



Author: Brian Marshall
Updated: 6/1/2017
Version: 1.0



Table of Contents

1	Introduction.....	3
1.1	Application Description.....	3
1.2	Requirements.....	3
1.3	Essbase Artifacts.....	4
1.4	Data Artifacts.....	4
1.5	Automation Artifacts.....	4
2	Installation.....	5
2.1	Essbase Creation.....	5
2.2	Automation Creation.....	10
3	Benchmark.....	12
3.1	Option 1.....	13
3.2	Option 2.....	13



1 Introduction

The purpose of this document is to provide a definitive set of benchmarks related to the hardware and software factors that impact the performance of Oracle's Hyperion Essbase database engine. The goal of this document is to assist both implementers and administrators of Essbase with decisions related to hardware and software selection. As an example, it's known that Essbase performs better on physical hardware than virtual hardware, but concrete testing of like applications isn't readily available. This document will change that.

1.1 Application Description

The EssBench application is an Essbase BSO application creating using fictional meta-data and data. The application is intended to mimic a multi-billion dollar retail chain with many stores and many products. This produces two large sparse dimensions. With only Income Statement data in the cube, the account dimension is reasonably sized. The full statistics are shown in the table below:

Dimension	Type	Members in Dimension	Members Stored
Account	Dense	1025	838
Period	Dense	19	14
Years	Sparse	6	6
Scenario	Sparse	3	3
Version	Sparse	4	4
Currency	Sparse	3	3
Entity	Sparse	8767	8709
Product	Sparse	8639	8639
EntityCurrency	Sparse (Attribute)	76	0

1.2 Requirements

The EssBench application is designed to utilize systems of many sizes. As a result, the settings are not designed for maximum performance, but instead for maximum compatibility. The hardware and software requirements for EssBench are found in the table below:

Item	Requirement
CPU	8 Logical Cores
Memory	16GB
Disk Space	20GB
Essbase Version	11.1.2.4+



1.3 Essbase Artifacts

In addition to the outline file, EssBench also includes one data load rule and four calculation scripts. The details of these artifacts are found in the table below:

Artifact	Type	Description
dtOPEX	Load Rule	Used for loading a CSV file containing OPEX data.
AggAct	Calculation Script	Used to aggregate the product and entity dimensions.
AggAlloc	Calculation Script	Used to aggregate the specific allocation account for the product and entity dimensions.
AllocP	Calculation Script	Used to allocate high-level values to the product and entity dimensions using a FIXPARALLEL statement.
Curr	Calculation Script	Used to convert USD back to local currency.

1.4 Data Artifacts

EssBench includes standard data sets along with the included Essbase artifacts. The details of these artifacts are found in the table below:

Artifact	Description
dtOPEX.csv	Comma separated values containing operating expenses for the fictional retailer. Used in conjunction with the data load rule dtOPEX .
EssBenchLO_*	Eight native Essbase format files containing level 0 data. Loaded directly without the need of a data load rule.

1.5 Automation Artifacts

The final component included with EssBench is the automation of the benchmark itself. This benchmark is performed using a PowerShell script and an MaxL script. The details of these artifacts are found in the table below:

Artifact	Description
EssBench.ps1	PowerShell script used to define variables necessary for the MaxL execution along with the execution and logging of the benchmark.
EssBench.msh	MaxL script used to execute the specific tests included with the benchmark.



2 Installation

The installation of EssBench is a manual process. There are two main parts to the installation. First, the creation of the Essbase application. Second, the creation of the automation process. Each part of the installation is provided in a ZIP file.

2.1 Essbase Configuration

Ensure that the Essbase.cfg is configured to include the following (modify for your preferences):

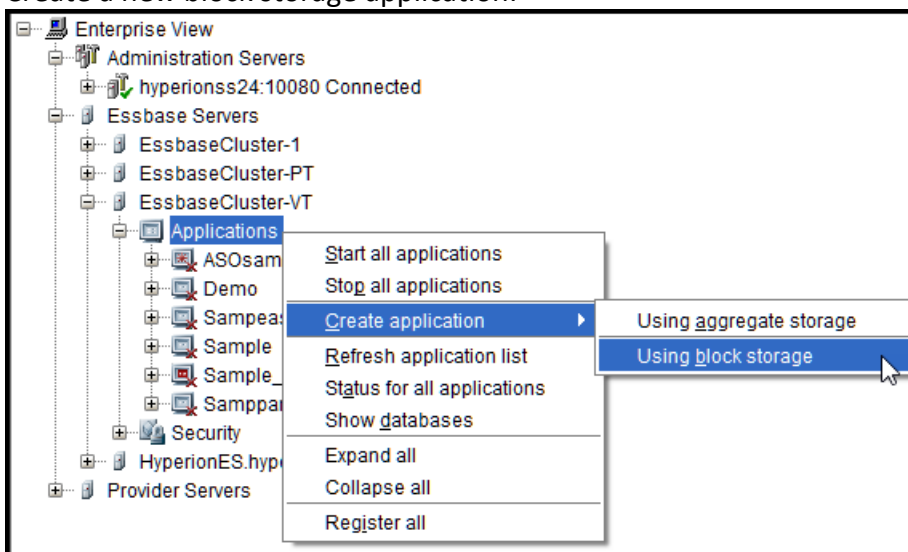
```
WORKERTHREADS 50
SERVERTHREADS 100
AGENTTHREADS 25

CALCPARALLEL 16
RESTRUCTURETHREADS 16
EXPORTTHREADS 16
```

Other settings in the config could throw off the test results for certain tests, so if you can remove other settings for testing purposes, that would be best. When submitting results, please note any other settings in the config file so that we can note that in the result set.

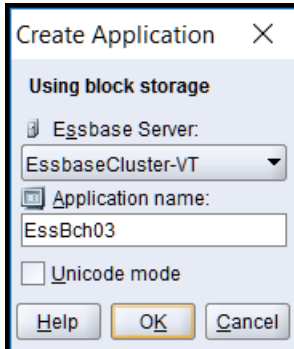
2.2 Essbase Creation

1. Create a new block storage application:

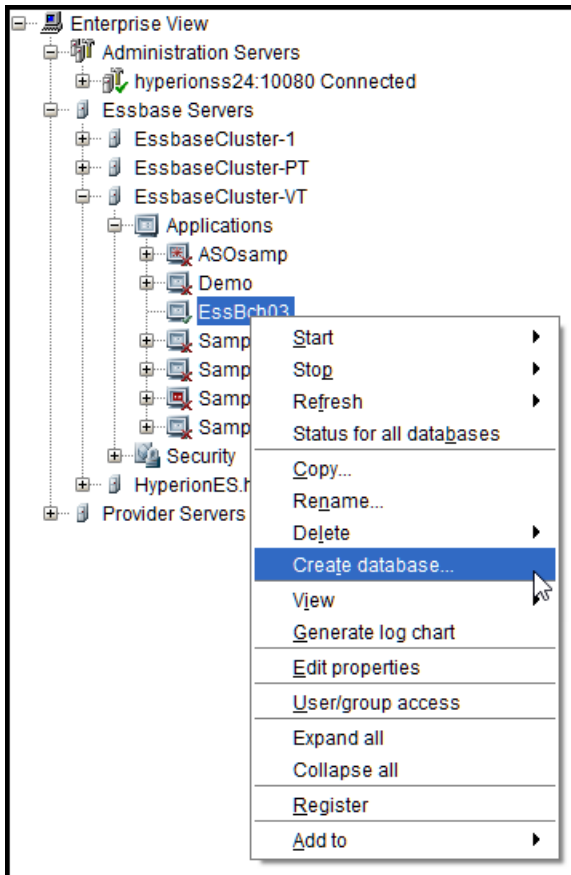




2. The application can be named anything:

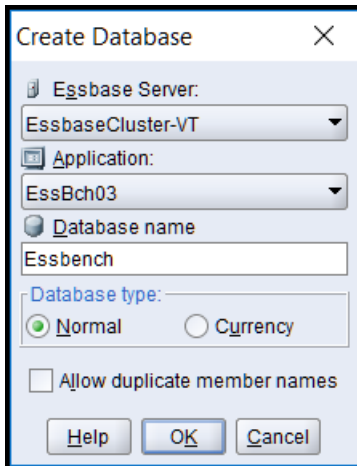


3. Create a new database:

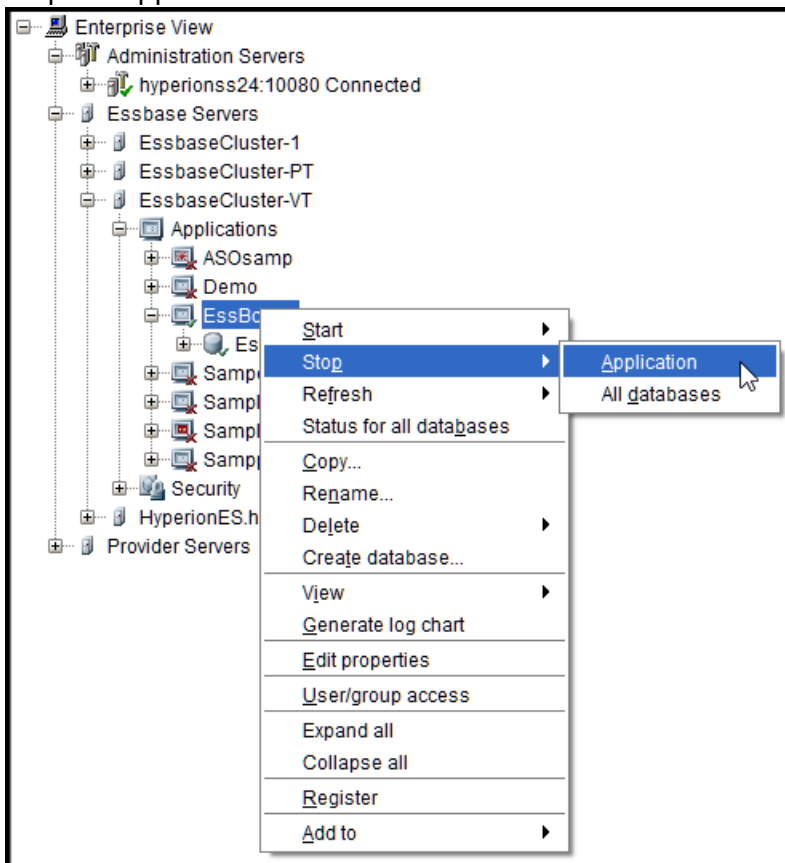




4. Name the database **Essbench**:



5. Stop the application:

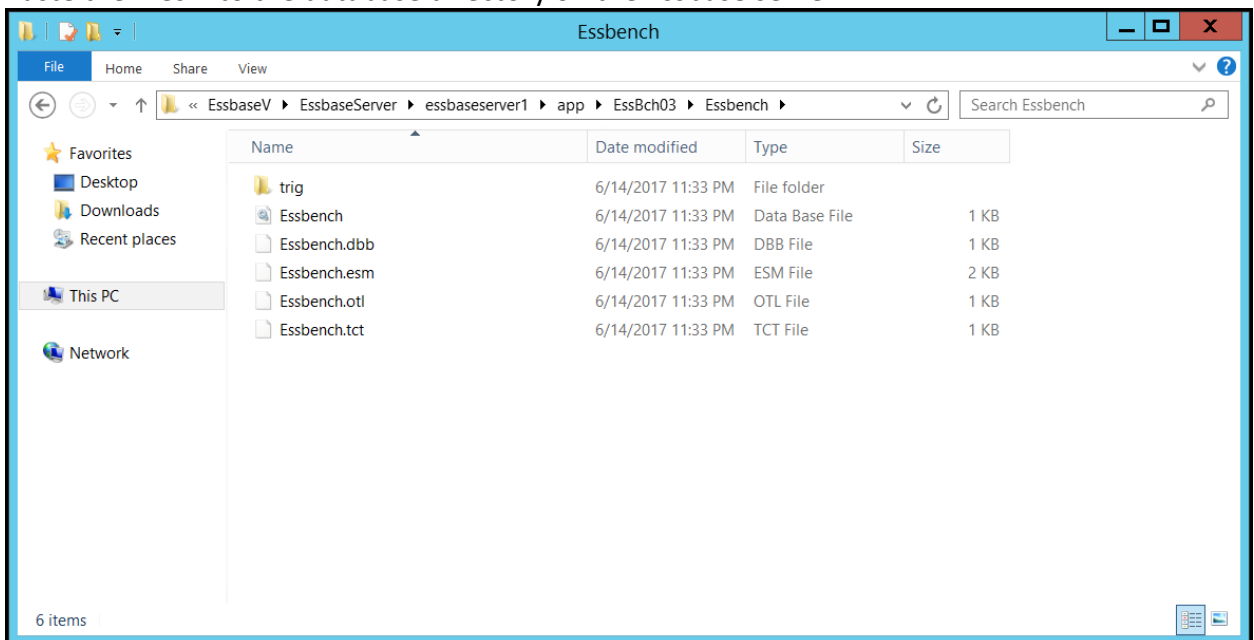




6. Copy the files included in the Essbase artifacts ZIP file:

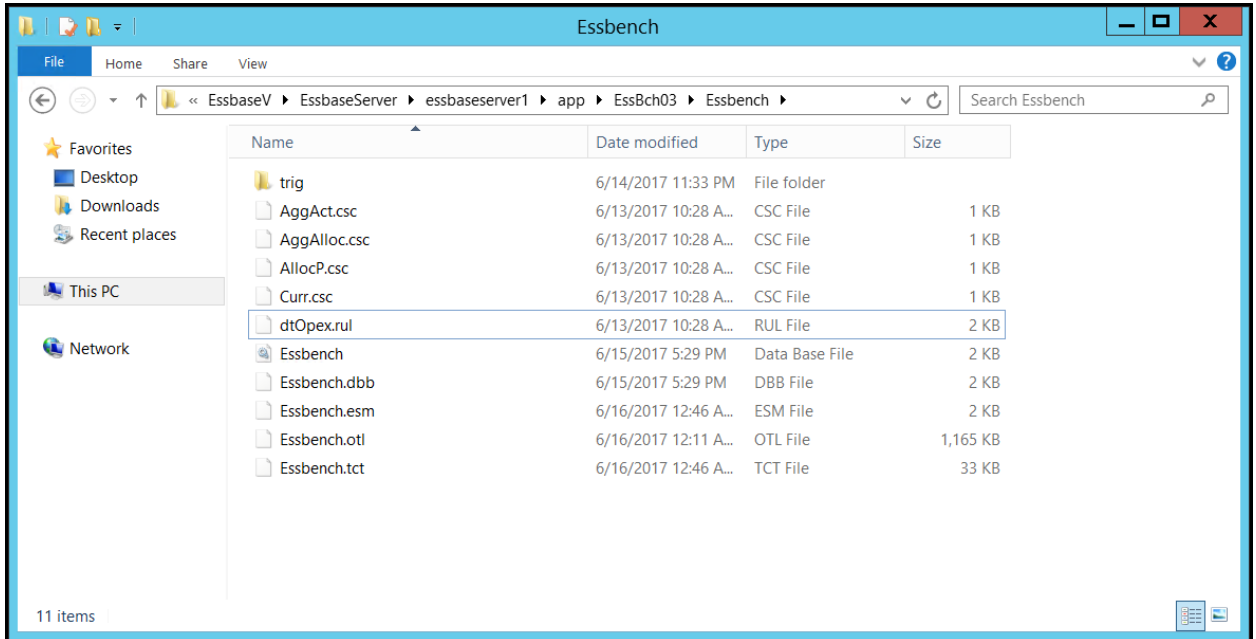
Name	Date modified	Type	Size
AggAct.csc	6/13/2017 10:28 A...	CSC File	1 KB
AggAlloc.csc	6/13/2017 10:28 A...	CSC File	1 KB
AllocP.csc	6/13/2017 10:28 A...	CSC File	1 KB
Curr.csc	6/13/2017 10:28 A...	CSC File	1 KB
dtOpex.rul	6/13/2017 10:28 A...	RUL File	2 KB
EssBench.otl	6/13/2017 6:41 PM	OTL File	1,165 KB

7. Paste the files into the database directory on the Essbase server:

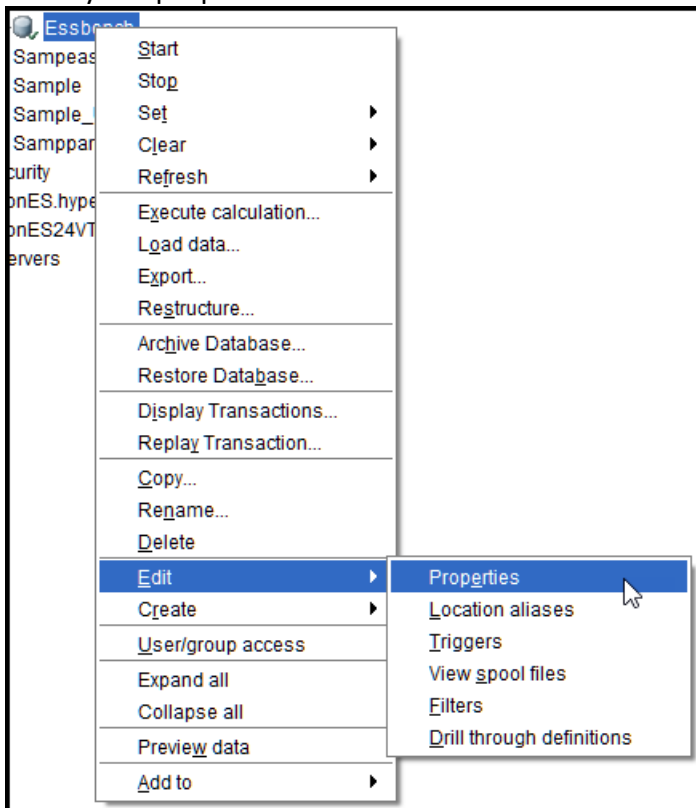




8. The folder should look like this:

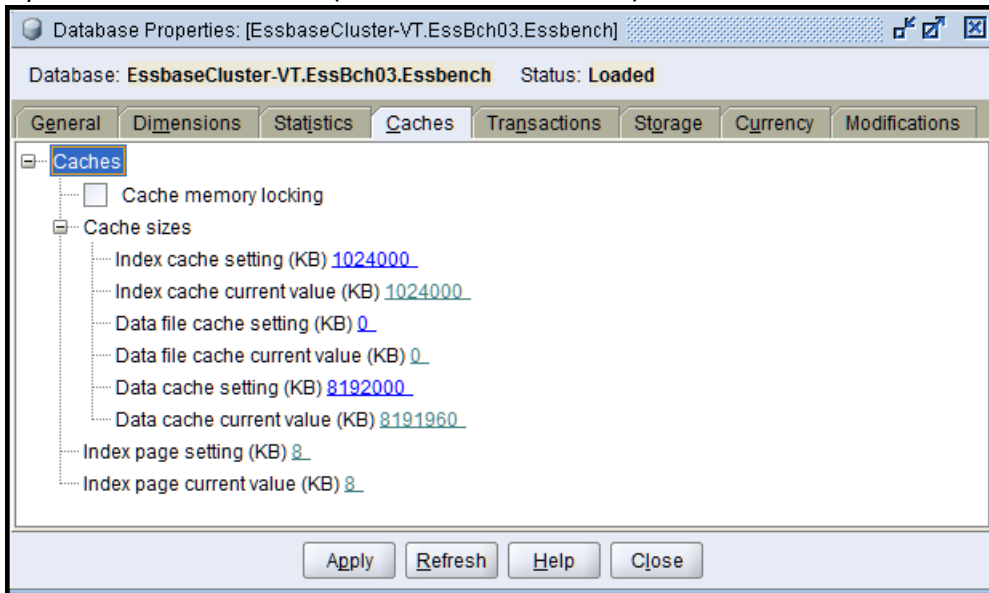


9. Modify the properties of the Essbench database in EAS:





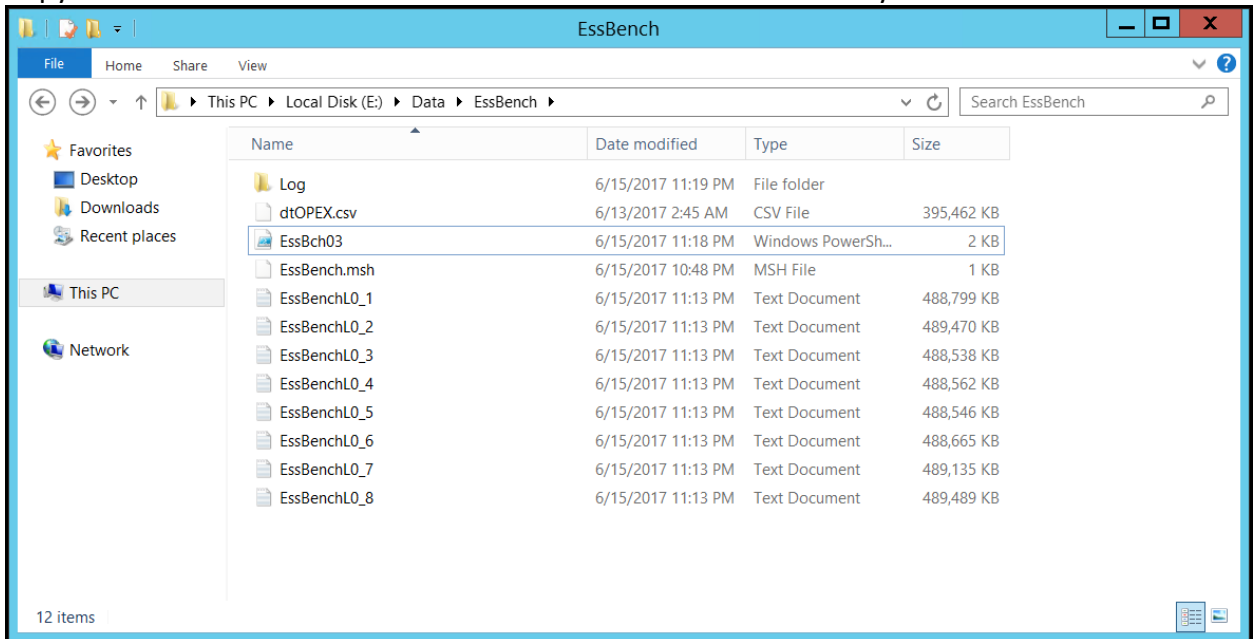
10. Update the cache values (1024000 and 8192000):



11. Apply the settings and restart the Essbase application.

2.3 Automation Creation

1. Copy the contents of the EssBench automation ZIP file to a directory on Essbase server:





2. Update the variable assignment section of the PowerShell file:

```
1 #####
2 #Created By: Brian Marshall
3 #Created Date: 6/1/2017
4 #Purpose: Essbase Performance Benchmark (EssBench)
5 #####
6
7 #####
8 #Variable Assignment
9 #####
10
11 $LogPath = "E:\Data\EssBench\Log\"
12 $MaxLPath = "E:\Oracle\Middleware\user_projects\EssbaseV\EssbaseServer\essbaseserver1\bin"
13 $MaxLUsername = "admin"
14 $MaxLPassword = " "
15 $MaxLServer = "hyperiones24vt.hyperion.local"
16 $EssbaseApp = "EssBch03"
17 $FilePath = "e:\data\EssBench\"
18 $TimeStamp = Get-Date -format "yyyyMMddHHmm"
19 $LogFileName = $LogPath + $EssbaseApp + $TimeStamp + ".log"
20
21 Start-Transcript -path $LogFileName
22
23 #####
24 #MaxL Execution - Pass 1
25 #####
26
27 & $MaxLPath\StartMaxL.bat EssBench.msh $MaxLUsername $MaxLPassword $MaxLServer $EssbaseApp $FilePath
28
29 #####
30 #MaxL Execution - Pass 2
31 #####
32
33 & $MaxLPath\StartMaxL.bat EssBench.msh $MaxLUsername $MaxLPassword $MaxLServer $EssbaseApp $FilePath
34
35 #####
36 #MaxL Execution - Pass 3
37 #####
38
39 & $MaxLPath\StartMaxL.bat EssBench.msh $MaxLUsername $MaxLPassword $MaxLServer $EssbaseApp $FilePath
40
41 Stop-Transcript
```



3 Benchmark

Now that everything is configured, the benchmark can be executed. Simply start the PowerShell script. Before we can actually execute a PowerShell script, we need to allow for remote signed scripts. Open up Windows PowerShell as an administrator and execute this command:

Set-ExecutionPolicy RemoteSigned

It should look something like this:

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Users\hypserservice> Set-ExecutionPolicy RemoteSigned

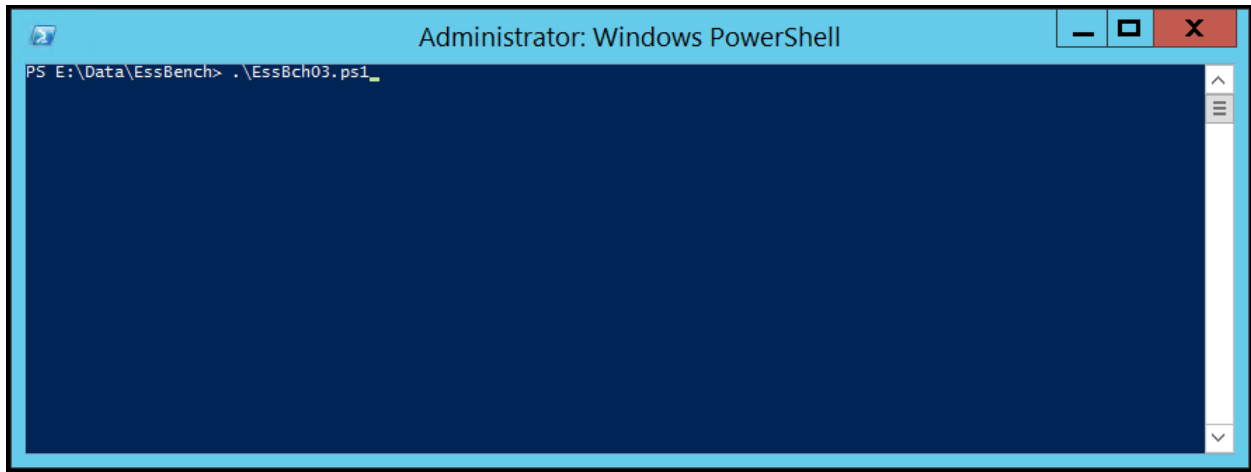
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
PS C:\Users\hypserservice> _
```



Once this command has executed successfully, there are then two options to execute the benchmark:

3.1 Option 1

Open Windows PowerShell and change to the directory containing the benchmark. Execute the benchmark:



3.2 Option 2

Right-click and run with PowerShell:

