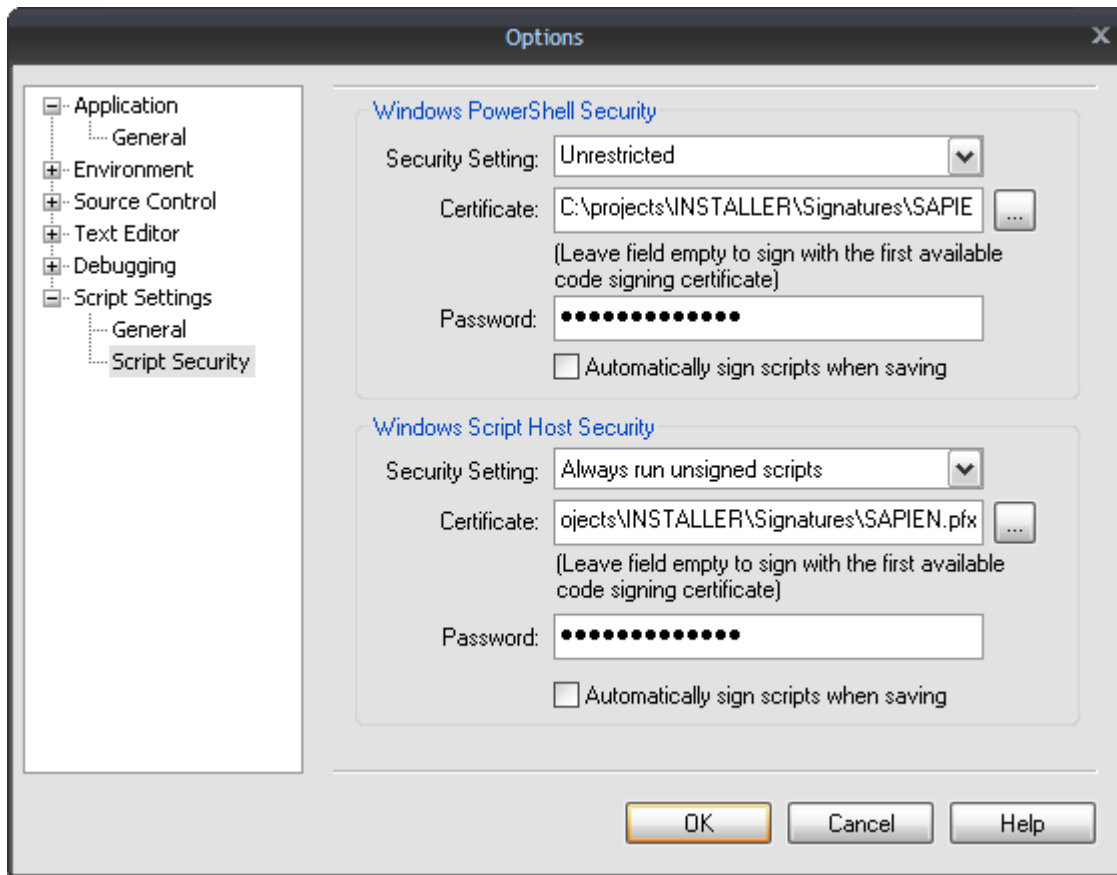
We hope that these enhancements make the find and replace functions even more useful to you. Please make sure to visit our wish list forum at <http://support.sapien.com> and let us know if there are any other enhancements you would like to see.

Code signing for VBScript files now can use PFX files

In many organizations and corporations code signing is mandatory for all scripts. Recent updates to Microsoft Windows have left many users with a script signing object that no longer works.

The “popular” file format for certificates has also changed to the PFX (Personal Information Exchange File) format, which is the same format Microsoft PowerShell prefers.

In response to this we have changed the script signing options in PrimalScript slightly:

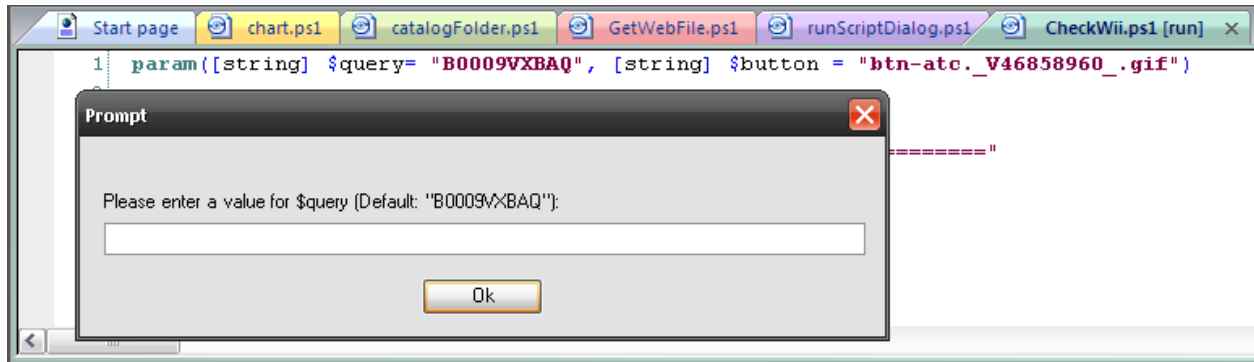


You can now specify a PFX file and its password just as you can for PowerShell. Please note that by using this option you no longer have to install that certificate on your computer. If you do not have a PFX file but any other format, please contact your certificate provider. There are also a number of conversion tools which are part of various Microsoft SDKs. Please be aware that we cannot redistribute these tools and that you need to get those from Microsoft.

Alternatively, you can leave the Certificate field empty. In that case PrimalScript will use Microsoft’s script signer object to sign your script. Just as before, it will use the first code signing certificate it finds.

Parameter prompting for PowerShell scripts

Testing a Powershell script with parameters can be tedious, especially if you want to test with a variety of arguments. To make this process a bit easier, PrimalScript will now prompt for anything defined in a scripts param block just prior to running it or starting a debugging session.

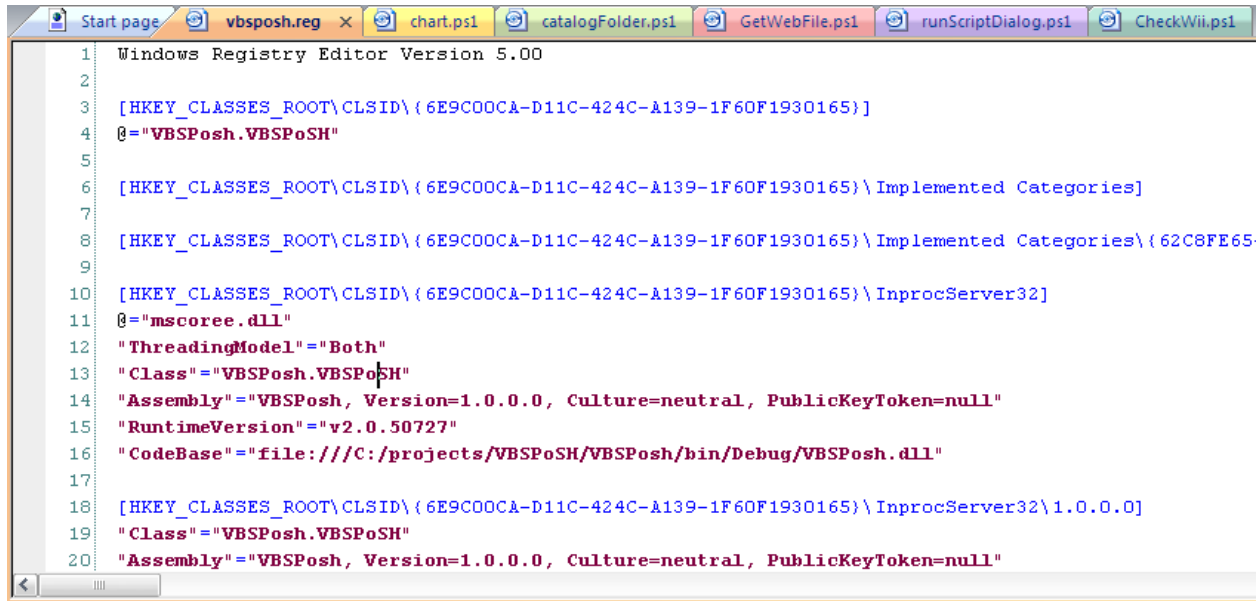


Note that any default values you have assigned to your parameters should show up in the prompt dialog and will be assigned in case you simply press "Ok" without entering a value.

Registry files are now a supported file type

A user in our wish list forum suggested that we add REG files as a supported file type to PrimalScript.

Since this is actually really easy to do, we went ahead and added this to the very next service build. If you open any REG file, it gives you basic coloring in order to keep sections and strings apart.



```
Windows Registry Editor Version 5.00
[HKKEY_CLASSES_ROOT\CLSID\{6E9C00CA-D11C-424C-A139-1F60F1930165}]
@="VBSPoSH.VBSPoSH"
[HKKEY_CLASSES_ROOT\CLSID\{6E9C00CA-D11C-424C-A139-1F60F1930165}\Implemented Categories]
[HKKEY_CLASSES_ROOT\CLSID\{6E9C00CA-D11C-424C-A139-1F60F1930165}\Implemented Categories\{62C8FE65-
[HKKEY_CLASSES_ROOT\CLSID\{6E9C00CA-D11C-424C-A139-1F60F1930165}\InprocServer32]
@="mscorlib.dll"
"ThreadingModel"="Both"
"Class"="VBSPoSH.VBSPoSH"
"Assembly"="VBSPoSH, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null"
"RuntimeVersion"="v2.0.50727"
"CodeBase"="file:///C:/projects/VBSPoSH/VBSPoSH/bin/Debug/VBSPoSH.dll"
[HKKEY_CLASSES_ROOT\CLSID\{6E9C00CA-D11C-424C-A139-1F60F1930165}\InprocServer32\1.0.0.0]
"Class"="VBSPoSH.VBSPoSH"
"Assembly"="VBSPoSH, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null"
```

“Running” a registry file will simply call Regedit.exe with the complete filename. We will add PrimalSense for registry paths and values at a later date.

If you have any suggestions, head over to our wish list forum at support.sapien.com and give it a try. Your wish feature may be added faster than you think.