Data Access in Client and Middle-Tier Programming

**Introduction to Data Adapters**

Data adapters are an integral part of ADO.NET managed providers, which are the set of objects used to communicate between a data source and a dataset. (In addition to adapters, managed providers include connection objects, data reader objects, and command objects.) Adapters are used to exchange data between a data source and a dataset. In many applications, this means reading data from a database into a dataset, and then writing changed data from the dataset back to the database. However, a data adapter can move data between any source and a dataset. For example, there could be an adapter that moves data between a Microsoft Exchange server and a dataset.

|  |
| --- |
| **Description: NoteNote:** |
| In the previous version of Visual Studio, data adapters were used for communicating between an application and a database. While data adapters are still a main component of [.NET Framework Data Providers (ADO.NET)](http://msdn.microsoft.com/en-us/library/a6cd7c08(VS.90).aspx), TableAdapters are designer-generated components that simplify the process of moving data between your application and a database. For more information on working with TableAdapters, see [TableAdapter Overview](http://msdn.microsoft.com/en-us/library/bz9tthwx(VS.90).aspx). |

Generally, adapters are configurable to allow you to specify what data to move into and out of the dataset. Often this takes the form of references to SQL statements or stored procedures that are invoked to read or write to a database.

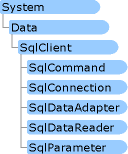
Visual Studio makes these data adapters available for use with databases:

* The [OleDbDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbdataadapter(VS.90).aspx) object is suitable for use with any data source exposed by an OLE DB provider.
* The [SqlDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqldataadapter(VS.90).aspx) object is specific to SQL Server. Because it does not have to go through an OLE DB layer, it is faster than the [OleDbDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbdataadapter(VS.90).aspx) class. However, it can only be used with SQL Server 7.0 or later.
* The [OdbcDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.odbc.odbcdataadapter(VS.90).aspx) object is optimized for accessing ODBC data sources.
* The [OracleDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.oracleclient.oracledataadapter(VS.90).aspx) object is optimized for accessing Oracle databases.

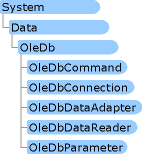
|  |
| --- |
| **Description: NoteNote:** |
| Data adapters, data connections, data commands, and data readers are the components that make up a .NET Framework data provider. Microsoft and third-party providers can make available other .NET Framework data providers that can be integrated into Visual Studio. For more information on the different .NET Framework data providers, see [.NET Framework Data Providers (ADO.NET)](http://msdn.microsoft.com/en-us/library/a6cd7c08(VS.90).aspx). |

You can create and manipulate adapters using the following portions of the .NET Framework managed-provider namespaces.

**SqlClient Managed-Provider Namespace**



**OleDb Managed-Provider Namespace**



Generally, each data adapter exchanges data between a single data-source table and a single [DataTable](http://msdn.microsoft.com/en-us/library/system.data.datatable(VS.90).aspx) object in the dataset. If the dataset contains multiple data tables, the usual strategy is to have multiple data adapters feeding data to it and writing its data back to individual data-source tables.

When you want to populate a table in a dataset, you call an adapter method that executes an SQL statement or stored procedure. The adapter creates a data reader object ([SqlDataReader](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqldatareader(VS.90).aspx), [OleDbDataReader](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbdatareader(VS.90).aspx), [OdbcDataReader](http://msdn.microsoft.com/en-us/library/system.data.odbc.odbcdatareader(VS.90).aspx), or [OracleDataReader](http://msdn.microsoft.com/en-us/library/system.data.oracleclient.oracledatareader(VS.90).aspx)) to read the data into a dataset.

|  |
| --- |
| **Description: NoteNote:** |
| You can read data from the database without having to store it in a dataset, which can be very efficient for situations involving read-only data. For more information, see "Read-Only Data" below. You can also execute SQL statements directly without using them to populate a dataset. For more information, see [Commands and Parameters (ADO.NET)](http://msdn.microsoft.com/en-us/library/ms254953(VS.90).aspx). |

Similarly, when you want to update the database, you invoke an adapter method that calls an appropriate SQL statement or stored procedure to make the actual update in the database.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifData Adapters and Related Tables

An implication of having separate tables in the dataset is that a data adapter typically does not reference SQL commands or stored procedures that join tables. Instead, information from the related tables is read separately into the dataset by different adapters. Then a [DataRelation](http://msdn.microsoft.com/en-us/library/system.data.datarelation(VS.90).aspx) object is used to manage constraints between the dataset tables (such as cascading updates) and to allow you to navigate between related master and child records.

For example, imagine that you are working with two related tables in the Northwind database, Customers and Orders. Rather than specifying a join to combine both tables into a single result set, you would most commonly define two adapters, one to populate a Customers table in the dataset and a second adapter to read Order records into a different dataset table. The individual adapters would probably include selection criteria to limit the number of records in the data tables.

In the dataset you would also define a [DataRelation](http://msdn.microsoft.com/en-us/library/system.data.datarelation(VS.90).aspx) object specifying that order records are related to customer records by the CustomerID field. You can still manage the tables individually, which would not be possible if you had joined tables before fetching records from the data source. For situations where you wanted to work with related records, you can invoke properties and methods of the [DataRelation](http://msdn.microsoft.com/en-us/library/system.data.datarelation(VS.90).aspx) object.

For more information about data relations, see [Relationships in Datasets](http://msdn.microsoft.com/en-us/library/dbwcse3d(VS.90).aspx).

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifConnection Objects

A data adapter needs an open connection to a data source to read and write data. Therefore, an adapter uses connection objects ([SqlConnection](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection(VS.90).aspx), [OleDbConnection](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbconnection(VS.90).aspx), [OdbcConnection](http://msdn.microsoft.com/en-us/library/system.data.odbc.odbcconnection(VS.90).aspx), or [OracleConnection](http://msdn.microsoft.com/en-us/library/system.data.oracleclient.oracleconnection(VS.90).aspx)) to communicate with a data source. (The adapter can contain up to four connection references, one for each type of action it can perform: Select, Update, Insert, and Delete.)

The following table lists connection objects in the **Data** tab of the **Toolbox**:

|  |  |
| --- | --- |
| **Connection Object** | **Description** |
| [SqlConnection](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection(VS.90).aspx) | A connection to a SQL Server 7.0 or later database. |
| [OleDbConnection](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbconnection(VS.90).aspx) | A connection to any OLE DB data source. |
| [OdbcConnection](http://msdn.microsoft.com/en-us/library/system.data.odbc.odbcconnection(VS.90).aspx) | A connection to an ODBC data source. |
| [OracleConnection](http://msdn.microsoft.com/en-us/library/system.data.oracleclient.oracleconnection(VS.90).aspx) | A connection to an Oracle database. |

In all cases, the connection object represents a unique session within the data source. All connection objects provide properties to establish and modify connection details (such as user ID and password, and connection-timeout setting). They also provide methods to begin, commit, and roll back database transactions. For more information about connection objects, see [Connecting to a Data Source (ADO.NET)](http://msdn.microsoft.com/en-us/library/32c5dh3b(VS.90).aspx).

|  |
| --- |
| **Description: Security noteSecurity Note:** |
| Storing sensitive information (such as the server name, user name, and password) can have implications for the security of your application. Using Windows Authentication (also known as integrated security) is a more secure way to control access to a database. |

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifADO.NET Command Objects

Using an adapter, you can read, add, update, and delete records in a data source. To allow you to specify how each of these operations should occur, an adapter supports the following four properties:

* SelectCommand – reference to a command (SQL statement or stored procedure name) that retrieves rows from the data store.
* InsertCommand – reference to a command for inserting rows into the data store.
* UpdateCommand – reference to a command for modifying rows in the data store.
* DeleteCommand – reference to a command for deleting rows from the data store.

The properties are themselves objects — they are instances of the [SqlCommand](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand(VS.90).aspx), [OleDbCommand](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbcommand(VS.90).aspx), [OdbcCommand](http://msdn.microsoft.com/en-us/library/system.data.odbc.odbccommand(VS.90).aspx), or [OracleCommand](http://msdn.microsoft.com/en-us/library/system.data.oracleclient.oraclecommand(VS.90).aspx) class. The objects support a CommandText property containing a reference to an SQL statement or stored procedure.

|  |
| --- |
| **Description: NoteNote:** |
| The command class must match the connection class. For example, if you are using a [SqlConnection](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection(VS.90).aspx) object to communicate with a SQL Server, you must also use commands that derive from the [SqlCommand](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand(VS.90).aspx) class. |

Although you can explicitly set the text of a command object, you do not always need to; in many situations, Visual Studio will generate the SQL statements needed. In addition, the adapter can automatically generate appropriate SQL statements at run time if the UpdateCommand, InsertCommand, or DeleteCommand objects are not specified. For more information, see [Generating Commands with CommandBuilders (ADO.NET)](http://msdn.microsoft.com/en-us/library/tf579hcz(VS.90).aspx).

However, you can manipulate command objects at design time and run time in order to have more direct control over how the commands are executed. For example, you can create or modify the command associated with a SelectCommand object just before it is executed.

You can also execute commands yourself, independently of the data adapter. This allows you to pass arbitrary SQL commands through the data adapter, such as those used to define or modify database definitions. You can also call stored procedures directly that do not return record sets — for example, a stored procedure that validates a user entry against a database. For more information, see [Executing a Command (ADO.NET)](http://msdn.microsoft.com/en-us/library/tyy0sz6b(VS.90).aspx).

|  |
| --- |
| **Description: Security noteSecurity Note:** |
| When using data commands with a [CommandType](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand.commandtype(VS.90).aspx) property set to Text, carefully check information that is sent from a client before passing it to your database. Malicious users might try to send (inject) modified or additional SQL statements in an effort to gain unauthorized access or to damage the database. Before you transfer user input to a database, you should always verify that the information is valid; it is a best practice to always use parameterized queries or stored procedures when possible. For more information, see [Script Exploits Overview](http://msdn.microsoft.com/en-us/library/w1sw53ds(VS.90).aspx). |

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifCommand Parameters

The commands in a data adapter are usually parameter-driven. The command for the SelectCommand property, for example, often has a parameter in its WHERE clause so you can specify at run time what records to get from the database. The other commands use parameters that allow you to pass at run time the data to write into a record and what record in the database to update. For more information about how parameters are used in data adapters, see [Parameters in Data-Adapter Commands](http://msdn.microsoft.com/en-us/library/ebxy9a8b(VS.90).aspx).

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifReading and Updating with Data Adapters

The primary purpose of the data adapter is to communicate data between a data store and a dataset. The adapter supports specific methods to move the data back and forth between the two.

|  |
| --- |
| **Description: NoteNote:** |
| If you just want to read data (not update it), you do not have to store it in a dataset. Instead, you can read directly out of the database and into an application. For more information, see "Read-Only Data" below. |

You can use a data adapter to perform the following operations:

* Retrieve rows from a data store into corresponding data tables within the dataset.

To retrieve rows into a dataset, use the Fill method on a data adapter object ([SqlDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqldataadapter(VS.90).aspx), [OleDbDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbdataadapter(VS.90).aspx), [OdbcDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.odbc.odbcdataadapter(VS.90).aspx), or [OracleDataAdapter](http://msdn.microsoft.com/en-us/library/system.data.oracleclient.oracledataadapter(VS.90).aspx)). When you invoke the Fill method, it transmits an SQL SELECT statement to the data store.

* Transmit changes made to a dataset table to the corresponding data store.

To transmit a dataset table of the dataset to the data store, use the adapter's Update method. When you invoke the method, it executes whatever SQL INSERT, UPDATE or DELETE statements are needed, depending on whether the affected record is new, changed, or deleted.

For more information about how updates are performed using data adapters, see [Updating Data Sources with DataAdapters (ADO.NET)](http://msdn.microsoft.com/en-us/library/33y2221y(VS.90).aspx).

**Read-Only Data**

If your program needs to perform a sequential, read-only pass through a query result, you can use a data reader object in place of filling a dataset. A data reader object fetches the data from the data source and passes it through directly to your application. Typically, data reader objects are used for read-only, forward-only access to the data at times when you do not need to cache data in a dataset. (The data adapter itself uses a data reader object to populate a dataset.) An example is a Web Forms page that displays database information; because the Web Forms page is recreated with each roundtrip, it is often not useful to store the data in a dataset.

Visual Studio supplies four data reader objects, a [SqlDataReader](http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqldatareader(VS.90).aspx), [OleDbDataReader](http://msdn.microsoft.com/en-us/library/system.data.oledb.oledbdatareader(VS.90).aspx), [OdbcDataReader](http://msdn.microsoft.com/en-us/library/system.data.odbc.odbcdatareader(VS.90).aspx), and [OracleDataReader](http://msdn.microsoft.com/en-us/library/system.data.oracleclient.oracledatareader(VS.90).aspx). For more information about using the data reader object for efficient read-only access, see [DataAdapters and DataReaders (ADO.NET)](http://msdn.microsoft.com/en-us/library/ms254931(VS.90).aspx).

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifTable Mappings

By default, when you use Visual Studio tools to generate a dataset from database tables, the names of the tables and columns are the same in the dataset as in the database. However, you might find this impractical. For example, you might find the names used in the database too terse or verbose; or the names might be in a foreign language. If you are working with an existing schema, you might also find that the names defined in the schema do not match those used in your database.

Therefore, the names in the database and dataset do not have to match. Instead, you can create new table and column names in the dataset command, and then map those to the names used in the database. Adapters use the [TableMappings](http://msdn.microsoft.com/en-us/library/system.data.common.dataadapter.tablemappings(VS.90).aspx) collection to maintain the correspondence between structures of the dataset (data tables and data columns) and structures of the data store (tables and columns). For more information about table mappings, see [Table Mapping in Data Adapters](http://msdn.microsoft.com/en-us/library/byt00a0h(VS.90).aspx).