Configure Kerberos Authentication for SharePoint 2010 Products

Tom Wisnowski

Guide

Microsoft

Configure Kerberos Authentication for SharePoint 2010 Products

Tom Wisnowski

Contributors: Philippe-Joseph Arida, Luca Bandinelli, Kevin Donovan, PejJavaheri , Denny Lee, Cephas Lin, Dave Manning, Carl Rabeler, PrashShirolkar, Norm Warren, Josh Zimmerman

Summary: This document covers the concepts of identity in SharePoint 2010 products, how Kerberos authentication plays a critical role in authentication and delegation in business intelligence scenarios, and the situations where Kerberos authentication should be leveraged or may be required in solution designs. It also covers how to configure Kerberos authentication end-to-end within your environment, including scenarios which use various service applications in SharePoint Server. Additional tools and resources are described to help you test and validate Kerberos configuration.

Category: Guide
Applies to: SharePoint 2010
Source: White paper (link to source content)
E-book publication date: May 2012
220 pages

Microsoft

Table of Contents

Configure Kerberos authentication for SharePoint 2010 Products

Published: July 15, 2010

This document gives you information that will help you understand the concepts of identity in Microsoft SharePoint 2010 Products, how Kerberos authentication plays a very important role in authentication and delegation scenarios, and the situations where Kerberos authentication should be used or may be required in solution designs. Scenarios include business intelligence implementations which secure access to external data sources such as SQL Server.

The document also shows how to configure Kerberos authentication end-to-end within your environment, including scenarios that use various service applications in Microsoft SharePoint Server. Additional tools and resources are described to help you test and validate Kerberos configuration. The "Step-by-Step Configuration" sections of this document cover the following scenarios for SharePoint Server 2010.

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP
- Scenario 3: Identity Delegation for SQL Analysis Services
- Scenario 4: Identity Delegation for SQL Reporting Services
- Scenario 5: Identity Delegation for Excel Services
- Scenario 6: Identity Delegation for PowerPivot for SharePoint
- Scenario 7: Identity Delegation for Visio Services
- Scenario 8: Identity Delegation for PerformancePoint Services
- Scenario 9: Identity Delegation for Business Connectivity Services

The same information about Configuring Kerberos authentication for SharePoint 2010 Products is available is also available as a set of articles here in the TechNet Library. It begins here: Overview of Kerberos authentication for Microsoft SharePoint 2010 Products.

Overview of Kerberos authentication for Microsoft SharePoint 2010 Products

Published: December 2, 2010

Microsoft SharePoint 2010 Products introduce significant improvements in how identity is managed in the platform. It is very import to understand how these changes affect solution design and platform configuration to enable scenarios that require user identity to be delegated to integrated systems. The Kerberos version 5 protocol plays a key role in enabling delegation and sometimes may be required in these scenarios.

This set of articles gives you information that helps you do the following:

- Understand the concepts of identity in SharePoint 2010 Products
- Learn how Kerberos authentication plays a very important role in authentication and delegation scenarios
- Identify the situations where Kerberos authentication should be leveraged or may be required in solution designs
- Configure Kerberos authentication end-to-end within your environment, including scenarios that use various service applications in SharePoint Server
- Test and validate that Kerberos authentication is configured correctly and working as expected
- Find additional tools and resources to help you configure Kerberos authentication in your environment

This set of articles is divided in two major sections:

- This overview of Kerberos authentication in SharePoint 2010 Products

  This article contains conceptual information about how to manage identity in SharePoint 2010 Products, the Kerberos protocol, and how Kerberos authentication plays a key role in SharePoint 2010 solutions.

- Step-by-step configuration

  This group of articles discusses the steps that are required to configure Kerberos authentication and delegation in various SharePoint solution scenarios.

Who should read these articles about Kerberos authentication?

Identity and delegation in SharePoint 2010 Products is a broad topic, with many facets and depths of understanding. This set of articles addresses the topic from both conceptual and technical levels and is written to address the needs of various audiences:

Beginning to end

"Tell me everything there is to know about Identity and Kerberos authentication in SharePoint 2010 Products"

If you are only starting out and learning about SharePoint 2010 Products, Kerberos authentication, and claims authentication, you will want to the read the first section of this document. It covers the basic concepts of identity and delegation and offers primers about Claims and Kerberos authentication. Be sure to follow the links to external articlesand additional information to build a solid foundation of knowledge before continuing on to the step-by-step configuration articles.

Upgrading from Office SharePoint Server 2007

"Tell me what is changed from 2007 and what I should prepare for in upgrading to 2010"

If you have an existing Microsoft Office SharePoint Server 2007 environment already configured to use Kerberos authentication and Kerberos delegation, you should read the following articles:

- Identity scenarios in SharePoint 2010 Products
- Claims primer
- Kerberos authentication changes in Windows 2008 R2 and Windows 7
- Kerberos configuration changes in SharePoint 2010 Products
- Considerations when you are upgrading from Office SharePoint Server 2007

If you have additional questions about how to configuration delegation for a particular feature or scenario, read the step-by-step configuration articles, especially the configuration checklists. This will help you ensure that your environment is configured correctly after upgrade.

Step-by-step walkthrough

"I want detailed step-by-step instructions on how to configure Kerberos delegation in SharePoint Server and applicable SharePoint Server service applications"

The step-by-step configuration articles cover several SharePoint 2010 Products scenarios which can be configured to use Kerberos delegation. Each scenario is covered in detail, including a configuration checklist and step-by-step instructions to help you successfully configure Kerberos authentication in your environment. The scenarios covered include the following:

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP
- Scenario 3: Kerberos Authentication for SQL Analysis Services
- Scenario 4: Identity Delegation for SQL Reporting Services
- Scenario 5: Identity Delegation for Excel Services
- Scenario 6: Identity Delegation for PowerPivot for SharePoint 2010
- Scenario 7: Identity Delegation for Visio Services
- Scenario 8: Identity Delegation for Performance Point Services
- Scenario 9: Identity Delegation for Business Connectivity Services

Be sure to thoroughly review the first core configuration scenario, because it is a prerequisite for all the scenarios that follow.

Note:

The scenarios include "SetSPN" commands that you may choose to copy from this document and paste in a Command Prompt window. These commands include hyphen characters. Microsoft Word has an AutoFormat feature that tends to convert hyphens to dash characters. If you have this feature turned on in Word and then do a copy-and-paste operation, the commands will not work correctly. Change the dashes to hyphens to fix this error. To turn off this AutoFormat feature in Word, select **Options** from the **File** menu, click the **Proofing** tab, and then open the **Auto Correct** dialog box.

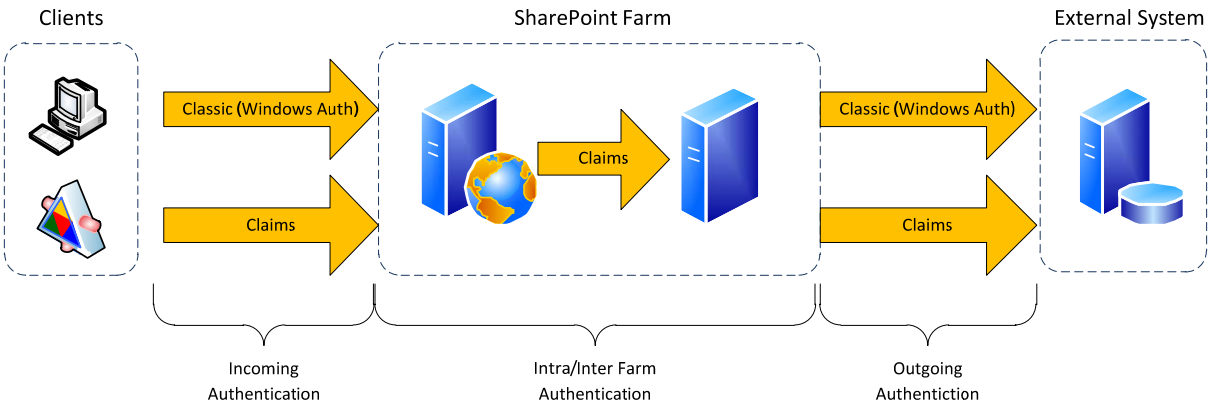
Existing SharePoint 2010 Product environments

"I have an existing SharePoint 2010 Product environment and I cannot seem to get Kerberos authentication working. How do I validate and debug my configuration?"

The Step-by-step configuration articles contain several checklists to help triage your environment in various scenarios. Pay special attentions to Scenario 1, Core configuration, which covers basic tools and techniques to triage Kerberos configuration.

Identity scenarios in SharePoint 2010 Products

When learning about identity in the context of authentication in SharePoint 2010 Products, you can conceptually look at how the platform handles identity in three key scenarios: Incoming authentication, inter/intra-farm authentication and outgoing authentication.



Incoming Identity

The incoming authentication scenario represents the means in which a client presents its identity to the platform, or in other words *authenticates* with the web application or web service. SharePoint Server will use the client's identity to authorize the client to access SharePoint Server secured resources such as web pages, documents, and so on.

SharePoint 2010 Products support two modes in which a client can authenticate with the platform: Classic mode and Claims mode.

Classic mode

Classic mode allows the typical Internet Information Services (IIS) authentication methods that you may already be familiar with from previous versions of SharePoint Server. When a SharePoint Server 2010 Web Application is configured to use classic mode, you have the option of using the following IIS authentication methods:

Integrated Windows authentication

Integrated Windows authentication enables Windows clients to seamlessly authenticate with SharePoint Server without having to manually provide credentials (user name/password). Users accessing SharePoint Server from Internet Explorer will authenticate by using the credentials that the Internet Explorer process is running under — by default the credentials that the user used to log on to the desktop. Services or applications that access SharePoint Server in Windows integrated mode attempt to authenticate by using the credentials of the running thread, which, by default, is the identity of the process.

NTLM

NT LAN Manager (NTLM) is the default protocol type when Integrated Windows authentication is selected. This protocol takes advantage of a three-part challenge-response sequence to authenticate clients. For more information about NTLM, see Microsoft NTLM (http://go.microsoft.com/fwlink/?LinkId=196643).

Pros:

- It is easy to configure and typically requires no additional infrastructure/environment configuration to function
- It works when the client is not part of the domain, or is not in a domain trusted by the domain that SharePoint Server resides in

Cons:

- It requires SharePoint Server to contact the domain controller every time that a client authentication response needs validation, increasing traffic to the domain controllers.
- It does not allow delegation of client credentials to back-end systems, otherwise known as the double-hop rule. It is a proprietary protocol.
- It is a proprietary protocol.
- It does not support server authentication.
- It is considered less secure than Kerberos authentication

Kerberos protocol

The Kerberos protocol is a more secure protocol that supports ticketing authentication. A Kerberos authentication server grants a ticket in response to a client computer authentication request, if the request contains valid user credentials and a valid Service Principal Name (SPN). The client computer then uses the ticket to access network resources. To enable Kerberos authentication, the client and server computers must have a trusted connection to the domain Key Distribution Center (KDC). The KDC distributes shared secret keys to enable encryption. The client and server computers must also be able to access Active Directory directory services. For Active Directory, the forest root domain is the center of Kerberos authentication referrals. For more information about the Kerberos protocol, see How the Kerberos Version 5 Authentication Protocol Works(http://go.microsoft.com/fwlink/?LinkId=196644) and Microsoft Kerberos. (http://go.microsoft.com/fwlink/?LinkId=196645)

Pros:

- Most secure Integrated Windows authentication protocol
- Allows delegation of client credentials
- Supports mutual authentication of clients and servers
- Produces less traffic to domain controllers
- Open protocol supported by many platforms and vendors

Cons:

- Requires additional configuration of infrastructure and environment to function correctly
- Requires clients have connectivity to the KDC (Active Directory domain controller in Windows environments) over TCP/UDP port 88 (Kerberos), and TCP/UDP port 464 (Kerberos Change Password – Windows)

Other methods

In addition to NTLM and Kerberos authentication, SharePoint Server supports other kinds of IIS authentication such as basic, digest, and certificate-based authentication, which are not covered in this document. For more information about how these protocols function, see Authentication Methods Supported in IIS 6.0 (IIS 6.0) (http://go.microsoft.com/fwlink/?LinkId=196646).

Claims-based authentication

Support for claims authentication is a new feature in SharePoint 2010 Products and is built on Windows Identity Foundation (WIF). In a claims model, SharePoint Server accepts one or more *claims* about an authenticating client to identify and authorize the client. The claims come in the form of SAML tokens and are facts about the client stated by a trusted authority. For example, a claim could state, "Bob is a member of the Enterprise Admins group for the domain Contoso.com."If this claim came from a provider trusted by SharePoint Server, the platform could use this information to authenticate Bob and to authorize him to access SharePoint Server resources. For more information about claims authentication, see A Guide to Claims-based Identity and Access Control (http://go.microsoft.com/fwlink/?LinkID=187911).

The kind of claims that SharePoint 2010 Products support for incoming authentication areWindows-Claims, forms-based authentication-Claims, and SAML-Claims.

Windows-Claims

In the Windows-claims mode sign in, SharePoint Server authenticates the client using standard Integrated Windows authentication (NTLM/Kerberos), and then translates the resulting Windows Identity into a Claims Identity.

Forms-based authentication Claims

In Forms-based authentication claims mode, SharePoint Server redirects the client to a logon page that hosts the standard ASP.NET logon controls. The page authenticates the client by using ASP.NET membership and role providers, similar to how forms-based authentication functioned in Office SharePoint Server 2007. After the identity object that represents the user is created, SharePoint Server then translates this identity into a claims identity object.

SAML-Claims

In SAML-Claims mode, SharePoint Server accepts SAML tokens from a trusted external Security Token Provider (STS). When the user attempts to log on, see comment is directed to an external claims provider (for example, Windows Live ID claims provider) which authenticates the user and produces a SAML token. SharePoint Server accepts and processes this token, augmenting the claims and creating a claims identity object for the user.

For more information about claims-based authentication in SharePoint 2010 Products, seeSharePoint Claims-Based Identity.

Note about incoming claims authentication and the Claims to Windows Token Service (C2WTS)

Some service applications require that you use the Windows Identity Foundation (WIF) Claims to Windows Token Service (C2WTS) to translate claims within the farm to Windows credentials for outbound authentication. It is important to understand that C2WTS only functions if the incoming authentication method is either classic mode or Windows claims. Ifclaims is configured, the C2WTS requires only Windows claims; the web application cannot use multiple forms of claims on the web application, otherwise the C2WTS will not function.

Identity within a SharePoint 2010 Products environment

SharePoint 2010 Products environments use claims authentication for intra- and inter-farm communications with most SharePoint service applications and SharePoint

integrated products regardless of the incoming authentication mechanism used. This means that even where classic authentication is used to authenticate with a particular web application, SharePoint Products convert the incoming identity into a claims identity to authenticate with SharePoint Service Applications and products that are claims-aware. By standardizing on the claims model for intra/inter farm communications, the platform can abstract itself from the incoming protocols that are used.

Note:

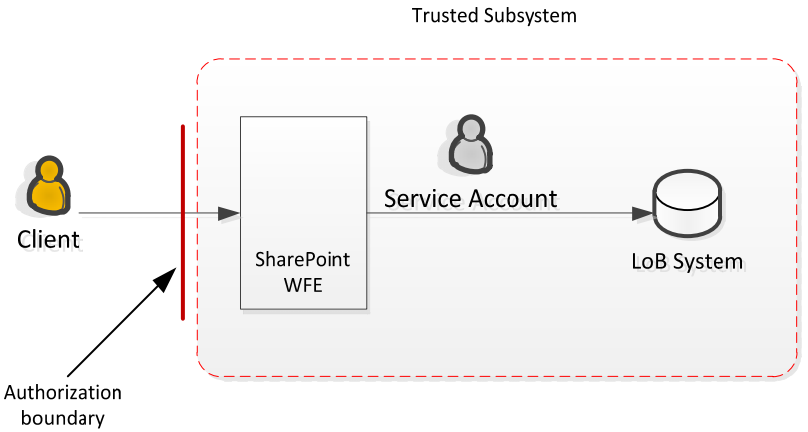
Some products integrated with SharePoint Server, such as SQL Server Reporting Services, are not claims-aware and do not take advantage of the intra-farm claims authentication architecture. SharePoint Server may also rely on classic Kerberos delegation and claims in other scenarios, for example when the RSS viewer web part is configured to consume an authenticated feed. Refer to each product or service application's documentation to determine whether it can support claims-based authentication and identity delegation.

Outbound identity

Outbound identity in SharePoint 2010 Products represents the scenarios where services within the farm have to authenticate with external line-of-business systems and services. Depending on the scenario, authentication can be performed in one of two basic conceptual models:

Trusted subsystem

In the trusted subsystem, the front-end service authenticates and authorizes the client, and then authenticates with additional back-end services without passing the client identity to the back-end system. The back-end system *trusts* the front-end service to do authentication and authorization on its behalf. The most common way to implement this model is to use shared service account to authenticate with the external system:

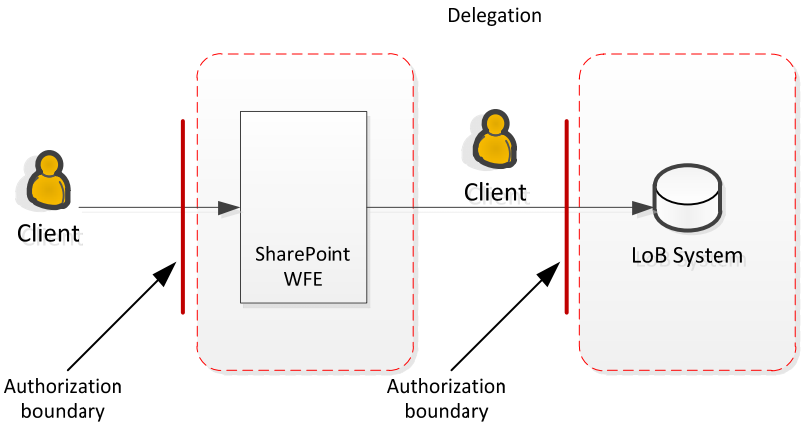


In SharePoint Server, this model can be implemented in various ways:

- Using the IIS application pool identity — usually achieved by running code in the web application that elevates permissions while making a call to an external system. Other methods such as using RevertToSelf can also use the application pool's identity to authenticate with external systems.
- Using a service account — typically achieved by storing application credentials in the Secure Store then using those credentials to authenticate with an external system. Other methods include storing the service account credentials in other ways such as embedded connection strings.
- Anonymous Authentication — this is where the external system requires no authentication. Therefore the front-end SharePoint Server service does not have to pass any identity to the back-end system.

Delegation

In the Delegation model, the front-end service first authenticates the client, and then uses the client's identity to authenticate with another back-end system that performs its own authentication and authorization:



In SharePoint 2010 Products, this model can be implemented in various ways:

- Kerberos delegation — If the client authenticates with the front-end service by using Kerberos authentication, Kerberos delegation can be used to pass the client's identity to the back-end system.
- Claims — claims authentication allows the client's claims to be passed between services as long as there is trust between the two services and both are claims-aware.

Note:

Currently, most of the service applications that are included with SharePoint Server do not allow for outbound claims authentication, but outbound claims is a platform capability that will be taken advantage of in the future. Further, many of the most common line-of-business systems today do not support incoming claims authentication, which means that using outbound claims authentication may not be possible or will require additional development to work correctly.

Delegation across domain and forest boundaries

The scenarios in this set of articles about Kerberos authentication require that the SharePoint Server service and external data sources reside in the same Windows domain, which is required for Kerberos constrained delegation. The Kerberos protocol supports two kinds of delegation, basic (unconstrained) and constrained. Basic Kerberos delegation can cross domain boundaries in a single forest, but cannot cross a forest

boundary regardless of trust relationship. Kerberos constrained delegation cannot cross domain or forest boundaries in any scenario.

Some SharePoint Server services can be configured to use basic Kerberos delegation, but other services require that you use constrained delegation. Any service that relies on the Claims to Windows token service (C2WTS) must use Kerberos constrained delegation to allow the C2WTS to use Kerberos protocol transition to translate claims into Windows credentials.

The following service applications and products require the C2WTS and Kerberos constrained delegation:

- Excel Services
- PerformancePoint Services
- Visio Services

The following service applications and products are not affected by these requirements, and therefore can use basic delegation, if it is required:

- Business Data Connectivity service and Microsoft Business Connectivity Services
- InfoPath Forms Services
- Access Services
- Microsoft SQL Server Reporting Services (SSRS)
- Microsoft Project Server 2010

The following service application does not allow delegation of client credentials and therefore is not affected by these requirements:

- Microsoft SQL Server PowerPivot for Microsoft SharePoint

Claims primer

For an introduction to Claims concepts and Claims base authentication, see An Introduction to Claims (http://go.microsoft.com/fwlink/?LinkId=196648) and SharePoint Claims-Based Identity (http://go.microsoft.com/fwlink/?LinkID=196647).

Kerberos protocol primer

For a conceptual overview of the Kerberos protocol, see Microsoft Kerberos (Windows) (http://go.microsoft.com/fwlink/?LinkID=196645), Kerberos Explained (http://go.microsoft.com/fwlink/?LinkId=196649), and Ask the Directory Services Team: Kerberos for the Busy Admin (http://go.microsoft.com/fwlink/?LinkId=196650).

Benefits of the Kerberos protocol

Before examining the details of how one configures SharePoint Server (or any web application) to use the Kerberos protocol, let's talk about the Kerberos protocol generally and why you might want to use it.

Typically there are three main reasons to use the Kerberos protocol:

1. **Delegation of client credentials**—The Kerberos protocol allows a client's identity to be impersonated by a service to allow the impersonating service to pass that identity to other network services on the client's behalf. NTLM does not allow this delegation. (This limitation NTLM is called the "double-hop rule"). Claims authentication, like Kerberos authentication, can be used to delegate client credentials but requires the back-end application to be claims-aware.

2. **Security**—Features such as AES encryption, mutual authentication, support for data integrity and data privacy, just to name a few, make the Kerberos protocol more secure than its NTLM counterpart.

3. **Potentially better performance** — Kerberos authentication requires less traffic to the domain controllers compared with NTLM (depending on PAC verification, see Microsoft Open Specification Support Team Blog: Understanding Microsoft Kerberos PAC Validation). If PAC verification is disabled or not needed, the service that authenticates the client does not have to make an RPC call to the DC (see: You experience a delay in the user-authentication process when you run a high-volume server program on a domain member in Windows 2000 or Windows Server 2003). Kerberos authentication also requires less traffic between client and server compared with NTLM. Clients can authenticate with web servers in two request/responses vs. the typical three-leg handshake with NTLM. However, this improvement is typically not noticed on low latency networks on a per-transaction basis, but can typically be noticed in overall system throughput. Remember that many environmental factors can affect authentication performance; therefore Kerberos authentication and NTLM should be performance-tested in your own environment before you determine whether one method performs better than the other.

This is an incomplete list of the advantages of using the Kerberos protocol. There are other reasons like mutual authentication, cross platform interoperability, and transitive cross domain trust, to name a few. However, in most cases one typically finds delegation and security to be the primary drivers in adoption of the Kerberos protocol.

Kerberos delegation, constrained delegation, and protocol transition

The Kerberos version 5protocol on the Windows platform supports two kinds of identity delegation: basic (unconstrained) delegation and constrained delegation:

Type	Advantages	Disadvantages
Basic delegation	• Can cross domain boundaries in a single forest • Requires less configuration than constrained delegation.	• Does not support protocol transition • Secure. If the front-end service is compromised, client identity can be delegated to any service in the forest that accepts Kerberos authentication.
Constrained delegation	• Can transition non-Kerberos incoming authentication protocol to Kerberos (example: NTLM to Kerberos, Claims to Kerberos) • More secure. Identities can only be delegated to specified service.	• Cannot cross domain boundaries • Requires additional setup configuration

Kerberos enabled services can delegate identity multiple times across multiple services and multiple hops. As an identity travels from service to service, the delegation method can change from Basic to Constrained but not in reverse. This is an important design detail to understand: if a back-end service requires Basic delegation (for example to

delegate across a domain boundary), all services in front of the back-end service must use basic delegation. If any front-end service uses constrained delegation, the back-end service cannot change the constrained token into an unconstrained token to cross a domain boundary.

Protocol transition allows a Kerberos enabled authenticating service (front-end service) to convert a non-Kerberos identity into a Kerberos identity that can be delegated to other Kerberos enabled services (back-end service). Protocol transition requires Kerberos constrained delegation and therefore protocol-transitioned identities cannot cross domain boundaries. Depending on the user rights of the front-end service, the Kerberos ticket returned by protocol transition can be an identification token or an impersonation token. For more information about constrained delegation and protocol transition, see the following articles:

- Kerberos Protocol Transition and Constrained Delegation (http://technet.microsoft.com/en-us/library/cc739587(WS.10).aspx)
- Protocol Transition with Constrained Delegation Technical Supplement (http://msdn.microsoft.com/en-us/library/ff650469.aspx)
- Kerberos Constrained Delegation May Require Protocol Transition in Multi-hop Scenarios (http://support.microsoft.com/kb/2005838)

As a general best practice, if Kerberos delegation is required, one should use constrained delegation, if it is possible. If delegation across domain boundaries is required, then all services in the delegation path must use basic delegation.

Kerberos authentication changes in Windows 2008R2 and Windows 7

Windows Server 2008 R2 and Windows 7 introduce new features to Kerberos authentication. For an overview of the changes, see Changes in Kerberos Authentication (http://go.microsoft.com/fwlink/?LinkId=196655) and Kerberos Enhancements(http://go.microsoft.com/fwlink/?LinkId=196656). In addition, you should make yourself familiar with IIS 7.0 Kernel Mode authentication (Internet Information Services (IIS) 7.0 Kernel Mode Authentication Settings, (http://go.microsoft.com/fwlink/?LinkId=196657)) even though it is not supported in SharePoint Server farms.

Kerberos configuration changes in SharePoint 2010 Products

Most of the basic concepts of configuring Kerberos authentication in SharePoint 2010 Products have not changed. You still have to configure service principal names and you still have to configure delegation settings on computer and service accounts. However there are several changes that you should be aware of:

- Constrained Delegation — required for services which use the Claims to Windows Token Service. Constrained delegation is required to allow protocol transition to convert claims to Windows tokens.
- Service Applications — In Office SharePoint Server 2007, the SSP services required special SPNs and server registry changes to enable delegation. In SharePoint 2010 Products, service applications use claims authentication and the Claims to Windows Token service, so these changes are no longer needed.
- Windows Identity Foundation (WIF) — the WIF Claims to Windows Token Service (C2WTS) is a new service leveraged by SharePoint 2010 Products to convert claims to Windows tokens for delegation scenarios.

Considerations when you are upgrading from Office SharePoint Server 2007

If you are upgrading an Office SharePoint Server 2007 farm to SharePoint Server 2010, there are several things you should consider as you complete the upgrade:

- If web applications are changing URLs, make sure that you update the Service Principle Names to reflect the DNS names.
- Delete the SSP service principal names, because they are no longer needed in SharePoint Server 2010.
- Start the Claims to Windows Token Service on the servers that are running service applications that require delegation (for example, Excel Services, Visio Graphics Service).
- Configure Kerberos constrained delegation with "use any authentication protocol" to allow Kerberos constrained delegation with the C2WTS.
- Ensure Kernel mode authentication is disabled in IIS.

Configuring Kerberos authentication: Step-by-step configuration (SharePoint Server 2010)

Published: December 2, 2010

In the scenario articles that follow, we build out a SharePoint Server 2010 environment to demonstrate how to configure delegation in a number of common scenarios you might encounter in the enterprise. The walkthroughs assume you are building out a scaled-out SharePoint farm similar to what is described in the following section.

Note:

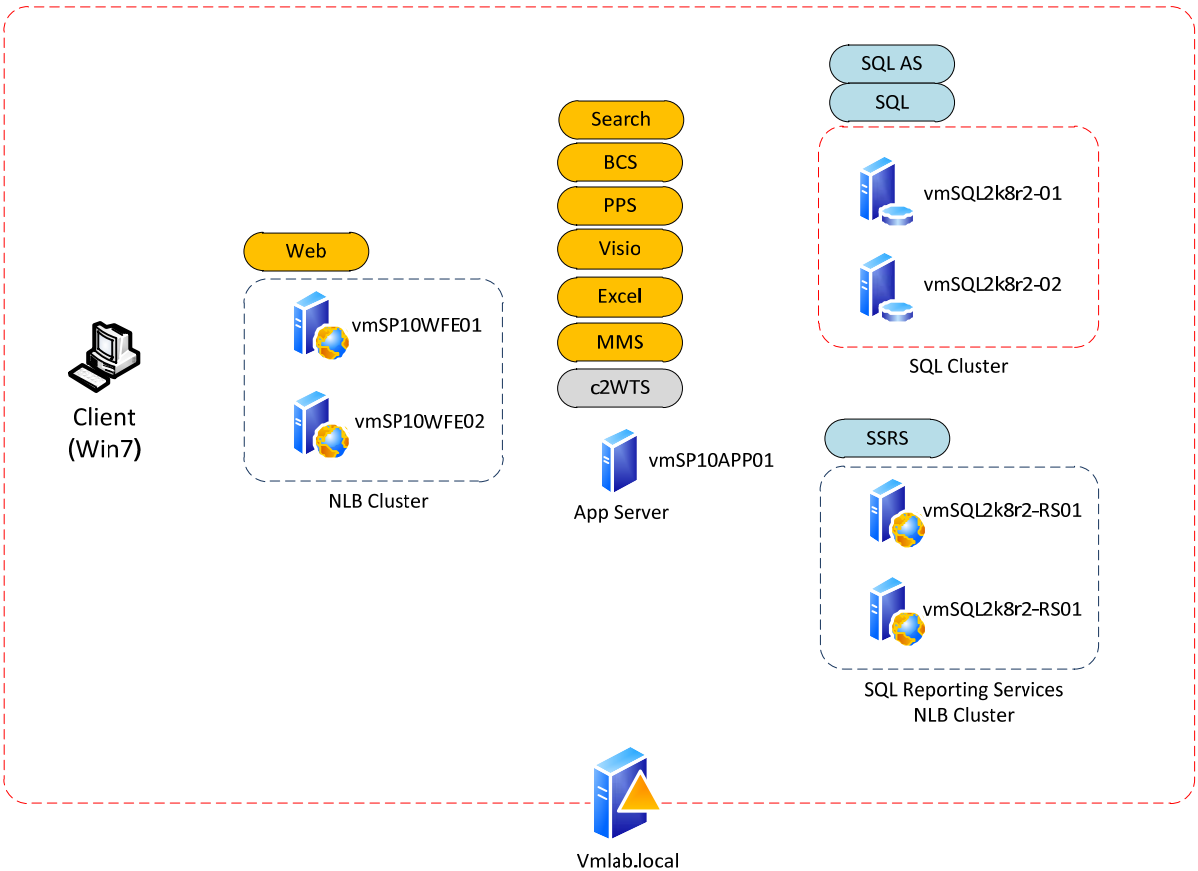
Some of the configuration steps may change, or may not work in certain farm topologies. For instance, a single server install does not support the Windows Identity Foundation C2WTS services so claims to windows token delegation scenarios are not possible with this farm configuration.

Environment and farm topology

The following diagram illustrates the farm topology used when configuring the scenarios in the sections below. The farm topology is load balanced and scaled out between multiple tiers to demonstrate how identity delegation would work in multi-server, multi hop scenarios.

Note:

The farm configuration in the demonstrations is not meant to be a reference architecture or an example of how to design a topology for production environments. For example, the demo topology runs all SharePoint Server 2010 service applications on a single server which creates a single point of failure for these services. For more information on how to design and build a production SharePoint Server environment, see SharePoint Server 2010 – Physical and Logical Architecture and Topologies for SharePoint Server 2010.



Note:

The scenario walkthroughs assume that all computers that are running SharePoint Server and the data sources used in the scenario below reside in a single domain. An explanation and walkthrough of multi-domain/multi-forest configuration is not covered in this document.

Environment specification

All computers in the demonstration environment are virtualized running on Windows Server 2008 R2 Hyper-V. The computers are joined to a single Windows domain, vmlab.local, running in Windows Server 2008 Forest and Domain function levels.

- Client Computer
    - Windows 7 Professional, 64 bit
- SharePoint Server front-end Webs
    - Windows Server 2008 R2 Enterprise, 64 bit
    - Services:
    - Web Application Service
    - Load balanced with Windows NLB
- SharePoint Server Application Server
    - Windows Server 2008 R2 Enterprise, 64 bit
    - Microsoft SharePoint Server 2010 (RTM)
    - Services:
    - WIF Claims to Windows Token Service
    - Managed Metadata Service
    - SharePoint Index
    - SharePoint Query
    - Excel Services
    - Visio Graphics Service

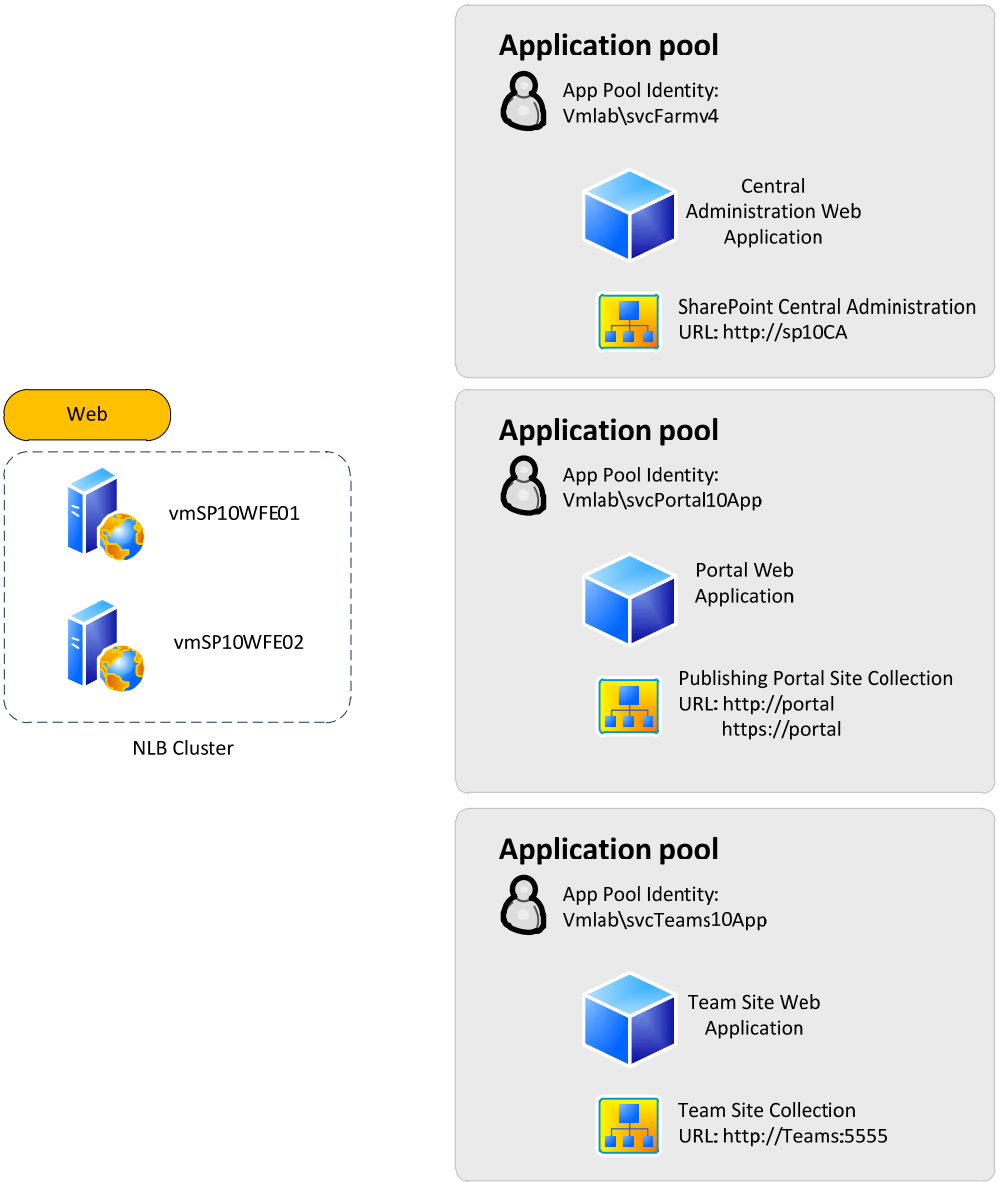
    - Business Connectivity Services
    - Performance Point Services
- SQL Services
    - Windows Sever 2008 R2 Enterprise, 64 bit
    - Microsoft SQL Server 2008 R2 Enterprise, 64 bit
    - Active/Passive Configuration
    - SQL Server Services:
    - SQL Data Engine
    - SQL Server Analysis Services
    - SQL Agent
    - SQL Browser
- SQL Reporting Server
    - Windows Server 2008 R2 Enterprise, 64 bit (RTM)
    - Microsoft SQL 2008 R2 Enterprise, 64 bit (RTM)
    - Microsoft SharePoint Server 2010 (RTM)
    - Windows NLB, Load balanced
    - Reporting Services SharePoint integration mode
    - Reporting Services, scaled-out mode

Web Application specification

The scenarios in the walkthrough reference a set of SharePoint Server 2010 web applications you will configure in Scenario 1. The following web applications are load balanced using Windows NLB across the two SharePoint Server web front ends in the demonstration environment:

- **http://sp10CA** The Central Administration web application for the farm. Scenario 1 will not walk through the configuration of this web application.

- **http://portal and https://portal** Web application with demonstration publishing portal. It is used to demonstrate how to configure delegation for web applications running on standard ports (HTTP 80, HTTPS 443)
- **http://teams:5555** Web application with demonstration team site. It is used to demonstrate how to configure delegation for web applications running on non-standard ports, in this example port 5555.



SSL configuration

Some of the walkthrough scenarios will use SSL to demonstrate how to configure delegation with HTTPS. It is assumed that the certificates being used come from a trusted root certificate authority, either internal or public, or you have configured all computers to trust the certificates being used. The document will not cover how to properly configure certificate trust nor will it provide guidance about debugging issues related to SSL certificate installation. It is highly recommended to review these topics and test your SSL configuration before configuring Kerberos constrained delegation with SSL protected services. For more information see:

- Active Directory Certificate Services Overview (http://go.microsoft.com/fwlink/?LinkId=196660)
- Active Directory Certificate Services Step-by-Step Guide (http://go.microsoft.com/fwlink/?LinkId=196661)
- Configuring Server Certificates in IIS 7 (http://go.microsoft.com/fwlink/?LinkId=196662)
- How to Set Up SSL on IIS 7: Configuring Security : Installing and Configuring IIS 7 : The Official Microsoft IIS Site (http://go.microsoft.com/fwlink/?LinkID=193447)
- Add a Binding to a Site (IIS 7) (http://go.microsoft.com/fwlink/?LinkId=196663)
- Configure a Host Header for a Web Site (IIS 7) (http://go.microsoft.com/fwlink/?LinkId=196664) — (How to use SSL with host headers)
- Create a Self-Signed Server Certificate in IIS 7 (http://go.microsoft.com/fwlink/?LinkId=196665)

Load balancing

Load balancing on the SharePoint Server front-end Webs and SQL Server Reporting Services servers was implemented by using Windows Server 2008 Network Load Balancing (NLB). How to configure NLB and NLB best practices are not covered in this document. For more information on NLB, refer to Overview of Network Load Balancing.

SQL aliasing

The farm was built using a SQL client alias to connect to the SQL cluster. This is typically a best practice and was done to demonstrate how Kerberos authentication is configured

when SQL aliasing is used. Scenario 2 assumes the environment is configured in this manner, but it is not required to use SQL aliases to complete any of the scenarios below. For more information on how to configure SQL aliases see How to: Create a Server Alias for Use by a Client (SQL Server Configuration Manager).

SharePoint Server Services and service accounts

The scenarios below implement a least-privilege model where each service in the SharePoint farm leverages a separate, distinctActive Directory account for its service identity. Using a least-privilege model has advantages and disadvantages:

Advantages:

- **The administrator can control the permissions of each service in a fine-grained way** This includes domain permissions, local permissions and privileges, delegation rights and other settings.
- **Better auditing and traceability** By ensuring each service leverages its own identity, an administrator can track network and system activity back to the specific service based on the identity captured in audit files. For example, if a server audit log shows logon activity for a particular account, the account could be used to trace the activity to a particular service.
- **Better security** By leveraging separate accounts for each service, an administrator assures that if one account is compromised it potentially limits the damage due to the security issue because only the service that is using the compromised account is affected. Note that if any account becomes compromised, a holistic security assessment of the entire environment should be performed to determine the most appropriate action to address the security issue.

Disadvantages:

- **Increased account management complexity** Having more service accounts translates to more Active Directory configuration and more password management policies to enforce.

- **Additional configuration** As seen in the step by step guide below, once a SharePoint Server administrator makes the decision to leverage a least-privilege model, there are additional steps she or he must perform to configure the environment correctly.

- **Increased administration complexity** The probability of misconfiguration increases as the complexity of the environment increases. When you leverage multiple accounts, there is a chance that certain services willbe misconfigured, which can lead to functionality issues and triage needed to correct the issues.

Be aware that using separate service accounts is not a requirement of SharePoint Server but a general recommendation for production environments. The steps in the rest of this paper outline how to configure SharePoint Server when you are using separate accounts; some of these steps may not apply when you are using shared accounts.

C2WTS Service Identity

The steps below assume a least-privilege security model and leverage discrete service accounts for each SharePoint Server service. The C2WTS is configured to use a separate Active Directory account instead of the default local system account to follow this design tenant. When you use a distinct account, the individual delegation rights granted to the C2WTS can be managed separately from other services on the server that are also using the local system account. Note that this is not a product requirement, but a recommended practice.

Tips for working through the scenarios

The scenarios below walk through various activities needed to configure Kerberos delegation across different functions of the SharePoint Server platform. As you go through each section:

All the scenarios assume you have your web applications configured for incoming classic authentication (Kerberos). Some scenarios below require classic authentication and will not function as documented with incoming claims authentication.

- Get the SharePoint Server services working first without delegation to ensure the service applications are configured correctly before moving on to more challenging configurations with delegation.

- Try to pay special attention to each step and avoid skipping any steps
- Work through scenario 1 and spend time using the debugging tools mentioned in the scenario as they can be used in other scenarios to triage configuration issues.
- Remember to work through scenario 2. You'll need a computer running SQL Server that is configured to accept Kerberos authentication and will require the test database that you setup in this scenario for some of the later scenarios.
- Always double-check SPN configuration in each scenario by using **SetSPN -X**and **SetSPN -Q**. See the appendix for more information.
- Always be sure to check the server event logs and ULS logs when attempting to debug a configuration issue. There are typically good pointers in these logs which can quickly point out the issues you are encountering.
- Turn up diagnostic logging for SharePoint Foundation->Claims Authentication and any service applications that you are attempting to triage if issues occur.
- Remember that each scenario may be affected by service application caching. If you make configuration changes but do not see changes in platform behavior, try restarting the service's application pool or windows service. If this has no effect, sometimes a system reboot will help.
- Remember that Kerberos tickets are cached once requested. If you are using a tool like NetMon to view TGT and TGS requests, you may need to empty the ticket cache if you don't see the request traffic you expect. Scenario 1, Configuring Kerberos authentication: Core configuration (SharePoint Server 2010) explains how to do this with the KLIST and KerbTray utilities.
- Remember to run NetMon with Administrative privileges to capture Kerberos traffic.
- For advanced debugging scenarios you may want to turn on WIF tracing for the Claims to Windows Token Service and WCF tracing for the SharePoint Service Applications (WCF services). See:
  - WIF Tracing
  - How to: Enable Tracing
  - Configuring Tracing

Configuring Kerberos authentication: Core configuration (SharePoint Server 2010)

Published: December 2, 2010

In the first scenario you will configure two SharePoint Server 2010 web applications to use the Kerberos protocol for authenticating incoming client requests. For demonstration purposes, one web application will be configured to use standard ports (80/443) and the other will use a non-default port (5555). This scenario will be the basis of all the following scenarios which assume the activities below have been completed.

Important:

It is a requirement to configure your web applications with classic Windows authentication using Kerberos authentication to ensure that the scenarios work as expected. Windows-Claims authentication can be used in some scenarios but may not produce the results detailed in the scenarios below.

Note:

If you are installing on Windows Server 2008, you may need to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

In this scenario you do the following things:

- Configure two web applications with default zones that use the Kerberos protocol for authentication
- Create two test site collections, one in each web application
- Verify the IIS configuration of the web application

- Verify that clients can authenticate with the web application and ensure that the Kerberos protocol is used for authentication
- Configure the RSS Viewer web part to display RSS feeds in a local and remote web application
- Crawl each web application and test searching content in each test site collection

Configuration checklist

Area of Configuration	Description
DNS	Register a DNS A Record for the web applications networked loaded balanced (NLB) virtual IP (VIP)
Active Directory	Create a service accounts for the web applications' IIS application pool Register Service Principal Names (SPN) for the web applications on the service account created for the web application's IIS application pool Configure Kerberos constrained delegation for service accounts
SharePoint Web App	Create SharePoint Server managed accounts Create the SharePoint Search Service Application Create the SharePoint web applications
IIS	Validate that Kerberos authentication is Enabled Verify Kernel-mode authentication is disabled Install certificates for SSL
Windows 7 Client	Ensure web application URLs are in the intranet zone, or a zone configured to automatically authenticate with integrated Windows authentication
Firewall	Open firewall ports to allow HTTP traffic in on default and

Area of Configuration	Description
Configuration	non-default ports Ensure clients can connect to Kerberos Ports on the Active Directory
Test Browser Authentication	Verify authentication works correctly in the browser Verify Logon information on the web server's security event log Use third party tools to confirm Kerberos authentication is configured correctly
Test SharePoint Server Search Index and Query	Verify browser access from the index server(s) Upload sample content and perform a crawl Test search
Test WFE Delegation	Configure RSS Feed sources on each site collection Add RSS view web parts to the home page of each site collection

Step-by-step configuration instructions

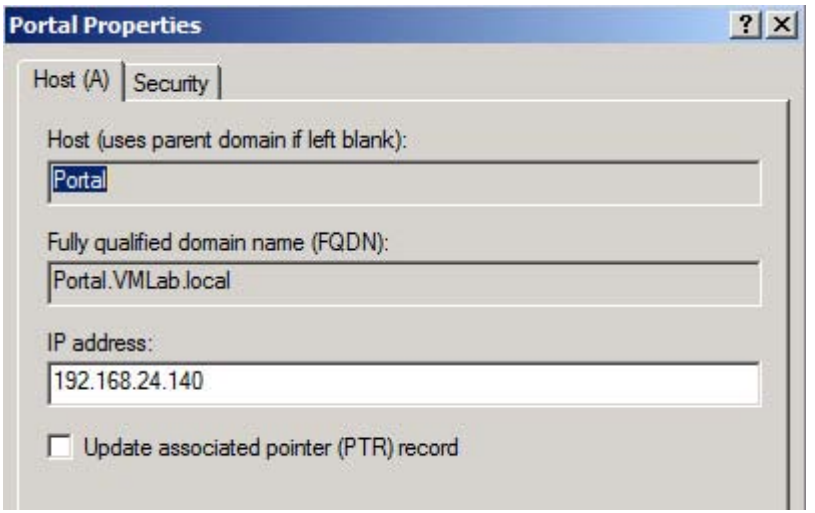
Configure DNS

Configure DNS for the web applications in your environment. In this example we have 2 web applications, http://portal and http://teams:5555, which both resolve to the same NLB VIP (192.168.24.140/24)

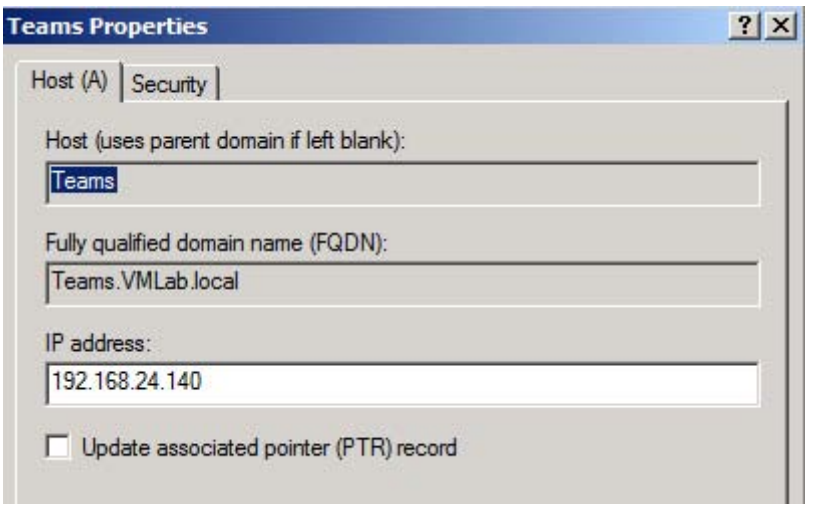
For general information about how to configure DNS, seeManaging DNS Records.

SharePoint Server Web apps

http://portal — Configure a new DNS A Record for the portal web application. In this example we have a host "portal" configured to resolve to the load balanced VIP.



http://teams:5555 — Configure a new DNS A Record for the for the team's web application



Note:

It is important to ensure the DNS entries are A Records and not CNAME aliases for Kerberos authentication to work successfully in environments with more than one web application running with host headers and separate dedicated service accounts. See Kerberos configuration known issues (SharePoint Server 2010) for an explanation of the known issue with using CNAME with Kerberos enabled web applications.

Configure Active Directory

Next you will configure the Active Directory accounts for the web applications in your environment. As a best practice you should configure each web application to run in its own IIS application pool with its own security context (application pool identity).

SharePoint Service Application Service Accounts

In our example we have two SharePoint Server web applications running in two separate IIS application pools running with their own application pool identities.

Web Application (default zone)	IIS App Pool Identity
http://portal	vmlab\svcPortal10App
http://teams:5555	vmlab\ svcTeams10App

Service Principal Names (SPNs)

For each service account, configure a set of service principal names that map to the DNS host names assigned to each web application.

Use SetSPN, a command line tool in Windows Server 2008, to configure a new service principal name. For a full explanation of how to use SetSPN, see Setspn. To learn about SetSPN improvements in Windows Server 2008, see Care, Share and Grow! : New features in SETSPN.EXE on Windows Server 2008.

All SharePoint Server web applications, regardless of port number, use the following SPN format:

- HTTP/<DNS HOST name>
- HTTP/<DNS FQDN>

Example:

- HTTP/portal
- HTTP/portal.vmlab.local

For Web applications running on non-default ports (ports other than 80/443) register additional SPNs with port number:

- HTTP/<DNS Host Name>:<port>
- HTTP/<DNS FQDN>:<port>

Example:

- HTTP/teams:5555
- HTTP/teams.vmlab.local:5555

Note:

See the appendix for an explanation of why it is recommended to configure SPNs with and without port number for HTTP services running on non-default ports (80, 443). Technically the correct way to refer to a HTTP service that runs on a non-default port is to include the port number in the SPN but because of known issues described in the appendix we need to configure SPNs without port as well. Note that the SPNs without port for the **teams** web application does not mean services will be accessed using the default ports (80, 443) in our example.

In our example we configured the following service principal names for the two accounts we created in the previous step:

DNS Host	IIS App Pool Identity	Service Principal Names
Portal.vmlab.local	vmlab\svcPortal10App	HTTP/portal HTTP/portal.vmlab.local
Teams.vmlab.local	vmlab\ svcTeams10App	HTTP/Teams HTTP/Teams.vmlab.local HTTP/Teams:5555 HTTP/Teams.vmlab.local:5555

To create the service principal names the following commands were executed:

```
SetSPN -S HTTP/Portal vmlab\svcportal10App

SetSPN -S HTTP/Portal.vmlab.localvmlab\svcportal10App

SetSPN -S HTTP/Teams vmlab\svcTeams10App

SetSPN -S HTTP/Teams.vmlab.localvmlab\ svcTeams10App

SetSPN -S HTTP/Teams:5555vmlab\ svcTeams10App

SetSPN -S HTTP/Teams.vmlab.local:5555 vmlab\ svcTeams10App
```

Important:

Do not configure service principal names with HTTPS even if the web application uses SSL.

In our example we used a new command line switch, -S, introduced in Windows Server 2008 that checks for the existence of the SPN before creating the SPN on the account. If the SPN already exists, the new SPN is not created and you see an error message.

```
C:\>SetSPN -S http/Teams vmlab\svcTeamsApp
Checking domain DC=VMLab,DC=local
CN=svcTeams10App,OU=014,OU=Service Accounts,DC=VMLab,DC=local
        HTTP/Teams.vmlab.local
        HTTP/Teams

Duplicate SPN found, aborting operation!
```

If duplicate SPNs are found, you have to resolve the issue by either using a different DNS name for the web application, thereby changing the SPN, or by removing the existing SPN from the account it was discovered on.

Important:

Before deleting an existing SPN, be sure it is no longer needed, otherwise you may break Kerberos authentication for another application in your environment.

Service Principal Names and SSL

It is common to confuse Kerberos Service Principal Names with URLs for http web applications because the SPN and URI formats are very similar in syntax, but it's important to understand that they are two very separate and unique things. Kerberos service principal names are used to identify a service, and when that service is an http application, the service scheme is "HTTP" regardless if the service is access with SSL or not. This means that even if you access the web application using "https://someapp"you do not, and should not, configure a service principal name with HTTPS, for example "HTTPS/someapp".

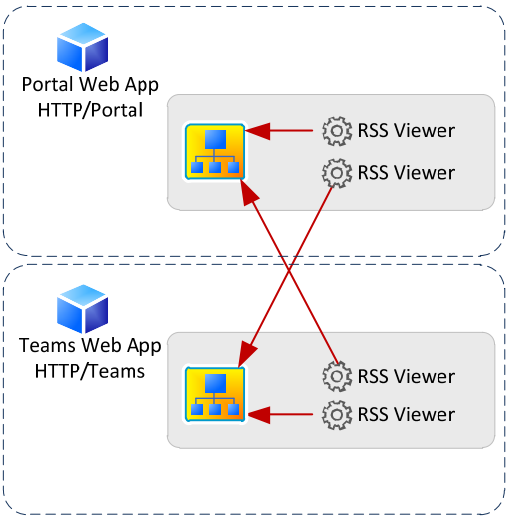
Configure Kerberos constrained delegation for computers and service accounts

Depending on the scenario, some functionality in SharePoint Server 2010 may require constrained delegation to function properly. For example, if the RSS viewer web part is configured to display a RSS feed from an authenticated source it will require delegation to consume the source feed. In other situations it may be required to configure constrained delegation to allow service applications (such as Excel Services) to delegate the client's identity to back-end systems.

In this scenario we will configure Kerberos constrained delegation to allow the RSS view web part to read a secured local RSS feed and secured remote RSS feed from a remote

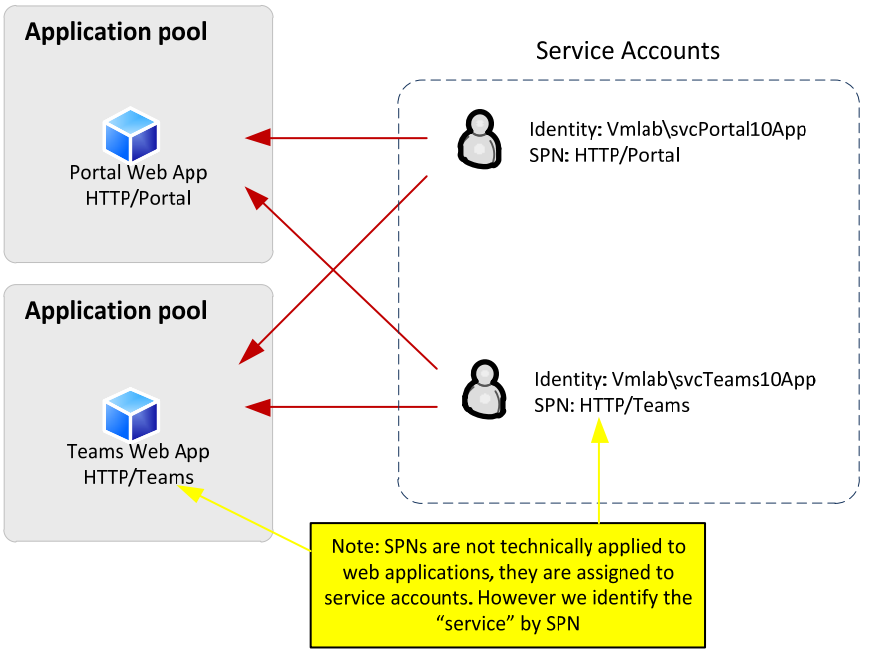
web application. In later scenarios we will configure Kerberos constrained delegation for other SharePoint Server 2010 service applications.

The following diagram conceptually describes what will be configured in this scenario:



We have two web applications, each with their own site collection with a site page hosing two RSS viewer web parts. The web applications each have a single default zone configured to use Kerberos authentication so all feeds coming from these web sites will require authentication. In each site one RSS viewer will be configured to read a local RSS feed from a list and the other will be configured to read an authentication feed in the remote site.

To accomplish this, Kerberos constrained delegation will be configured to allow delegation between the IIS application pool service accounts. The following diagram conceptually describes the constrained delegation paths needed:



Remember that we identify the web application by service name using the Service Principal Name (SPN) assigned to the identity of the IIS application pool. The service accounts processing requests must be allowed to delegate the client identity to the designated services. All together we have the following constrained delegation paths to configure:

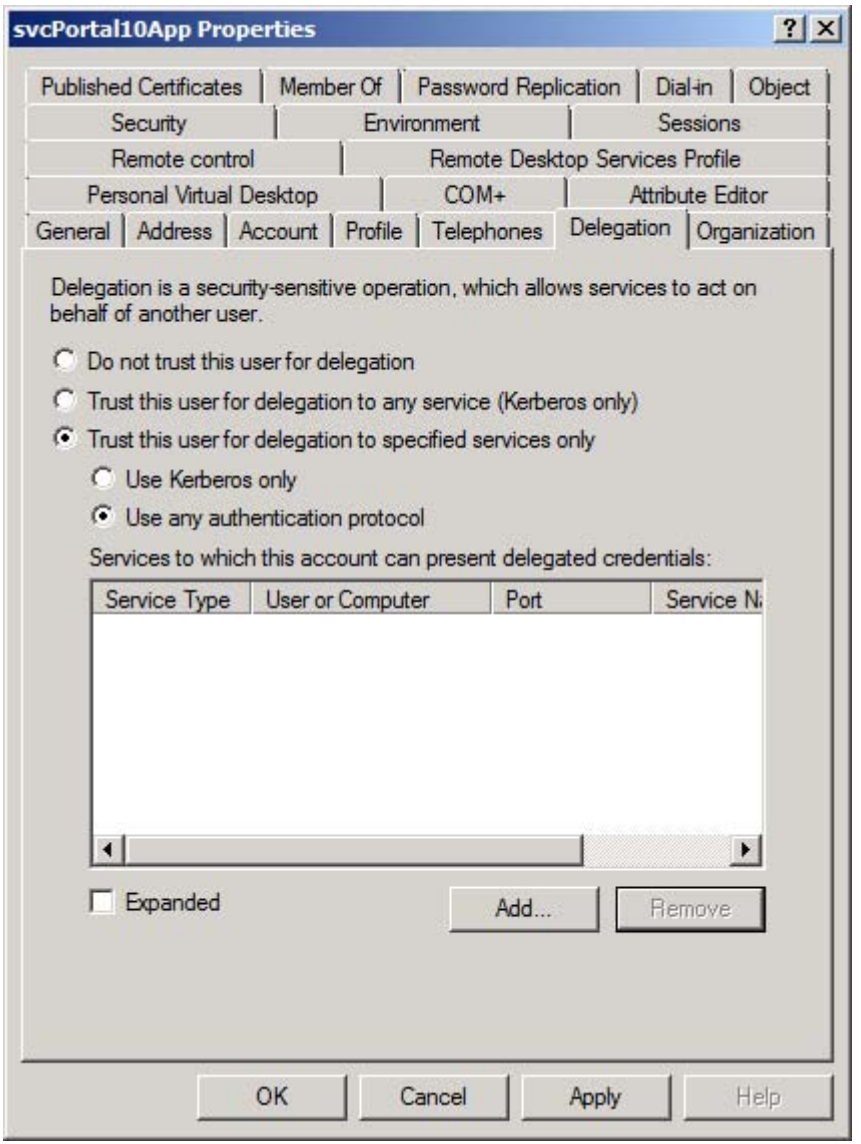
Principal Type	Principal Name	Delegates To Service
User	svcPortal10App	HTTP/Portal HTTP/Portal.vmlab.local HTTP/Teams HTTP/Teams.vmlab.local HTTP/Teams:5555 HTTP/Teams.vmlab.local:5555

Principal Type	Principal Name	Delegates To Service
User	svcTeams10App	HTTP/Portal HTTP/Portal.vmlab.local HTTP/Teams HTTP/Teams.vmlab.local HTTP/Teams:5555 HTTP/Teams.vmlab.local:5555

Note:

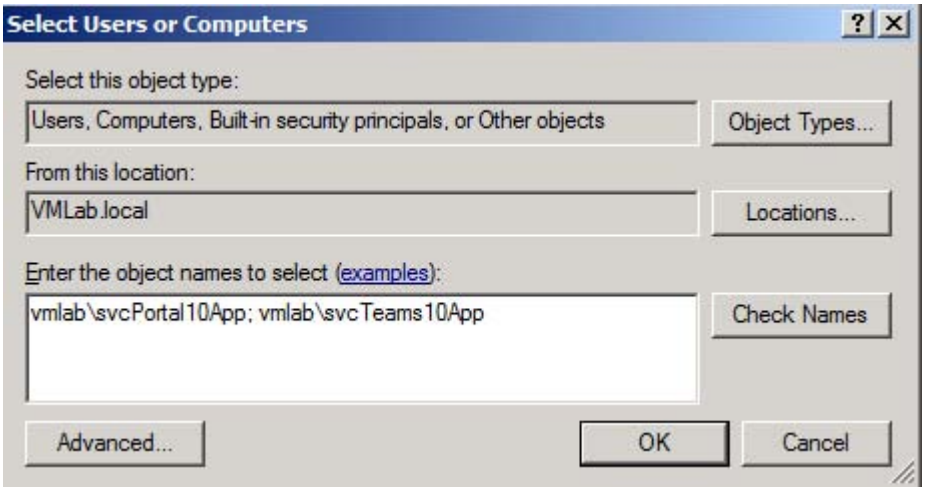
It may seem redundant to configure delegation from a service to itself, such as the portal service account delegating to the portal service application, but this is required in scenarios where you have multiple servers running the service. This is to address the scenario where one server may need to delegate to another server running the same service; for instance a WFE processing a request with a RSS viewer which uses the local web application as the data source. Depending on farm topology and configuration there is a possibility that the RSS request may be serviced by a different server which would require delegation to work correctly.

To configure delegation you can use the Active Directory Users and Computer snap-in. Right-click each service account and open the properties dialog. In the dialog you will see a tab for delegation (note that this tab only appears if the object has an SPN assigned to it; computers have an SPN by default). On the delegation tab, select **Trust this user for delegation to specified services only**, then select **Use any authentication protocol**.

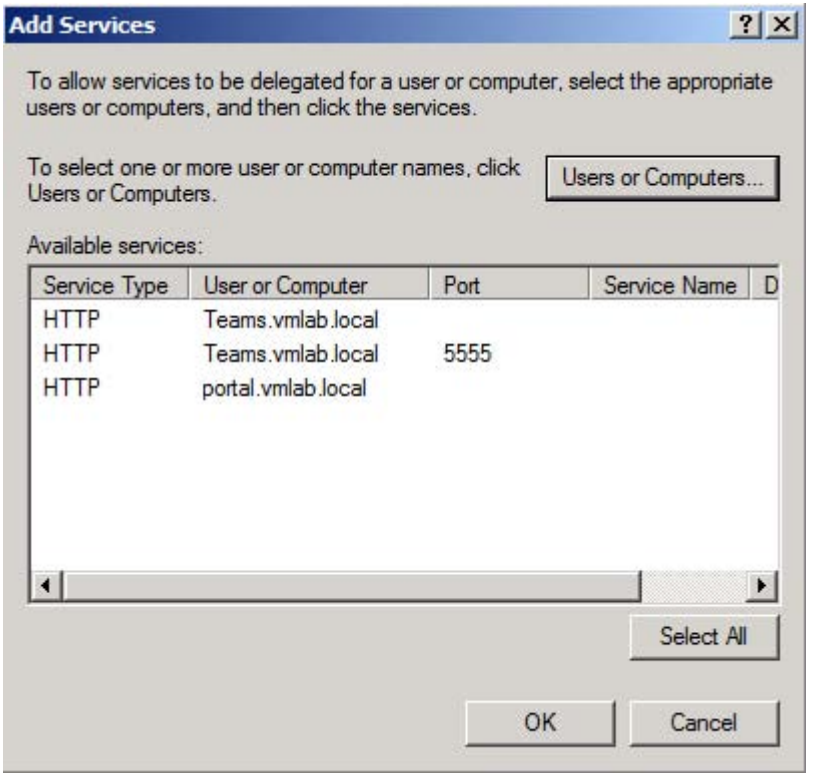


Click the **Add** button to add the services the user (service account) will be allowed to delegate to. To select a SPN, you will look up the object the SPN is applied to. In our instance, we are trying to delegate to a HTTP service which means we search for the service account of the IIS application pool that the SPN was assigned to in the previous step.

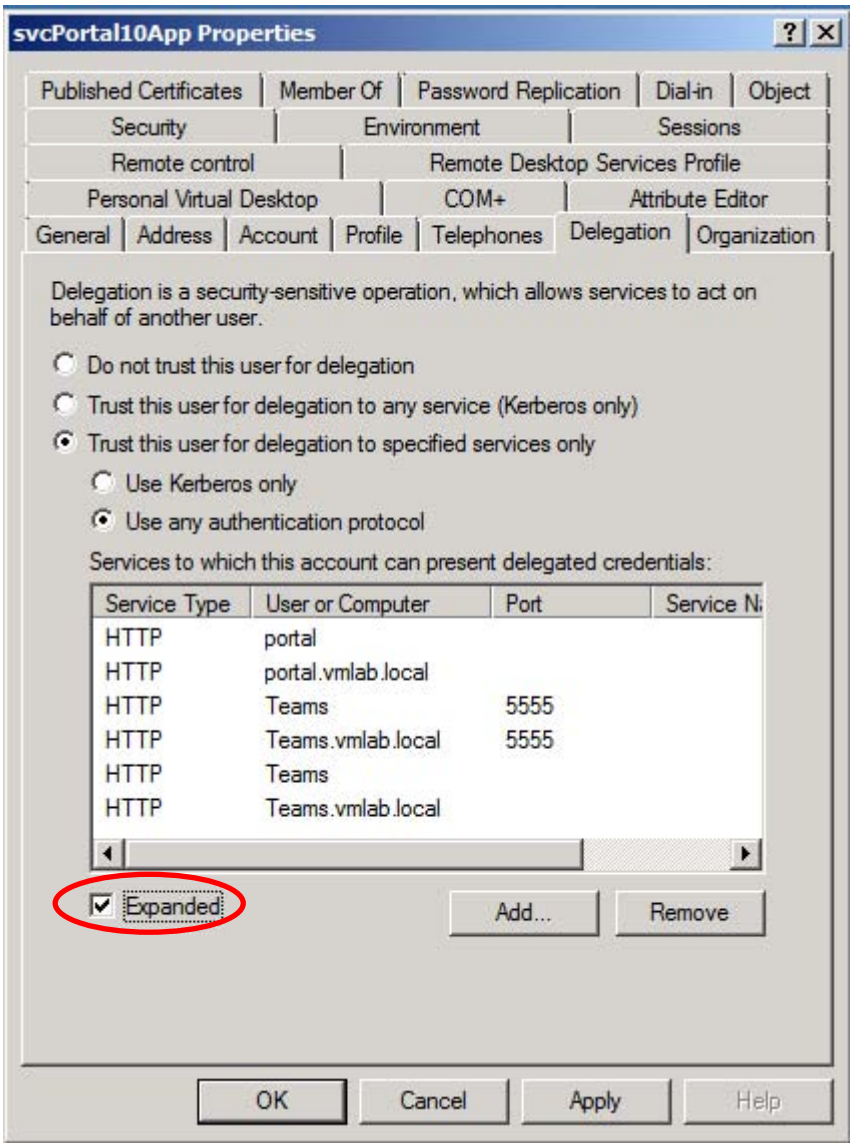
On the **Select Users or Computers** dialog box, click **Users and Computers**, search for the IIS application pool service accounts (in our example **vmlab\svcPortal10App** and **vmlab\svcTeams10App**"and then click **OK**:



You will then be prompted to select the services assigned to the objects by service principal name.



On the **Add Services** dialog box, click **Select All** then click **OK**. Note that when you return to the delegation dialog you do not actually see all the SPNs selected. To see all SPNs, check the **Expanded** check box in the lower left hand corner.



Perform these steps for each service account in your environment that requires delegation. In our example this is the service accounts list

Configure SharePoint Server

Once Active Directory and DNS are configured, it's time to create the web application in your SharePoint Server 2010 Farm. This paper assumes that the installation of

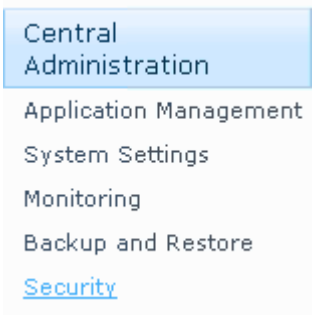
SharePoint Server is complete at this point and the farm topology and supporting infrastructure, for instance load balancing, is configured. For more information about how to install and configure your SharePoint farm, see: Deployment for SharePoint Server 2010.

Configure managed service accounts

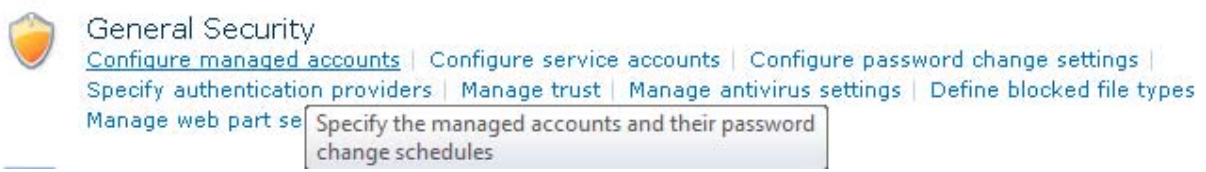
Before creating your web applications, configure the services accounts created in the previous steps as managed service accounts in SharePoint Server. Doing so ahead of time will allow you to skip this step when creating the web applications themselves.

To configure a managed account

1. In SharePoint Central Administration, click **Security**.



2. Under **General Security** click **Configure managed accounts**:



3. Click **Register Managed Account** and create a managed account for each service account. In this example we created five managed service accounts:

Account	Purpose
VMLAB\svcSP10Search	SharePoint Search Service Account
VMLAB\svcSearchAdmin	SharePoint Search Administration Service Account

Account	Purpose
VMLAB\svcSearchQuery	SharePoint Search Query Service Account
VMLAB\svcPortal10App	Portal Web App IIS Application Pool Account
VMLAB\svcTeams10App	Teams Web App IIS Application Pool Account

Note:

Managed accounts in SharePoint Server 2010 are not the same as managed service accounts in Windows Server 2008 R2 Active Directory.

Create the SharePoint Server Search Service Application

In this example we will configure the SharePoint Server Search Service Application to ensure the newly create web application can be crawled and searched upon successfully. Create a new SharePoint Server Search Web Application and place the Search, Query and Administration Services on the application server, in our example vmSP10App01. For a detailed explanation on how to configure the Search Service Application, see Step-by-Step: Provisioning the Search Service Application.

Note:

The placement of all Search Services on a single application server is for demonstration purposes only. A complete discussion about SharePoint Server 2010 Search Topology options and best practices is out of scope for this document.

Create the Web Application

Browse to Central Administration and navigate to **Manage Web Applications** in the **Application Management** section. In the toolbar, select **New** and create your web application. Ensure that the following is configured:

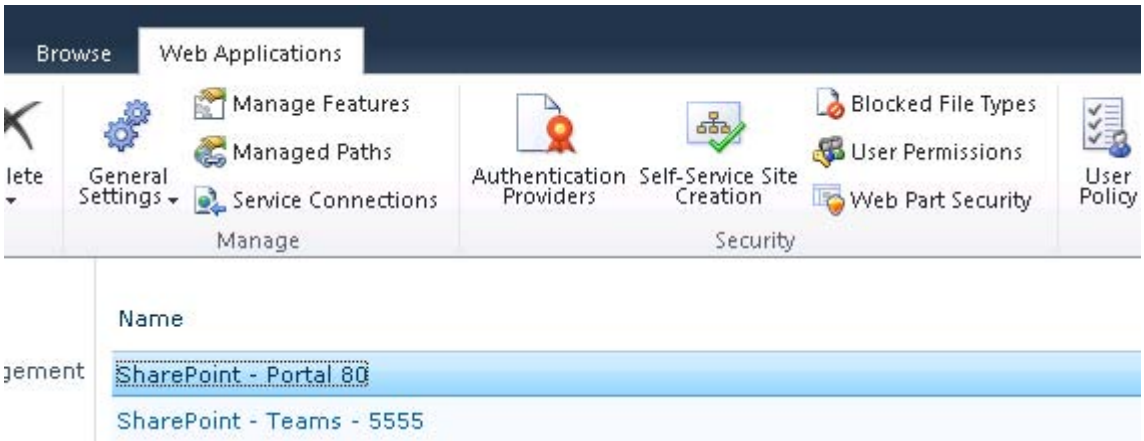
- Select **Classic Mode Authentication**.
- Configure the port and host header for each web application.

- Select **Negotiate** as the Authentication Provider.
- Under application pool, select **Create new application pool** and select the managed account created in the previous step.

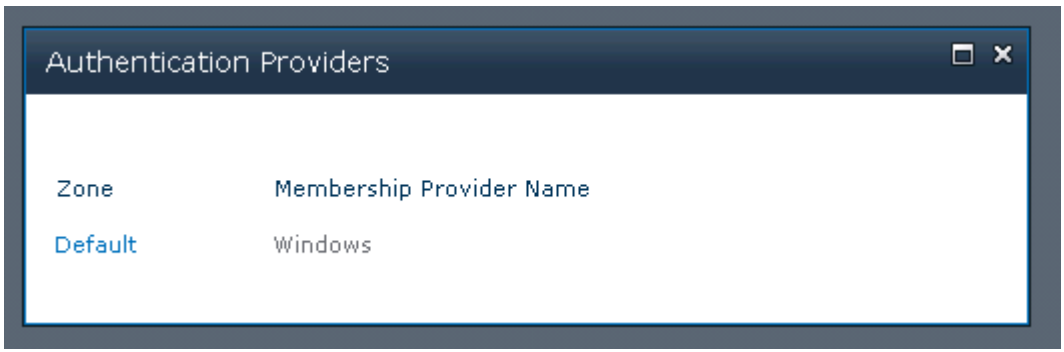
In this example, two web applications were created with the following settings:

Setting	http://Portal Web Application	http://Teams Web Application
Authentication	Classic Mode	Classic Mode
IIS Web Site	Name: SharePoint – Portal – 80 Port: 80 Host Header: Portal	Name: SharePoint – Teams – 5555 Port: 80 Host Header: Teams
Security Configuration	Auth Provider: Negotiate Allow Anonymous: No Use Secure Socket Layer: No	Auth Provider: Negotiate Allow Anonymous: No Use Secure Socket Layer: No
Public URL	http://Portal:80	http://Teams:5555
Application Pool	Name: SharePoint – Portal80 Security Account: vmlab\svcPortal10App	Name: SharePoint – Teams5555 Security Account: vmlab\svcTeams10App

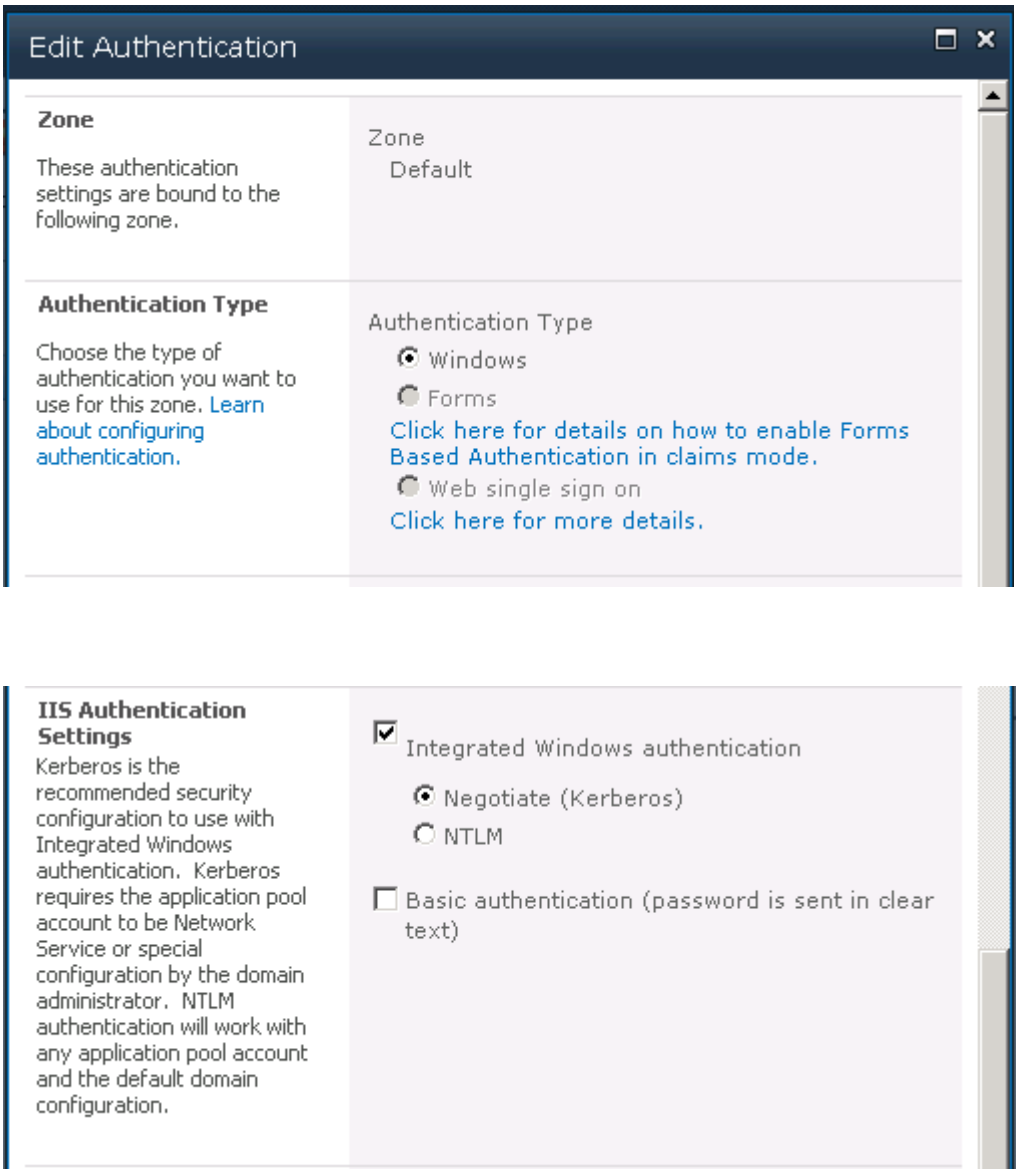
When creating the new web application you are also create a new zone, the default zone, configured to use the Windows authentication provider. You can see the provider and it's settings for the zone in web application management by first selecting the web application, then clicking **Authentication Providers** in the toolbar. The authentication providers dialog box lists all the zones for the selected web application along with the authentication provider for each zone. By selecting the zone, you will see the authentication options for that zone.



The authentication providers dialog will list all the zones for the selected web application along with the authentication provider for each zone:



By selecting the zone, you will see the authentication options for that zone:



If you misconfigured the Windows settings and selected NTLM when the web application was created, you can use the edit authentication dialog for the zone to switch the zone from NTLM to Negotiate. If **classic mode** was not selected as the authentication mode, you must either create a new zone by extending the web application to a new IIS web site or delete and recreate the web application.

Create site collections

To test whether authentication is working correctly, you will need to create at least one site collection in each web application. The creation and configuration of the site collection will not affect Kerberos functionality, so follow existing guidance on how to create a site collection in Create a site collection (SharePoint Foundation 2010).

For this example, two site collections were configured:

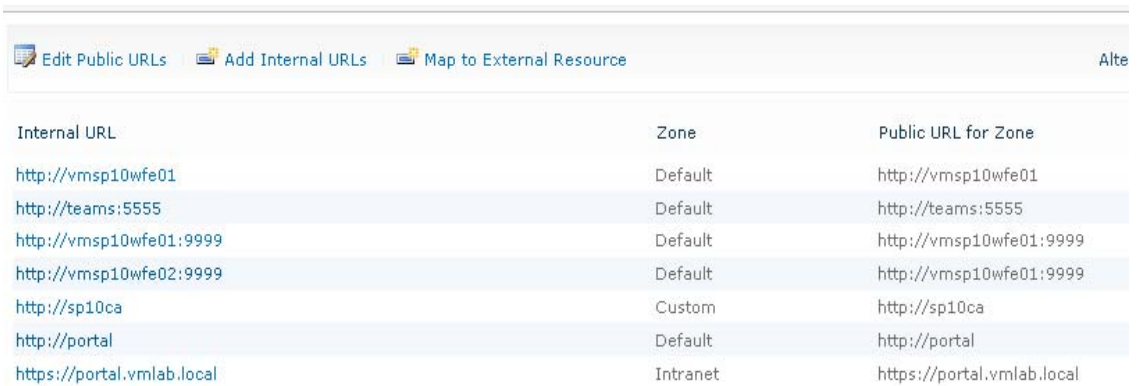
Web Application	Site Collection Path	Site Collection Template
http://portal	/	Publishing Portal
http://teams:5555	/	Team Site

Create alternate access mappings

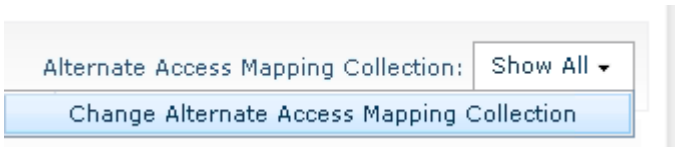
The portal web application will be configured to use HTTPS as well as HTTP to demonstrate how delegation works with SSL protected services. To configure SSL, the portal web application will need a second SharePoint Server alternate access mapping (AAM) for the HTTPS endpoint.

To configure alternate access mappings

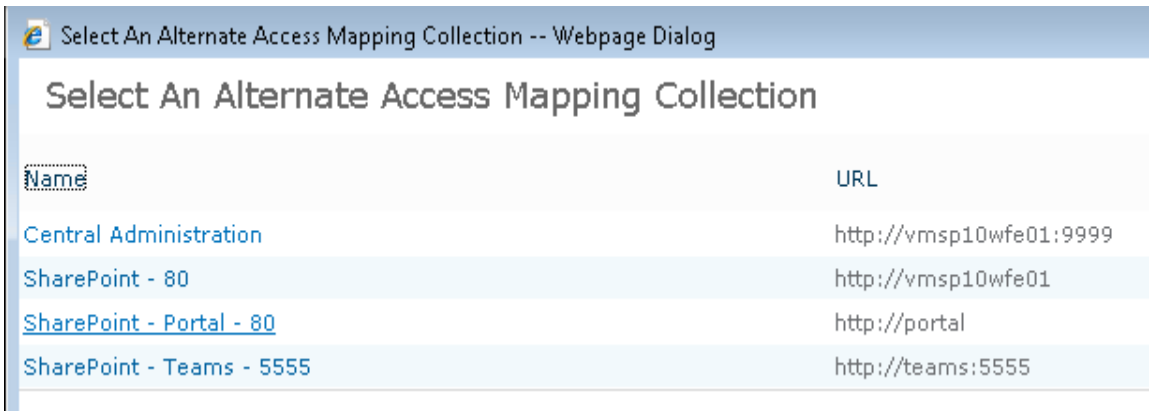
1. In Central Administration, click **Application Management**.
2. Under **Web Applications** click **configure alternate access mappings**.



3. In the **Select Alternate Access Mapping Collection** dropdown, select the **Change Alternate Access Mapping Collection**.

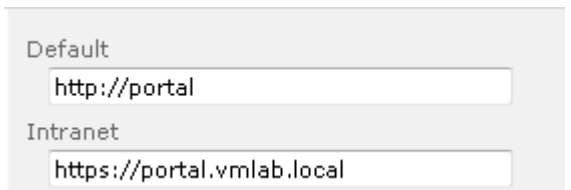


4. Select the portal web application.



5. Click **Edit Public Urls** in the top toolbar.

6. In a free zone, add the https URL for the web application. This URL will be the name on the SSL certificate you will create in the next steps.



7. Click **Save**.

   You should now see the HTTPS URL in the zone list for the web application.

Internal URL	Zone	Public URL for Zone
http://portal	Default	http://portal
https://portal.vmlab.local	Intranet	https://portal.vmlab.local

IIS configuration

Install SSL certificates

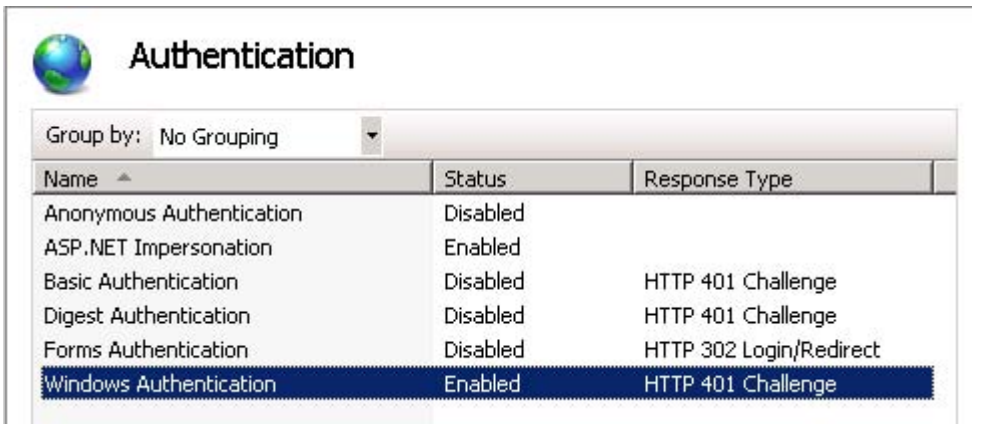
You will need to configure an SSL certificate on each SharePoint Server hosting the web application service for each web application that uses SSL. Again, the topic of how to configure an SSL certificate and certificate trust is out of scope for this document. See the SSL Configuration section in this document for references to material about configuring SSL certificates in IIS.

Verify that Kerberos authentication is enabled

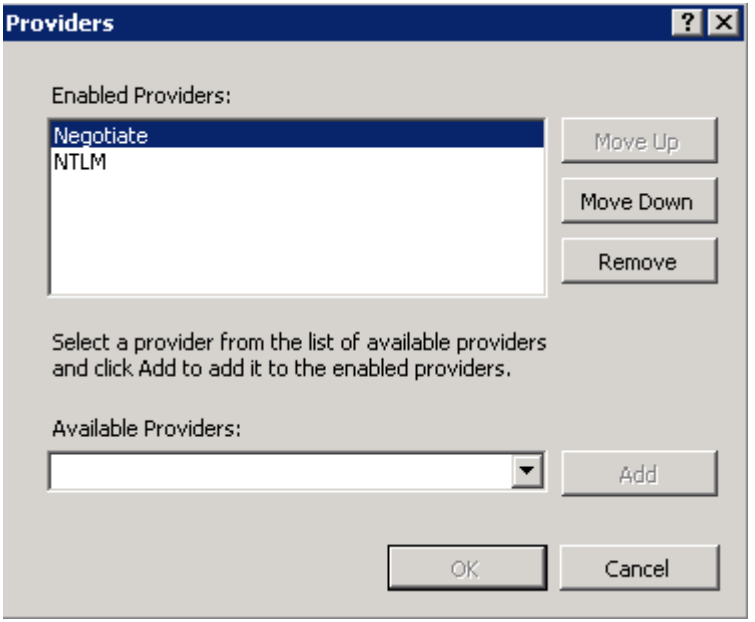
To verify that Kerberos authentication is enabled on the web site

1. Open IIS manager.
2. Select the IIS web site to verify.
3. In Features View, under IIS, double click **Authentication**.

4. Select **Windows Authentication** which should be enabled.



5. On the right hand side under **Actions**, select **Providers**. Verify **Negotiate** is at the top of the list.

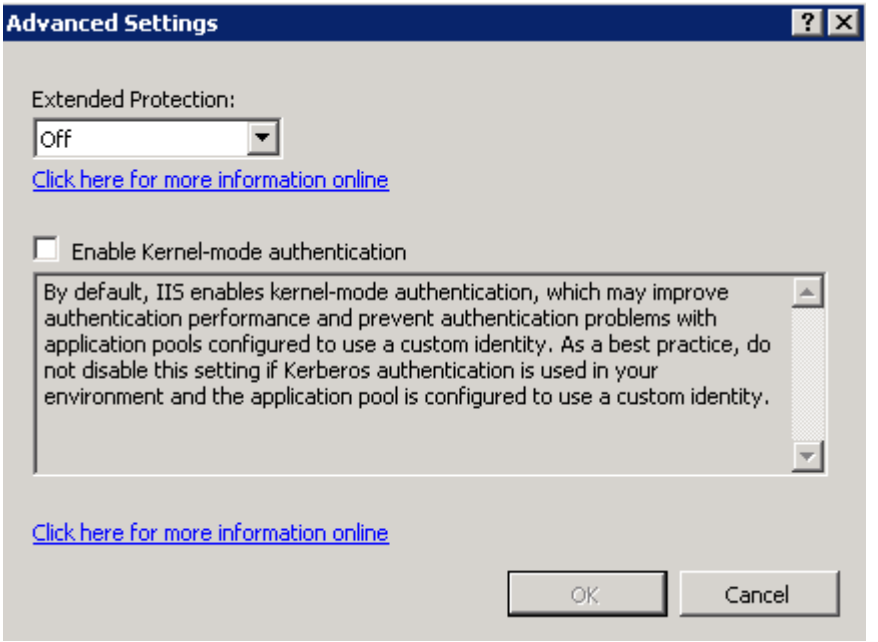


Verify that Kernel mode authentication is disabled

Kernel mode authentication is not supported in SharePoint Server 2010. By default, all SharePoint Server Web Applications should have Kernel Mode Authentication disabled by default on their corresponding IIS web sites. Even in situations where the web application was configured on an existing IIS web site, SharePoint Server disables kernel mode authentication as it provisions a new web application on the existing IIS site.

To verify that kernel mode authentication is disabled

1. Open IIS manager.
2. Select the IIS web site to verify.
3. In Features View, under IIS, double click **Authentication**.
4. Select **Windows Authentication**, which should be enabled.
5. Click **Advanced Settings**.
6. Verify both EAP and Kernel Mode Authentication are disabled.

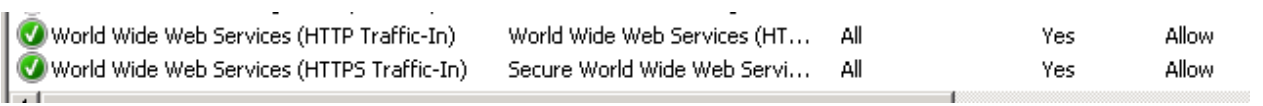


Configure the firewall

Before testing authentication, ensure clients can access the SharePoint Server web applications on the configured HTTP ports. In addition, ensure clients can authenticate with Active Directory and request Kerberos tickets from the KDC over the standard Kerberos ports.

Open firewall ports to allow HTTP traffic in on default and non-default ports

Typically you have to configure the firewall on each front-end Web to allow incoming requests over ports TCP 80 and TCP 443. Open Windows Firewall with Advanced Security and browse to the following Inbound Rules:



- World Wide Web Services (HTTP Traffic-In)
- World Wide Web Services (HTTPS Traffic-In)

Make sure the appropriate ports are open in your environment. In our example, we access SharePoint Server over HTTP (port 80), so this rule was enabled.

In addition, we have to open the non-default port used in our example (TCP 5555). If you have web sites running on non-default ports, you also have to configure custom rules to allow HTTP traffic on those ports.

Ensure that clients can connect to Kerberos ports on the Active Directory role

To use Kerberos authentication, clients will have to request ticket granting tickets (TGT) and service tickets (ST) from the Key Distribution Center (KDC) over UDP or TCP port 88. By default, when you install the Active Directory Role in Windows Server 2008 and later, the role will configure the following incoming rules to allow this communication by default:

- Kerberos Key Distribution Center – PCR (TCP-In)
- Kerberos Key Distribution Center – PCR (UDP-In)
- Kerberos Key Distribution Center (TCP-In)
- Kerberos Key Distribution Center (UDP-In)

In your environment ensure these rules are enabled and that clients can connect to the KDC (domain controller) over port 88.

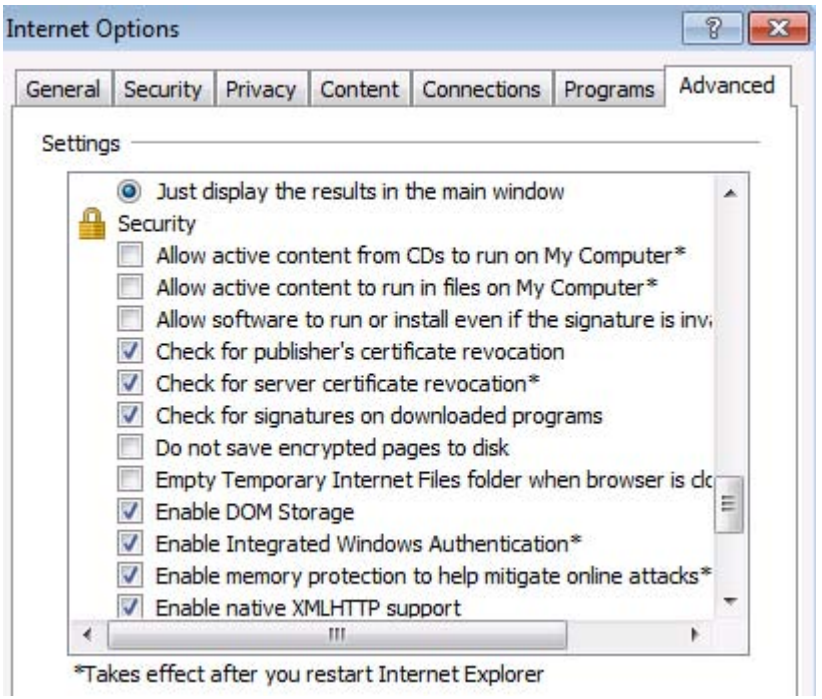
Test browser authentication

After configuring Active Directory, DNS and SharePoint Server you can now test whether Kerberos authentication is configured correctly by browsing to your web applications. When testing in the browser, ensure the following conditions are met:

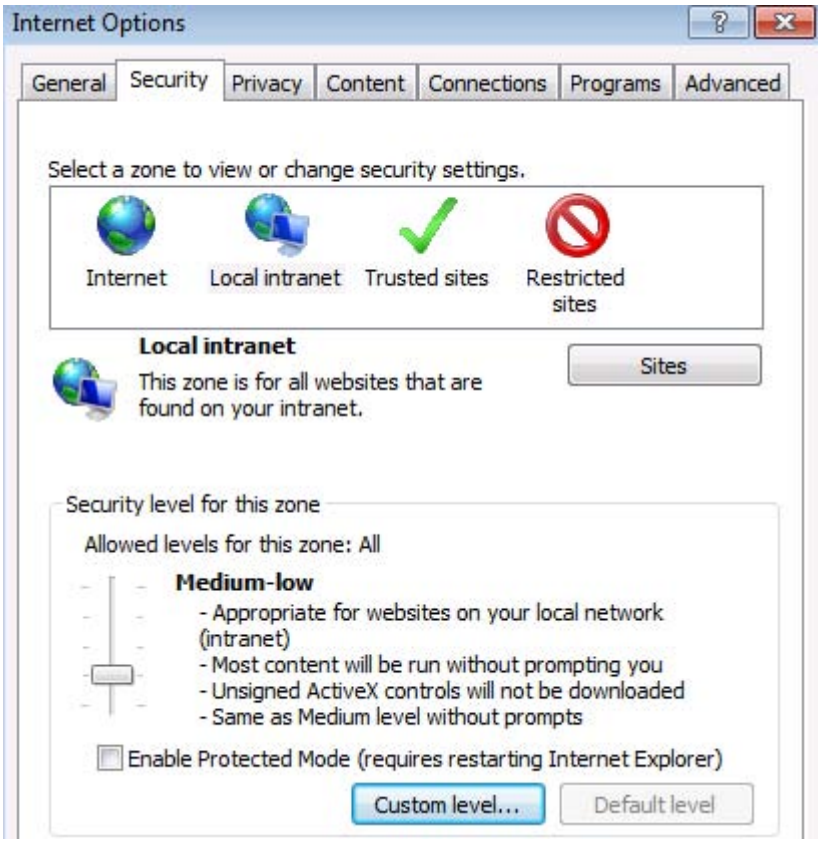
1. The test user is logged into a Windows XP, Vista, or Windows 7 computer joined to the domain that SharePoint Server is installed in, or is logged into a domain trusted by the SharePoint Server domain.

2. The test user is using Internet Explorer 7.0 or later (Internet Explorer 6.0 is no longer supported in SharePoint Server 2010; see Plan browser support (SharePoint Server 2010)).

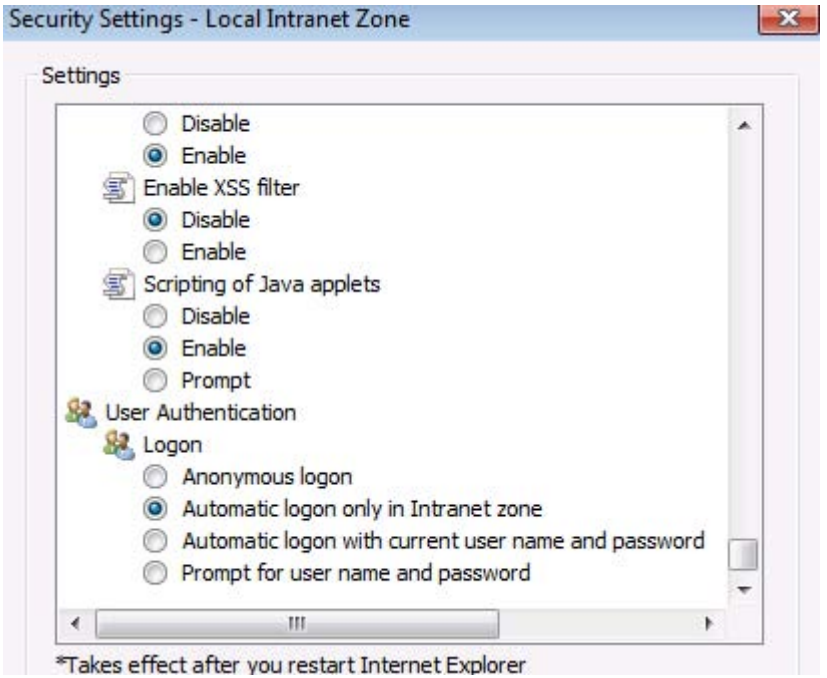
3. Integrated Windows authentication is enabled in the browser. Under **Internet Options** in the **Advanced** tab, make sure **Enable Integrated Windows Authentication*** is enabled in the Security section:



4. Local intranet is configured to automatically logon clients. Under Internet explorer option, in the **Security** tab, select **Local Intranet** and click the **Custom level** button. Scroll down and make sure that **Automatic logon only in Intranet zone** is selected.



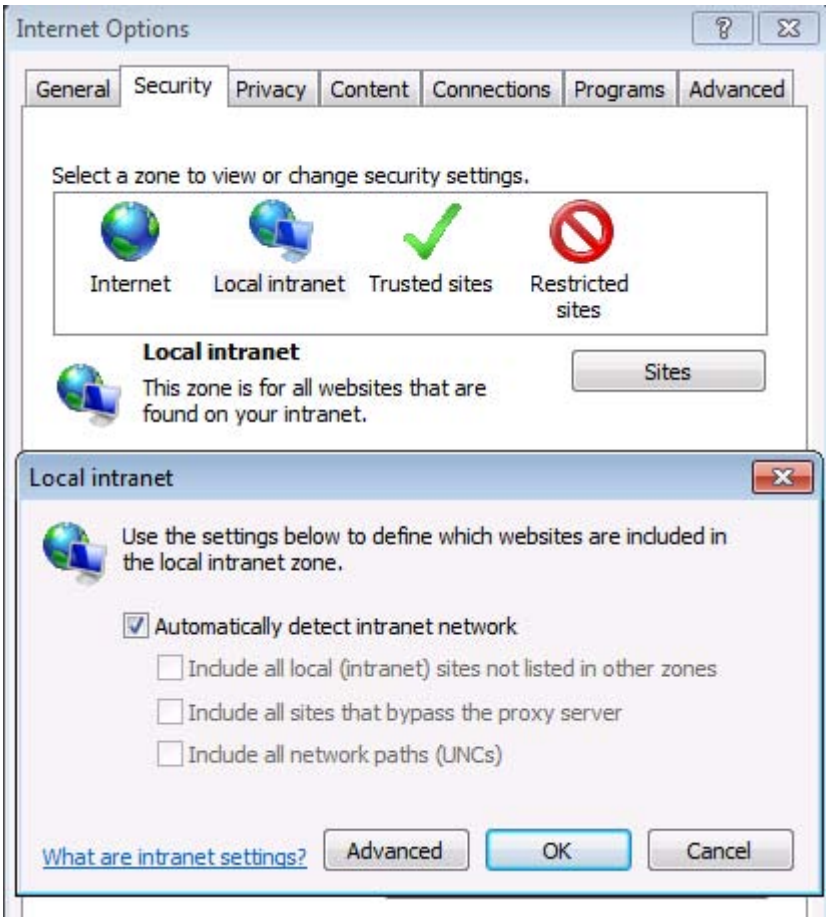
Scroll down and make sure “Automatic logon only in Intranet zone” is selected:



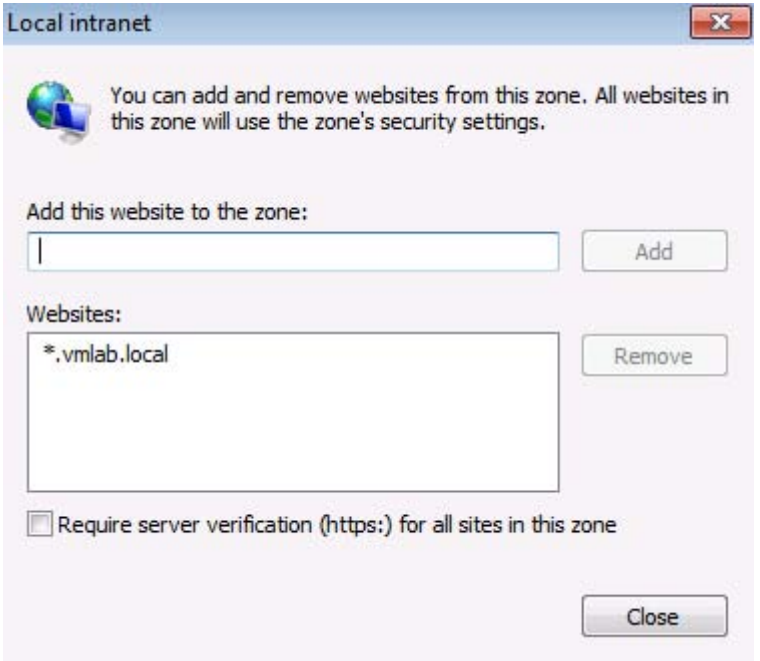
Note:

It is possible to configure automatic logon on other zones but the topic of IE security zones best practices it outside the scope of this paper. For this demonstration the intranet zone will be used for all tests.

5. Ensure that **Automatically detect intranet network** is selected in **Internet options->Security->Intranet Zone->Sites**.



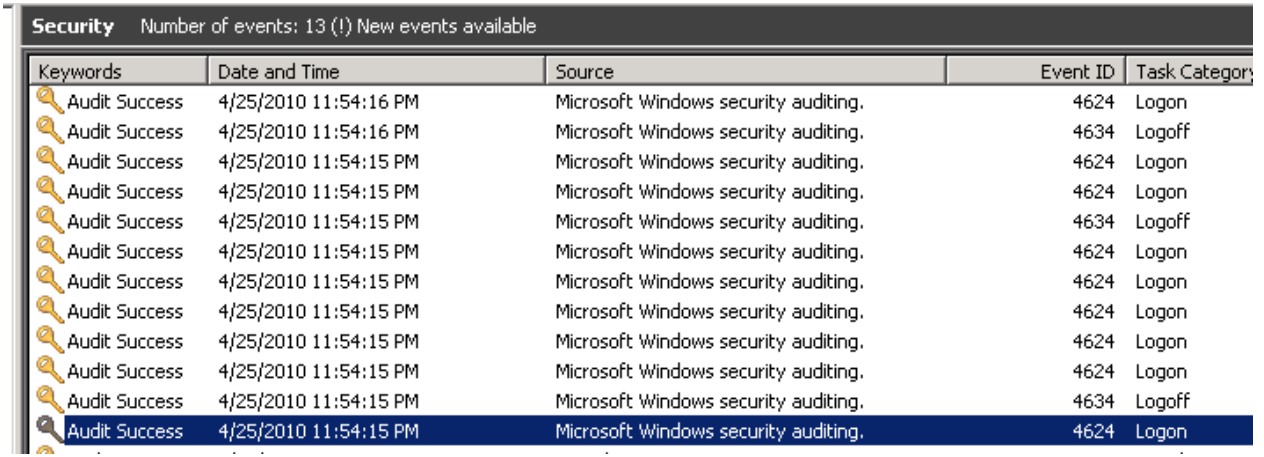
6. If you are using fully qualified domain names to access the SharePoint Server web applications, ensure that the FQDNs are included in the intranet zone, either explicitly or by wildcard inclusion (for example, “*.vmlab.local”).



The easiest way to determine if Kerberos authentication is being used is by logging into a test workstation and navigating to the web site in question. If the user isn't prompted for credentials and the site is rendered correctly, you can assume Integrated Windows authentication is working. The next step is to determine if the negotiate protocol was used to negotiate Kerberos authentication as the authentication provider for the request. This can be done in the following ways:

Front-end Web security logs

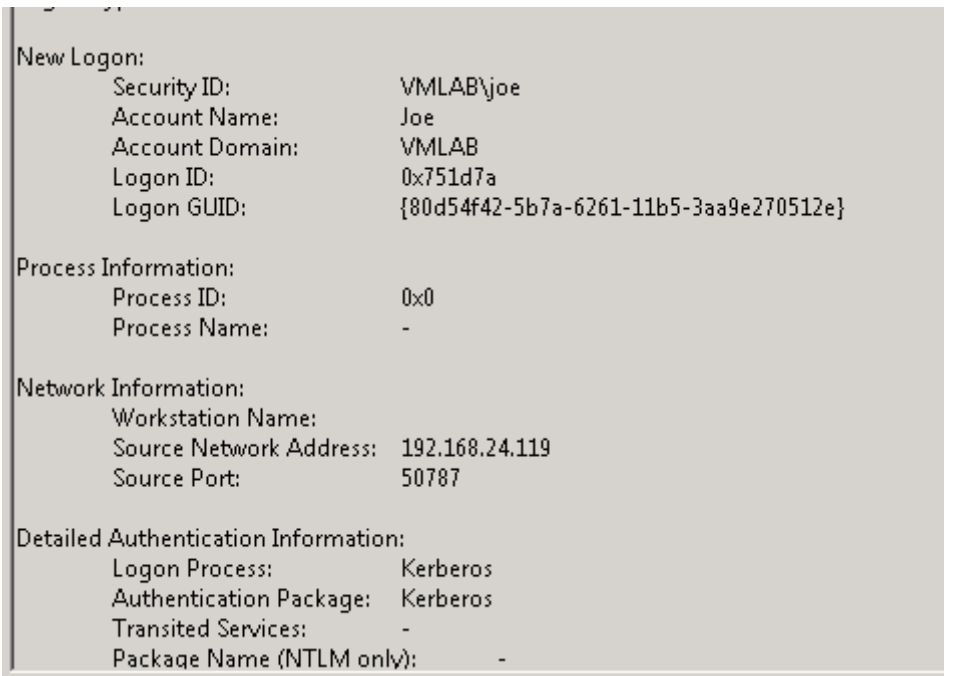
If Kerberos authentication is working correctly you will see Logon events in the security event logs on the front-end webs with event ID = 4624.



Security Number of events: 13 (!) New events available

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	4/25/2010 11:54:16 PM	Microsoft Windows security auditing.	4624	Logon
Audit Success	4/25/2010 11:54:16 PM	Microsoft Windows security auditing.	4634	Logoff
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4624	Logon
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4624	Logon
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4634	Logoff
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4624	Logon
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4624	Logon
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4624	Logon
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4624	Logon
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4634	Logoff
Audit Success	4/25/2010 11:54:15 PM	Microsoft Windows security auditing.	4624	Logon

In the general information for these events you should see the security ID being logged onto the computer and the Logon Process used, which should be **Kerberos**.



New Logon:
- Security ID: VMLAB\joe
- Account Name: Joe
- Account Domain: VMLAB
- Logon ID: 0x751d7a
- Logon GUID: {80d54f42-5b7a-6261-11b5-3aa9e270512e}

Process Information:
- Process ID: 0x0
- Process Name: -

Network Information:
- Workstation Name:
- Source Network Address: 192.168.24.119
- Source Port: 50787

Detailed Authentication Information:
- Logon Process: Kerberos
- Authentication Package: Kerberos
- Transited Services: -
- Package Name (NTLM only): -

KList

KList is a command line utility included in the default installation of Windows Server 2008 and Windows Server 2008 R2 which can be used to list and purge Kerberos tickets

on a given computer. To run KLIST, open a command prompt in Windows Server 2008 and type **Klist**.

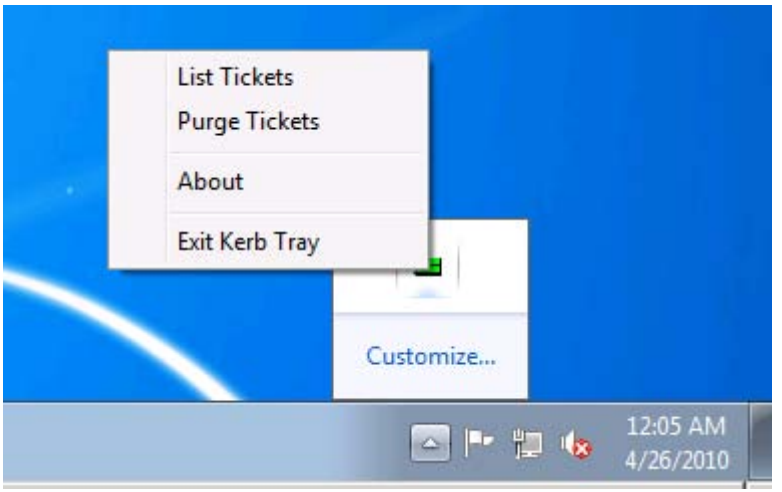
```
C:\>klist

Current LogonId is 0:0x53a11

Cached Tickets: (7)

#0>     Client: administrator @ VMLAB.LOCAL
        Server: krbtgt/VMLAB.LOCAL @ VMLAB.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a00000 -> forwardable forwarded renewable pre_authent
        Start Time: 5/30/2010 9:39:09 (local)
        End Time:   5/30/2010 18:41:15 (local)
        Renew Time: 6/3/2010 2:41:03 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96


#1>     Client: administrator @ VMLAB.LOCAL
        Server: krbtgt/VMLAB.LOCAL @ VMLAB.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 5/30/2010 8:41:15 (local)
        End Time:   5/30/2010 18:41:15 (local)
        Renew Time: 6/3/2010 2:41:03 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96


#2>     Client: administrator @ VMLAB.LOCAL
        Server: cifs/VMLabAD.VMLab.local @ VMLAB.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
```

If you want to purge the ticket cache, run Klist with the optional **purge** parameter: **Klist purge**

KerbTray

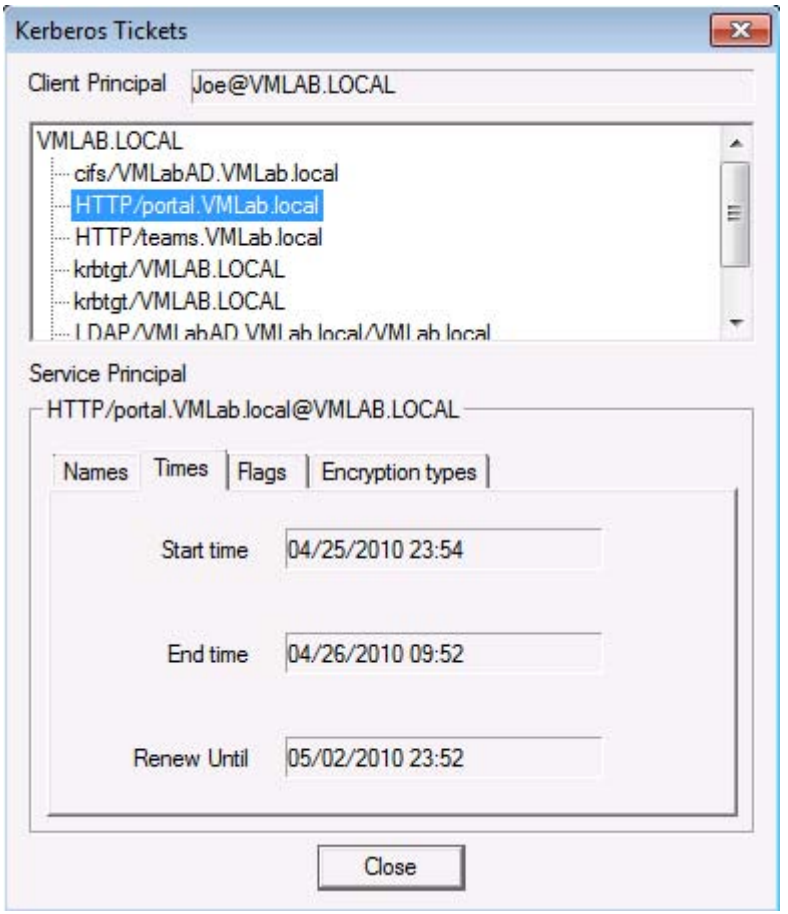
KerbTray is a free utility included with the Windows Server 2000 Resource Kit Tool that can be installed on your client computer to view the Kerberos ticket cache. Download and install from Windows 2000 Resource Kit Tool: Kerbtray.exe. Once you have it installed, perform the following actions:

1. Navigate to the web sites that use Kerberos Authentication.
2. Run KerbTray.exe.
3. View the Kerberos Ticket cache by right clicking on the kerb tray icon in the system tray and selecting **List Tickets**.



4. Validate the service tickets for the web applications you authenticated are in the list of cached tickets. In our example we navigated to the following web sites which have the following SPNs registered:

Web Site URL	SPN
http://portal	HTTP/Portal.vmlab.local
http://teams:5555	HTTP/Teams.vmlab.local

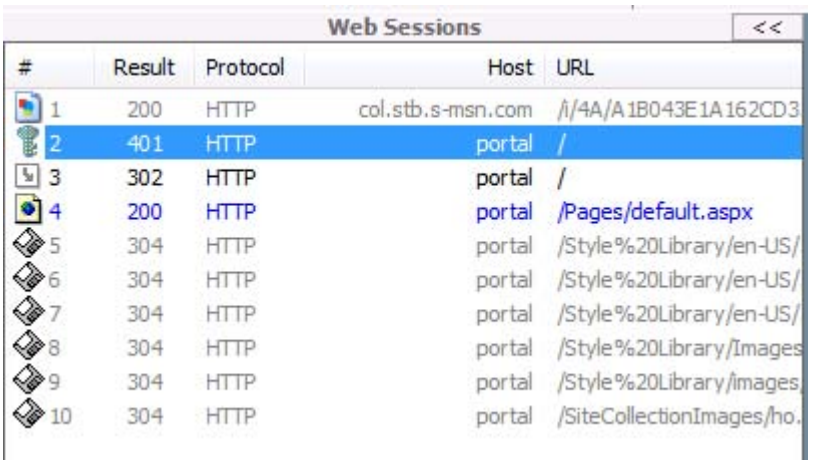

Kerberos Tickets
Client Principal
Joe@VMLAB.LOCAL
VMLAB.LOCAL
cifs/VMLabAD.VMLab.local
HTTP/portal.VMLab.local
HTTP/teams.VMLab.local
krbtgt/VMLAB.LOCAL
krbtgt/VMLAB.LOCAL
Service Principal
HTTP/portal.VMLab.local@VMLAB.LOCAL
Names
Times
Flags
Encryption types
Start time
04/25/2010 23:54
End time
04/26/2010 09:52
Renew Until
05/02/2010 23:52
Close

Fiddler

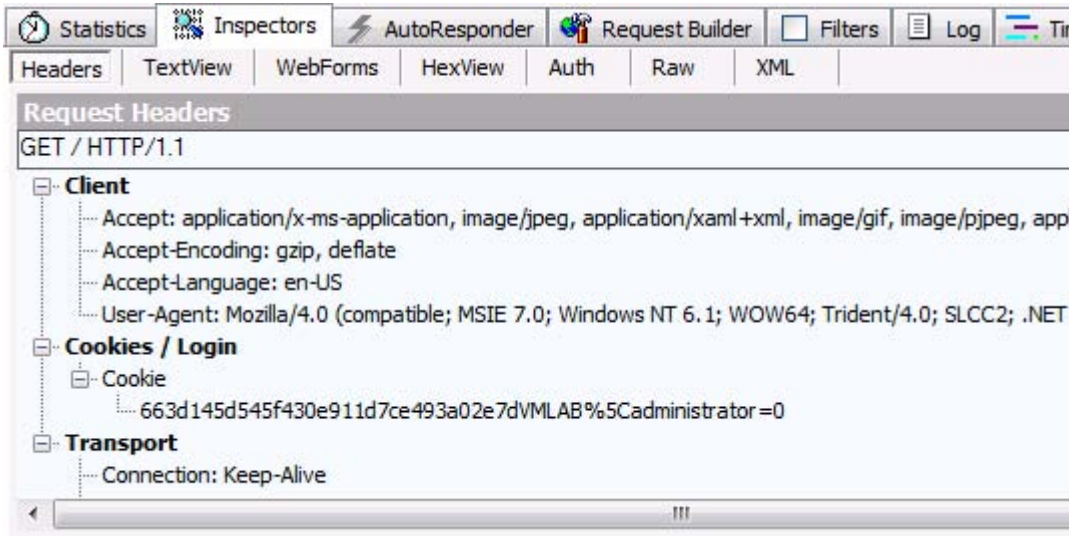
Fiddler is a free HTTP traffic analyzer that can be downloaded from the following location: http://www.fiddlertool.com/. In fiddler you will see the client and server negotiate Kerberos authentication and you will be able to see the client send the Kerberos Service Tickets to the server in the HTTP headers of each request. To validate that Kerberos authentication is working correctly using fiddler perform the following actions:

1. Download and install Fiddler (www.fiddlertool.com) on the client computer.
2. Log out of the desktop and log back in to flush any cached connections to the web server and force the browser to negotiate Kerberos authentication and perform the authentication handshake.
3. Start Fiddler.
4. Open Internet Explorer and browse to the web application (http://portal in our example).

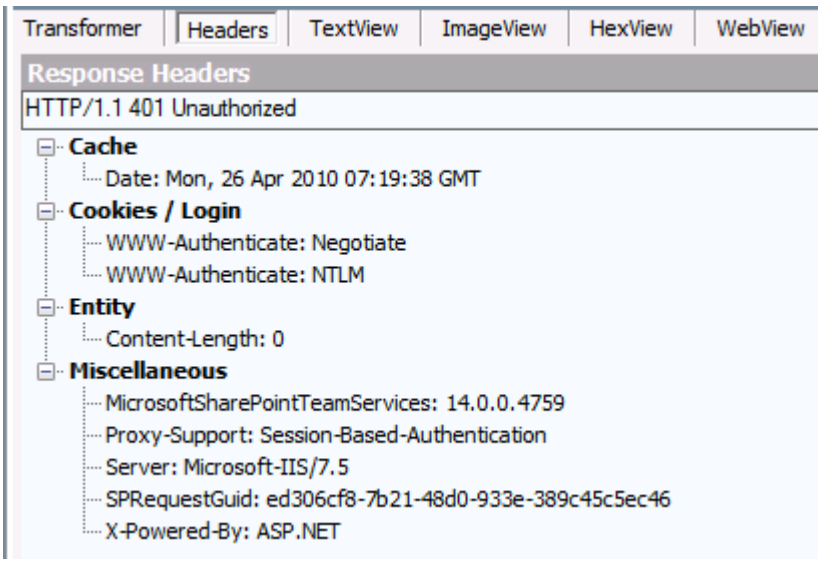
You should see the requests and responses to the SharePoint Server front-end web in Fiddler.



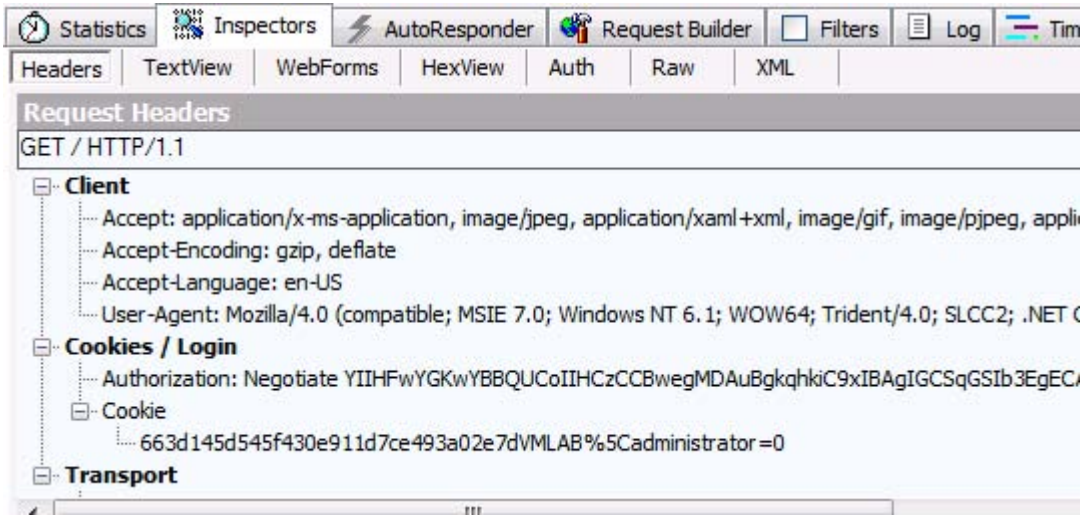
The first HTTP 401 is the browser attempt to do the GET request without authentication.



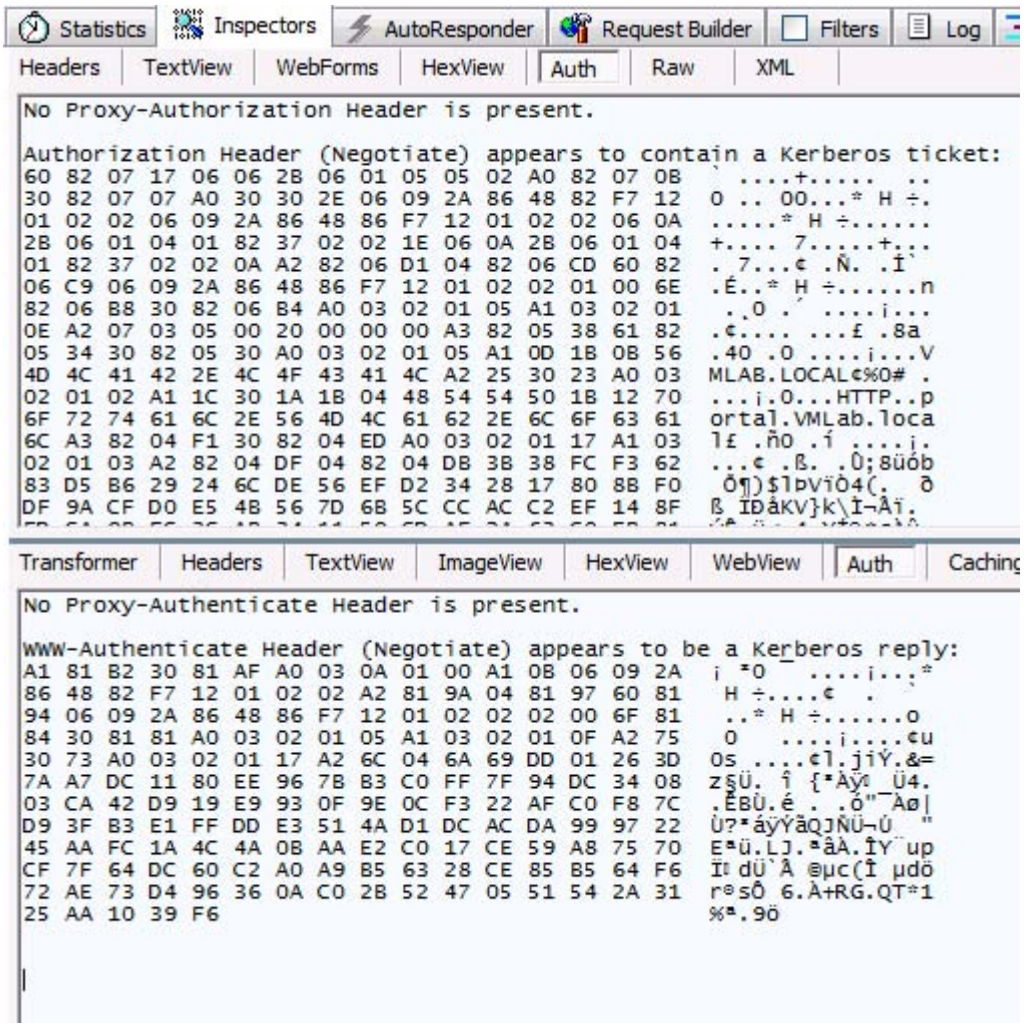
In response, the server sends back an "HTTP 401 – unauthorized" and in this response indicates what authentication methods it supports:



In the next request, the client resends the previous request, but this time sends the service ticket for the web application in the headers of the request:



If you select the “Auth” view within the Fiddler inspector window you will also see the Kerberos ticket in the request and the Kerberos response:



If authenticated successfully, the server will send back the requested resource.

NetMon 3.4

NetMon 3.4 is a free network packet analyzer from Microsoft that can be downloaded from the Microsoft Download Center: [Microsoft Network Monitor 3.4].

In NetMon you see all TCP request and responses to the KDC and the SharePoint Server web servers, giving you a complete view of traffic that makes up a complete authentication request. To validate that Kerberos authentication is working by using netmon, perform the following actions:

Configuring Kerberos authentication: Core configuration (SharePoint Server 2010)

1. Download and install NetMon 3.4 (Microsoft Network Monitor 3.4).

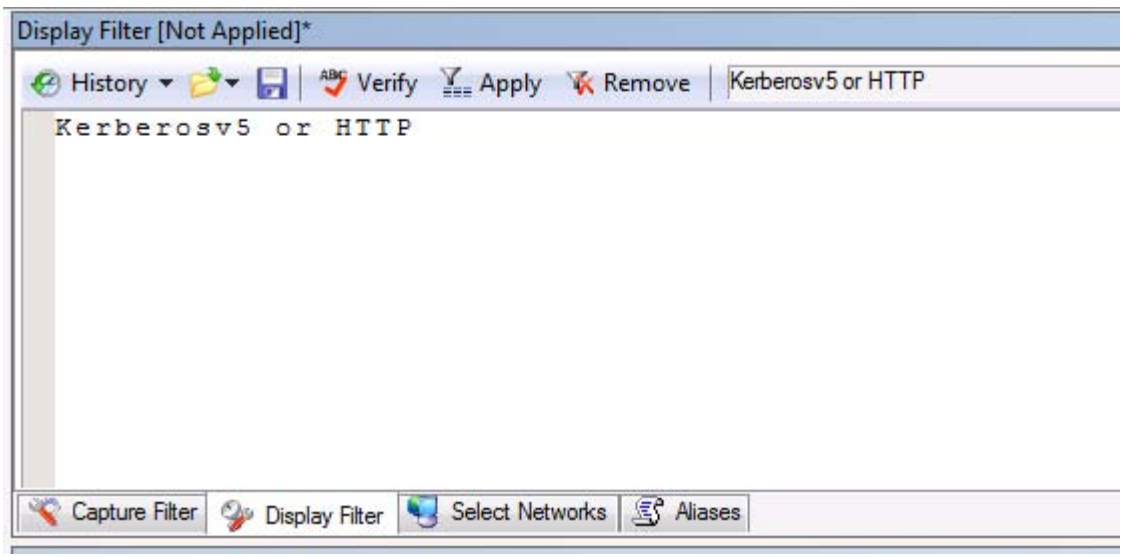
2. Log out of the client then log back in to flush the Kerberos ticket cache. Optionally you can use KerbTray to purge the ticket cache by right clicking on KerbTray and selecting **Purge Tickets**.

3. Start NetMon in administrator mode. Right-click the NetMonshortcut, and select **Run as Administrator**.

4. Start a new capture on the interfaces that connect to the active directory controller in your environment and the web front ends.

5. Open internet explorer and browse to the web application.

6. After the web site renders, stop the capture and add a display filter to show the frames for Kerberos authentication and HTTP traffic.



7. In the frames window you should see both HTTP and KerberosV5 traffic.

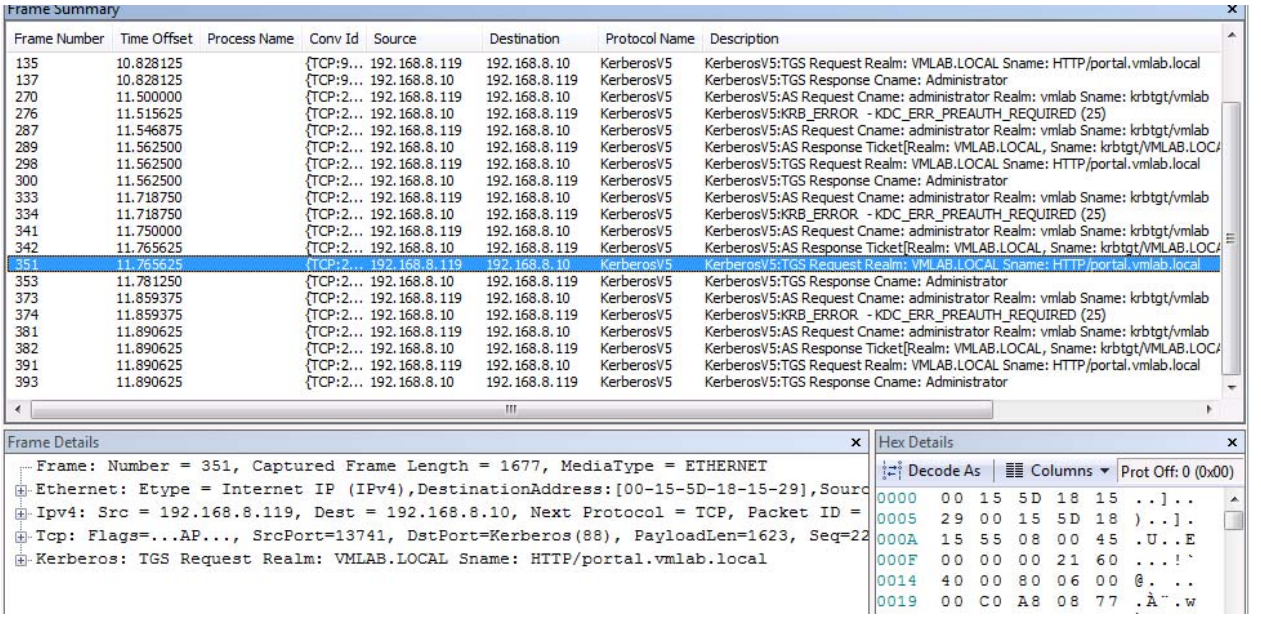
Source	Destination	Protocol Name	Description
192.168.24.119	192.168.24.140	HTTP	HTTP:Request, GET /
192.168.24.140	192.168.24.119	HTTP	HTTP:Response, HTTP/1.1, Status Code = 401, URL: / , Using NTLMX-Powered-By: Authentication
192.168.8.119	192.168.8.10	KerberosV5	KerberosV5:AS Request Cname: Administrator Realm: VMLAB.LOCAL Sname: krbtgt/VMLAB.LOCAL
192.168.8.10	192.168.8.119	KerberosV5	KerberosV5:KRB_ERROR - KDC_ERR_PREAUTH_REQUIRED (25)
192.168.8.119	192.168.8.10	KerberosV5	KerberosV5:AS Request Cname: Administrator Realm: VMLAB.LOCAL Sname: krbtgt/VMLAB.LOCAL
192.168.8.10	192.168.8.119	KerberosV5	KerberosV5:AS Response Ticket[Realm: VMLAB.LOCAL, Sname: krbtgt/VMLAB.LOCAL]
192.168.8.119	192.168.8.10	KerberosV5	KerberosV5:TGS Request Realm: VMLAB.LOCAL Sname: HTTP/portal.VMLab.local
192.168.8.10	192.168.8.119	KerberosV5	KerberosV5:TGS Response Cname: Administrator
192.168.24.119	192.168.24.140	HTTP	HTTP:Request, GET / , Using SPNEGO Authorization
192.168.24.119	192.168.24.140	HTTP	HTTP:Request, GET /Pages/default.aspx , Using SPNEGO Authorization

a. The first two frames are the original request/response where the client and server negotiate the use of Kerberos for authentication
b. The following KerberosV5 frames are the client requests for Ticket Granting Ticket for the VMLAL.Local Realm and the Kerberos service tickets for the SPN HTTP/portal.VMLAB.local
c. Finally the last HTTP frames are the client using the service tickets to authenticate with the web server and the server successfully authenticating the client and returning the response

Test Kerberos Authentication over SSL

To clearly demonstrate the SPNs requested when a client accesses an SSL protected resource, you can use a tool like Netmon to capture the traffic between client and server and examine the Kerberos service ticket requests.

1. Either logout and then re-login into the client computer, or clear all cached Kerberos tickets by using KerbTray.
2. Start a new NetMon capture on the client computer. Be sure to start NetMon with administrator permissions.
3. Browse to the web application by using SSL (in this example, https://portal.)
4. Stop the NetMon capture and examine the KerberosV5 traffic. For instructions on how to filter the capture display, see the instructions in the NetMon 3.4 section of this article.
5. Look for the TGS request the client sends. In the request you will see the SPN requested in the "Sname" parameter.



Note that the "Sname" is HTTP/portal.vmlab.local and not HTTPS/portal.vmlab.local.

Test SharePoint Server Search Index and Query

Verify browser access from the index server(s)

Before running a crawl, ensure that the index server can access the web applications and authenticate successfully. Log into the index server and open the test site collections in the browser. If the sites render successfully and no authentication dialogs appear, proceed to the next step. If any issues occur while accessing the sites in the browsers, go back over the previous steps to ensure all configuration actions were performed correctly.

Upload sample content and perform a crawl

In each site collection upload a "seed" document (one that is easily identifiable in search) to a document library in the site collection. For instance, create a text document containing the words "alpha, beta, delta" and save it to a document library in each site collection.

Next, browse to SharePoint Central Administration and start a full crawl on the Local SharePoint Sites content source (which should contain the two test site collections by default).

Test search

If indexing completed successfully, you should see searchable items in your index and no errors in the crawl log.

System Status

Crawl status	Online for crawling
Background activity	None
Recent crawl rate	0.00 items per second
Searchable items	59
Recent query rate	0.00 queries per minute
Propagation status	Idle
Default content access account	VMLAB\svcSP10Search
Contact e-mail address	someone@example.com
Proxy server	None
Scopes update status	Idle
Scopes update schedule	Automatically scheduled
Scopes needing update	0
Search alerts status	Off Enable
Query logging	On Disable

Crawl History

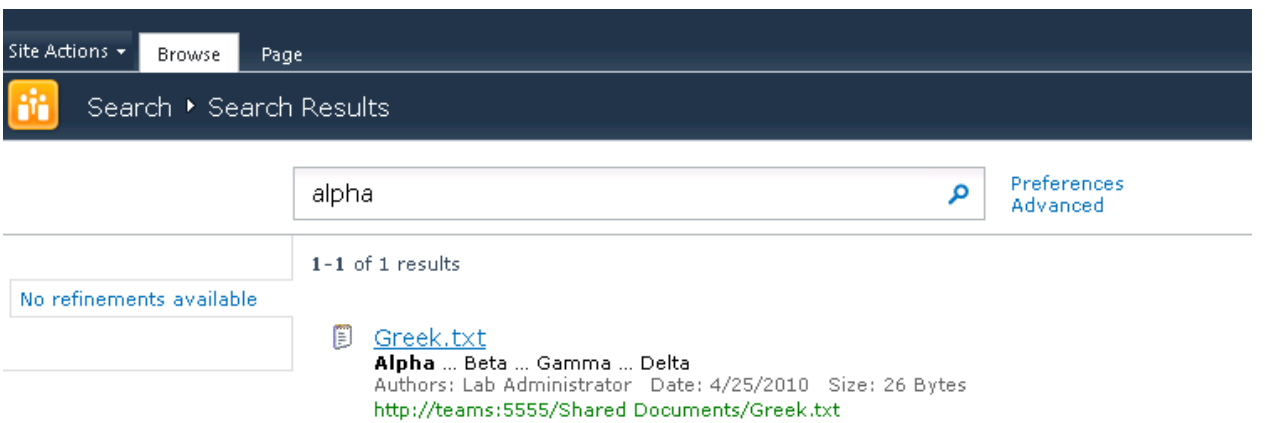
Content Source	Type	Start Time	End Time	Duration	Success	All Errors
Local SharePoint sites	Full	4/25/2010 7:53 PM	4/25/2010 7:58 PM	00:04:40	127	0

Note: If you have configured the User Profile Application (UPA) and are performing a crawl on the profile store be sure to configure the appropriate permissions on the UPA to allow the content access account to access profile data. If you have not configured the UPA permissions you will receive errors in the crawls logs indicating the crawler could not access the profile service because it received an HTTP 401 when trying to access the service. The 401 returned is not due to Kerberos, but instead is due to the content access account not having permissions to read profile data.

Note:

If you have configured the User Profile Application (UPA) and are performing a crawl on the profile store, be sure to configure the appropriate permissions on the UPA to allow the content access account to access profile data. If you have not configured the UPA permissions you will receive errors in the crawls logs indicating the crawler could not access the profile service because it received an HTTP 401 when trying to access the service. The 401 returned is not due to Kerberos, but instead is due to the content access account not having permissions to read profile data.

Next, browse to each site collection and perform a search for the seed document. Each site collection's search query should return the seed document uploaded.



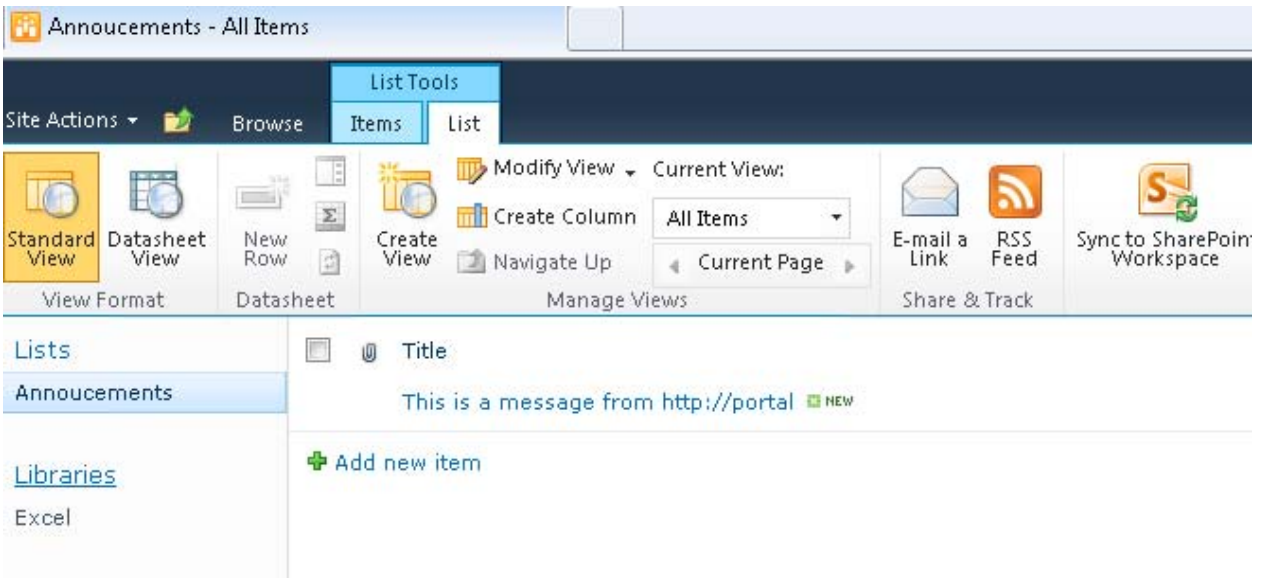
Test front-end Web delegation

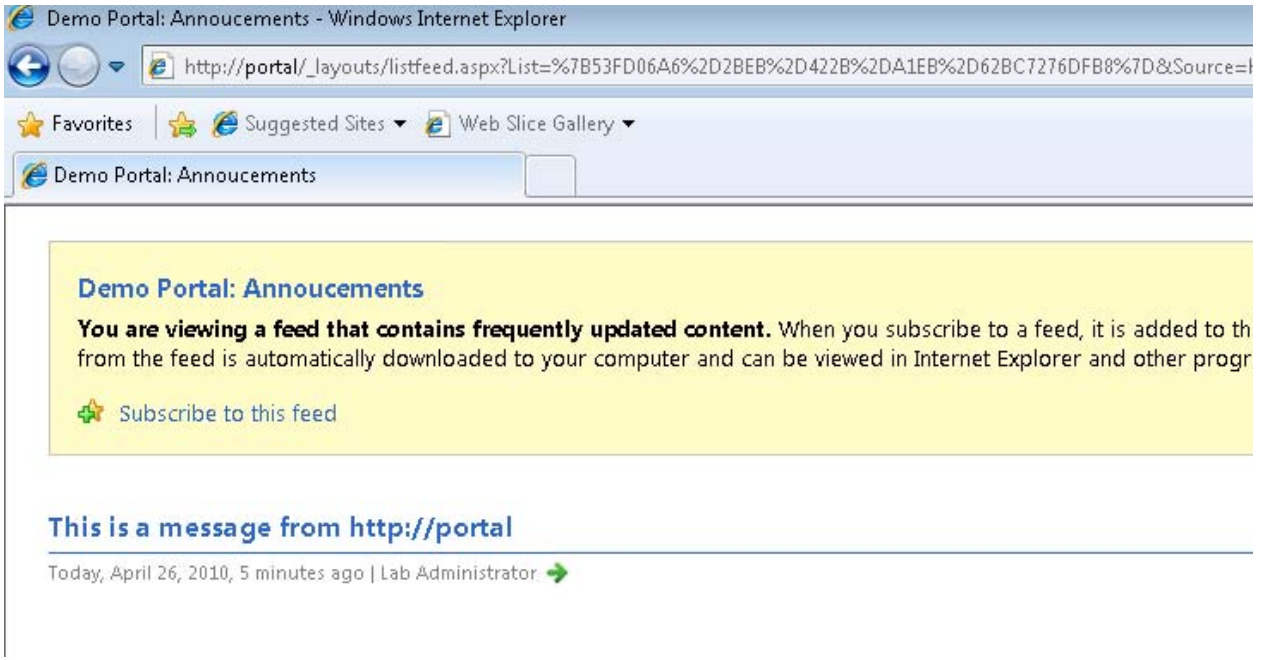
As a last step in this scenario, you use the RSS viewer web part on each site collection to ensure that delegation is working both locally and remotely.

Configure RSS feed sources on each site collection

For the portal application you have to enable RSS feeds on the Site Collection. To turn on RSS feeds follow the instructions in Manage RSS Feeds on Office.com.

Once RSS feeds are enabled, create a new custom list and add a list item for testing purposes. Navigate to the List toolbar menu and click **RSS Feed** to view the RSS feed. Copy the feed URL to use it in the following steps.





Perform this step for each site collection.

Add RSS view web parts to the home page of each site collection

On the portal application you'll need to enable the SharePoint Enterprise Features site collection feature to use the RSS viewer web part. Once enabled add two RSS viewer web parts to the home page. For the first web part, configure the feed URL to point at the local RSS feed you created in the previous step. For the second web part, configure the feed URL to point at the remote feed URL. When completed, you should see both web parts successfully rendering content from the local and remote RSS feeds.

documents, but they cannot publish the pages, images, or documents. Workflow is enabled in the Pages library, and content approval is enabled in the Documents and Images libraries.

RSS Viewer [2]

Demo: Announcements

This is a message from http:Teams:5555

Get Started with Microsoft SharePoint Foundation!

RSS Viewer [1]

Demo Portal: Annoucements

This is a message from http://portal

Kerberos authentication for SQL OLTP (SharePoint Server 2010)

Published: December 2, 2010

In this scenario we walk through the process of configuring Kerberos authentication for the SQL Server cluster in our sample environment. Once that process is complete, we validate that SharePoint Server services are authenticated with the cluster by using the Kerberos protocol.

In this scenario, you do the following things:

- Configure an existing SQL Server 2008 R2 cluster to use Kerberos authentication
- Verify that the client can authenticate with the cluster by using Kerberos authentication
- Create a test database and sample data to be used in later scenarios

Note:

It is not required to use Kerberos authentication for SQL Server for core SharePoint Server data services (for example, connections to platform databases). The sample environment has a sole SQL Server cluster that hosts additional sample databases used in later scenarios. For delegation to work correctly in these scenarios, the SQL Server cluster must accept Kerberos authenticated connection.

Note:

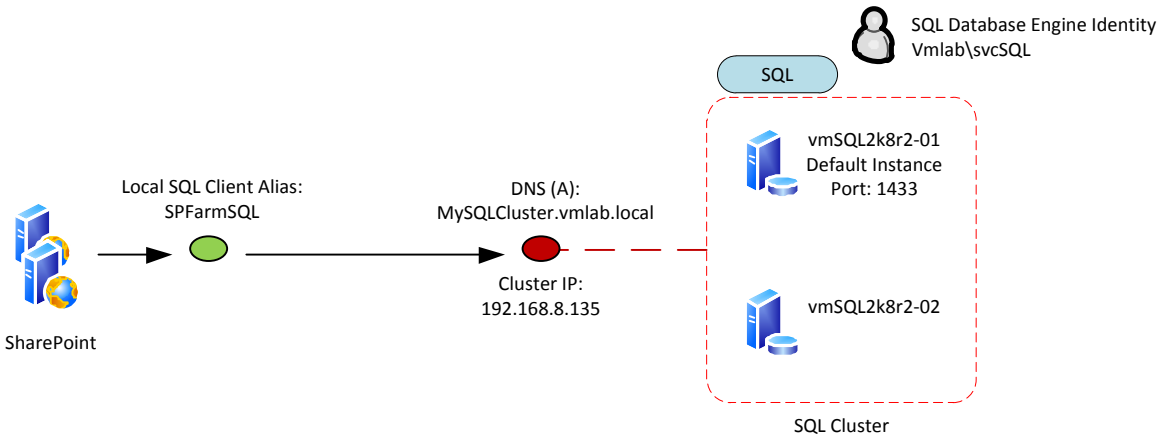
If you are installing on Windows Server 2008, you may need to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

Configuration checklist

Area of configuration	Description
Configure DNS	Create DNS (A) host records for the SQL Server cluster IP
Configure Active Directory	Create Service Principal Names (SPNs) for the SQL Server service
Verify SQL Server Kerberos configuration	Use SQL Server Management Studio to query SQL connection metadata to ensure the Kerberos authentication protocol is used

Scenario environment details



This scenario demonstrates a SharePoint Server farm configured to use a SQL alias for a connection to a SQL Server cluster that is configured to use Kerberos authentication.

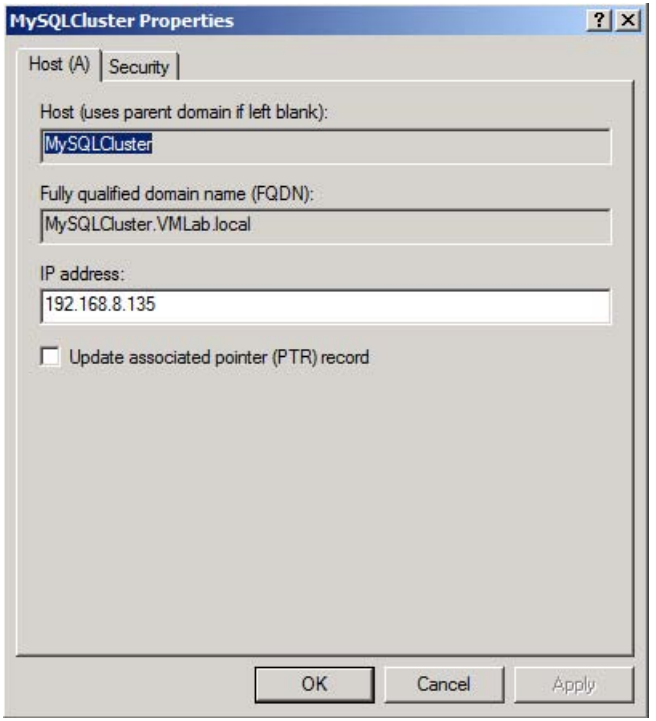
Step-by-step configuration instructions

Configure DNS

Configure DNS for the SQL Server cluster in your environment. In this example we have one SQL Server cluster, MySqlCluster.vmlab.local, running on port 1433 at cluster IP 192.168.8.135/4. The cluster is Active/Passive with the SQL Server database engine running on the default instance of the first node.

For general information about how to configure DNS, see Managing DNS Records.

In this example, we configured a DNS (A) record for the SQL Server cluster.



Note:

Technically, because SQL Server SPNs include an instance name (if you are using the second-named instance on the same computer), you can register the DNS host for the cluster as a CNAME alias and avoid the CNAME issue described in Appendix A, Kerberos configuration known issues (SharePoint Server 2010). However, if you choose to use CNAMEs, you have to register an SPN using the DNS (A) record host name the CNAME aliases.

Configure Active Directory

For SQL Server to authenticate clients using Kerberos authentication, you have to register a service principal name (SPN) on the service account that is running SQL Server. Service principal names for the SQL Server database engine use the following format for configurations that are using the default instance and not a SQL Server named instance:

MSSQLSvc/<FQDN>:port

For more information about registering SPNs for SQL Server 2008, see Registering a Service Principal Name.

In our example, we configured the SQL Server SPN on the SQL Server database engine service account (vmlab\svcSQL) with the following SetSPN command:

```
SetSPN -S MSSQLSVC/MySQLCluster.vmlab.local:1433 vmlab\svcSQL
```

SQL Server named instances

If you use SQL Server named instances instead of the default instance, you have to register SPNs specific to the SQL Server instance and for the SQL Server browser service. See the following articles for more information about configuring Kerberos authentication for names instances:

- Registering a Service Principal Name
- An SPN for the SQL Server Browser service is required when you establish a connection to a named instance of SQL Server 2005 Analysis Services or of SQL Server 2005

SQL aliases

As a best practice, when building your farm you should consider using SQL aliases for connections to your SQL Server computer. If you choose to use SQL aliases, the Kerberos SPN format for those connections does not change. You continue to use the registered DNS host name (A record) in the SPN for SQL Server. For example, if you register an alias "SPFARMSQL" for "MySQLCluster.vmlab.local" the SPN when you are connecting to SPFarmSQLremains "MSSQLSVC/MySQLCluster.vmlab.local:1433".

Verify SQL Server Kerberos configuration

When DNS and Service Principal Names are configured, you can reboot the computers that are running SharePoint Server and verify that SharePoint Server services now authenticate with SQL Server by using Kerberos authentication.

To verify the cluster configuration

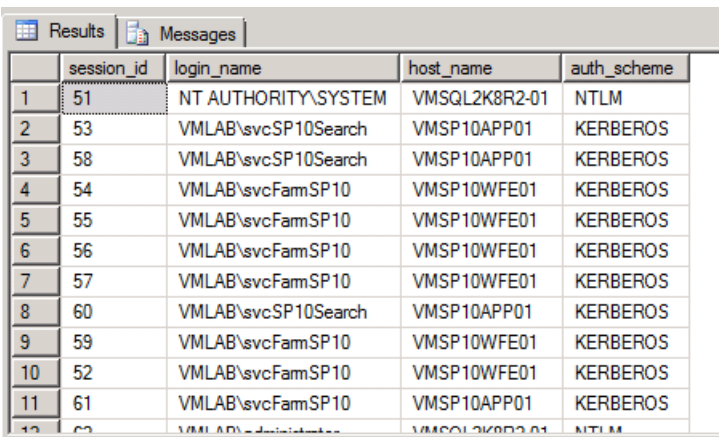
1. **Reboot the computers that are running SharePoint Server** This action restarts all services and forces them to re-connect and re-authenticate by using Kerberos authentication.

2. Open SQL Server Management Studio and run the following query:

```
Select
    s.session_id,
    s.login_name,
    s.host_name,
    c.auth_scheme
from
sys.dm_exec_connections c
innerjoin
sys.dm_exec_sessions s
on c.session_id = s.session_id
```

The query returns metadata about each session and connection. The session data helps identify the connection source, and the session information reveals the authentication scheme for the connection.

3. Verify that the SharePoint Server services are authenticating by using Kerberos authentication:



Results | Messages

	session_id	login_name	host_name	auth_scheme
1	51	NT AUTHORITY\SYSTEM	VMSQL2K8R2-01	NTLM
2	53	VMLAB\svcSP10Search	VMSP10APP01	KERBEROS
3	58	VMLAB\svcSP10Search	VMSP10APP01	KERBEROS
4	54	VMLAB\svcFarmSP10	VMSP10WFE01	KERBEROS
5	55	VMLAB\svcFarmSP10	VMSP10WFE01	KERBEROS
6	56	VMLAB\svcFarmSP10	VMSP10WFE01	KERBEROS
7	57	VMLAB\svcFarmSP10	VMSP10WFE01	KERBEROS
8	60	VMLAB\svcSP10Search	VMSP10APP01	KERBEROS
9	59	VMLAB\svcFarmSP10	VMSP10WFE01	KERBEROS
10	52	VMLAB\svcFarmSP10	VMSP10WFE01	KERBEROS
11	61	VMLAB\svcFarmSP10	VMSP10APP01	KERBEROS

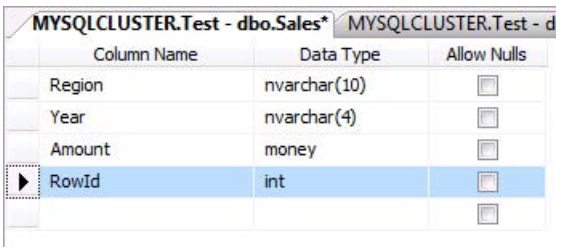
4. . If Kerberos authentication is configured correctly, you see **Kerberos** in the **auth_scheme** column of the query results.

Create a test SQL Server database and test table

To test delegation across the various SharePoint Server service applications covered in the scenarios in this document, you have to configure a test data source for those services to access. In the final step of this scenario, you configure a test database called "Test" and a test table called "Sales" to be used later.

1. In SQL Server Management Studio, create a new database called "Test". Keep the default settings when creating this database.

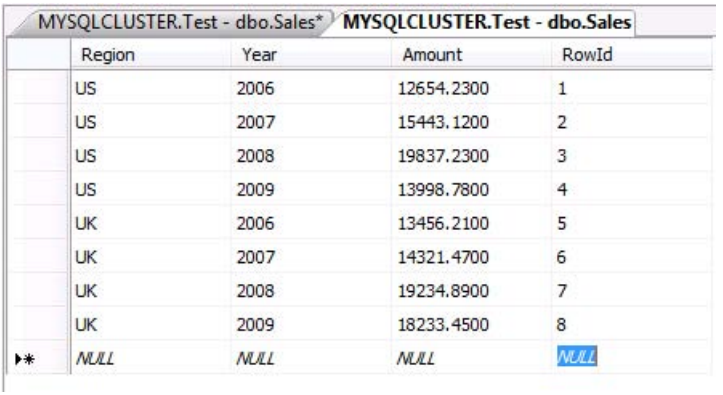
2. In the Test database, create a new table with the following schema:



Column Name	Data Type	Allow Nulls
Region	nvarchar(10)	No
Year	nvarchar(4)	No
Amount	money	No
RowId	int	No

3. Save the table with the name "Sales".

4. In Management Studio, populate the table with test data. The data itself does not matter and does not affect the function of later scenarios. A few rows of data will suffice. In the example environment we populated the table with the following data:



MYSQLCLUSTER.Test - dbo.Sales* | MYSQLCLUSTER.Test - dbo.Sales

	Region	Year	Amount	RowId
	US	2006	12654.2300	1
	US	2007	15443.1200	2
	US	2008	19837.2300	3
	US	2009	13998.7800	4
	UK	2006	13456.2100	5
	UK	2007	14321.4700	6
	UK	2008	19234.8900	7
	UK	2009	18233.4500	8
▶*	NULL	NULL	NULL	NULL

Kerberos authentication for SQL Server Analysis Services (SharePoint Server 2010)

Published: December 2, 2010

In this scenario you do the following things:

- Configure Analysis Service instances in the SQL Server 2008 R2 cluster to use Kerberos authentication
- Verify that the client can authenticate with the cluster by using Kerberos authentication

Enabling Kerberos authentication for SQL Server Analysis Services is similar to SQL Server

Note:

If you are installing on Windows Server 2008, you may have to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

Configuration checklist

Area of configuration	Description
Configure Active Directory	Create Service Principal Names (SPNs) for the Analysis Services instance
Verify SQL Kerberos	Connect to the Analysis Services instance in Excel

Area of configuration	Description
Configuration	2010

Step-by-step configuration instructions

Configure Active Directory

For SQL Server Analysis Services to authenticate clients by using Kerberos authentication, you have to register a service principal name (SPN) on the service account that is running SQL Server. The SPN for a default Analysis Services instance uses the following format:

MSOLAPSvc.3/<FQDN>

If you are using a named instance of Analysis Services, note that you cannot specify a port after the colon. If you do, it is interpreted as part of the hostname or domain name. Instead, you must use the actual instance name for all functionality to work correctly.

MSOLAPSvc.3/<FQDN>:instanceName

For more information about registering SPNs for SQL Server 2008, see http://support.microsoft.com/kb/917409.

This scenario assumes a default Analysis Services instance. We will configure the Analysis Services SPN on the Analysis Services service account (vmlab\svcSQLAS) with the following SetSPN command:

```
SetSPN -S MSOLAPSvc.3/MySQLCluster.vmlab.localvmlab\svcSQLAS
```

SQL Server named instances

If you use SQL Server named instances instead of the default instance, you have to register SPNs specific to the SQL Server instance and for the SQL Server browser service. See the following articles for more information about configuring Kerberos authentication for named instances:

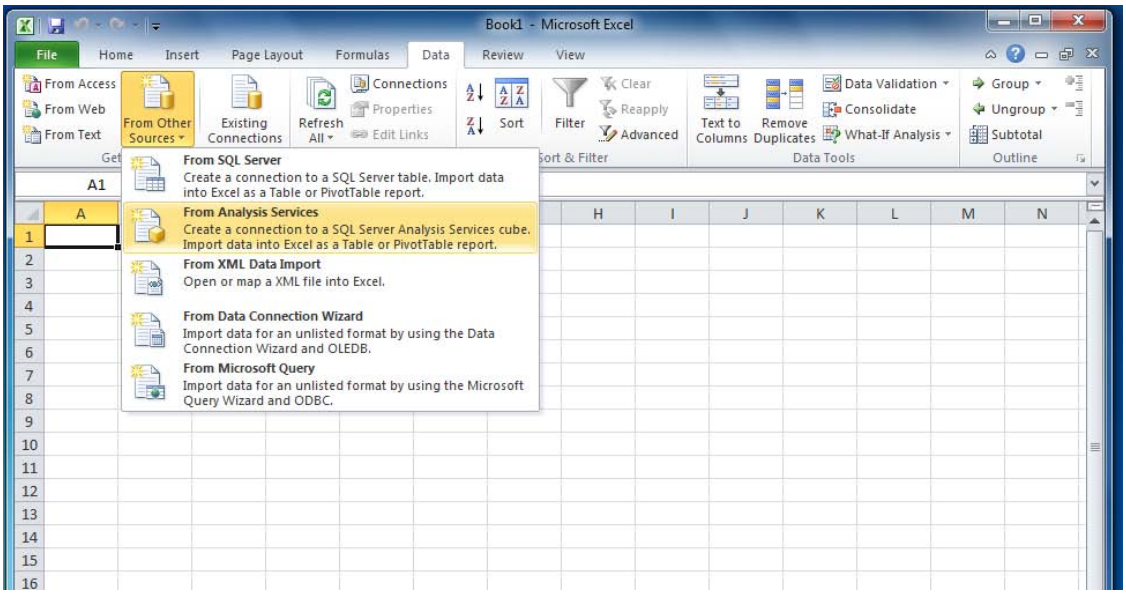
- Registering a Service Principal Name

- An SPN for the SQL Server Browser service is required when you establish a connection to a named instance of SQL Server 2005 Analysis Services or of SQL Server 2005

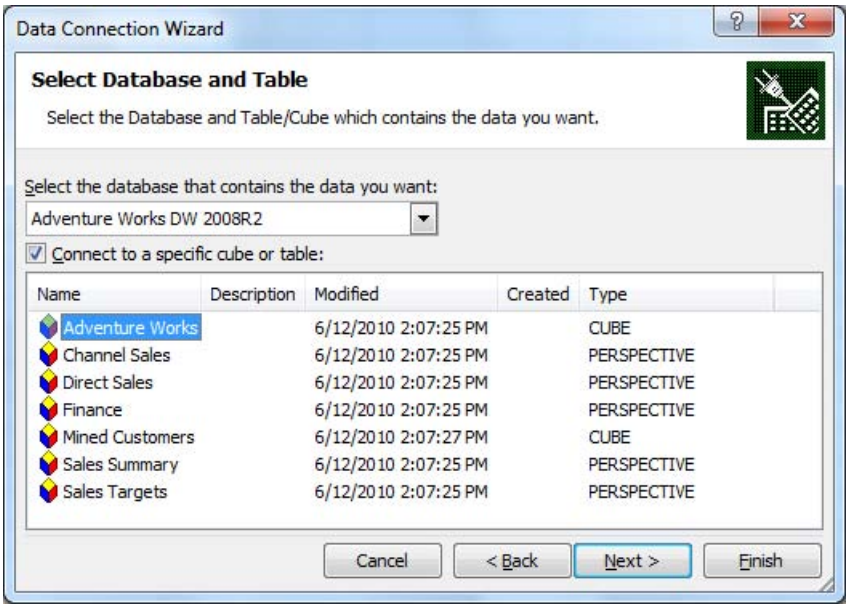
Verify SQL Server Kerberos configuration

Once the SPN is configured, verify the Kerberos connection to the cluster by using Excel 2010.

1. Open Excel 2010 on the client computer by using a domain account that has access to at least one database in the Analysis Services instance and open a data connection to your Analysis Services instance by selecting the **Data** tab, clicking **From Other Sources**, and then clicking **From Analysis Services**.



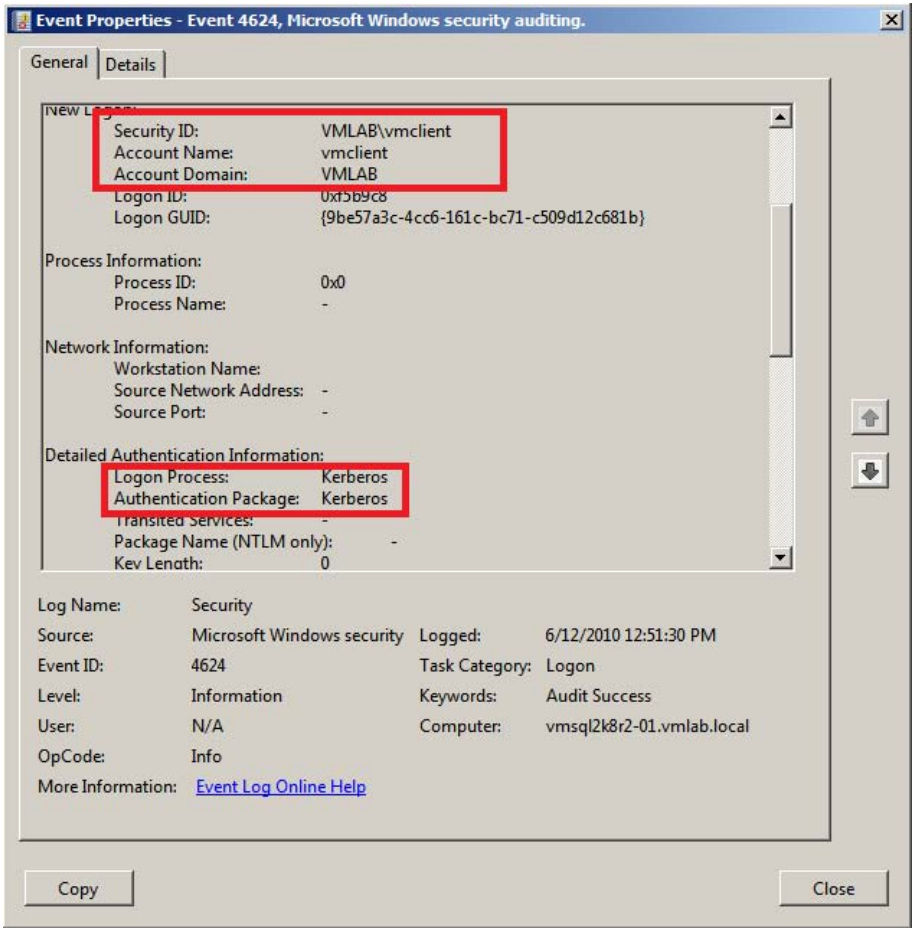
2. In the Data Connection Wizard, type **MySQLCluster** in the **Server name** box, then click **Next**. If Kerberos authentication is working, then you can see all the databases that you already have the permission to see.



Note:

To use the AdventureWorks 2008 R2 sample databases, download from Microsoft SQL Server Community Projects & Samples and follow the installation instructions.

3. Open the event viewer on the database server (vmsql2k8r2-01). You should now be able to see an audit success in the security log similar to the one you see in the verification steps for Scenario 2, Kerberos authentication for SQL OLTP (SharePoint Server 2010).


Event Properties - Event 4624, Microsoft Windows security auditing.
General
Details
Security ID: VMLAB\vmclient
Account Name: vmclient
Account Domain: VMLAB
Logon ID: 0xf5b9c8
Logon GUID: {9be57a3c-4cc6-161c-bc71-c509d12c681b}
Process Information:
Process ID: 0x0
Process Name: -
Network Information:
Workstation Name:
Source Network Address: -
Source Port: -
Detailed Authentication Information:
Logon Process: Kerberos
Authentication Package: Kerberos
Transited Services: -
Package Name (NTLM only): -
Key Length: 0
Log Name: Security
Source: Microsoft Windows security
Logged: 6/12/2010 12:51:30 PM
Event ID: 4624
Task Category: Logon
Level: Information
Keywords: Audit Success
User: N/A
Computer: vmsql2k8r2-01.vmlab.local
OpCode: Info
More Information: Event Log Online Help
Copy
Close

Identity delegation for SQL Server Reporting Services (SharePoint Server 2010)

Published: December 2, 2010

In this scenario you configure a pair of load-balanced SQL Server Reporting Services (SSRS) servers in a scaled-out configuration running in SharePoint integrated mode. The servers are configured to accept Kerberos authentication and they delegate authentication to a back-end SQL Server cluster.

In this scenario, the SharePoint Server farm and Reporting Services data source are both in the same domain; therefore in this scenario we configure Kerberos constrained delegation to allow identity delegation to the back-end data source. If you are required to authenticate with data sources in other domains within the same forest, you have to configure basic (unconstrained) Kerberos delegation. Remember that Reporting Services does not leverage the C2WTS and therefore can use basic delegation.

Note:

If you are installing on Windows Server 2008, you may have to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

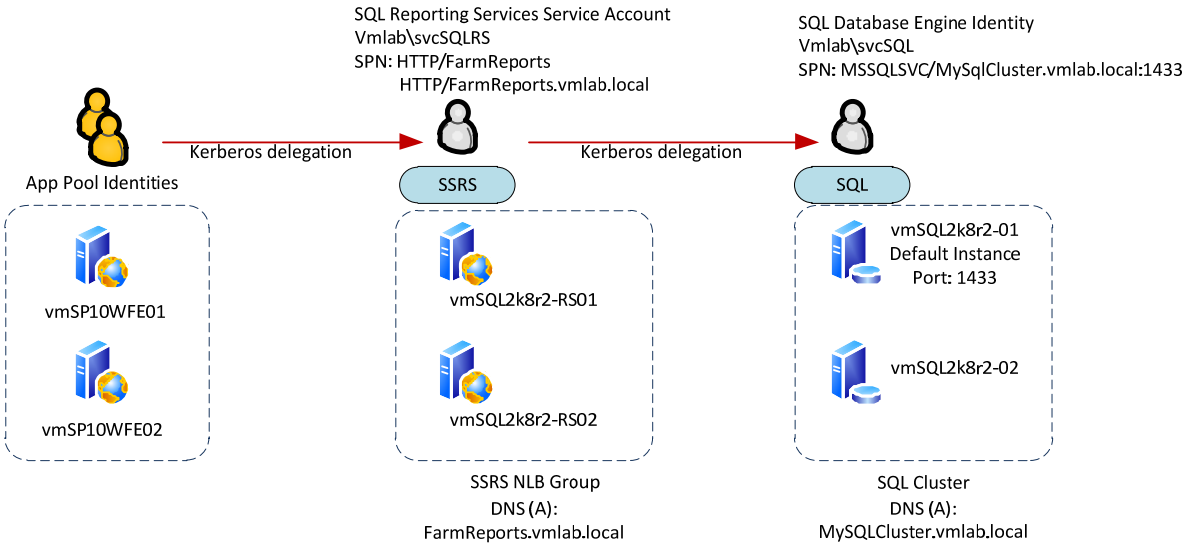
Scenario dependencies

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP
- (Optional) Scenario 3: Kerberos Authentication for SQL Analysis Services

Configuration checklist

Area of configuration	Description
Active Directory	Create SSRS service account Configure Kerberos constrained delegation
SQL Server Reporting Services	Install and configure SSRS in load-balanced, scale out mode Modify Web.Config Modify ReportingServer.config
Configure SharePoint Server	Configure Reporting Services integration Add a report server to the integration Set server defaults
Verify configuration	Create a document library for reports Configure site collection setting for Reporting Services Create and publish a test report in SQL Server Business Intelligence Studio View the test report in Internet Explorer

Scenario environment details



In this scenario, the Internet Information Services (IIS) application pool service accounts are configured to delegate to the SQL Server Reporting Services (SSRS) service. The SSRS service account is configured to delegate credentials to the SQL Server service. Note that SQL Server Reporting Services in SharePoint integrated mode does not leverage intra-farm Claims authentication and requires Kerberos authentication for delegated authentication. For more information, see Claims Authentication and Reporting Services.

Cross-domain Kerberos delegation

In this example, the data source that SSRS connects to resides in the same domain as the SSRS servers. In some situations you may want to access data sources outside of the domain that SSRS resides in. To authenticate with delegation cross domain, you have to configure basic (unconstrained) delegation on the SSRS service account. Remember that this is possible because the SSRS service does not rely on the Claims to Windows Token Service (C2WTS), therefore does not require protocol transition through Kerberos constrained delegation. Also note that cross-forest delegation is not possible, even with basic delegation.

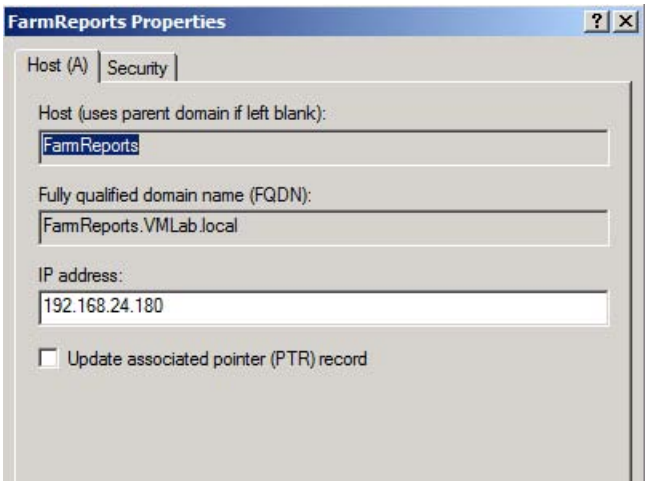
Step-by-step configuration instructions

Configure DNS

Configure DNS for the SSRS NLB server group in your environment. In this example we have two SSRS servers, VMSSRS01 and VMSSRS02, which are load-balanced and resolve to the same NLB VIP (192.168.24.180/24). The VIP will be mapped to the host FarmReports and will have the URL http://FarmReports.

For general information about how to configure DNS, see Managing DNS Records.

Configure a new DNS A Record for the SSRS host. In this example we have a host FarmReports configured to resolve to the load balanced VIP.



Active Directory directory service

Create SSRS service account

As a best practice, SQL Server Reporting Services should run under its own domain identity. In this example, the following accounts were created:

Service	Service Identity
SQL Server Reporting Services	vmlab\svcSQLRS

Configure Service Principal Names

For SSRS to connect and authenticate with external data sources using Kerberos authentication, the Report Server Web Service and Report Manager service accounts and the service account for the external data source must have service principal names configured. Refer to scenarios 1 and 2 (Core configuration and Kerberos authentication for SQL OLTP) in this series of articles to configure and validate the necessary SPNS on the SharePoint Server web applications and SQL Server service accounts. For the SSRS servers, the following SPNs were defined:

DNS Host	IIS App Pool Identity	Service Principal Names
FarmReports.vmlab.local	vmlab\svcSQLRS	HTTP/FarmReports HTTP/ FarmReports.vmlab.local

In this example the following commands were executed:

```
SetSPN -S HTTP/FarmReportsvmlab\svcSQLRS

SetSPN -S HTTP/FarmReports.vmlab.localvmlab\svcSQLRS
```

Configure delegation

Kerberos delegation must be configured for SSRS to delegate the client's identity to back-end data source. In this example, SSRS queries data from a SQL Server transactional database by using the client's identity, therefore Kerberos delegation is required. Kerberos constrained delegation (KCD) is not a requirement in this scenario (because protocol transition is not needed), but KCD is configured as a best practice.

The SSRS service account that is running the SSRS services must be trusted to delegate credentials to each back-end service. In our example, the following delegation paths are needed:

Principal type	Principal name	Delegates to service

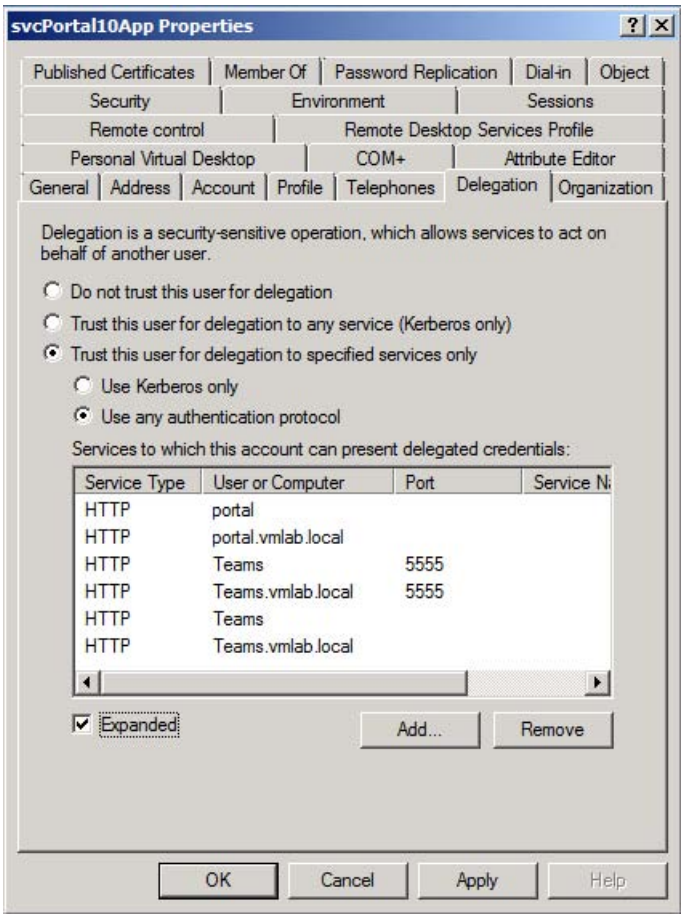
Principal type	Principal name	Delegates to service
User	Vmlab\svcPortal10App	HTTP/FarmReports HTTP/FarmReports.vmlab.local
User	Vmlab\svcSQLRS	MSSQLSVC/MySqlCluster.vmlab.local:1433

Optionally, if you wish to report against Analysis Services data sources, configure the following delegation paths:

Principal type	Principal name	Delegates to service
User	Vmlab\svcSQLRS	MSOLAPSvc.3/MySqlCluster.vmlab.local

To configure constrained delegation

1. Open the Active Directory Object's properties in Active Directory Users and Computers.
2. Navigate to the **Delegation** tab.



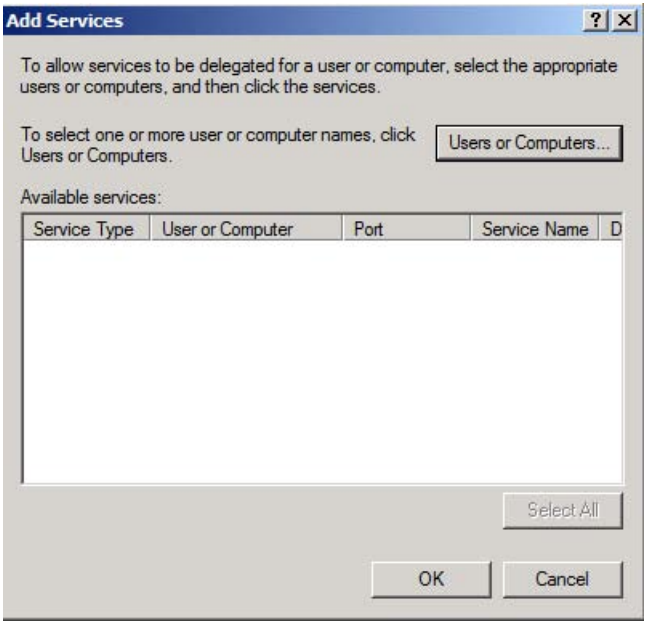
3. Select **Trust this user for delegation to specified services only**.

   Note:

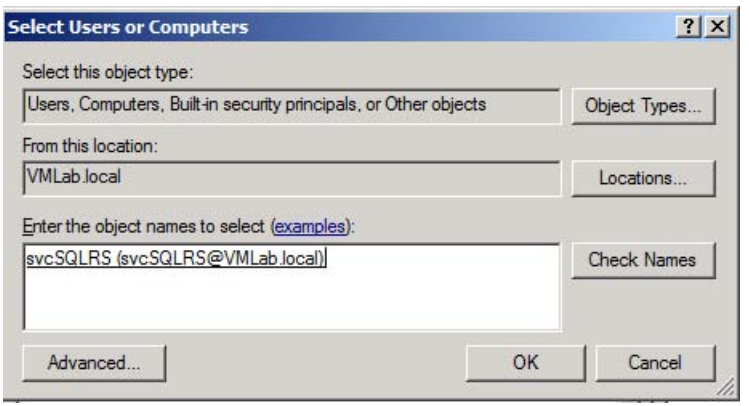
   For the SSRS service account, if you need to authenticate with data sources within the same forest but outside of the domain that the SSRS server resides in, configure basic delegation instead of constrained delegation. You can do this by selecting **Trust this computer for delegation to any service**. Remember that cross-forest Kerberos delegation is not possible.

4. Optionally select **Use any authentication protocol**. This enables protocol transition.

5. Click the **Add** button to select the service principal that can be delegate to.



6. Select **User and Computers**.

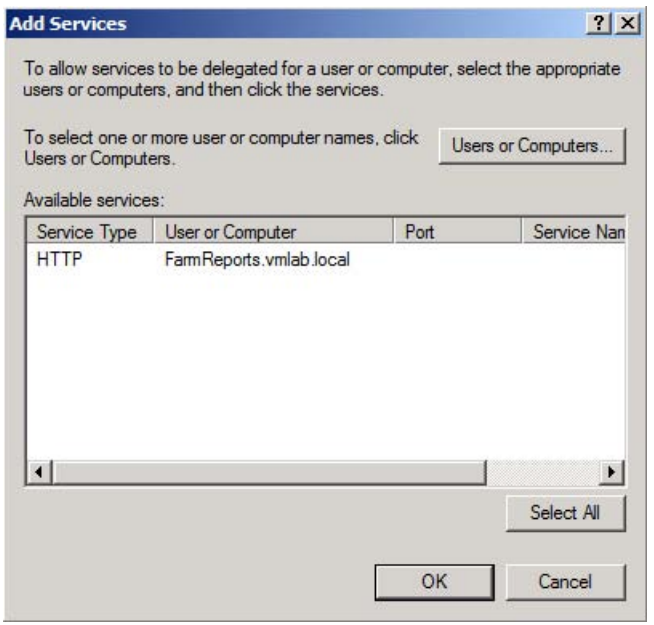


7. Select the service account that is running the service you want to delegate to. In this example, it is the service account for the SQL Server Reporting Service.

Note:

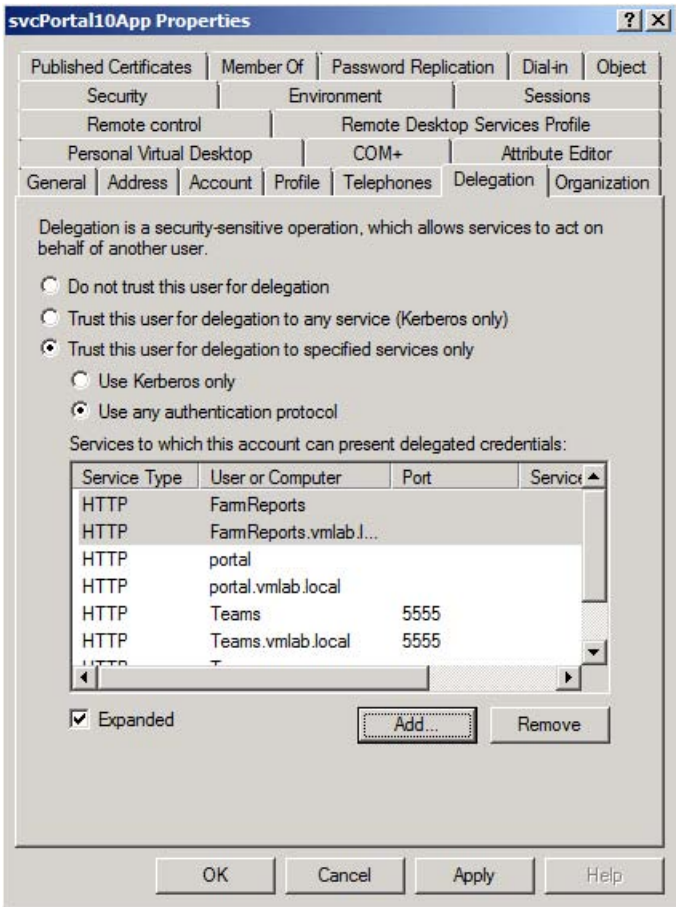
The service account selected must have an SPN applied to it. In our example, the SPN for this account (HTTP/FarmReports.vmlab.local) was configured earlier in the scenario.

8. Click **OK**. You are then asked to select the SPNs you want to delegate to on the following page.



9. Select the service or **Select All** and click **OK**.

You should now see the selected SPNs in the **services to which this account can present delegated credentials** list:



10. Repeat these steps for each delegation path identified earlier in this section. You have to configure delegation from the SQL Server Reporting Services service account to one or more back-end data sources (SQL OLTP or SQL AS in our scenarios).

Note:

For the SSRS service account, if you need to authenticate with data sources within the same forest but outside of the domain the SSRS server resides in, configure basic delegation instead of constrained delegation. To do so, select **Trust this computer for delegation to any service**. Remember that cross-forest Kerberos delegation is not possible.

Verify MSSQLSVC SPN for the service account running the service on SQL Server (performed in Scenario 2)

Verify that the SPN for the Analysis Services service account (vmlab\svcSQL) exists by using the following SetSPN command:

```
SetSPN -L vmlab\svcSQL
```

You should see the following:

```
MSSQLSVC/MySqlCluster

MSSQLSVC/MySqlCluster.vmlab.local:1433
```

Verify MSOLAPSvc.3 SPN for the Service Account running the SSAS service on the SQL Server Analysis Services server (performed in Scenario 3)

Verifythat the SPN for the SQL Server service account (vmlab\svcSQLAS) exists by using the following SetSPN command:

```
SetSPN -L vmlab\svcSQLAS
```

You should see the following:

```
MSOLAPSvc.3/MySqlCluster

MSOLAPSvc.3/MySqlCluster.vmlab.local
```

SQL Server Reporting Services

Install SharePoint Server 2010

SQL Server Reporting Services requires SharePoint Server 2010 to be installed on each SSRS server to run SSRS in SharePoint integrated mode. Install SharePoint Server 2010 on each reporting server and join each server to the SharePoint Server farm.

Install and configure SSRS in load-balanced, scaled out mode

Detailed step by step instructions on how to configure SQL Server Reporting Services in a load-balanced, scaled-out configuration is beyond the scope of this document. For detailed instructions on how to install SSRS, see Deployment Topologies for Reporting Services in SharePoint Integrated Mode. Once SSRS is installed, be sure to complete the additional SSRS configuration steps outlined below to complete the install.

Modify Web.config on the SSRS Servers

The following changes have to be made to the web.config files on each SSRS server. The web.config file can be found in the Program Files directory where SSRS is installed:

Add the <machineKey> element

SSRS servers in a load-balanced configuration need the same machine key set across all servers. The machine key element should be added as a child of the `<system.web>` element in web.config. Below is an example machine key:

```
<machineKey
validationKey="54AEBD3BC893726E9B84D30F4970CB58F2086C2DAEE2F8D34A65A0632F4676DDB
BC38779F2972C6596931E 13BD07A772BD4B9395BE38A43E461079E45D594E53"
decryptionKey="" validation="SHA1" decryption="AES" />
```

Important:

DO NOT USE THE SAMPLE MACHINE KEY IN OUR ENVIRONMENT. Generate your own key values for your environment.

Modify ReportingServer.config

The following changes have to be made to the ReportingServer.config files on each SSRS server. The ReportingServer.config file can be found in the program files directory where SSRS is installed:

Enable Kerberos authentication

To enable Kerberos authentication, set the authentication type to "RSWindowsNegotiate". Change the `<AuthenticationTypes/>`

element and add `<RSWindowsNegotiate/>`

```
<AuthenticationTypes><RSWindowsNegotiate/></AuthenticationTypes>
```

Modify the URL root

Add the URL for the report server to the `<UrlRoot>` tag found in the `<service>` tag of ReportingServer.Config

```
<UrlRoot>http://FarmReports/reportserver</UrlRoot>
```

Configure BackConnectionHostNames in the registry

To allow SQL Server Reporting Services to authenticate with each other on a single computer, NTLM loopback detection needs to be addressed. Instead of disabling loopback detection, a better practice is to configure the BackConnectionHostNames value in the registry of each SSRS server. For more information about

BackConnectionHostNames, see You receive an error message when you use SQL Server 2008 Reporting Services.

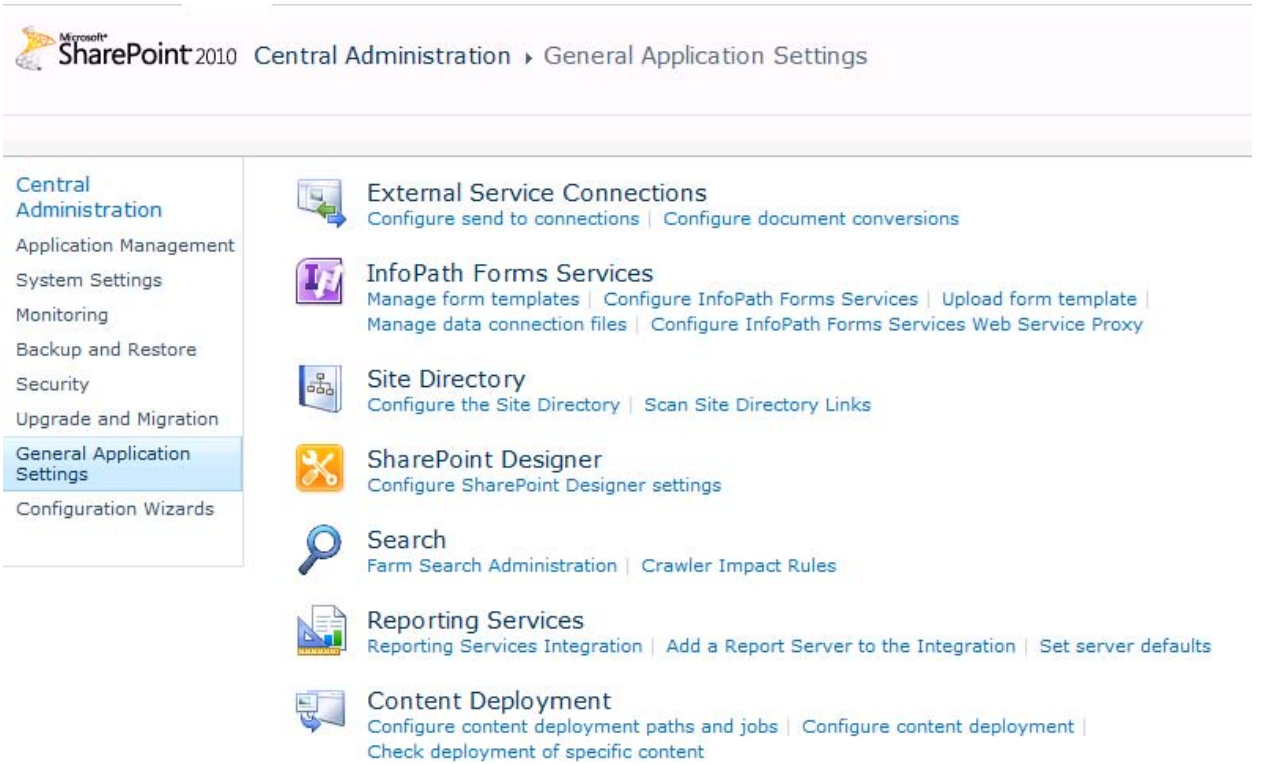
In our example, we configure the following values for BackConnectionHostNames:

- FarmReports
- FarmReports.vmlab.local

Once the BackConnectionHostNames values are set, reboot the SSRS server.

Configure SharePoint Server

In Central Administration, you find the farm configuration options for SSRS. Note that in SharePoint Server 2010 you do not need to install a separate SSRS component installation for SSRS administration and Web Parts. To access the SSRS farm options, navigate to Central Administration and then see **Reporting Services** in the **General Application Settings** section.



Grant the Reporting Services service account permissions on the web application content database

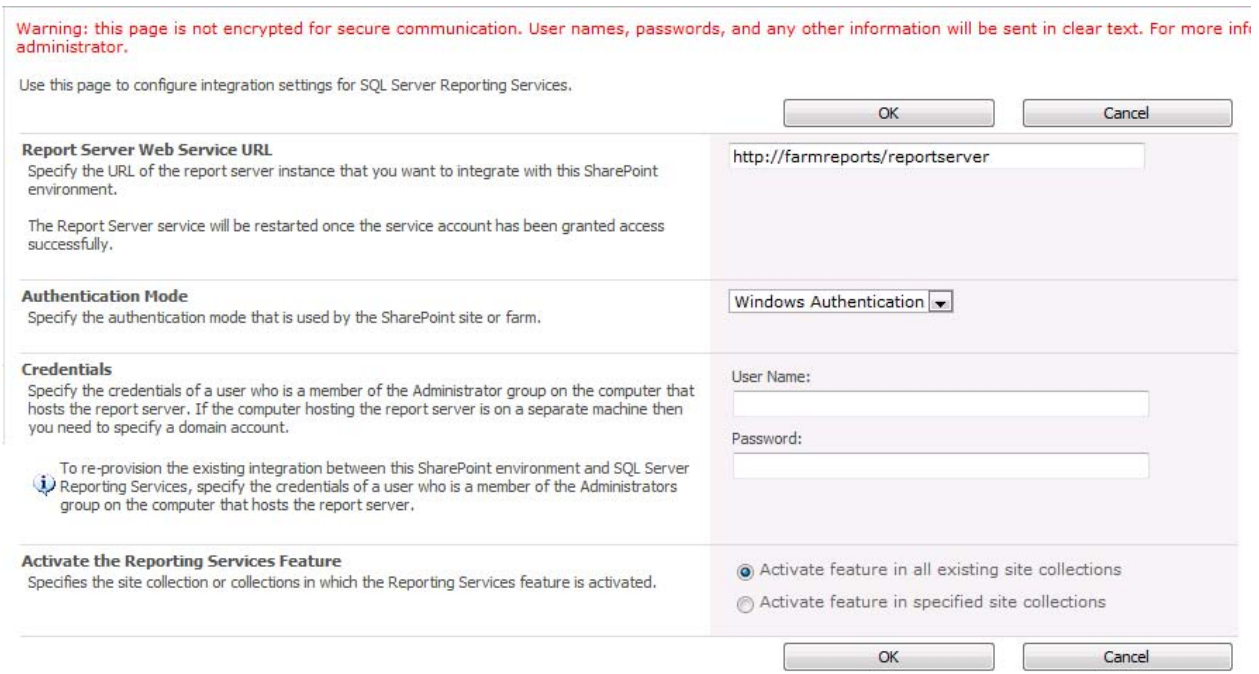
A required step in configuring SQL Server Reporting Services in SharePoint integrated mode is allowing the Reporting Services service account access to the content databases for web applications hosting reports. In this example, we grant the Reporting Services account access to the "portal" web application's content database through Windows PowerShell.

Run the following command from the SharePoint 2010 Management Shell:

```
$w = Get-SPWebApplication -Identity http://portal

$w.GrantAccessToProcessIdentity("vmlab\svcSQLRS")
```

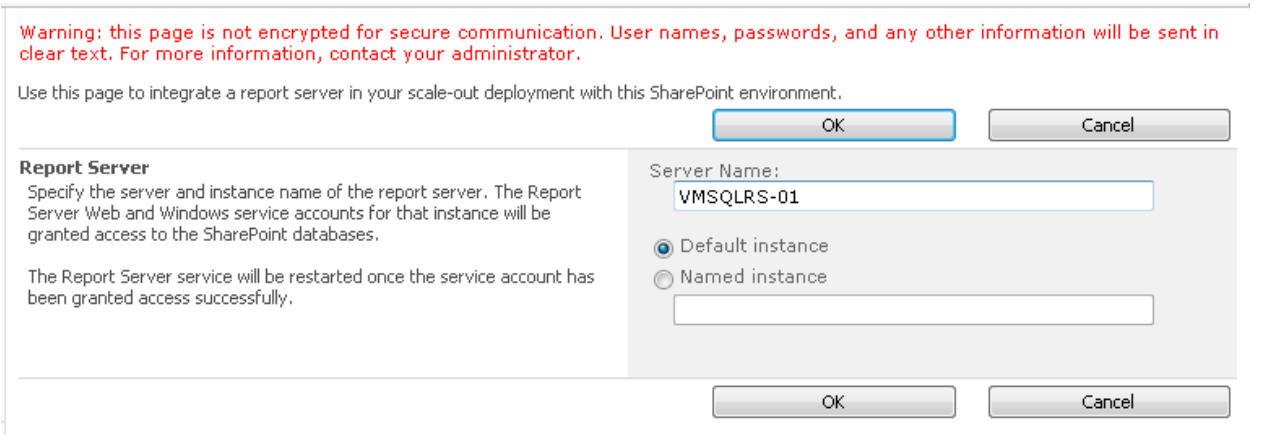
Configure Reporting Services Integration

In the **Reporting Service Integration** dialog box, specify the load-balanced URL of the report server. Also, select the **Activate feature in all exiting collections** option to automatically activate the Reporting Services feature in your site collections.



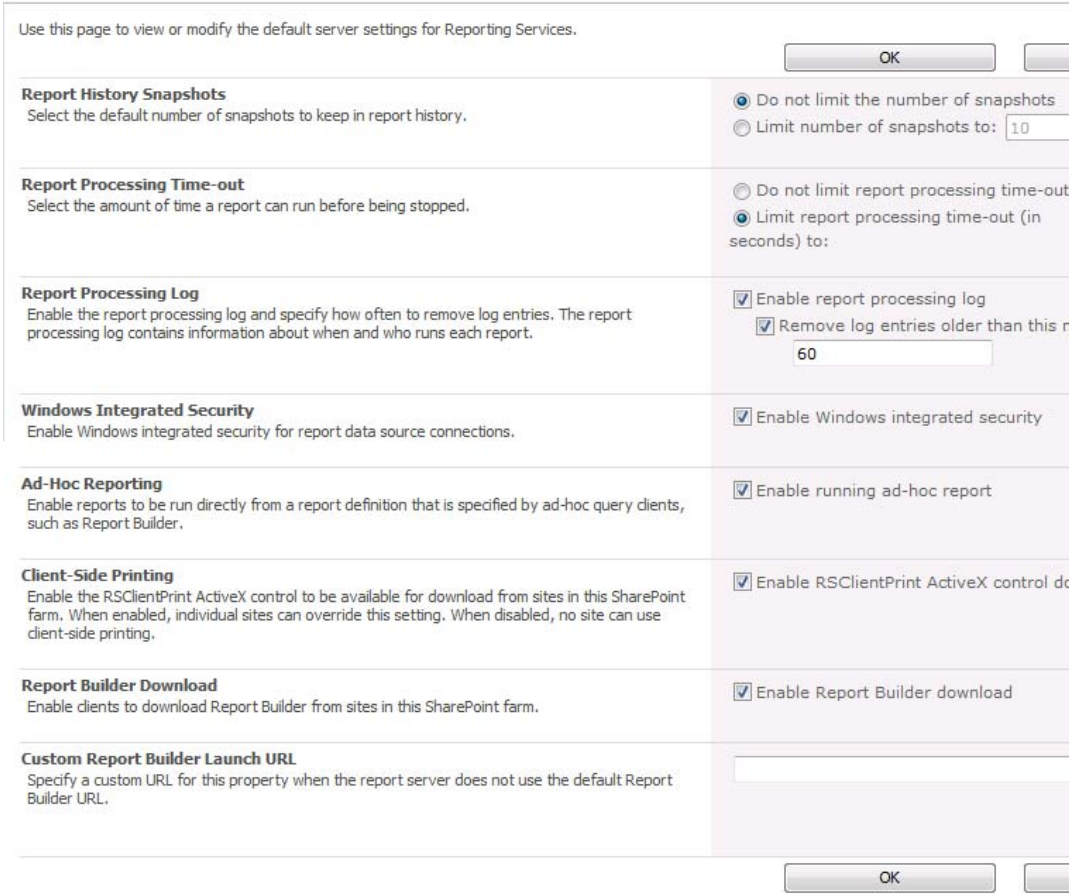
Add each report server to the integration

In the **Add a report server to the integration** dialog box, specify each of the nodes of the Reporting Services NLB group. You have to open this dialog box for each server that you are adding to the integration; there is no way to add multiple servers in a single operation.



Set server defaults

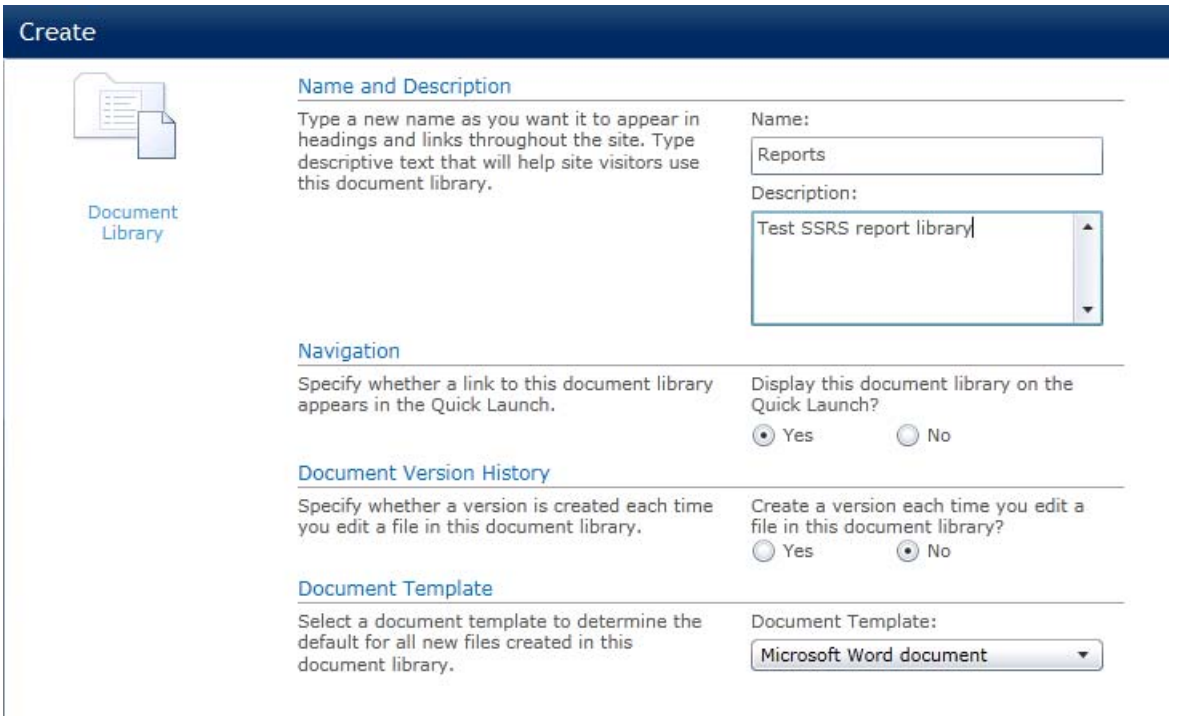
At this point SSRS integration should be configured. To validate the configuration, open the Server Defaults page. No changes are required for the example in this document.



Verify configuration

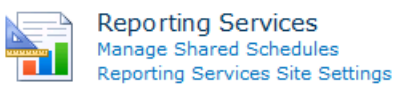
Create a document library for reports

Create a document library to host SSRS reports in your SharePoint site. In this example, we assume the existence of a document library called "reports" at http://portal/reports.



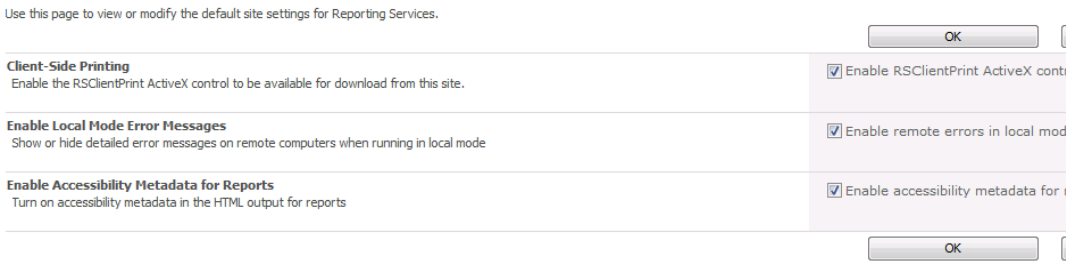
Validate site collection settings for Reporting Services

In the browser, navigate to the Site Settings of the site that is hosting the document library for SSRS reports. In Site Settings you should see a new category called **Reporting Services**.



If you do not see the Reporting Services feature in the site collections features list, you may need to activate it from Central Administration. For more information, see How to: Activate the Report Server Feature in SharePoint Central Administration (http://go.microsoft.com/fwlink/?LinkId=196878).

Click the Reporting Services site settings link to ensure the settings are accessible.



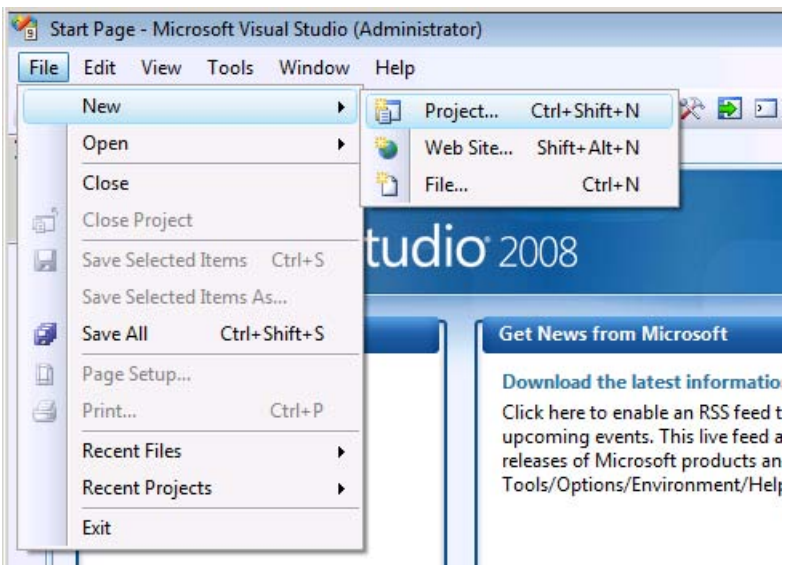
Note:

No changes to Reporting Services Site Settings are required for this demonstration.

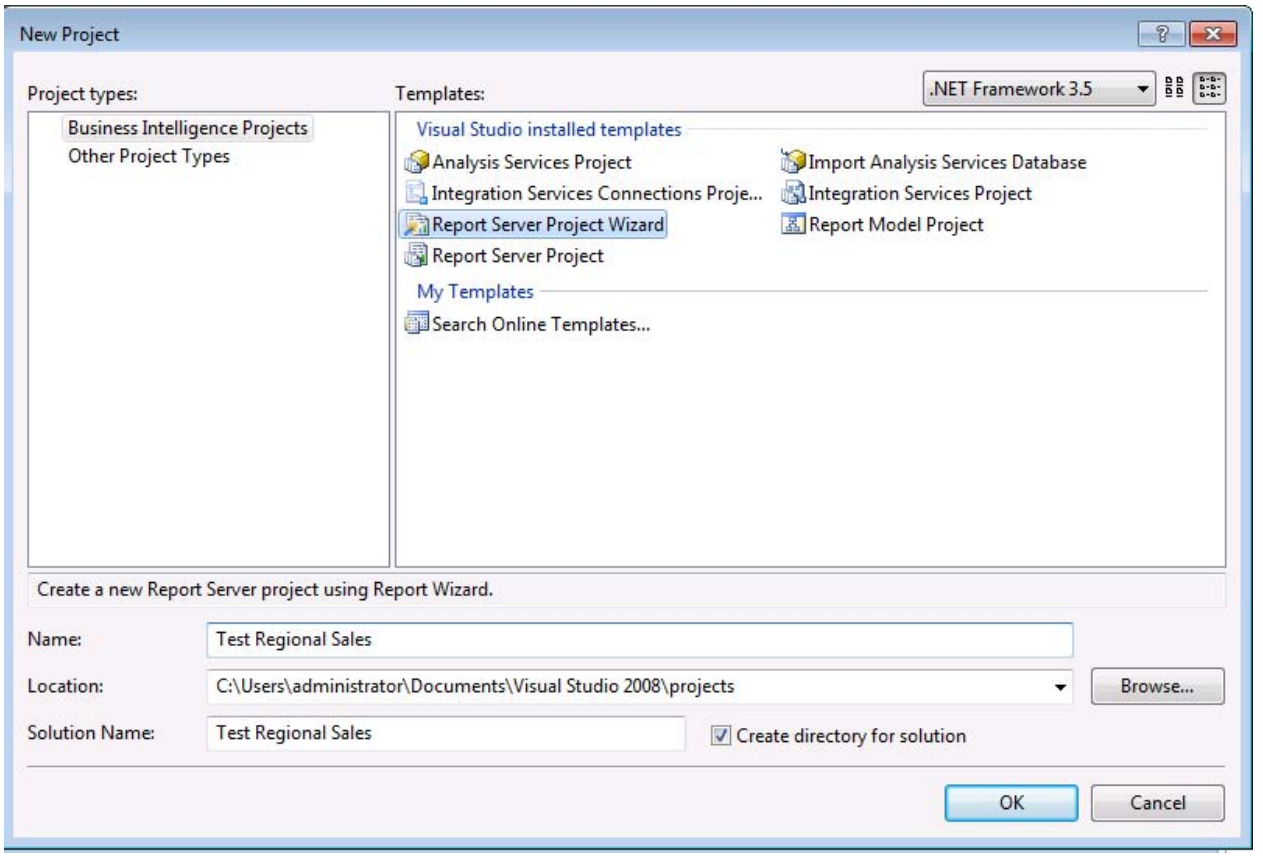
Create and publish a test report in SQL Server Business Intelligence Development Studio

After you configure SSRS and the integration with SharePoint Server, you create a test report to ensure identity delegation is working correctly.

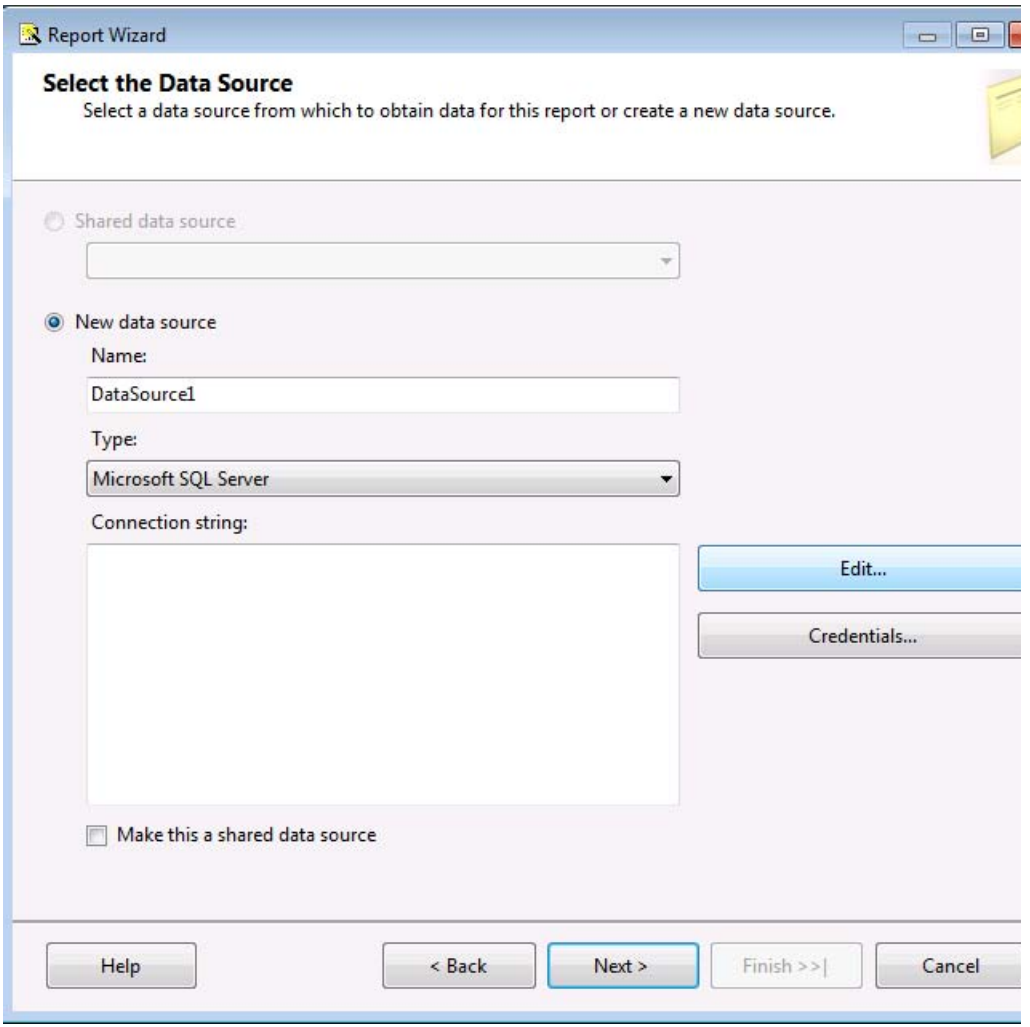
1. Open SQL Server Business Intelligence Development Studio. Click **File**, point to **New**, and then click **Project**.



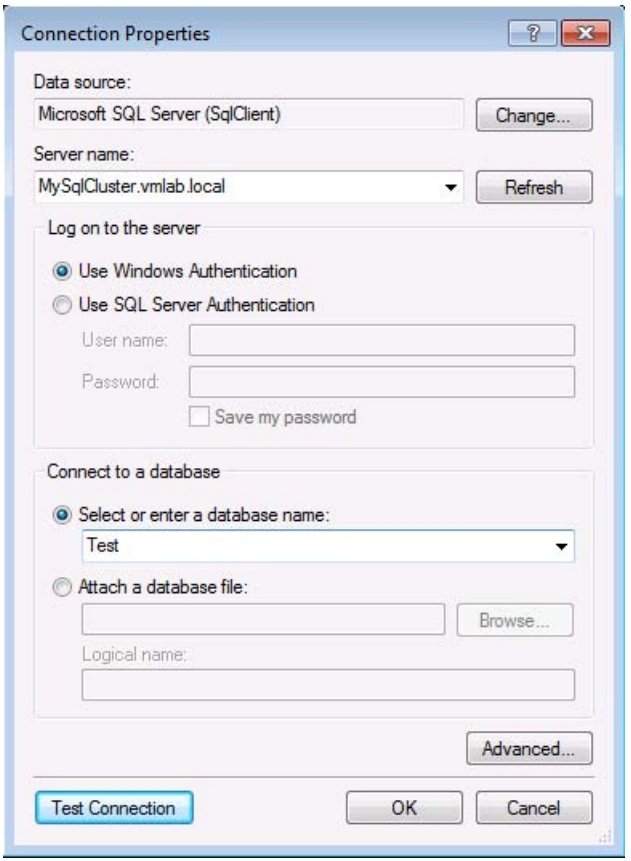
2. Select **Report Server Project Wizard** and enter a project name.



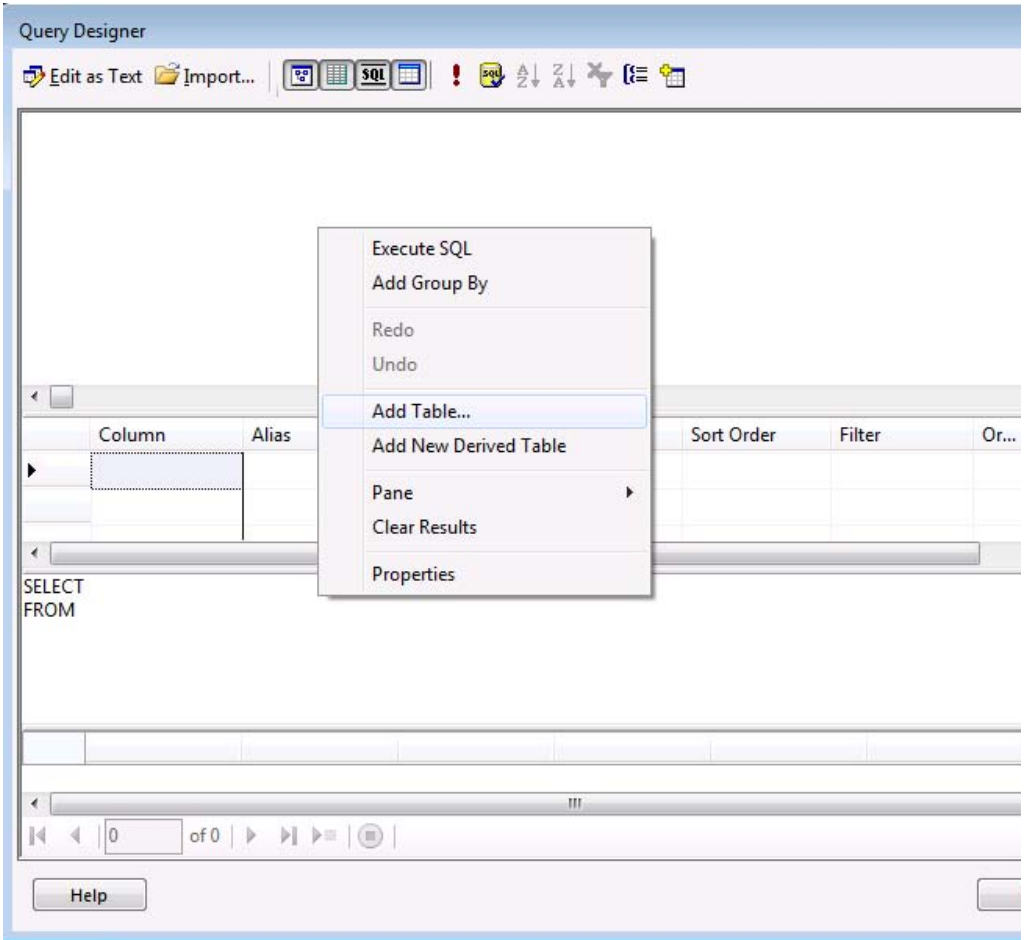
3. Next configure a new data source. Choose the type **Microsoft SQL Server** and click the **Edit** button.



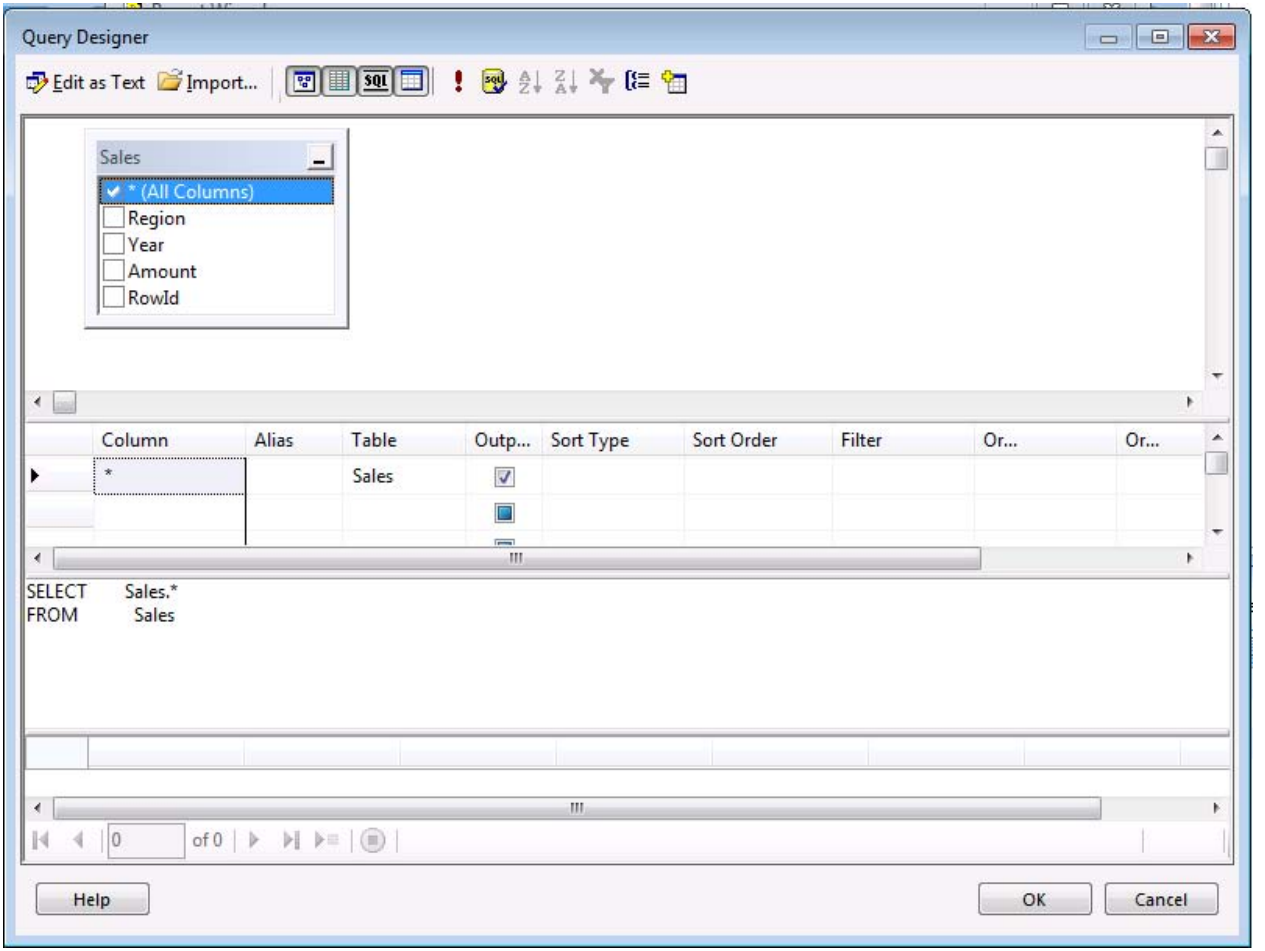
4. In **Connection Properties** enter the information to connect to the demo SQL Server cluster created in scenario 2.



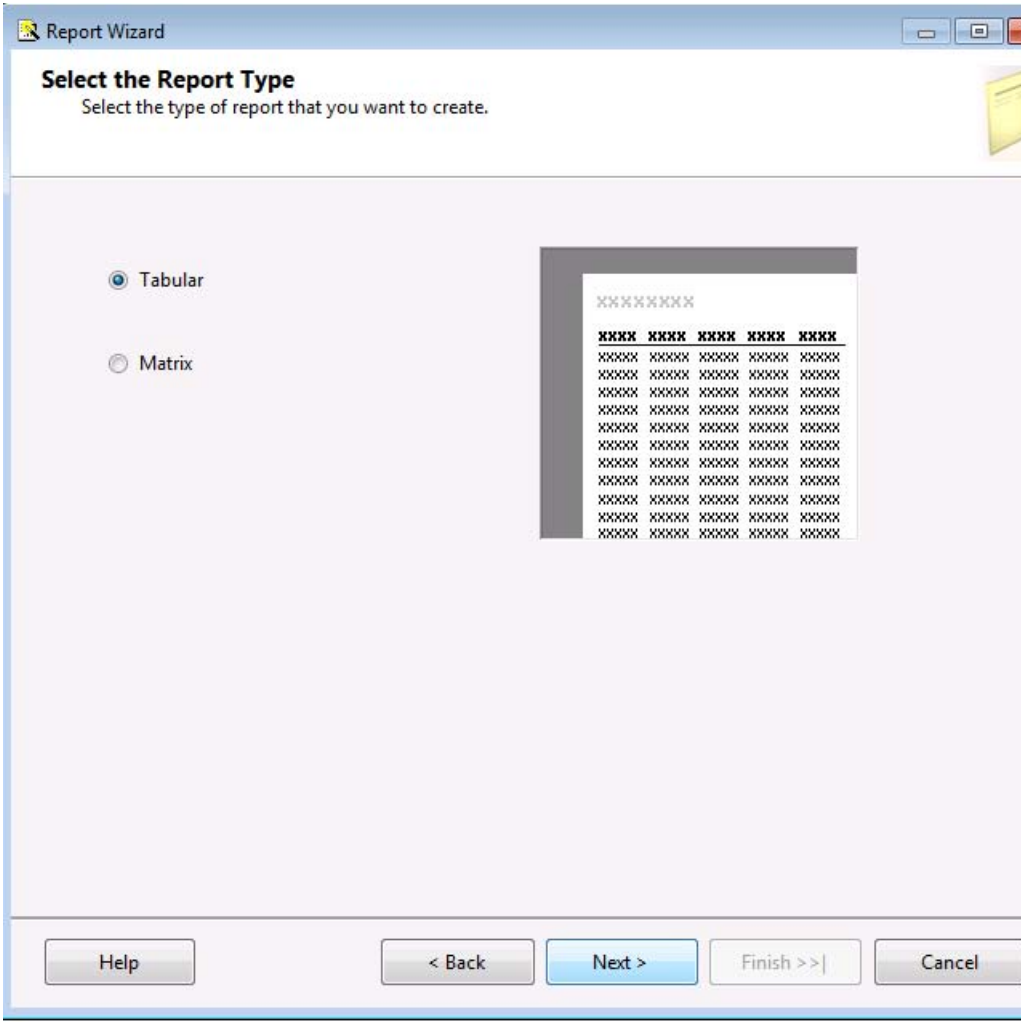
5. Open query designer, right-click the query window and select **Add table.**



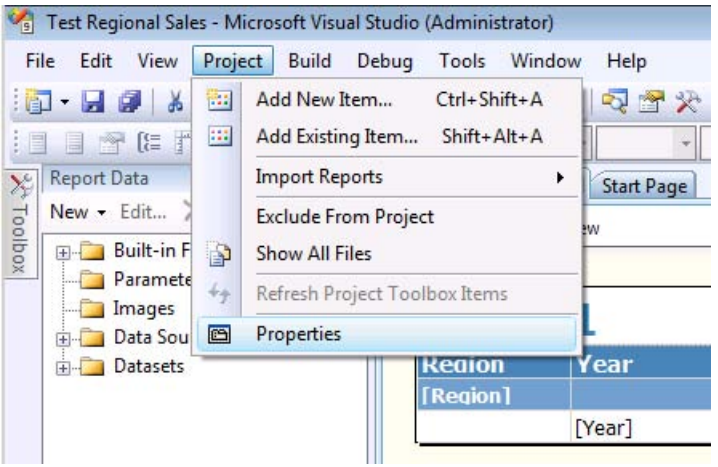
6. Choose the **Sales** table (created in scenario 2) and select **All Columns**.



7. Select a tabular report type.



8. In our example we group by region; you can skip this step if you want to.
9. Once the project is created, open the project properties on the **Project** menu.



10. Configure the following project properties:

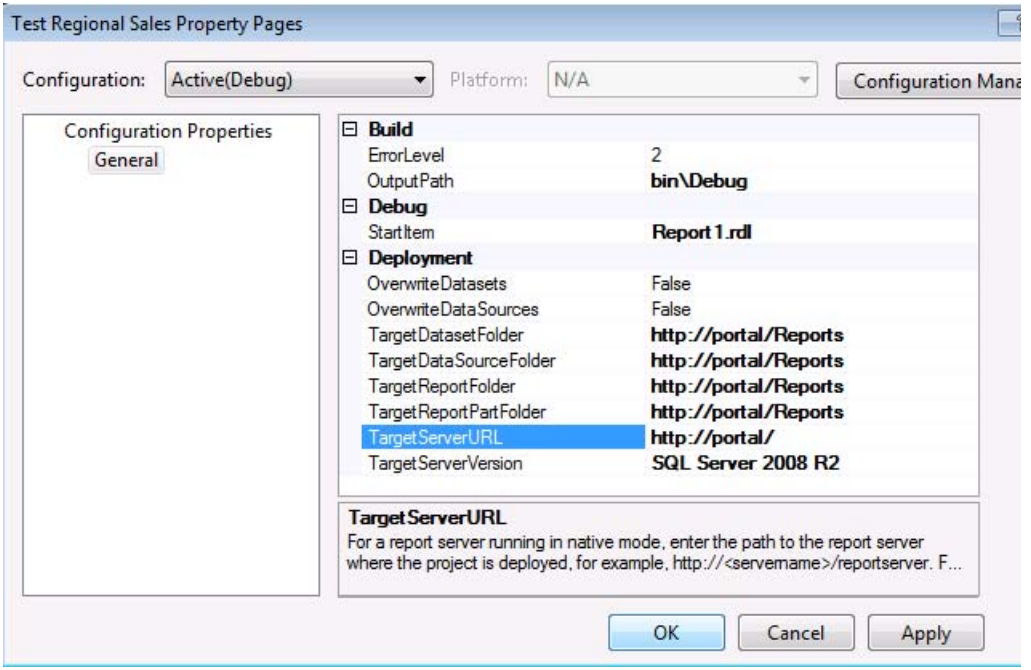
    a) TargetDatasetFolder — Set it to the test report folder created earlier

    b) TargetDatasetFolder — Set it to the test report folder created earlier

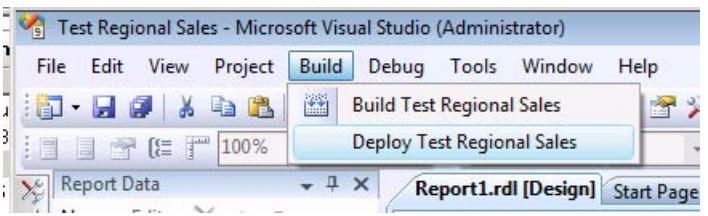
    c) TargetReportFolder — Set it to the test report folder created earlier

    d) TargetReportPartFolder — Set it to the test report folder created earlier

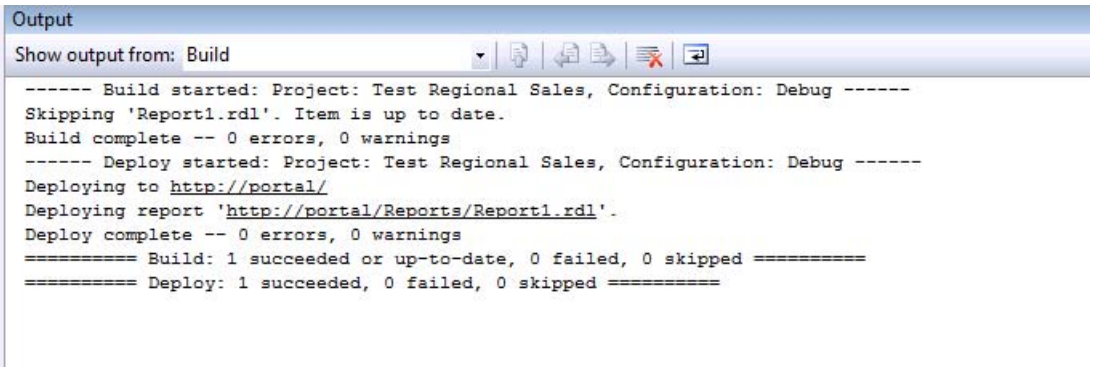
    e) TargetServerURL — Set to the web application URL that is hosting the report



11. Deploy the report to the SharePoint library. On the build menu select **Deploy <project name>**".

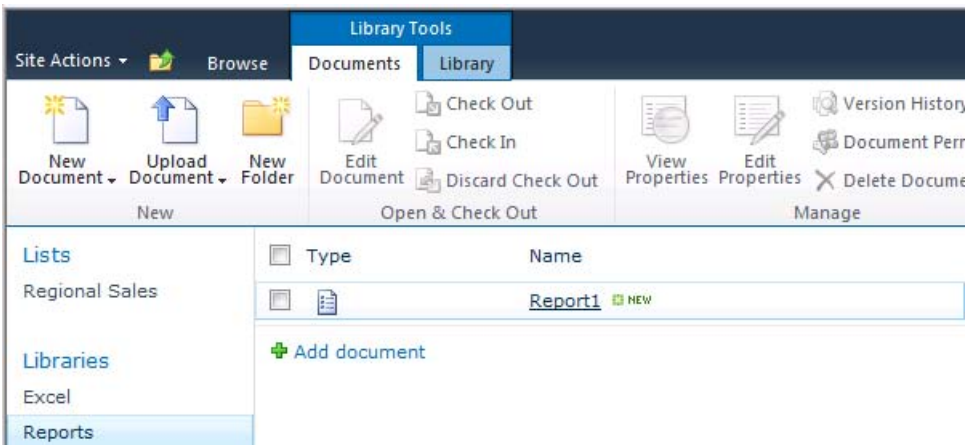


12. If it is successful, you will see the deployment succeeded message in the **Output** window.

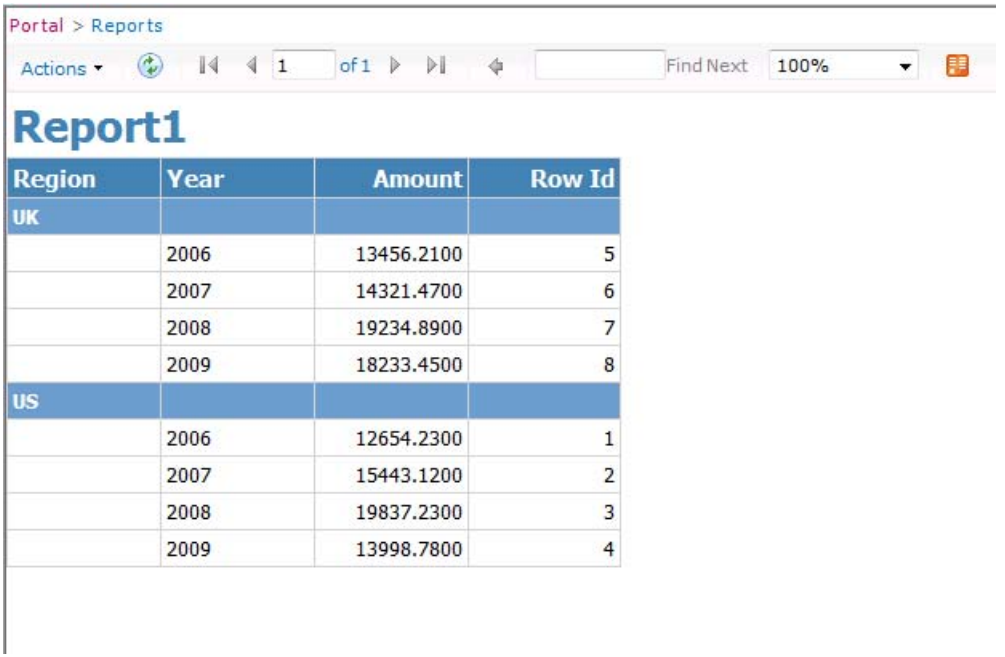


View the test report in Internet Explorer

Open the report document library created in previous steps of this scenario in the browser. You should see the report file you just published. If you do not see the report, you may need to activate the Reporting Services features in your site collection. For more information, seeHow to: Activate the Report Server Feature in SharePoint Central Administration (http://go.microsoft.com/fwlink/?LinkID=196878).



Click the report and it will render in the browser.



To further verify delegation and the data connection, changed the source data in SQL Server Management Studio and refresh the SSRS report data connection in the browser. You should see the data changes reflected in the report.

SSL configuration for Reporting Services

In some environments it may be required to protect communications between front-end Web and SSRS servers with SSL. A detailed walkthrough of how to configure SSL for Reporting Services is out of scope for this paper, but at a high level these are the steps you have to take:

1. Configure each reporting server for SSL. See Configuring a Report Server for Secure Sockets Layer (SSL) Connections (http://go.microsoft.com/fwlink/?LinkId=196881).

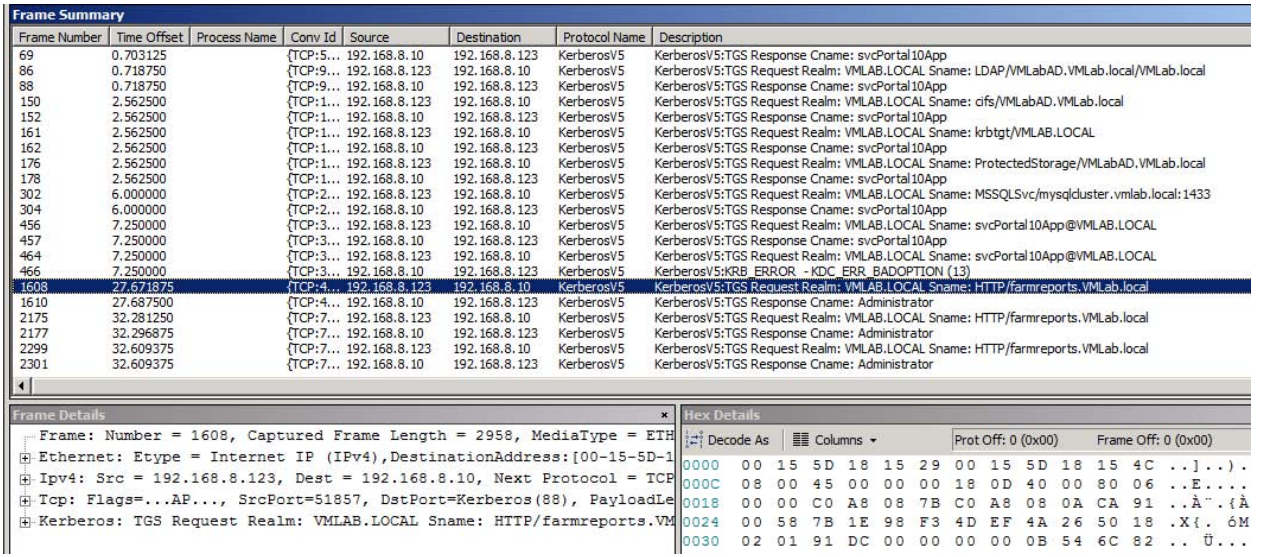
2. Update ReportingServer.config. Change the `<UrlRoot>` to the new https:// URL.

3. Restart the SQL Server Reporting Services service.

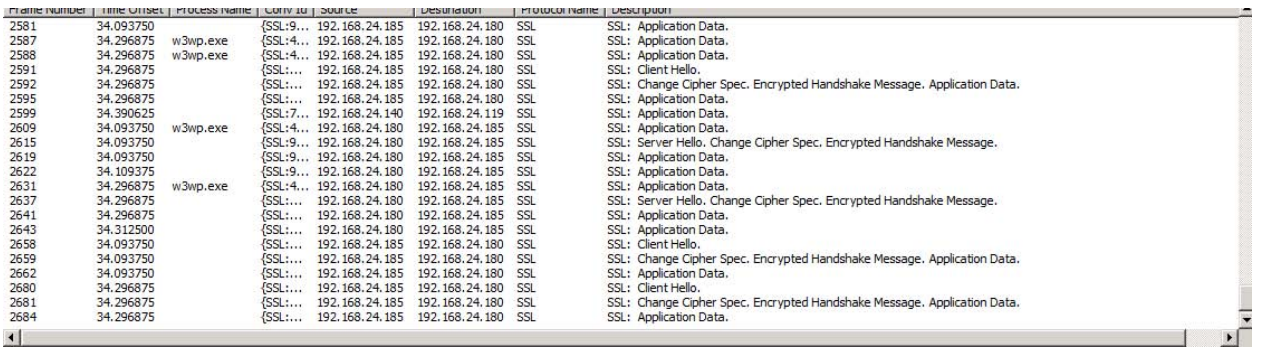
4. In Central Administration, change the Reporting Services integration settings and change the Report Server Web Service URL to the new https:// URL.

5. Restart IIS on each instance of SharePoint Server that is running the web application service.

You do not need to change any of the SPNs created when configuring Reporting Services with HTTP in the previous steps. The SPN for an HTTP service over SSL remains HTTP/<service>. You can see this by usingNetMonto view the front-end web server that is communicating with the Reporting Services Server.



Notice the ticket granting service request highlighted and the Sname requested. The reporting server service was accessed using https:// and the SName in the ticket request remained HTTP/ as expected. To ensure the WFE was actually using SSL to communicate with the reporting server, additional traffic was captured and analyzed:



Notice that all requests from the WFE to the reporting server are protected over SSL. This confirms SSL was used for communications between the web front ends and the reporting server.

Identity delegation for Excel Services (SharePoint Server 2010)

Published: December 2, 2010

In this scenario you add the Excel Services service application to the SharePoint Server environment and configure Kerberos constrained delegation to allow the service to refresh data in a worksheet from an external SQL Server data source.

Note:

If you are installing on Windows Server 2008, you may have to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

Scenario dependencies

To complete this scenario you need to have completed the following articles:

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP

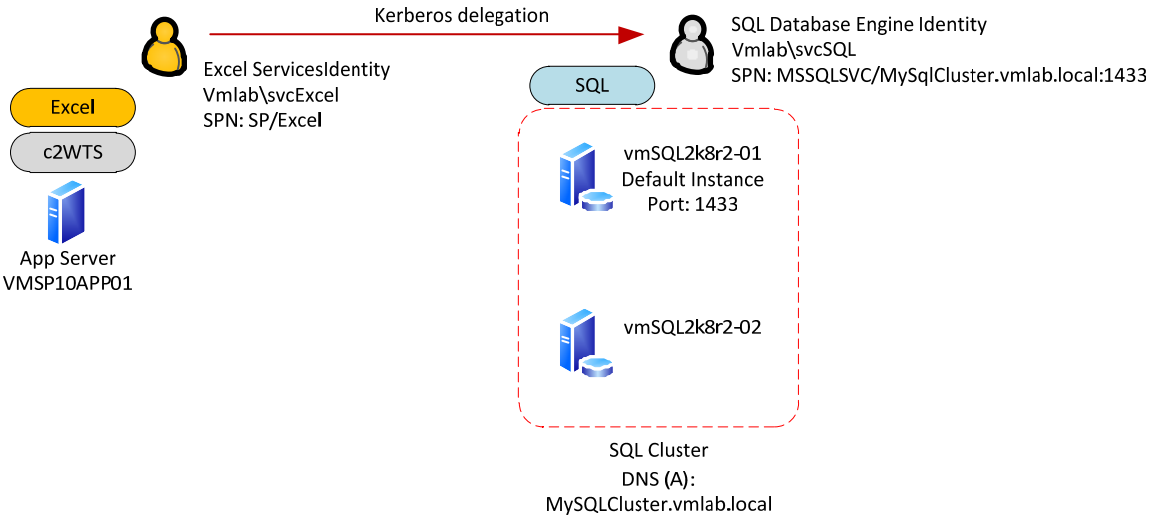
Configuration checklist

Area of Configuration	Description
Active Directory Configuration	Create Excel Services service account Configure SPN on Excel Services service account

Area of Configuration	Description
	Configure Kerberos constrained delegation for servers running Excel Services Configure Kerberos constrained delegation for the Excel Services service account
SharePoint Server configuration	Start Claims to Windows Token Service on Excel Services Servers Start the Excel Services service instance on the Excel Services server Create the Excel Services service application and proxy Configure Excel services trusted file location and authentication settings
Verify Excel Service Constrained Delegation	Create document library to host test workbook Create test SQL database and test table Create test Excel workbook with SQL data connection Publish workbook to SharePoint Server and refresh data connection

Scenario environment details

Kerberos constrained delegation paths

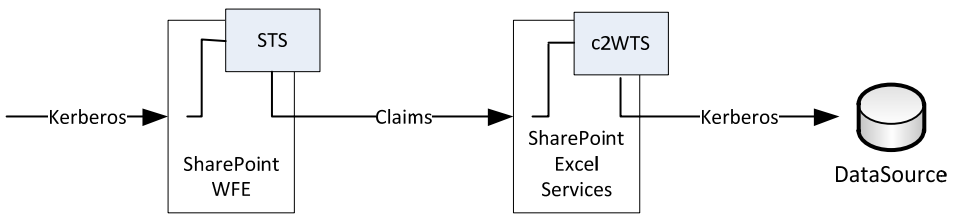


In this scenario we will configure the SharePoint Server Excel Services service account for Kerberos constrained delegation to the SQL Server service.

Note:

In this scenario we will configure the Claims to Windows Token Services (C2WTS) to use a dedicated service account. If you leave the C2WTS configured to use **Local System** you will need to configured constrained delegation on the computer account for the computer running the C2WTS and Excel Services.

SharePoint Server logical authentication



Authentication in this scenario begins with the client authenticating with Kerberos authentication at the web front end. SharePoint Server 2010 will convert the Windows authentication token into a claims token using the local Security Token Service (STS). The excel service application will accept the claims token and convert it into a windows token (Kerberos) using the local Claims to Windows Token Service (C2WTS) that is a part of Windows Identity Framework (WIF). The excel service application will then use the client's Kerberos ticket to authenticate with the backend DataSource.

Step-by-step configuration instructions

Active Directory configuration

Create Excel Services service account

As a best practice Excel Services should run under its own domain identity. To configure the Excel Service Application an Active Directory accounts must be created. In this example the following accounts were created:

SharePoint Server Service	IIS App Pool Identity
Excel Services	vmlab\svcExcel

Configure SPN on the Excel Services service account

Kerberos constrained delegation must be configured if Excel Services is going to delegate the client's identity to a back end data source. In this example Excel services will query data from a SQL transactional database, therefore Kerberos delegation is required.

The Active Directory Users and Computers MMC snap-in is typically used to configure Kerberos delegation. To configure the delegation settings within the snap-in, the Active Directory object being configured must have a service principal name applied; otherwise the **delegation** tab for the object will not be visible in the object's properties dialog. Although Excel Services does not require a SPN to function, we will configure one for this purpose.

On the command line, run the following command:

```
SETSPN -S SP/ExcelServices
```

Note:

The SPN is not a valid SPN. It is applied to the specified service account to reveal the delegation options in the AD users and computers add-in. There are other supported ways of specifying the delegation settings (specifically the msDS-AllowedToDelegateTo AD attribute) but this topic will not be covered in this document.

Configure Kerberos constrained delegation for Excel Services

To allow excel services to delegate the clients identity Kerberos constrained delegation must be configured. It is required to configure constrained delegation with protocol transition for the conversion of claims token to windows token via the WIF C2WTS.

Each server running excel services must be trusted to delegate credentials to each back-end service excel will authenticate with. In additional, the excel services service account must also be configured to allow delegation to the same back-end services.

In our example the following delegation paths are defined:

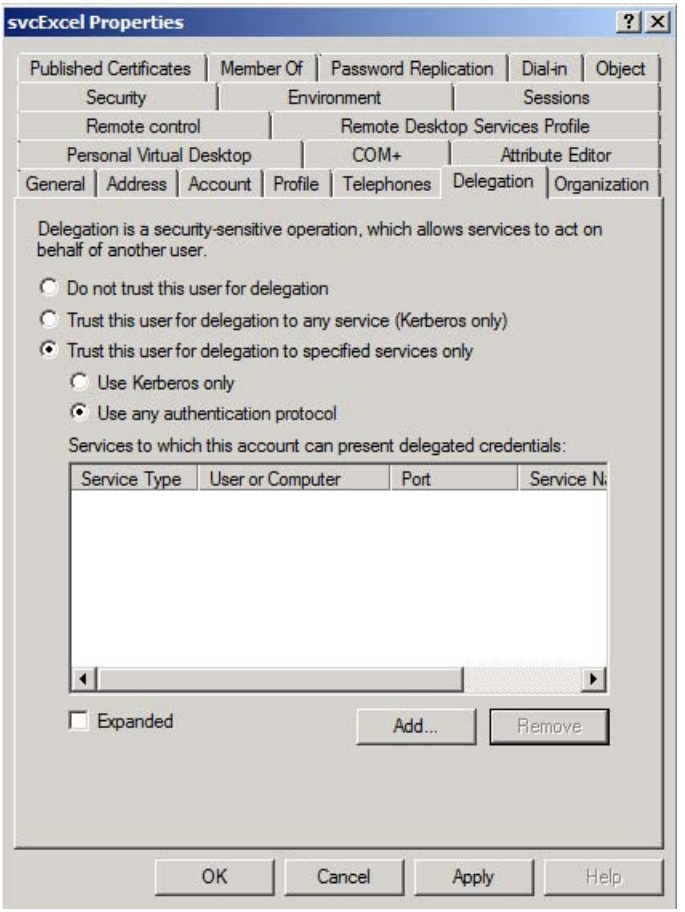
Principal Type	Principal Name	Delegates To Service
User	svcExcel	MSSQLSVC/MySqlCluster.vmlab.local:1433
*User	svcC2WTS	MSSQLSVC/MySqlCluster.vmlab.local:1433
**Computer	VMSP10APP01	MSSQLSVC/MySqlCluster.vmlab.local:1433

* Configured later in this scenario

** Only required if the C2WTS is running as local system

To configure constrained delegation

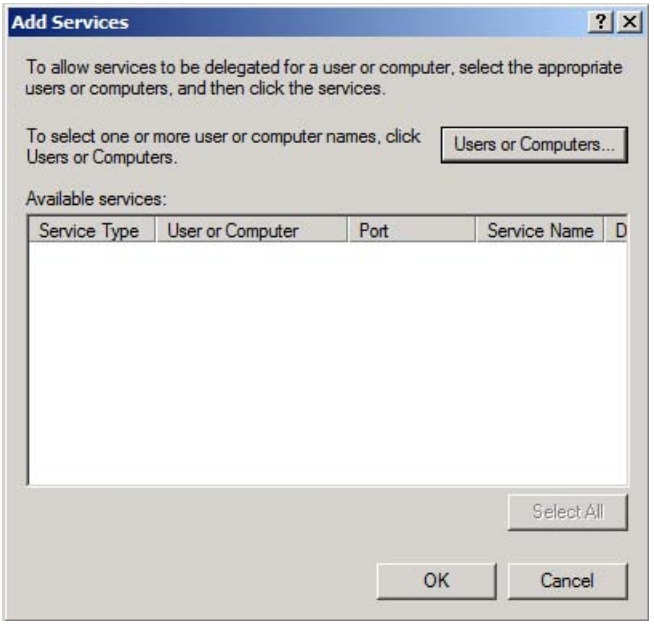
1. Open the Active Directory Object's properties in Active Directory Users and Computers.
2. Navigate to the **Delegation** tab.



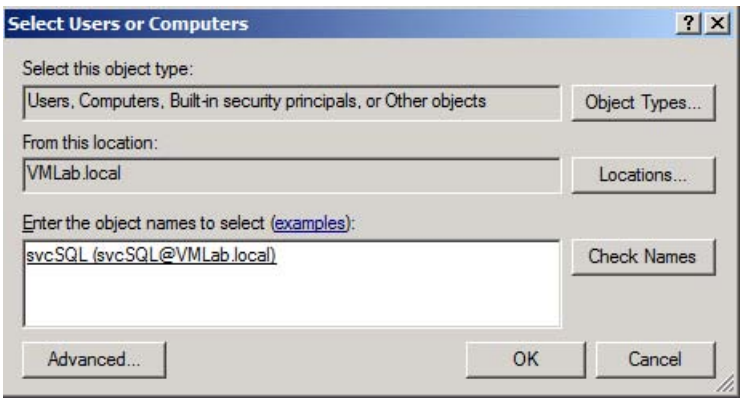
3. Select **Trust this user for delegation to specified services only**.

4. Select **Use any authentication protocol**. This enables protocol transition and is required for the service account to use the C2WTS.

5. Click the add button to select the service principal allowed to delegate to.



6. Select **User and Computers**.

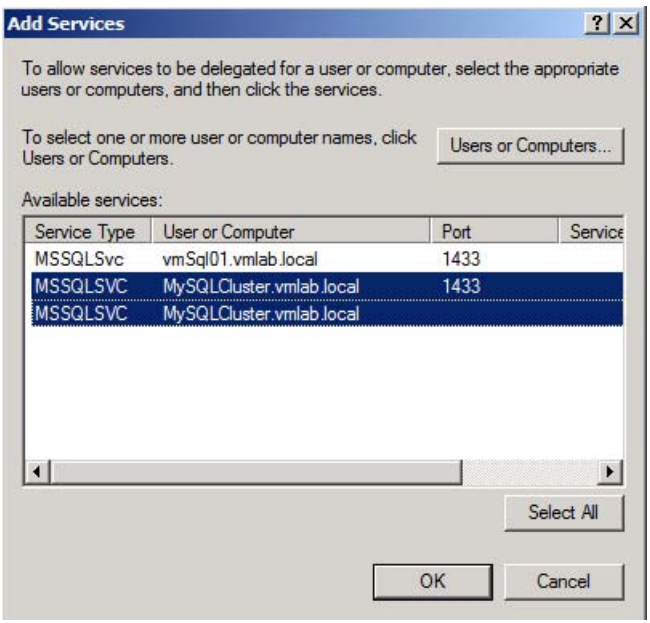


7. Select the service account running the service you wish to delegate to. In this example it is the service account for the SQL service.

Note:

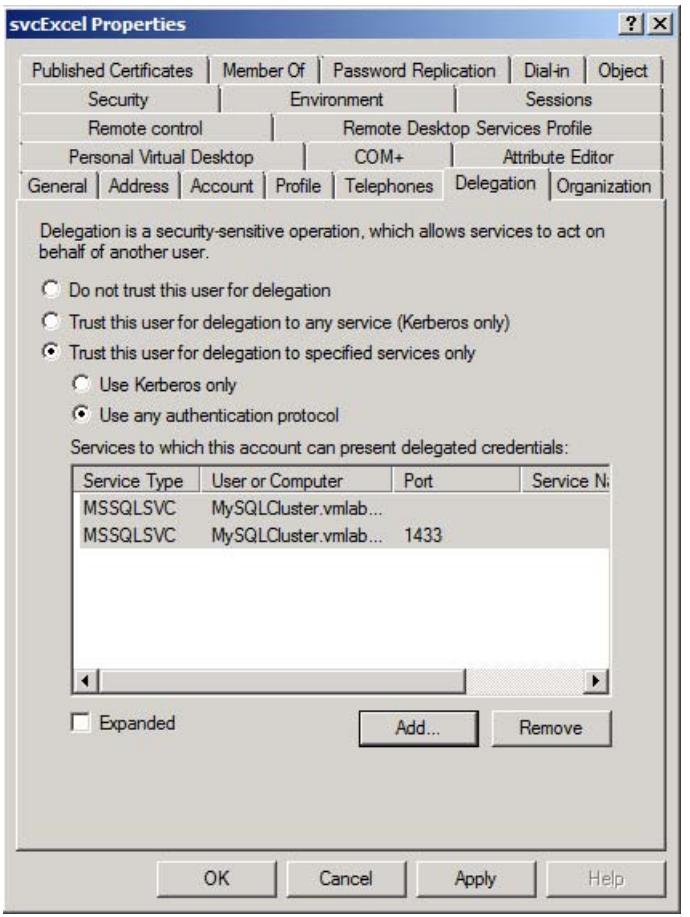
The service account selected must have a SPN applied to it. In our example the SPN for this account was configured in a previous scenario.

8. Click **OK**. You will then be asked to select the SPNs you would like to delegate to in the following window.



9. Select the services for the SQL cluster and click OK.

10. You should now see the selected SPNS in the **services to which this account can presented delegated credentials** list.



11. Repeat these steps for each delegation path defined in the beginning of this section.

Verify MSSQLSVC SPN for the Service Account running the service on the SQL Server (performed in Scenario 2)

Verify the SPN for Analysis Services service account (vmlab\svcSQL) exists with the following SetSPN command:

```
SetSPN -L vmlab\svcSQL
```

You should see the following:

```
MSSQLSVC/MySqlCluster

MSSQLSVC/MySqlCluster.vmlab.local:1433
```

SharePoint Server configuration

Configure and Start the Claims to Windows Token Service on Excel Services Servers

The Claims to Windows Token Service (C2WTS) is a component of the Windows Identity Foundation (WIF) which is responsible for converting user claim tokens to windows tokens. Excel services uses the C2WTS to convert the user's claims token into a windows token when the services needs to delegate credentials to a back-end system which uses Integrated Windows authentication. WIF is deployed with SharePoint Server 2010 and the C2WTS can be started from Central Administration.

Each Excel Services Application server must run the C2WTS locally. The C2WTS does not open any ports and cannot be accessed by a remote caller. Further, the C2WTS service configuration file must be configured to specifically trust the local calling client identity.

As a best practice you should run the C2WTS using a dedicated service account and not as Local System (the default configuration). The C2WTS service account requires special local permissions on each server the service runs on so be sure to configure these permissions each time the service is started on a server. Optimally you should configure the service account's permissions on the local server before starting the C2WTS, but if done after the fact you can restart the C2WTS from the Windows services management console (services.msc).

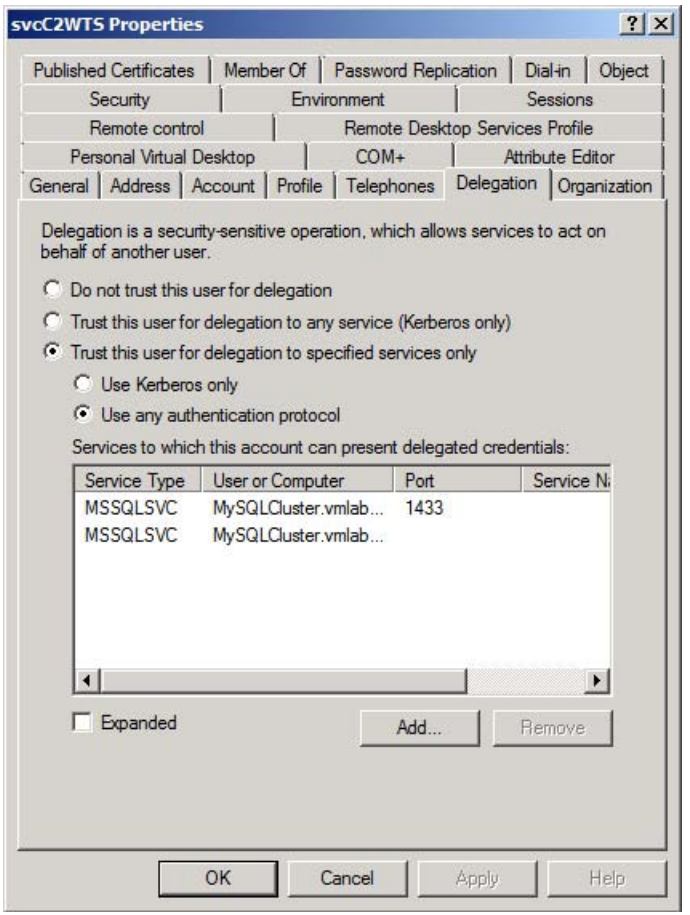
To start the C2WTS

1. Create a service account in Active Directory to run the service under. In this example we created vmlab\svcC2WTS.

2. Add an arbitrary Service Principal Name (SPN) to the service account to expose the delegation options for this account in Active Directory Users and Computers. The SPN can be any format because we do not authenticate to the C2WTS using Kerberos authentication. It is recommended to not use an HTTP SPN to avoid potentially creating duplicate SPNs in your environment. In our example we registered SP/C2WTS to the vmlab\svcC2WTS using the following command:

   ```
   SetSPN -S SP/C2WTS vmlab\svcC2WTS
   ```

3. Configure Kerberos constrained delegation on the C2WTS services account. In this scenario we will delegate credentials to the SQL service running with the MSSQLSVC/MySqlCluster.vmlab.local:1433 service principal name.



4. Next, configure the required local server permissions that the C2WTS requires. You will need to configure these permissions on each server the C2WTS runs on. In our example this is VMSP10APP01. Log onto the server and give the C2WTS the following permissions:

    a) Add the service account to the local Administrators Groups.

    b) In local security policy (secpol.msc) under user rights assignment give the service account the following permissions:

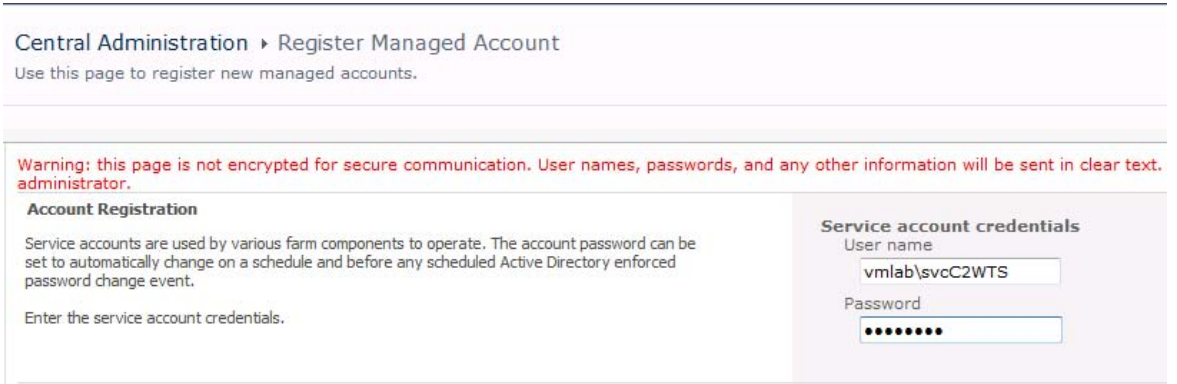
        i. Act as part of the operating system

        ii. Impersonate a client after authentication

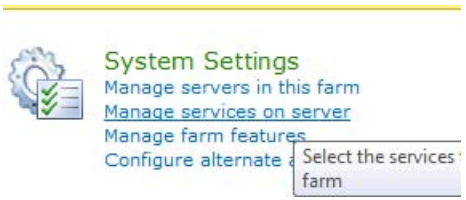
        iii. Log on as a service

5. Open Central Administration.

6. Under Security->Configure Managed Service Accounts, Register the C2WTS service account as a managed account.

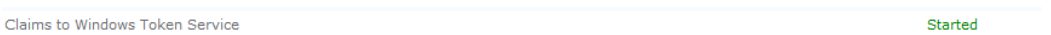


7. Under services, select **Manage services on server**.

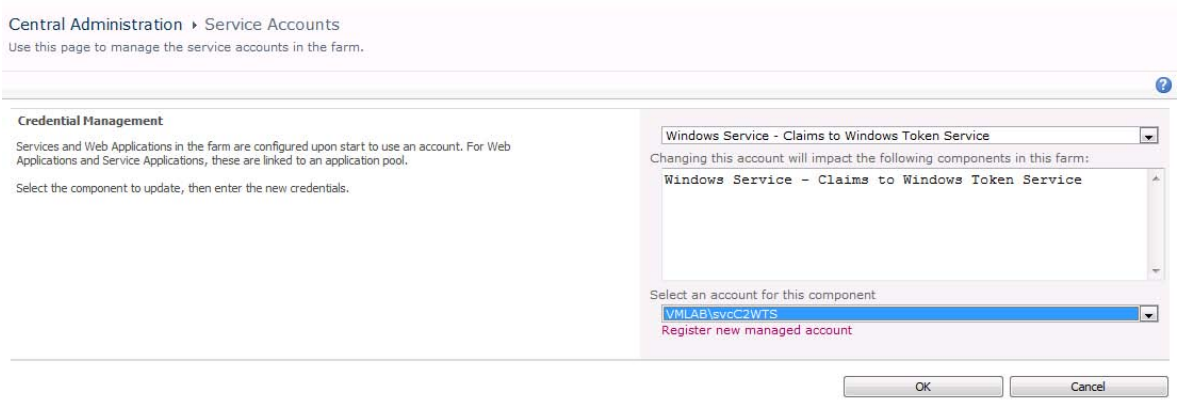


8. In the server selection box in the upper right hand corner select the server(s) running excel services. In this example it is VMSP10APP01:

9. Find the Claims to Windows Token Service and start it:



10. Go to Security->Manage Service Accounts. Change the identity of the C2WTS to the new managed acount.



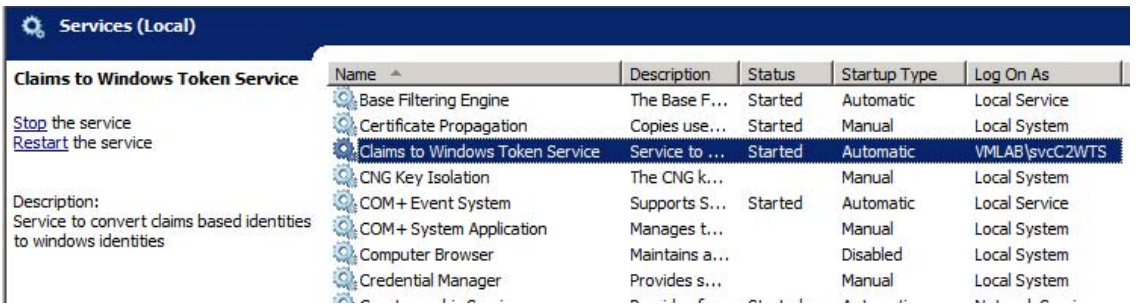
Note:

If the C2WTS was already running before configuring the dedicated service account, or if you need to changes the permissions of the service account after the C2WTS is running you must restart the C2WTS from the services console.

In addition, if you experience issues with the C2WTS after restarting the service it may also be required to reset the IIS application pools that communicate with the C2WTS.

Add Startup dependencies the WIF C2WTS service

There is a known issue with the C2WTS where it may not automatically startup successfully on system reboot. A workaround to the issue is to configure a service dependency on the Cryptographic Services service:

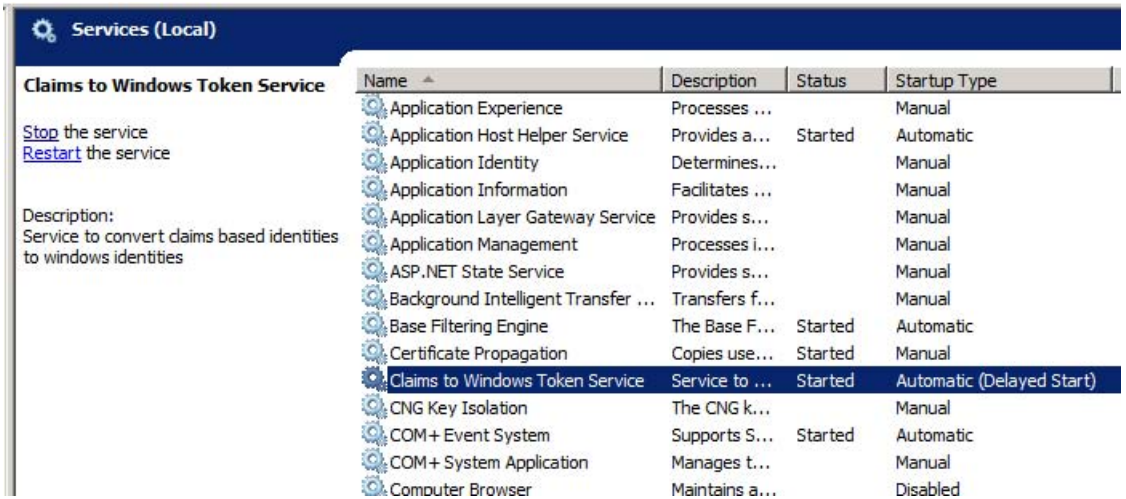


In addition, if you experience issues with the C2WTS after restarting the service it may also be required to reset the IIS application pools that communicate with the C2WTS.

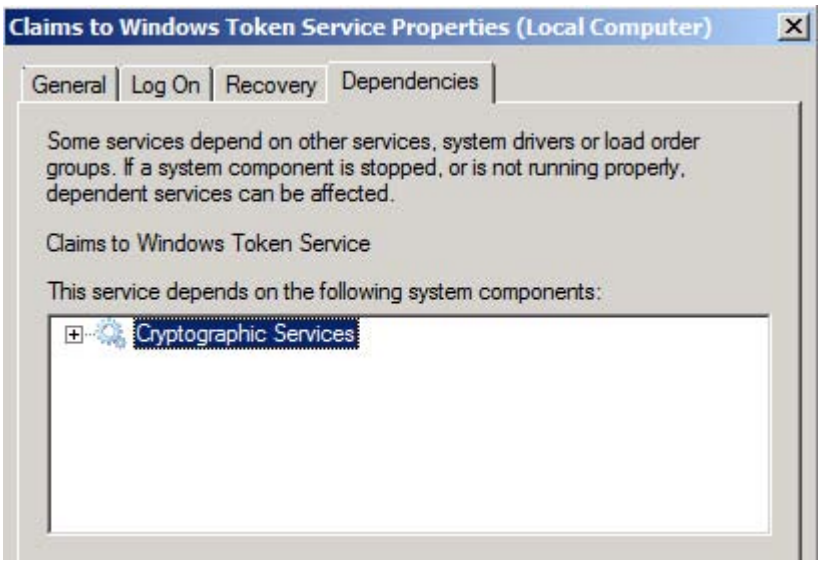
1. Open the Command Prompt window.
2. Type: **scconfig"c2wts" depend= CryptSvc**

```
C:\>sc config "c2wts" depend= CryptSvc
[SC] ChangeServiceConfig SUCCESS
```

3. Find the Claims to Windows Token Service in the services console.
4. Open the properties for the service.



5. Check the **Dependencies** tab. Make sure **Cryptographic Services** is listed.



6. Click **OK**.

Grant the Excel Services service account permissions on the web application content database

A required step in configuring SharePoint Server 2010 Office Web Applications is allowing the web application's service account access to the content databases for a given web application. In this example, we will grant the Excel Services service account access to the "portal" web application's content database by using Windows PowerShell.

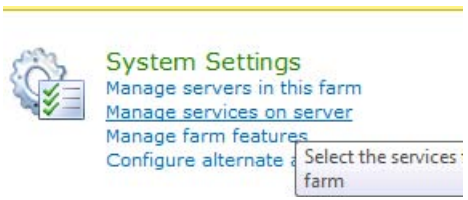
Run the following command from the SharePoint 2010 Management Shell:

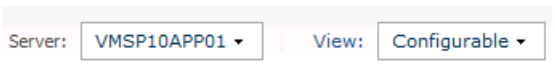
```
$w = Get-SPWebApplication -Identity http://portal

$w.GrantAccessToProcessIdentity("vmlab\svcExcel")
```

Start the Excel Services service instance on the Excel Services server

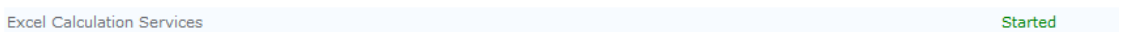
Before creating an Excel Services service application, start the excel services serve service on the designated Farm servers.

1. Open Central Administration.
2. Under services, select **Manage services on server**.





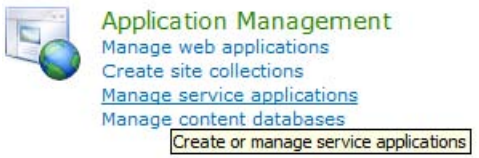
3. In the server selection box in the upper right hand corner select the server(s) running excel services. In this example it is VMSP10APP01.
4. Start the Excel Calculation Services service.



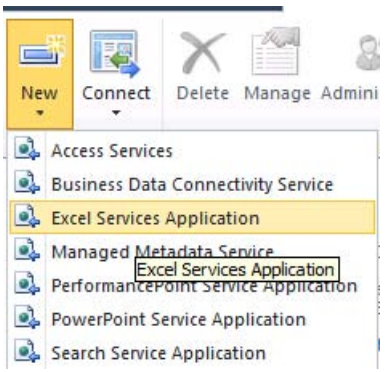
Create the Excel Services service application and proxy

Next configure a new Excel Services service application and application proxy to allow web applications to consume Excel Services:

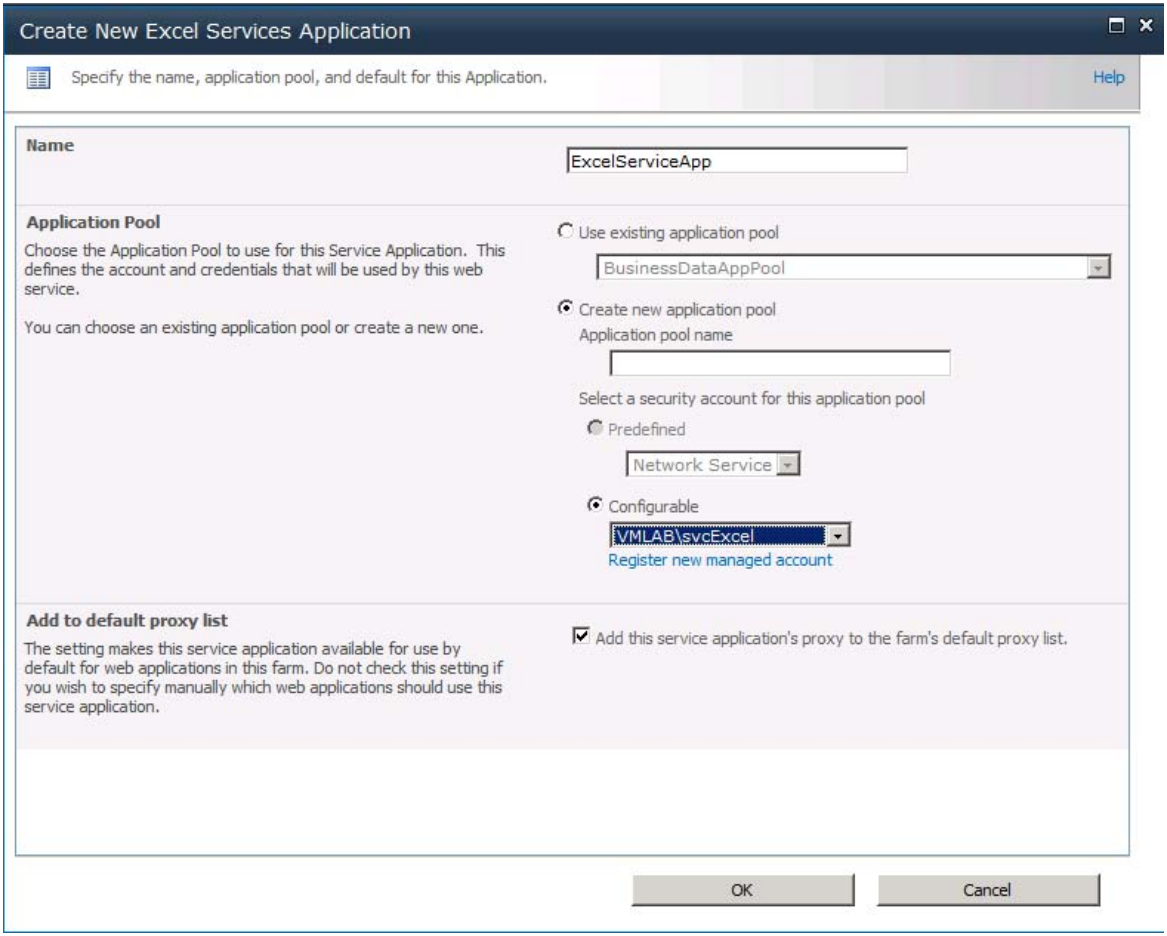
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



3. Select **New**, and then click **Excel Services Application**.



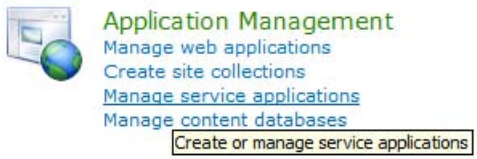
4. Configure the new service application. Be sure to select the correct service account (create a new managed account if the excel service account is not in the list).



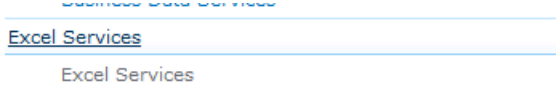
Configure Excel services trusted file location and authentication settings

Once the Excel Services application is created, configure the properties on the new service application to specify a trusted host location and authentication settings.

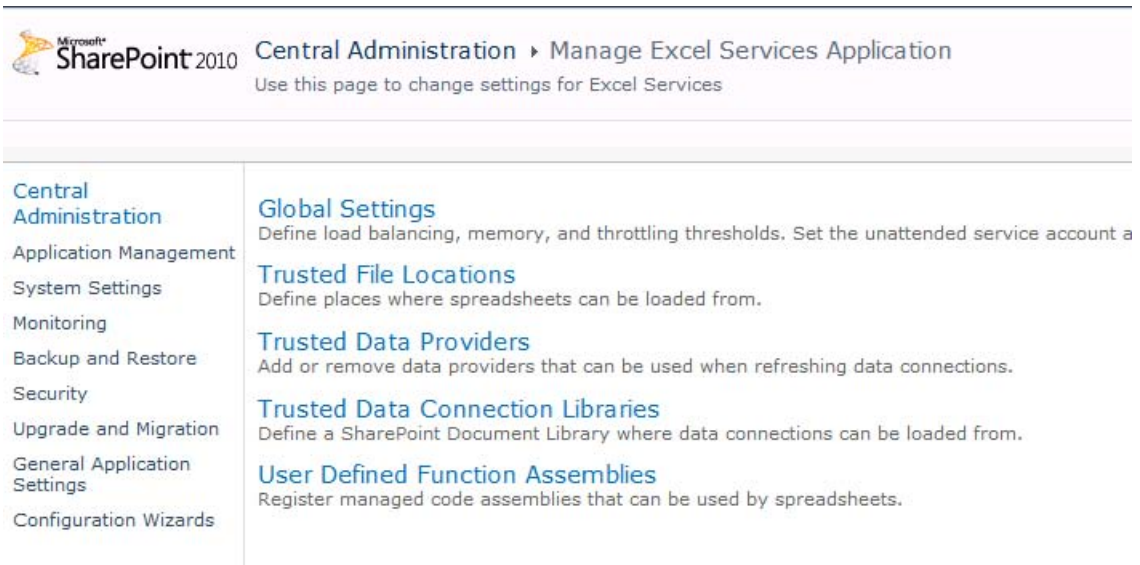
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



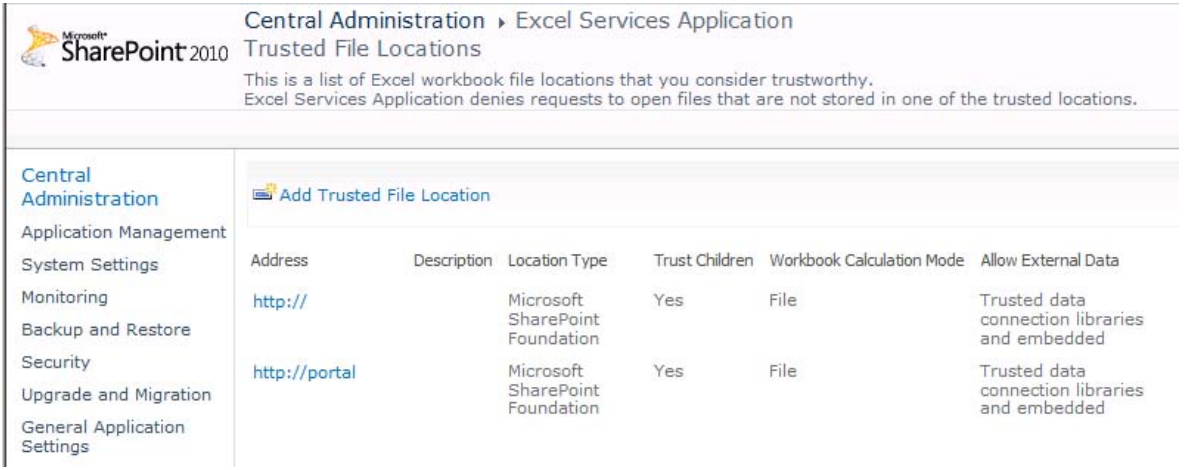
3. Click the link for the new Service Application, **Excel Services** in this example.



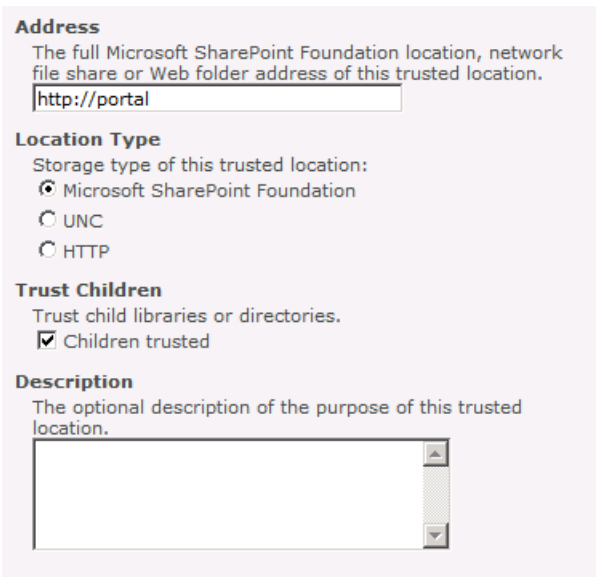
4. In the Excel Services management page, click "Trusted File Locations".



5. Add a new trusted file location.



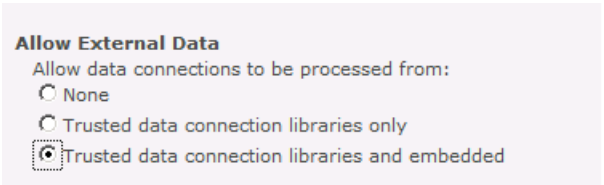
6. Specify the location to your test library.



Note:

In our example, we trust the root web application URL and all children. In a production environment you may choose to constrain the trust to a more granular location.

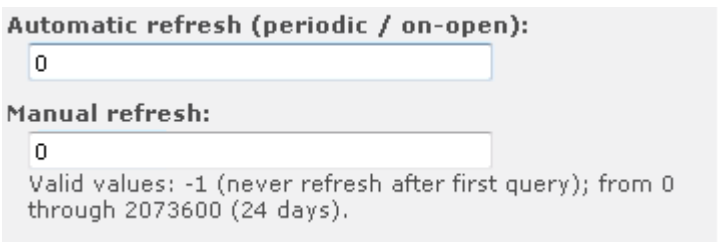
7. In **External Data** Select **trusted data connection libraries and embedded**.



Note:

This example will use an embedded connection to connect to SQL Server. In your environment you may choose to create a separate connection file and store it in a trusted data connection library. In that case you might select Trusted data connection libraries only.

8. Change the External Data Cache Age — For testing purposes, it is convenient to change the external data cache lifetime to ensure data refreshes are coming from the data source and not the cache. Under External Data, change the following settings:



Automatic refresh (periodic / on-open) = 0

Manual refresh = 0

Note:

In a production environment you will want to configure a cache setting higher than 0. Setting the cache to 0 is for testing purposes only.

Verify Excel Services constrained delegation

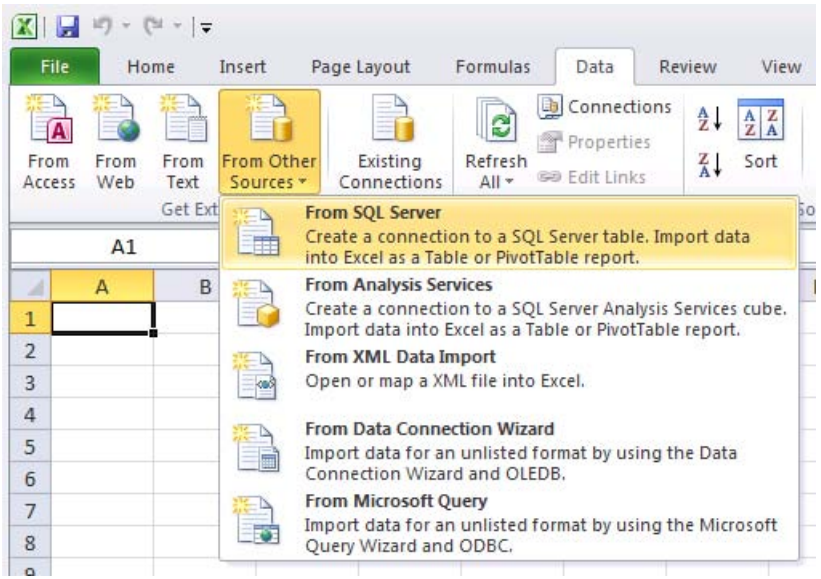
Create document library to host the test workbook

Open a site in the trusted path that was configured in the previous step. Create a new document library to host a test Excel workbook.

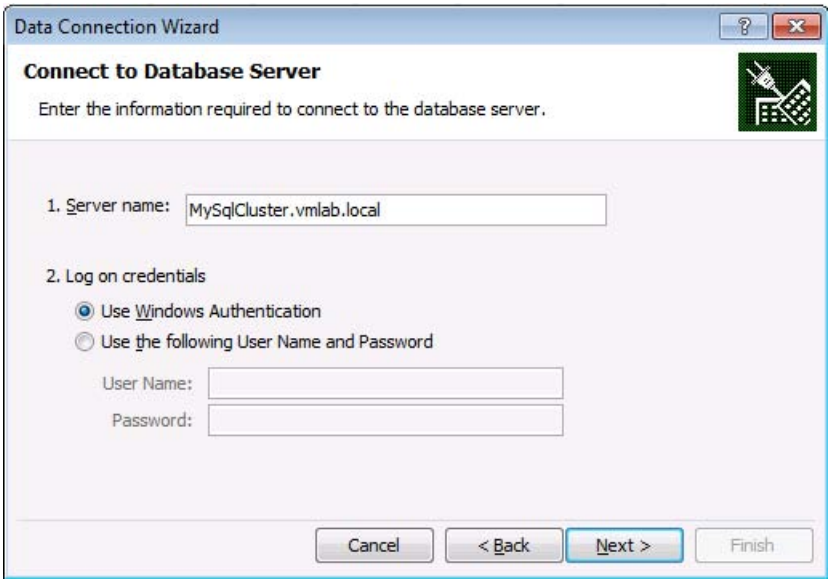
Create test Excel workbook with SQL data connection

Next create an Excel workbook with a data connection to the new test database:

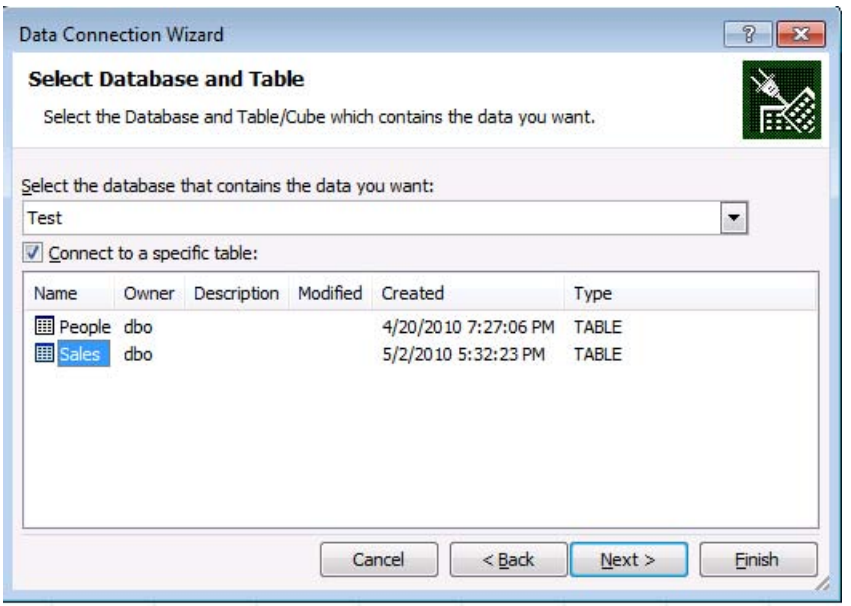
1. Open Excel.
2. On the **Data** tab, select **From other sources**->**From SQL Server**.



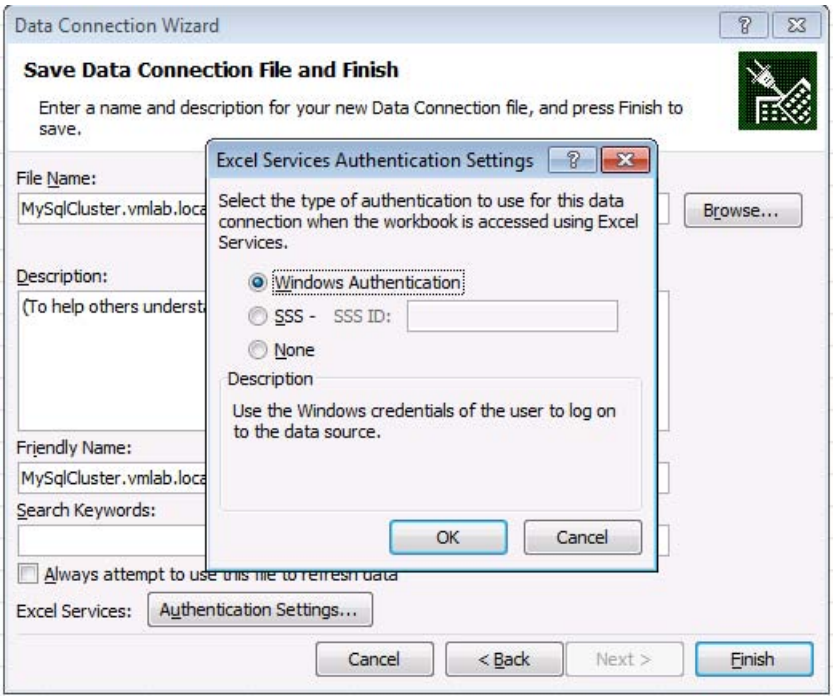
3. Connect to the test SQL data source.



4. Select the test database and the test table (**Sales in our example**).

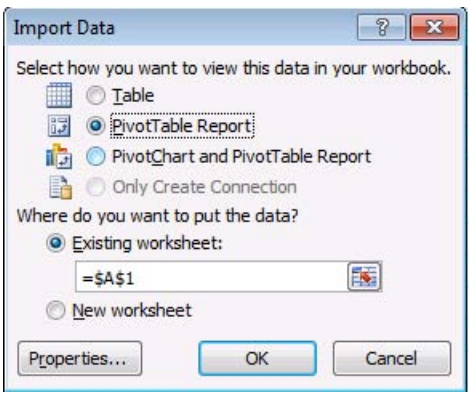


5. Click Next. Click the authentication settings button. Ensure Windows Authentication is specified.



6. Click Finish.

7. Select Pivot Table Report.



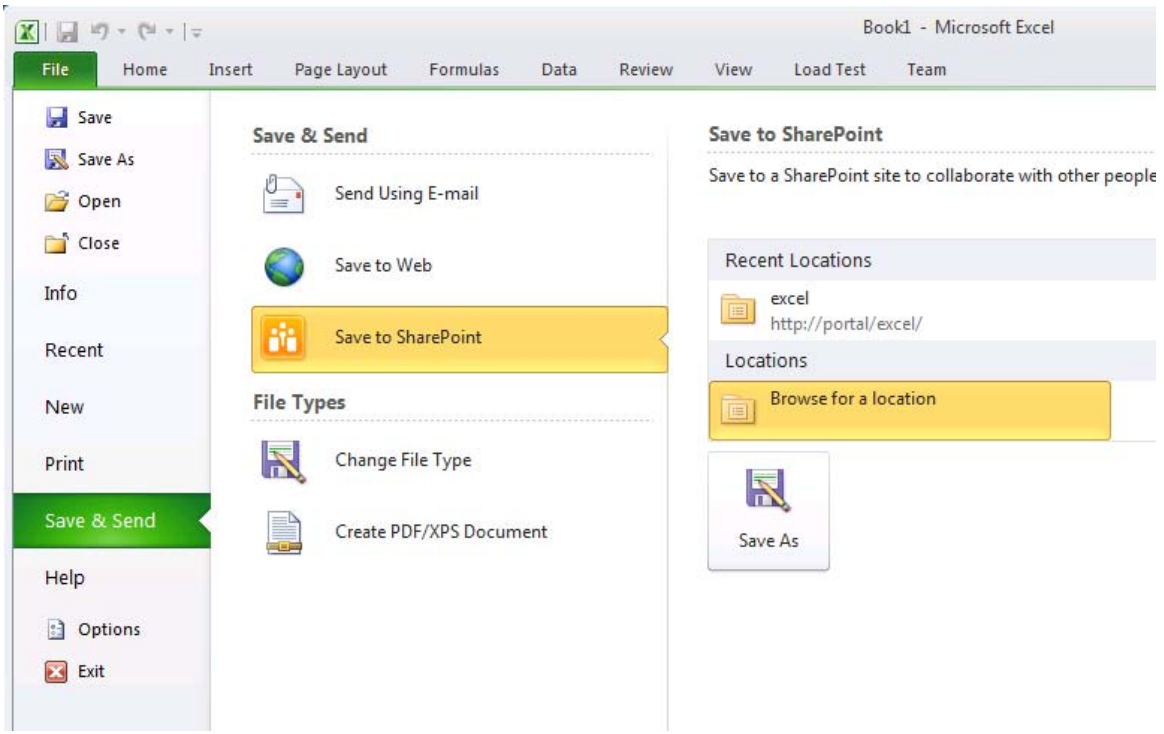
8. Configure the pivot table. Ensure data is returned from the SQL source.

	A	B	C	D	E
1	Sum of Amount	Column Labels			
2	Row Labels	UK	US	Grand Total	
3	2006	13456.21	12654.23	26110.44	
4	2007	14321.47	15443.12	29764.59	
5	2008	19234.89	19837.23	39072.12	
6	2009	18233.45	13998.78	32232.23	
7	Grand Total	65246.02	61933.36	127179.38	
8					
9					

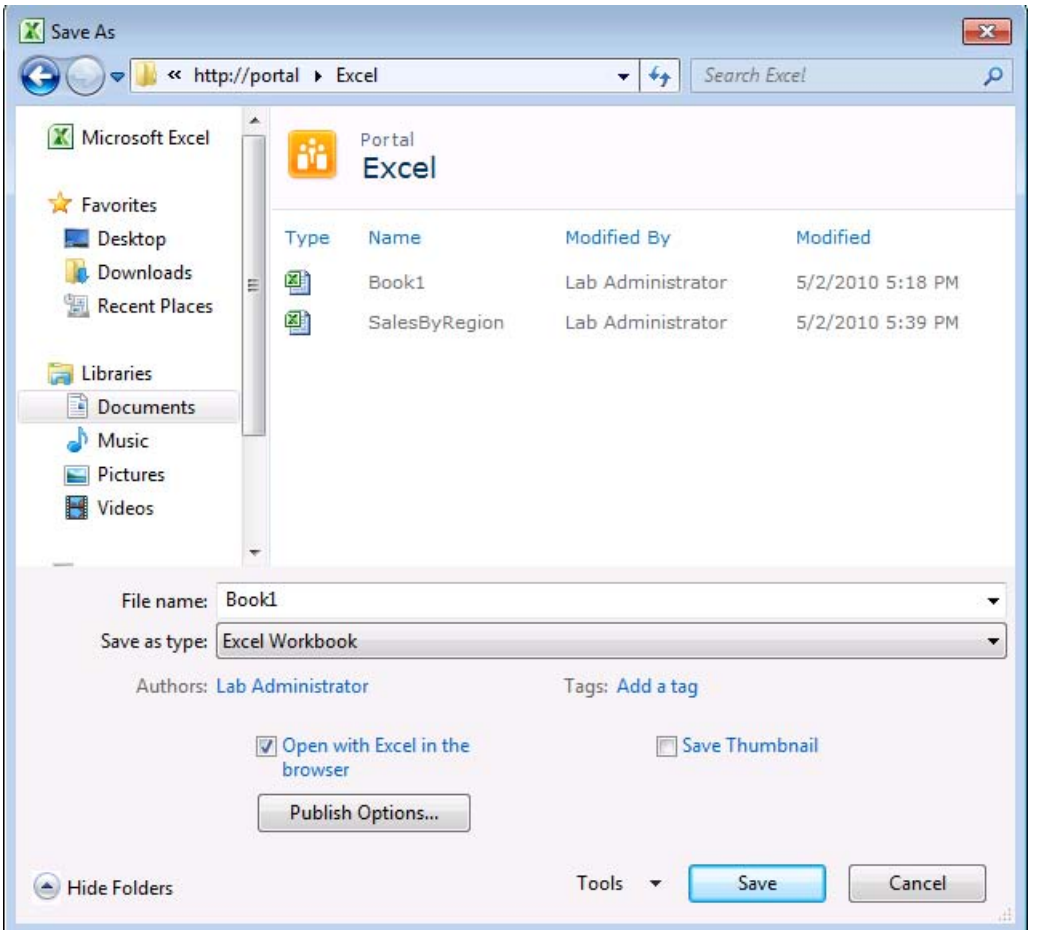
Publish workbook to SharePoint Server and refresh data connection

The last step to validate the Excel Services application is to publish the workbook and test refreshing the embedded SQL connection.

1. Click the **File** tab.
2. Click **Save and Send**, then click **Save to SharePoint**, and then click **Browse for a location**.

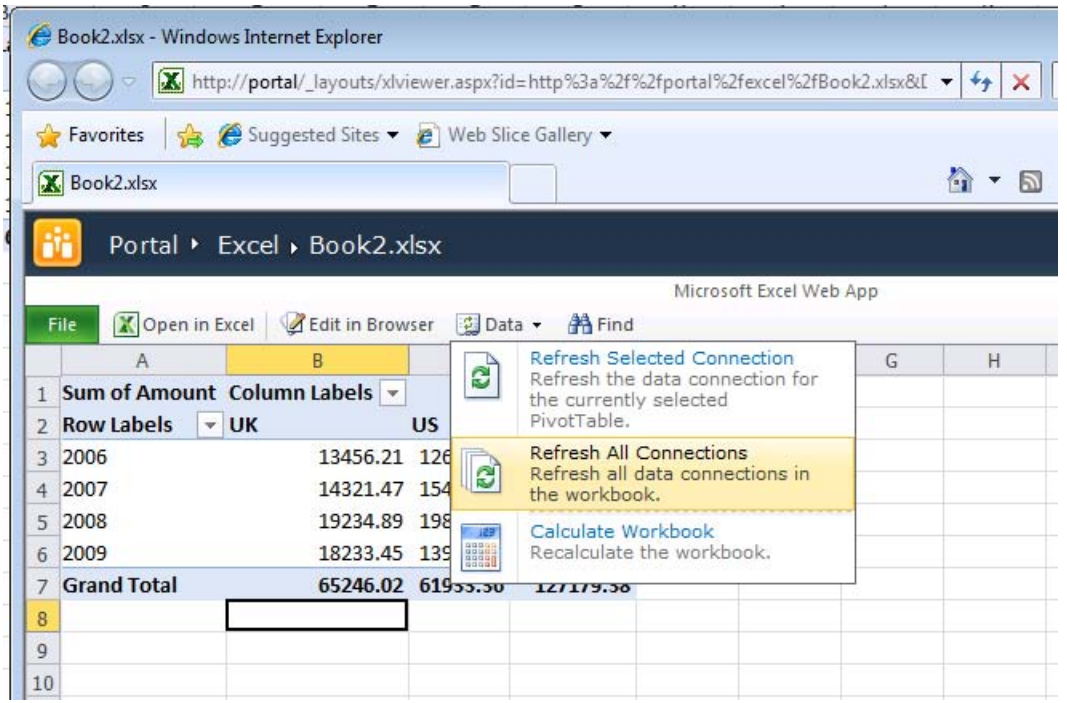


3. Enter the location to the trusted library created in previous steps.



4. Ensure **Open with Excel in the browser** is selected.

   A new browser window will open at this point with your test workbook displayed. Once the workbook renders, refresh the data connection by clicking **Data** and then clicking **Refresh All Connections**.



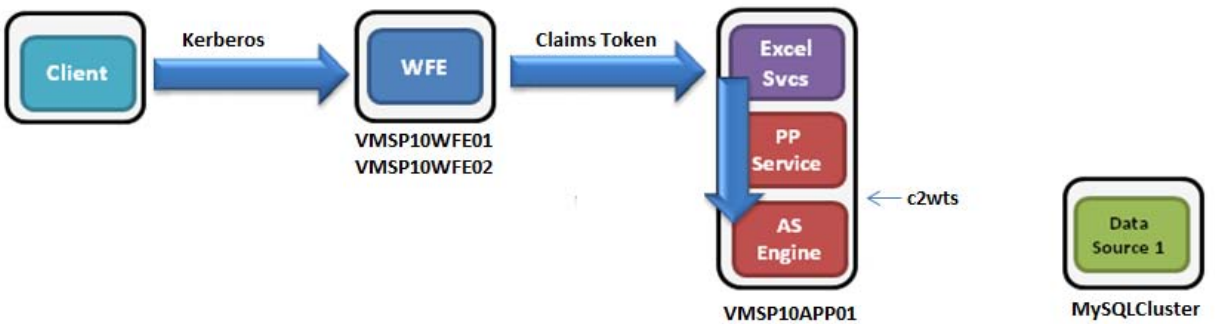
If the data connection refreshes you have successfully configured Kerberos delegation for excel services. To further test connectivity, change the source data via SQL Management Studio then refresh the connection. You should see the newly changed data in your workbook. If you do not see any changes, and you do not receive any errors on refresh you are most likely seeing cached data. By default, Excel Services will cache data from external sources for five minutes. You can change this cache setting; see Configure Excel services trusted file location and authentication settings in this article for more information.

Identity delegation for PowerPivot for SharePoint 2010 (SharePoint Server 2010)

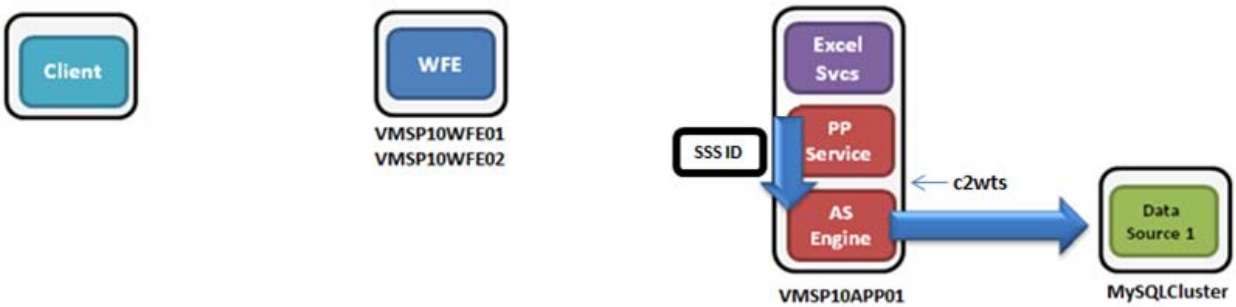
Published: December 2, 2010

The farm topology described in Environment and farm topology does not require Kerberos authentication for PowerPivot for Microsoft SharePoint 2010 to work. The PowerPivot System Service is claims aware, and uses the Claims To Windows Token Service (C2WTS) to recreate the client's Windows identity using the client's claims token in order to connect with the Analysis Service Vertipaq engine that runs on the application server.

When a PowerPivot workbook is uploaded in SharePoint Server, it already contains the PowerPivot data that the workbook uses. When the user opens the PowerPivot workbook in Excel Web Access and interacts with the slicers, the PowerPivot System Service loads the data in the workbook directly into its Analysis Services engine. No access is made to the data connection embedded in the workbook.



When a data refresh job for a PowerPivot workbook starts executing, the PowerPivot System Service performs a Windows login using the credentials stored in the SharePoint Server Secure Store Service. Since the Windows identity is created on the application server, the connection from the PowerPivot Analysis Services Vertipaq engine (on the same computer, VMSP10APP01) to MySQLCluster is the first NTLM hop.



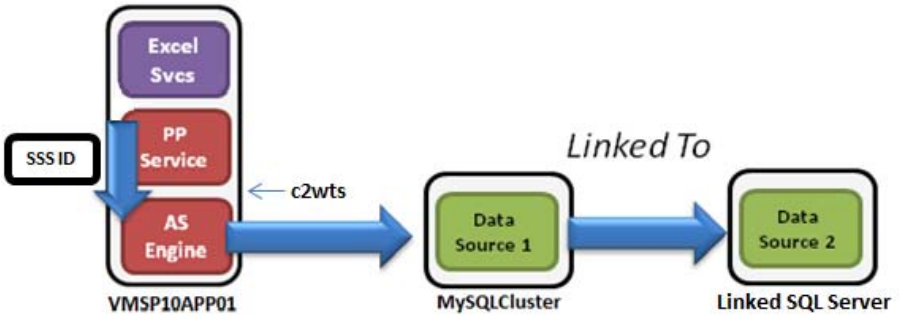
Note:

If you are installing on Windows Server 2008, you may have to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

Scenarios requiring Kerberos authentication

As you can see from the discussion above, most common situations with PowerPivot do not require Kerberos authentication. However, there are some unusual edge cases where Kerberos authentication would be required. For example, if your PowerPivot workbook contains a data connection to a SQL Server instance that is linked to yet another SQL Server instance on a separate computer, you will need to configure Kerberos authentication with identity delegation for data refresh to work. For example, if MySQLCluster is linked to another remote SQL Server instance, then the link from MySQLCluster to the linked remote server is the second hop. In this case, NTLM is no longer adequate. You must configure Kerberos delegation for the data refresh to process successfully.



While they are outside the scope of the scenarios defined in this paper, the major steps to configure identity delegation for PowerPivot are as follows:

1. Change the service account of the C2WTS Windows service to a domain account (e.g. VMLAB\svcC2WTS). Configuring the C2WTS is a large topic and is covered in detail in the other scenarios in this document:
    - Configure and Start the Claims to Windows Token Service on Excel Services Servers
    - Configure and Start the Claims to Windows Token Service on Visio Graphics Servers
    - Configure and Start the Claims to Windows Token Service on PerformancePoint Services Servers
2. Configure delegation from the VMLAB\svcSQL account to the SPN for the linked SQL Server instance Configuration Checklist.

Area of configuration	Description
PowerPivotinstallation	Install SQL Server PowerPivot for SharePoint on the application server

Scenario dependencies

Strictly speaking, the following Kerberos authentication scenarios are not required by PowerPivot for SharePoint. However it expedites your PowerPivot for SharePoint

installation process if you successfully completed them, as the components themselves are prerequisites for PowerPivot for SharePoint.

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP
- (Optional) Scenario 3: Kerberos Authentication for SQL Analysis Services
- Scenario 5: Identity Delegation for Excel Services

Configuration instructions

Install PowerPivot for SharePoint on the application server (vmsp10app01). For detailed instructions, see How to: Install PowerPivot for SharePoint in a Three-tier SharePoint Farm in the MSDN Library online. If you have already performed the dependent scenarios in this paper, you can skip the sections in the MSDN article that have already been covered by the scenario dependencies.

Important:

The application pool for the SQL Server PowerPivot Service Application must be run using the domain account of the SharePoint Server farm administrator. In no other user context can the PowerPivot System Service retrieve the unattended account credentials from the Secure Store Service.

Identity delegation for Visio Services (SharePoint Server 2010)

Published: December 2, 2010

In this scenario, you add a Visio Services service application to the SharePoint Server environment and configure Kerberos constrained delegation to allow the service to refresh data from an external SQL Server data source in a Visio web drawing.

Note:

If you are installing on Windows Server 2008, you may have to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

Scenariodependencies

To complete this scenario you will need to have completed:

- Scenario 1: Core Configuration
- Scenario 2: Kerberos authentication for SQL OLTP

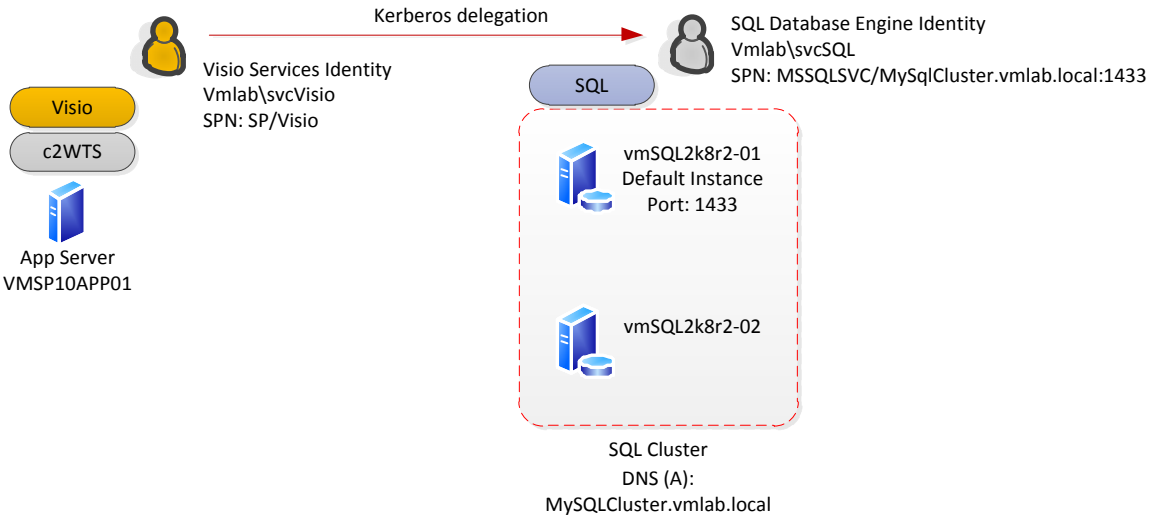
Configurationchecklist

Area of Configuration	Description
Active Directory Configuration	Create Visio Services service account Configure SPN on Visio Services service account

Area of Configuration	Description
	Configure Kerberos constrained delegation for servers running Visio Services Configure Kerberos constrained delegation for the Visio Services service account
SharePoint Server configuration	Start Claims to Windows Token Service on Visio Services Servers Grant the Visio Services service account permissions on the web application content database Start the Visio Services service instance on the Visio Services server Create the Visio Services service application and proxy
Verify Visio Services Constrained Delegation	Configure the Visio services cache settings Create document library to host test Visio Diagram Create a test Visio web drawing with SQL Server data connected shapes Publish the Visio drawing to SharePoint Server and refresh data connection

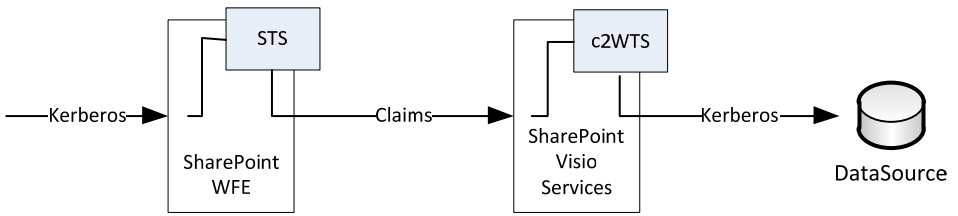
Scenario environment details

Kerberos constrained delegation paths



In this scenario, we will configure the SharePoint Server Visio services application servers and service accounts for Kerberos constrained delegation to the SQL Server service.

SharePoint Server logical authentication



Authentication in this scenario begins with the client authenticating with Kerberos authentication at the web front end. SharePoint Server 2010 will convert the Windows authentication token into a claims token using the local Security Token Service (STS). The Visio service application will accept the claims token and convert it into a windows token (Kerberos) using the local Claims to Windows Token Service (C2WTS) that is a part

of Windows Identity Foundation (WIF). The Visio service application will then use the client's Kerberos ticket to authenticate with the backend data source.

Step-by-step configuration instructions

Active Directory configuration

Create Visio Services service account

As a best practice, Visio Services should run under its own domain identity. To configure the Excel Service Application, an Active Directory account must be created. In this example, the following accounts were created:

SharePoint Server service	IIS App Pool Identity
Visio Services	vmlab\svcVisio

Configure SPN on Visio Services service account

Kerberos constrained delegation must be configured if Visio Services is going to delegate the client's Windows identity to back end data source. In this example Visio services will query data from a SQL Server transactional database as the client therefor Kerberos delegation is required.

The Active Directory Users and Computers MMC snap-in is typically used to configure Kerberos delegation. To configure the delegation settings within the snap-in, the Active Directory object being configured must have a service principal name applied; otherwise the **delegation** tab for the object will not be visible in the object's properties dialog. Although Visio Services does not require a SPN to function, we will configure one for this purpose.

On the command line, run the following command:

```
SETSPN -S SP/VisioServices svc\VisioServices
```

Note:

The SPN is not a valid SPN. It is applied to the specified service account to reveal the delegation options in the AD users and computers add-in. There are other supported ways of specifying the delegation settings (specifically the msDS-AllowedToDelegateTo AD attribute) but this topic will not be covered in this document.

Configure Kerberos constrained delegation for Visio Services

To allow Visio Services to delegate the client's identity Kerberos constrained delegation must be configured. It is required to configure constrained delegation with protocol transition for the conversion of claims token to windows token via the WIF C2WTS.

Each server running Visio services must be trusted to delegate credentials to each back-end service Visio will authenticate with. In additional, the Visio services service account must also be configured to allow delegation to the same back-end services.

In our example the following delegation paths are defined:

Principal Type	Principal Name	Delegates To Service
User	Vmlab\svcVisio	MSSQLSVC/MySqlCluster.vmlab.local:1433
*User	Vmlab\svcC2WTS	MSSQLSVC/MySqlCluster.vmlab.local:1433
**Computer	Vmlab\vmsp10app01	MSSQLSVC/MySqlCluster.vmlab.local:1433

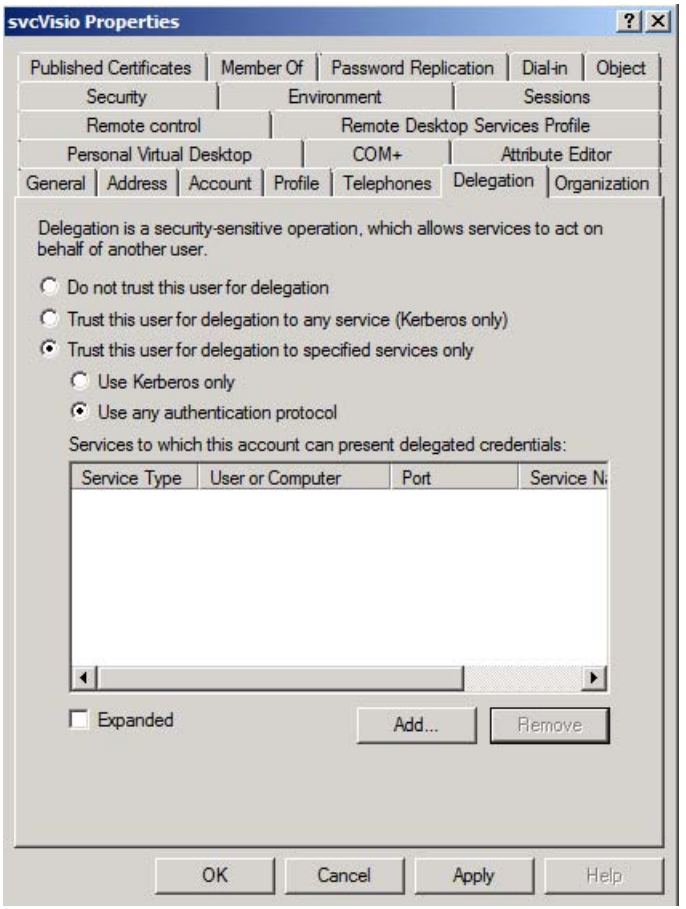
* Configured later in this scenario

** Optional. Constrained delegation on the computer account is only required when running the C2WTS as Local System

To configure constrained delegation

1. Open the Active Directory Object's properties in Active Directory Users and Computers.

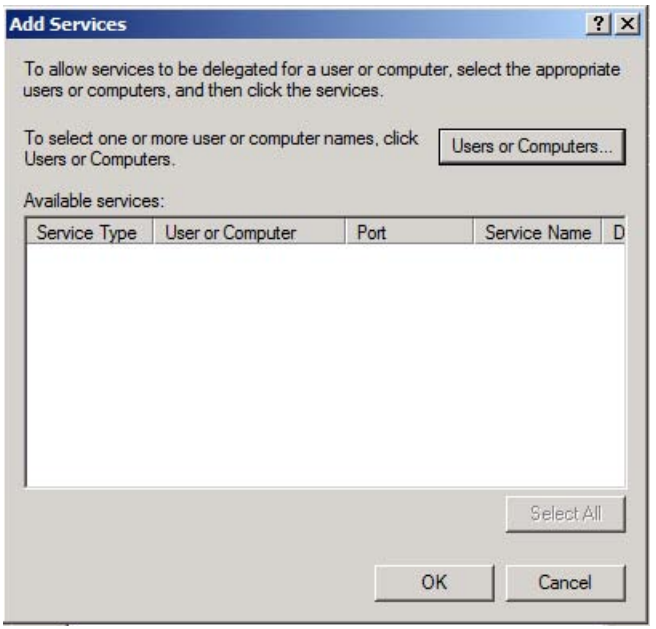
2. Navigate to the **Delegation** tab.



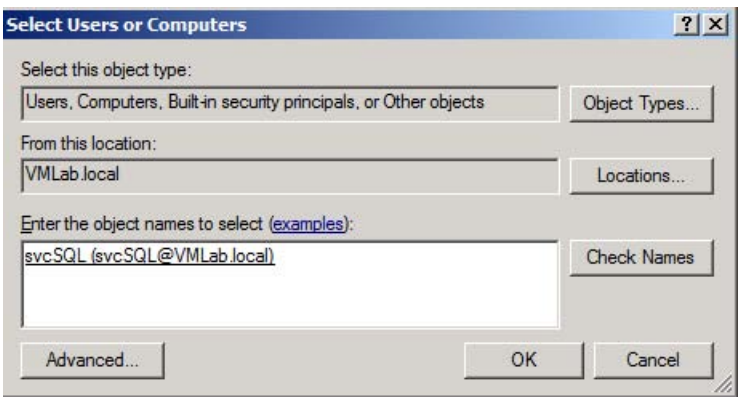
3. Select **Trust this user for delegation to specified services only**.

4. Select **Use any authentication protocol**. This enables protocol transition and is required for the Visio service account to use the C2WTS.

5. Click the add button to select the service principal allowed to delegate to.



6. Select **User and Computers**.

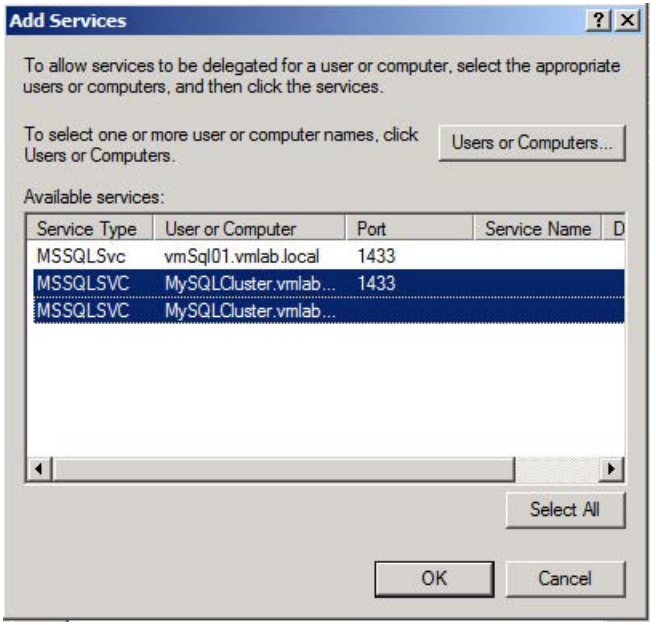


7. Select the service account running the service you wish to delegate to. In this example it is the service account for the SQL Server service.

Note:

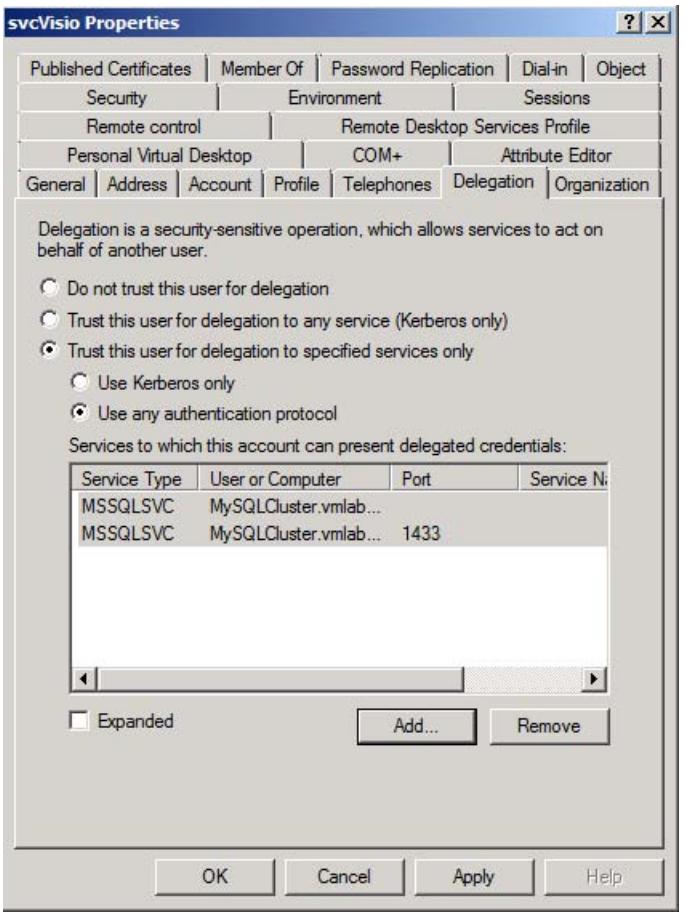
the service account selected must have a SPN applied to it. In our example the SPN for this account was configured in a previous scenario.

8. Click **OK**. You will then be asked to select the SPNs you would like to delegate to.



9. Select the services for the SQL Server cluster and click **OK**.

10. You should now see the selected SPNS in the **services to which this account can presented delegated credentials** list.



11. Repeat these steps for each delegation path (Computer and User) defined in the beginning of this section.

Verify MSSQLSVC SPN for the Service Account running the service on the SQL Server (performed in Scenario 2)

Verify the SPN for Analysis Services service account (vmlab\svcSQL) exists with the following SetSPN command:

```
SetSPN -L vmlab\svcSQL
```

You should see the following:

```
MSSQLSVC/MySqlCluster

MSSQLSVC/MySqlCluster.vmlab.local:1433
```

SharePoint Server configuration

Configure and Start the Claims to Windows Token Service on Visio Graphics Servers

The Claims to Windows Token Service (C2WTS) is a component of the Windows Identity Foundation (WIF) which is responsible for converting user claim tokens to windows tokens. The Visio graphics service uses the C2WTS to convert the user's claims token into a windows token when the services needs to delegate credentials to a back-end system which uses Windows authentication. WIF is deployed with SharePoint Server 2010 and the C2WTS can be started from Central Administration.

Each Visio Graphics Service application server must run the C2WTS locally. The C2WTS does not open any ports and cannot be accessed by a remote caller. Further, the C2WTS service configuration file must be configured to specifically trust the local calling client identity.

As a best practice you should run the C2WTS using a dedicated service account and not as Local System (the default configuration). The C2WTS service account requires special local permissions on each server the service runs on so be sure to configure these permissions each time the service is started on a server. Optimally you should configure the service account's permissions on the local server before starting the C2WTS, but if done after the fact you can restart the C2WTS from the Windows services management console (services.msc).

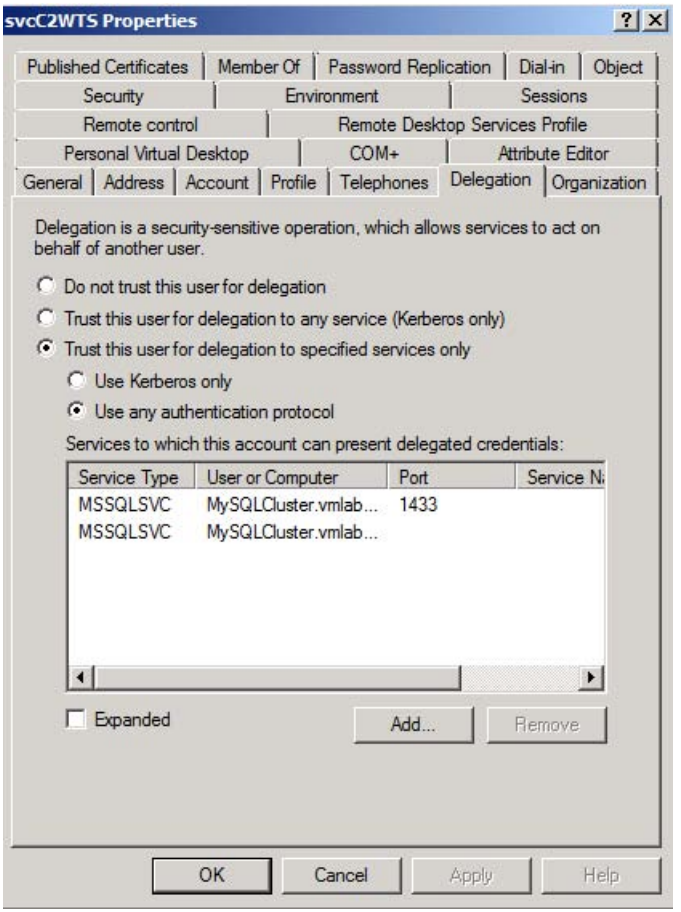
To start the C2WTS

1. Create a service account in Active Directory to run the service under. In this example we created vmlab\svcC2WTS.

2. Add an arbitrary Service Principal Name (SPN) to the service account to expose the delegation options for this account in Active Directory Users and Computers. The SPN can be any format because we do not authenticate to the C2WTS using Kerberos authentication. It is recommended to not use an HTTP SPN to avoid potentially creating duplicate SPNs in your environment. In our example we registered SP/C2WTS to the vmlab\svcC2WTS using the following command:

   ```
   SetSPN -S SP/C2WTS vmlab\svcC2WTS
   ```

3. Configure Kerberos constrained delegation on the C2WTS services account. In this scenario we will delegate credentials to the SQL Server service running with the MSSQLSVC/MySqlCluster.vmlab.local:1433 service principal name.



4. Configure the required local server permissions that the C2WTS requires. You will need to configure these permissions on each server the C2WTS runs on. In our example, this is VMSP10APP01. Log on to the server and give the C2WTS the following permissions:

    a) Add the service account to the local Administrators Groups.

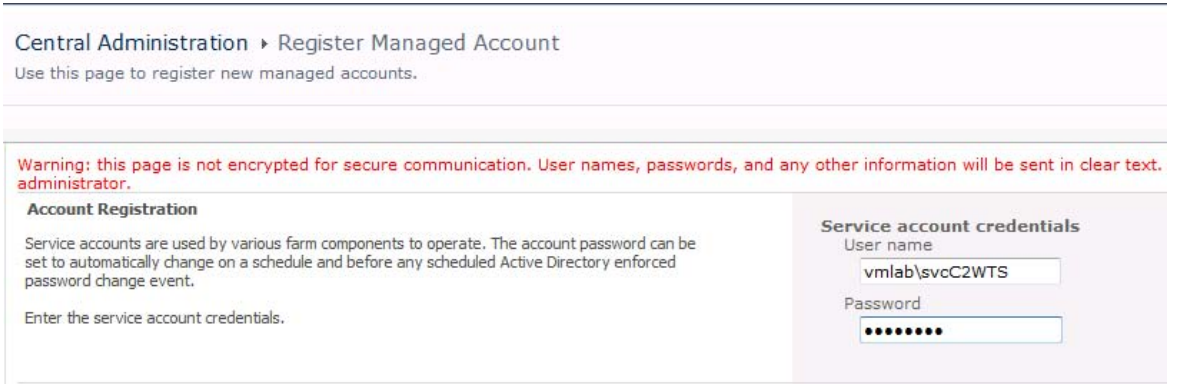
    b) In local security policy (secpol.msc) under user rights assignment give the service account the following permissions:

        i. **Act as part of the operating system**

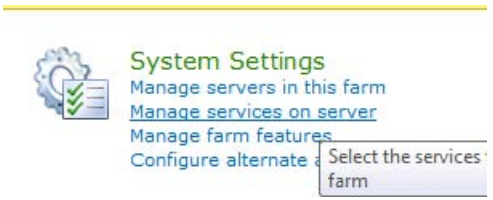
        ii. **Impersonate a client after authentication**

        iii. **Log on as a service**

5. Open Central Administration.
6. In **Security**, in the **Configure Managed Service Accounts** section, register the C2WTS service account as a managed account.

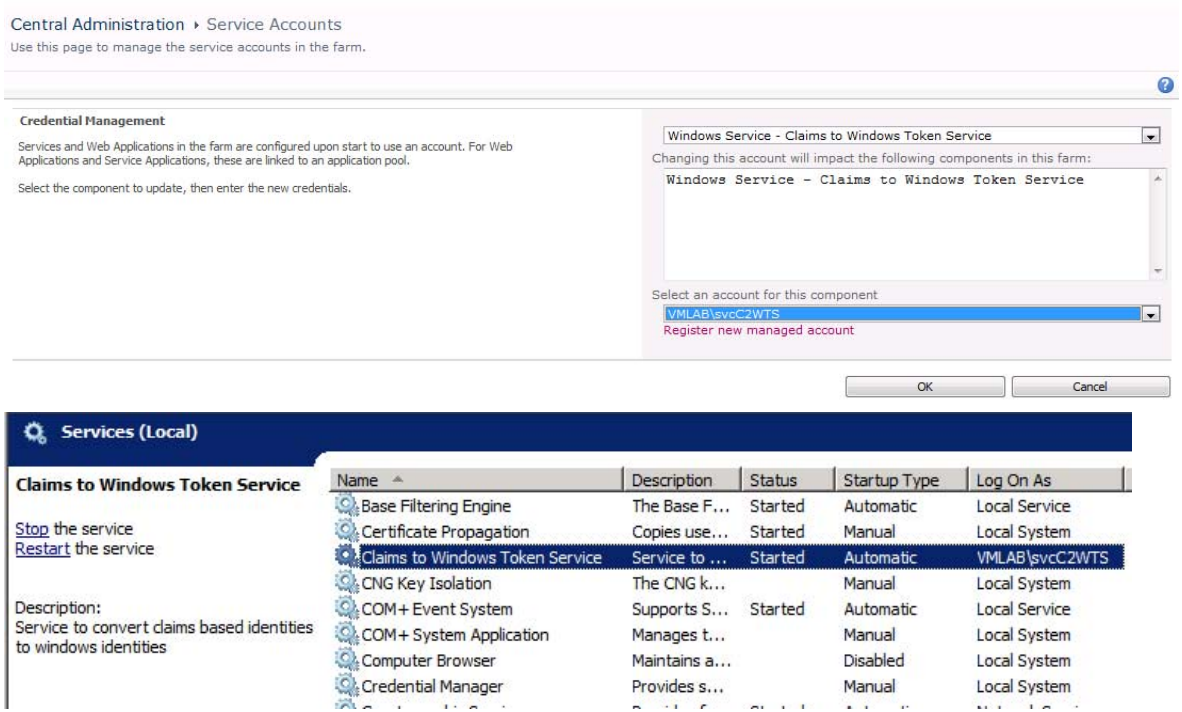


7. Under services, select **Manage services on server**.



8. In the server selection box in the upper right corner, select the server(s) that is or are running the Visio Graphics Service. In this example it is VMSP10APP01.

9. Find the **Claims to Windows Token Service** and start it.
10. Go to **Manage Service Accounts** in the **Security** section. Change the identity of the C2WTS to the new managed acount.



Note:

If the C2WTS was already running before configuring the dedicated service account, or if you need to changes the permissions of the service account after the C2WTS is running you must restart the C2WTS from the services console.

In addition, if you experience issues with the C2WTS after restarting the service it may also be necessary to reset the IIS application pools that communicate with the C2WTS.

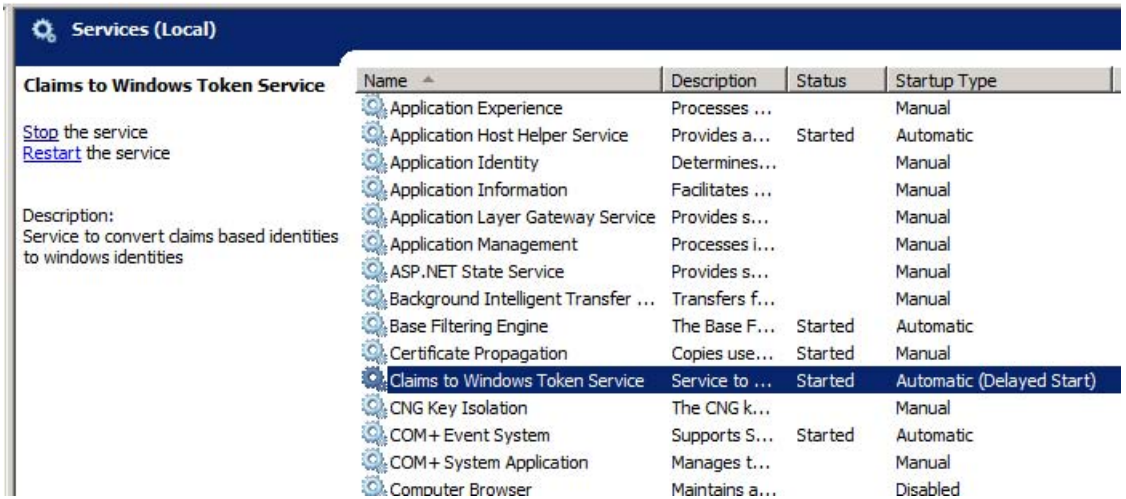
Add Startup dependencies the WIF C2WTS service

There is a known issue with the C2WTS where it may not automatically start up successfully on system reboot. A workaround to the issue is to configure a service dependency on the Cryptographic Services service:

1. Open a Command Prompt window.
2. Type: **scconfig"c2wts" depend= CryptSvc**

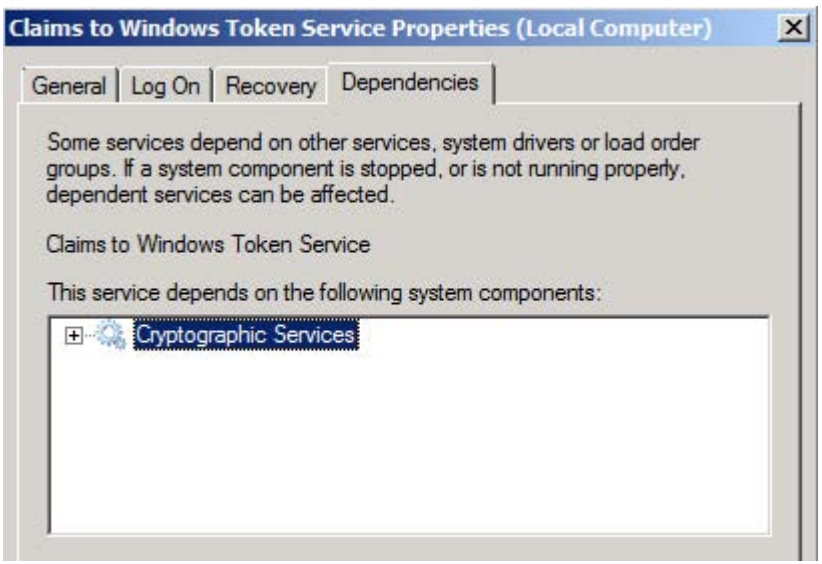
```
C:\>sc config "c2wts" depend= CryptSvc
[SC] ChangeServiceConfig SUCCESS
```

3. Find the Claims to Windows Token Service in the services console.



4. Open the properties for the service.

5. Check the **Dependencies** tab. Make sure **Cryptographic Services** is listed:



6. Click **OK**.

Grant the Visio Services service account permissions on the web application content database

A required step in configuring SharePoint Server 2010 Office Web Applications is allowing the web application's service account access to the content databases for a given web application. In this example, we will grant the Visio Graphics Service account access to the **portal** web application's content database by using Windows PowerShell.

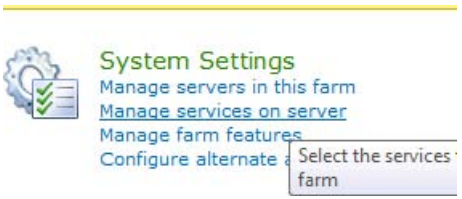
Run the following command from the SharePoint 2010 Management Shell:

```
$w = Get-SPWebApplication -Identity http://portal

$w.GrantAccessToProcessIdentity("vmlab\svcVisio")
```

Start the Visio Graphics Service instance on the Visio server

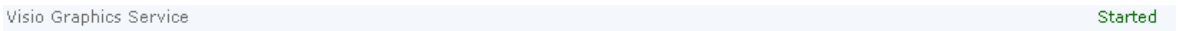
Before creating a Visio Services service application, start the Visio services server service on the designated Farm servers.

1. Open Central Administration.
2. Under services, select **Manage services on server**.



3. In the server selection box in the upper right hand corner select the server(s) running excel services. In this example it is VMSP10APP01.

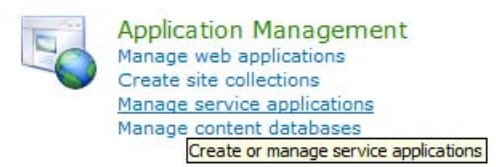
4. Start the **Visio Graphics Service**.



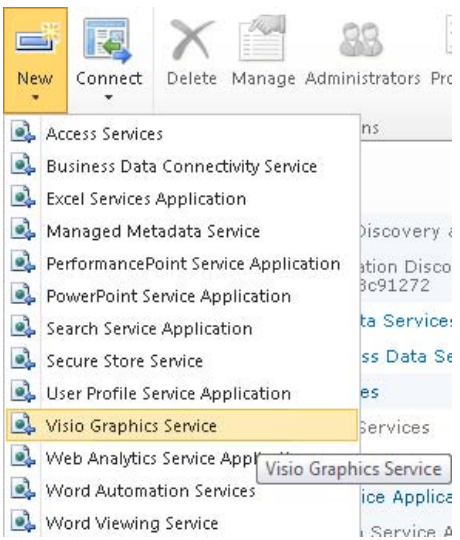
Create the Visio Graphics Service application and proxy

Next, configure a new Excel Services service application and application proxy to allow Web applications to consume Excel Services (if one does not already exist):

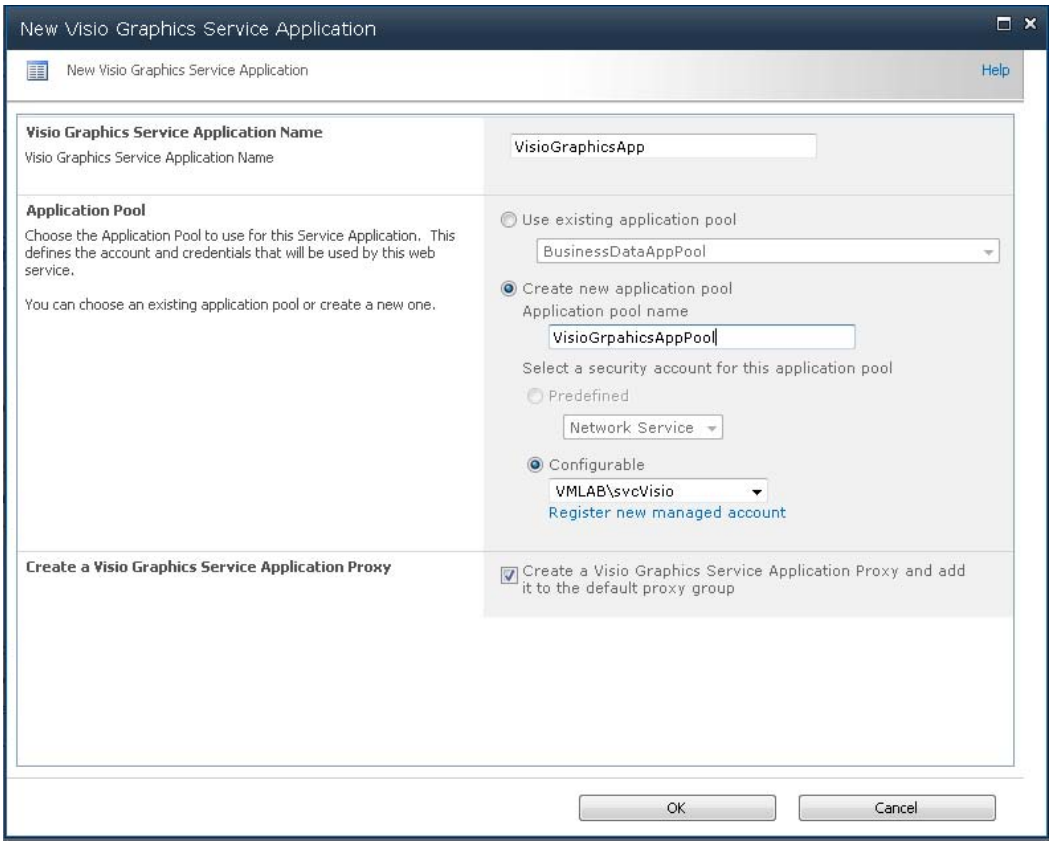
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



3. Select **New**, and then select **Visio Graphics Service**.



4. Configure the new service application. Be sure to select the correct service account (create a new managed account if the Visio service account is not in the list).



Verify Visio Graphic Service Constrained Delegation

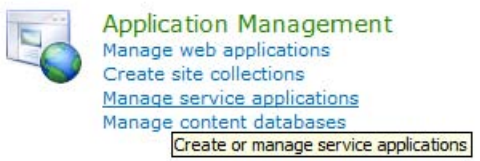
Configure the Visio services cache settings

By default, the Visio Graphics service will cache the web drawings it renders for web clients for a number of minutes based on the service's cache settings. To test delegation we will configure the service to not cache drawings to easily check data refresh in a Visio web drawing.

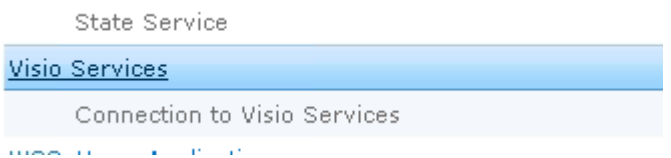
Note:

Disabling the rendering cache is not recommended for production environments. Remember to re-enable the cache once you have completed testing delegation in Visio

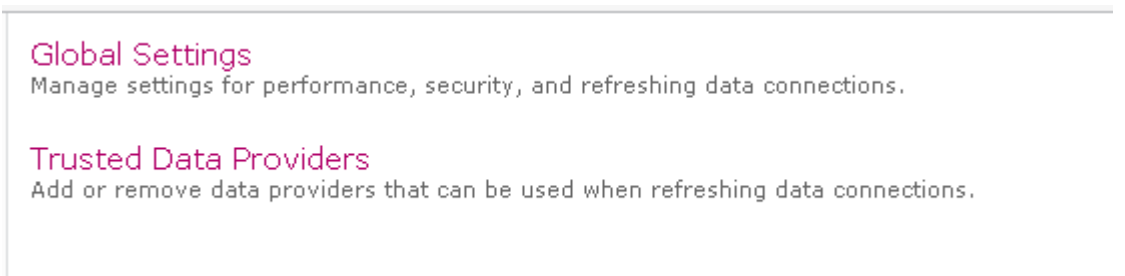
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



3. Select the Visio Graphics Service application created in the previous step.



4. Select **Global Settings**".



5. In the **Minimum Cache Age** setting, set the cache to 0 (no cache).

Note:

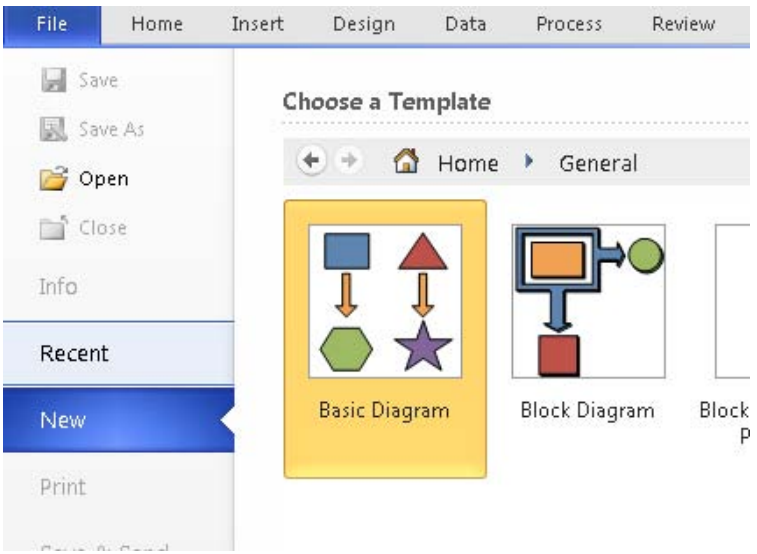
Setting the minimum cache age to 0 is for testing purposes only and should not be used in a production environment.

Create document library to host a test Visio Web Drawing

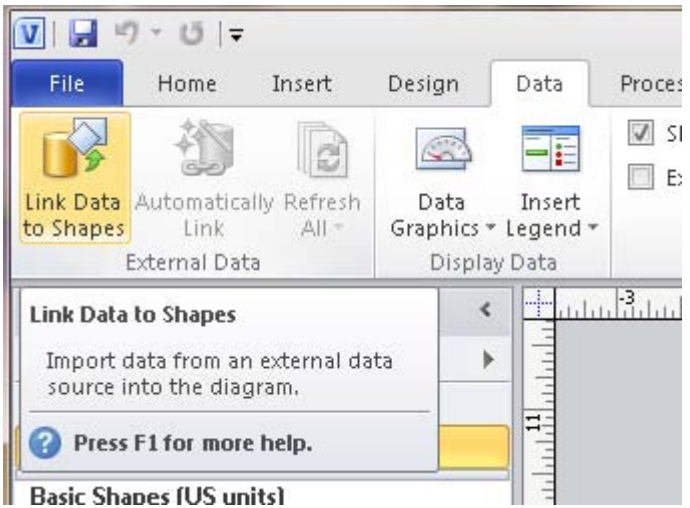
Navigate to the portal application (http://portal). Create a new document library to host a test Visio workbook. The default document library

Create a test Visio web drawing with SQL Server data-connected shapes

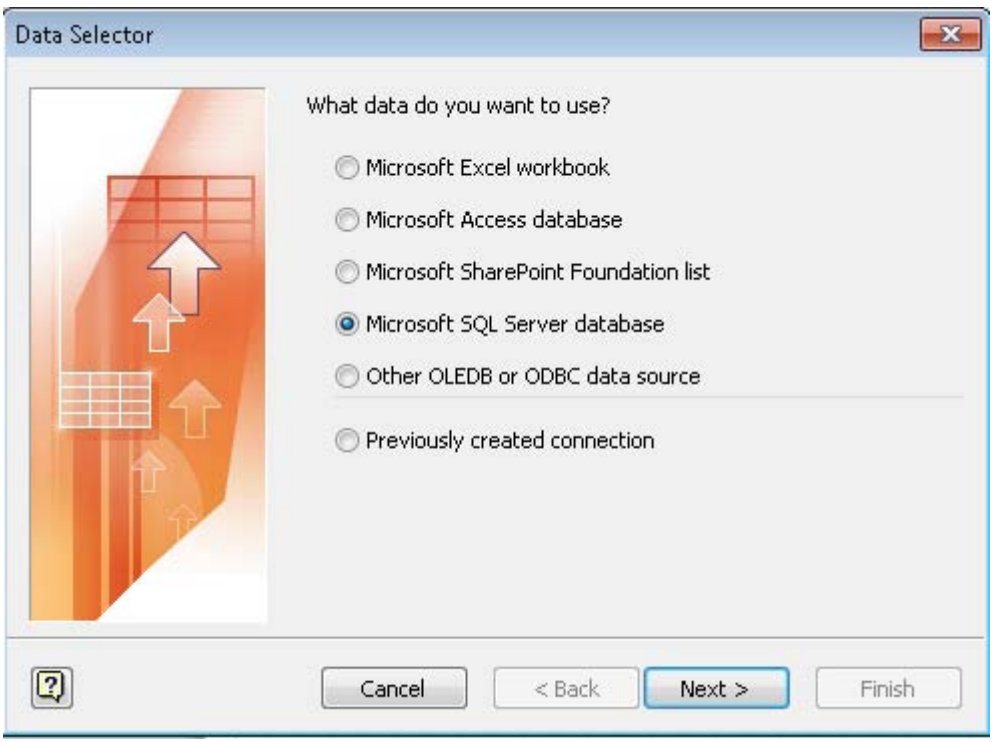
1. Start Visio 2010.
2. Create a new **Basic Diagram** in the General section under **Home**.



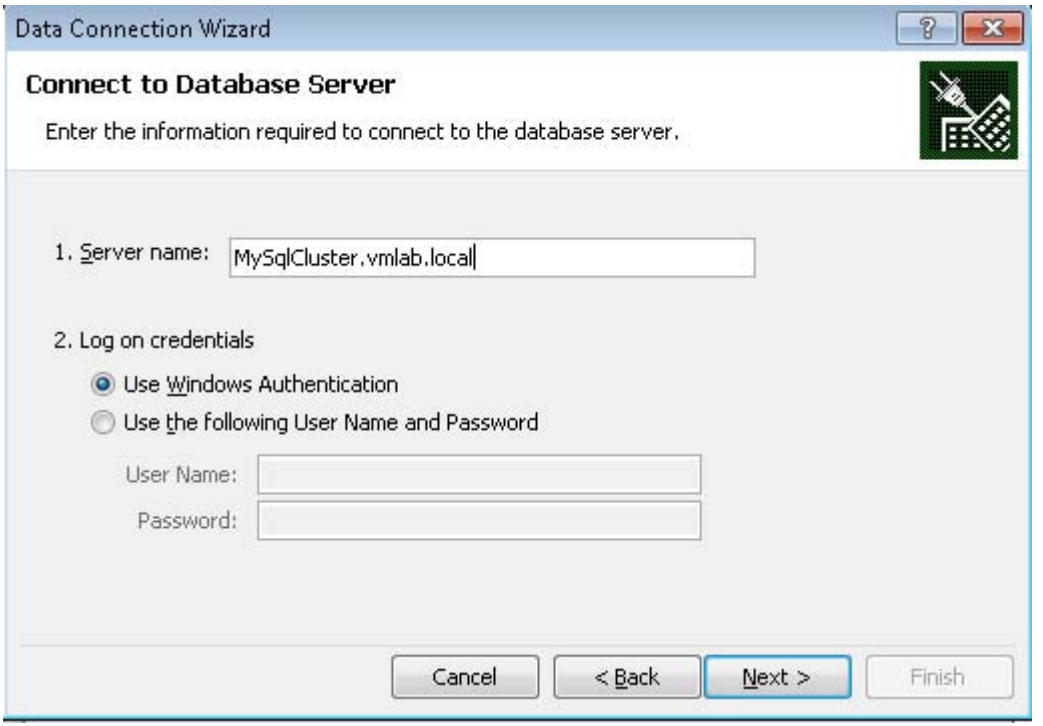
3. On the **Data** Ribbon Tab, select **Link Data to Shapes**.



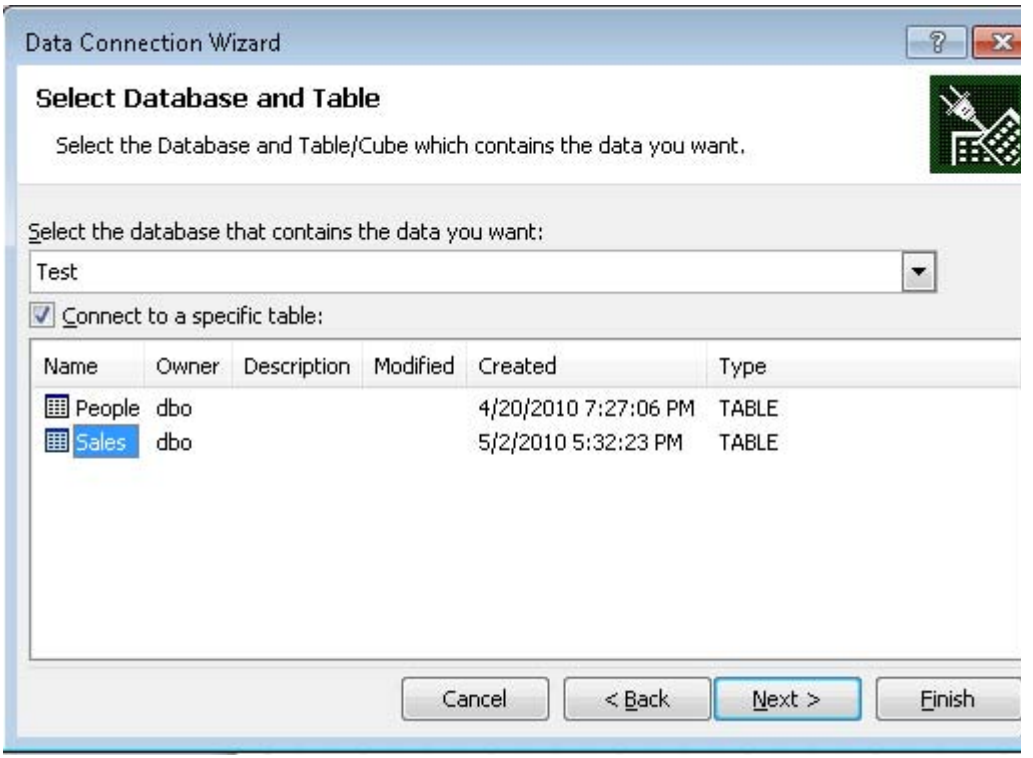
4. In the data selector dialog box, select **Microsoft SQL Server database**.



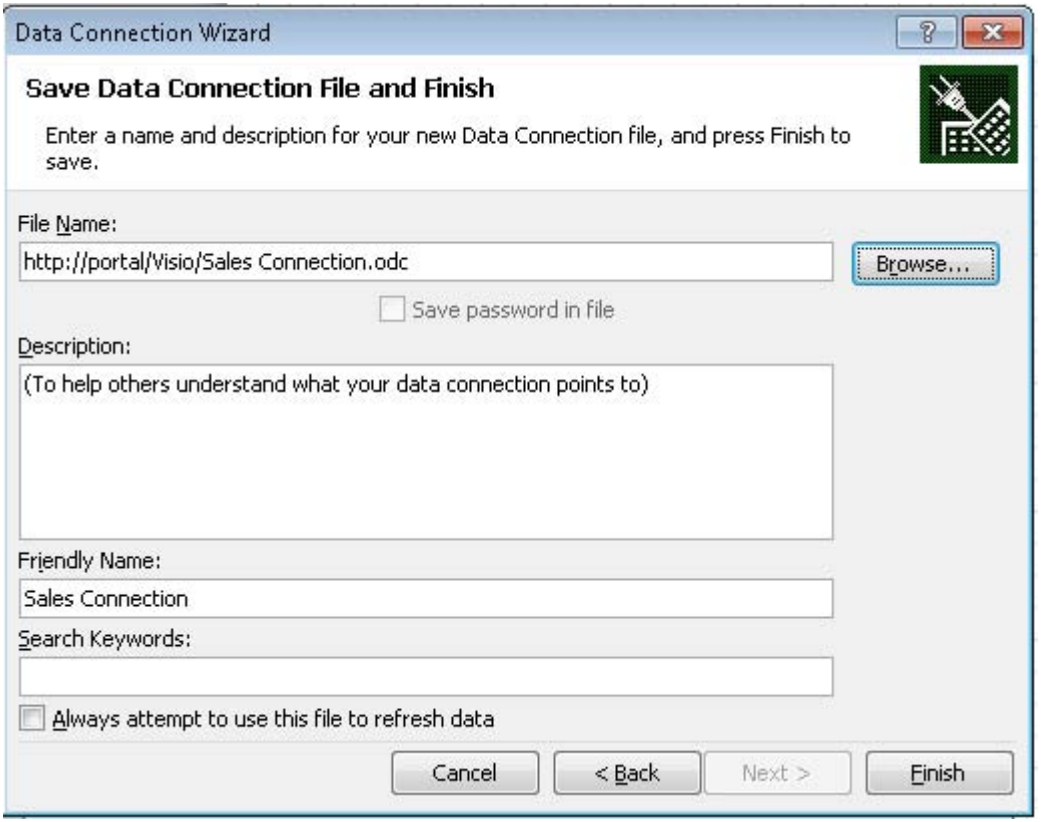
5. Specify the SQL Server cluster created in Scenario 2 and select **Windows Authentication**.



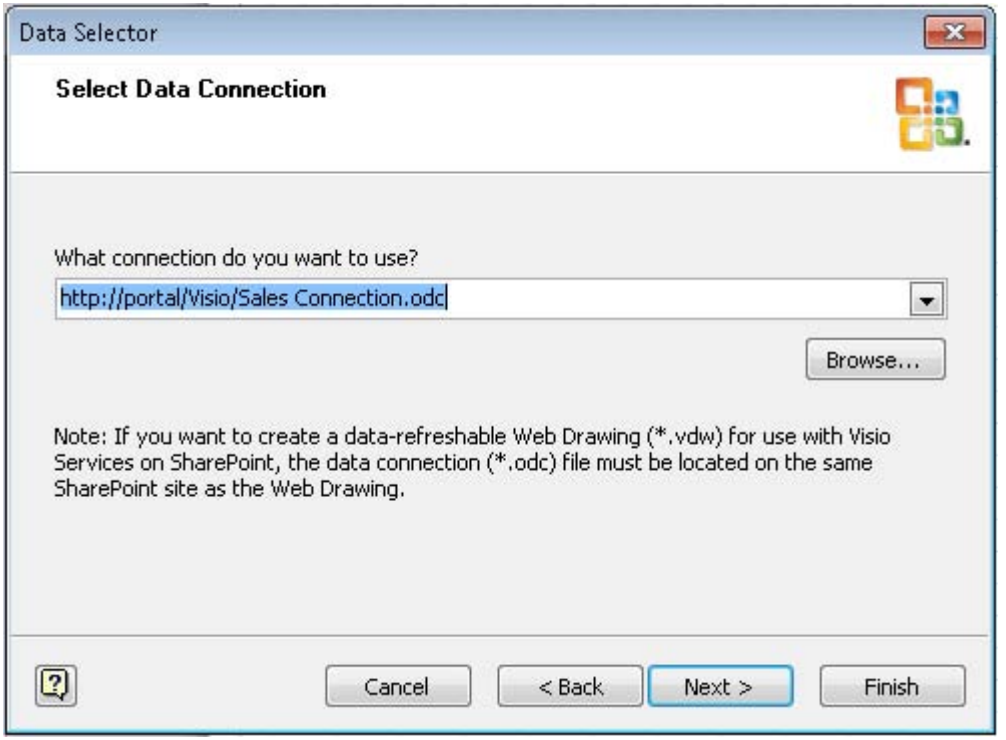
6. Select the **Test** database and the **Sales** Table.



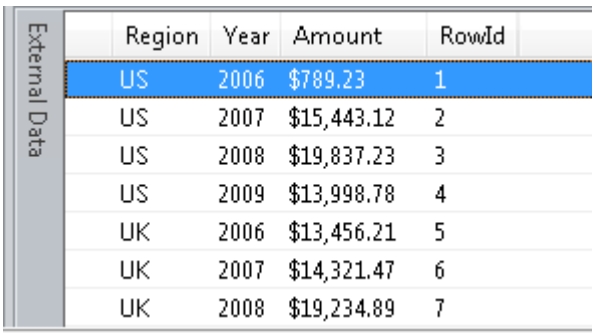
7. Specify a friendly name for the connection and save the connection to the document library created in the previous step.



8. In the **Data Selector** dialog, select the newly created connection and press **Finish**.

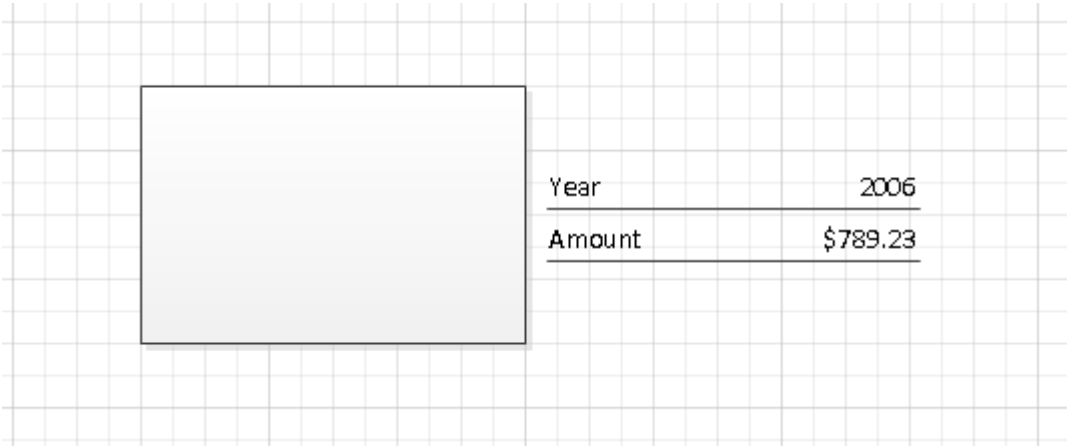


You should now see the external data window at the bottom of the drawing window with the sample data that was created earlier.



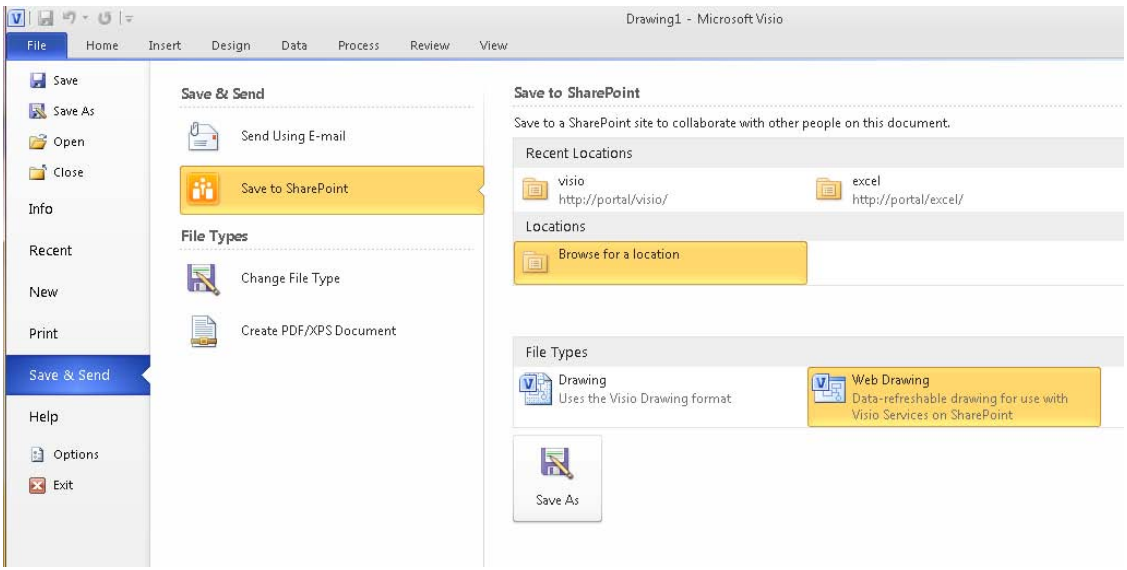
Region	Year	Amount	RowId
US	2006	$789.23	1
US	2007	$15,443.12	2
US	2008	$19,837.23	3
US	2009	$13,998.78	4
UK	2006	$13,456.21	5
UK	2007	$14,321.47	6
UK	2008	$19,234.89	7

9. Drag the first data row onto the drawing surface. This will create a new shape that is linked to the data row. Note that the test drawing is meant to test delegation and is not meant to demonstrate how to create a fully functioning, production ready web drawing.

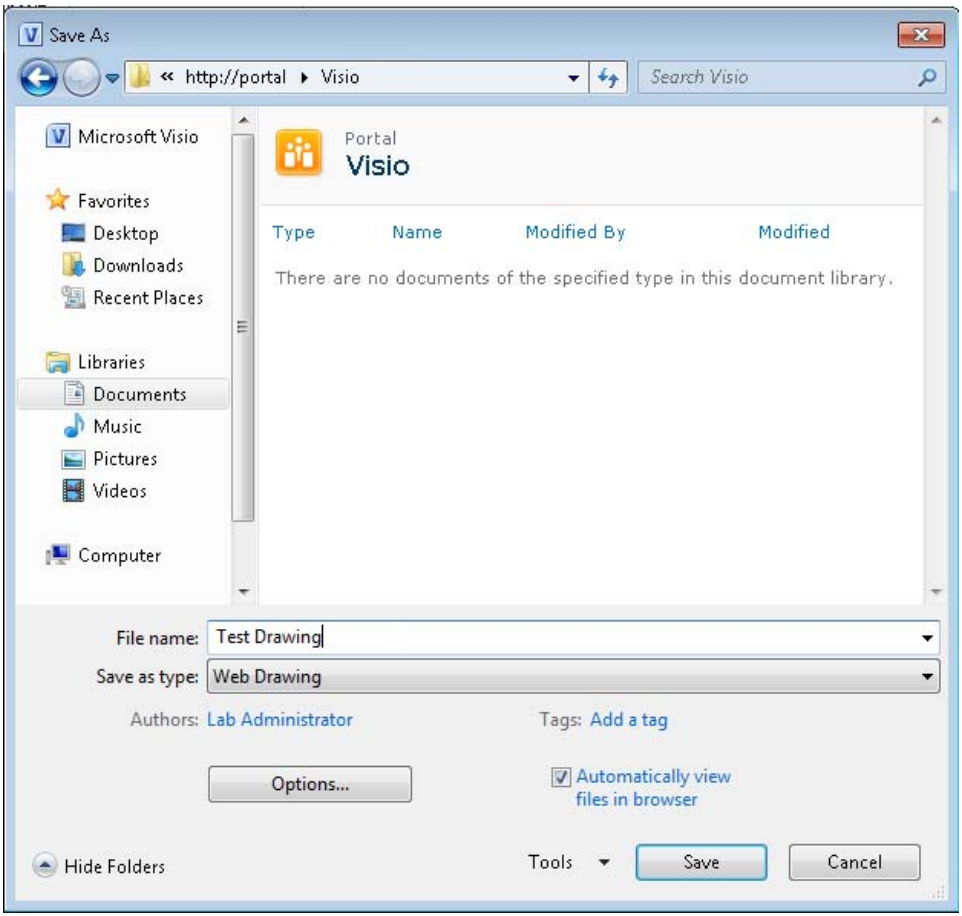


Publish the Visio drawing to SharePoint Server and refresh the data connection

1. Publish the drawing to the test SharePoint document library. On the **File** tab click **Save and Send**, **Save to SharePoint**, **Browse for a location**, and then **Web Drawing**.

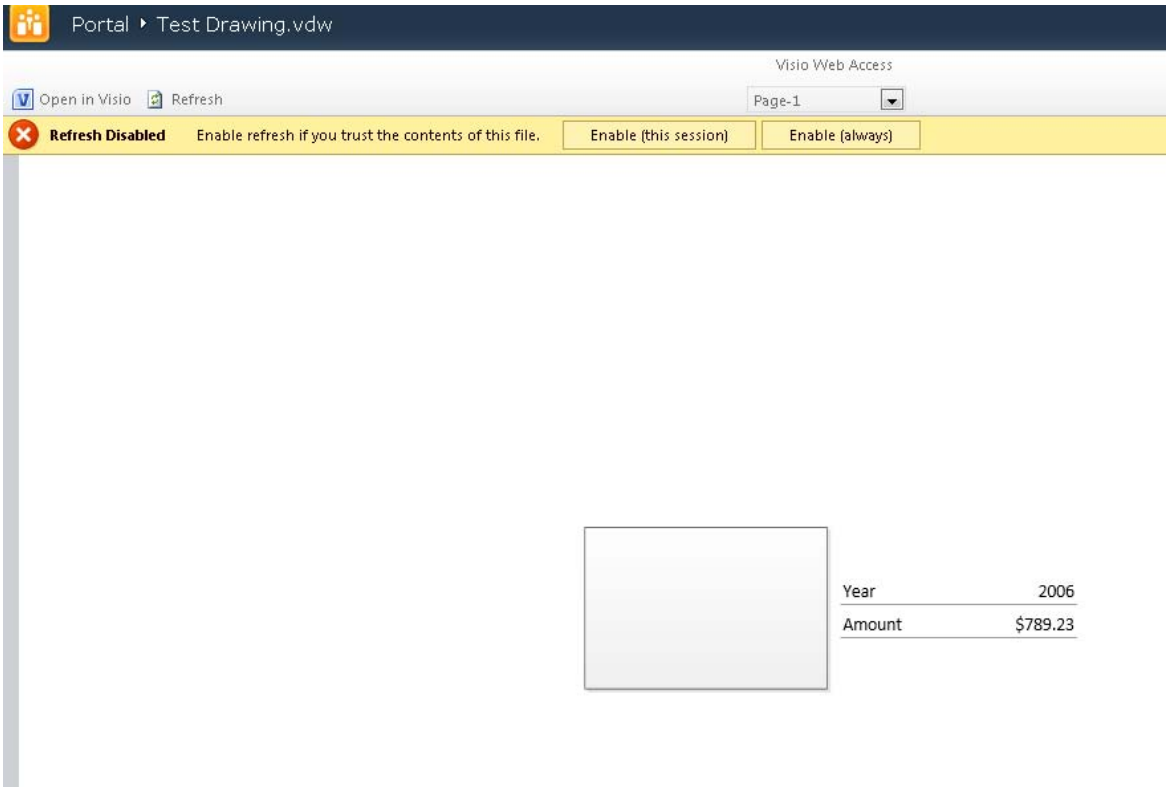


2. Browse to the test document library, specify a name for the test drawing, and then click **Save**.



The drawing opens in the browser.

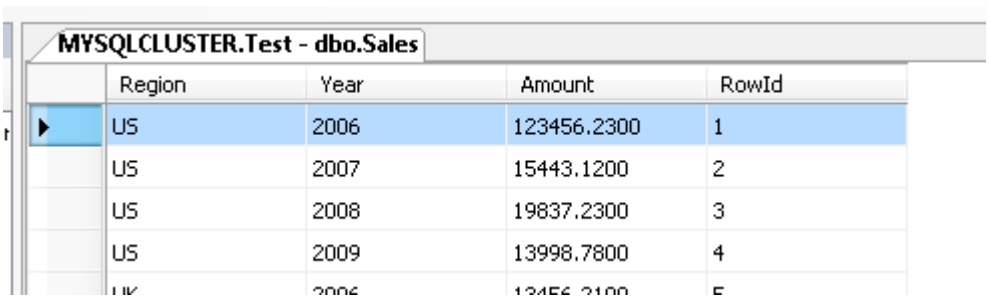
3. In the refresh disabled notification, select **Enable (always)**.



4. The data connection should automatically refresh and no errors should occur.

5. Open SQL Server Management Studio and modify the data row displayed in the web drawing.

6. Refresh the data connection by pressing the **Refresh** button at the top of the drawing window. If delegation is configured correctly you should see your data refresh.



Region	Year	Amount	RowId
US	2006	123456.2300	1
US	2007	15443.1200	2
US	2008	19837.2300	3
US	2009	13998.7800	4

Year	2006
Amount	$123,456.23

Identity delegation for PerformancePoint Services (SharePoint Server 2010)

Published: December 2, 2010

In this scenario, you will add the PerformancePoint Services service application to the SharePoint Server environment and configure Kerberos constrained delegation to allow the service to pull data from an external Analysis Services cube and have the option to pull data from SQL Server.

Note:

If you are installing on Windows Server 2008, you may need to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

Scenario dependencies

To complete this scenario you will need to have completed:

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP (optional)
- Scenario 3: Kerberos Authentication for SQL Server Analysis Services

Configuration checklist

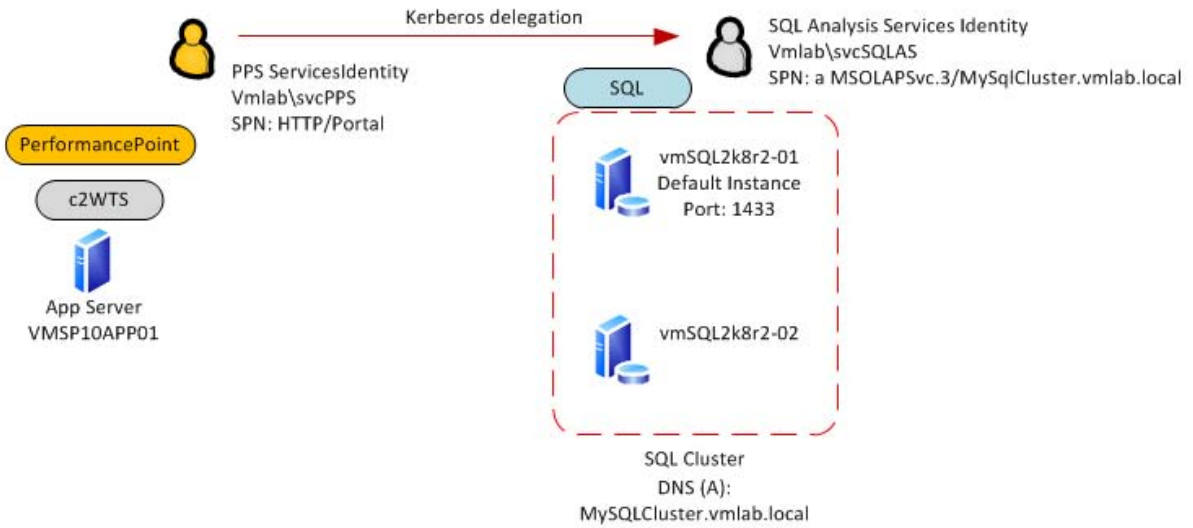
Area of configuration	Description

Area of configuration	Description
Active Directory configuration	Create PerformancePoint Services service account Create an SPN for the service account running the PerformancePoint Service on the Application Server Verify Analysis Services SPN on SQL Server Analysis Services service account, vmlab\svcSQLAS (performed in Scenario 3) and (Optional) verify the SQL Server database engine service account, vmlab\svcSQL(performed in Scenario 2). Configure Kerberos constrained delegation for Claims to Windows Services service account to Analysis Services Configure Kerberos constrained delegation for the PerformancePoint Services service account to Analysis Services
SharePoint Server configuration	Start Claims to Windows Token Service on PerformancePoint Services Servers Start the PerformancePoint Services service instance on the PerformancePoint Services server Create the PerformancePoint Services service application and proxy Check the identity on PerformancePoint application Grant the PerformancePoint Services service account permissions on the web application content database Configure PerformancePoint services trusted file location and authentication settings
Verify PerformancePoint Service constrained	Create document library to host a test dashboard Create a data source that reference an existing SQL Server

Area of configuration	Description
delegation	Analysis Services cube Create a trusted PerformancePoint content list Create test PerformancePoint dashboard Publish dashboard to SharePoint Server

Scenario environment details

Kerberos constrained delegation paths

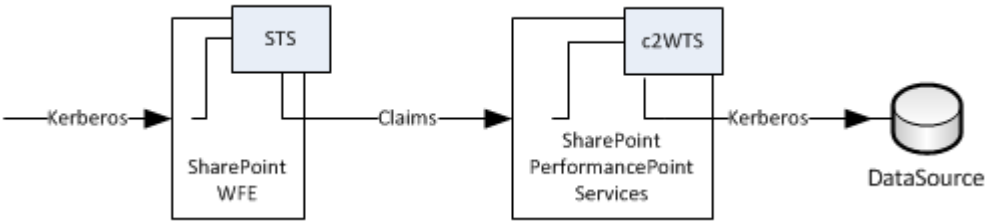


In this scenario we will configure the PerformancePoint Services service account for Kerberos constrained delegation to the SQL Server service.

Note:

In this scenario we will configure the Claims to Windows Token Services (C2WTS) to use a dedicated service account. If you leave the C2WTS configured to use **Local System** you will need to configure constrained delegation on the computer account for the computer running the C2WTS and Excel Services.

SharePoint Server logical authentication



Authentication in this scenario begins with the client authenticating with Kerberos authentication at the web front end. SharePoint Server 2010 will convert the Windows authentication token into a claims token using the local Security Token Service (STS). The PerformancePoint service application will accept the claims token and convert it into a Windows token (Kerberos) using the local Claims to Windows Token Service (C2WTS) that is a part of Windows Identity Framework (WIF). The PerformancePoint service application will then use the client's Kerberos ticket to authenticate with the backend DataSource.

Step-by-step Configuration instructions

Active Directory configuration

Create PerformancePoint Services service account

As a best practice PerformancePoint Services should run under its own domain identity. To configure the PerformancePoint Service Application, an Active Directory account must be created and registered as a managed account in SharePoint Server 2010. For more information see Managed Accounts in SharePoint 2010. In this example the following account is created and registered later in this scenario:

SharePoint Server service	IIS App Pool Identity
PerformancePoint Services	vmlab\svcPPS

* NOTE: You can optionally reuse a single domain account for multiple services. This configuration is not covered in the following sections.

Create an SPN for the Service Account that is running the PerformancePoint service on the Application Server

The Active Directory Users and Computers MMC snap-in is typically used to configure Kerberos delegation. To configure the delegation settings within the snap-in, the Active Directory object being configured must have a service principal name applied; otherwise the **delegation** tab for the object will not be visible in the object's properties dialog. Although PerformancePoint Services does not require a SPN to function, we will configure one for this purpose. Note that if the service account already has an SPN applied (in the case of sharing accounts across services) this step is not required.

On the command line, run the following command:

```
SETSPN -S SP/PPSvmlab\svcPPS
```

Note:

The SPN is not a valid SPN. It is applied to the specified service account to reveal the delegation options in the AD users and computers add-in. There are other supported ways of specifying the delegation settings (specifically the msDS-AllowedToDelegateTo AD attribute) but this topic will not be covered in this document.

Verify Analysis Services SPN on SQL Server Analysis Services service account, vmlab\svcSQLAS(performed in Scenario 3) AND (Optional) Verify the SQL Server database engine service account, vmlab\svcSQL(performed in Scenario 2)

Verify that the SPN for the SQL Analysis Services account (vmlab\svcSQLAS) exists with the following SetSPN command:

```
SetSPN -L vmlab\svcSQLAS
```

You should see the following:

```
MSOLAPSvc.3/MySqlCluster
```

Verify the SPN for the SQL Server service account (vmlab\svcSQL) exists with the following SetSPN command:

```
SetSPN -L vmlab\svcSQL
```

You should see the following:

```
MSSQLSVC/MySqlCluster
```

Configure Kerberos constrained delegation from the PerformancePoint Services Service account to the SSAS Service and optionally for SQL Server service

To allow PerformancePoint services to delegate the client's identity, Kerberos constrained delegation must be configured. You must also configure constrained delegation with protocol transition for the conversion of claims token to Windows token via the WIF C2WTS.

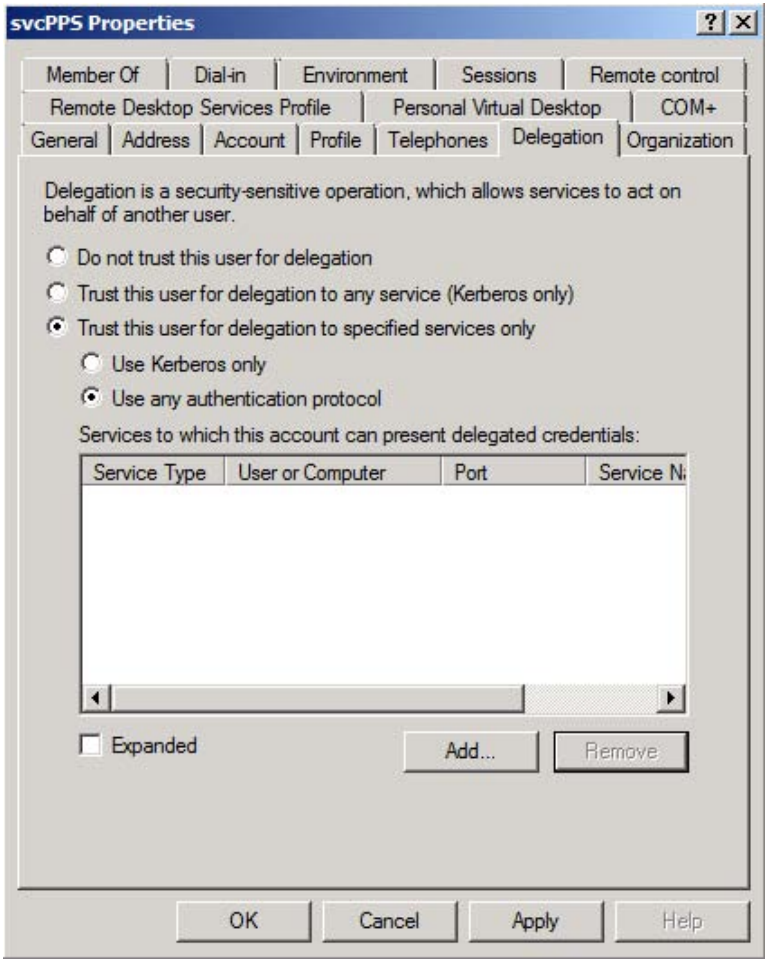
Each server running PerformancePoint services must be trusted to delegate credentials to each back-end service with which PerformancePoint will authenticate. In addition, the PerformancePoint services service account must also be configured to allow delegation to the same back-end services. Notice also that HTTP/Portal and HTTP/Portal.vmlab.local are configured to delegate in order to include a SharePoint list as an optional data source for your PerformancePoint dashboard.

In our example the following delegation paths are defined:

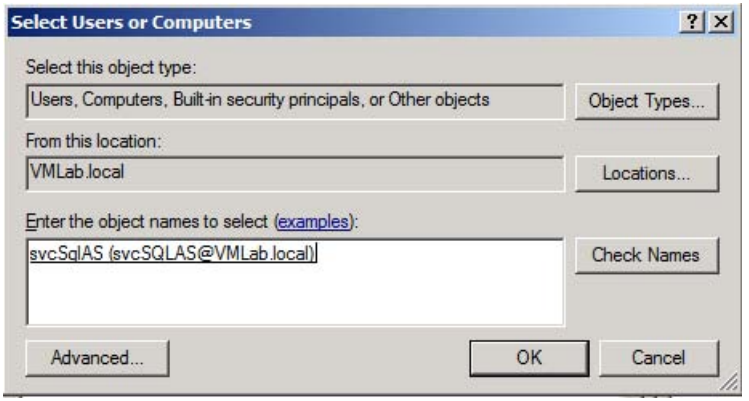
Principal Type	Principal Name
User	Vmlab\svcC2WTS
User	Vmlab\svcPPS

To configure constrained delegation

1. Open the Active Directory Object’s properties in Active Directory Users and Computers.
2. Navigate to the **Delegation** tab.



3. Select **Trust this computer for delegation to specified services only**.
4. Select **Use any authentication protocol**.
5. Click the add button to select the service principal.
6. Select **User and Computers**.



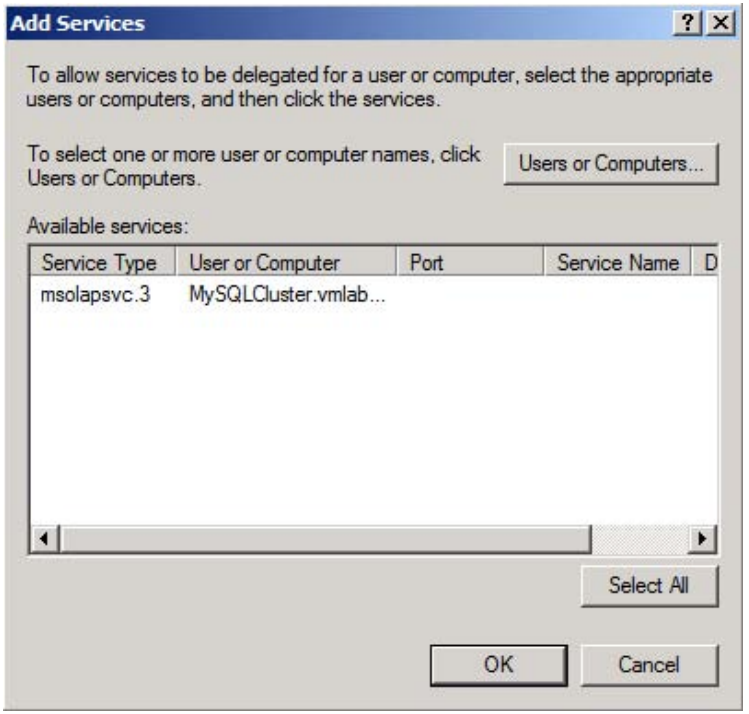
7. Select the service account running the service you wish to delegate to (SQL Server, SQL Server Analysis Services, or both).

   Note:

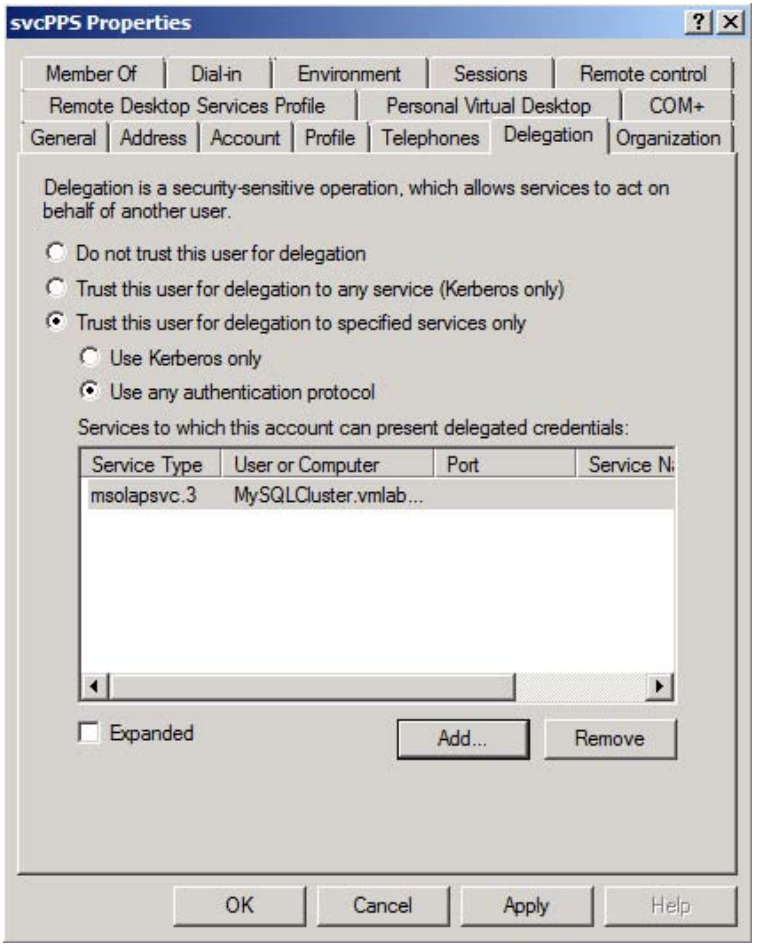
   The service account selected must have an SPN applied to it. In our example, the SPN for this account was configured in a previous scenario. See the Kerberos Authentication for SQL OLTP and Kerberos Authentication for SQL Analysis Services sections of this document.

8. Click **OK**.

9. Select the SPNs you would like to delegate to, and then click **OK**.



10. You should now see the selected SPNS in the **services to which this account can presented delegated credentials** list.



11. Repeat these steps for each delegation path defined in the beginning of this section.

SharePoint Server configuration

Configure and Start the Claims to Windows Token Service on PerformancePoint Services Servers

The Claims to Windows Token Service (C2WTS) is a component of the Windows Identity Foundation (WIF) which is responsible for converting user claim tokens to Windows tokens. PerformancePoint Services uses the C2WTS to convert the user's claims token into a windows token when the services needs to delegate credentials to a back-end system which uses Windows authentication. WIF is deployed with SharePoint Server 2010 and the C2WTS can be started from Central Administration.

Each PerformancePoint Services Application server must run the C2WTS locally. The C2WTS does not open any ports and cannot be accessed by a remote caller. Further, the C2WTS service configuration file must be configured to specifically trust the local calling client identity.

As a recommended practice you should run the C2WTS using a dedicated service account and not as Local System (the default configuration). The C2WTS service account requires special local permissions on each server that the service runs on.Be sure to configure these permissions when you choose to run the C2WTS with a domain account. To ensure the C2WTS account picks up the needed privileges, reboot the serverafter you have configured the C2WTS.

*NOTE: If you choose to configure the C2WTS as local system you do not need to configure any additional local privileges.

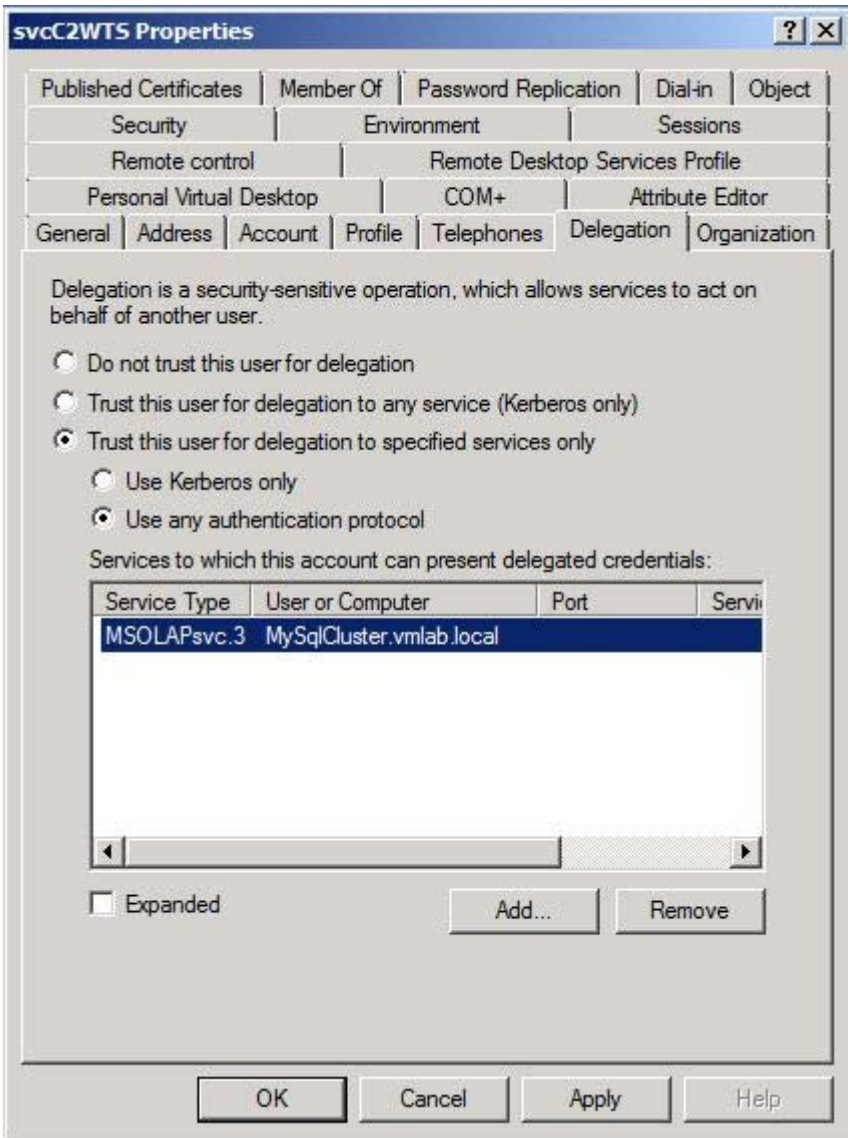
To start the C2WTS

1. Create a service account in Active Directory to run the service under. In this example we created vmlab\svcC2WTS.

2. Add an arbitrary Service Principal Name (SPN) to the service account to expose the delegation options for this account in Active Directory Users and Computers. The SPN can be any format because we do not authenticate to the C2WTS using Kerberos authentication. It is recommended to not use an HTTP SPN to avoid potentially creating duplicate SPNs in your environment. In our example we registered SP/C2WTS to the vmlab\svcC2WTS using the following command:

```
SetSPN -S SP/C2WTS vmlab\svcC2WTS
```

3. Configure Kerberos constrained delegation on the C2WTS services account. In this scenario we delegate credentials to the SQL Server service that is running with the MSOLAPsvc.3/MySqlCluster.vmlab.local service principal name.



4. Next, configure the required local server permissions the C2WTS requires. You have to configure these permissions on each server the C2WTS runs on. In our example this is VMSP10APP01. Log onto the server and give the C2WTS the following permissions:

    a) Add the service account to the local Administrators Groups.

b) In local security policy (secpol.msc) under user rights assignment give the service account the following permissions:

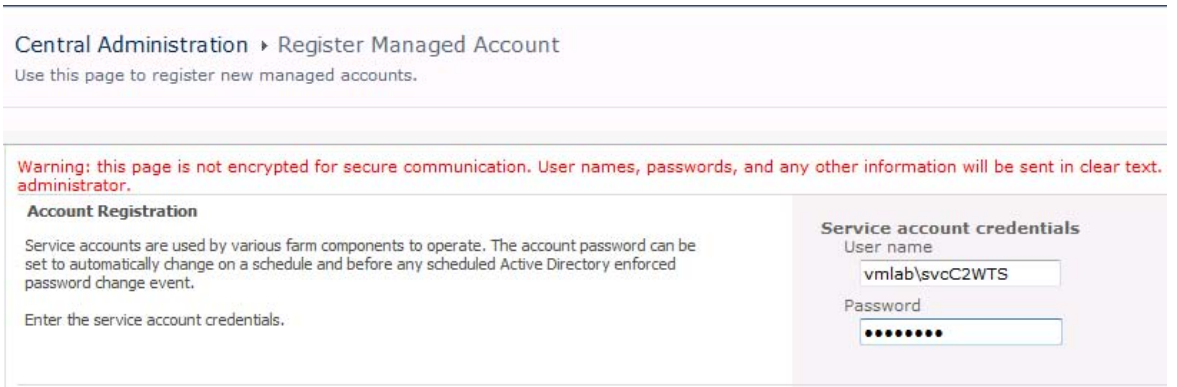
i. **Act as part of the operating system**

ii. **Impersonate a client after authentication**

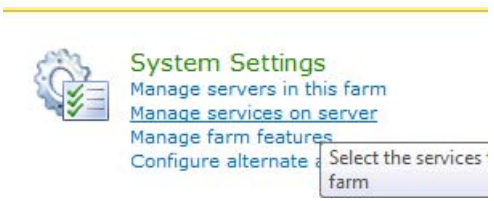
iii. **Log on as a service**

5. Open Central Administration.

6. In the **Security** section, under **Configure Managed Service Accounts**, register the C2WTS service account as a managed account.

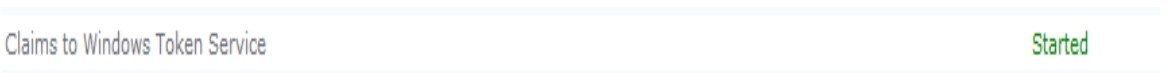


7. Under services, select **Manage services on server**.

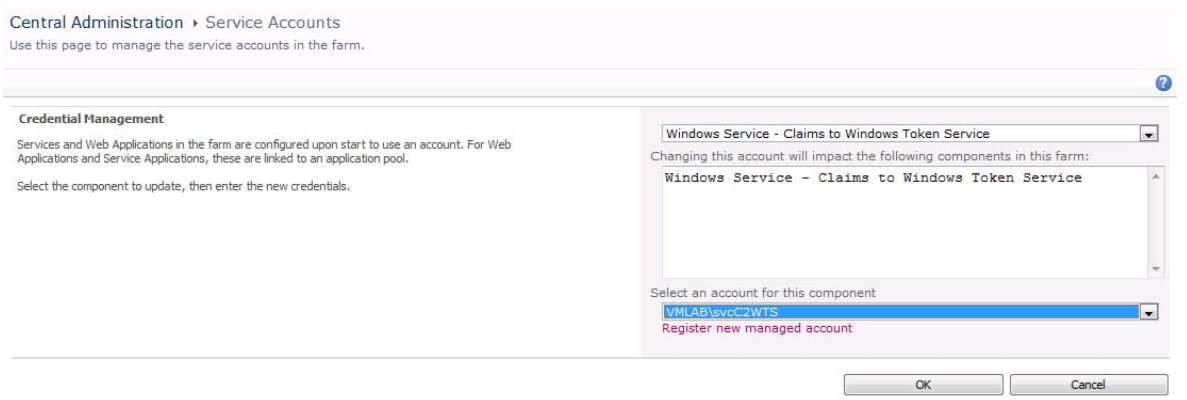


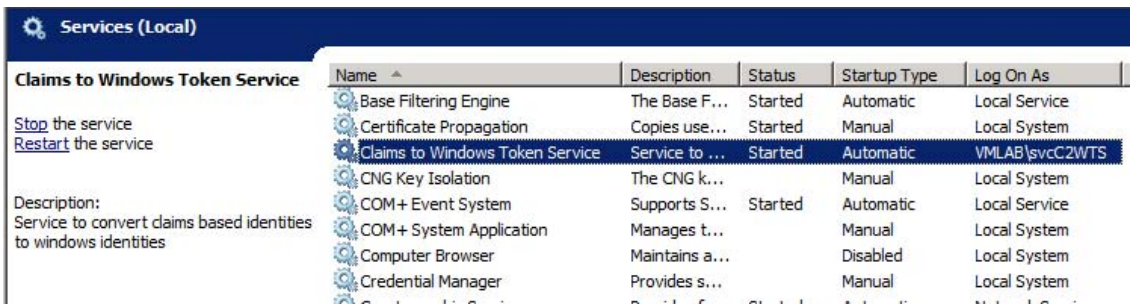
8. In the server selection box in the upper right hand corner select the server(s) running PerformancePoint services. In this example it is VMSP10APP01.

9. Find the **Claims to Windows Token Service** and start it:



10. Go to **Manage Service Accounts** in the **Security** section. Change the identity of the C2WTS to the new managed acount.





Note:

If the C2WTS was already running before configuring the dedicated service account, or if you need to changes the permissions of the service account after the C2WTS is running you must restart the C2WTS from the services console.

In addition, if you experience issues with the C2WTS after restarting the service it may also be required to reset the IIS application pools that communicate with the C2WTS.

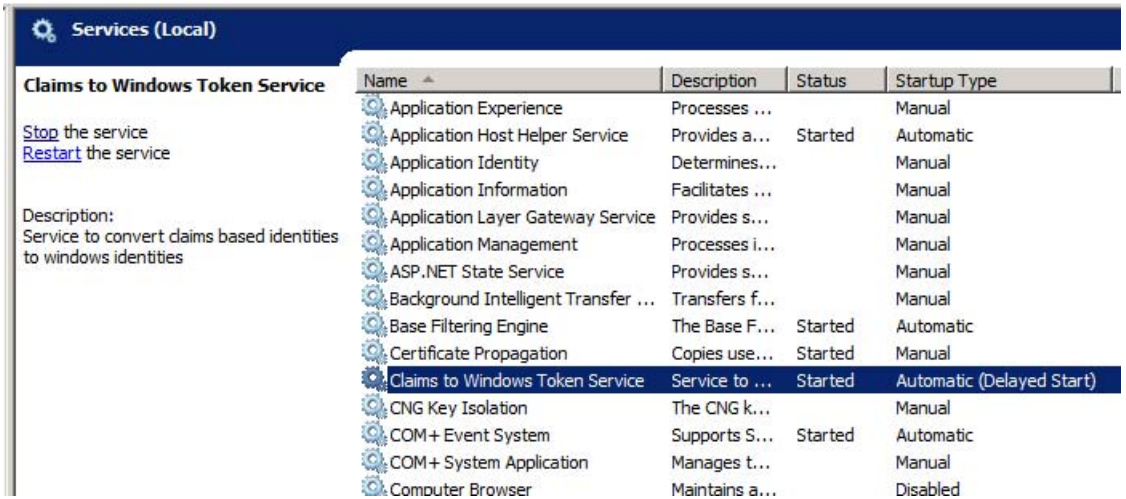
Add startup dependencies to the WIF C2WTS service

There is a known issue with the C2WTS where it may not automatically startup successfully on system reboot. A workaround to the issue is to configure a service dependency on the Cryptographic Services service:

1. Open the command-prompt window.
2. Type: **scconfig c2wts depend= CryptSvc**

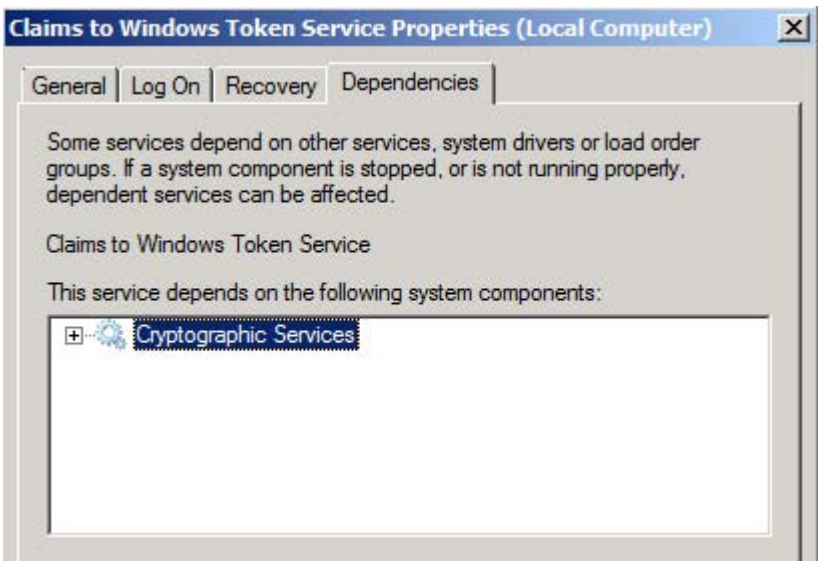
```
C:\>sc config "c2wts" depend= CryptSvc
[SC] ChangeServiceConfig SUCCESS
```

3. Find the Claims to Windows Token Service in the services console.



4. Open the properties for the service.

5. Check the **Dependencies** tab. Make sure **Cryptographic Services** is listed:



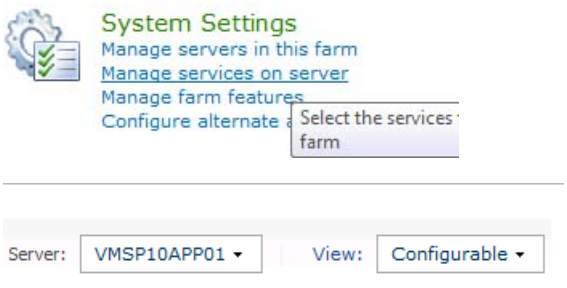
6. Click **OK**.

7. Reboot the server. Make sure that the C2WTS has started once the computer reboots.

Start the PerformancePoint Services service instance on the PerformancePoint Services server

Before creating a PerformancePoint Services service application, start the PerformancePoint services serve service on the designated Farm servers. To learn more about PerformancePoint Services configuration, see PerformancePoint Services administration on Microsoft TechNet.

1. Open Central Administration.
2. Under services, select **Manage services on server**.
3. In the server selection box in the upper right hand corner select the server(s) running PerformancePoint services. In this example it is VMSP10APP01:

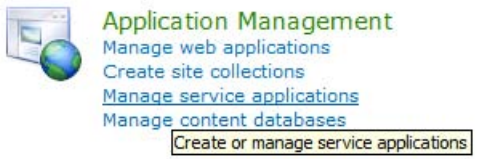


4. Start the **PerformancePoint Services** service.

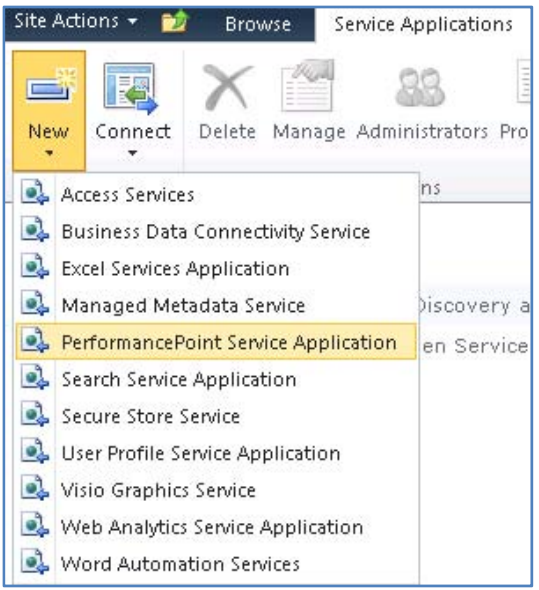
Create the PerformancePoint Services service application and proxy

Next configure a new PerformancePoint Services service application and application proxy to allow web applications to consume PerformancePoint Services:

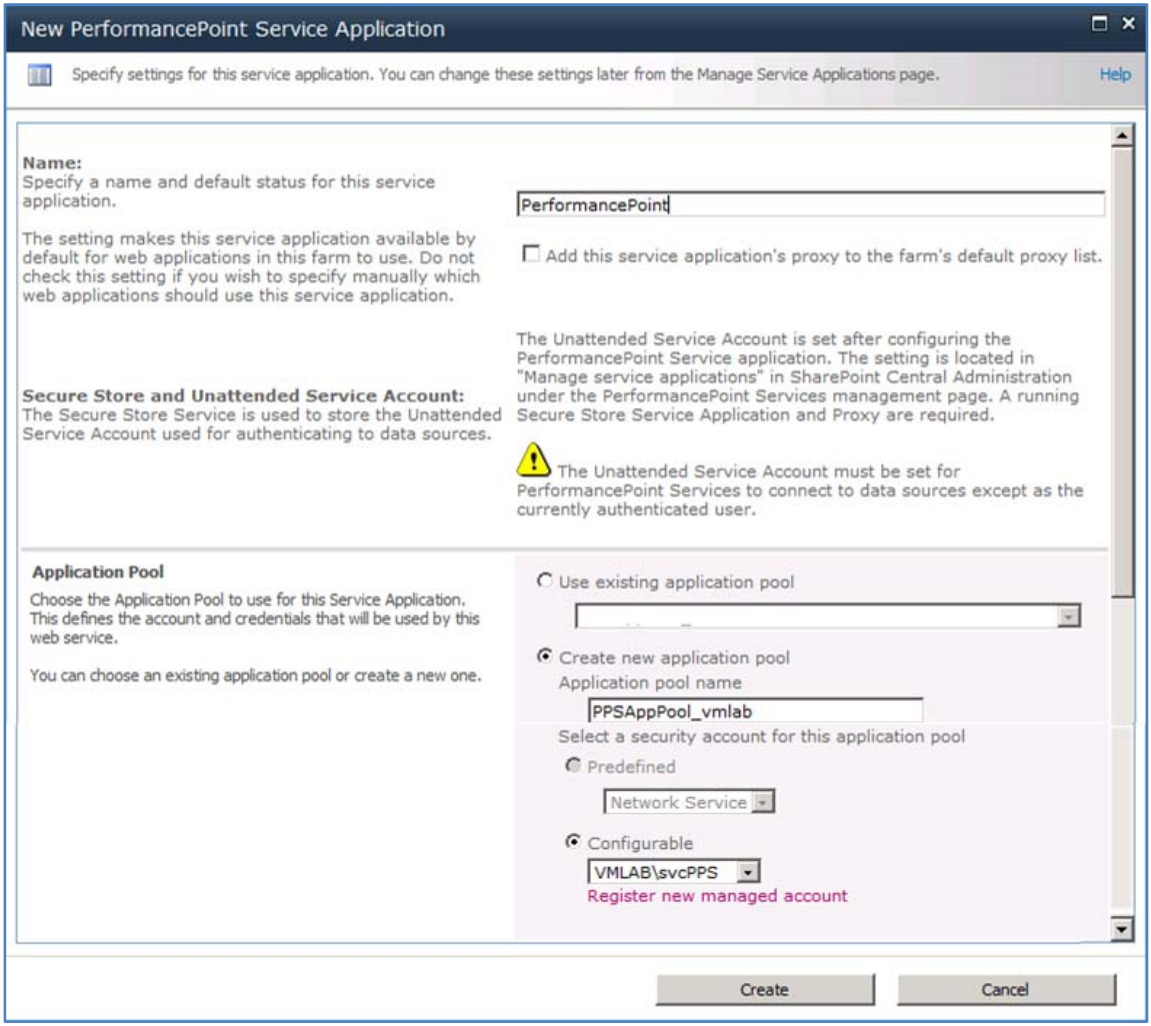
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



3. Select **New**, and then click **PerformancePoint Services Application**.



4. Configure the new service application. Be sure to select the correct service account or create a new managed account if you did not perform this step previously.



Note:

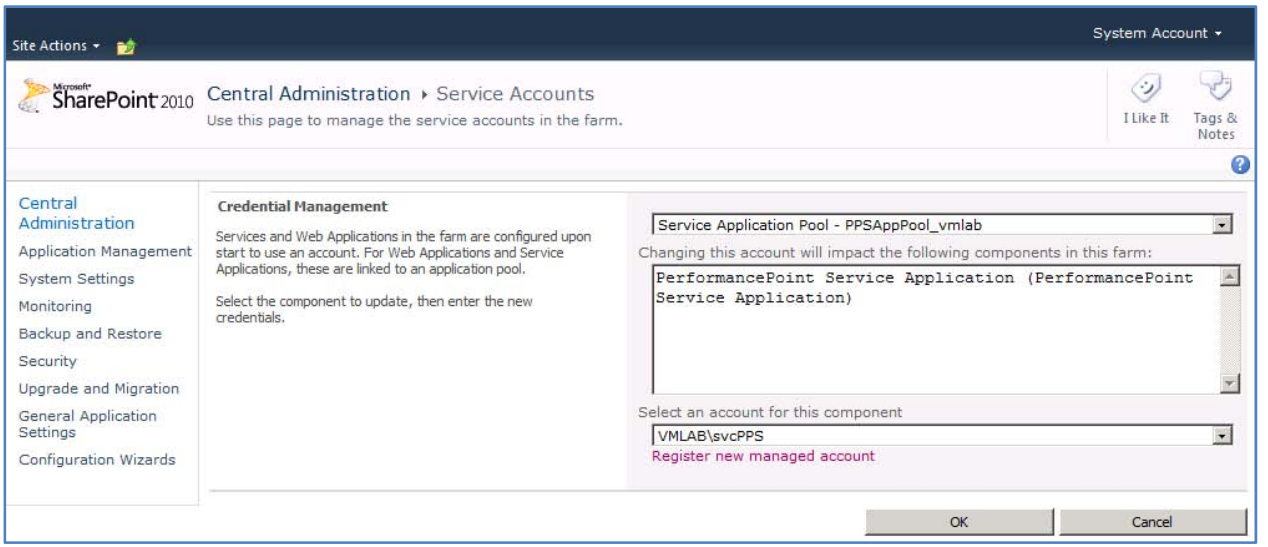
Configuring the Unattended Services Account is optional in this scenario and only used if you want to also test NTLM authentication.

You can create and register a new service account for an existing application pool dedicated for PerformancePoint Services before this step or when you create the new PerformancePoint Service. To associate the service account with an existing application pool dedicated to PerformancePoint or verify an existing account, do the following.

1. Navigate to SharePoint Central Administration. Find **Configure managed accounts** in the **Security** section.

2. Select the drop-down box and select the application pool.

3. Select the Active Directory account.



Grant the PerformancePoint Services service account permissions on the web application content database

A required step in configuring SharePoint Server 2010 Office Web Applications is allowing the web application's service account access to the content databases for a given web application. In this example, we will grant the PerformancePoint Services account access to the "portal" web application's content database by using Windows PowerShell.

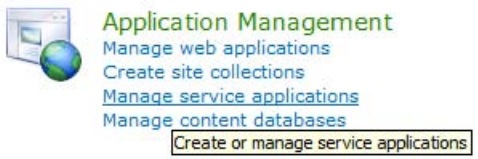
Run the following command from the SharePoint 2010 Management Shell:

```
$w = Get-SPWebApplication -Identity http://portal

$w.GrantAccessToProcessIdentity("vmlab\svcPPS")
```

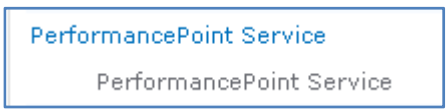
Configure PerformancePoint Services trusted file location and authentication settings

Once the PerformancePoint Services application is created, you must configure the properties on the new service application to specify a trusted host location and authentication settings.

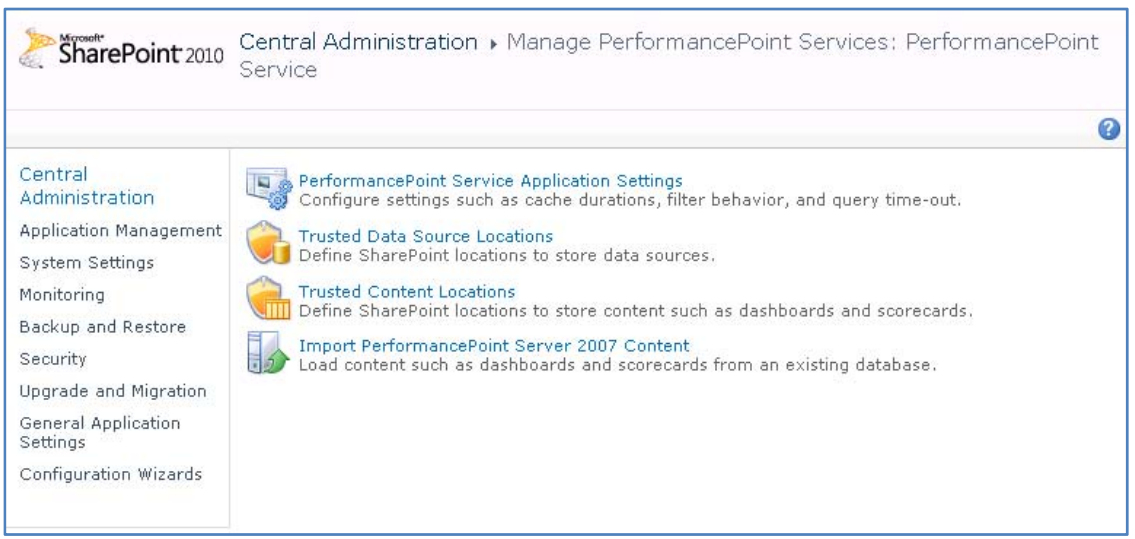
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



3. Click the link for the new Service Application, **PerformancePoint Services** and click the **Manage** button in the ribbon.

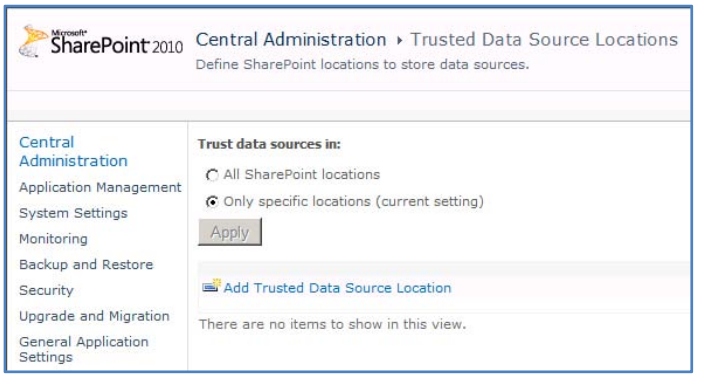


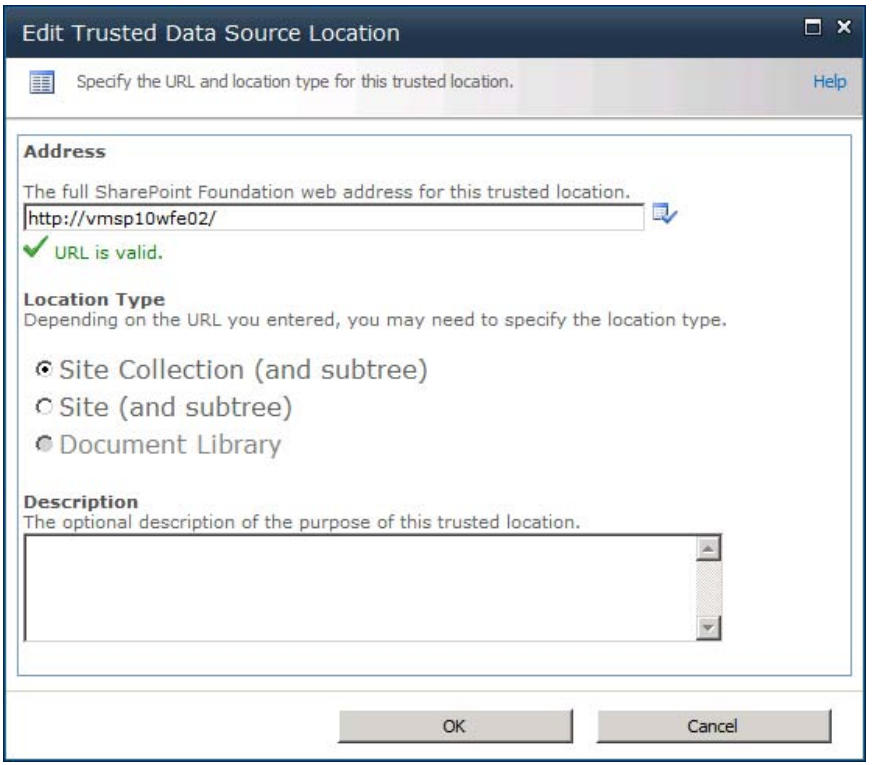
4. In the PerformancePoint services management screen, click **Trusted Data Source Locations**.



5. Select the **Only specific locations** option and click **Add Trusted Data Source Location**.

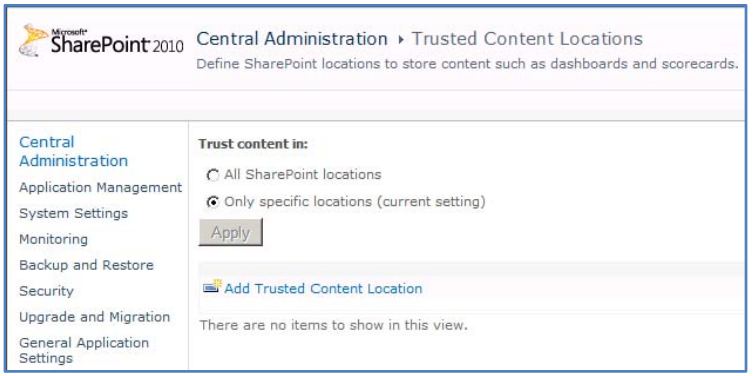
6. Type the URL of the location, select the **Site Collection (and subtree)** option, and then click **OK**.

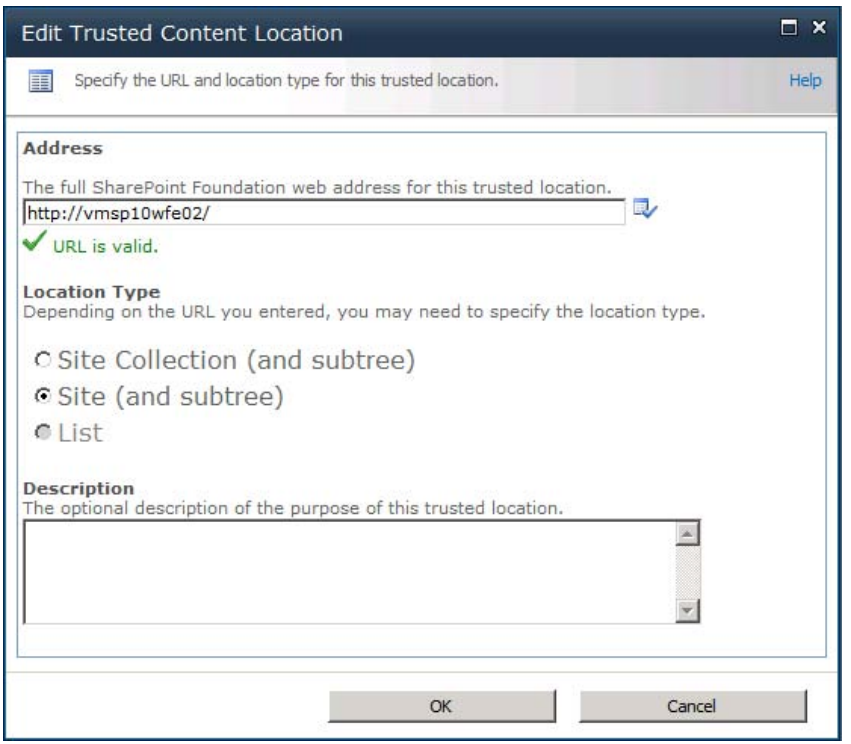




7. Select the **Only specific locations** option and click **Add Trusted Data Source Location**.

8. Type the URL of the location, select the **Site (and subtree)** option, and then click **OK**.





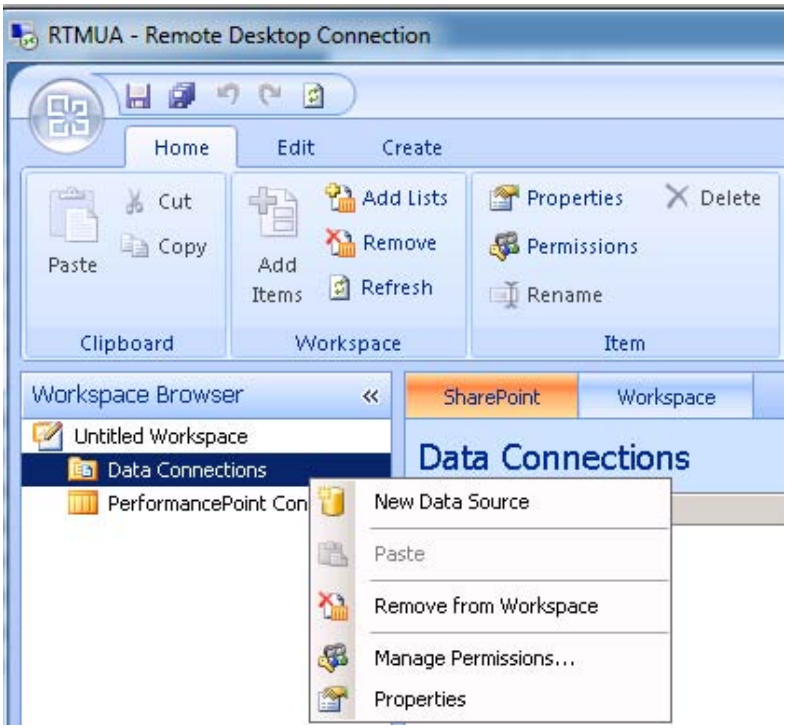
Verify PerformancePoint Service Constrained Delegation

Note:In largerenvironments with multiple Active Directory servers, you may need to wait for Active Directory replication to finishbeforeyou verify your configuration.

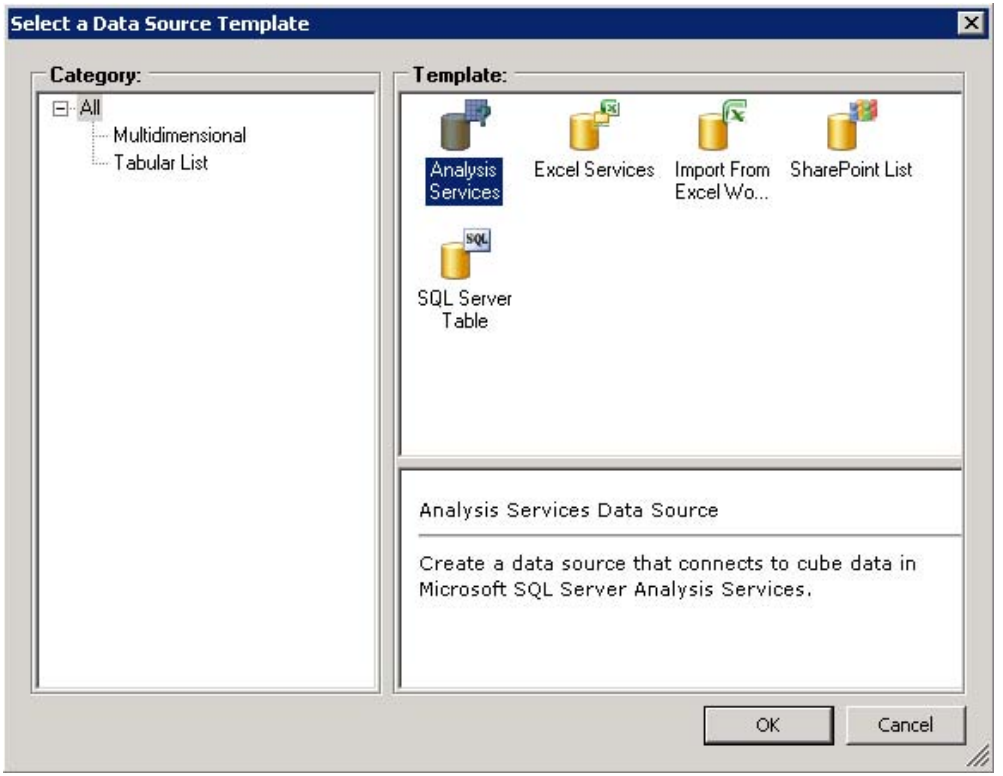
Create test PerformancePoint dashboard with a SQL Server AS data connection

Next, open PerformancePoint Dashboard Designer and create an Analysis Services data connection.

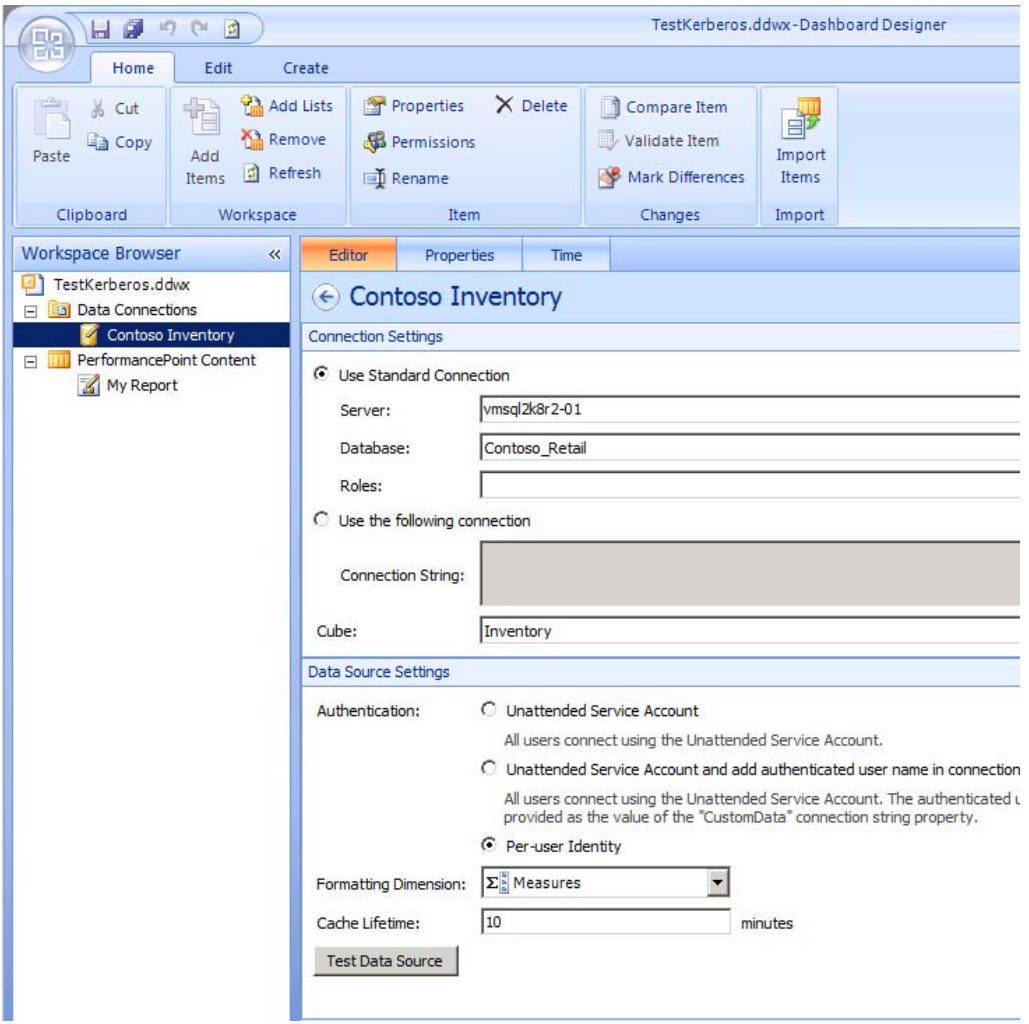
1. Open PerformancePoint Dashboard Designer and right-click data source to create a connection.



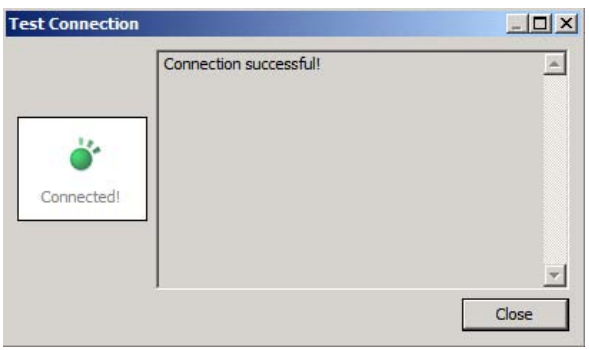
2. Select **Analysis Services**.



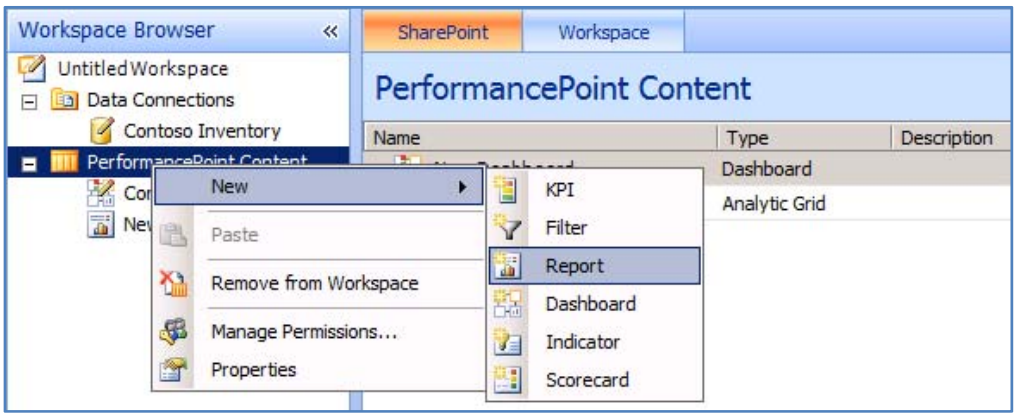
3. Specify the server, database, and cube and select **Per-user Identity**.



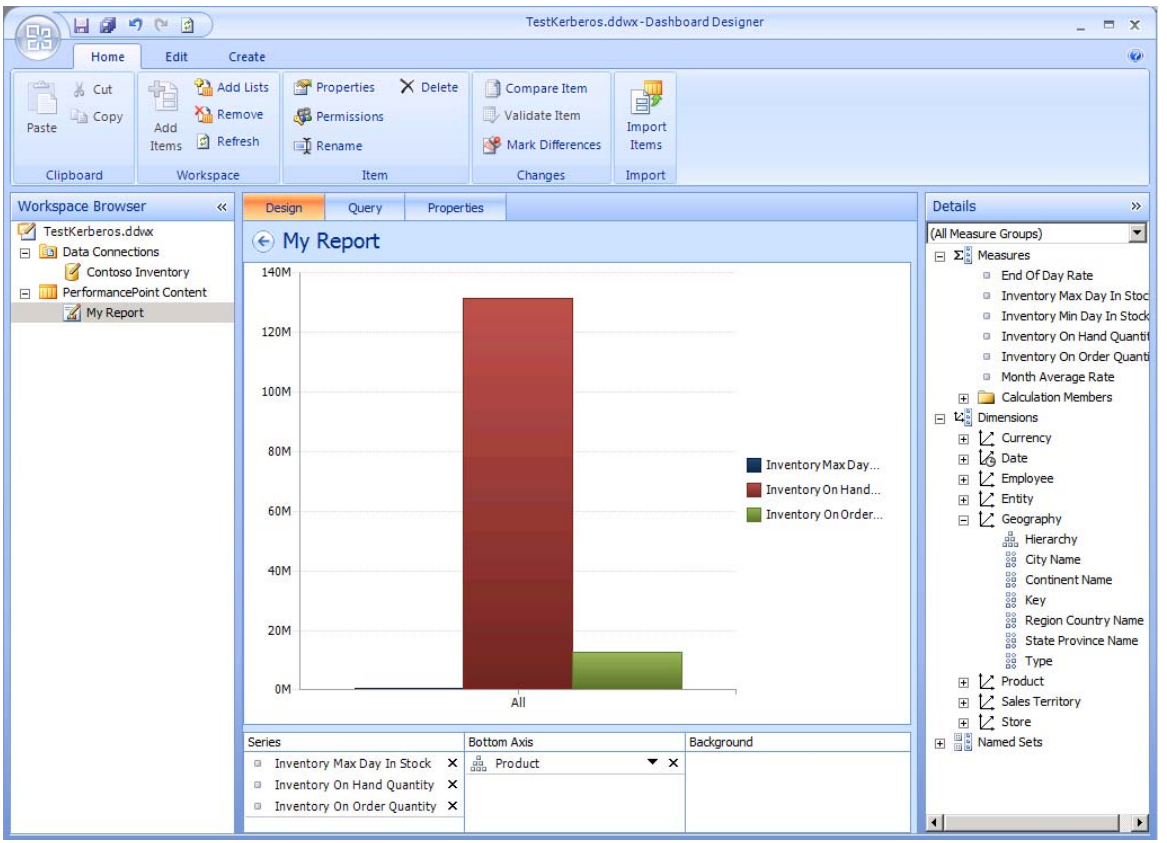
4. Click **Test Data Source** to test the connection.



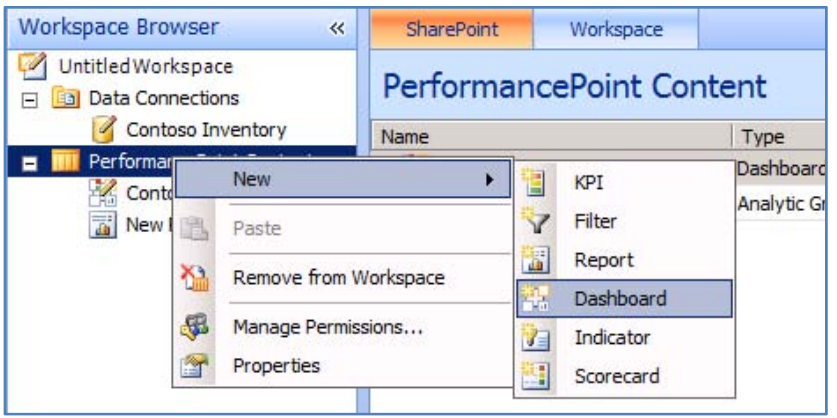
5. Create a report and dashboard.



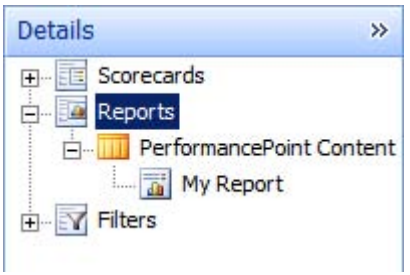
6. Make sure you have a data connection by dragging measures and dimensions from the details pain into the report designer.



7. Your report can be included in the dashboard.



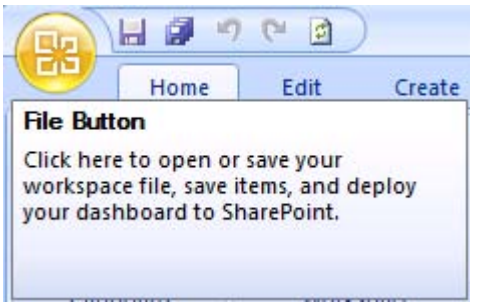
Select **Reports** and then drag My Report onto the Dashboard Content page.



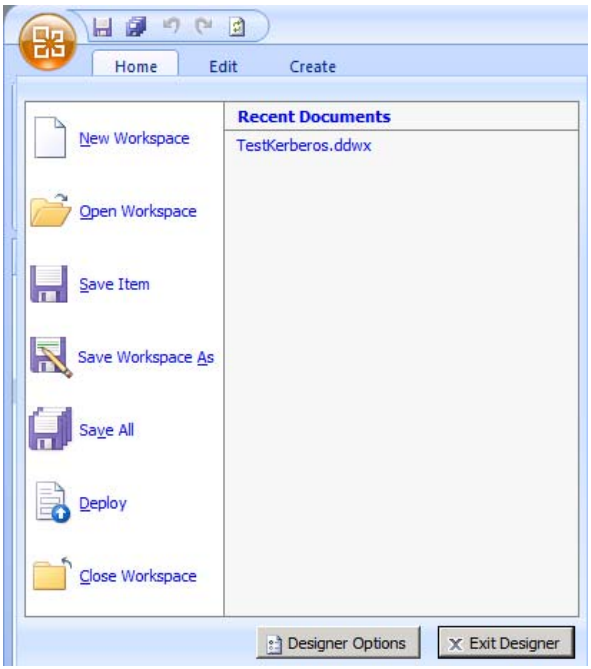
Publish the dashboard to SharePoint Server

The last step to validate the PerformancePoint Services application is to publish the dashboard and test refreshing and viewing the Analysis Services data. To do this:

1. Select the bright file button icon.



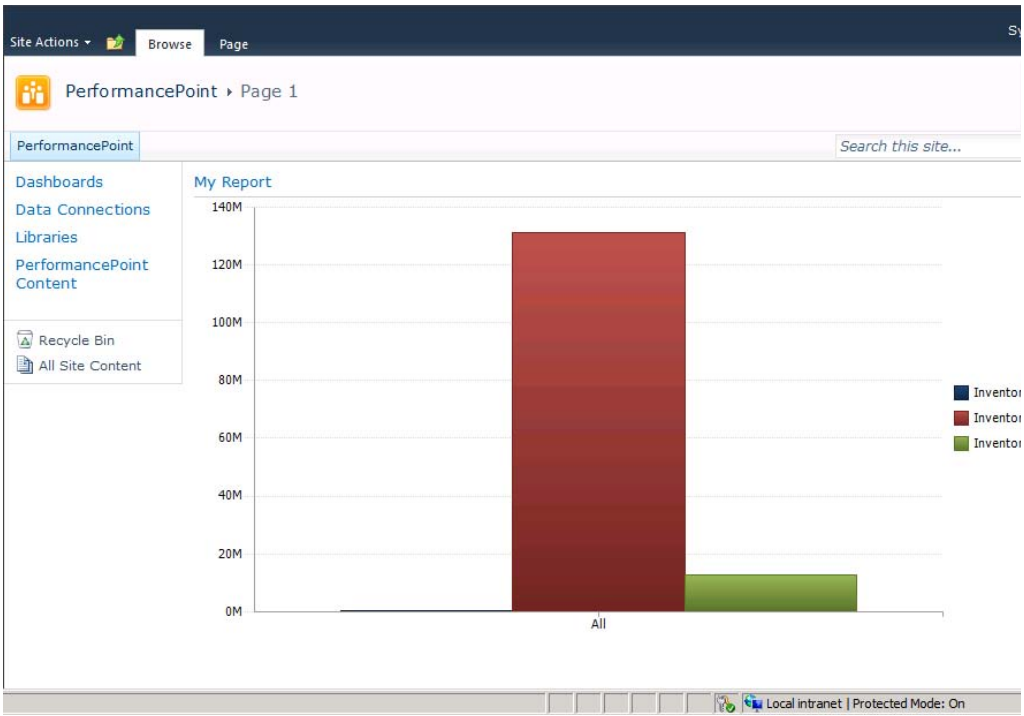
2. Click **Deploy** in the file selection.



3. Select a Master Page to which you want to publish.

4. Click the refresh button in your browser.

   If the data connection refreshes, you have successfully configured Kerberos delegation for PerformancePoint Services.


Site Actions
Browse
Page
PerformancePoint ▸ Page 1
PerformancePoint
Search this site...
Dashboards
Data Connections
Libraries
PerformancePoint Content
Recycle Bin
All Site Content
My Report
140M
120M
100M
80M
60M
40M
20M
0M
All
Inventor
Inventor
Inventor
Local intranet | Protected Mode: On

Identity delegation for Business Connectivity Services (SharePoint Server 2010)

Published: December 2, 2010

In this scenario you configure the Business Data Connectivity service application to use Kerberos constrained delegation to authenticate with SQL Server. Once it is configured, you create a new external content type and external list to test authentication and read operations within a SharePoint site.

In this scenario, the SharePoint Server Farm and BCS data source are both in the same domain. Therefore, we configure Kerberos constrained delegation to allow identity delegation to the back-end data source. If you are required to authenticate with data sources in other domains within the same forest, you have to configure basic (unconstrained) Kerberos delegation. Remember that BCS does not leverage the C2WTS; therefore you can use basic delegation.

Note:

If you are installing on Windows Server 2008, you may have to install the following hotfix for Kerberos authentication:

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used (http://support.microsoft.com/kb/969083)

Scenario dependencies

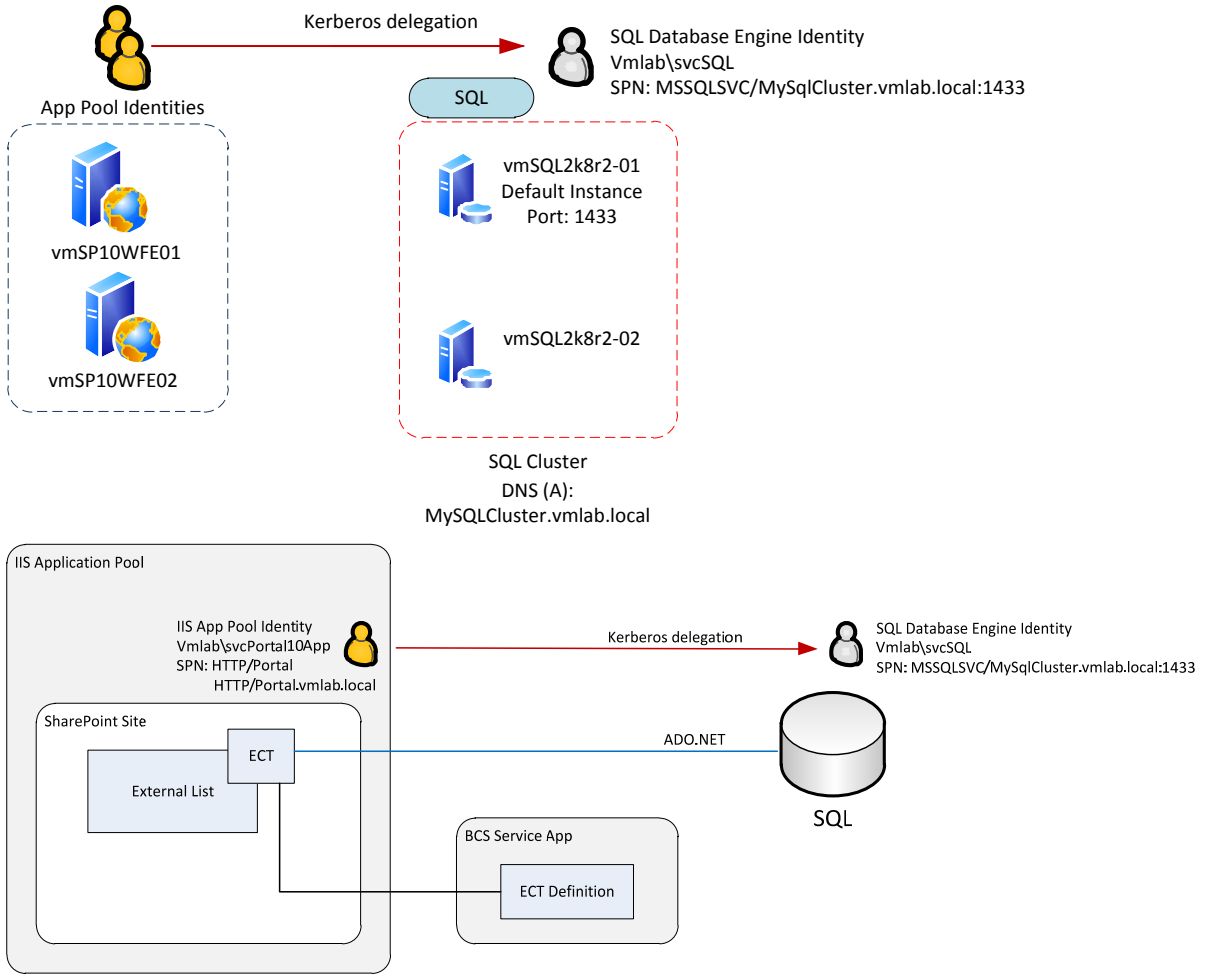
To complete this scenario you have to have completed the following:

- Scenario 1: Core Configuration
- Scenario 2: Kerberos Authentication for SQL OLTP

Configuration checklist

Area of configuration	Description
Active Directory configuration	Create BCS Application Service Account Validate Service Principal Names Configure Delegation
SharePoint Server configuration	Start the BCS Service Instance Create the BCS Service Application
Verification	Create a BCS External Content Type Configure BCS Security Create a BCS External List Open the external list in the browser

Scenario Environment Details



Step-by-step configuration instructions

Active Directory configuration

Create BCS Application Service Account

As a best practice Business Connectivity Services should run under its own domain identity. To configure the BCS Application an Active Directory account must be created. In this example the following accounts were created:

SharePoint Server service	IIS App Pool Identity
Business Connectivity Service	vmlab\svcBDC

Validate Service Principal Names

BCS external content types run within the context of the IIS application pool using the ECT type when BCS data is used in SharePoint sites. For BCS to connect and authenticate with external data sources using Kerberos authentication the IIS application pool service account and the service account for the external data source must have service principal names configured. Refer to scenarios 1 & 2 in this document to configure and validate the necessary SPNS on the web applications and SQL Server service accounts.

Configure delegation

To allow BCS to delegate the client's identity Kerberos delegation must be configured. Although constrained delegation is technically not required like Excel Services, unconstrained delegation can be used for BCS, it is a best practice to limit the scope of delegation the service is allowed to perform therefore constrained delegation will be configured in this example.

Each IIS application pool service account hosting the site running the ECT must be configured to allow delegation to the back-end services.

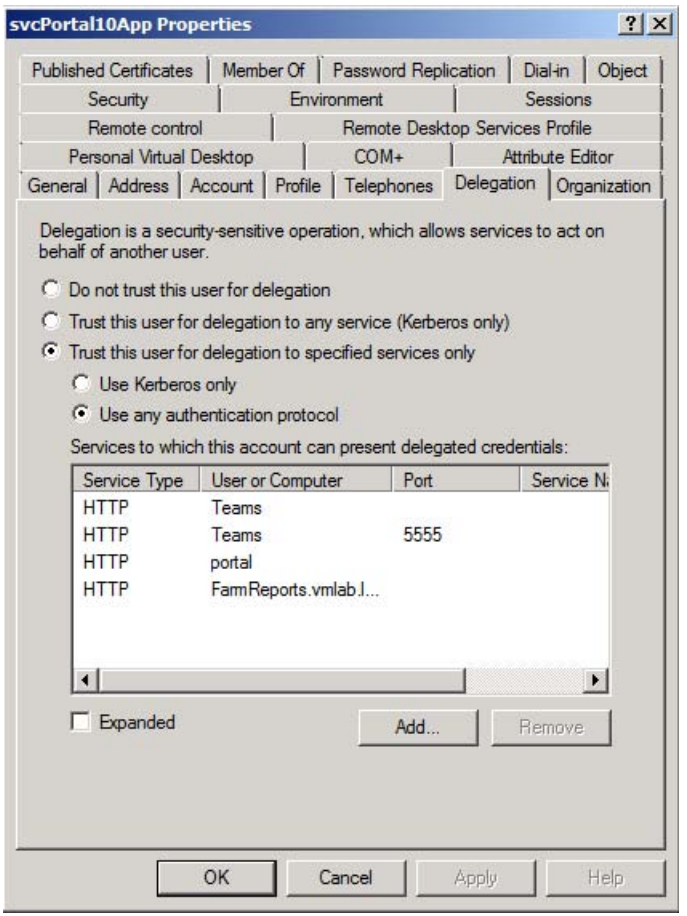
In our example the following delegation paths are needed:

Principal Type	Principal Name	Delegates To Service
User	svcPortal10App	MSSQLSVC/MySqlCluster.vmlab.local:1433
User	svcTeams10App	MSSQLSVC/MySqlCluster.vmlab.local:1433

To configure constrained delegation

1. Open the Active Directory Object's properties in Active Directory Users and Computers.

2. Navigate to the **Delegation** tab.

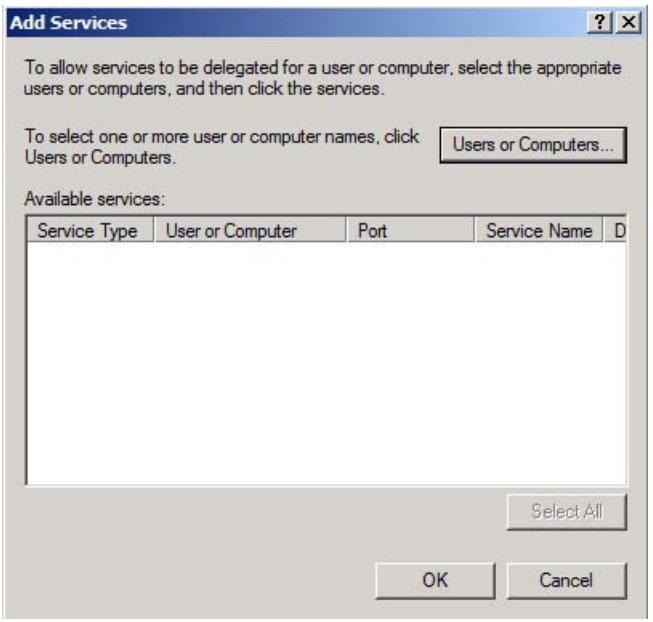


3. Select **Trust this user for delegation to specified services only**.

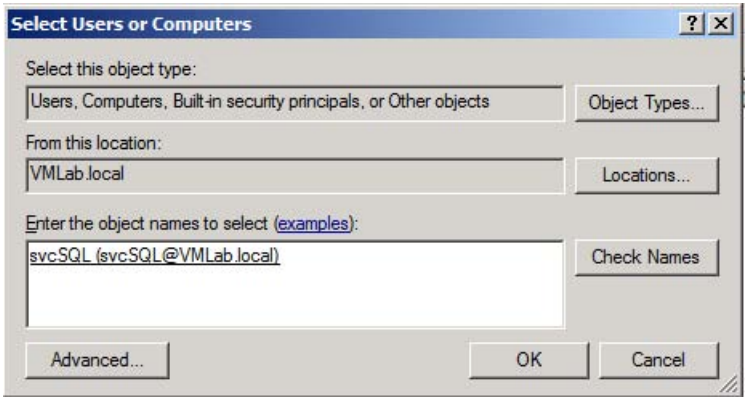
   Note:

   If you need BCS to authenticate with data sources within the same forest but outside of the domain that SharePoint Server resides in you will want to select **Trust this computer for delegation to any service** to configure basic delegation instead of constrained delegation. The BCS external content type will execute in the web application's IIS worker process and does not leverage the C2WTS. Remember that cross forest Kerberos delegation is not possible.

4. Click the **Add** button to select the service principal allowed to delegate to.



5. Select **User and Computers**.



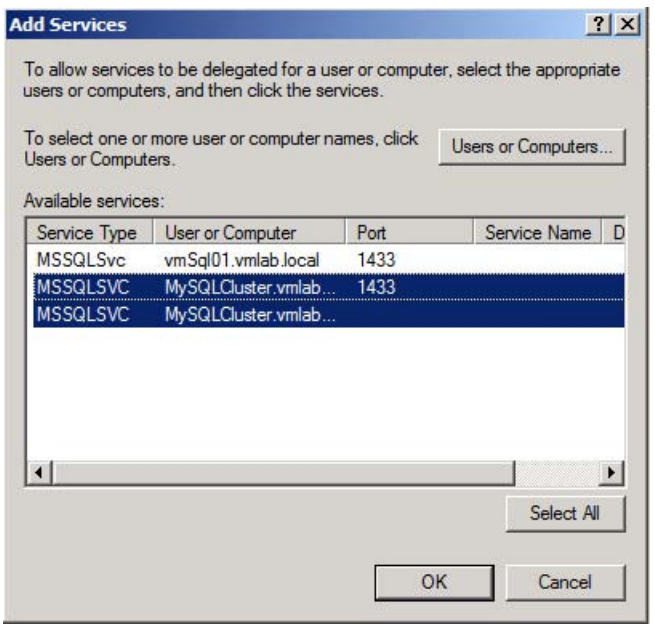
6. Select the service account running the service you wish to delegate to. In this example it is the service account for the SQL Server service.

   Note:

   The service account selected must have a SPN applied to it. In our example the SPN for this account was configured in a previous scenario.

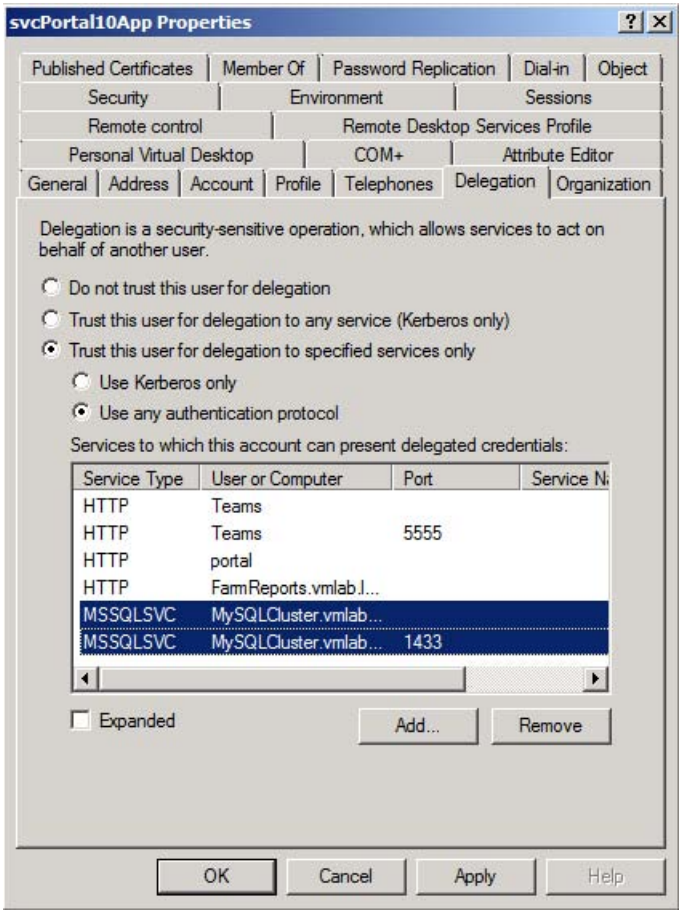
7. Click **OK**.

8. Select the SPNs you would like to delegate, and then click **OK**.



9. Select the services for the SQL Server cluster and click **OK**.

   You should now see the selected SPNS in the **services to which this account can presented delegated credentials** list.



10. Repeat these steps for each delegation path identified earlier in this section.

Verify MSSQLSVC SPN for the Service Account running the service on the SQL Server (performed in Scenario 2)

Verify the SPN for Analysis Services service account (vmlab\svcSQL) exists with the following SetSPN command:

```
SetSPN -L vmlab\svcSQL
```

You should see the following:

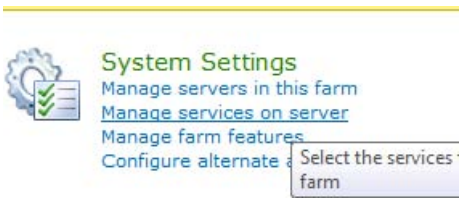
```
MSSQLSVC/MySqlCluster

MSSQLSVC/MySqlCluster.vmlab.local:1433
```

SharePoint Server configuration

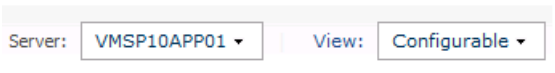
Start the BCS service instance

Before creating a BCS service application, start the BCS service on the designated farm servers.

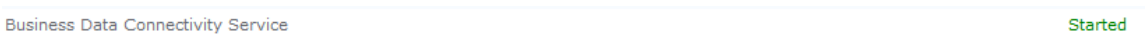
1. Open Central Administration.
2. Under Services, select **Manage services on server**.



3. In the Server Selection box in the upper-right corner, select the server(s) running Excel Services. In this example, it is VMSP10APP01.



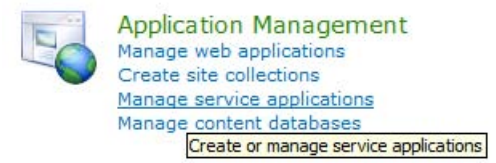
4. Start the **Business Data Connectivity Service**service.



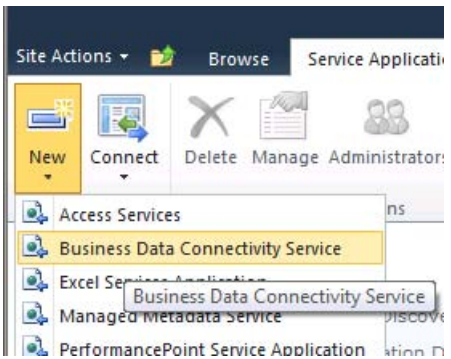
Create the BCS service application

Next, configure a new BDC service application and application proxy to allow web applications to consume BDC services:

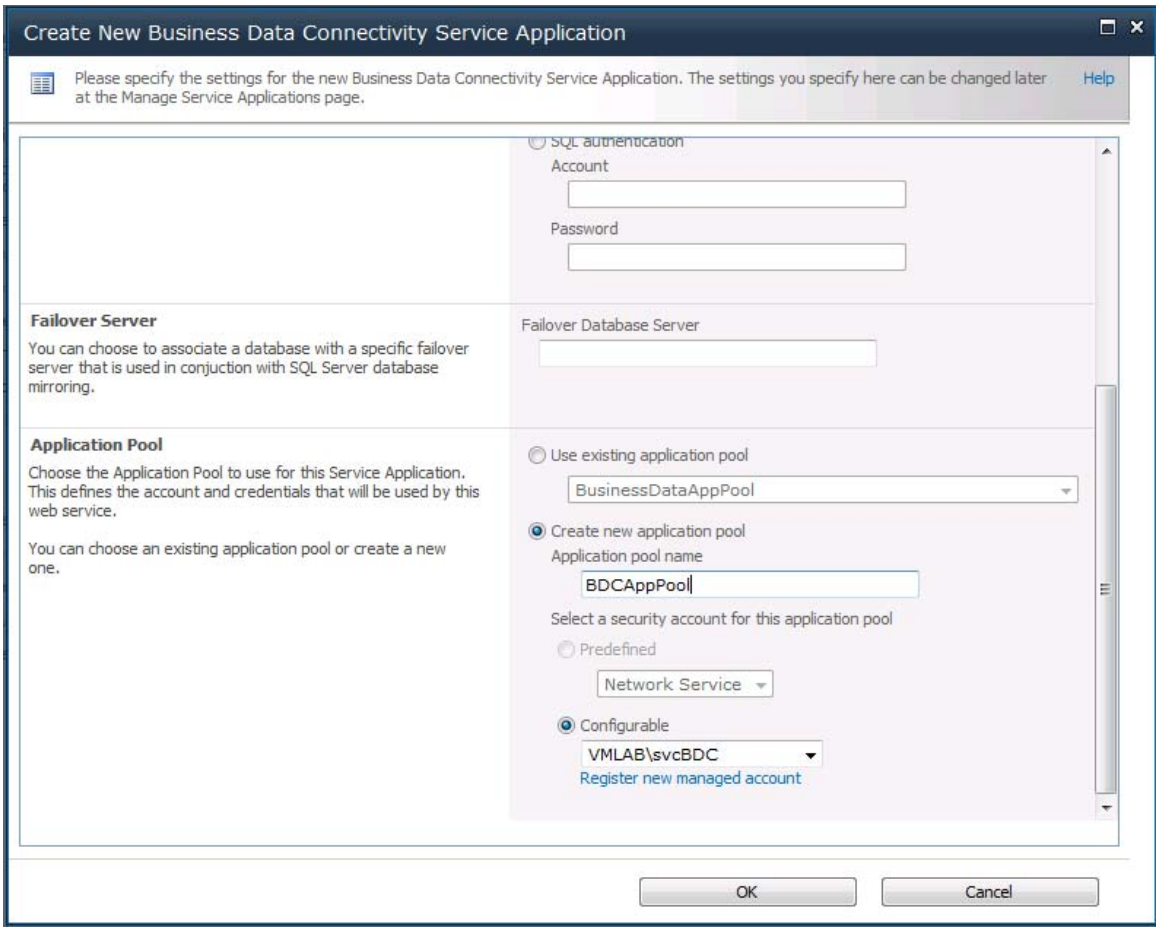
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



3. Select **New** then **Business Data Connectivity Service**.



4. Configure the new service application. Be sure to select the correct service account (create a new managed account if the BDC service account is not in the list).

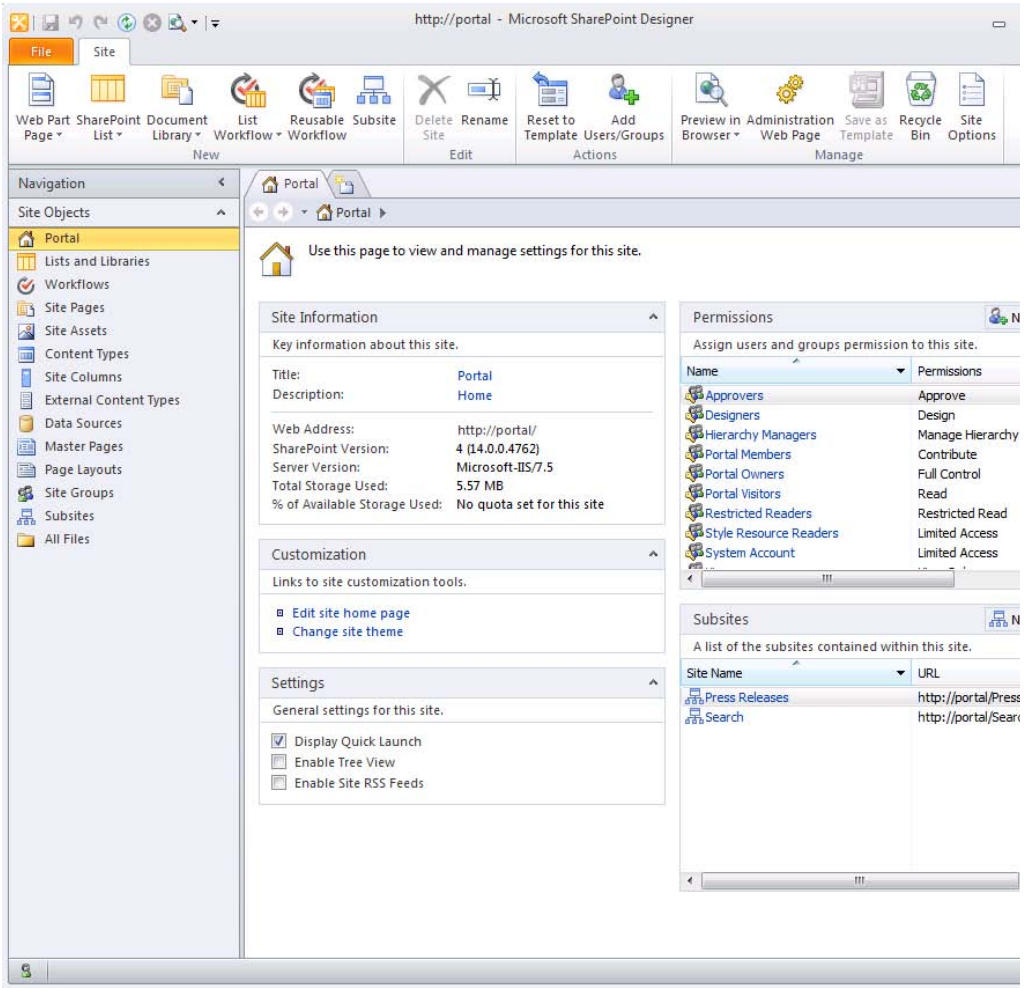


Verification

Create a BCS external content type

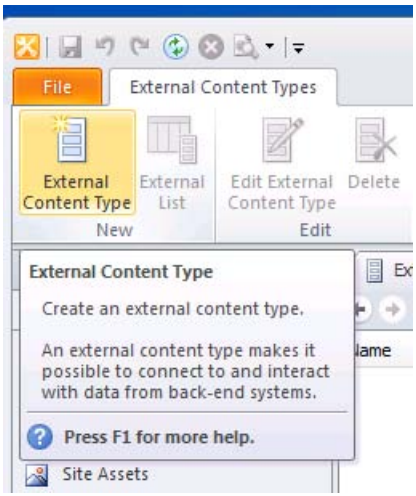
To access external data through BDC a BDC eternal content type must be created. In this example we will use SharePoint Designer 2010 to create the external content type in the Portal web application (http://portal):

1. Open SharePoint Designer 2010.
2. Open the test site collection at http://portal.

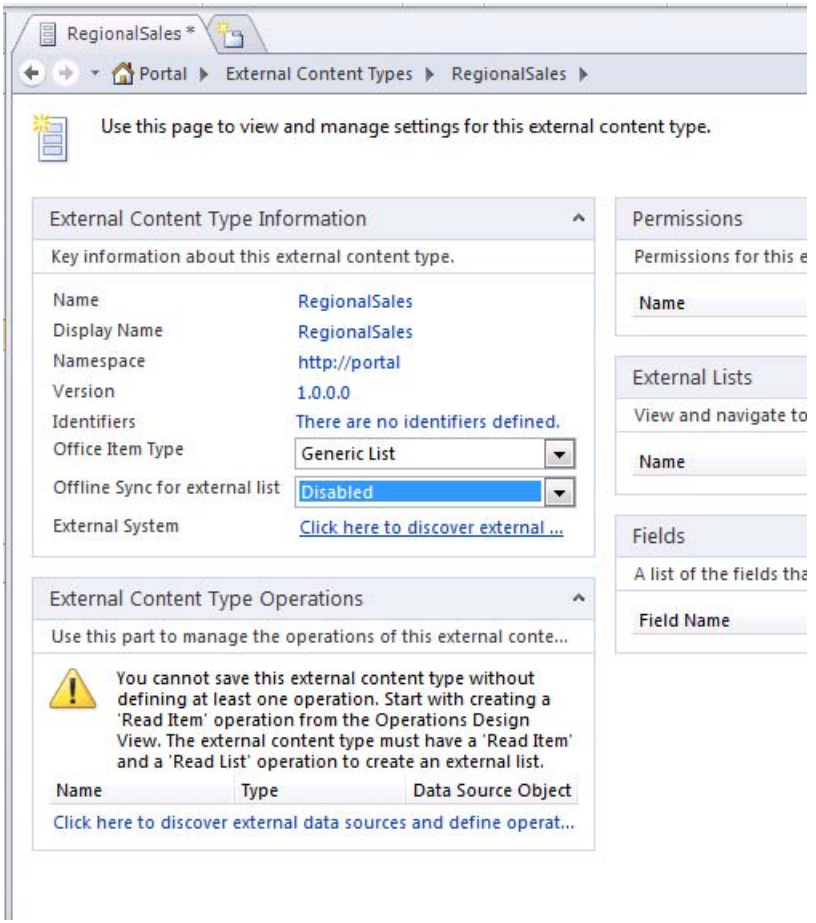


3. On the left hand navigation, click **External Content Types**.

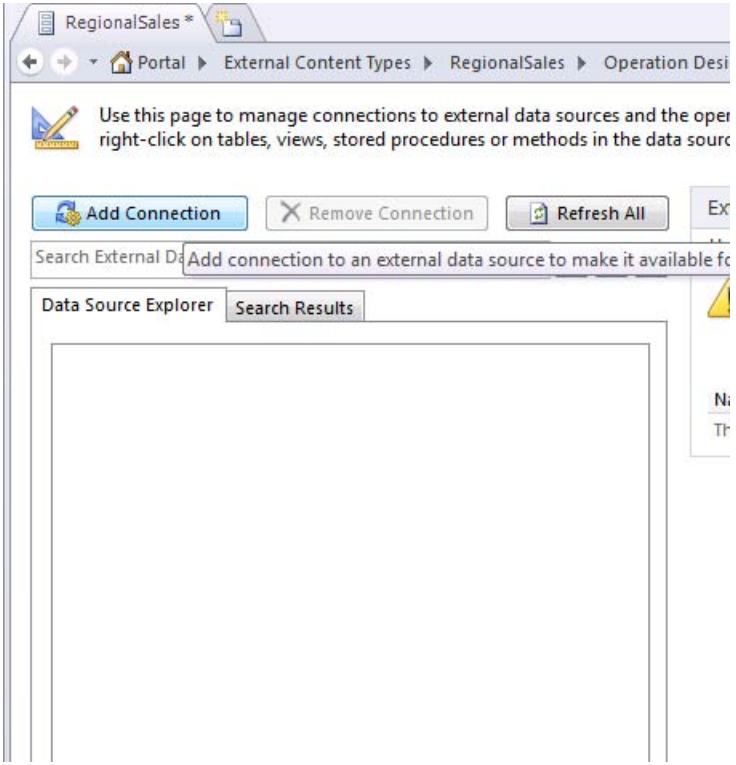
4. Select **External Content Type** in the **New** section of the ribbon in the upper left hand corner of the page.



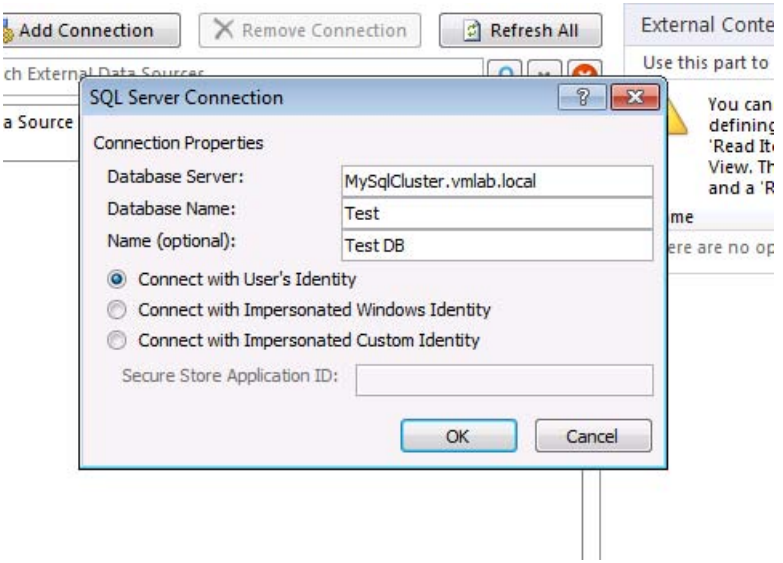
5. Give the External Content Type a display name.



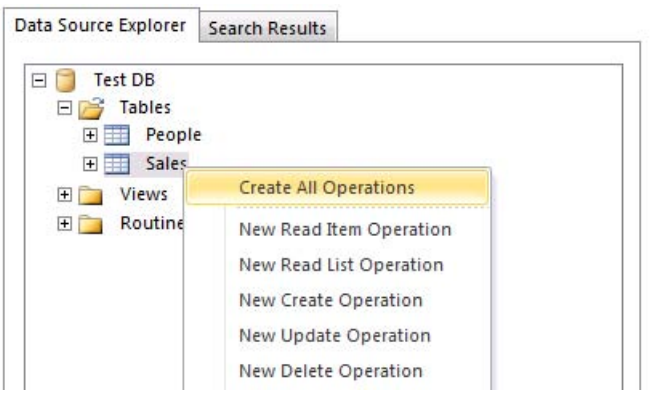
6. Then select **Click here to discover external data sources and define operations**.
7. Click **Add Connection**.



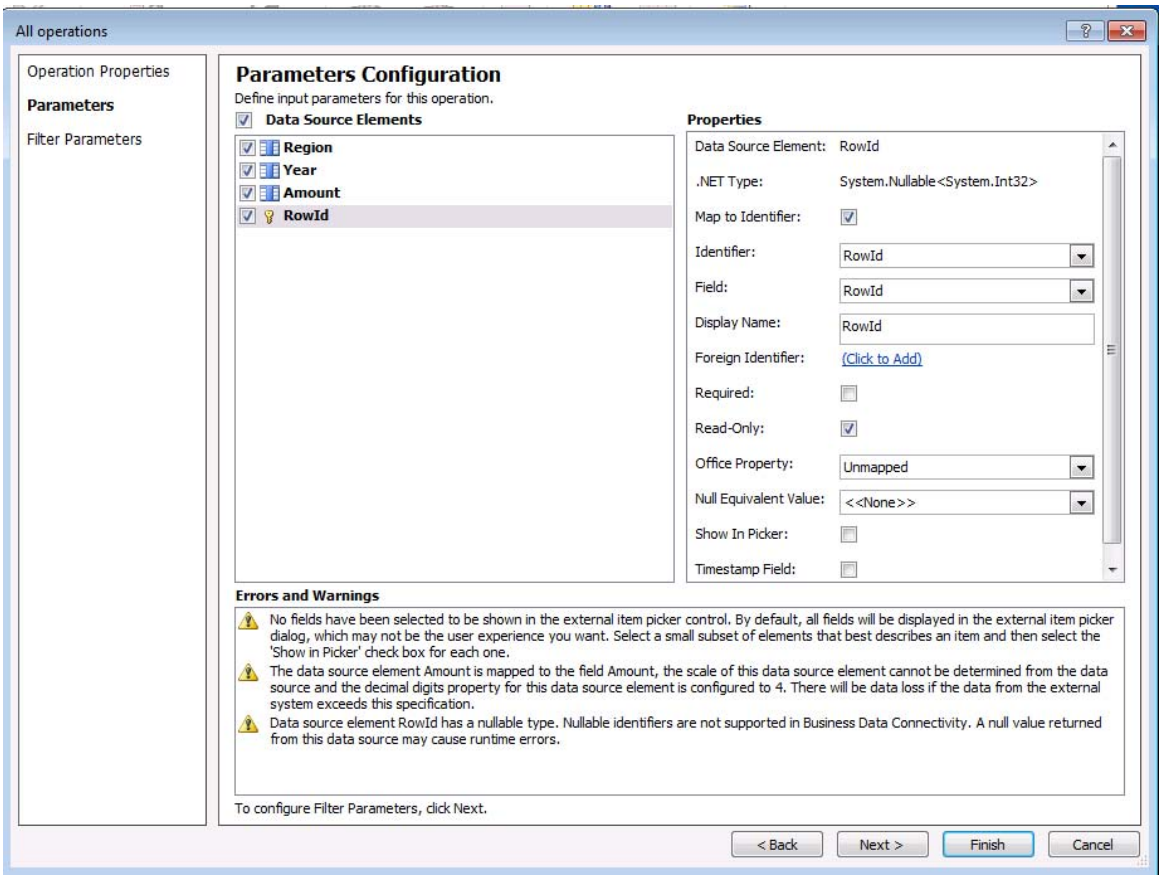
8. Select **SQL Server** from the **Data Source Type** dropdown list and add the information to connect to the test database. Be sure to select **Connect with the User's Identity** to test delegation.



9. Expand the new connection. Right-click the test table (**Sales**) and select **Create All Operations**.



10. You should see an error explaining there isn't a unique identifier defined. Select the identifier column and select the **Map to Identifier** check box. Click **Finish** to accept the default options and create the ECT operations.

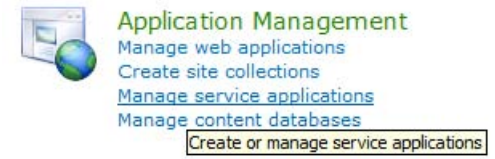


11. Click Save (CTRL+S). This will publish the ECT to the BDC service application metadata store.

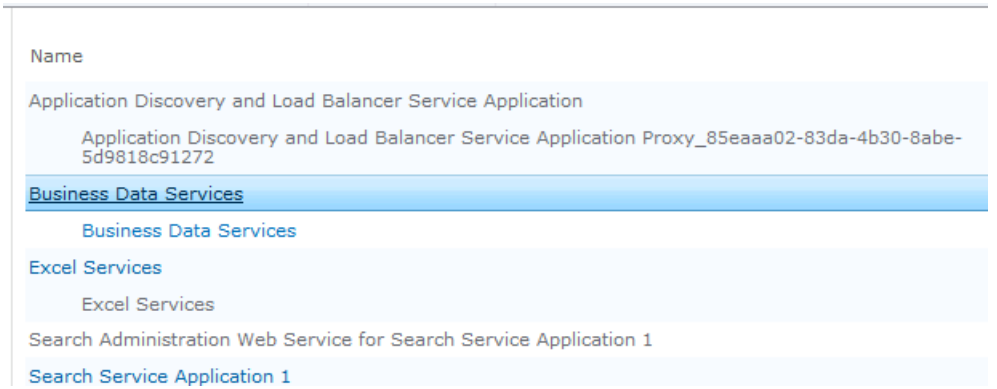
Configure BCS security

Before clients can use the BCS external content type in the portal web application BCS permissions must be configured. BCS supports a granular permission model but for the purposes of this demo we will configure secure at the Metadata store level and propagate the security changes to all objects in the store.

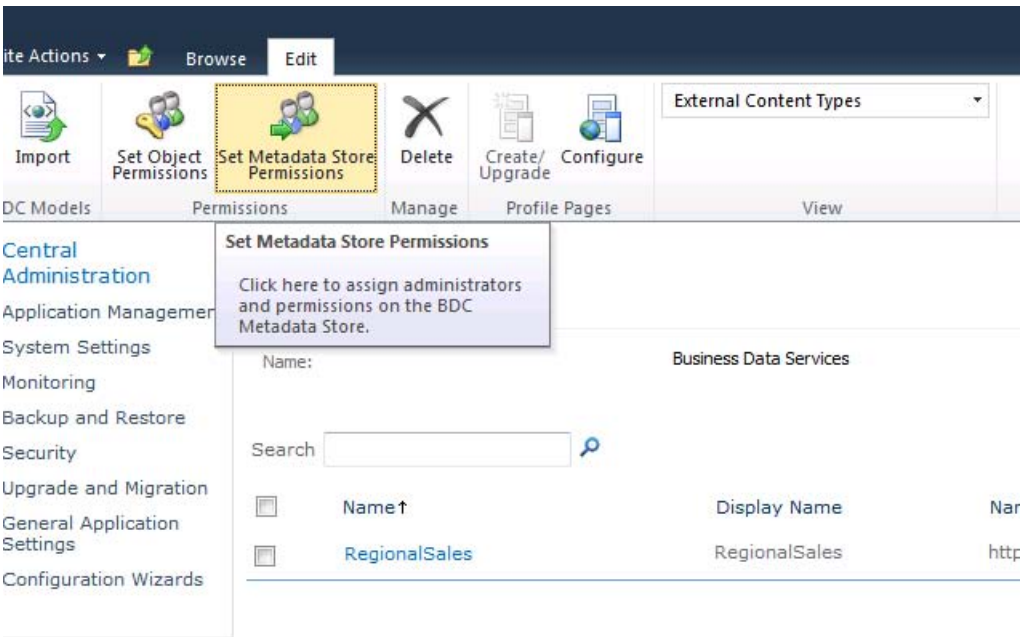
1. Open Central Administration.
2. Select **Manage Service Applications** under **Application Management**.



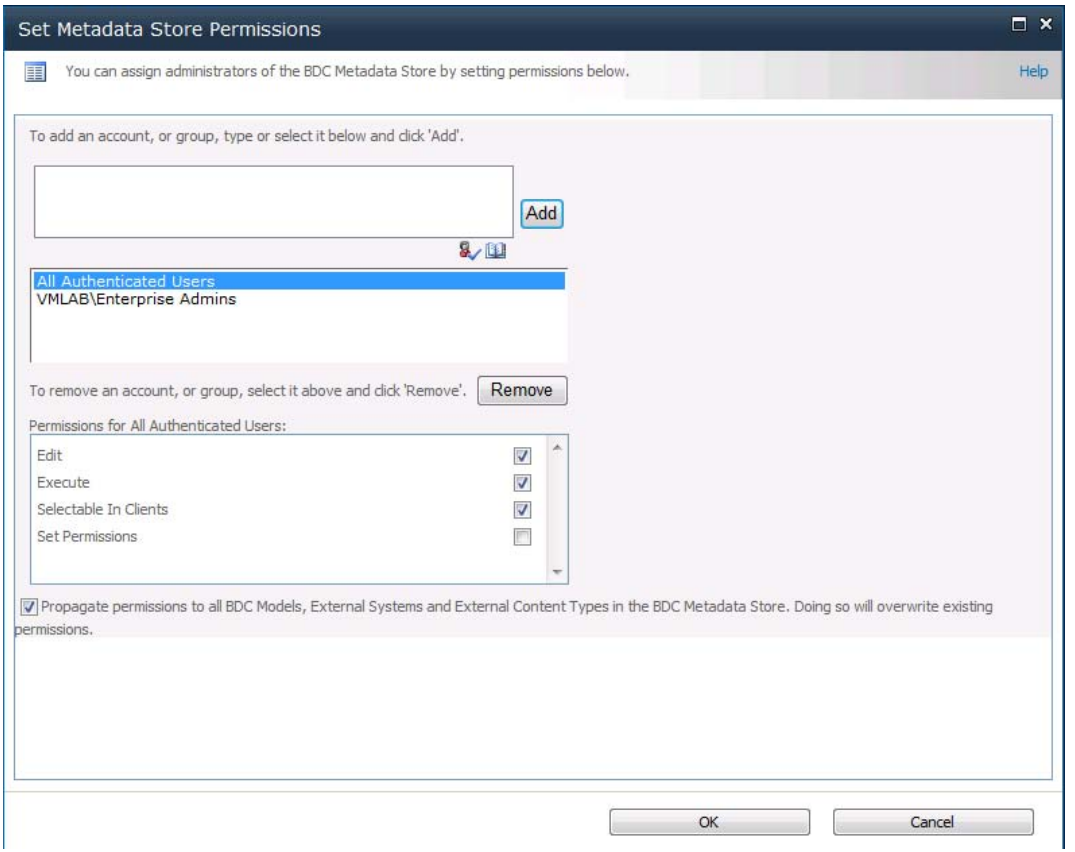
3. Click the link for the new Service Application, **Business Data Services** in this example.



4. Select **Set Metadata Store Permissions**.



5. In our example, we configured Enterprise Admins with all permissions and All Authenticated Users with all permissions except the **Set Permissions** permission.

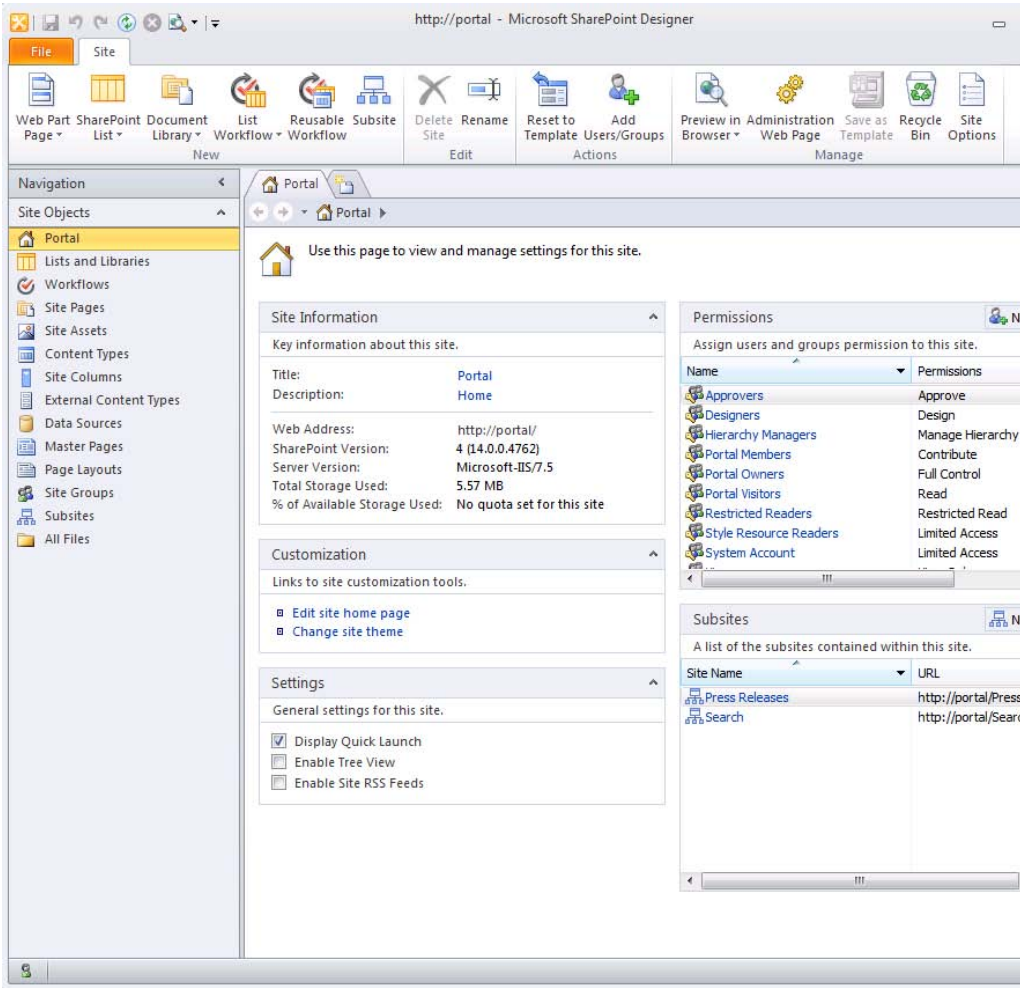


6. Ensure the **Propagate permissions** check box is selected and click **OK** to save your changes.

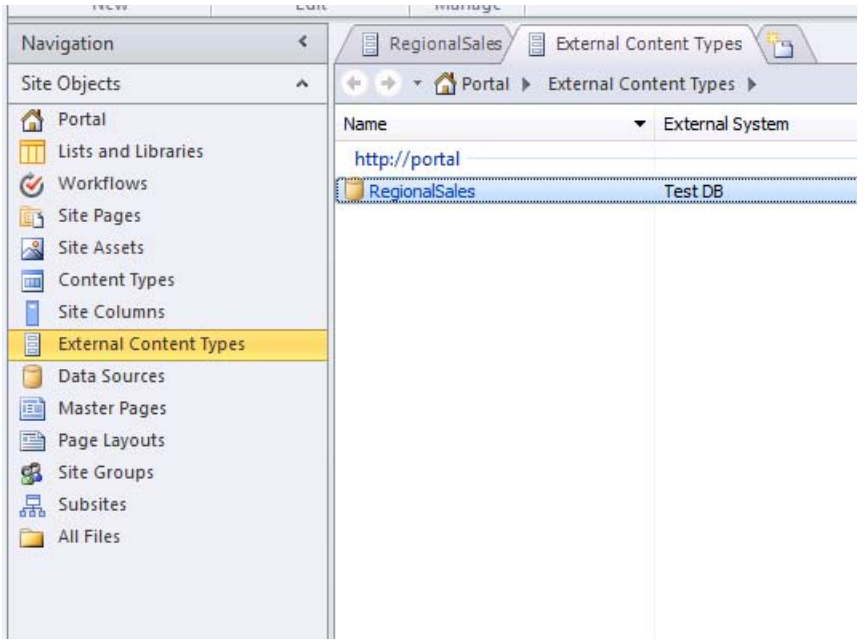
Create a BCS External List

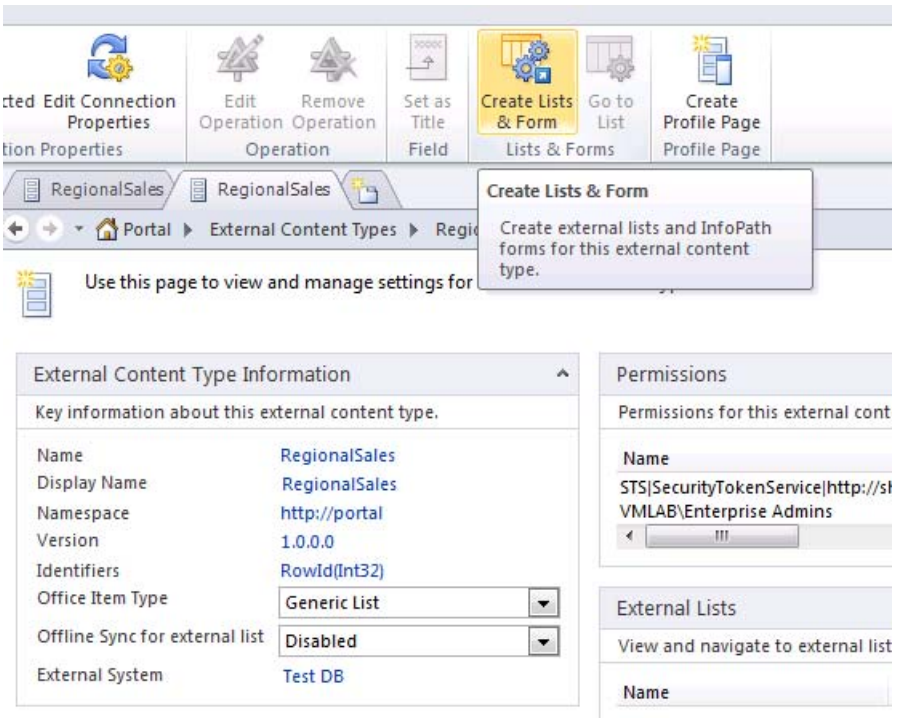
To test the external content type we will configure an external list to display the external data in the portal application:

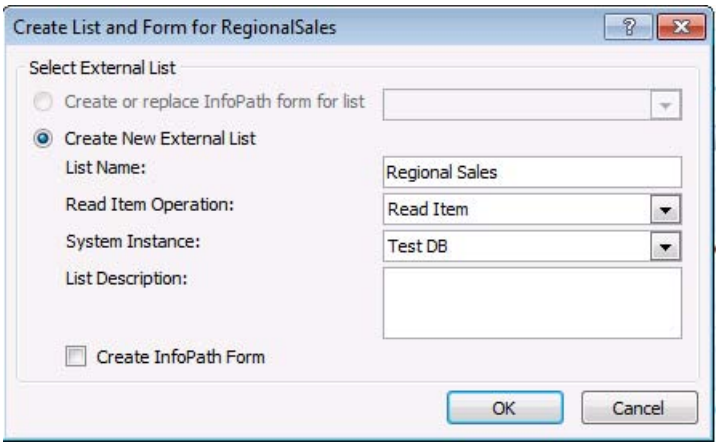
1. Open SharePoint Designer 2010.
2. Open the test site collection at http://portal.



3. Select **External Content Types** on the left side.


Navigation
Site Objects
Portal
Lists and Libraries
Workflows
Site Pages
Site Assets
Content Types
Site Columns
External Content Types
Data Sources
Master Pages
Page Layouts
Site Groups
Subsites
All Files
RegionalSales
External Content Types
Portal ▸ External Content Types
Name
External System
http://portal
RegionalSales
Test DB

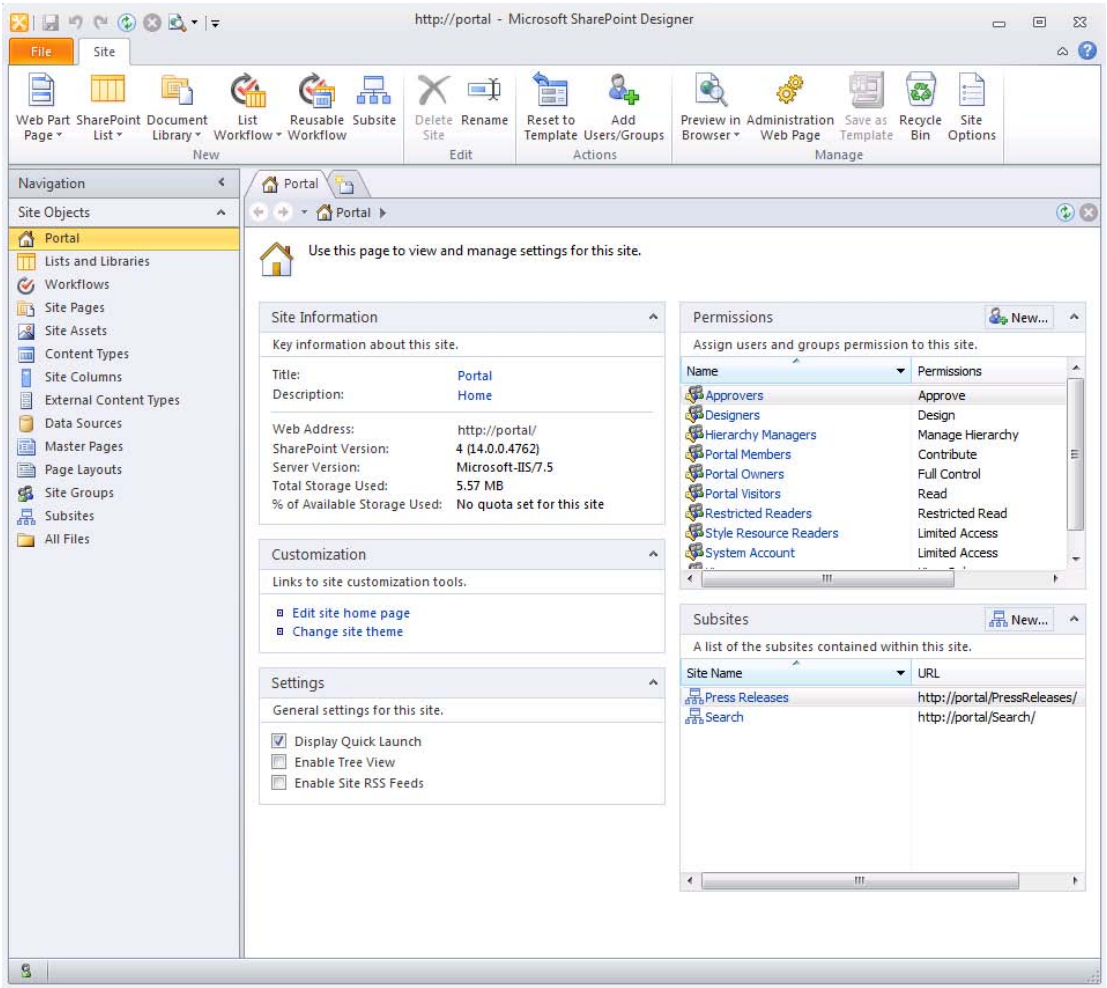

Edit Connection Properties
Edit Operation
Remove Operation
Operation
Set as Title
Field
Create Lists & Form
Go to List
Lists & Forms
Create Profile Page
Profile Page
RegionalSales
RegionalSales
Portal ▸ External Content Types
Create Lists & Form
Create external lists and InfoPath forms for this external content type.
Use this page to view and manage settings for
External Content Type Information
Key information about this external content type.
Name RegionalSales
Display Name RegionalSales
Namespace http://portal
Version 1.0.0.0
Identifiers RowId(Int32)
Office Item Type Generic List
Offline Sync for external list Disabled
External System Test DB
Permissions
Permissions for this external cont
Name
STS|SecurityTokenService|http://s
VMLAB\Enterprise Admins
External Lists
View and navigate to external list
Name



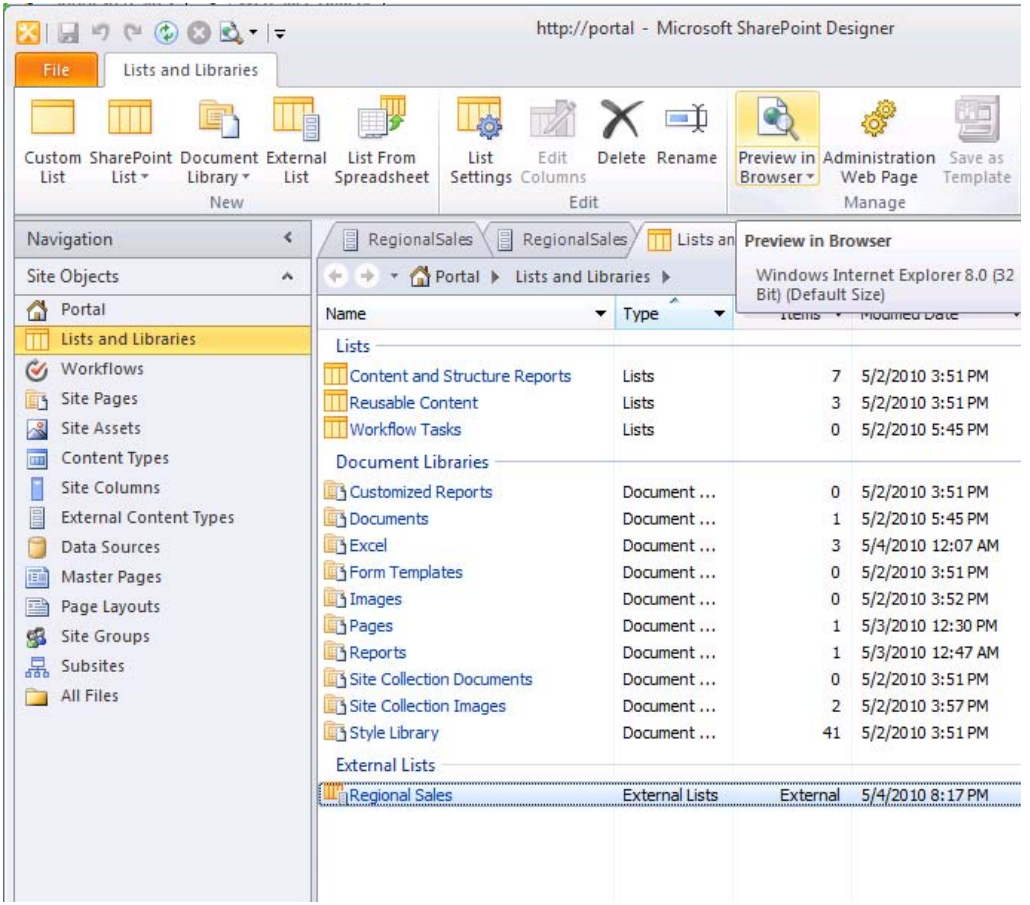
4. Click the content type that you created earlier.
5. In the ribbon, click **Create Lists & Form**.
6. If you are prompted to save the external content type, click **Yes**.
7. On the **Create List and Form** dialog box, type a list name in the **List Name** text box, and then click **OK**.

Open the external list in the browser

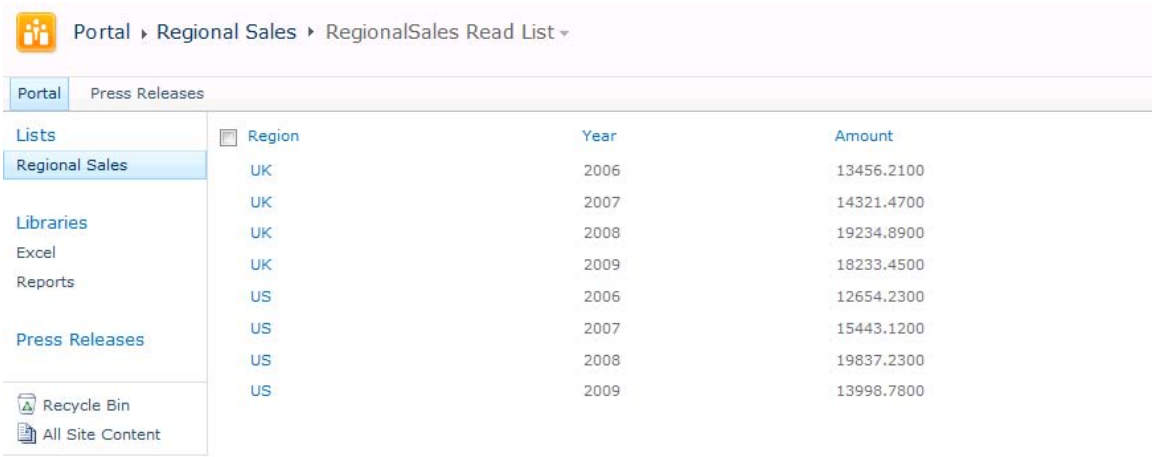
1. Open SharePoint Designer 2010.
2. Open the test site collection at http://portal.



3. Click "Lists and Libraries" in the left hand navigation.
4. Select the external list at the bottom of the **List and Libraries** list.
5. Click the **Preview in Browser** button.



Internet Explorer will open and display the selected site and external list.



6. Validate the external data is displayed correctly. To further validate the connection, change the source data in SQL Server Management Studio and refresh the browser page. You should see the data changes reflected in the browser.

Kerberos configuration known issues (SharePoint Server 2010)

Published: December 2, 2010

Kerberos authentication and non-default ports

There is a known issue where some Kerberos clients (.NET Framework, Internet Explorer 7 and 8 included) do not correctly form service principal names when attempting to authenticate with Kerberos enabled web applications that are configured on non-default ports (ports other than 80 and 443). The root of the problem is the client does not properly form the SPN in the TGS request by specifying it without the port number (as seen in the Sname of the TGS request).

Example:

If the web application is running at http://intranet.contoso.com:1234, the client will request a ticket for a service with a SPN equal to http/intranet.contoso.com instead of http/intranet.contoso.com:1234.

Details regarding the issue can be found in the following articles:

- Internet Explorer 6 cannot use the Kerberos authentication protocol to connect to a Web site that uses a non-standard port in Windows XP and in Windows Server 2003 (http://support.microsoft.com/kb/908209/en-us)
- Configure Kerberos authentication (Office SharePoint Server 2007) (http://go.microsoft.com/fwlink/?LinkId=196987)

To work around this issue, register SPNs with and without port number. Example:

- http://intranet.contoso.com:12345
- http/intranet
- http/intranet.contoso.com
- http/intranet:12345

- http/intranet.contoso.com:12345

We recommend that you register the non-default port to ensure that if the issue is resolved in some future service pack or hot fix, the applications using the workaround will still continue to function.

Note that this workaround will not work if the following conditions are true:

- There is more than one web application running on a non-default port
- The web applications either bind to the host name of the server or bind to the same host header (on different ports)
- The web application IIS application pools use different service accounts
- http://server.contoso.com:5000 AppPool Id: contoso\svcA
- http://server.contoso.com:5001 AppPool Id: contoso\svcB

If these conditions are true, following the recommendation in this workaround will yield duplicate SPNs registered to different service accounts which will break Kerberos authentication.

If you have multiple web sites sharing a common host name running on multiple ports, and you use different IIS application pool identities for the web applications, then you cannot use Kerberos authentication on all web sites. (One application can use Kerberos, the rest will require another authentication protocol.) To use Kerberos on all applications in this scenario, you would need to either:

1. Run all web applications under 1 shared service account
2. Run each site with its own host header

Kerberos authentication and DNS CNAMEs

There is a known issue with some Kerberos clients (Internet Explorer 7 and 8 included) that attempt to authenticate with Kerberos enabled services that are configured to resolve using DNS CNAMEs instead of A Records. The root of the problem is the client does not correctly form the SPN in the TGS request by creating it using the host name (A Record) instead of the alias name (CNAME).

Example:

A Record: wfe01.contoso.com

CNAME: intranet.contoso.com (aliases wfe01.contoso.com)

If the client attempts to authenticate with http://intranet.contoso.com, the client does not correctly form the SPN and requests a Kerberos ticket for http/wfe01.contoso.com instead of http/intranet.contoso.com

Details regarding the issue can be found in the following articles:

http://support.microsoft.com/kb/911149/en-us

http://support.microsoft.com/kb/938305/en-us

To work around this issue, configure Kerberos enabled services using DNS A records instead of CNAME aliases. The hotfix mentioned in KB article will correct this issue for Internet Explorer but will not correct the issue for the .NET framework (which is used by Microsoft Office SharePoint Server for web service communication).

Kerberos authentication and Kernel Mode Authentication

Note:

Kernel Mode Authentication is not supported in SharePoint 2010 Products. This information is provided for informational purposes only.

Beginning in IIS version 7.0, there is a new authentication feature called Kernel Mode Authentication. When an IIS web site is configured to use Kernel Mode authentication, HTTP.sys will authenticate the client's requests instead of the application pool's worker process. Because HTTP.sys runs in kernel mode this yields better performance but also introduces a bit of complexity when configuring Kerberos. This is due to HTTP.sys running under the computer's identity and not under the identity of the worker process. When HTTP.sys receives a Kerberos ticket, by default it will attempt to decrypt the ticket using the server's encryption key (aka secret) and not the key for the identity the worker process is running under.

If a single web server is configured to use Kernel Mode authentication, Kerberos will work without any additional configuration or additional SPNs because the server will automatically register a HOST SPN when it is added to the domain. If multiple web

servers are load balanced, the default Kernel Mode Authentication configuration will not work, or at least will intermittently fail, because the client has no way of ensuring the service ticket they received in the TGS request will work with the server authenticating the request.

To work around this issue you can do the following:

- Turn off Kernel Mode Authentication
- Configure HTTP.sys to use the IIS application pool's identity when decrypting service tickets. See Internet Information Services (IIS) 7.0 Kernel Mode Authentication Settings.

  You may also need a hotfix when configuring HTTP.sys to use the application pool's credentials: FIX: You receive a Stop 0x0000007e error message on a blue screen when the AppPoolCredentials attribute is set to true and you use a domain account as the application pool identity in IIS 7.0

Kerberos authentication and session-based authentication

You may notice increased authentication traffic when using Kerberos authentication with IIS 7.0 and greater. This may be related to Windows authentication settings in IIS, in particular:

Setting	Description
AuthPersistNonNTLM	Optional Boolean attribute. Specifies whether IIS automatically re-authenticates every non-NTLM (for example, Kerberos) request, even those on the same connection. Falseenables multiple authentications for the same connections. The default is False. **Note:** A setting of Truemeans that the client will be authenticated only once on the same connection. IIS will cache a token or ticket on the server for a TCP session that

Setting	Description
	stays established.
authPersistSingleRequest	Optional Boolean attribute. Setting this flag to True specifies that authentication persists only for a single request on a connection. IIS resets the authentication at the end of each request, and forces re-authentication on the next request of the session. The default value is False.

For instructions on how to configure authentication persistence in IIS 7.0, see You may experience slow performance when you use Integrated Windows authentication together with the Kerberos authentication protocol in IIS 7.0 and Implementing Access Control.

Kerberos authentication and duplicate/missing SPN issues

When configuring Kerberos authentication, it is easy to accidentally configure duplicate service principal names, especially if you use **SetSPN -A**orthe ADSI Edit (adsiedit.msc) tool to create your SPNs. The general recommendation is to use **SetSPN -S** to create SPNs because the -S switch will check for a duplicate SPN before creating the specified SPN.

If you suspect you have duplicate SPNs in your environment, use the **SetSPN -X** command to query for all duplicate SPNs in your environment (Windows 2008 or greater only). If any SPNs are returned you should investigate why the SPNs have been registered and delete any SPNs that are duplicates and are not needed. If you have two services running with two different identities and both use the same SPN (duplicate SPN issue) you need to reconfigure one of those services to either use a different SPN or share a common service identity.

```
C:\Users\administrator.VMLAB>setspn -x
Checking domain DC=VMLab,DC=local
Processing entry 0
found 0 group of duplicate SPNs.
```

If you suspect a SPN has not been registered, or not registered in a format required, you can use the SetSPN -Q <insert SPN> to query for the existence of a particular SPN.

Kerberos Max Token Size

In some environments, users may be members of many Active Directory groups, which can increase the size of their Kerberos tickets. If the tickets grow too large, Kerberos authentication can fail. For more information about how to adjust the max token size, see New resolution for problems with Kerberos authentication when users belong to many groups (http://support.microsoft.com/kb/327825).

Note:

When adjusting maximum token size, be aware that if you configure the maximum token size beyond the maximum value for the registry setting, you may see Kerberos authentication errors. We recommend not exceeding 65535 decimal, FFFF hexadecimal, for maximum token size.

Kerberos authentication hotfixes for Windows Server 2008 and Windows Vista

A Kerberos authentication fails together with the error code 0X80090302 or 0x8009030f on a computer that is running Windows Server 2008 or Windows Vista when the AES algorithm is used(http://support.microsoft.com/kb/969083).

You may need to install a hotfix for Kerberos authentication on all computers that are running Windows Server 2008 or Windows Vista in your environment. This includes all computers that are running SharePoint Server 2010, SQL Server, or Windows Server 2008 that SharePoint Server attempts to authenticate with by using Kerberos

authentication. Follow the instructions in the support page to apply the hotfix if you experience the symptoms documented in the support case.

How to reset the Claims to Windows Token Service account (SharePoint Server 2010)

Published: December 2, 2010

Scenario: The Claims to Windows Token Service account is changed unintentionally or otherwise needs to be reset back to default.

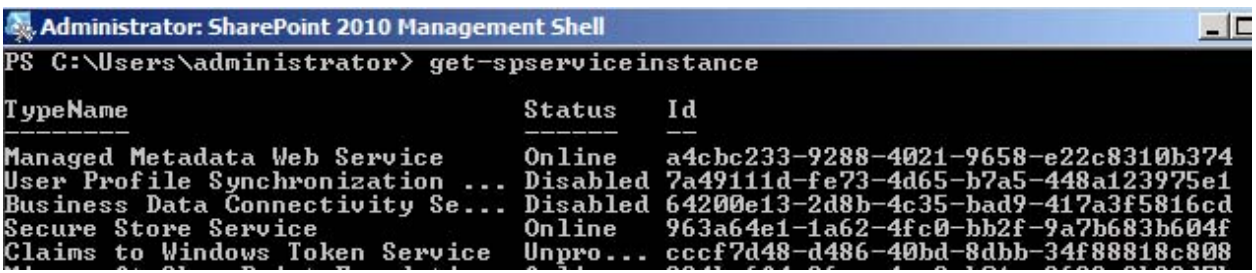
Solution

The Claims to Windows Token Service cannot be reset to the Local System account by using Central Administration. The following Windows PowerShell cmdlets can be used to reset the Claims to Windows Token Service back to Local System.

Launch the SharePoint Management Shell from the computer that is running SharePoint Server.

Run the following cmdlet to view a list of services.

```
Get-SPServiceInstance
```

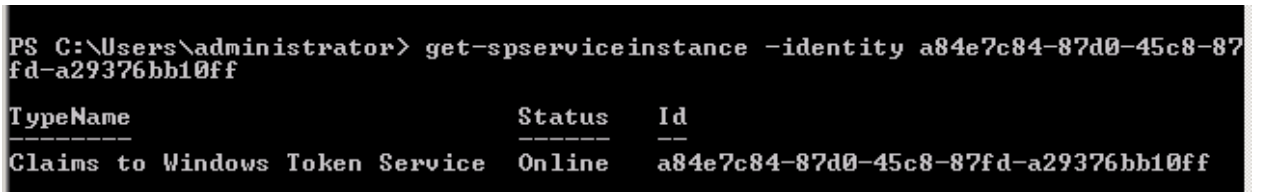


Find and copy the Id of the Claims To Windows Token Service. Right-click in the Windows PowerShell window and choose **Mark**. This will allow you to select and copy the Id with your mouse cursor. After highlighting the Id, press ENTER on your keyboard.

Test your Id by running the following cmdlet.

```
Get-SPServiceInstance -identity <Paste the C2WTS Id>
```

Right-click in the PowerShell window and paste the Id you copied earlier.



Next, set a variable by running this cmdlet:

```
$claims = get-spserviceinstance -identity <Paste the C2WTS Id>
```

Run these cmdlets to reset the C2WTS back to Local System:

```
$claims.Service.ProcessIdentity.CurrentIdentityType=0  // The 0 in the preceding
line is IdentityType.LocalSystem $claims.Service.ProcessIdentity.Update()
$claims.Service.ProcessIdentity.Deploy() $claims.Service.ProcessIdentity  //
This output demonstrates that the cmdlet was successful CurrentIdentityType :
LocalSystemCurrentSecurityIdentifier : S-1-5-18 ManagedAccount : ProcessAccount
: S-1-5-18 Username : NT AUTHORITY\SYSTEM
```