



## Microsoft SQL Server PowerPivot for SharePoint: Transitioning from a Single Server to a SharePoint Farm

SQL Server Best Practices Technical White Paper

**Authors:** Dave Wickert, Denny Lee, Ed Campbell

**Technical Reviewers:** Zohar Raz, Mark Souza

**Published:** June 2010

**Applies to:** SQL Server 2008 R2

**Summary:** This white paper is a How-to document that provides you with the steps to transition from a single-server Microsoft® SQL Server® PowerPivot for SharePoint® installation to a multi-server PowerPivot for SharePoint farm. As your PowerPivot for SharePoint environment becomes more heavily utilized with more PowerPivot workbooks and more concurrent users, server resources such as CPU, memory, disk, and network become saturated. To alleviate resource bottlenecks, SharePoint itself is designed as a scale-out solution; as more resources are required, more servers can be added to the farm. Therefore, it is important to transition your existing PowerPivot for SharePoint single-server installation into a multi-server farm as resources are required.

# Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

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

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, Excel, SharePoint, SQL Server, VertiPaq, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

# Contents

- Introduction ..... 4
- Installation Workflow ..... 5
- Adding a Dedicated Server Instance of SQL Server..... 7
  - Why Should You Have a Dedicated Server Instance of SQL Server for Your SharePoint Databases? ..... 7
  - Moving the SharePoint Databases to a Dedicated Server Instance of SQL Server..... 8
    - 1) Install PowerPivot ..... 8
    - 2) Move the Databases ..... 9
- Adding a SharePoint Application Server ..... 13
  - Moving the Applications..... 14
- Basic Three-Server PowerPivot for SharePoint Farm..... 15
- Conclusion ..... 15
- Appendix..... 16
  - Appendix A..... 16
  - Appendix B..... 19
  - Appendix C..... 21

## Introduction

This white paper describes an approach for expanding a single PowerPivot machine into a farm of servers. This approach does not replace the normal sequence of events that you would use for multiple machines for your SharePoint farm, because in many cases you should architect and construct your farm as you normally would. This How-to white paper applies to sites that start off with a single machine and then over time need to expand their capabilities as load and resource consumption increase. The transition will initially take you from a single “all-in-one” server to two machines, including one that is dedicated to the SharePoint SQL Server databases. As more resources are utilized within Excel® Services, PowerPivot System Service, and Analysis Services in VertiPaq™ mode, you will need to add additional SharePoint application servers. With this last step, you will have a basic three-server SharePoint farm that includes a SharePoint Web front end (WFE) (rendering and delivering Web services), a SharePoint application server (with PowerPivot and dependent services enabled), and a SharePoint SQL Server databases server. As more resources are required, you can scale out from this basic setup (for example, add more application servers). This technique offers the following benefits:

- The initial startup is easier for a single-server installation than it is for a multi-server farm.
- You transition to a multi-server farm only when necessary.
- The steps involved in moving to a multi-server farm are fewer and less complicated when you scale up from a single-server installation.

This white paper focuses on the transition from a single-server installation to a farm. This is only part of the larger planning and deployment process for an expanding organization; research on your organization’s capacity planning and needs must also be a part of your process. For more information about capacity planning, see the following resources:

- Capacity management for Microsoft SharePoint Server 2010:  
<http://technet.microsoft.com/en-us/sharepoint/ff601870.aspx>
- Capacity management and sizing for SharePoint Server 2010:  
<http://technet.microsoft.com/en-us/library/cc261700.aspx>
- Storage and SQL Server capacity planning and configuration (SharePoint Server 2010):  
<http://technet.microsoft.com/en-us/library/cc298801.aspx>

For more information about PowerPivot, see the following resources:

- Academy Live: A Preview to PowerPivot Server Best Practices:  
<http://academyliv.vo.msecnd.net/CALT/SQL90CALT/20091215-CALT-SQL90CALT-WMV.wmv>
- PowerPivot Technical Diagram: PowerPivot Client/Server Architecture:  
<http://sqlcat.com/whitepapers/archive/2010/04/23/powerpivot-technical-diagram-powerpivot-client-server-architecture.aspx>

Stay tuned to the SQL Customer Advisory Team (<http://sqlcat.com>) Web site. We plan to publish a PowerPivot for SharePoint capacity planning document.

## Installation Workflow

When you are working with PowerPivot for SharePoint, you can:

- Build a single-server PowerPivot for SharePoint installation by following the instructions provided on either PowerPivotGeek.com or SQL Server 2008 R2 Books Online. For more information, see [Single Server Install](http://powerpivotgeek.com/server-installation/single-server-install/) (<http://powerpivotgeek.com/server-installation/single-server-install/>) or [How to: Install PowerPivot for SharePoint on a New SharePoint Server](http://msdn.microsoft.com/en-us/library/ee210708(v=SQL.105).aspx) ([http://msdn.microsoft.com/en-us/library/ee210708\(v=SQL.105\).aspx](http://msdn.microsoft.com/en-us/library/ee210708(v=SQL.105).aspx)).
- Build an existing multi-server PowerPivot for SharePoint farm. For more information, see [Existing Farm Install](http://powerpivotgeek.com/server-installation/existing-farm-install/) (<http://powerpivotgeek.com/server-installation/existing-farm-install/>) or [How to: Install PowerPivot for SharePoint on an Existing SharePoint Server](http://msdn.microsoft.com/en-us/library/ee210616(v=SQL.105).aspx) ([http://msdn.microsoft.com/en-us/library/ee210616\(v=SQL.105\).aspx](http://msdn.microsoft.com/en-us/library/ee210616(v=SQL.105).aspx)).

Although the single-server farm is a very simple and straightforward installation (five steps), you cannot expand it if more resources are required. The multi-server farm is a great scale-out solution, but it requires over twenty steps to perform the installation.

What we show in this document is a third way to build your multi-server PowerPivot for SharePoint farm by transitioning from your single-server installation. Figure 1 depicts the installation workflow.

The steps from a single-server installation to a basic three-server SharePoint farm are:

1. In Step [1], you install the single-server (for more information, see [Single Server Install](http://powerpivotgeek.com/server-installation/single-server-install/)).
2. In Step [2], you add a dedicated instance of SQL Server (for more information, see “Adding a Dedicated Instance of SQL Server” in this white paper).
3. In Step [3], you add an application server (for more information, see “Adding an Application Server” in this white paper).

After the third step, you will have a basic three-server PowerPivot for SharePoint farm with one SharePoint WFE, one SharePoint application server, and one SharePoint SQL Server database server. As your organization requires more resources, you can add more SharePoint application servers (the more likely action) and SharePoint WFE servers as appropriate.

Figure 1 shows the process in detail.

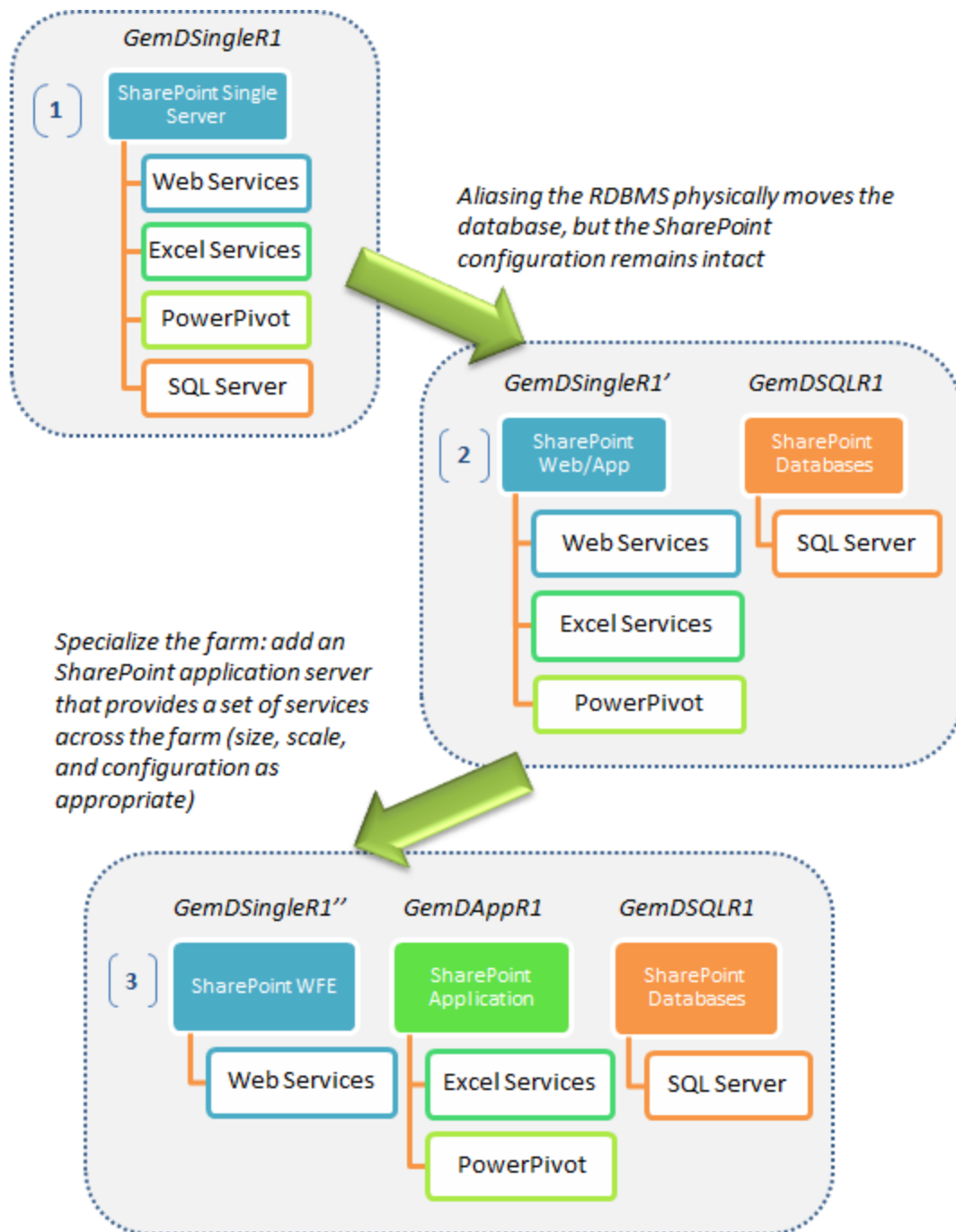


Figure 1: Transitioning from a single server to SharePoint farm (PowerPivot for SharePoint)

## Adding a Dedicated Server Instance of SQL Server

One of the first things to do to alleviate resource contention is to place the SharePoint databases on a separate server as shown in figure 2.

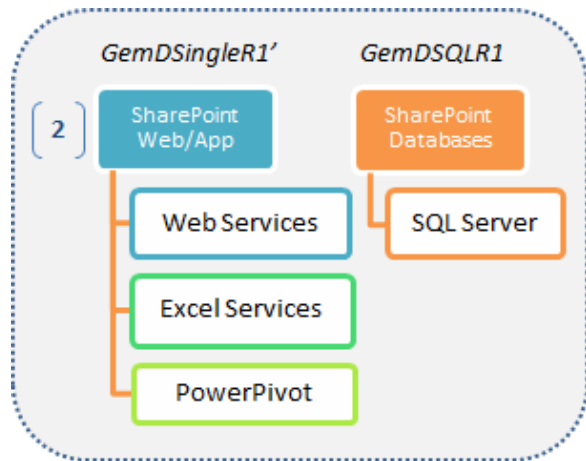


Figure 2: Adding a dedicated server instance of SQL Server to your PowerPivot for SharePoint single-server installation

### Why Should You Have a Dedicated Server Instance of SQL Server for Your SharePoint Databases?

The PowerPivot for SharePoint single-server setup is designed for smaller departmental scenarios where there are not a lot of users or workbooks (in terms of number and/or concurrency). But as more people start using your PowerPivot for SharePoint environment, it will be beneficial to move the SQL Server databases that SharePoint uses to a separate dedicated server instance of SQL Server, because a high volume of transactions will interact with these databases.

By moving these databases to their own server instance of SQL Server, you can dedicate memory, CPU, and I/O resources for just the SharePoint SQL Server databases. This is also one of the first steps you perform in order to convert your smaller departmental PowerPivot for SharePoint environment into a multi-server SharePoint farm (that is PowerPivot enabled) because you will need to have a separate dedicated set of instances of SQL Server to support your environment. For more information about SharePoint topologies, see the [SharePoint Server 2010 Planning and Architecture Technical Diagrams](http://technet.microsoft.com/en-us/library/cc263199(office.14).aspx) ([http://technet.microsoft.com/en-us/library/cc263199\(office.14\).aspx](http://technet.microsoft.com/en-us/library/cc263199(office.14).aspx)).

---

**Lesson Learned** During the PowerPivot TAP, we learned that having a separate dedicated instance of SQL Server for your SharePoint databases can in many cases improve upload and download performance of PowerPivot workbooks, especially if they are large.

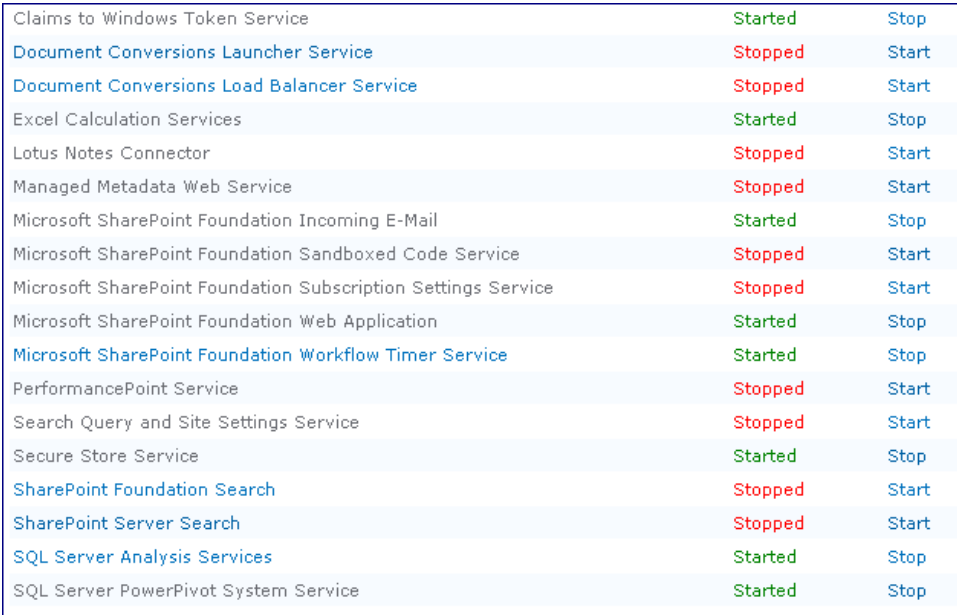
---

# Moving the SharePoint Databases to a Dedicated Server Instance of SQL Server

This section discusses the steps you need to perform to switch from a PowerPivot single-server installation to a PowerPivot single-server installation with a dedicated server instance of SQL Server.

## 1) Install PowerPivot

If you haven't installed PowerPivot for SharePoint, follow the instructions in [Single Server Install](#) or [How to: Install PowerPivot for SharePoint on a New SharePoint Server](#). After you complete the installation, verify that everything is up and running. In SharePoint Central Administration, in System Settings, click **Manage services on server**.



Claims to Windows Token Service	Started	Stop
Document Conversions Launcher Service	Stopped	Start
Document Conversions Load Balancer Service	Stopped	Start
Excel Calculation Services	Started	Stop
Lotus Notes Connector	Stopped	Start
Managed Metadata Web Service	Stopped	Start
Microsoft SharePoint Foundation Incoming E-Mail	Started	Stop
Microsoft SharePoint Foundation Sandboxed Code Service	Stopped	Start
Microsoft SharePoint Foundation Subscription Settings Service	Stopped	Start
Microsoft SharePoint Foundation Web Application	Started	Stop
Microsoft SharePoint Foundation Workflow Timer Service	Started	Stop
PerformancePoint Service	Stopped	Start
Search Query and Site Settings Service	Stopped	Start
Secure Store Service	Started	Stop
SharePoint Foundation Search	Stopped	Start
SharePoint Server Search	Stopped	Start
SQL Server Analysis Services	Started	Stop
SQL Server PowerPivot System Service	Started	Stop

Figure 3: Managing Services on a Server view within SharePoint Central Administration

In figure 3, the services that are required for PowerPivot for SharePoint are all running, including Claims to Windows® Token Service, Excel Calculation Services, Microsoft SharePoint Foundation Web Application, Microsoft SharePoint Foundation Workflow Timer Service, Secure Store Service, SQL Server Analysis Services, and SQL Server PowerPivot System Service.

On the same server, in Central Administration, in Upgrade and Migration, click **Review database status**. Note that all of the databases are located on the same server (in figure 4, the names have been blurred out).

SQL Instance	Database	Type	Status
\POWERPIVOT	SharePoint_Admin_0fff507037704eb9af55d85e47514034	Content Database	No action required
\POWERPIVOT	SharePoint_Content_014191608ad74a94b466892285045214	Content Database	No action required
\POWERPIVOT	DefaultPowerPivotServiceApplicationDBc26cea5043aa40ad9fda7d65e509547a	GeminiServiceDatabase	No action required
\POWERPIVOT	Secure Store Service	SecureStoreServiceDatabase	No action required
\POWERPIVOT	SharePoint_Config_c0c859e4ac6747be9f6e1f3f08ac772b	Configuration Database	No action required
\POWERPIVOT	WSS_Logging	SPUsageDatabase	No action required

Figure 4: Review Database Status view from SharePoint Central Administration

## 2) Move the Databases

The starting point for moving the databases was the SharePoint 2007 article [Move all databases to a different database server](http://technet.microsoft.com/en-us/library/cc512725.aspx) (http://technet.microsoft.com/en-us/library/cc512725.aspx).

Because SharePoint 2010 no longer has Shared Services Providers (SSPs), the process is a little simpler; here are the steps.

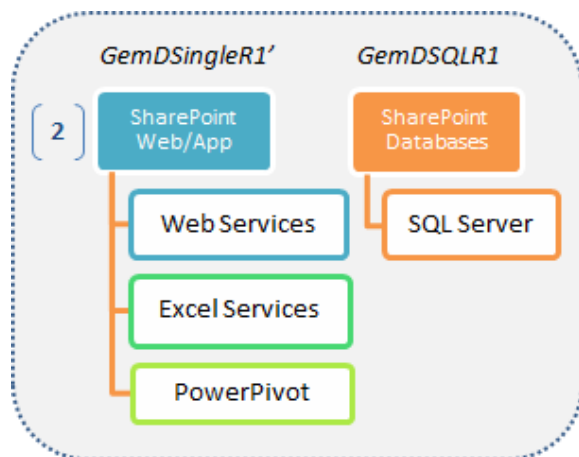


Figure 5: Server service breakdown when a dedicated instance of SQL Server is added (going from GemDSingleR1 to GemDSingleR1' and GemDSQLR1)

- 1) Prepare your new database server (GemDSQLR1) as described in the “Install SQL Server on the machine you designate as the DB Server” section of [Existing Farm Install](#). Be sure to use the same Installation/Administrator and service accounts that were used in the *PowerPivot for SharePoint Single Server* (GemDSQLR1) installation.

- 2) Shut down Microsoft Internet Information Services (IIS) on your original *PowerPivot for SharePoint Single Server* (GemDSingleR1) instance by typing **iisreset /stop** at an administrative command prompt. This prevents new transactions or entries from being added to the SharePoint databases.
  
- 3) Copy databases and logins between servers:
  - a. Back up all of the databases from the *PowerPivot for SharePoint Single Server* (GemDSingleR1) instance.
  
  - b. Create the required logins and relate them to the server roles on your new *dedicated SharePoint Databases Server* (GemDSQLR1). For a simple PowerPivot for SharePoint Single Server instance, there may only be one or two logins involved (such as Farm Administrator or Services Accounts). If you have more complex configurations, see the article [How to transfer the logins and the passwords between instances of SQL Server 2005 and SQL Server 2008](http://support.microsoft.com/kb/918992/) (http://support.microsoft.com/kb/918992/). As noted in figure 6, the login GEMDOMAIN\GeminiService in our environment has been re-created and given the same server roles from the original *PowerPivot for SharePoint Single Server* (GemDSingleR1) instance to the *dedicated SharePoint Databases Server* (GemDSQLR1).
  
  - c. Copy the databases from the original *PowerPivot for SharePoint Single Server* (GemDSingleR1) instance to the *dedicated SharePoint Databases Server* (GemDSQLR1) and restore the databases. As shown in figure 6, the databases are now restored to the *dedicated SharePoint Databases Server* (GemDSQLR1) instance, and they are removed from the original instance

---

**Note:** This white paper provides SQL scripts that you can use to automate steps 3a, 3b, and 3c. Appendix A contains an SQL script that can be run on the source *PowerPivot for SharePoint Single Server* (GemDSingleR1) instance, and Appendix B contains an SQL script that can be run on the destination *dedicated SharePoint Database Server* (GemDSQLR1).

---

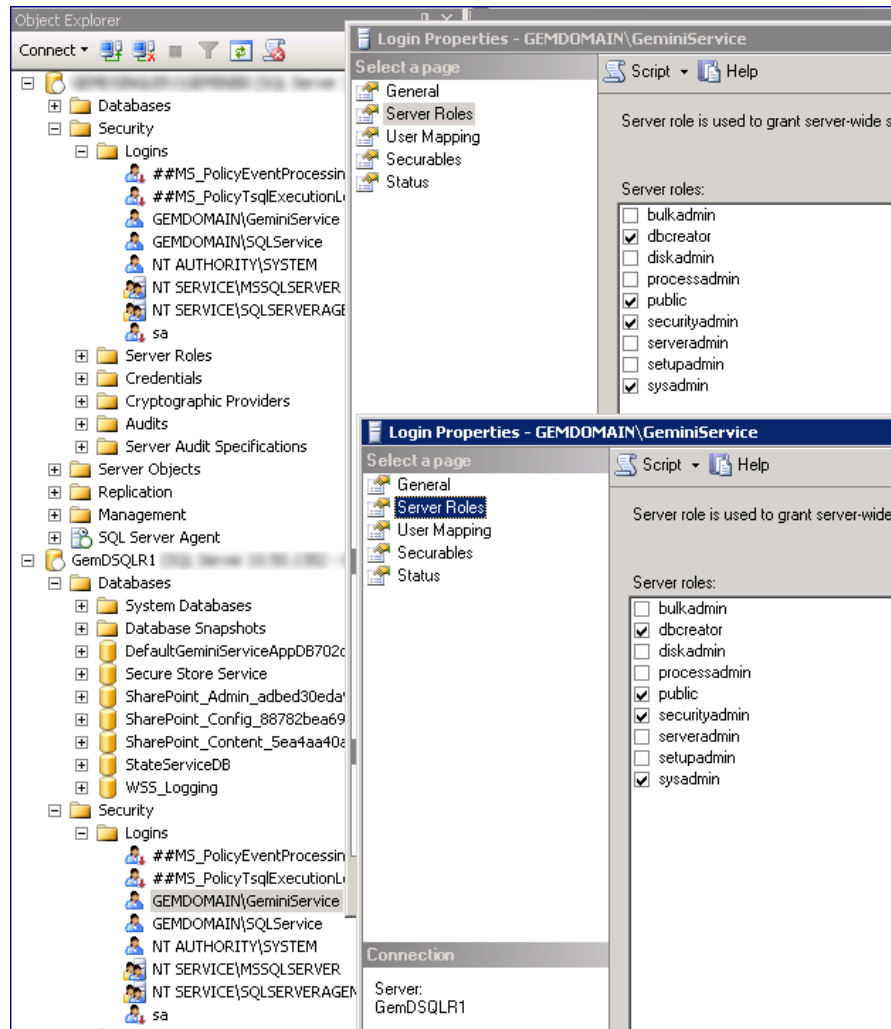


Figure 6: SQL Server Management Studio view of the PowerPivot databases and GEMDOMAIN\GeminiService login

- 4) By using the SQL Server Configuration Manager, create both 32-bit and 64-bit aliases on the original *PowerPivot for SharePoint Single Server* (GemDSingleR1') instance to point to the new *dedicated SharePoint Databases Server* (GemDSQLR1). Notice in figure 7 that there is an alias for GemDSingleR1\GEMINIBI (*PowerPivot for SharePoint Single Server* instance) that points to the GemDSQLR1 (*dedicated SharePoint Databases server*).

---

**Note:** The entire alias should be in capital letters (for example, POWERPIVOT), not lowercase or mixed case.

---

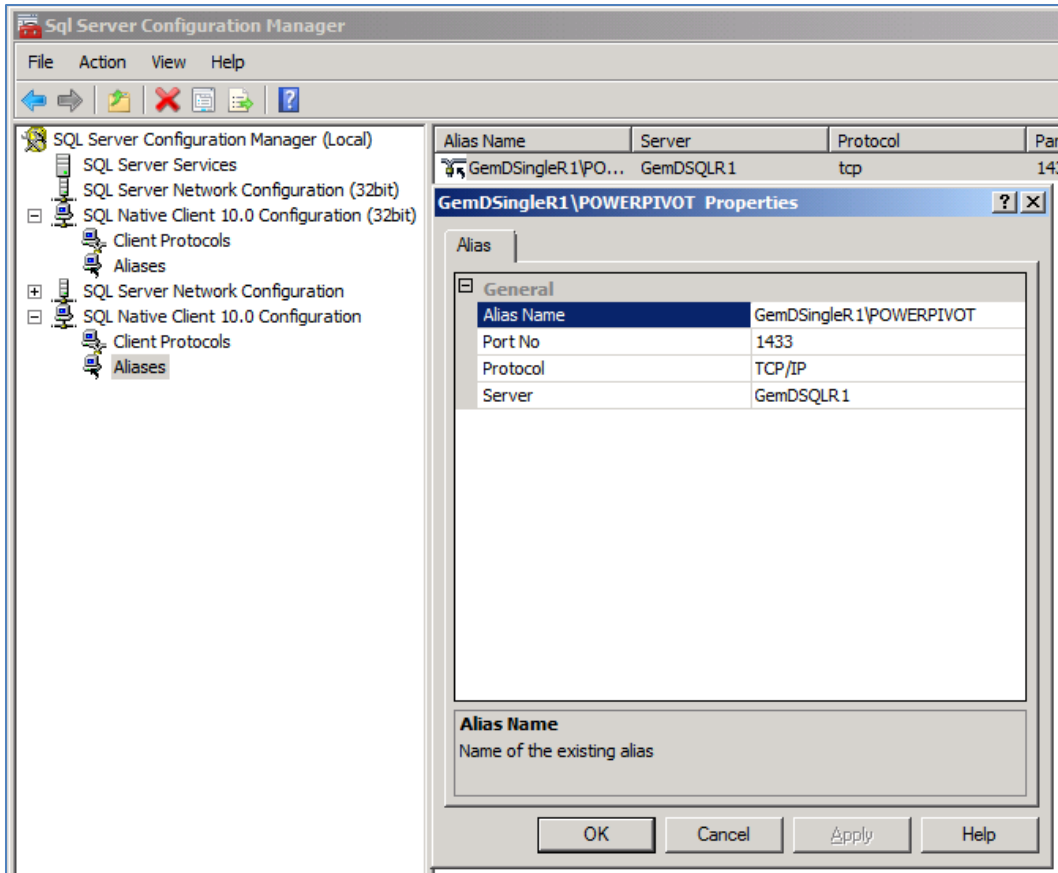


Figure 7: Aliasing the database names to the new server

- 5) Because the instance of SQL Server is no longer used on the original *PowerPivot for SharePoint Single Server* (GemDSingleR1') instance, use the SQL Server Configuration Manager to shut down the SQL Server instance (that is, the GemDSingleR1\POWERPIVOT instance) and change the Start Mode from Automatic to Disabled so that it will not start if the server is rebooted. When time permits, disable all \POWERPIVOT instances services altogether and remove these instances of SQL Server on the *PowerPivot for SharePoint Single Server* (GemDSingleR1') instance. If you want to remove the databases, use the script in Appendix C.
  
- 6) Restart IIS on your original *PowerPivot for SharePoint Single Server* (GemDSingleR1') instance (which is now your *PowerPivot for SharePoint WFE/App server*) by typing **iisreset /start** at an administrative command prompt.

From this point forward, all transactions to the *PowerPivot for SharePoint WFE/App Server* (GemDSingleR1') are utilizing the instance of SQL Server on your *dedicated SharePoint Databases server* (GemDSQLR1). Note that within the **Manage Databases Upgrade Status** page it will still state that the instance of SQL Server is the original *PowerPivot for SharePoint Single Server* (GemDSingleR1') instance, but because of the aliasing in step 4, the transactions actually occur on the *dedicated SharePoint Databases server* instance (GemDSQLR1). This is the primary limitation to this aliasing instead of migrating SharePoint, because you cannot add this to the farm.

---

**Lesson Learned** You may also be able to improve performance even more by using the SQL Server 2008 R2 Remote Blob Store feature; for more information, see [Why you may want to use RBS for your PowerPivot for SharePoint Farm](http://dennyglee.com/2010/02/23/why-you-may-want-to-use-rbs-for-your-powerpivot-for-sharepoint-farm/) (<http://dennyglee.com/2010/02/23/why-you-may-want-to-use-rbs-for-your-powerpivot-for-sharepoint-farm/>).

---

## Adding a SharePoint Application Server

After you have the SharePoint databases on a *dedicated SharePoint Databases server* (GemDSQLR1) by following the process described earlier in this paper, the next step is to move the applications, primarily Excel Calculation Services and PowerPivot Services (PowerPivot System Service, Analysis Services in VertiPaq mode), to a *dedicated SharePoint application server* (GemDAppR1).

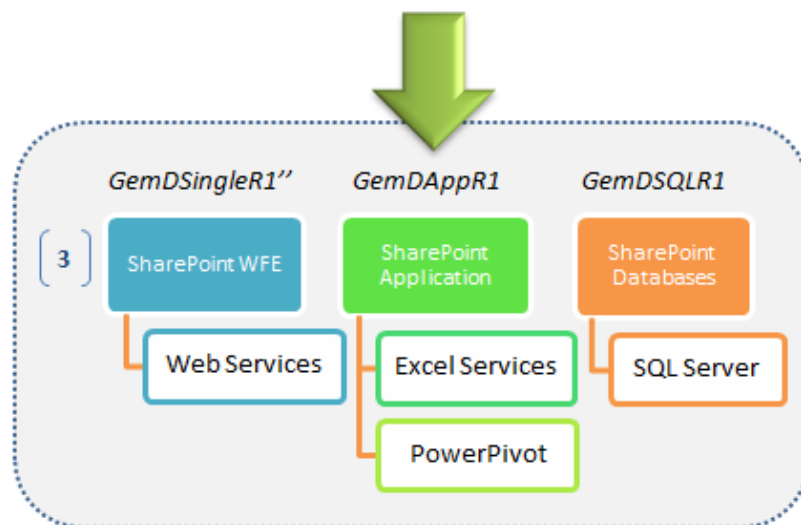


Figure 8: Adding a new SharePoint application server

You now have a basic three-server farm that you can further expand as needed.

## Moving the Applications

This section references the instructions in [Existing Farm Install](#), as well as steps discussed in previous sections of this white paper. Perform the following steps on the new server:

- 1) Install SQL Server Management and client tools on the new application server (GemDAppR1). Then create the alias defined earlier in step 4 of “Move the Databases.”
- 2) Follow **Step 6: Install SharePoint 2010 on PowerPivotServer1** (GemDAppR1) in “Existing Farm Install.”
- 3) Follow **Step 7: Run the SharePoint Configuration Wizard on PowerPivotServer1** (GemDAppR1) in “Existing Farm Install.”
- 4) Follow **Step 10: Run Existing Farm Setup on PowerPivotServer1** (GemDAppR1) in “Existing Farm Install.”
- 5) After you complete step 10 in “Existing Farm Install”, go to **Services on Server** in SharePoint Central Administration and shut down the following services on the *PowerPivot for SharePoint WFE/App Server*:
  - Excel Calculation Services
  - SQL Server Analysis Services
  - SQL Server PowerPivot System Service

Your server is now a WFE server only.

Now, GemDSingleR1' (which was a WFE/ application server) is now GemDSingleR1'' (now a SharePoint WFE).

- 6) When time permits, remove the Analysis Server instance on the *PowerPivot for SharePoint WFE Server* (that is, the GemDSingleR1'' \POWERPIVOT instance).
- 7) Use the **Services on Server** in SharePoint Central Administration to stop the Microsoft SharePoint Foundation Web Application on the new application server (GemDAppR1).

# Basic Three-Server PowerPivot for SharePoint Farm

If you have followed the steps so far, you have set up and configured your three-server PowerPivot for SharePoint farm with your dedicated SharePoint WFE, dedicated SharePoint application server, and dedicated SharePoint SQL Server databases server.

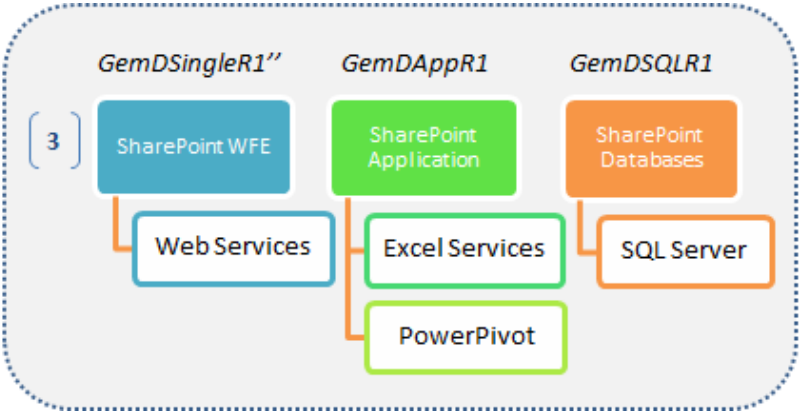


Figure 9: Basic three-server farm

With this configuration, you have the basic PowerPivot for SharePoint infrastructure. As more workbooks are created and viewed, more resources will be required to render and calculate your PowerPivot workbooks within SharePoint. In most cases, you will add more SharePoint Application servers running PowerPivot and its dependent services.

## Conclusion

The easiest way to jump-start your organization’s implementation of PowerPivot for SharePoint is to build a single-server SQL Server PowerPivot for SharePoint installation. But as support for more concurrent users and larger PowerPivot workbooks becomes necessary, your organization will need to scale out your installation to alleviate resource bottlenecks. This white paper provides you with steps that help you transition from a single-server setup to a three-server PowerPivot for SharePoint farm. We hope that by using the information in this white paper, your organization will benefit from an easier initial startup, some flexibility in timing the transition to a multi-server farm, and a less complicated process when you do decide to make that move.

## Appendix

The Appendix section contains the scripts that you can use to perform the database actions required to transition from a single-server installation to a multi-server farm:

- Appendix A and B contain move database scripts to facilitate steps 3a, 3b, and 3c in “Move the Databases.”
- Appendix C contains the script to drop the moved databases on the destination server. This step may not be required for your installation.

### Appendix A

Appendix A and B contain move database scripts to facilitate steps 3a, 3b, and 3c in “Move the Databases.” The following script can be run on the server that is the source for the databases. You need to either create the folder `C:\dbBackSP\` or change the value of `@bakPath` to an existing folder.

```
-----  
-- Create a database to hold move information.  
CREATE DATABASE [dbMoveSP]  
GO  
  
ALTER DATABASE [dbMoveSP] SET RECOVERY SIMPLE  
GO  
  
DECLARE @bakPath NVARCHAR(256) -- path for backup files  
SET @bakPath = N'C:\dbBackSP\  
  
DECLARE @dbName NVARCHAR(128) -- database name  
DECLARE @bakFileName NVARCHAR(256) -- file name for backup  
  
-----  
-- Store database names to be moved  
SELECT name  
into dbMoveSP.dbo.DBNames  
FROM master.dbo.sysdatabases  
WHERE name NOT IN ('master', 'model', 'msdb', 'tempdb')  
  
-----  
-- Store security information  
CREATE TABLE dbMoveSP.dbo.DBLogins (Login NVARCHAR(2000))  
  
DECLARE @name sysname  
DECLARE @type varchar (1)  
DECLARE @hasaccess int  
DECLARE @denylogin int  
DECLARE @is_disabled int  
DECLARE @SID_varbinary varbinary (85)
```

```

DECLARE @SID_string varchar (514)
DECLARE @tmpstr varchar (1024)
DECLARE @is_policy_checked varchar (3)
DECLARE @is_expiration_checked varchar (3)
DECLARE @defaultdb sysname

DECLARE login_curs CURSOR FOR
    SELECT p.sid, p.name, p.type, p.is_disabled,
p.default_database_name, l.hasaccess, l.denylogin FROM
sys.server_principals p LEFT JOIN sys.syslogins l
    ON ( l.name = p.name ) WHERE p.type IN ( 'G', 'U' ) AND p.name
not like 'nt%'

OPEN login_curs

FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @type,
@is_disabled, @defaultdb, @hasaccess, @denylogin
IF (@@fetch_status = -1)
BEGIN
    PRINT 'No login(s) found.'
    CLOSE login_curs
    DEALLOCATE login_curs
-- RETURN -1
END
ELSE
BEGIN
SET @tmpstr = '/* sp_help_revlogin script '
PRINT @tmpstr
SET @tmpstr = '** Generated ' + CONVERT (varchar, GETDATE()) + ' on '
+ @@SERVERNAME + ' */'
PRINT @tmpstr
PRINT ''
WHILE (@@fetch_status <> -1)
BEGIN
    IF (@@fetch_status <> -2)
    BEGIN
        PRINT ''
        SET @tmpstr = '-- Login: ' + @name
        PRINT @tmpstr
        IF (@type IN ( 'G', 'U'))
        BEGIN -- NT authenticated account/group

            SET @tmpstr = 'CREATE LOGIN ' + QUOTENAME( @name ) + ' FROM
WINDOWS WITH DEFAULT_DATABASE = [' + @defaultdb + ']'
            END
        IF (@denylogin = 1)
        BEGIN -- login is denied access

```

```

        SET @tmpstr = @tmpstr + '; DENY CONNECT SQL TO ' + QUOTENAME (
@name )
    END
    ELSE IF (@hasaccess = 0)
    BEGIN -- login exists but does not have access
        SET @tmpstr = @tmpstr + '; REVOKE CONNECT SQL TO ' + QUOTENAME (
@name )
    END
    IF (@is_disabled = 1)
    BEGIN -- login is disabled
        SET @tmpstr = @tmpstr + '; ALTER LOGIN ' + QUOTENAME ( @name ) +
' DISABLE'
    END
    PRINT @tmpstr
    INSERT INTO dbMoveSP.dbo.DBLogins VALUES (@tmpstr)
END

    FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @type,
@is_disabled, @defaultdb, @hasaccess, @denylogin
    END
CLOSE login_curs
DEALLOCATE login_curs
END

```

-----  
-- Back up the databases

```

DECLARE db_cursor CURSOR FOR
SELECT name
FROM dbMoveSP.dbo.DBNames --master.dbo.sysdatabases

OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @dbName

WHILE @@FETCH_STATUS = 0
BEGIN

    SET @bakFileName = @bakPath + @dbName + N'.BAK'
    BACKUP DATABASE @dbName TO DISK = @bakFileName
        WITH NOFORMAT, NOINIT,
            NAME = @dbName, SKIP, NOREWIND, NOUNLOAD
    FETCH NEXT FROM db_cursor INTO @dbName

END

CLOSE db_cursor
DEALLOCATE db_cursor
GO

```

## Appendix B

Appendix A and B contain move database scripts to facilitate steps 3a, 3b, and 3c in “Move the Databases.” The following script can be run on the server that is the destination for the databases. You need to first copy the folder that was used for @bakPath in the script in Appendix A to the destination server then make sure that @bakPath in this script is set correctly. Also either create folder C:\dbMoveSP\ or change the value of @sqlPath to an existing folder.

```
USE [master]
GO

-----
-- Restore move database -- dbMoveSP
DECLARE @bakPath      NVARCHAR(256) -- path for backup files
SET @bakPath = N'C:\dbBackSP\'

DECLARE @sqlPath      NVARCHAR(256) -- path for database files
SET @sqlPath = N'C:\dbMoveSP\'

DECLARE @dbName       NVARCHAR(128) -- database name
DECLARE @bakFileName  NVARCHAR(256) -- file name for backup
DECLARE @ldfFileName  NVARCHAR(256) -- file name for log
DECLARE @mdfFileName  NVARCHAR(256) -- file name for data
DECLARE @ldfName      NVARCHAR(256) -- logical name for log
DECLARE @mdfName      NVARCHAR(256) -- logical name for data
DECLARE @sql          NVARCHAR(256)
DECLARE @login        NVARCHAR(2000) -- database name

SET @dbName = N'dbMoveSP'
SET @ldfName = @dbName + N'_log'
SET @mdfName = @dbName
SET @bakFileName = @bakPath + @dbName + N'.BAK'
SET @ldfFileName = @sqlPath + @ldfName + N'.LDF'
SET @mdfFileName = @sqlPath + @mdfName + N'.MDF'

RESTORE DATABASE @dbName FROM DISK = @bakFileName
    WITH Move @ldfName TO @ldfFileName
        ,Move @mdfName TO @mdfFileName

-----
-- Create logins
SET @sql = 'DECLARE db_cursor CURSOR FOR
SELECT login
FROM '
+ @dbName + N'.dbo.DBLogins'

print @sql
exec (@sql)
```

```

OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @login

WHILE @@FETCH_STATUS = 0
BEGIN
    EXEC (@login)
    FETCH NEXT FROM db_cursor INTO @login

END

CLOSE db_cursor
DEALLOCATE db_cursor

-----
-- Restore SharePoint databases
SET @sql = 'DECLARE db_cursor CURSOR FOR
SELECT name
FROM '
+ @dbName + N'.dbo.DBNames'
+ ' WHERE name NOT IN ('' + @dbName + '')'

print @sql
exec (@sql)

OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @dbName

WHILE @@FETCH_STATUS = 0
BEGIN
    SET @ldfName = @dbName + N'_log'
    SET @mdfName = @dbName
    SET @bakFileName = @bakPath + @dbName + N'.BAK'
    SET @ldfFileName = @sqlPath + @ldfName + N'.LDF'
    SET @mdfFileName = @sqlPath + @mdfName + N'.MDF'

    print @bakFileName
    RESTORE DATABASE @dbName From DISK = @bakFileName
        WITH Move @ldfName TO @ldfFileName
            ,Move @mdfName TO @mdfFileName

    FETCH NEXT FROM db_cursor INTO @dbName

END

CLOSE db_cursor
DEALLOCATE db_cursor
GO

```

## Appendix C

The following script can be used to drop the moved databases on the destination server if necessary. It cannot be run on the source server after step 4 to delete the databases because the alias created in step 4 would cause it to actually try to drop the databases on the destination server. However, it could be run from another machine (that does not have the alias) against the source instance of SQL Server (in our example, GemDSingleR1\POWERPIVOT).

Before you run this script, enable the TCP/IP and Named Pipes protocols, and allow access through the firewall on the source server (for more information, see [How to: Configure a Windows Firewall for Database Engine Access](http://msdn.microsoft.com/en-us/library/ms175043.aspx) at <http://msdn.microsoft.com/en-us/library/ms175043.aspx>).

If you run this script on the source server before step 4, use Services.msc to temporarily stop the SharePoint services.

```
USE [master]
GO
-----
-- Drop moved SharePoint databases
DECLARE @dbName      NVARCHAR(128)
DECLARE @sql         NVARCHAR(256)

SET @dbName = N'dbMoveSP'
SET @sql = 'DECLARE db_cursor CURSOR FOR SELECT name FROM '
+ @dbName + N'.dbo.DBNames'
+ ' WHERE name NOT IN ('' + @dbName + '')'
exec (@sql)

OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @dbName

WHILE @@FETCH_STATUS = 0
BEGIN
    SET @sql = N'DROP DATABASE ' + @dbName
    exec (@sql)
    FETCH NEXT FROM db_cursor INTO @dbName
END

CLOSE db_cursor
DEALLOCATE db_cursor

SET @dbName = N'dbMoveSP'
SET @sql = N'DROP DATABASE ' + @dbName

exec (@sql)
GO
```