

Safer

Source:

<http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.jscript/2007-10/msg00053.html>

- *From:* "Toad" <toad@xxxxxxx>
 - *Date:* Sat, 13 Oct 2007 02:24:25 GMT
-

Ok, so XP Home does not have Group Policy Editor like XP Pro that allows you to set Software Restriction Policies. But, XP Home does support the software restriction policies themselves.

So, here is a WSF script to allow one to set a SRP for a path that works on XP home and pro... I find it useful. The default policy is to disallow a path...

```
<package>
```

```
<job>
```

```
<runtime>
```

```
<description>SAFER adds or deletes a software restriction policy for a path and works on XP Home and Pro<\/description>
```

```
<named name="path" helpstring="Path to use for software restriction policy" type="string" required="true" />
```

```
<named name="level" helpstring="Set path policy using level (trust, limit, constrain, untrust, disallow)" type="string" required="false" />
```

```
<named name="user" helpstring="Set path policy for current user rather than local machine" type="simple" required="false" />
```

```
<named name="delete" helpstring="Delete existing policy for path" type="simple" required="false" />
```

```
<named name="interactive" helpstring="Run Safer interactively" type="simple" required="false" />
```

```
<\/runtime>
```

```
<script language="JScript">
```

```
/**Start Encode**
```

```
// Safer
```

```
var oShell = WScript.CreateObject("WScript.Shell");
```

```
var oTypeLib = WScript.CreateObject("Scriptlet.TypeLib");
```

```
var oFileSystem = WScript.CreateObject("Scripting.FileSystemObject");
```

```
var oLevelDictionary = WScript.CreateObject("Scripting.Dictionary");
```

```
var oArgs = WScript.Arguments;
```

```
var oArgsNamed = WScript.Arguments.Named;
```

```
var sGUID = oTypeLib.GUID.toString().substr(0,38);
```

Safer

```
var sRegKey = "HKLM";
var sSaferLevel = "0";
var sRegValue = "";
var sDeleteRegValue = "";
var sOldRegValue = "";

try
{
if (oArgsNamed.Exists("interactive"))
{
oShell.Run("\"" + WScript.FullName + "\" //Job:interactive \"\" +
WScript.ScriptFullName + "\" ");
}
// If required program path provided and it exists
else if (oArgsNamed.Exists("path")) // &&
oFileSystem.FileExists(oArgsNamed.Item("path")))
{
// if user option specified use HKCU
if (oArgsNamed.Exists("user")) sRegKey = "HKCU";

// Set safer level parameter passed
if (oArgsNamed.Exists("level"))
{
// Populate dictionary
oLevelDictionary.Add("trust", "262144");
oLevelDictionary.Add("trusted", "262144");
oLevelDictionary.Add("limit", "131072");
oLevelDictionary.Add("limited", "131072");
oLevelDictionary.Add("constrain", "65536");
oLevelDictionary.Add("constrained", "65536");
oLevelDictionary.Add("untrust", "4096");
oLevelDictionary.Add("untrusted", "4096");
oLevelDictionary.Add("disallow", "0");
oLevelDictionary.Add("disallowed", "0");

// If entry exists for passed level then get its safer level id
if (oLevelDictionary.Exists(oArgsNamed.Item("level")))
{
sSaferLevel =
oLevelDictionary.Item(oArgsNamed.Item("level"));
}
}

// Build the registry value to track it for deleting
// Replace back slashes in program path with |
sDeleteRegValue = "HKLM\\Software\\SaferList\\" +
oArgsNamed.Item("path").replace(/\\/g, "|");

// Delete previous safer entry for the program
try
{
```

Safer

Safer

```
oShell.RegDelete(oShell.RegRead(sDeleteRegValue));
oShell.RegDelete(sDeleteRegValue);
if (oArgsNamed.Exists("delete")) oShell.Popup("Deleted
software restriction policy for path :\n\n" +
oArgsNamed.Item("path"),-1,"Safer", 48);
}
catch (error)
{
if (oArgsNamed.Exists("delete")) oShell.Popup("No software
restriction policy exists for path:\n\n" +
oArgsNamed.Item("path"),-1,"Safer", 48);
}

if (!oArgsNamed.Exists("delete"))
{
// Build the registry value
sRegValue = sRegKey +
"\\SOFTWARE\\Policies\\Microsoft\\Windows\\Safer\\CodeIdentifiers\\" +
sSaferLevel + "\\Paths\\" + sGUID + "\\";

// Write new entry for program
oShell.RegWrite(sDeleteRegValue, sRegValue, "REG_SZ");

// Write safer entry for program
oShell.RegWrite(sRegValue + "ItemData",
oArgsNamed.Item("path"), "REG_SZ");

oShell.Popup("\nAdded new software restriction policy for
path:\n\n" + oArgsNamed.Item("path"),-1,"Safer", 64);
}
}
else
{
oArgs.ShowUsage();
}
}
catch (error)
{
oShell.Popup("Runtime error : " + error, -1, "Safer", 16);
}

</script>
</job>

<job id="interactive">

<script language="VBScript">
Function WSHInputBox(Message, Title, Value)
WSHInputBox = InputBox(Message, Title, Value)
End Function
</script>
```

Safer

```
<script language="JScript">

var oShell = WScript.CreateObject("WScript.Shell")
var sCommandLine = "\"" + WScript.FullName + "\" \"\" +
WScript.ScriptFullName + "\" ";
var sInput = "";

try
{
sInput = WSHInputBox("Enter path to set restriction policy for :",
"Safer Path", "");
if (sInput == null) sInput = "";

if (sInput != "")
{
sCommandLine += " /path:\"\" + sInput + "\"\"";

sInput = "";
sInput = WSHInputBox("Enter additional options :\n\n/level – set
restriction level\n/user – set user policy\n/delete – delete policy",
"Safer Options", "");
if (sInput == null) sInput = "";

if (sInput != "")
{
sCommandLine += " " + sInput;
}
}

//WScript.Echo(sCommandLine);
oShell.Run(sCommandLine);
}
catch (error)
{
oShell.Popup("Runtime error : " + error, -1, "Safer", 16);
}

</script>

</job>
</package>
```

--

.