```
#################################################################################
#
# Name:                 Add2Admin.ps1
# Author:               Jarvis Davis
# Company:              Campus Crusade for Christ
# Creation Date:  April 2, 2008
#
# Purpose:              To quickly and easily add/remove a domain user to/from
#                       the local administrators group on a computer
#
# Inputs:         It accepts the first four strings after the script name
#                       and puts them into variables
#
# Usage:          add2admin.ps1 username domainname computername action
#                       Actions are: Add and Remove. If no action is specified
#                       the script does not modify the group.
#
# Note:                 In an environment with one domain, you could modify the
#                       script to hardcode in the domain and shift the $args
#                       that are accepted to fit the variables.
#
# Acknowledgements:     Portions of this script were originally posted on the
#                       following websites. A big thanks to the original authors!
#
#       http://myitforum.com/cs2/blogs/yli628/archive/2007/08/30/powershell-script-to-add-remove-
#                       a-domain-user-to-the-local-administrators-group-on-a-remote-machine.aspx
#       http://keithhill.spaces.live.com/blog/cns!5A8D2641E0963A97!676.entry
#       http://www.microsoft.com/technet/scriptcenter/resources/qanda/mar08/hey0311.mspx
#
#################################################################################

$username = $args[0]
$domain = $args[1]
$strComputer = $args[2]
$action = $args[3]

function IsNullOrEmpty($str)
        {
        if ($str)
        {""}
        else
        {"Please specify parameters for the script."
        "Usage: [Add2Admin.ps1 Username UserDomain ComputerName Action (Add/Delete)]."
        ""
        break
        }
        }
# Check to make sure that at least three arguments were passed via the command line.
IsNullOrEmpty $strComputer
#IsNullOrEmpty $username
#IsNullOrEmpty $domain

$computer = [ADSI]("WinNT://" + $strComputer + ",computer")
$computer.name

$Group = $computer.psbase.children.find("administrators")

# This will list what's currently in Administrator Group so you can verify the result
```

```
function ListAdministrators
{$members= $Group.psbase.invoke("Members") | %{$_.GetType().InvokeMember("Name", 'GetProperty', $null, $_, $null)}
$members}

ListAdministrators
""

# Depending on the value of the $action variable ($args[3]), either add/remove/do nothing to the admins group

if ($action -eq "Add")
	{$Group.Add("WinNT://" + $domain + "/" + $username)}
elseif ($action -eq "Remove")
	{$Group.Remove("WinNT://" + $domain + "/" + $username)}
else
	{"No action was specified, so no action was taken."
	"Usage: [Add2Admin.ps1 Username UserDomain ComputerName Action (Add/Delete)]."
	""
	}
ListAdministrators
""
```