

ADO.NET

Abril 2015

ADO.NET

- ADO.NET es un conjunto de clases que exponen servicios de acceso a datos para programadores de .NET Framework. ADO.NET ofrece abundancia de componentes para la creación de aplicaciones de uso compartido de datos distribuidas.

ADO.NET

- Constituye una parte integral de .NET Framework y proporciona acceso a datos relacionales, XML y de aplicaciones. ADO.NET satisface diversas necesidades de desarrollo, como la creación de clientes de base de datos front-end y objetos empresariales de nivel medio que utilizan aplicaciones, herramientas, lenguajes o exploradores de Internet.

ADO.NET

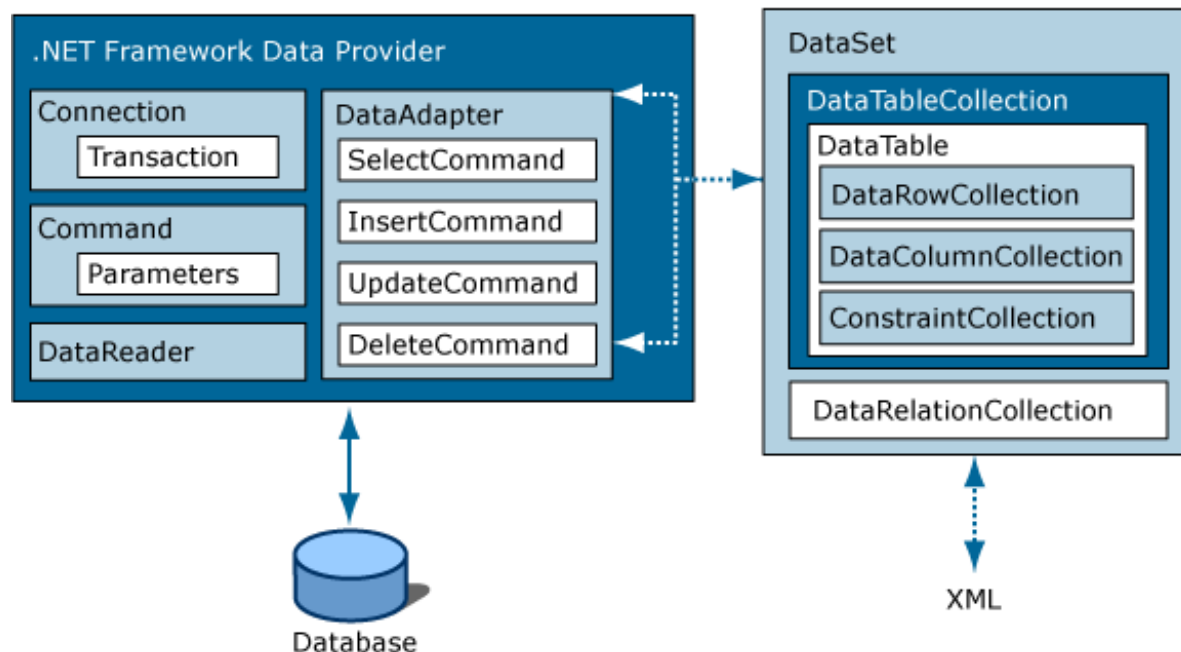
- ADO.NET proporciona acceso coherente a orígenes de datos como SQL Server y XML, así como a orígenes de datos expuestos mediante OLE DB y ODBC.
- Las aplicaciones de consumidor que comparten datos pueden utilizar ADO.NET para conectar a estos orígenes de datos y recuperar, controlar y actualizar los datos contenidos

ADO.NET

- ADO.NET separa el acceso a datos de la manipulación de datos y crea componentes discretos que se pueden utilizar por separado o conjuntamente.
- ADO.NET incluye proveedores de datos .NET Framework para conectarse a una base de datos, ejecutar comandos y recuperar resultados.
- Los resultados se procesan directamente o se colocan en un objeto [DataSet](#) de ADO.NET.

Arquitectura

- Las clases de ADO.NET se encuentran en System.Data.dll y se integran con las clases de XML incluidas en System.Xml.dll.



Proveedores de datos .NET Framework

- Los proveedores de datos .NET Framework son componentes diseñados explícitamente para la manipulación de datos y el acceso rápido a datos. El objeto **Connection** proporciona conectividad a un origen de datos.
- El objeto **Command** permite tener acceso a comandos de base de datos para devolver datos, modificar datos, ejecutar procedimientos almacenados y enviar o recuperar información sobre parámetros.
- **DataReader** proporciona un flujo de datos de alto rendimiento desde el origen de datos.
- El objeto **DataAdapter** proporciona el puente entre el objeto DataSet y el origen de datos.
- **DataAdapter** utiliza objetos Command para ejecutar comandos SQL en el origen de datos tanto para cargar **DataSet** con datos y reconciliar en el origen de datos los cambios aplicados a los datos incluidos en el **DataSet**.

ADO.NET Entity Framework

ADO.NET Entity Framework

- ADO.NET Entity Framework está diseñado para permitir que los desarrolladores creen aplicaciones de acceso a los datos programando en un modelo de aplicación conceptual en lugar de programar directamente en un esquema de almacenamiento relacional.
- El objetivo es reducir la cantidad de código y mantenimiento que se necesita para las aplicaciones orientadas a datos. Para obtener más información, vea ADO.NET Entity Framework.

Ventajas (1/2)

- Las aplicaciones de Entity Framework ofrecen las siguientes ventajas:
 - Las aplicaciones pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones.
 - Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento.
 - Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.

Ventajas (2/2)

- Los desarrolladores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes.
- Se pueden asignar varios modelos conceptuales a un único esquema de almacenamiento.
- La compatibilidad con Language Integrated Query (LINQ) proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.