SE Consolidation Data Processing

Formerly Known as

Employee Readiness Program

Consolidation PROCESS

Information in this document is subject to change without notice. The names of companies, products, people, characters, and /or data mentioned here in are fictitious and are in no way intended to represent any real individual, company, or event, unless otherwise noted. Complying, with all applicable copyright laws is the responsibility of the user. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Scalability Experts Inc. If however, your only means of access is electronic, permission to print one copy is hereby granted.

Scalability Experts Inc. may have patents, patent applications, trademarks, copyright, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Scalability Experts Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyright, or intellectual property.

**©2010 Scalability Experts Inc. All rights reserved.**

Contents

Contents

[Introduction 4](#_Toc327284119)

[Description 5](#_Toc327284120)

[Program Pre-Requisites 5](#_Toc327284121)

[Manager and PMO Approval 5](#_Toc327284122)

[SQL Server Basic Knowledge 5](#_Toc327284123)

[**DAY 1 – Understanding Phases of Consolidation Process** 6](#_Toc327284124)

[**PHASE 1: Understand the phases of the consolidation engagement** 6](#_Toc327284125)

[**PHASE 2: Understand the Consolidation Methodology** 6](#_Toc327284126)

[**PHASE 3: Understand the Resources and Tools used for Consolidation engagement** 6](#_Toc327284127)

[**PHASE 4: Understand the Consolidation strategies** 7](#_Toc327284128)

[**PHASE 5: Understand the Analysis process** 7](#_Toc327284129)

[**Deliverables** 7](#_Toc327284130)

[**DAY 2 – Data Collection Process** 8](#_Toc327284131)

[**Deliverables** 8](#_Toc327284132)

[**DAY 3 – Analysis** 9](#_Toc327284133)

[**PHASE 1: Understand the Data Sent by SE Consultant** 9](#_Toc327284134)

[**PHASE 2: Understand There are Subtle, but Important Differences to be Aware Of** 10](#_Toc327284135)

[**Deliverables** 10](#_Toc327284136)

[**DAY 4 – PerfMon Data Loading** 11](#_Toc327284137)

[**PHASE 1: Create Database to Store the PerfMon Data** 11](#_Toc327284138)

[**PHASE 2: Creating DSN Connection** 12](#_Toc327284139)

[**PHASE 3: Uploading PerfMon Data Into SQL Server Tables with RELOG** 14](#_Toc327284140)

[**PHASE 4: Alter Column Data Types, Indexing, Adding CPU Factor Table, & PowerPivot Views** 15](#_Toc327284141)

[**PHASE 5: Import CPU “FACTOR” from an Excel Spreadsheet** 15](#_Toc327284142)

[**DAY 5 – PowerShell’s Windows & SQL Server Meta Data Loading** 16](#_Toc327284143)

[**PHASE 1: Create a database in SQL Instance** 16](#_Toc327284144)

[**PHASE 2: Loading PowerShell Data using SSIS Package** 16](#_Toc327284145)

[**PHASE 3: verify the Variable lists are visible** 16](#_Toc327284146)

[**PHASE 4**: **Executing the SSIS Package** 18](#_Toc327284147)

[**PHASE 5**: **Creating Views for PowerPivot Analysis** 18](#_Toc327284148)

[**Deliverables** 18](#_Toc327284149)

[**DAY 6 – PerfMon Analysis (SES)** 19](#_Toc327284150)

[**PHASE 1: Processing PerfMon Data with the Unified Script** 19](#_Toc327284151)

[**PHASE 2: Placing Results Into Spreadsheets** 19](#_Toc327284152)

[**Tiering the Servers** 20](#_Toc327284153)

[**CPU\_Analysis** 21](#_Toc327284154)

[**Deliverables** 22](#_Toc327284155)

[**DAY 7 – Analysis of the PowerShell Server Meta Data** 23](#_Toc327284156)

[**Memory Analysis** 23](#_Toc327284157)

[**Database Analysis** 23](#_Toc327284158)

[**Server List** 23](#_Toc327284159)

[**Data File Info Analysis** 24](#_Toc327284160)

[**Log File Info Analysis** 24](#_Toc327284161)

[**Physical Disk Info Analysis** 24](#_Toc327284162)

[**Deliverables** 24](#_Toc327284163)

[**Final actions after ALL the processing** 24](#_Toc327284164)

[**DAY 8 – PowerPivot Setup (new technique For Advanced Analysis)** 25](#_Toc327284165)

[**PerfMon Data Snowflake Schema** 26](#_Toc327284166)

[**Server Meta-Data Schema** 26](#_Toc327284167)

[Building PerfMon Data Into a PowerPivot Display 27](#_Toc327284168)

[Building PowerShell Data Into a PowerPivot Display 28](#_Toc327284169)

# Introduction

The Employee Readiness Program (ERP) at Scalability Experts is a program designed to train new hires on corporate policies, procedures, framework(s) and methodologies around all its service offerings. In addition, this program will also be used to cross-train existing employees on getting familiar and ready to deliver services to the customer that they have either not delivered before.

Since this is periodically updated to incorporate latest and greatest information and changes relating to a particular practice, it also allows employees to keep abreast with current methodologies within the organization.

# Description

This document will go through the details of corporate standard relating to the Consolidation Process. Although the ERP for this process will be one week long and will walk through the steps required to get acquainted with it, the actual time taken by the employee, to go through one iteration of the entire steps outlined in this document can vary based on prior knowledge and level of experience etc.

The following sections of the document are structured around a weekly program, with activities to be completed on each of the week long schedule.

### Program Pre-Requisites

Typically, once a consultant has gone through this program, there would not be a need to go through it again until the process has either been updated with additional details or the methodology has been modified as a result of change in tool and/or technology. Alternatively, a consultant may feel the need to skim through this document in a situation when a long period of time, 6 months or more since the last time the consultant had the opportunity to work on a project requiring this skill set.

Either way, each consultant who plans intends to go through this readiness program must complete the following pre-requisites prior to their initiation

### Manager and PMO Approval

If executed on a full-time basis, on an average, this program will take approximately one week to be completed in its entirety. Irrespective of whether a consultant is given a week off to complete this program or run through this on a time-permitting basis, there will be an impact to some extent on the project(s) assigned to the consultant as a result. It is therefore a mandate for the consultant to discuss and seek a written approval from the Consulting and the PMO managers.

### SQL Server Basic Knowledge

Prior to starting on this program, all fresh college graduate consultants must complete the ‘Introduction to SQL Server’ training program. Please refer to the **OLTP Readiness – Introduction to SQL Server** document for details.

## **DAY 1 – Understanding Phases of Consolidation Process**

**Objective**

At the end of the day-1, you will have learned the different phases of the consolidation process and the tools that could be used to perform the consolidation tasks. The Scalability Experts consolidation engagement in itself executed as a consulting project. In order to execute the consolidation engagement in a disciplined fashion with quality and also to ensure smooth transition among the consultants it is imperative that the consultant closely follow the Scalability Experts defined process.

**Tasks Description**

The following are the task descriptions

### **PHASE 1: Understand the phases of the consolidation engagement**

The consultant is required to understand each phase of the consolidation engagement, it is important so that the various divisions in Scalability Experts, namely the consulting, project management, sales and other divisions operate smoothly. The following are the consolidation engagement phases

* 1. Scope
  2. Data Collection
  3. Analysis (SES)
  4. Analysis (SE)
  5. Final Client Presentation
  6. Deployment

### **PHASE 2: Understand the Consolidation Methodology**

The Consolidation Methodology is a set of collection of methods (procedures or tasks) that are Scalability Experts best practices and/or Microsoft suggested best practices. These methods will help the consultant quickly leverage is past experience and SQL Server knowledge base outside of Scalability Experts to execute on the various consolidation tasks.

### **PHASE 3: Understand the Resources and Tools used for Consolidation engagement**

Scalability Experts make heavy use of automated tools and rely on internal and external resources to ensure that the consolidation engagement is delivered efficiently and incorporate the latest developments in the industry. This task will help you learn the tools used for consolidation engagement.

### **PHASE 4: Understand the Consolidation strategies**

The consultant needs to understand the Microsoft Consolidation Strategies, and their pro’s and con’s. This will help the consultant recommend an optimal strategy to the customer.

### **PHASE 5: Understand the Analysis process**

The most important phase of the consolidation engagement is the Analysis phase. It is the most important because this is the phase that defined the tasks associated with next phases and hence is also the phase that can truly identify the time and effort involved in the consolidation engagement. The candidate needs to learn the tasks in the Analysis phase since they are critical to the quality of the engagement.

### **Deliverables**

At the end of Day-1, the following deliverables must be completed

1. Review the following presentation SEConsolidation.ppt
2. Create a project execution plan for an consolidation engagement

The expectation is that the consultant will create a consolidation engagement presentation which explains the execution plan. The project execution plan should explain the steps that will be performed in the remainder of the engagement, as the consultant envisions. This task has minimal description in order to encourage independent thinking and creativity. However in the training session, it is suggested that each student presents his execution plan and other students play the role of the customer. The customer objective is to understand how the consolidation will be executed end-to-end and the student objective to prepare before-hand answers to all the questions anticipated.

## **DAY 2 – Data Collection Process**

**Objective**

At the end of day-2, you will have learnt the data collection process, which is the pre-requisite for the actual consolidation. Prior to executing on the analysis phase, the consultant is required to collect all data that will be required for the analysis phase. The data collected includes SQL Server version information (WMI Data), Consolidation incompatibilities, System’s PerfMon Data and SQL Server Configuration information and so on. This task is mostly carried out by consultant at SE Office. SES Consultant is sent the collected data to perform the analysis.

**Tasks Description**

The following are the task descriptions

1. Understand the data requirements for consolidation

As mentioned above this task will help the consultant identify all the data required for consolidation analysis. Typically the consolidation engagement starts with a customer/SE Consultant kick-off call discussing the engagement expectations and introductions. Once the consultant is introduced to the scope of the engagement, the consultant typically sends a list of data collection items and requirements to the database administrators prior to the start of the engagement. This helps the customer prepare for the engagement.

### **Deliverables**

At the end of Day-2, the following deliverables must be completed.

1. Create a list of all the data that needs to be collected for an consolidation engagement
2. Document how you will collect each piece of data

## **DAY 3 – Analysis**

**Objective**

At the end of day-3, you will have learned the process of initiating consolidation analysis by understanding the data and its mappings. The past two-days the students would have understood the consolidation process and the data that needs to be collected from the consolidation engagement. Today the student will learn how to analyze the data and the various mappings done during the Analysis process.

**Tasks Description**

The following are the task descriptions

### **PHASE 1: Understand the Data Sent by SE Consultant**

In this task the candidate is expected to understand at the minimum the following

1. Number of Servers in Scope
2. PerfMon Data for the same
3. Target Server Model
4. WMI/PowerShell Data
5. Client Questionnaire
6. Reports that need to be sent

* **PerfMon Data:** It is the information regarding the performance of the server. Perfmon Data is gathered by running it on the respective system for a minimum of 24 hours. A month of data collection is the best, but often is not possible due to various constraints.
* **PowerShell Data:** It is the information regarding the server configuration, databases, user and logins with their access rights etc. PowerShell Data is gathered by running it on the respective system during the non-peak hours. (Peak hours are those hours of the day when the usage of the various resources of the system is maximum.)
* This data is uploaded on the FTP site by the architect in zipped format.
* The SES consultant will download the data from the FTP site and will check for the latency of the data (i.e. PerfMon Data and PowerShell Data) is incomplete or corrupt.
* Data is also collected by asking the client to fill out a customer questionnaire which has information about the servers.

### **PHASE 2: Understand There are Subtle, but Important Differences to be Aware Of**

The most critical task of the consolidation process is the identification the data sent. Client environment can have clusters with logical/physical names. Identifying their mapping with the data present is crucial.

### **Deliverables**

At the end of Day-3, the following deliverables must be completed

1. Understand and Plan Quick ways to download the data
2. Review a set of dummy data
3. Review a dummy Client Questionnaire

Observe the practical complexity and the issues that are identified

* 1. Select some consolidation issues and identify their resolution
  2. We encourage an healthy discussion on how different ways to present the findings

## **DAY 4 – PerfMon Data Loading**

**Objective:** At the end of day-4, you will have learnt how to populate a database with the PerfMon data sent.

**Tasks Description:** PerfMon Data loading

After analyzing the data, if all the files are correct then SES Consultant can begin the data analyzing part. This includes the following steps:

### **PHASE 1: Create Database to Store the PerfMon Data**

Database creation recommendations: Create the CONSOLIDATIONDB database in **simple** recovery model. For file sizing you can use these sizes as a reference point.

625MB BLG

256MB CSV same PerfMon files just converted to csv.

Both the above files will make the database’s:

Data file after import and indexing grew to 7,900 MB

Log file after import and indexing grew to 1,500 MB

The time it took to process alter table and index script was 23 minutes on core i3 processor and 8GB RAM.

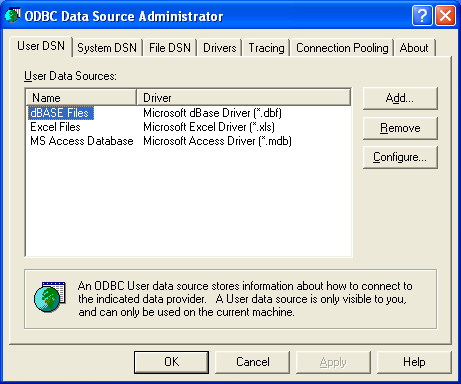
The database naming convention that should be used is **ConPerf\_*customer\_name*** where *customer\_name* is set for the particular engagement.

  
The create table script is not necessary as the RELOG operation will create the tables, but you can still use the script if you desire.

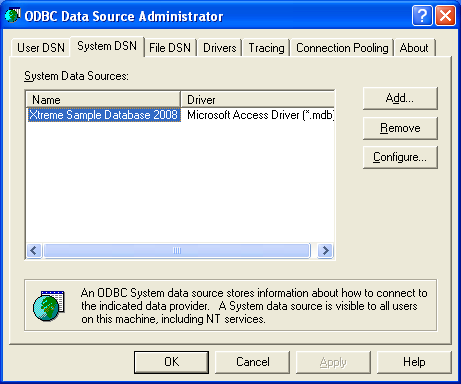
### **PHASE 2: Creating DSN Connection**

Create a DSN to connect to SQL Server for loading the PerfMon data. Following are the steps to create the DSN connection.

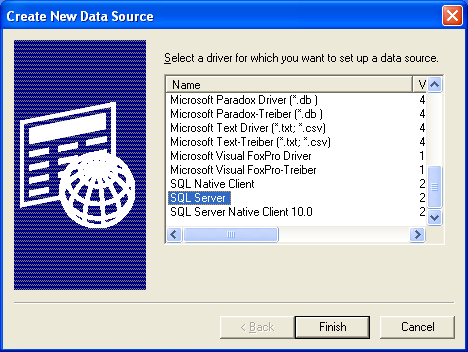
**Step 1:** Start🡺 All Programs🡺Administrative tools🡺ODBC Sources



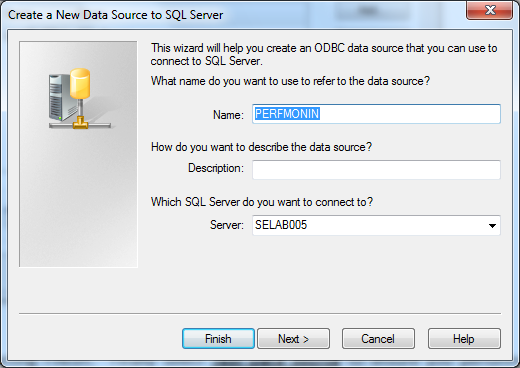
**Step 2:** 🡺Select the System DSN tab, and then click the **Add** button



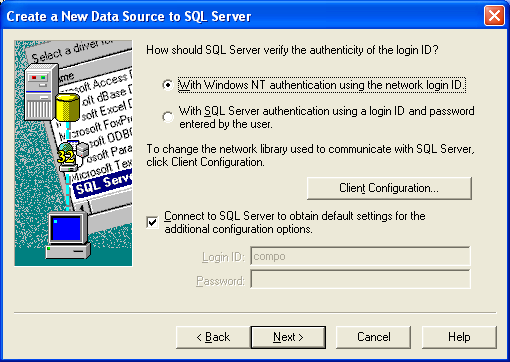
**Step 3:** 🡺 Select **SQL Server**

🡺 Select Finish.

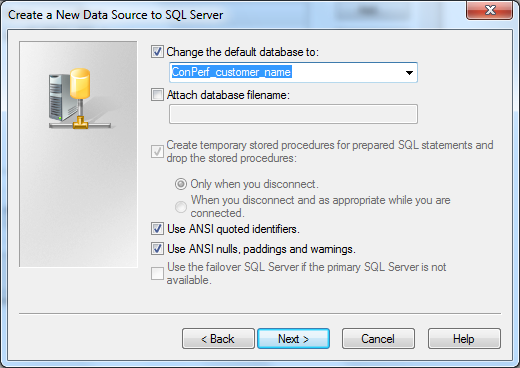
**Step 4:** For **Name** enter **PERFMONIN** (this will be your DSN name) and for **Server** your SQL Server name.

🡺 Click Next

**Step 5:** 🡺 Just click Next



**Step 6:** 🡺 Change the database which is to be connected from Master to **ConPerf\_*customer\_name***.

 🡺 Click Next

**Step 7:**  🡺 Complete the steps by selecting FINISH. Finally select **Test Data Source** to ensure you performed the steps properly.

### **PHASE 3: Uploading PerfMon Data Into SQL Server Tables with RELOG**

There are a couple of approaches to take here. One is to write out the names of all the files as in the example attachment. There is no need to convert from one form to the other prior to loading into SQL Server. 

Another approach to loading the PerfMon data that does not call for writing out the file names of every file is to use a looping batch file that will go through all the files. Below in **RED** is the text for creating the batch file **RelogIntoSQL.BAT**. Before creating the script, modify accordingly.

1. For the line in the code that says [**dir** C:\ Customername\perfmon\\*.BLG ] you have to first copy all the PerfMon files to that directory.
2. Change the file extension in the script to match your files (either BLG or CSV).
3. Underneath the path of [C:\ Customername\perfmon\] create a directory named [Perfmon\_done]
4. Remember to ensure the ODBC DSN named PERFMONIN is pointing to the correct database, ConPerf\_*customername*

For all the files successfully imported into your database, they will have been moved to the **PerfMon\_done** directory.

:: Loads PerfMon Files in Sequential Order

::

@ECHO OFF

time /t

IF "%1" == "\*TASKS\*" GOTO TASKS

SET NUMBER=1

:COPY-PROJ

echo 1

**dir** C:\ Customername\perfmon\\*.BLG

Echo COPY-Proj

echo 2

FOR %%F IN (C:\Customername\perfmon\\*.BLG) DO CALL "RelogIntoSQL.bat" \*TASKS\* %%F

GOTO DISPLAY

:TASKS

echo RELOG command for %2 runs here.

relog %2 -o SQL:PERFMONIN!%NUMBER% -f SQL

Move %2 C:\ Customername\perfmon \Perfmon\_done\

SET /MATH NUMBER=%NUMBER% + 1

GOTO END

:DISPLAY

ECHO This is it.

SET NUMBER=

time /t

:END

### **PHASE 4: Alter Column Data Types, Indexing, Adding CPU Factor Table, & PowerPivot Views**

To convert data type from varchar () to smalldatetime for the counterdatetime column of **CounterData** table and then create indexes on **CounterData** and **CounterDetails** tables. The below scripts does the following:

* Alters the temporary table created earlier using the script.
* Moves all data from old table “counterdata” to temporary table “Tmp\_counterdata” and then drop the “counterdata” table.
* Renames the temporary table to “counterdata”.
* Creates indexes to make processing go faster.
* Creates a **CPUFactor** table.
* Adds views for supporting Power Pivot that may be used later.

****

### **PHASE 5: Import CPU “FACTOR” from an Excel Spreadsheet**

Before performing this step, be sure to save the spreadsheet as a CSV file format.

In SSMS, right click the **ConPerf\_*customer\_name*** database and select Tasks, then Import Data … In the “Choose a Data Source” dialog screen, for Data Source select Flat File Source and then pick the applicable CSV file, and if the file has column headers such as server name, then be sure to also check the box “**Column names in the first data row**”. “Choose a Destination” select SQL Server Native Client 10.0, for “Select Source Tables and Views” choose the **CPUFactor** table. “Review Data Type Mapping” screen will reveal if there are problems with the spreadsheet’s data types. The table’s ServerName is varchar, while the CPUFactor column is numeric. If the Convert boxes are checked, you should be OK. When done importing the CPUFactor data, double-check your work to make sure it worked, execute *select \* from the CPUFactor* to make sure the data appears as it should.

**Deliverables:** At the end of Day-4, the following deliverables must be completed

1. PerfMon Database with tables and indexes built.
2. Get familiar with the data in the tables

## **DAY 5 – PowerShell’s Windows & SQL Server Meta Data Loading**

At the end of day-5, you will have learnt PowerShell data loading of the consolidation engagement

### **PHASE 1: Create a database in SQL Instance**

Create a database in the simple recovery mode using the naming convention **CONSHELL\_customername** this is where the SSIS package is going to build the tables and then fill them. 

### **PHASE 2: Loading PowerShell Data using SSIS Package**

For loading PowerShell Data we use an SSIS Package. Copy the file **BatchLoadPowerShellData\_V5.dtsx** from SharePoint’s “**Consulting**” division, **SQL Toolbox**, Consolidation folder to your local disk.

Start by opening BIDS, then select **creating** a New Project - Integration Project. Then click on Integration Services Project, and save it and name it where ever you desire. Then click on the **Solution Explorer** shortcut button. Then on the right-hand side of the screen, right click on SSIS Package and select Add Existing Package. Next to “**Package Location**”, select in the drop down File System. Browse the Package Path, and select the file **BatchLoadPowerShellData\_V5.dtsx**. Click Ok.

Package consists of Control and Data Flow Tasks. These tasks represent a file type of CSV data that is collected. Each file type goes into the database as a table with data for every server.

### **PHASE 3: verify the Variable lists are visible**

* Follow the steps given below to load data into SQL Server.
* After the Package is loaded in the SSIS, verify the Variable lists are visible on the left side of the screen. If not, then click on the Control flow tab of the designer, then from menu bar at the top click **SSIS**-> **Variables**.

**To run this package you need to edit 3 of the 4 variables that are at the package level. Variable values are as follows:**

**DestDB** = This string is the name of the DATABASE where PowerShell data should be loaded. While the database  
 does need to exist, the tables to be loaded are not required to exist since the first task in the package   
 will drop the tables if they exist, and will then create them.

**DestServer =** This string is the name of the SQL Server (and instance if not the default instance) where the database   
 listed in DestDB exists.   
 Example of Servers:   
 Localhost\SQL2k8R2 --local server with an instance named SQL2k8R2   
 MyServer --Server named MyServer using the DEFAULT instance   
 10.92.166.32\SqlInst2 --Server with IP 10.92.166.32 and instance named SqlInst2

**RootDir =** Is a string with a path where the sequence will start looking for the .CSV files. It will check all the sub   
 directories under the root directory. The path needs to end in "\"  
 Often the path to the CSV files will look like C:\customername\SE\Data\servername\instancename,   
 In such a situation, you would enter **C:\customername\SE\Data\**

**When done, now hit the SAVE button.**

**FileNameWithPath** = **\*\*\*\*Do not edit this VARIABLE\*\*\*\*** It must exist at the root of the package so expressions for the file path in the Flat File connections.   
 But its value needs to be changed at RUNTIME for each file found in the ForEach File Loop.   
 But don’t worry if what is in the path at present is something you do not have.   
 the package will rewrite itself.

***Note:*** Changing anything other than these variables will likely cause errors in the package or invalid results.   
If you want to add to the tables loaded without deleting the data already loaded disable this first task ("Build Tables If Needed") then run the package. If you want to load everything fresh then make sure this task ("Build Tables If Needed") is enabled before running.

### **PHASE 4**: **Executing the SSIS Package**

After all the Editing, on the right pane, browse through **Solution Explorer**-> **SSIS Package**-> **BatchLoadPowerShellData\_V5.dtsx**. Right click on **BatchLoadPowerShellData\_V5.dtsx** and click Execute Package. After successful Execution check in your database whether complete data has been loaded successfully or not.

### **PHASE 5**: **Creating Views for PowerPivot Analysis**

Execute the following script in the **ConShell\_customername** database.



### **Deliverables**

At the end of Day-5, the following deliverables must be completed

1. PowerShell Database
2. Get familiar with the data in the tables

## **DAY 6 – PerfMon Analysis (SES)**

**Objective**

By the end of day-6, you will have learned how to prepare reports from PerfMon data.

### **PHASE 1: Processing PerfMon Data with the Unified Script**

PerfMon Analysis INCLUDES preparing following Analysis Sheets:

* Network Analysis Sheet
* Storage Analysis Sheet
* Tier Analysis Sheet(s)
* CPU Analysis Sheet

The bulk of the processing is done with the embedded script which is an evolution from the many smaller scripts once used. Run the code below.

### **PHASE 2: Placing Results Into Spreadsheets**

Select out the data from the various titled outputs and place into various tabs of an Excel workbook. Title each tab with the appropriate name.

For Network analysis you have to analyze the following with the understanding that servers can have many NIC’s and each NIC can have many ports. All of them have to be summed so we understand the complete load the server is generating and receiving.

**Avg\_NW\_Bytes\_PHPM**. This is the hour and maximum value of Network Bytes/sec for each server out of all the hourly averages.

**Avg\_NW\_Packets\_PHPM**. This is the hour and maximum value of Network Packets/sec for each server out of all the hourly averages.

Storage Analysis includes the following data points, much in the same way as the above networking data; summarized by calculating the average of each hour surveyed, and then selecting the heaviest hour. This data is used to create the Storage Analysis sheet.

* AVG\_Disk\_Reads\_sec
* AVG\_Disk\_Writes\_sec
* AVG\_Disc\_Transfer\_sec
* Avg\_Disk\_Read\_Bytes\_sec
* Avg\_Disk\_Write\_Bytes\_sec
* AVG\_Disk\_Read\_Queue\_Length
* AVG\_Disk\_Write\_Queue\_Length
* AVG\_Disk\_sec\_Read
* AVG\_Disk\_sec\_Write

Once the peak hour for each of the servers are calculated import this data into a sheet and naming it as **storage\_op**.

**Avg\_CPU\_PHPM**. This script calculated the average value of % processor time for each server for every hour that data was captured. It then selected the maximum hour out of all those hours. Thus, we now have the maximum value for each server, or in other words, their peak hour. For instance, if PerfMon was executed on the server for just 24hrs then we will have 24 values for that server. So we have 24 values for every server in scope of Consolidation. It is preferred to have much more than just 24 hours of data, but each customer is different, and sometimes, that’s all the time afforded the consultant. Once we have these details with us copy the results onto a spreadsheet naming the tab as **Avg\_CPU\_PHPM**.

### **Tiering the Servers**

The script assigned numbers to our counters depending upon the corresponding values which will be used in tiering the Servers. Copy the results into an Excel Sheet and name it TIER\_ALLin2.

* Tier 1 query will give all the servers which have overly high resource loading of one sort or another.
* Tier 2 query will give all the servers which are approaching (but not there yet) high resource loading of one sort or another.
* Tier 3 query will give all the servers which are of low resource loading.

Unfortunately, tiering calculations sometimes needs manual intervention. That is the case for the categories of **Processor Queue Length** (PQL) and **Context Switching**. Why? Because, these two values are reported by PerfMon on a per machine basis, but in the modern era that perspective is meaningless. We need to **evaluate these values on a per core basis** not on a per server, or even per socket basis. PerfMon has not kept up with the times for some measurements. These categories were first created when Windows typically ran on machines with 1 CPU socket which had just 1 core. That is not the case anymore, many sockets, and many cores in each socket. For instance, if you see a context switch value of 35,000, think about this number relative to the number of **cores** (avoid thinking in terms of logical processors (LP’s) because an LP could also include hyper threading, which is not relevant to this particular topic.). The Tier1 threshold is 5,000 **per core**. Thus, if for example the machine has 8 cores, does this mean it should be labeled as Tier1 in the category of context switching? NO!!! 8 x 5,000 = 40,000. Tier2 then? Perhaps, but absolutely not Tier1! Evaluate the Processor Queue Length (PQL) values in the same manner, based upon per cores.

### **CPU\_Analysis**

In the **CPU\_Analysis** sheet we evaluate the old systems against the intended model of server and chipset based upon what the architect\consultant has determined will be the object of comparison. The choice of intended model of server and chipset is done in coordination with the end customer and the engagement sponsor. Never assume this point. It must be clearly communicated to you, as no two customers and engagements are the same, and thus the target consolidation platform is not always the same. Follow the below steps to fill up the sheet.

**Step 1:**

We prepare a draft with the headers as specified in the attached document. These headers are considered to be standard and headers could be increased on demand.



**Step 2:**

The Client uses some server models which can be extracted from the PowerShell Database. Query out all the distinct Server Models and the Servers which fall into that category and paste in into the Excel Sheet.

**Step 3:**

Insert the respective Type CPU and No. of Sockets for the Servers by using the Client Questionnaire or Database tables.

**Step 4:**

Use the below Sheet to find out the Source’s Max Comparative Capacity by following the below Steps.

**Note: This Sheet is subject to change with respect to the latest models. Please check for the latest sheet on our SharePoint portal** “**Consulting**” division, **SQL Toolbox**, Consolidation folder**.**



* Select the Server which is latest and to be included into the scope. Copy Paste the values corresponding to it in “Plug in your new server data here”.
* Search the Server Model in the above sheet. If you find the correct Server Model Check the Type CPU and also the respective Server Model.
* If you can’t find the Server Model and corresponding Type CPU, check that server has 2, 4 or 8 Sockets. Accordingly search in 2way, 4way or 8way.
* Now check the type CPU you were trying to search.
* In the corresponding row try to search the Server Model you were trying to find.
* If you found the Chipset Model, Select the value at the beginning of the row.
* In case you can’t find the Type CPU in specific number of Sockets try searching in other Socket number. If you found the Type CPU use the value and in the Comment Section of the sheet specify what assumptions have been made.
* Repeat the steps to find out Source’s Max Comparative Capacity of all the Servers.
* Fill the Target % CPU from the CPU Processor value sheet. Calculate the Load transferred (number with no decimal values) by multiplying Source’s Max comparative Capacity and Target % CPU.

### **Deliverables**

At the end of Day-6, the PerfMon Data Analysis and report must be completed.

## **DAY 7 – Analysis of the PowerShell Server Meta Data**

**Objective**

At the end of day-7, you will have learnt preparing reports from PowerShell data

**Tasks Description**

PowerShell Analysis includes preparing following Analysis Sheets:

* Memory Analysis
* Database Analysis
* Server List
* Datafile Info Analysis
* Logfile Info Analysis
* Physical disk Info Analysis

### **Memory Analysis**

In Memory Analysis Sheet you need to include the details of memory of each server. The details can be extracted from one or multiple tables. It should at least include ServerName and Total Physical Memory in MB. At the end take a sum of the Total Physical Memory. Other details can be provided. It depends solely on the request or decision of Consultant.

### **Database Analysis**

In Database Analysis Sheet you need to include details of databases present in each server. The details can be extracted from one or multiple tables. It should at least include Database Id, ServerName, InstanceName, Database Name, Size and Space Available. Other details can be provided. It depends solely on the request or decision of Consultant.

### **Server List**

In Server List Sheet you need to include details of servers’ present scope of consolidation. The details can be extracted from one or multiple tables. It should at least include ServerName, InstanceName, Collation, Version, Edition, Server Model, Manufacturer and Domain. Other details can be provided. It depends solely on the request or decision of Consultant.

### **Data File Info Analysis**

In Datafile Info Analysis Sheet you need to include details of data files (mdf). The details can be extracted from one or multiple tables. It should at least include ServerName, Database ID, Database Name, filename, Size and Used Space. Other details can be provided. It depends solely on the request or decision of Consultant.

### **Log File Info Analysis**

In Logfile Info Analysis Sheet you need to include details of log files (ldf). The details can be extracted from one or multiple tables. It should at least include ServerName, Database ID, Database Name, Filename, Size and Used Space. Other details can be provided. It depends solely on the request or decision of Consultant.

### **Physical Disk Info Analysis**

In Physical Disk Info Analysis Sheet you need to include details of log files (ldf). The details can be extracted from one or multiple tables. It should at least include ServerName, Name, Size and Free Space. Other details can be provided. It depends solely on the request or decision of Consultant.

### **Deliverables**

* PowerShell Analysis and prepare Report.
* Create a single Consolidation presentation by merging the presentations developed in previous days.
* It is not required to present this, however this is the time when the student will learn to create a final presentation for delivery. Also the approach of creating smaller presentations and then gluing them together, will help the consultant (especially in the first few engagements) to focus on smaller tasks to achieve the overall bigger task of successfully delivery. All the best ☺
* Discuss in an open forum on consolidation questions
* Document how you will improve the consolidation engagement
* Document any lessons learnt
* Evaluation and Sign-Off

## **Final actions after ALL the processing**

The last actions required are to use SQL Server’s PAGE compression on all the tables of both databases, then shrink the data and log files, then backup both databases using compression, and post them to the FTP site so that the engagement’s architects and\or consultants can restore the databases and conduct further analysis by way of Power Pivot.

## **DAY 8 – PowerPivot Setup (new technique For Advanced Analysis)**

The following technique is very powerful and can aid the architect in easily and quickly double checking the results from the previous analysis. Moreover, you can use it to drill into the data more deeply into any particular area.

Software Requirements:

* An install of SQL Server 2012, but can still be done with SQL Server 2008 R2.
* An install of **PowerPivot for SQL Server 2012 and Excel 2010**, download from here:

<http://www.microsoft.com/en-us/download/details.aspx?amp;amp;displaylang=en&id=29074>

* + PowerPivot for SQL Server 2012 and Excel 2010 pre-requisites are:
    - Microsoft .NET Framework 4

<http://www.microsoft.com/en-us/download/confirmation.aspx?id=17851>

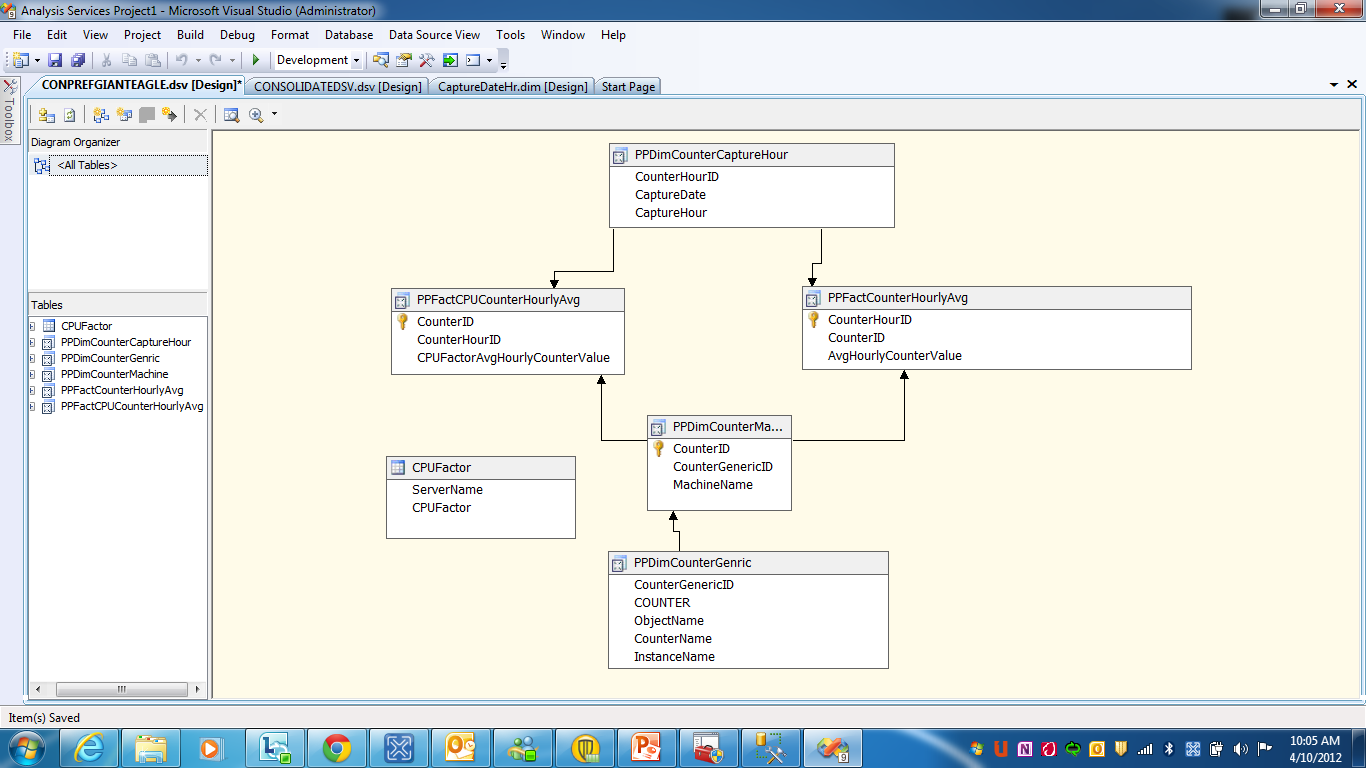
* + - Visual Studio 2010 Tools for Office Runtime

<http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=20479>

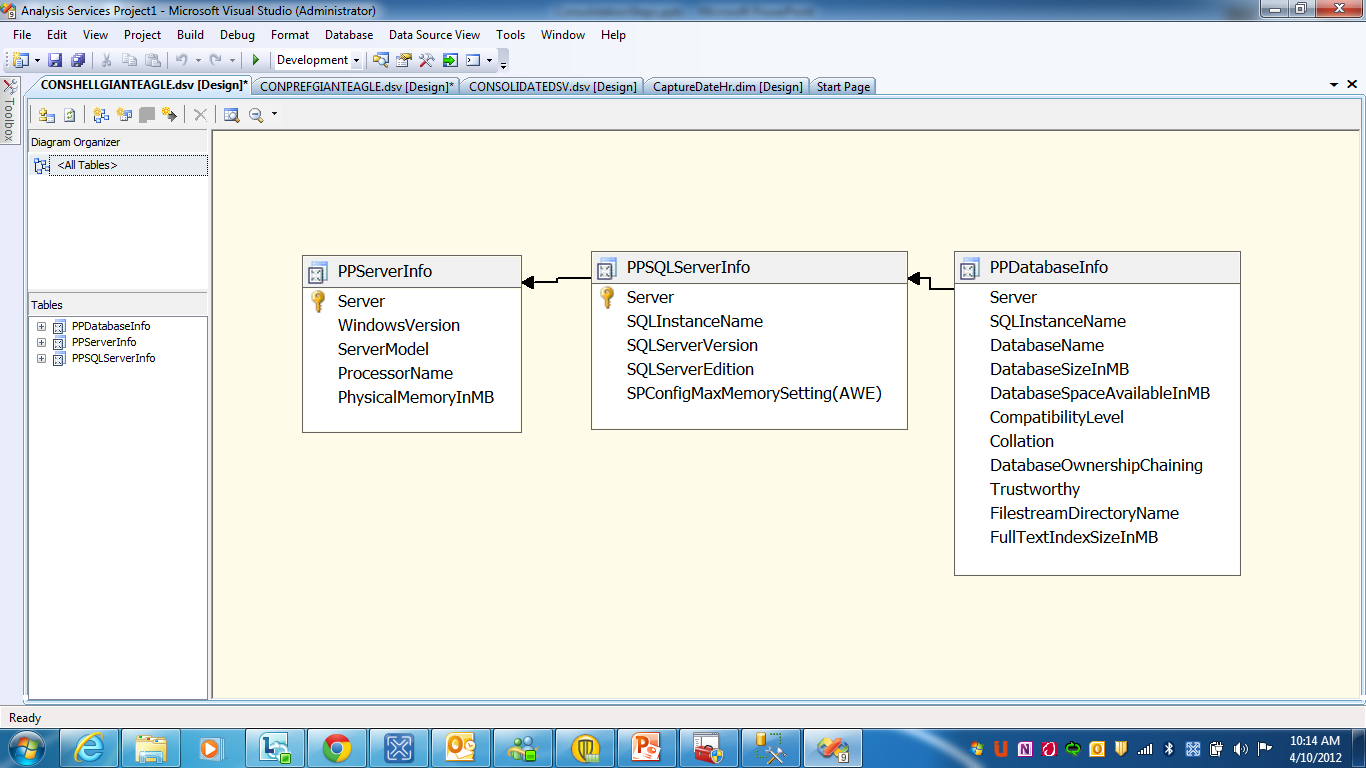
I recommend your view the two videos **PowerPivot Consolidation webcast Pt1.wmv** and **PowerPivot Consolidation webcast Pt2.wmv** to familiarize yourself with the process if you are not already. They should be attached to the zip file this document belongs to.

The following two diagrams are crucial to understanding how to use PowerPivot for the rest of this section. Be sure you understand it well.

### **PerfMon Data Snowflake Schema**



### **Server Meta-Data Schema**



## Building PerfMon Data Into a PowerPivot Display

1. Open Excel and then click **PowerPivot** menu tab of Excel
2. Click “**PowerPivot Window**” shortcut menu icon.
3. Click “**From Database**” menu button.
4. Select your intended SQL Server instance.
5. Select your intended ConPerf\_xxx database.
6. Select all the views with names starting with **PP…** as well as the table named **CPUFactor.**
7. (**If** you have installed the Excel add-on named **PowerPivot for SQL Server 2012**, then hit the *Diagram View* button).
8. You will see 6 tables figures displayed, now you have to draw relationships between the views. You should heavily reference the snowflake schema picture in the preceding page of this document. It is the basis for what you do next.
9. Drag and drop from one view’s column to another’s.
   1. PPDimCounterCaptureHour (CounterHourID) to PPFactCPUCounterHourlyAvg (CounterHourID)
   2. PPDimCounterCaptureHour (CounterHourID) to PPFactCounterHourlyAvg (CounterHourID)
   3. PPDimCounterMachine (CounterID) to PPFactCPUCounterHourlyAvg (CounterID)
   4. PPDimCounterMachine (CounterID) to PPFactCounterHourlyAvg (CounterID)
   5. PPDimCounterGeneric (CounterGenericID) to PPDimCounterMachine (CounterGenericID)
10. From the menu buttons on the top ribbon, select **PivotTable** and then **PivotTable** again.
11. Select **new worksheet**.
12. Now you define the measures. The value is derived from the fact tables.
    1. Expand PPFactCounterHourlyAvg and place a checkmark on column AvgHourlyCounterValue
       1. After doing so look in the lower right-hand corner in the box labeled **∑ Values**, you should now see inside the box the expression “Sum of AvgHourlyCounterValue”.
    2. Expand PPDimCountMachine and place a checkmark on column MachineName.
       1. This will show up in the box named **Row Labels**.
    3. This time, instead of placing a checkmark on a particular view’s column, you need to do a drag and drop. Drag DimCounterCaptureHour column CaptureDate, to the box labeled **Column Labels**.
    4. Repeat using the column CaptureHour.
       1. Make sure when you are done, that in the **Column Labels** box, that CaptureDate is above CaptureHour.
13. Now you are slicing and dicing.
    1. Drag DimCounterGeneric column Countername, to the box labeled **Slicers Vertical**.
    2. Drag DimCounterGeneric column Objectname, to the box labeled **Slicers Horizontal**.
14. Now you turn your attention away from right-hand’s side of the worksheet and the section called PowerPivot Field List, and to the main body of the worksheet.
15. Look for a column heading named “**Row Labels**”. Right click on the “Row Labels” and select “**PivotTable options…”**. Then select the tab **Display**, and then place a checkmark for **Classic PivotTable** **layout** and then OK.
16. Next, right click on “**MachineName**” and remove the checkmark for “**SubTotal MachineName**”.
17. Next, right click on “**CaptureDate**” and remove the checkmark for “**SubTotal CaptureDate**”.
18. Next, click on the menu ribbon’s **Insert** tab, then inside the **Charts** section of the ribbon, click on the **Column** button to make a chart.
19. Move your mouse inside the chart and right click in the white space, and select Move Chart, and then New Sheet and name it as PerfMon Chart.
20. Now you use your imagination on what you want to look at and how you want to display it.

Now go and explore and see what else you can learn from your data.

## Building PowerShell Data Into a PowerPivot Display

The following are steps to get you started. They are not the only things you can or even should do.

1. While still in the same Excel workbook, but on a clean worksheet, click **PowerPivot** menu tab of Excel
2. Click “**PowerPivot Window**” shortcut menu icon.
3. Click “**From Database**” menu button.
4. Select your intended SQL Server instance.
5. Select your intended ConShell\_xxx database.
6. Select all the views with names starting with **PP...**
7. Next, look for the **PivotTable** button on the top menu ribbon, in the PivotTable (drop down menu) select **Flattened Pivot Table**.
8. On the far right, expand the **PPServerInfo** view; put a checkmark on the column named **Server**, then **ServerModel**.
9. Then in the upper right hand section of the spreadsheet, right click on the **Server** column header, and remove the checkmark for **SubTotal “Server”**.
10. Now go and explore and see what else you can learn from your data.