|  |
| --- |
| ASLB Setup |
|  |

Table of Contents

[1. Getting Started 3](#_Toc253066715)

[1.1. Unzip the ASLB Setup files 3](#_Toc253066716)

[2. Configuring the Analysis Services Query Servers 3](#_Toc253066717)

[2.1. Configuring IIS when you are using IIS 7.0 or IIS 7.5 3](#_Toc253066718)

[2.2. Configuring IIS when you are using IIS 6.0 9](#_Toc253066719)

[2.3. Configuring and Verifying MSMDPump 10](#_Toc253066720)

[3. Configuring the Relational Server(s) 10](#_Toc253066721)

[3.1. Configure the SQL Server Agent service account 11](#_Toc253066722)

[3.2. Creating the ALSB database on the primary server 11](#_Toc253066723)

[3.3. Registering the Microsoft.AnalysisServices.dll assembly 11](#_Toc253066724)

[3.4. Compiling and registering the ASLBMonitor assembly 12](#_Toc253066725)

[3.5. Create stored procedures and jobs, and populate Metadata Tables 12](#_Toc253066726)

[3.6. Mirror ASLB database on to second relational server (optional) 13](#_Toc253066727)

[4. Configuring the Web Server(s) 14](#_Toc253066728)

[4.1. Configuring IIS for ASLB when you are using IIS 7.0 or IIS 7.5 14](#_Toc253066729)

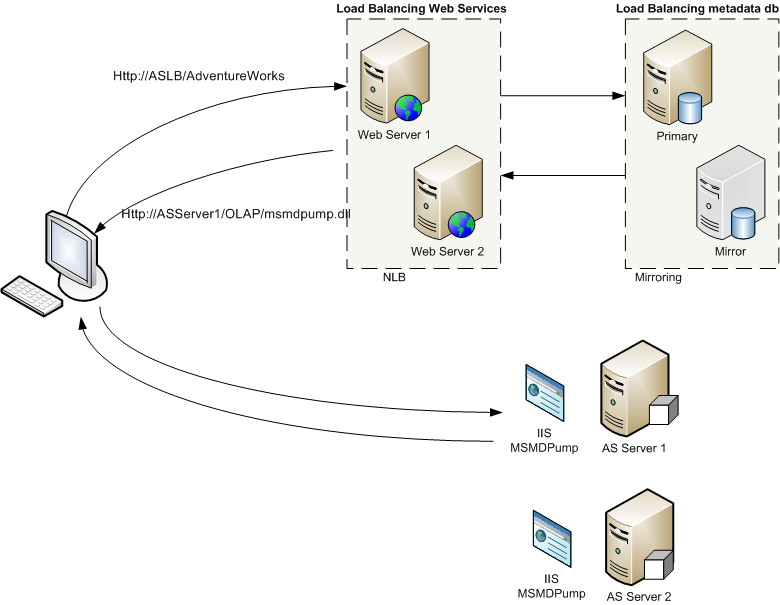
[4.2. Configuring IIS for ASLB when you are using IIS 6.0 17](#_Toc253066730)

[5. Testing 23](#_Toc253066731)

# ASLB Overview

ASLB is a custom Analysis Services load balancing solution that consists of a load balancing Web service backed by a SQL Server load balancing metadata database to load balance MDX queries across multiple query servers. The metadata database contains information about each Analysis Services query server to which queries can be redirected by the Web service.

|  |  |
| --- | --- |
| Load Balancing Web Services | Location of the Analysis Services load balancing (ASLB) Web application. Internet Information Services (IIS) is required on these servers. These servers can use Network Load Balancing (NLB) for failover. For more information about using Network Load Balancing, see [Network Load Balancing Deployment Guide](http://technet.microsoft.com/en-us/library/cc754833(WS.10).aspx). |
| Load Balancing metadata db | Location of ASLB database. The ASLB database can be mirrored for failover (optional, but recommended for availability). For more information about mirroring, see [Database Mirroring Best Practices and Performance Considerations](http://sqlcat.com/whitepapers/archive/2007/11/19/database-mirroring-best-practices-and-performance-considerations.aspx). |
| AS servers with IIS and MSMDPump | Location of Microsoft® SQL Server® Analysis Services databases. IIS and MSMDPump.DLL must be installed on each of these servers. |



**Important:** For more information, including a discussion of the problems that are solved by ASLB that are not solved by other software and hardware load balancing solutions, see the Microsoft SQL Server 2008 Analysis Services Consolidation Best Practices paper on the [SQLCAT](http://www.sqlcat.com) web site.

# Getting Started

The ALSB setup files are available for download with this article.

## Unzip the ASLB Setup files

Unzip the ASLB setup files to a location of your choice. The folders listed in the following table will be created. However, there are hard-coded references in these files to C:\ASLBSetup; if you use an alternate path, you will need to modify the paths within a number of files.

|  |  |
| --- | --- |
| Scripts | Contains the Transact-SQL script to configure the ASLB database |
| ASLBMonitor | Contains the ASLBMonitor Microsoft Visual Studio® 2008 database project |
| ASLB | Contains the ASLB Web application files |

# Configuring the Analysis Services Query Servers

On each of your Analysis Services query servers, configure IIS to connect to the Analysis Services instance on that server via MSMDPump.dll.

This section contains two sets of instructions: one for configuring IIS 7.0 and IIS 7.5, and one for configuring IIS 6.0.

## Configuring IIS when you are using IIS 7.0 or IIS 7.5

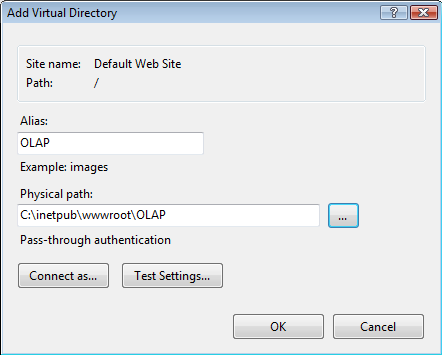
If you are using IIS 7.0 or IIS 7.5 on the Analysis Services computer, perform the following steps to configure IIS 7.0 or IIS 7.5 to use MSMDPump.dll:

* + 1. In Server Manager, add the Web Server (IIS) role, adding ASP.NET and Windows Authentication. For more information, see [Web Server (IIS)](http://technet.microsoft.com/en-us/library/cc753433(WS.10).aspx).
    2. Copy the ISAPI binaries from **..\OLAP\bin\isapi** folder to a new folder (such as ..\inetpub\wwwroot\OLAP) to use as the base for an IIS virtual directory.

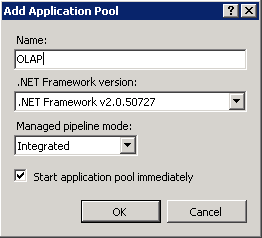
**Tip**: Make sure IIS has permissions to this folder if you do not place it in the ..\inetpub\wwwroot folder structure.

* + 1. In Internet Information Services (IIS) Manager, create a virtual directory in the default Web site based on the folder that you created in the previous step.

**Note**: You can use a site other than default port 80.

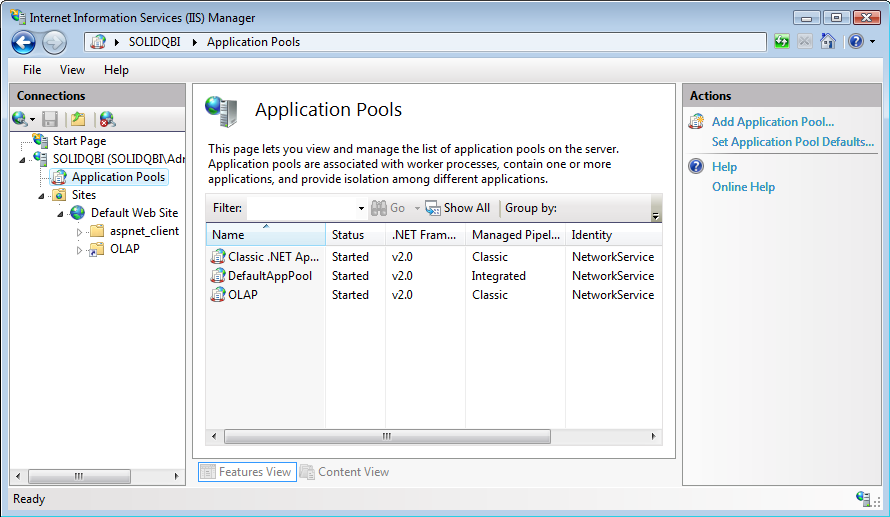


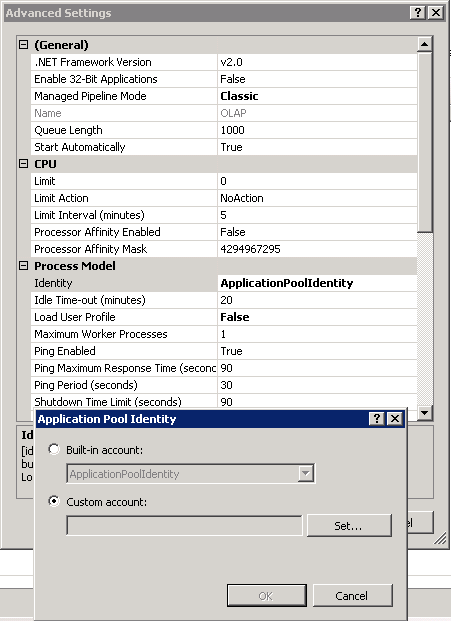
* + 1. Create a new application pool called OLAP and select **Integrated** for the managed pipeline mode. Utilizing a separate application pool for each Web application enables you to set security separately for each Web application.



* + 1. Configure the appropriate identity for your environment for the OLAP application pool in the **Advanced Settings** dialog box.

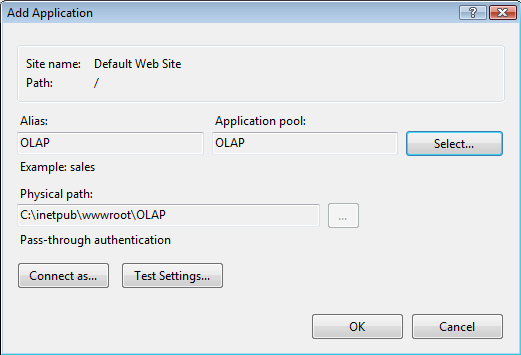
In IIS 7.0, the default application pool identity is NetworkService; in IIS 7.5, it is ApplicationPoolIdentity. ApplicationPoolIdentity was introduced in Windows Server® 2008 SP2, but not made the default. For more information about Application Pool Identities, see [Application Pool Identities](http://learn.iis.net/page.aspx/624/application-pool-identities). When using Kerberos, the identity for the OLAP application pool must be a domain user account, for which you register a Service Principal Name (SPN). The account that you use as the identity for the OLAP application pool is also the account that you will specify as the anonymous access account for HTTP connectivity to Analysis Services, and the account to which you will grant access rights and authorization permissions within Analysis Services if you elect to use anonymous access.





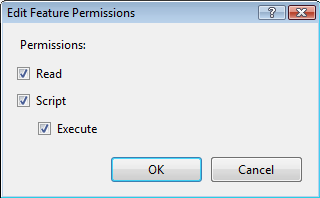
* + 1. Convert the OLAP directory to an application and specify OLAP as its application pool.

Right-click **OLAP** in the Default Web Site node and then click **Convert to Application**. Click **Select**, click **OLAP** in the Application Pool drop-down list, and then click **OK**.



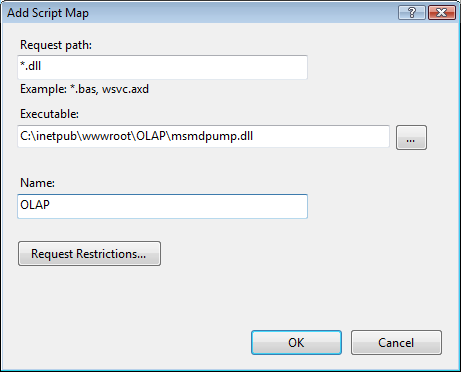
* + 1. Set directory properties for the OLAP virtual directory in IIS to enable Execute permissions.

In the Default Web Site node in the Connections pane, click **OLAP**. In the **/OLAP Home** pane (the center pane), double-click **Handler Mappings** in the IIS group. In the Actions pane, click **Edit Feature Permissions** and then select the **Execute** check box.



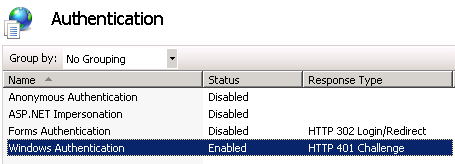
* + 1. Add the Script Map for the type \*.dll in the OLAP folder.

In the Actions pane, click **Add Script Map**. In the **Request path:** text box, type **\*.dll**. In the **Executable** text box, type **C:\inetpub\wwwroot\OLAP\msmdpump.dll** (or the appropriate path for your environment) and in the **Name** text box, type **OLAP**. Click **OK**. When you are prompted to allow this ISAPI extension, click **Yes**.



* + 1. Configure security settings for your environment for users to connect to Analysis Services. Typically, you will enable Windows Authentication and disable all other authentication types. However, for some environments, you may want to enable Anonymous Authentication and configure Analysis Services to grant the IIS anonymous user account access to Analysis Services objects.

In the Default Web Site node in the Connections pane, click **OLAP**. In the **/OLAP Home** pane (the center pane), double-click **Authentication** in the IIS group. Notice that Anonymous Authentication is enabled by default, and that all other authentication methods are disabled. If Anonymous Authentication is enabled as well as another authentication mechanism, anonymous authentication will be used.



* + 1. Enable persistent authentication by following the steps in: <http://support.microsoft.com/kb/954873>.

**Note**: Enabling persistent authentication has performance benefits in environments with slow network connections (such as long distance connectivity or RAS), but this configuration may consume slightly more memory.

## Configuring IIS when you are using IIS 6.0

If you are using IIS 6.0 on the Analysis Services computer, perform the following steps to configure IIS 6.0 to use MSMDPump.dll.

* + 1. In the Configure Your Server Wizard, add the Web Server (IIS) role, adding ASP.NET and Windows Authentication. For more information, see [Installing IIS](http://technet.microsoft.com/en-us/library/cc782498(WS.10).aspx).

**Note**: Make sure ASP is already installed – if not, install it first.

* + 1. Configure HTTP Access to Analysis Services by following the directions at: <http://technet.microsoft.com/en-us/library/cc917711.aspx>.
    2. Right-click the OLAP application pool and click Properties.
    3. On the Identity tab, configure the appropriate identity for your environment for the ASLB application pool.

In IIS 6.0, the default application pool identity is NetworkService. When using Kerberos, the identity for the OLAP application pool must be a domain user account, for which you register a Service Principal Name (SPN). The account that you use as the identity for the OLAP application pool is also the account that you will specify as the anonymous access account for HTTP connectivity to Analysis Services, and the account to which you will grant access rights and authorization permissions within Analysis Services if you elect to use anonymous access.

* + 1. If you are running Windows Server 2003 SP2 or later, enable persistent authentication by performing the following steps.

(If you are running Windows Server 2003 SP1, proceed to step 2.3.5.)

* + - * Click **Start**, click **Run**, type **regedit**, and then click **OK**.
      * Locate and then click the following registry subkey: HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters
      * On the **Edit** menu, point to **New**, and then click **DWORD Value**.
      * Type **EnableKerbAuthPersist** for the entry name, and then press ENTER.
      * Right-click **EnableKerbAuthPersist**, and then click **Modify**.
      * In the **Value data** box, type **1** or any nonzero value to enable the hotfix, and then click **OK**.

**Note** To return the IIS service to the default behavior, set the value to 0.

* + - * Exit Registry Editor.
      * Restart the IIS service.
    1. If you are running Windows Server 2003 **SP1**, follow the steps in the following article: <http://support.microsoft.com/kb/917557>, which includes a hotfix (**SP1 ONLY**).

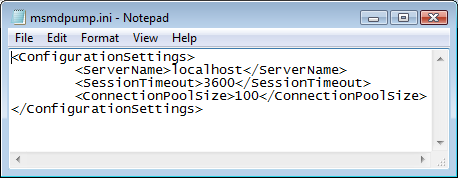
**Note**: Enabling persistent authentication has performance benefits in environments with slow network connections (such as long distance connectivity or RAS), but this configuration may consume slightly more memory.

## Configuring and Verifying MSMDPump

Configure the target Analysis Services instance and verify connectivity.

* + 1. Set the target Analysis Services server in the msmdpump.ini file.

In the Default Web Site node in the Connections pane, click **OLAP** and then click **Explore** in the Actions pane. Open the **msmdpump.ini** file in Notepad. The default value for the <ServerName> property is localhost. For a default instance, do not change this value. For a named instance, change the value to localhost\<instancename>.



* + 1. Verify connectivity via HTTP by using SQL Server Management Studio (SSMS).

Open SQL Server Management Studio and connect to each instance of Analysis Services by using HTTP. For example, to connect to Analysis Services on the local computer, type http://localhost/olap/msmdpump.dll; to connect to Analysis Services on a remote computer, type http://<machinename>/olap/msmdpump.dll.

**Troubleshooting tip**: You may need to reboot your server in some case to apply group policies to the ASP.NET client.

**Performance tip**: To enable the OLAP application pool to utilize multiple threads to service requests, increase the maximum number of worker processes in the advanced settings section for the application pool. You may need to modify this setting if you see connection failures as concurrency increases.

# Configuring the Relational Server(s)

On the primary relational server, create the ASLB database, configure and compile the ASLBMonitor project, and then execute the install scripts. Optionally, mirror to a secondary relational server for ASLB database availability.

**Important**: You must use Visual Studio 2008 to compile the ASLBMonitor project.

## Configure the SQL Server Agent service account

SQL Server Agent monitors service availability and processor performance on each Analysis Services computer to determine the computer to which to forward query requests and to notify a specified user if Analysis Services stops running. Follow these steps to configure SQL Server Agent:

* + 1. Configure the service account for SQL Server Agent to use a domain user account.

Grant the SQL Server Agent domain user account performance to access the performance monitor counters on each Analysis Services computer. At a minimum, the domain user account used for the SQL Server Agent service account should be made a member of the Performance Monitor Users local group on each remote server that it is monitoring.

* + 1. Configure SQL Server Agent to start automatically.
    2. Start SQL Server Agent.

## Creating the ALSB database on the primary server

The web service reads values from the ASLB database on the primary server.

* + 1. Create a database on your primary relational server called ASLB with a data size of 50 MB and a log size of 5 MB.
    2. Execute the AllTables.sql script in the ..\ASLBSetup\Scripts\MISC1 folder.
    3. Execute the PopulateTables.sql script in the ..\ASLBSetup\Scripts\MISC2 folder.

This script populates values in the **Parameters** table for a number of parameters that are used to setup and operate ASLB. For example, this script sets ASLB to use server load if the round robin threshold of 40 percent is exceeded. To use round robin exclusively, change the value of the UseRoundRobin parameter in this script to 1 (or in the **Parameters** table after this script has been run). To change the threshold value, change the value of the RoundRobinThreshold parameter in this script to the value of your choice (or in the **Parameters** table after this script has been run).

**Note**: By default, this script does not populate a value a value for the @NotificationList parameter. If you have configured DBMail on your relational server and want to have notifications sent to group or list of users when ASLB detects that an Analysis Server is not responsive, specify the groups and users in this script as a semicolon-separated list for the @NotificationList variable (or in the **Parameters** table after this script has been run).

**Note**: If your folder structure does not match the structure in this document, modify the PopulateTables.sql script to match your folder structure.

## Registering the Microsoft.AnalysisServices.dll assembly

Register the Microsoft.Anlaysis Services.dll assembly in the ASLB database.

* + 1. Create a new folder at the root of your C drive called ..\ASLBSetup\Assembly.

**Note**: If you use a different file location, you will need to modify the PopulateTables script.

* + 1. Copy the Microsoft.AnalysisServices.dll assembly from ..\Program Files (x86)\Microsoft SQL Server\100\Setup Bootstrap\Release\x64 to ..\ASLBSetup\Assembly.

**Important**: Do not copy the assembly from ..\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Release\x64 – this one has a different digital signature and is not the one registered in the Global Assembly Cache (GAC).

**Note**: If Analysis Services client connectivity software was not installed on this computer, install the Microsoft SQL Server 2008 Analysis Services 10.0 OLE DB Provider from the most recent feature pack.

* + 1. Execute the InstallASAssembly.sql script in the ..\ASLBSetup\Scripts\MISC3.

This script enables CLR and creates the Microsoft.AnalysisServices assembly in the ASLB database.

* + 1. Verify that Microsoft.AnalysisServices assembly appears in the Assemblies folder in the Programmability folder in the ASLB database.

## Compiling and registering the ASLBMonitor assembly

Copy the Microsoft.AnalysisServices.dll to the Assembly folder, create tables and jobs in the ASLB database, compile the ASLBMonitor.dll assembly, and then copy it to its working folder:

* + 1. Using Visual Studio 2008, open the ALSBMonitor project in the ..\ASLBSetup\ASLBMonitor folder.
    2. Open the properties of the ASLBMonitor project and, on the **Database** tab, configure the **Connection String** property to connect to the ASLB database in the primary relational server.
    3. In the Microsoft Visual Studio dialog box that appears, click **No** – you do not need to enable SQL/CLR debugging on this connection.
    4. In the References folder in the ASLBMonitor project, add a reference to the Microsoft.AnalysisServices component.
    5. Deploy the ASLBMonitor project.
    6. Verify that ASLBMonitor assembly appears in the Assemblies folder in the Programmability folder in the ASLB database.
    7. Copy ASLBMonitor.dll from the ..\ASLBSetup\ASLBMonitor\bin\Release folder to ..\ASLBSetup\Assemblies folder.

## Create stored procedures and jobs, and populate Metadata Tables

Create stored procedures and jobs for ASLB, and populate the ASLB metadata tables to operate ASLB.

* + 1. Execute all of the Transact-SQL scripts in the ..\ASLBSetup\Scripts\SP1 folder (in no particular order).

**Important**: The user account for the application pool for the ASLB Web application must execute permissions on the GetSession stored procedure.

* + 1. Execute all of the Transact-SQL scripts in the ..\ASLBSetup\Scripts\Jobs folder (in no particular order).
    2. Open the Jobs folder and verify that three new jobs appear.
    3. Open and review the PopulateMetadata script in the ..\ASLBSetup\Scripts\Extra folder.

This script populates the metadata tables in the ASLB database for load balancing across two query servers; you can easily modify it to support more query servers.

* + 1. Change the values for the Server1 and Server2 parameters to the Analysis Services server names, change the value for the test database parameter (as appropriate), and then execute the PopulateMetadata script.
    2. To verify that everything is working, select all rows from the **dbo.Server** table.

The **LoadUpdateDate** column for each specified server should not contain a null; however, it takes a few moments for the jobs to run and populate these values. If you receive nulls in this column for a server, try the following:

* + - * + Rerun the query.
        + Review the job history for the ASLB\_CheckASAvailability job; however, note that you will not see any history unless the job actually fails.
        + Execute the dbo.MonitorASAvailability stored procedure.

## Mirror ASLB database on to second relational server (optional)

For data recovery and increased availability, mirror the ASLB database to a second relational server:

* + 1. Configure the service accounts for SQL Server and SQL Server Agent to use a domain user account.

A domain user account enables the SQL Server Agent jobs to connect to remote servers to verify the availability of the Analysis Services service and the performance of the Analysis Services computer.

* + 1. Configure SQL Server Agent to start automatically.
    2. Start SQL Server Agent.
    3. (Optional) Copy the ..\ASLBSetup\Assembly folder to the identical location on the secondary.
    4. Back up the ASLB database on the primary and restore it to the secondary.
    5. Execute all of the Transact-SQL scripts in the ..\ASLBSetup\Scripts\Jobs folder (in no particular order).

# Configuring the Web Server(s)

On each Web server, configure IIS for ASLB. After each Web server is configured, set up a software or hardware network load balancer.

## Configuring IIS for ASLB when you are using IIS 7.0 or IIS 7.5

If you are using IIS 7.0 or 7.5 on the Web server, perform the following steps to configure IIS 7.0 or IIS 7.5 to use ASLB.

* + 1. Copy the ASLB folder to each Web server as the base folder for the ASLB Web application.
    2. Ensure that the IIS\_IUSRS local group has the following permissions on the ASLB folder: Read & Execute, List Folder Contents, and Read.
    3. Using Notepad, open web.config in the ASLB folder.

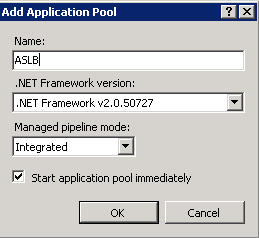
**Tip**: Use the Run as Administrator option if the ASLB folder is in the ..\inetpub\ folder and you are running on Windows Server 2008 or newer.

* + 1. Change the property values for the connectionString properties for your environment by replacing the Server1 and Server2 placeholders with the Analysis Services server names:

<add name="ASLBConnectionString" connectionString="Data Source=Server1;Failover Partner=Server2;Initial Catalog=ASLB;Integrated Security=True" providerName="System.Data.SqlClient" />

**Note**: If you are not mirroring the relational servers, remove the Failover Partner argument or, at a minimum, configure the same value for both arguments.

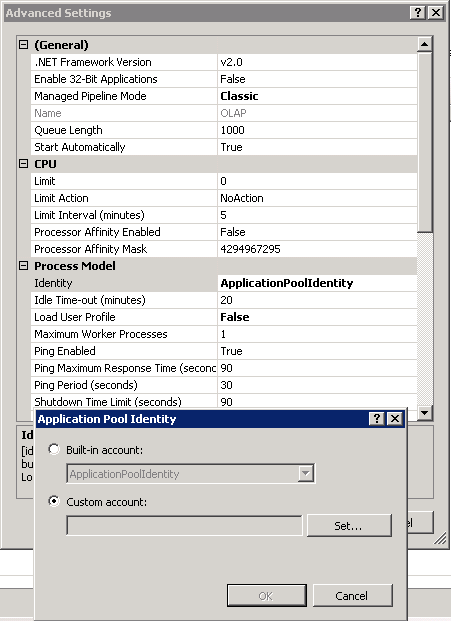
* + 1. In Internet Information Services (IIS) Manager, create a new application pool called ASLB and select **Integrated** for the managed pipeline mode.



* + 1. Configure the appropriate identity for your environment for the ASLB application pool in the **Advanced Settings** dialog box.

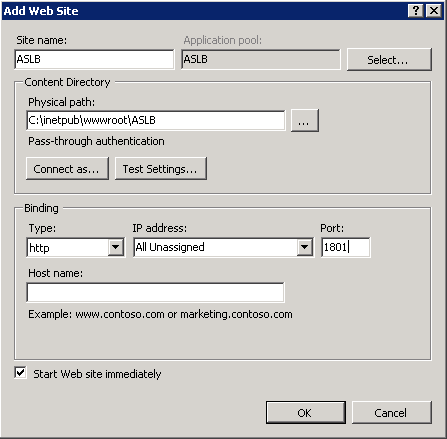
The identity for this application pool will be a user account that will access the data in the ASLB database to retrieve the necessary information to route user requests to the appropriate Analysis Services instance. The default application pool identity in IIS 7.0 is NetworkService, and in IIS 7.5 it is ApplicationPoolIdentity. ApplicationPoolIdentity was introduced in Windows Server 2008 SP2, but not made the default. For more information about Application Pool Identities, see [Application Pool Identities](http://learn.iis.net/page.aspx/624/application-pool-identities). For database access, you can grant access to the Web server’s machine account. However, for security reasons, you will generally want to use a dedicated domain user account.

**Important**: The user account for the application pool for the ASLB Web application must execute permissions on the GetSession stored procedure that you created in step 3.5.1.

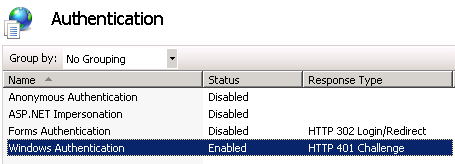


* + 1. Add a new Web site called ASLB. Specify the path to the ASLB folder as the physical path, and specify an unused port (such as 1801) for the port binding.

**Important:** You are creating a new site because the ASLB application has its own configuration file.

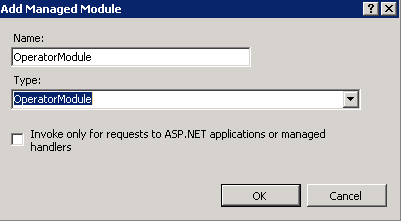


* + 1. Configure security settings for your environment to disable Anonymous Authentication and enable Windows Authentication.



* + 1. In the ASLB site, add the OperatorModule as a managed module.

In the Site node in the Connections pane, click **ASLB**. In the **ASLB** pane (the center pane), double-click **Modules** and then, in the Actions pane, click **Add Managed Module**.



* + 1. Enable persistent authentication by following the steps in: <http://support.microsoft.com/kb/954873>.

**Note**: Enabling persistent authentication has performance benefits in environments with slow network connections (such as long distance connectivity or RAS), but this configuration may consume slightly more memory.

## Configuring IIS for ASLB when you are using IIS 6.0

If you are using IIS 6.0 on the Web server, perform the following steps to configure IIS 6.0 to use ASLB.

* + 1. Copy the ASLB folder to each Web server as the base folder for the ASLB Web application.
    2. Ensure that the IIS\_IUSRS local group has the following permissions on the ASLB folder: Read & Execute, List Folder Contents, and Read.
    3. Using Notepad, open web.config in the ASLB folder.

**Tip**: Use the Run as Administrator option if the ASLB folder is in the ..\inetpub\ folder and you are running on Windows Server 2008 or newer.

* + 1. Change the property values for the connectionString properties for your environment by replacing the Server1 and Server2 placeholders with the Analysis Services server names:

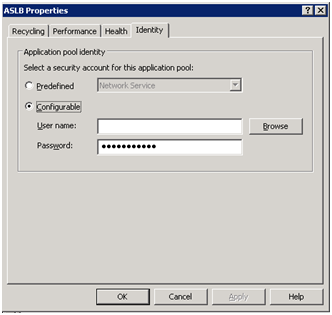
<add name="ASLBConnectionString" connectionString="Data Source=Server1;Failover Partner=Server2;Initial Catalog=ASLB;Integrated Security=True" providerName="System.Data.SqlClient" />

**Note**: If you are not mirroring the relational servers, remove the Failover Partner argument or, at a minimum, configure the same value for both arguments.

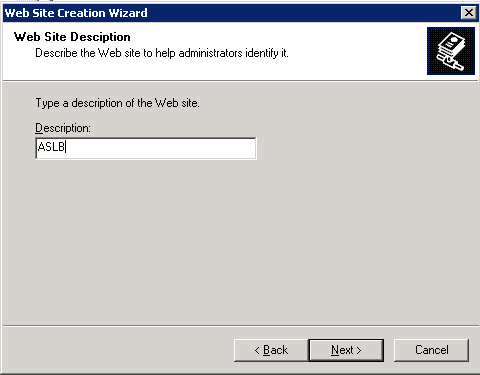
* + 1. In Internet Information Services (IIS) Manager, create a new application pool called ASLB.
    2. Right-click the ASLB application pool and click Properties.
    3. On the Identity tab, configure the appropriate identity for your environment for the ASLB application pool.

The identity for this application pool will be a user account that will access the data in the ASLB database to retrieve the necessary information to route user requests to the appropriate Analysis Services instance. The default application pool identity in IIS 6.0 is NetworkService. For database access, you can grant access to the Web server’s machine account. However, for security reasons, you will generally want to use a dedicated domain user account.

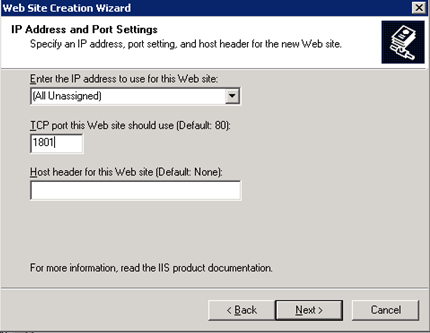
**Important**: The user account for the application pool for the ASLB Web application must execute permissions on the GetSession stored procedure that you created in step 3.5.1.



* + 1. Add a new Web site called ASLB.

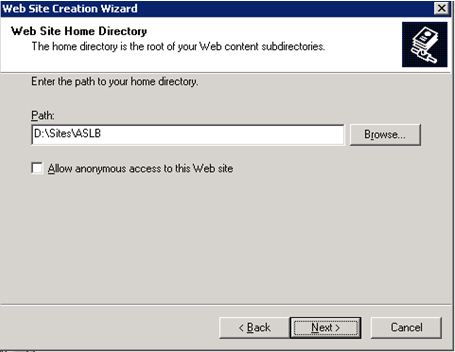


* + 1. Specify an unused port (such as 1801) for the port binding.

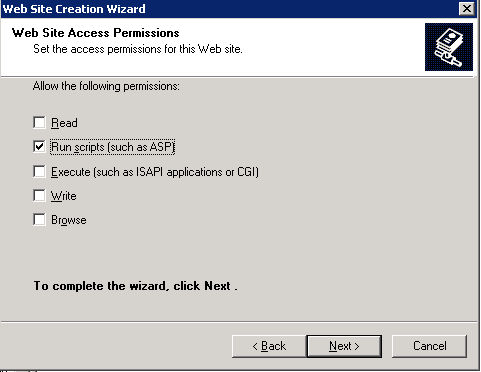


**Important:** You are creating a new site because the ASLB application has its own configuration file.

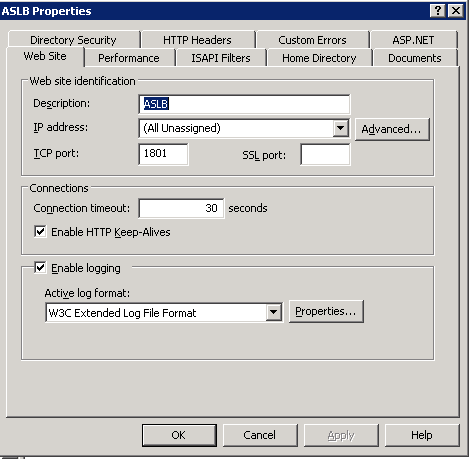
* + 1. Specify the path to the ASLB folder as the physical path, and clear the Allow anonymous access to this Web site checkbox.



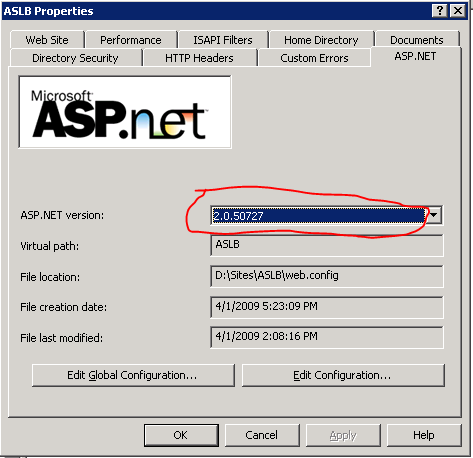
* + 1. Click Next and then check the Run Scripts checkbox.



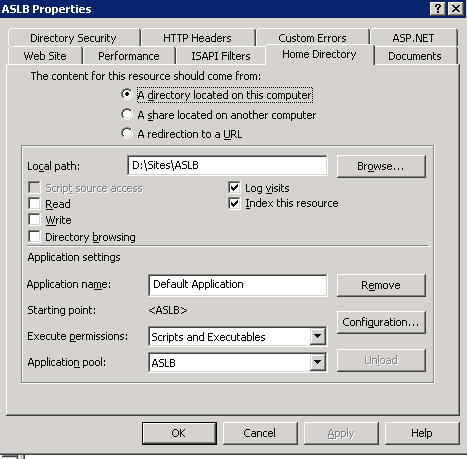
* + 1. Click Next and click Finished.
    2. Right-click the ASLB web site and click Properties.
    3. Verify that settings on the Web Site tab match the screenshot below.



* + 1. Click the ASP.Net tab and verify the settings match the screenshot below. You may need to modify the ASP.NET version on this tab.

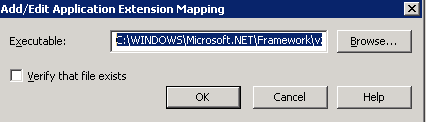


* + 1. Click Apply.
    2. Click the Home Directory tab. In the Application pool drop-down list box, select ASLB and then verify the settings match the screenshot below.

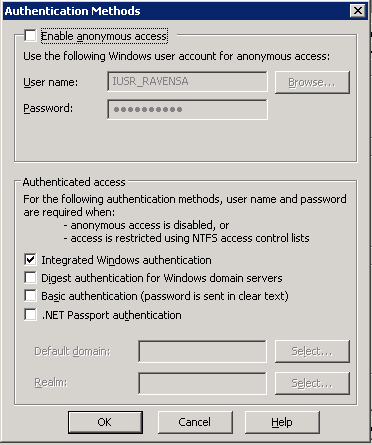


* + 1. Click Configuration and, under Wildcard application maps, click Insert.
    2. Clear the Verify that file exists checkbox and type the following path:

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet\_isapi.dll



* + 1. Click OK.
    2. Click the Directory Security tab, and, under Authentication and access control, click Edit. Verify the settings match the screenshot below.



* + 1. Click OK, and then click OK again to exit the ASLB properties window.
    2. If you are running Windows Server 2003 SP2 or later, enable persistent authentication by performing the following steps.

If you are running Windows Server 2003 SP1, proceed to step 4.2.24.)

* + - * Click **Start**, click **Run**, type **regedit**, and then click **OK**.
      * Locate and then click the following registry subkey: HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters
      * On the **Edit** menu, point to **New**, and then click **DWORD Value**.
      * Type **EnableKerbAuthPersist** for the entry name, and then press ENTER.
      * Right-click **EnableKerbAuthPersist**, and then click **Modify**.
      * In the **Value data** box, type **1** or any nonzero value to enable the hotfix, and then click **OK**.

**Note** To return the IIS service to the default behavior, set the value to 0.

* + - * Exit Registry Editor.
      * Restart the IIS service.
    1. If you are running Windows Server 2003 **SP1**, follow the steps in the following article: <http://support.microsoft.com/kb/917557>, which includes a hotfix (**SP1 ONLY**).

**Note**: Enabling persistent authentication has performance benefits in environments with slow network connections (such as long distance connectivity or RAS), but this configuration may consume slightly more memory.

# Testing

Install and test using the Adventure Works DW 2008 Enterprise database or another database of your choice.

* 1. Install one of the sample Adventure Works DW 2008 Analysis Services databases from Codeplex at: [Sample Databases](http://msftdbprodsamples.codeplex.com/).
  2. Using SQL Server Management Studio, connect to one of your query servers by connecting to the ASLB Web application on your Web server with a connection string of http://<servername>:<portnumber>/<testdatabasename>.
  3. To verify the server to which the ASLB Web application redirected you, select all rows from dbo.Session.
  4. Stop the Analysis Services service on the query server to which SQL Server Management Studio was redirected and then reconnect to the ASLB Web application on your Web server using the same connection string.
  5. To verify the server to which the ASLB Web application redirected you, select all rows from dbo.Session. If your installation was successful, you are redirected to another Web server.