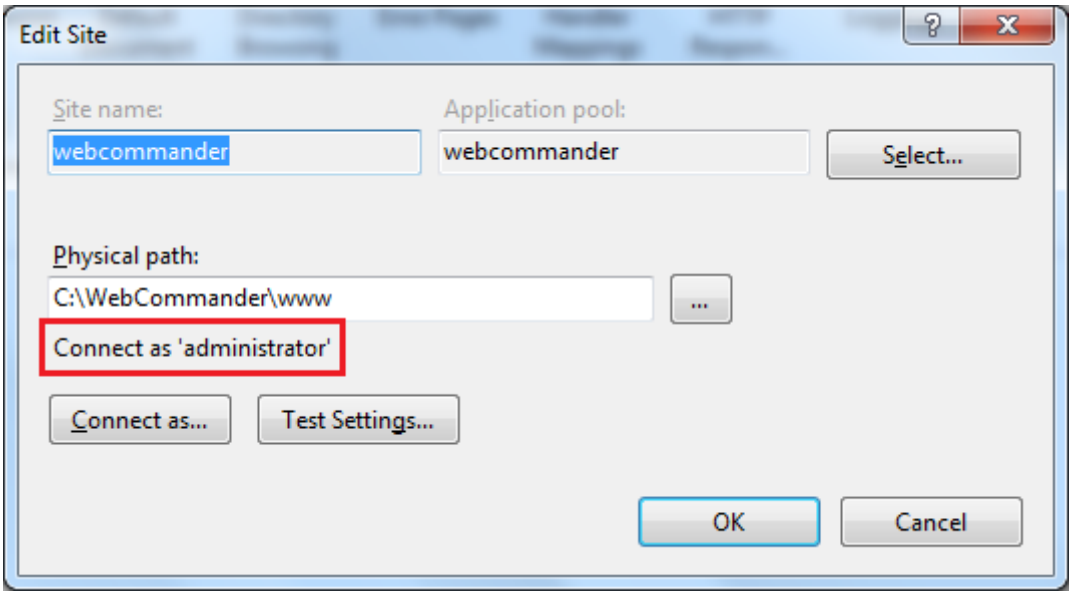
webCommander server installation and configuration guide

1. Install Windows Server 2012 on a VM or a physical machine by following this guide.
2. Install IIS on the server by following this guide. Make sure IIS Management Service is also installed.
3. Download PHP web installer from http://php.iis.net/.
4. Run the installer with default options.
5. Download the latest version of PowerCLI installer from VMware website.
6. Run the installer to install all required components.
7. If the installer requires dotNet 3.5, add it to the server by following this guide and then re-run the installer.
8. Download webCommander package from VMware Fling website.
9. Extract the package to c:\webcommander on the server
10. Open Powershell console (x86) and execute the following commands
    ```
    set-executionpolicy unrestricted -force
    c:\webcommander\setup.ps1
    ```
11. Make sure to connect as administrator in the basic settings of the web site.



When communicating with Windows machine, webCommander requires Powershell V2 or newer version is installed on the remote machine. For Windows XP, 2003, Vista and 2008 where Powershell is not built-in, webCommander automatically installs Powershell v2. Please also note that dotNet 2.0 with SP1 is also installed on Windows XP and 2003 because it's the pre-requisite of Powershell.

Neither Powershell or dotNet installers are included in webCommander package, please download them from Miscrosoft website and put them at the following locations:

postinstall\dotnet\NetFx20SP1_x64.exe
postinstall\dotnet\NetFx20SP1_x86.exe

postinstall\powershell\AMD64-all-windows6.0-kb968930-x64.msu
postinstall\powershell\AMD64-en-windowsserver2003-kb968930-x64-eng.exe
postinstall\powershell\WindowsServer2003.WindowsXP-KB926139-v2-x64-ENU.exe
postinstall\powershell\X86-all-windows6.0-kb968930-x86.msu
postinstall\powershell\X86-en-windowsserver2003-kb968930-x86-eng.exe
postinstall\powershell\X86-en-windowsxp-kb968930-x86-eng.exe

How to add a script

1. Edit c:\webcommander\www\webcmd.xml by adding the following lines of code:

```
<command name="myCommand" description="my command description">
    <script>myScriptName</script>
    <parameters>
        <parameter optional="0" name="parameter1" description="description1" />
        <parameter optional="0" name="parameter2" description="description2" />
        <parameter optional="0" name="parameter3" description="description3" />
        ...
    </parameters>
</command>
```

   Command name must be unique.

   Script value must be identical to the name of the Powershell script.

   All parameters the script takes must be defined.

   If a parameter is optional, set optional="0" otherwise optional="1"

   A parameter could be of one of the following 5 types:

   1. Text which is the default type
   2. Password
   3. File which is used to upload a file
   4. Option which is selectable from a drop down list
   5. SelectText which is similar to Option but allow inputting a custom value

2. Save and close webcmd.xml.
3. Put the Powershell script namely myScriptName.ps1 into c:\webcommander\powershell\.
4. Open myScriptName.ps1 and add the green lines right after parameter section

```
Param (
    $parameter1,
    $parameter2,
    $parameter3
)
foreach ($paramKey in $psboundparameters.keys) {
    $oldValue = $psboundparameters.item($paramKey)
    $newValue = [system.web.httputility]::urldecode("$oldValue")
    set-variable -name $paramKey -value $newValue
}
```

   This is because webcmd.php encodes those parameters to handle special characters, so we must decode them back. Please note that there is no need to escape Powershell special characters (such as white space and $) when input parameters on webCommander web pages.

How to remove a script

Simply remove or rename the Powershell script at c:\webcommander\powershell\. There is no need to remove the corresponding codes from webcmd.xml. When webcmd.php parses the xml file, it does not display a command of which the named Powershell script could not be found.

How to store the credentials in a script

webCommander stores a default password in an environment variable named defaultPassword. When setup.ps1 of webCommander is executed, the user is required to input a string to be used as the default password. Then the following line is added into Powershell profile:
$env:defaultPassoword=<what the user input>
Consequently, whenever a Powershell session is created, the environment variable is set. Powershell scripts can access the variable as described below

```
Param (
    ...
    $serverUser="root",
    $serverPassword=$env:defaultPassword,
    ...
    $guestUser=".\administrator",
    $guestPassword=$env:defaultPassword,
    ...
)
```

webCommander defines "root" as the default user for ESX or VC.
webCommander defines ".\administrator" as the default user for VM guest OS.