

# ASP.NET SQL Server Registration Tool

## (Aspnet\_regsql.exe)



### .NET Framework 4

8 out of 38 rated this helpful - [Rate this topic](#)

The ASP.NET SQL Server registration tool (Aspnet\_regsql.exe) is used to create a Microsoft SQL Server database that is used by the SQL Server providers in ASP.NET. The tool is also used to add or remove options from an existing database.

You can run Aspnet\_regsql.exe without any command-line arguments to run a wizard that will walk you through specifying connection information for your SQL Server installation, and installing or removing the database elements for the membership, role management, profile, Web Parts personalization, and health monitoring features.

You can also run Aspnet\_regsql.exe as a command-line tool to specify database elements for individual features to add or remove, using the options listed in the table below.

#### Note

For information about how to find the correct version of Aspnet\_regsql.exe, see [Finding the Correct Version of Aspnet\\_regsql.exe](#) later in this topic.

```
Aspnet_regsql.exe <options>
```

## SQL Connection Options

Option	Description
<b>-?</b>	Prints Aspnet_regsql.exe Help text in the command window.
<b>-W</b>	Runs the tool in wizard mode. This is the default if no command-line arguments are specified.
<b>-C</b> <connection string>	Specifies the connection string to the computer running SQL Server where the database will be installed, or is already installed. This option is not necessary if you specify only the server ( <b>-S</b> ) and login ( <b>-U</b> and <b>-P</b> , or <b>-E</b> ) information.
<b>-S</b> <server>	Specifies the name of the computer running SQL Server where the database will be installed, or is already installed.
<b>-U</b> <login ID>	The SQL Server user ID to log in with. This option also requires the password ( <b>-P</b> ) option. This option is not necessary if you are authenticating using Windows credentials ( <b>-E</b> ).

<b>-P</b> <password>	The SQL Server password to log in with. This option also requires the user ID ( <b>-U</b> ) option. This option is not necessary if you are authenticating using Windows credentials ( <b>-E</b> ).
<b>-E</b>	Authenticates using the Windows credentials of the currently logged-on user.
<b>-sqllexportonly</b> <filename>	Generates a SQL script file that can be used to add or remove the specified features. The specified actions are not performed.

## Application Services Options

Option	Description
<b>-A</b> <i>all m r p c w</i>	<p>Adds support for one or more ASP.NET application services. Service identifiers can be specified together or separately. The following identifiers are used for ASP.NET application services:</p> <p><b>all</b> - All services, including common tables and stored procedures shared by the services</p> <p><b>m</b> - Membership</p> <p><b>r</b> - Role Manager</p> <p><b>p</b> - Profile</p> <p><b>c</b> - Web Parts Personalization</p> <p><b>w</b> - Web Events</p>
<b>-R</b> <i>all m r p c w</i>	<p>Removes support for one or more application services. Service identifiers can be specified together or separately. The following identifiers are used for ASP.NET application services:</p> <p><b>all</b> - All services, including common tables and stored procedures shared by the services</p> <p><b>m</b> - Membership</p> <p><b>r</b> - Role Manager</p> <p><b>p</b> - Profile</p> <p><b>c</b> - Web Parts Personalization</p> <p><b>w</b> - Web Events</p>
<b>-Q</b>	Runs the tool in quiet mode and does not confirm before removing an application service.

<b>-d</b> <database>	Specifies the name of the database to create or modify for use with application services. If the database is not specified, the default database name of "aspnetdb" is used.
<b>Note</b>  The database elements that are installed in the application services database will always be owned by the SQL Server database owner account (dbo). You do not need to be a system administrator for the computer running SQL Server to install the application services database. To install the application services database, a SQL Server login must be permitted to the <b>db_ddladmin</b> and <b>db_securityadmin</b> roles for the SQL Server database.	

## Session State Options

Option	Description
<b>-d</b> <database>	Specifies the name of the database to store session state. This option must be used if <b>-sstype</b> is set to <b>c</b> .
<b>-ssadd</b>	Adds support for SQL Server mode session state.
<b>-ssremove</b>	Removes support for SQL Server mode session state.
<b>-sstype</b> t p c	Specifies the type of session state to use:  <b>t</b> - Temporary. Session state data is stored in the SQL Server tempdb database. Stored procedures for managing session state are installed in the SQL Server ASPState database. Data is not persisted if you restart SQL. This is the default.  <b>p</b> - Persisted. Both session state data and stored procedures are stored in the SQL Server ASPState database.  <b>c</b> - Custom. Both session state data and stored procedures are stored in a custom database. The database name must be specified using the <b>-d</b> option.

## Remarks

You can set several types of options with the ASP.NET SQL Server Registration tool. You can specify a SQL connection, specify which ASP.NET application services use SQL Server to manage information, and add or remove support for using SQL Server to store procedures and session state.

Several ASP.NET application services rely on a provider to manage storing and retrieving data from a data source. Each provider is specific to the data source. ASP.NET includes a SQL Server provider for the following ASP.NET features:

- Membership (the [SqlMembershipProvider](#) class).
- Role management (the [SqlRoleProvider](#) class).
- Profile (the [SqlProfileProvider](#) class).

- Web Parts personalization (the [SqlPersonalizationProvider](#) class).
- Web events (the [SqlWebEventProvider](#) class).

When you install ASP.NET, the Machine.config file for your server includes configuration elements that specify SQL Server providers for each of the ASP.NET features that rely on a provider. These providers are configured, by default, to connect to a local user instance of SQL Server Express. If you change the default connection string used by the providers, then before you can use any of the ASP.NET features configured in the machine configuration, you must install the SQL Server database and the database elements for your chosen feature using Aspnet\_regsql.exe. If the database that you specify with Aspnet\_regsql.exe does not already exist (aspnetdb will be the default database if one is not specified on the command line), then the current user must have rights to create databases in SQL Server as well as to create schema elements within a database.

Each SQL Server provider can use the same SQL Server database to store the data for that particular feature. Each feature can be used individually, or in conjunction with other features. For example, you could use role management by itself or in conjunction with the user information managed by membership.

For more information about SQL Server providers and ASP.NET, see [Implementing a Membership Provider](#), [Implementing a Profile Provider](#), and [Implementing a Role Provider](#).

## Session State

ASP.NET session state is designed to enable you to easily store user session data in different sources for your ASP.NET applications. By default, session state values and information are stored in memory within the ASP.NET process. Alternatively, you can store session data in a SQL Server database, where it can be shared by multiple Web servers. For more information about session state, see [Implementing a Session-State Store Provider](#) and [Session-State Modes](#).

If the database that you specify for session state with Aspnet\_regsql.exe does not already exist, then the current user must have rights to create databases in SQL Server as well as to create schema elements within a database. If the database does exist, then the current user must have rights to create schema elements in the existing database.

To install the session state database on SQL Server, run Aspnet\_regsql.exe tool supply the following information with the command:

- The name of the SQL Server instance, using the **-S** option.
- The logon credentials for an account that has permission to create a database on a computer running SQL Server. Use the **-E** option to use the currently logged-on user, or use the **-U** option to specify a user ID along with the **-P** option to specify a password.
- The **-ssadd** command-line option to add the session state database.

By default, you cannot use Aspnet\_regsql.exe to install the session state database on a computer running SQL Server Express. For more information about using SQL Server Express to store session state, see [Session-State Modes](#).

## Examples

You can run Aspnet\_regsql.exe without any command-line arguments to run a wizard that will walk you through specifying connection information for your SQL Server database and installing or removing the database elements for supported features.

You can also run Aspnet\_regsql.exe as a command-line tool to specify database elements for individual features to add or remove. To set up session state, you must use the command-line tool; the wizard will not set up session state.

To run the wizard, run Aspnet\_regsql.exe without any command-line arguments, as shown in the

following example.

```
aspnet_regsql.exe
```

The ASP.NET SQL Registration tool is commonly used with the **-A** or **-R** option to specify which features use a SQL Server provider. The **-A** option allows you to add support for one or more features, whereas the **-R** option allows you to remove a feature. The following command installs the database elements for membership and role management on the local computer running SQL Server using Windows authentication.

```
aspnet_regsql.exe -E -S localhost -A mr
```

The following command creates a database named ASPState on a SQL Server instance named "SampleSqlServer" and specifies that session data is also stored in the ASPState database.

```
aspnet_regsql.exe -S SampleSqlServer -E -ssadd -sstype p
```

## Finding the Correct Version of Aspnet\_regsql.exe

Aspnet\_regsql.exe is installed in the Microsoft.NET Framework directory. If the computer is running multiple .NET Framework versions side-by-side, multiple versions of the tool might be installed. The following table lists the locations where the tool is installed for different versions of the .NET Framework.

Version of .NET Framework	Location of Aspnet_regsql.exe file
.NET Framework version 2.0, version 3.0, and version 3.5 (32-bit systems)	%windir%\Microsoft.NET\Framework\v2.0.50727
.NET Framework version 2.0, version 3.0, and version 3.5 (64-bit systems)	%windir%\Microsoft.NET\Framework64\v2.0.50727
.NET Framework version 4 (32-bit systems)	%windir%\Microsoft.NET\Framework\v4.0.30319
.NET Framework version 4 (64-bit systems)	%windir%\Microsoft.NET\Framework64\v4.0.30319

## See Also

### Concepts

[Implementing a Membership Provider](#)

[Implementing a Profile Provider](#)

[Implementing a Role Provider](#)

Did you find this helpful?



Yes



No

## Community Content

### Example

Example where Windows authentication is indicated with -E, a server is specified with -S, a database with -d and a table with -t

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>aspnet_regsql.exe -S HP1\SQLEXPRESS08R2 -E -ed -d IWSqlCacheDependency_790 -et -t Banners
```

Enabling the database for SQL cache dependency.

Finished.

Enabling the table for SQL cache dependency.

Finished.

7/3/2012  
SartoriusN



© 2012 Microsoft. All rights reserved.