**How To Run Chapter 7 Example Code**

**PowerShell Scripts**

PdfPrint.ps1

This script provides a way to print a pdf file on disk to an IPP within Windows PowerShell.

1. Start Windows Powershell ISE located in Windows Tools.
2. In Windows Tools, open Windows PowerShell ISE
3. In PowerShell ISE, open \Code\PowerShell\PdfPrint.ps1
4. You will need a pdf file to print and an IPP printer to print it on.
5. Syntax: PdfPrint.psi <pdf file to print> <IPP printer>
6. Example: assuming you have navigated to the folder where the. psi script file exits, **.\PdfPrint.ps1 "C:\Temp\Test.pdf" HPM528** should work.

**.Net Utilities**

RmLocalPrinters

Utility to remove local (direct), non-IPP printers from remote machines before or after converting to IPP/IPPS. You will need appropriate rights on the remote computers to do so of you will receive an “Access-Denied” (5) error.

1. Navigate to: \Code\RmLocalPrinters\RmLocalPrinters\bin\Release
2. Open a Console Window.
3. Usage: rmLocalPrinters /c=<target\_computer> (this will show potential printers to remove)
4. Example: assuming you have navigated to the folder where the RmLocalPrinters utility exits, **rmLocalPrinters /c=SR36001A4**
5. You can use the “Run-As” command to open the Consoile Windows to gain appropriate rights over the target machine(s)
6. This utility requires that WMI be up and running on the remote machine, and the spooler service running.
7. There are optional switches that can be used with this utility. The /r switch will remove local, non-IPP printers – **each one will ask the user to reply yes or no before removing a printer identified.** The /l switch places logging information about run events into the RmLocalPrinters.log file – this is always placed in the user **Documents** folder – entries are appended.

AddSharedIppPrinters

Another utility to help upgrade computers to IPP/IPPS print. This utility queries the named print server and installs IPP print connections on the client machine.

1. Navigate to: \Code\AddSharedIppPrinters\AddSharedIppPrinters\bin\Release

2. Open a Console Window.

3. Usage: AddSharedIppPrinters </ps=print\_server\_name>

4. Optional parameters are either /purge (removes existing print connections first) or /pl=<Location string(s)> filter.

5. Example: assuming you have navigated to the folder where the AddSharedIppPrinters utility exits,

**AddSharedIppPrinters /ps=print\_server**

**or**

**AddSharedIppPrinters /ps=print\_server /purge**

**AddIppPrinter**

This utility can add local IPP/IPPS printers on a workstation or server. You will have to know the DNS name or IP address of the printer(s) to add and whether they should be IPP (port 631) or IPPS (port 443). This utility is included in the code section of chapter 2.

1. Navigate to AddIppPrinter\bin\Release.
2. Open a Console Window.
3. AddIppPrinter /p=<printer IP address> /s=<631 or 443>
4. Example: AddIppPrinter /p=101.34.44.56 /s=631
5. Ensure printers have IPP/IPPS enabled.
6. If the computer has a FQDN, you can use printer name instead of IP address.

Note: If you want to install several IPP/IPPS local printers at once, you can use the /l switch. In this mode, the command line should follow the following syntax:

AddIppPrinter /p=<printer IP address> /l=<file list of printer DNS names> /s=631 or 443>

* In list mode, the type (IPP/IPPS) chosen on the command line will be used by all the printers created from the file.

**CheckWppEnv**

This utility monitors the state of WPP on remote computers – i.e. is WPP enabled or disabled. It displays all IPP printers found on the remote machine as well. Provided the caller has sufficient rights, it can also enable or disable WPP on the remote machine as well.

1. Navigate to CheckWppEnv\bin\Release.

2. Open a Console Window.

3. CheckWppEnv /c=<computer>

4. Example: CheckWppEnv /c=<computer\_to\_monitor>

Note: This utility can also enable or disable WPP

Registry Scripts

Use WPP\_Enable.reg on machines to enable WPP

Use WPP\_Disable on machine to disable WPP