**ASP.NET Session State Overview**

This page is specific to

**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:

[Microsoft Visual Studio 2005/.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/ms178581(VS.80).aspx)

[.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/ms178581(VS.85).aspx)

[Microsoft Visual Studio 2010/.NET Framework 4](http://msdn.microsoft.com/en-us/library/ms178581(VS.100).aspx)

Use ASP.NET session state to store and retrieve values for a user.

This topic contains:

* [Background](http://msdn.microsoft.com/en-us/library/ms178581.aspx#Background)
* [Code Examples](http://msdn.microsoft.com/en-us/library/ms178581.aspx#CodeExamples)
* [Class Reference](http://msdn.microsoft.com/en-us/library/ms178581.aspx#ClassReference)

 Background

ASP.NET session state enables you to store and retrieve values for a user as the user navigates ASP.NET pages in a Web application. HTTP is a stateless protocol. This means that a Web server treats each HTTP request for a page as an independent request. The server retains no knowledge of variable values that were used during previous requests. ASP.NET session state identifies requests from the same browser during a limited time window as a session, and provides a way to persist variable values for the duration of that session. By default, ASP.NET session state is enabled for all ASP.NET applications.

Alternatives to session state include the following:

* Application state, which stores variables that can be accessed by all users of an ASP.NET application.
* Profile properties, which persists user values in a data store without expiring them.
* ASP.NET caching, which stores values in memory that is available to all ASP.NET applications.
* View state, which persists values in a page.
* Cookies.
* The query string and fields on an HTML form that are available from an HTTP request.

For a comparison of different state-management options, see [ASP.NET State Management Recommendations](http://msdn.microsoft.com/en-us/library/z1hkazw7.aspx).

### Session Variables

Session variables are stored in a [SessionStateItemCollection](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateitemcollection.aspx) object that is exposed through the [HttpContext..::.Session](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.session.aspx) property. In an ASP.NET page, the current session variables are exposed through the Session property of the Page object.

The collection of session variables is indexed by the name of the variable or by an integer index. Session variables are created by referring to the session variable by name. You do not have to declare a session variable or explicitly add it to the collection. The following example shows how to create session variables in an ASP.NET page for the first and last name of a user, and set them to values retrieved from [TextBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.textbox.aspx) controls.

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl75_ctl00_ctl04_code');" \o "Copy Code)

Session("FirstName") = FirstNameTextBox.Text

Session("LastName") = LastNameTextBox.Text

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl75_ctl00_ctl05_code');" \o "Copy Code)

Session["FirstName"] = FirstNameTextBox.Text;

Session["LastName"] = LastNameTextBox.Text;

Session variables can be any valid .NET Framework type. The following example stores an [ArrayList](http://msdn.microsoft.com/en-us/library/system.collections.arraylist.aspx) object in a session variable named StockPicks. The value returned by the StockPicks session variable must be cast to the appropriate type when you retrieve it from the [SessionStateItemCollection](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateitemcollection.aspx).

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl75_ctl00_ctl08_code');" \o "Copy Code)

' When retrieving an object from session state, cast it to

' the appropriate type.

Dim stockPicks As ArrayList = CType(Session("StockPicks"), ArrayList)

' Write the modified stock picks list back to session state.

Session("StockPicks") = stockPicks

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl75_ctl00_ctl09_code');" \o "Copy Code)

// When retrieving an object from session state, cast it to

// the appropriate type.

ArrayList stockPicks = (ArrayList)Session["StockPicks"];

// Write the modified stock picks list back to session state.

Session["StockPicks"] = stockPicks;

|  |
| --- |
| **NoteNote:** |
| When you use a session-state mode other than [InProc](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemode.inproc.aspx), the session-variable type must be either a primitive .NET type or serializable. This is because the session-variable value is stored in an external data store. For more information, see [Session-State Modes](http://msdn.microsoft.com/en-us/library/ms178586.aspx). |

### Session Identifiers

Sessions are identified by a unique identifier that can be read by using the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) property. When session state is enabled for an ASP.NET application, each request for a page in the application is examined for a [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value sent from the browser. If no [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value is supplied, ASP.NET starts a new session and the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value for that session is sent to the browser with the response.

By default, [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) values are stored in a cookie. However, you can also configure the application to store [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) values in the URL for a "cookieless" session.

A session is considered active as long as requests continue to be made with the same [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value. If the time between requests for a particular session exceeds the specified time-out value in minutes, the session is considered expired. Requests made with an expired [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value result in a new session.

|  |
| --- |
| **Security noteSecurity Note:** |
| P:System.Web.SessionState.HttpSessionState.SessionID values are sent in clear text, whether as a cookie or as part of the URL. A malicious user could get access to the session of another user by obtaining the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value and including it in requests to the server. If you are storing sensitive information in session state, it is recommended that you use SSL to encrypt any communication between the browser and server that includes the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value. |

#### Cookieless SessionIDs

By default, the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value is stored in a non-expiring session cookie in the browser. However, you can specify that session identifiers should not be stored in a cookie by setting the cookieless attribute to true in the [sessionState](http://msdn.microsoft.com/en-us/library/h6bb9cz9.aspx) section of the Web.config file.

The following example shows a Web.config file that configures an ASP.NET application to use cookieless session identifiers.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl75_ctl00_ctl26_code');" \o "Copy Code)

<configuration>

<system.web>

<sessionState cookieless="true"

regenerateExpiredSessionId="true" />

</system.web>

</configuration>

ASP.NET maintains cookieless session state by automatically inserting a unique session ID into the page's URL. For example, the following URL has been modified by ASP.NET to include the unique session ID lit3py55t21z5v55vlm25s55:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl75_ctl00_ctl27_code');" \o "Copy Code)

http://www.example.com/(S(lit3py55t21z5v55vlm25s55))/orderform.aspx

When ASP.NET sends a page to the browser, it modifies any links in the page that use an application-relative path by embedding a session ID value in the links. (Links with absolute paths are not modified.) Session state is maintained as long as the user clicks links that have been modified in this manner. However, if the client rewrites a URL that is supplied by the application, ASP.NET may not be able to resolve the session ID and associate the request with an existing session. In that case, a new session is started for the request.

The session ID is embedded in the URL after the slash that follows the application name and before any remaining file or virtual directory identifier. This enables ASP.NET to resolve the application name before involving the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) in the request.

|  |
| --- |
| **NoteNote:** |
| To improve the security of your application, you should allow users to log out of your application, at which point the application should call the [Abandon](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstatecontainer.abandon.aspx) method. This reduces the potential for a malicious user to get the unique identifier in the URL and use it to retrieve private user data stored in the session. |

#### Regenerating Expired Session Identifiers

By default, the session ID values that are used in cookieless sessions are recycled. That is, if a request is made with a session ID that has expired, a new session is started by using the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value that is supplied with the request. This can result in a session unintentionally being shared when a link that contains a cookieless [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value is used by multiple browsers. (This can occur if the link is passed through a search engine, through an e-mail message, or through another program.) You can reduce the chance of session data being shared by configuring the application not to recycle session identifiers. To do this, set the regenerateExpiredSessionId attribute of the [sessionState](http://msdn.microsoft.com/en-us/library/h6bb9cz9.aspx) configuration element to true. This generates a new session ID when a cookieless session request is made with an expired session ID.

|  |
| --- |
| **NoteNote:** |
| If the request that is made with the expired session ID is made by using the HTTP POST method, any posted data will be lost when regenerateExpiredSessionId is true. This is because ASP.NET performs a redirect to make sure that the browser has the new session identifier in the URL. |

#### Custom Session Identifiers

You can implement a custom class to supply and validate [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) values. To do so, create a class that inherits the [SessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.aspx) class and override the [CreateSessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.createsessionid.aspx) and [Validate](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.validate.aspx) methods with your own implementations. For an example, see the example provided for the [CreateSessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.createsessionid.aspx) method.

You can replace the [SessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.aspx) class by creating a class that implements the [ISessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.isessionidmanager.aspx) interface. For example, you might have a Web application that associates a unique identifier with non-ASP.NET pages (such as HTML pages or images) by using an ISAPI filter. You can implement a custom [SessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.aspx) class to use this unique identifier with ASP.NET session state. If your custom class supports cookieless session identifiers, you must implement a solution for sending and retrieving session identifiers in the URL.

### Session Modes

ASP.NET session state supports several storage options for session variables. Each option is identified as a session-state [Mode](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.mode.aspx) type. The default behavior is to store session variables in the memory space of the ASP.NET worker process. However, you can also specify that session state should be stored in a separate process, in a SQL Server database, or in a custom data source. If you do not want session state enabled for your application, you can set the session mode to [Off](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemode.off.aspx).

For more information, see [Session-State Modes](http://msdn.microsoft.com/en-us/library/ms178586.aspx).

### Session Events

ASP.NET provides two events that help you manage user sessions. The **Session\_OnStart** event is raised when a new session starts, and the **Session\_OnEnd** event is raised when a session is abandoned or expires. Session events are specified in the Global.asax file for an ASP.NET application.

The **Session\_OnEnd** event is not supported if the session [Mode](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.mode.aspx) property is set to a value other than [InProc](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemode.inproc.aspx), which is the default mode.

|  |
| --- |
| **NoteNote:** |
| If the Global.asax file or Web.config file for an ASP.NET application is modified, the application will be restarted and any values stored in application state or session state will be lost. Be aware that some anti-virus software can update the last-modified date and time of the Global.asax or Web.config file for an application. |

For more information, see [Session-State Events](http://msdn.microsoft.com/en-us/library/ms178583.aspx).

### Configuring Session State

Session state is configured by using the [sessionState](http://msdn.microsoft.com/en-us/library/h6bb9cz9.aspx) element of the system.web configuration section. You can also configure session state by using the [EnableSessionState](http://msdn.microsoft.com/en-us/library/system.web.configuration.pagessection.enablesessionstate.aspx) value in the @ Page directive.

The sessionState element enables you to specify the following options:

* The mode in which the session will store data.
* The way in which session identifier values are sent between the client and the server.
* The session [Timeout](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.timeout.aspx) value.
* Supporting values that are based on the session [Mode](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.mode.aspx) setting.

The following example shows a sessionState element that configures an application for [SQLServer](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemode.sqlserver.aspx) session mode. It sets the [Timeout](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.timeout.aspx) value to 30 minutes, and specifies that session identifiers are stored in the URL.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl75_ctl00_ctl56_code');" \o "Copy Code)

<sessionState mode="SQLServer"

cookieless="true "

regenerateExpiredSessionId="true "

timeout="30"

sqlConnectionString="Data Source=MySqlServer;Integrated Security=SSPI;"

stateNetworkTimeout="30"/>

You can disable session state for an application by setting the session-state mode to [Off](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemode.off.aspx). If you want to disable session state for only a particular page of an application, you can set the [EnableSessionState](http://msdn.microsoft.com/en-us/library/system.web.configuration.pagessection.enablesessionstate.aspx) value in the @ Page directive to false. The [EnableSessionState](http://msdn.microsoft.com/en-us/library/system.web.configuration.pagessection.enablesessionstate.aspx) value can also be set to ReadOnly to provide read-only access to session variables.

### Concurrent Requests and Session State

Access to ASP.NET session state is exclusive per session, which means that if two different users make concurrent requests, access to each separate session is granted concurrently. However, if two concurrent requests are made for the same session (by using the same [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value), the first request gets exclusive access to the session information. The second request executes only after the first request is finished. (The second session can also get access if the exclusive lock on the information is freed because the first request exceeds the lock time-out.) If the [EnableSessionState](http://msdn.microsoft.com/en-us/library/system.web.configuration.pagessection.enablesessionstate.aspx) value in the @ Page directive is set to ReadOnly, a request for the read-only session information does not result in an exclusive lock on the session data. However, read-only requests for session data might still have to wait for a lock set by a read-write request for session data to clear.

[Back to top](http://msdn.microsoft.com/en-us/library/ms178581.aspx#Introduction)

 Code Examples

[How to: Save Values in Session State](http://msdn.microsoft.com/en-us/library/6ad7zeeb.aspx)

[How to: Read Values from Session State](http://msdn.microsoft.com/en-us/library/03sekbw5.aspx)

[Implementing a Session-State Store Provider](http://msdn.microsoft.com/en-us/library/ms178587.aspx)

[Back to top](http://msdn.microsoft.com/en-us/library/ms178581.aspx#Introduction)

 Class Reference

The following table lists key classes that relate to session state are in the [System.Web.SessionState](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.aspx) namespace.

|  |  |
| --- | --- |
| **Member** | **Description** |
| [SessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.aspx) | Manages unique identifiers for ASP.NET session state. |
| [SessionStateItemCollection](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateitemcollection.aspx) | Used to store session state variables. |

**How to: Save Values in Session State**

This example uses the [HttpSessionState](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.aspx) object to persist values within an individual session.

 Example

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl17_ctl00_ctl00_code');" \o "Copy Code)

Dim firstName As String = "John"

Dim lastName As String = "Smith"

Dim city As String = "Seattle"

Session("FirstName") = firstName

Session("LastName") = lastName

Session("City") = city

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl17_ctl00_ctl01_code');" \o "Copy Code)

string firstName = "Jeff";

string lastName = "Smith";

string city = "Seattle";

Session["FirstName"] = firstName;

Session["LastName"] = lastName;

Session["City"] = city;

 Compiling the Code

This example requires:

* A Web Forms page or class that has access to the current request context using the [Current](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.current.aspx) property in an ASP.NET application that has session state enabled.

 Robust Programming

Session state can expire (by default, after 20 minutes of inactivity), and the information that you store there can be lost. You can control session-state lifetime using the timeout attribute of the [sessionState](http://msdn.microsoft.com/en-us/library/h6bb9cz9.aspx) configuration section.

Depending on your application requirements, you may want to consider an alternative to session state for storing information for each user. ASP.NET provides several other options for persisting data within an application. For a comparison of each, see [ASP.NET State Management Recommendations](http://msdn.microsoft.com/en-us/library/z1hkazw7.aspx).

**How to: Read Values from Session State**

This example accesses the [Item](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.item.aspx) property to retrieve the values in session state.

 Example

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl19_ctl00_ctl00_code');" \o "Copy Code)

Dim firstName as String = CType(Session.Item("FirstName"), String)

Dim lastName as String = CType(Session.Item("LastName"), String)

Dim city as String = CType(Session.Item("City"), String)

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl19_ctl00_ctl01_code');" \o "Copy Code)

string firstName = (string)(Session["First"]);

string lastName = (string)(Session["Last"]);

string city = (string)(Session["City"]);

 Compiling the Code

This example requires:

* A Web Forms page or class that has access to the current request context using the [Current](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.current.aspx) property in an ASP.NET application that has session state enabled.

 Robust Programming

No exception is thrown if you attempt to get a value out of session state that does not exist. To be sure that the value you want is in session state, check first for the existence of the object with a test such as the following:

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl21_ctl00_ctl00_code');" \o "Copy Code)

If Session.Item("FirstName") Is Nothing Then

' No such value in session state, take appropriate action.

End If

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl21_ctl00_ctl01_code');" \o "Copy Code)

if (Session["City"] == null)

// No such value in session state; take appropriate action.

If you attempt to use a nonexistent session state entry in some other way (for example, to examine its type), a [NullReferenceException](http://msdn.microsoft.com/en-us/library/system.nullreferenceexception.aspx) exception is thrown.

Session values are of type [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx). In Visual Basic, if you set Option Strict On, you must cast from type [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx) to the appropriate type when getting values out of session state, as shown in the example. In C#, you should always cast to the appropriate type when reading session values.

**Implementing a Session-State Store Provider**

Describes a custom session-state store provider implementation and demonstrates implementing a sample provider.

ASP.NET session state is designed to enable you to store user session data in different sources. By default, session state values and information are stored in memory within the ASP.NET process. One alternative is to store session data in a state server, which keeps session data in a separate process and retains it if the ASP.NET application is shut down and restarted. Another alternative is to store session data in a SQL Server database, where it can be shared by multiple Web servers.

You can use the session-state stores that are included with ASP.NET, or you can implement your own session-state store provider. You might create a custom session-state store provider for the following reasons:

* You need to store session-state information in a data source other than SQL Server, such as a FoxPro database or an Oracle database.
* You need to manage session-state information using a database schema that is different from the database schema used by the providers that ship with the .NET Framework. An example of this would be shopping cart data that is stored with a predefined schema in your existing SQL Server database.

You can implement a custom session-state store provider by creating a class that inherits the [SessionStateStoreProviderBase](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.aspx) class. For more information, see the "Required Classes" section later in this topic.

 The Session State Module

Session state is managed by the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) class, which calls the session-state store provider to read and write session data to the data store at different times during a request. At the beginning of a request, the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) instance retrieves data from the data source by calling the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method, or if the [EnableSessionState](http://msdn.microsoft.com/en-us/library/system.web.configuration.pagessection.enablesessionstate.aspx) page attribute has been set to **ReadOnly**, by calling the [GetItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitem.aspx) method. At the end of a request, if the session-state values have been modified, the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) instance calls the [SessionStateStoreProviderBase..::.SetAndReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setandreleaseitemexclusive.aspx) method to write the updated values to the session-state store. [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) calls additional members of the [SessionStateStoreProviderBase](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.aspx) implementation to initialize a new session as well as to delete session data from the data store when the [HttpSessionState..::.Abandon](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.abandon.aspx) method is called. Each member of the [SessionStateStoreProviderBase](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.aspx) class is discussed in more detail in the "Required Classes" section later in this topic.

The [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) class determines the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value itself, rather than relying on the session-state store provider to do so. If needed, you can implement a custom [SessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionidmanager.aspx) by creating a class that inherits the [ISessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.isessionidmanager.aspx) interface. For more information, see the "Remarks" section in [ISessionIDManager](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.isessionidmanager.aspx).

[SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) will revert to the ASP.NET process identity to access any secured resource, such as a database server. You can specify that the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) instance impersonate the identity supplied by IIS by setting the **useHostingIdentity** attribute of the [<sessionState>](http://msdn.microsoft.com/en-us/library/h6bb9cz9.aspx) configuration element to **false**. For example, if you have configured your IIS application to use Windows Integrated security and you want ASP.NET to impersonate the identity provided by IIS for session management, specify <identity impersonate="true" /> in the [<system.web>](http://msdn.microsoft.com/en-us/library/dayb112d.aspx) configuration section of the Web.config file for the application, and set the **useHostingIdentity** attribute of the [<sessionState>](http://msdn.microsoft.com/en-us/library/h6bb9cz9.aspx) configuration element to **false**. If the **useHostingIdentity** attribute is **true**, ASP.NET will impersonate the process identity, or the user credentials supplied to the [<identity>](http://msdn.microsoft.com/en-us/library/72wdk8cc.aspx) configuration element (if they exist) when connecting to the data source. For more information on the ASP.NET process identity, see [Configuring ASP.NET Process Identity](http://msdn.microsoft.com/en-us/library/dwc1xthy.aspx) and [ASP.NET Impersonation](http://msdn.microsoft.com/en-us/library/xh507fc5.aspx).

 Locking Session-Store Data

ASP.NET applications are multithreaded so they can respond to multiple concurrent requests. Multiple concurrent requests might attempt to access the same session information. Consider a scenario where multiple frames in a frameset all reference ASP.NET Web pages in the same application. The separate requests for each frame in the frameset might be executed on the Web server concurrently on different threads. If the ASP.NET pages for each frame access session-state variables, you could have multiple threads accessing the session store concurrently. To avoid data collisions at the session store and unexpected session-state behavior, the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) and [SessionStateStoreProviderBase](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.aspx) classes include functionality that exclusively locks the session-store item for a particular session during the execution of an ASP.NET page. Note that no lock is set on a session-store item if the [EnableSessionState](http://msdn.microsoft.com/en-us/library/system.web.configuration.pagessection.enablesessionstate.aspx) attribute is marked as **ReadOnly**. However, other ASP.NET pages in the same application might be able to write to the session store, so a request for read-only session data from the store might still have to wait for locked data to be freed.

A lock is set on session-store data at the beginning of the request in the call to the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method. When the request completes, the lock is released during the call to the [SetAndReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setandreleaseitemexclusive.aspx) method.

If the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) instance encounters locked session data during the call to either the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) or [GetItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitem.aspx) method, it will re-request the session data at half-second intervals until either the lock is released or the amount of time specified in the [ExecutionTimeout](http://msdn.microsoft.com/en-us/library/system.web.configuration.httpruntimesection.executiontimeout.aspx) property has elapsed. If the request times out, [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) calls the [ReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.releaseitemexclusive.aspx) method to free the session-store data and request the session-store data at that time.

Locked session-store data might have been freed by a call to the [ReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.releaseitemexclusive.aspx) method on a separate thread, before the call to the [SetAndReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setandreleaseitemexclusive.aspx) method for the current response. This could cause the [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) instance to set and release session-state store data that has already been released and modified by another session. To avoid this situation, [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) includes a lock identifier with each request to modify locked session-store data. Session-store data is only modified if the lock identifier in the data store matches the lock identifier supplied by [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx).

 Deleting Expired Session-Store Data

When the [Abandon](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.abandon.aspx) method is called for a session, the data for that session is deleted from the data store using the [RemoveItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.removeitem.aspx) method. Otherwise, the data remains in the session data store to serve future requests for the session.

The mechanism for deleting expired session data depends on the capabilities of your data source. If your data source can be configured to delete expired session data according to the session [Timeout](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.timeout.aspx) property, you can use the [SetItemExpireCallback](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setitemexpirecallback.aspx) method to reference the delegate for the **Session\_OnEnd** event and raise it when deleting expired session data.

 ApplicationName

To maintain session scope, session-state providers store session information uniquely for each application. This allows multiple ASP.NET applications to use the same data source without running into a conflict if duplicate session identifiers are encountered.

Because session-state store providers store session information uniquely for each application, you must ensure that your data schema, queries, and updates include the application name. For example, the following command might be used to retrieve session data from a database.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl157_ctl00_ctl00_code');" \o "Copy Code)

SELECT \* FROM Sessions

WHERE SessionID = 'ABC123' AND ApplicationName = 'MyApplication'

Alternatively, you can store a combination of the session identifier and the application name as the unique identifier for an item in the session-state data store.

 Required Classes

To implement a session-state store provider, create a class that inherits the [SessionStateStoreProviderBase](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.aspx) abstract class. The [SessionStateStoreProviderBase](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.aspx) class in turn inherits the [ProviderBase](http://msdn.microsoft.com/en-us/library/system.configuration.provider.providerbase.aspx) abstract class, so you must implement the required members of the [ProviderBase](http://msdn.microsoft.com/en-us/library/system.configuration.provider.providerbase.aspx) class as well. The following tables list the properties and methods that you must implement from the [ProviderBase](http://msdn.microsoft.com/en-us/library/system.configuration.provider.providerbase.aspx) and [SessionStateStoreProviderBase](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.aspx) abstract classes and provides a description of each. To view an implementation of each member, see [Sample Session-State Store Provider](http://msdn.microsoft.com/en-us/library/ms178588.aspx).

**Required ProviderBase Members**

|  |  |
| --- | --- |
| **Member** | **Description** |
| [Initialize](http://msdn.microsoft.com/en-us/library/system.configuration.provider.providerbase.initialize.aspx) method | Takes as input the name of the provider and a [NameValueCollection](http://msdn.microsoft.com/en-us/library/system.collections.specialized.namevaluecollection.aspx) instance of configuration settings. This method is used to set property values for the provider instance, including implementation-specific values and options specified in the configuration file (Machine.config or Web.config). |

**Required SessionStateStoreProvider Members**

|  |  |
| --- | --- |
| **Member** | **Description** |
| [InitializeRequest](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.initializerequest.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request and performs any initialization required by your session-state store provider. |
| [EndRequest](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.endrequest.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request and performs any cleanup required by your session-state store provider. |
| [Dispose](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.dispose.aspx) method | Frees any resources no longer in use by the session-state store provider. |
| [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request and the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value for the current request. Retrieves session values and information from the session data store and locks the session-item data at the data store for the duration of the request. The [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method sets several output-parameter values that inform the calling [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) about the state of the current session-state item in the data store.  If no session item data is found at the data store, the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method sets the *locked* output parameter to **false** and returns **null**. This causes [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) to call the [CreateNewStoreData](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.createnewstoredata.aspx) method to create a new [SessionStateStoreData](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoredata.aspx) object for the request.  If session-item data is found at the data store but the data is locked, the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method sets the *locked* output parameter to **true**, sets the *lockAge* output parameter to the current date and time minus the date and time when the item was locked, sets the *lockId* output parameter to the lock identifier retrieved from the data store, and returns **null**. This causes [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) to call the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method again after a half-second interval, to attempt to retrieve the session-item information and obtain a lock on the data. If the value that the *lockAge* output parameter is set to exceeds the [ExecutionTimeout](http://msdn.microsoft.com/en-us/library/system.web.configuration.httpruntimesection.executiontimeout.aspx) value, [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) calls the [ReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.releaseitemexclusive.aspx) method to clear the lock on the session-item data and then call the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method again.  The *actionFlags* parameter is used with sessions whose [Cookieless](http://msdn.microsoft.com/en-us/library/system.web.configuration.sessionstatesection.cookieless.aspx) property is **true**, when the **regenerateExpiredSessionId** attribute is set to **true**. An *actionFlags* value set to [InitializeItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateactions.initializeitem.aspx) (1) indicates that the entry in the session data store is a new session that requires initialization. Uninitialized entries in the session data store are created by a call to the [CreateUninitializedItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.createuninitializeditem.aspx) method. If the item from the session data store is already initialized, the *actionFlags* parameter is set to zero.  If your provider supports cookieless sessions, set the *actionFlags* output parameter to the value returned from the session data store for the current item. If the *actionFlags* parameter value for the requested session-store item equals the [InitializeItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateactions.initializeitem.aspx) enumeration value (1), the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method should set the value in the data store to zero after setting the *actionFlags* out parameter. |
| [GetItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitem.aspx) method | This method performs the same work as the [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method, except that it does not attempt to lock the session item in the data store. The [GetItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitem.aspx) method is called when the [EnableSessionState](http://msdn.microsoft.com/en-us/library/system.web.configuration.pagessection.enablesessionstate.aspx) attribute is set to **ReadOnly**. |
| [SetAndReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setandreleaseitemexclusive.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request, the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value for the current request, a [SessionStateStoreData](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoredata.aspx) object that contains the current session values to be stored, the lock identifier for the current request, and a value that indicates whether the data to be stored is for a new session or an existing session.  If the *newItem* parameter is **true**, the [SetAndReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setandreleaseitemexclusive.aspx) method inserts a new item into the data store with the supplied values. Otherwise, the existing item in the data store is updated with the supplied values, and any lock on the data is released. Note that only session data for the current application that matches the supplied [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value and lock identifier values is updated.  After the [SetAndReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setandreleaseitemexclusive.aspx) method is called, the [ResetItemTimeout](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.resetitemtimeout.aspx) method is called by [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) to update the expiration date and time of the session-item data. |
| [ReleaseItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.releaseitemexclusive.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request, the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value for the current request, and the lock identifier for the current request, and releases the lock on an item in the session data store. This method is called when the [GetItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitem.aspx) or [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) method is called and the data store specifies that the requested item is locked, but the lock age has exceeded the [ExecutionTimeout](http://msdn.microsoft.com/en-us/library/system.web.configuration.httpruntimesection.executiontimeout.aspx) value. The lock is cleared by this method, freeing the item for use by other requests. |
| [RemoveItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.removeitem.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request, the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value for the current request, and the lock identifier for the current request, and deletes the session information from the data store where the data store item matches the supplied [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value, the current application, and the supplied lock identifier. This method is called when the [Abandon](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.abandon.aspx) method is called. |
| [CreateUninitializedItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.createuninitializeditem.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request, the [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value for the current request, and the lock identifier for the current request, and adds an uninitialized item to the session data store with an **actionFlags** value of [InitializeItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateactions.initializeitem.aspx).  The [CreateUninitializedItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.createuninitializeditem.aspx) method is used with cookieless sessions when the **regenerateExpiredSessionId** attribute is set to **true**, which causes [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) to generate a new [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value when an expired session ID is encountered.  The process of generating a new [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value requires the browser to be redirected to a URL that contains the newly generated session ID. The [CreateUninitializedItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.createuninitializeditem.aspx) method is called during an initial request that contains an expired session ID. After [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) acquires a new [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value to replace the expired session ID, it calls the [CreateUninitializedItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.createuninitializeditem.aspx) method to add an uninitialized entry to the session-state data store. The browser is then redirected to the URL containing the newly generated [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value. The existence of the uninitialized entry in the session data store ensures that the redirected request with the newly generated [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value is not mistaken for a request for an expired session, and instead is treated as a new session.  The uninitialized entry in the session data store is associated with the newly generated [SessionID](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.sessionid.aspx) value and contains only default values, including an expiration date and time, and a value that corresponds to the **actionFlags** parameter of the [GetItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitem.aspx) and [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) methods. The uninitialized entry in the session state store should include an **actionFlags** value equal to the [InitializeItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateactions.initializeitem.aspx) enumeration value (1). This value is passed to [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) by the [GetItem](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitem.aspx) and [GetItemExclusive](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.getitemexclusive.aspx) methods and specifies for [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) that the current session is a new session. [SessionStateModule](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatemodule.aspx) will then initialize the new session and raise the **Session\_OnStart** event. |
| [CreateNewStoreData](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.createnewstoredata.aspx) method | Takes as input the [HttpContext](http://msdn.microsoft.com/en-us/library/system.web.httpcontext.aspx) instance for the current request and the [Timeout](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.timeout.aspx) value for the current session, and returns a new [SessionStateStoreData](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoredata.aspx) object with an empty [ISessionStateItemCollection](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.isessionstateitemcollection.aspx) object, an [HttpStaticObjectsCollection](http://msdn.microsoft.com/en-us/library/system.web.httpstaticobjectscollection.aspx) collection, and the specified [Timeout](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.httpsessionstate.timeout.aspx) value. The [HttpStaticObjectsCollection](http://msdn.microsoft.com/en-us/library/system.web.httpstaticobjectscollection.aspx) instance for the ASP.NET application can be retrieved using the [GetSessionStaticObjects](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateutility.getsessionstaticobjects.aspx) method. |
| [SetItemExpireCallback](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstatestoreproviderbase.setitemexpirecallback.aspx) method | Takes as input a delegate that references the **Session\_OnEnd** event defined in the Global.asax file. If the session-state store provider supports the **Session\_OnEnd** event, a local reference to the [SessionStateItemExpireCallback](http://msdn.microsoft.com/en-us/library/system.web.sessionstate.sessionstateitemexpirecallback.aspx) parameter is set and the method returns **true**; otherwise, the method returns **false**. |

 Sample Provider

To view an example implementation of a custom session-state store provider that manages session information in an Access database, see [Sample Session-State Store Provider](http://msdn.microsoft.com/en-us/library/ms178588.aspx).