

# Re: Script to Rename Computer Name in Domain

---

*Source:*

<http://www.tech-archive.net/Archive/Windows/microsoft.public.windows.server.scripting/2009-07/msg00024.html>

---

- *From:* "Mark D. MacLachlan" <[markdmac@xxxxxxxxx](mailto:markdmac@xxxxxxxxx)>
  - *Date:* Wed, 08 Jul 2009 12:16:06 -0700
- 

Richard Mueller [MVP] wrote:

Skepper wrote:

i looking a script to rename computer name in domain server 2003  
something in VB + MessageBox [Enter New Computer Name]  
include user name + password to change it (domain remember ...)

To rename a computer you bind to the parent OU/Container of the computer object in AD and use the MoveHere method. See this link:

<http://www.microsoft.com/technet/scriptcenter/scripts/ad/computer/cptrvb13.mspx>

This changes the "Common Name" of the object. If you want to rename the local computer (change the NetBIOS name), you must run a script must run a script locally at the computer that uses WMI. See this link:

<http://www.microsoft.com/technet/scriptcenter/scripts/ad/computer/cptrvb14.mspx>

In either case, you can prompt for the new name with an InputBox statement in VBScript. I assume you intend the script to be run at the computer in question. Otherwise, you would need to also prompt for the current name. Assuming the script is run locally, and you are changing the Common Name, the VBScript program could be similar to:  
===== Option Explicit

## Re: Script to Rename Computer Name in Domain

```
Dim objSysInfo, strComputerDN, objComputer  
Dim strNewName, strParentAdsPath, objParent
```

```
' Prompt for new computer name.  
strNewName = InputBox("Enter new computer name")
```

```
' Retrieve local computer Distinguished Name.  
Set objSysInfo = CreateObject("ADSystemInfo")  
strComputerDN = objSysInfo.ComputerName
```

```
' Bind to the local computer object.  
Set objComputer = GetObject("LDAP:// & strComputerDN")
```

```
' Bind to the parent OU/container of computer object.  
strParentAdsPath = objComputer.Parent  
Set objParent = GetObject(strParentAdsPath)
```

```
' Rename the computer object.  
objParent.MoveHere objComputer.AdsPath, "cn=" & strNewName)
```

=====

The above assumes the user has permissions to rename the computer object. If not, you can use alternate credentials. I would recommend prompting for the credentials, rather than hard coding a password in the script. For example, you would need the additional code below (assumes you have already assigned a value to variable strComputerDN): ===== Option Explicit

```
Dim objSysInfo, strComputerDN, objComputer  
Dim strNewName, strParentAdsPath, objParent  
Dim strUser, strPassword, objNS
```

```
Const ADS_SECURE_AUTHENTICATION = &H1  
Const ADS_SERVER_BIND = &H200
```

```
' Prompt for new computer name.  
strNewName = InputBox("Enter new computer name")
```

```
' Prompt for administrator user name and password.  
strUser = InputBox("Enter admin user name in form Domain\AdmName")  
strPassword = InputBox("Enter password")
```

```
' Retrieve local computer Distinguished Name.  
Set objSysInfo = CreateObject("ADSystemInfo")  
strComputerDN = objSysInfo.ComputerName
```

```
' Bind to the local computer object.  
Set objComputer = GetObject("LDAP:// & strComputerDN")
```

```
' Bind to the parent OU/container of computer object using alternate  
credentials. strParentAdsPath = objComputer.Parent  
Set objNS = GetObject("LDAP:")
```

Re: Script to Rename Computer Name in Domain

Set objParent = objNS.OpenDSObject(strParentAdsPath, strUser, strPassword, ADS\_SERVER\_BIND Or ADS\_SECURE\_AUTHENTICATION)

' Rename the computer object.

objParent.MoveHere objComputer.AdsPath, "cn=" & strNewName)

=====

The user name can be in the format

<NT name of user>\<NetBIOS name of domain>

where the NT name of the user is the "pre-Windows 2000 logon name" (sAMAccountName). Or it can be the Distinguished Name of the user, or it can be the userPrincipalName, such as "jsmith@xxxxxxxxxxxxx".

If you want to change the NetBIOS name of the computer (per the second link above) using WMI, you would provide alternate credentials using the SWbemLocator object. See this link:

[http://www.microsoft.com/technet/scriptcenter/guide/sas\\_wmi\\_ciga.msp](http://www.microsoft.com/technet/scriptcenter/guide/sas_wmi_ciga.msp)

I think a much easier method is to use the Netdom resource kit utility which is easily scripted and can be done remotely.

--

.