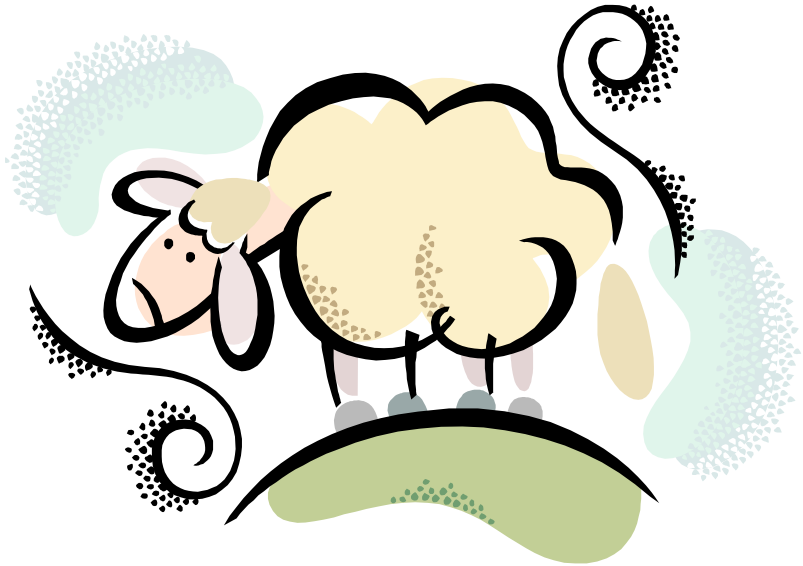

# C# Programming IV-4F:

## *System.Data.SqlTypes Namespace*

# Legal Stuff



This presentation as well as other presentations and documents found on the sheepsqueezers.com website may contain quoted material from outside sources such as books, articles and websites. It is our intention to diligently reference all outside sources. Occasionally, though, a reference may be missed. No copyright infringement whatsoever is intended, and all outside source materials are copyright of their respective author(s).


# .NET Lecture Series

| | | | | |
|---|---|---|---|---|
| | C# Programming I: Concepts of OOP | C# Programming II: Beginning C# | C# Programming III: Advanced C# | |
| C# Programming IV-1: System Namespace | C# Programming IV-2: System.Collections Namespace | C# Programming IV-3: System.Collections. Generic Namespace | C# Programming IV-4A: System.Data Namespace | C# Programming IV-4B: System.Data.Odbc Namespace |
| C# Programming IV-4C: System.Data.OleDb Namespace | C# Programming IV-4D: Oracle.DataAccess.Client Namespace | C# Programming IV-4E: System.Data.SqlClient Namespace | **C# Programming IV-4F: System.Data.SqlTypes Namespace** | C# Programming IV-5: System.Drawing/(2D) Namespace |
| C# Programming IV-6: System.IO Namespace | C# Programming IV-7: System.Numerics | C# Programming IV-8: System.Text and System.Text. RegularExpressions Namespaces | C# Programming V: Introduction to LINQ | |

C# Self-Inflicted Project #1 Address Cleaning

C# Self-Inflicted Project #2 Large Intersection Problem

# Charting Our Course

- The `System.Data.SqlTypes` Namespace
- What Next?

# The `System.Data.SqlTypes` Namespace


The `System.Data.SqlTypes` namespace is defined by Microsoft as follows:

*The System.Data.SqlTypes namespace provides classes for native data types in SQL Server. These classes provide a safer, faster alternative to the data types provided by the .NET Framework common language runtime (CLR).* ***Using the classes in this namespace helps prevent type conversion errors caused by loss of precision.*** *Because other data types are converted to and from SqlTypes behind the scenes, explicitly creating and using objects within this namespace also yields faster code. Each data type in SqlTypes has its equivalent data type in SQL Server, with the same underlying data representation. Many of them also have equivalent data types in the CLR. However, SqlDateTime, SqlDecimal, and SqlString have different underlying data structures with their corresponding .NET Framework data types.*

# The `System.Data.SqlTypes` Namespace


The following table maps the members of the **SqlTypes** namespace to Microsoft SQL Server data types and to the members of the SqlDbType enumeration.

| .NET Framework SqlTypes | Native SQL Server | .NET Framework SqlDbType |
|---|---|---|
| SqlBinary | binary, image, timestamp, varbinary | Binary, Image, TimeStamp, VarBinary |
| SqlBoolean | bit | Bit |
| SqlByte | tinyint | TinyInt |
| SqlDateTime | datetime, smalldatetime | DateTime, SmallDateTime |
| SqlDecimal | numeric, decimal | Decimal |
| SqlDouble | float | Float |
| SqlFileStream | varbinary | VarBinary |
| SqlGuid | uniqueidentifier | UniqueIdentifier |
| SqlInt16 | smallint | SmallInt |
| SqlInt32 | int | Int |
| SqlInt64 | bigint | BigInt |
| SqlMoney | money, smallmoney | Money, SmallMoney |
| SqlSingle | real | Real |
| SqlString | char, nchar, text, ntext, nvarchar, varchar | Char, NChar, Text, Ntext, NVarChar, VarChar |
| SqlXml | xml | Xml |

# The System.Data.SqlTypes Namespace


```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;

class MainProgram {

 public static void Main() {

  String sConn = @"Data Source=SCOTT-PC\SQLTEST;Initial Catalog=TESTDB;Integrated Security=SSPI;";
  SqlConnection oConn = new SqlConnection(sConn);
  SqlCommand oCmd = new SqlCommand("SELECT DISTINCT ZIPCODE,POPULATION FROM POPBYZIP",oConn);
  oConn.Open();

  //Create a table to hold the POPBYZIP table
  DataSet dsTESTDB = new DataSet("TESTDB");
  DataTableCollection dtcTESTDB = dsTESTDB.Tables;
  dtcTESTDB.Add("POPBYZIP");

  //Create two columns that appear in the POPBYZIP table
  DataColumn dcZIP = new DataColumn("ZIPCODE",typeof(SqlString)); dcZIP.AllowDBNull = true;
dcZIP.MaxLength=5;
  DataColumn dcPOP = new DataColumn("POPULATION",typeof(SqlInt32)); dcPOP.AllowDBNull = true;

  //Next, add columns to our POPBYZIP table.
  dtcTESTDB["POPBYZIP"].Columns.Add(dcZIP);
  dtcTESTDB["POPBYZIP"].Columns.Add(dcPOP);

  //Set up a data reader to the database table POPBYZIP
  SqlDataReader oDR = oCmd.ExecuteReader();
  dsTESTDB.Tables["POPBYZIP"].Load(oDR);

  ..continued on the next slide…
```

# The System.Data.SqlTypes Namespace


```
    //Close the reader
    oDR.Close();

    //Close the connections (since we did not use USING)
    oConn.Close();

    //Printout the data
    DataTableReader dtrPOPBYZIP = new DataTableReader(dsTESTDB.Tables["POPBYZIP"]);
    while(dtrPOPBYZIP.Read()) {
     Console.WriteLine("There are {1} people living in the zipcode {0}.",dtrPOPBYZIP[0],dtrPOPBYZIP[1]);
    }

  }

}
```


# The `System.Data.SqlTypes` Namespace

- → Classes
  - → `SqlBytes`

# SqlBytes


The `SqlBytes` class represents a mutable reference type that wraps either a Buffer or a Stream.

## Constructors

- SqlBytes() - Initializes a new instance of the SqlBytes class.
- SqlBytes(Byte[]) - Initializes a new instance of the SqlBytes class based on the specified byte array.
- SqlBytes(SqlBinary) - Initializes a new instance of the SqlBytes class based on the specified SqlBinary value.
- SqlBytes(Stream) - Initializes a new instance of the SqlBytes class based on the specified Stream value.

## Properties

- Buffer - Returns a reference to the internal buffer.
- IsNull - Gets a Boolean value that indicates whether this SqlBytes is null.
- Item - Gets or sets the SqlBytes instance at the specified index.
- Length - Gets the length of the value that is contained in the SqlBytes instance.
- MaxLength - Gets the maximum length of the value of the internal buffer of this SqlBytes.
- Null - Returns a null instance of this SqlBytes.
- Storage - Returns information about the storage state of this SqlBytes instance.
- Stream - Gets or sets the data of this SqlBytes as a stream.
- Value - Returns a managed copy of the value held by this SqlBytes.

## Methods

- Equals(Object) - Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
- Finalize - Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
- GetHashCode - Serves as a hash function for a particular type. (Inherited from Object.)
- GetType - Gets the Type of the current instance. (Inherited from Object.)
- GetXsdType - Returns the XML Schema definition language (XSD) of the specified XmlSchemaSet.
- MemberwiseClone - Creates a shallow copy of the current Object. (Inherited from Object.)


## Methods (continued)

- Read - Copies bytes from this SqlBytes instance to the passed-in buffer and returns the number of copied bytes.
- SetLength - Sets the length of this SqlBytes instance.
- SetNull - Sets this SqlBytes instance to null.
- ToSqlBinary - Constructs and returns a SqlBinary from this SqlBytes instance.
- ToString - Returns a string that represents the current object. (Inherited from Object.)
- Write - Copies bytes from the passed-in buffer to this SqlBytes instance.

## Operators

- Explicit(SqlBinary to SqlBytes) - Converts a SqlBinary structure to a SqlBytes structure.
- Explicit(SqlBytes to SqlBinary) - Converts a SqlBytes structure to a SqlBinary structure.


# The `System.Data.SqlTypes` Namespace

- → Classes
  - → `SqlChars`

# SqlChars


The `SqlChars` class is a mutable reference type that wraps a Char array or a SqlString instance.

## Constructors

- SqlChars()  Initializes a new instance of the SqlChars class.
- SqlChars(Char[])  Initializes a new instance of the SqlChars class based on the specified character array.
- SqlChars(SqlString)  Initializes a new instance of the SqlChars class based on the specified SqlString value.

## Properties

- Buffer  Returns a reference to the internal buffer.
- IsNull  Gets a Boolean value that indicates whether this SqlChars is null.
- Item  Gets or sets the SqlChars instance at the specified index.
- Length  Gets the length of the value that is contained in the SqlChars instance.
- MaxLength  Gets the maximum length in two-byte characters of the value the internal buffer can hold.
- Null  Returns a null instance of this SqlChars.
- Storage  Returns information about the storage state of this SqlChars instance.
- Value  Returns a managed copy of the value held by this SqlChars.

## Methods

- Equals(Object)  Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
- Finalize  Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
- GetHashCode  Serves as a hash function for a particular type. (Inherited from Object.)
- GetType  Gets the Type of the current instance. (Inherited from Object.)
- GetXsdType  Returns the XML Schema definition language (XSD) of the specified XmlSchemaSet.
- MemberwiseClone  Creates a shallow copy of the current Object. (Inherited from Object.)
- Read  Copies characters from this SqlChars instance to the passed-in buffer and returns the number of copied characters.
- SetLength  Sets the length of this SqlChars instance.
- SetNull  Sets this SqlChars instance to null.
- ToSqlString  Converts this SqlChars instance to its equivalent SqlString representation.
- ToString  Returns a string that represents the current object. (Inherited from Object.)
- Write  Copies characters from the passed-in buffer to this SqlChars instance.

# SqlChars

sheepsqueezers.com

## Operators

- Explicit(SqlChars to SqlString)  Converts a SqlChars structure to a SqlString structure.
- Explicit(SqlString to SqlChars)  Converts a SqlString structure to a SqlChars structure.


# The `System.Data.SqlTypes` Namespace

- → Classes
  - → `SqlFileStream`

# SqlFileStream


The `SqlFileStream` class exposes SQL Server data that is stored with the FILESTREAM column attribute as a sequence of bytes. According to Microsoft's website: *The SqlFileStream class is used to work with varbinary(max) data stored with the FILESTREAM attribute in a SQL Server 2008 database. You must install the .NET Framework 3.5 SP1 (or later) to use System.Data.SqlTypes.SqlFileStream to work with FILESTREAM data. Specifying the FILESTREAM attribute on a varbinary(max) column causes SQL Server to store the data in the local NTFS file system instead of in the database file. Transact-SQL statements provide data manipulation capabilities within the server, and Win32 file system interfaces provide streaming access to the data. Individual files stored in a FILESTREAM column cannot be opened directly from the NTFS file system. Streaming FILESTREAM data works only in the context of a SQL Server transaction. The SqlFileStream class is derived from the Stream class, which represents an abstraction of a sequence of bytes from some arbitrary data source such as a file or a block of memory. You can read from a FILESTREAM by transferring data from a stream into a data structure such as an array of bytes. You can write to a FILESTREAM by transferring the data from a data structure into a stream. You can also seek within the stream, which allows you to query and modify data at the current position within the stream.*

## Constructors

- SqlFileStream(String, Byte[], FileAccess) - Initializes a new instance of the SqlFileStream class.
- SqlFileStream(String, Byte[], FileAccess, FileOptions, Int64) - Initializes a new instance of the SqlFileStream class.

# SqlFileStream


## Properties

- CanRead - Gets a value indicating whether the current stream supports reading. (Overrides Stream.CanRead.)
- CanSeek - Gets a value indicating whether the current stream supports seeking. (Overrides Stream.CanSeek.)
- CanTimeout - Gets a value indicating whether the current stream can time out. (Overrides Stream.CanTimeout.)
- CanWrite - Gets a value indicating whether the current stream supports writing. (Overrides Stream.CanWrite.)
- Length - Gets a value indicating the length of the current stream in bytes. (Overrides Stream.Length.)
- Name - Gets the logical path of the SqlFileStream passed to the constructor.
- Position - Gets or sets the position within the current stream. (Overrides Stream.Position.)
- ReadTimeout - Gets or sets a value, in milliseconds, that determines how long the stream will attempt to read before timing out. (Overrides Stream.ReadTimeout.)
- TransactionContext - Gets or sets the transaction context for this SqlFileStream object.
- WriteTimeout - Gets or sets a value, in milliseconds, that determines how long the stream will attempt to write before timing out. (Overrides Stream.WriteTimeout.)

## Methods

- BeginRead - Begins an asynchronous read operation. (Overrides Stream.BeginRead(Byte[], Int32, Int32, AsyncCallback, Object).)
- BeginWrite - Begins an asynchronous write operation. (Overrides Stream.BeginWrite(Byte[], Int32, Int32, AsyncCallback, Object).)
- Close - Closes the current stream and releases any resources (such as sockets and file handles) associated with the current stream. (Inherited from Stream.)
- CopyTo(Stream) - Reads the bytes from the current stream and writes them to the destination stream. (Inherited from Stream.)
- CopyTo(Stream, Int32) - Reads all the bytes from the current stream and writes them to a destination stream, using a specified buffer size. (Inherited from Stream.)
- CreateObjRef - Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object. (Inherited from MarshalByRefObject.)
- CreateWaitHandle - Obsolete. Allocates a WaitHandle object. (Inherited from Stream.)
- Dispose() - Releases all resources used by the Stream. (Inherited from Stream.)
- Dispose(Boolean) - Releases the unmanaged resources used by the Stream and optionally releases the managed resources. (Inherited from Stream.)
- EndRead - Waits for the pending asynchronous read to complete. (Overrides Stream.EndRead(IAsyncResult).)
- EndWrite - Ends an asynchronous write operation. (Overrides Stream.EndWrite(IAsyncResult).)

# SqlFileStream


## Methods

- Equals(Object) - Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
- Finalize - Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
- Flush - clears all buffers for this stream and causes any buffered data to be written to the underlying device. (Overrides Stream.Flush().)
- GetHashCode - Serves as a hash function for a particular type. (Inherited from Object.)
- GetLifetimeService - Retrieves the current lifetime service object that controls the lifetime policy for this instance. (Inherited from MarshalByRefObject.)
- GetType - Gets the Type of the current instance. (Inherited from Object.)
- InitializeLifetimeService - Obtains a lifetime service object to control the lifetime policy for this instance. (Inherited from MarshalByRefObject.)
- MemberwiseClone() - Creates a shallow copy of the current Object. (Inherited from Object.)
- MemberwiseClone(Boolean) - Creates a shallow copy of the current MarshalByRefObject object. (Inherited from MarshalByRefObject.)
- ObjectInvariant - Infrastructure. Provides support for a Contract. (Inherited from Stream.)
- Read - Reads a sequence of bytes from the current stream and advances the position within the stream by the number of bytes read. (Overrides Stream.Read(Byte[], Int32, Int32).)
- ReadByte - Reads a byte from the stream and advances the position within the stream by one byte, or returns -1 if at the end of the stream. (Overrides Stream.ReadByte().)
- Seek - Sets the position within the current stream. (Overrides Stream.Seek(Int64, SeekOrigin).)
- SetLength - Sets the length of the current stream. (Overrides Stream.SetLength(Int64).)
- ToString - Returns a string that represents the current object. (Inherited from Object.)
- Write - Writes a sequence of bytes to the current stream and advances the current position within this stream by the number of bytes written. (Overrides Stream.Write(Byte[], Int32, Int32).)
- WriteByte - Writes a byte to the current position in the stream and advances the position within the stream by one byte. (Overrides Stream.WriteByte(Byte).)


# The `System.Data.SqlTypes` Namespace

- → Classes
  - → `SqlXml`

# SqlXml


The `SqlXml` class represents XML data stored in or retrieved from a server. Accoding to Microsoft's website: *This class contains an instance of an XmlReader-derived type, and adds SQL-specific features such as database-style null semantics by implementing the INullable interface. When you use SqlXml, the XML value that you assign to the SqlXml instance must be consumable by an XmlReader. For unicode data, the Byte Order Mark (BOM) must be present in the stream of data.*

## Constructors

- SqlXml() - Creates a new SqlXml instance.
- SqlXml(Stream) - Creates a new SqlXml instance, supplying the XML value from the supplied Stream-derived instance.
- SqlXml(XmlReader) - Creates a new SqlXml instance and associates it with the content of the supplied XmlReader.

## Properties

- IsNull - Indicates whether this instance represents a null SqlXml value.
- Null - Represents a null instance of the SqlXml type.
- Value - Gets the string representation of the XML content of this SqlXml instance.

## Methods

- CreateReader - Gets the value of the XML content of this SqlXml as a XmlReader.
- Equals(Object) - Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
- Finalize - Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
- GetHashCode - Serves as a hash function for a particular type. (Inherited from Object.)
- GetType - Gets the Type of the current instance. (Inherited from Object.)
- GetXsdType - Returns the XML Schema definition language (XSD) of the specified XmlSchemaSet.
- MemberwiseClone - Creates a shallow copy of the current Object. (Inherited from Object.)
- ToString - Returns a string that represents the current object. (Inherited from Object.)


# The `System.Data.SqlTypes` Namespace

→ Attributes

# Attributes

The are no attributes in the `System.Data.SqlTypes` namespace.


# The `System.Data.SqlTypes` Namespace

→ EventArgs

# EventArgs

The are no EventArgs in the `System.Data.SqlTypes` namespace.


# The `System.Data.SqlTypes` Namespace

→ Structures

# Structures

The are several structures available in the `System.Data.SqlTypes` namespace.


## Structures

- SqlBinary - Represents a variable-length stream of binary data to be stored in or retrieved from a database.
- SqlBoolean - Represents an integer value that is either 1 or 0 to be stored in or retrieved from a database.
- SqlByte - Represents an 8-bit unsigned integer, in the range of 0 through 255, to be stored in or retrieved from a database.
- SqlDateTime - Represents the date and time data ranging in value from January 1, 1753 to December 31, 9999 to an accuracy of 3.33 milliseconds to be stored in or retrieved from a database. The SqlDateTime structure has a different underlying data structure from its corresponding .NET Framework type, DateTime, which can represent any time between 12:00:00 AM 1/1/0001 and 11:59:59 PM 12/31/9999, to the accuracy of 100 nanoseconds. SqlDateTime actually stores the relative difference to 00:00:00 AM 1/1/1900. Therefore, a conversion from "00:00:00 AM 1/1/1900" to an integer will return 0.
- SqlDecimal - Represents a numeric value between - 10^38 +1 and 10^38 - 1, with fixed precision and scale.
- SqlDouble - Represents a floating-point number within the range of -1.79E +308 through 1.79E +308 to be stored in or retrieved from a database.
- SqlGuid - Represents a GUID to be stored in or retrieved from a database.
- SqlInt16 - Represents a 16-bit signed integer to be stored in or retrieved from a database.
- SqlInt32 - Represents a 32-bit signed integer to be stored in or retrieved from a database.
- SqlInt64 - Represents a 64-bit signed integer to be stored in or retrieved from a database.
- SqlMoney - Represents a currency value ranging from -2 63 (or -922,337,203,685,477.5808) to 2 63 -1 (or +922,337,203,685,477.5807) with an accuracy to a ten-thousandth of currency unit to be stored in or retrieved from a database.
- SqlSingle - Represents a floating point number within the range of -3.40E +38 through 3.40E +38 to be stored in or retrieved from a database.
- SqlString - Represents a variable-length stream of characters to be stored in or retrieved from the database. SqlString has a different underlying data structure from its corresponding .NET Framework String data type.

Note that these structures contain their own constructors, properties, methods, operators, and fields. On the next slide, we show details for the `SqlDecimal` structure specifically. For the remaining structures, please see Microsoft's website.

# Structures

## `SqlDecimal` Structure in Detail


### Constructors

- SqlDecimal(Decimal) - Initializes a new instance of the SqlDecimal structure using the supplied Decimal value.
- SqlDecimal(Double) - Initializes a new instance of the SqlDecimal structure using the supplied double parameter.
- SqlDecimal(Int32) - Initializes a new instance of the SqlDecimal structure using the supplied integer value.
- SqlDecimal(Int64) - Initializes a new instance of the SqlDecimal structure using the supplied long integer value.
- SqlDecimal(Byte, Byte, Boolean, Int32[]) - Initializes a new instance of the SqlDecimal structure using the supplied parameters.
- SqlDecimal(Byte, Byte, Boolean, Int32, Int32, Int32, Int32) - Initializes a new instance of the SqlDecimal structure using the supplied parameters.

### Properties

- BinData - Get the binary representation of the value of this SqlDecimal structure as an array of bytes.
- Data - Gets the binary representation of this SqlDecimal structure as an array of integers.
- IsNull - Indicates whether this SqlDecimal structure is null.
- IsPositive - Indicates whether the Value of this SqlDecimal structure is greater than zero.
- Precision - Gets the maximum number of digits used to represent the Value property.
- Scale - Gets the number of decimal places to which Value is resolved.
- Value - Gets the value of the SqlDecimal structure. This property is read-only.

### Methods

- Abs - The Abs method gets the absolute value of the SqlDecimal parameter.
- Add - Calculates the sum of the two SqlDecimal operators.
- AdjustScale - The scale of the SqlDecimal operand will be adjusted to the number of digits indicated by the digits parameter. Depending on the value of the fRound parameter, the value will either be rounded to the appropriate number of digits or truncated.
- Ceiling - Returns the smallest whole number greater than or equal to the specified SqlDecimal structure.
- CompareTo(Object) - Compares this SqlDecimal instance to the supplied Object and returns an indication of their relative values.
- CompareTo(SqlDecimal) - Compares this SqlDecimal instance to the supplied SqlDecimal object and returns an indication of their relative values.
- ConvertToPrecScale - Adjusts the value of the SqlDecimal operand to the indicated precision and scale.
- Divide - The division operator calculates the results of dividing the first SqlDecimal operand by the second.
- Equals(Object) - Compares the supplied Object parameter to the Value property of the SqlDecimal instance. (Overrides ValueType.Equals(Object).)
- Equals(SqlDecimal, SqlDecimal) - Performs a logical comparison of the two SqlDecimal operands to determine whether they are equal.

# Structures

## Methods (continued)


- Finalize - Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
- Floor - Rounds a specified SqlDecimal number to the next lower whole number.
- GetHashCode - Returns the hash code for this instance. (Overrides ValueType.GetHashCode().)
- GetType - Gets the Type of the current instance. (Inherited from Object.)
- GetXsdType - Returns the XML Schema definition language (XSD) of the specified XmlSchemaSet.
- GreaterThan - Performs a logical comparison of two SqlDecimal structures to determine whether the first is greater than the second.
- GreaterThanOrEqual - Performs a logical comparison of the two SqlDecimal parameters to determine whether the first is greater than or equal to the second.
- LessThan - Performs a logical comparison of two SqlDecimal structures to determine whether the first is less than the second.
- LessThanOrEqual - Performs a logical comparison of the two SqlDecimal parameters to determine whether the first is less than or equal to the second.
- MemberwiseClone - Creates a shallow copy of the current Object. (Inherited from Object.)
- Multiply - The multiplication operator computes the product of the two SqlDecimal parameters.
- NotEquals - Performs a logical comparison of the two SqlDecimal parameters to determine whether they are not equal.
- Parse - Converts the String representation of a number to its SqlDecimal equivalent.
- Power - Raises the value of the specified SqlDecimal structure to the specified exponential power.
- Round - Gets the number nearest the specified SqlDecimal structure's value with the specified precision.
- Sign - Gets a value that indicates the sign of a SqlDecimal structure's Value property.
- Subtract - Calculates the results of subtracting the second SqlDecimal operand from the first.
- ToDouble - Returns the a double equal to the contents of the Value property of this instance.
- ToSqlBoolean - Converts this SqlDecimal structure to SqlBoolean.
- ToSqlByte - Converts this SqlDecimal structure to SqlByte.
- ToSqlDouble - Converts this SqlDecimal structure to SqlDouble.
- ToSqlInt16 - Converts this SqlDecimal structure to SqlInt16.
- ToSqlInt32 - Converts this SqlDecimal structure to SqlInt32.
- ToSqlInt64 - Converts this SqlDecimal structure to SqlInt64.
- ToSqlMoney - Converts this SqlDecimal structure to SqlMoney.
- ToSqlSingle - Converts this SqlDecimal structure to SqlSingle.
- ToSqlString - Converts this SqlDecimal structure to SqlString.
- ToString - Converts this SqlDecimal structure to String. (Overrides ValueType.ToString().)
- Truncate - Truncates the specified SqlDecimal structure's value to the that you want position.

# Structures


## Operators

- Addition - Calculates the sum of the two SqlDecimal operators.
- Division - The division operator calculates the results of dividing the first SqlDecimal operand by the second.
- Equality - Performs a logical comparison of the two SqlDecimal operands to determine whether they are equal.
- Explicit(Double to SqlDecimal) - Converts the Double parameter to SqlDecimal.
- Explicit(SqlBoolean to SqlDecimal) - Converts the supplied SqlBoolean structure to SqlDecimal.
- Explicit(SqlDecimal to Decimal) - Converts the SqlDecimal parameter to Decimal.
- Explicit(SqlDouble to SqlDecimal) - Converts the supplied SqlDouble structure to SqlDecimal.
- Explicit(SqlSingle to SqlDecimal) - Converts the supplied SqlSingle structure to SqlDecimal.
- Explicit(SqlString to SqlDecimal) - Converts the supplied SqlString parameter to SqlDecimal.
- GreaterThan - Performs a logical comparison of two SqlDecimal structures to determine whether the first is greater than the second.
- GreaterThanOrEqual - Performs a logical comparison of the two SqlDecimal parameters to determine whether the first is greater than or equal to the second.
- Implicit(Decimal to SqlDecimal) - Converts the Decimal value to SqlDecimal.
- Implicit(Int64 to SqlDecimal) - Converts the supplied Int64 structure to SqlDecimal.
- Implicit(SqlByte to SqlDecimal) - Converts the supplied SqlByte structure to SqlDecimal.
- Implicit(SqlInt16 to SqlDecimal) - Converts the supplied SqlInt16 structure to SqlDecimal
- Implicit(SqlInt32 to SqlDecimal) - Converts the supplied SqlInt32 structure to SqlDecimal.
- Implicit(SqlInt64 to SqlDecimal) - Converts the supplied SqlInt64 structure to SqlDecimal.
- Implicit(SqlMoney to SqlDecimal) - Converts the SqlMoney operand to SqlDecimal.
- Inequality - Performs a logical comparison of the two SqlDecimal parameters to determine whether they are not equal.
- LessThan - Performs a logical comparison of two SqlDecimal structures to determine whether the first is less than the second.
- LessThanOrEqual - Performs a logical comparison of the two SqlDecimal parameters to determine whether the first is less than or equal to the second.
- Multiply - The multiplication operator computes the product of the two SqlDecimal parameters.
- Subtraction - Calculates the results of subtracting the second SqlDecimal operand from the first.
- UnaryNegation - The unary minus operator negates the SqlDecimal parameter.

## Fields

- MaxPrecision - A constant representing the largest possible value for the Precision property.
- MaxScale - A constant representing the maximum value for the Scale property.
- MaxValue - A constant representing the maximum value of a SqlDecimal structure.
- MinValue - A constant representing the minimum value for a SqlDecimal structure.
- Null - Represents a DBNull that can be assigned to this instance of the SqlDecimalclass.

# Structures


One comment on Microsoft's website should convince you of how important it is to use the `SqlDecimal` (as well as other Sql* types) when pulling data from or pushing data to SQL Server: *SqlDecimal has different underlying data structures from its corresponding .NET Framework Decimal data type. Decimal has no concept of precision. It uses 3 bytes to store the actual data, and therefore has a maximum scale of 28. The data range is -79,228,162,514,264,337,593,543,950,335 through 79,228,162,514,264,337,593,543,950,335. SqlDecimal has both precision and scale. It uses 4 unsigned 4-byte integers to store the actual data, and therefore has maximum precision and scale of 38. The data range is - 10^38 +1 through 10^38 - 1.*

Please see the following website for more information: http://msdn.microsoft.com/en-us/library/ms172136.aspx. Here is a comment from that webpage: *SQL Server and the .NET Framework are based on different type systems, which can result in potential data loss. To preserve data integrity, the .NET Framework Data Provider for SQL Server (System.Data.SqlClient) provides typed accessor methods for working with SQL Server data. You can use the enumerations in the SqlDbType classes to specify SqlParameter data types.*

Don't forget that your parameters should also be set to use the data types in `System.Data.SqlTypes` as well!


# The `System.Data.SqlTypes` Namespace

→ Interfaces

# Interfaces

There is one interfaces available in the `System.Data.SqlTypes` namespace.


## Interfaces

- INullable - All the System.Data.SqlTypes objects and structures implement the INullable interface.


# The `System.Data.SqlTypes` Namespace

→ Delegates

# Delegates

There are no delegates in the `System.Data.SqlTypes` namespace.


# The `System.Data.SqlTypes` Namespace

→ Enumerations

# Enumerations

The following are the enumerations available in the `System.Data.SqlTypes` namespace.


## Enumerations

- SqlCompareOptions - Specifies the compare option values for a SqlString structure.
- StorageState - The StorageState enumeration is not intended for use as a stand-alone component, but as an enumeration from which other classes derive standard functionality.

## The `SqlCompareOptions` enumeration contains the following members:

- None - Specifies the default option settings for SqlString comparisons.
- IgnoreCase - Specifies that SqlString comparisons must ignore case.
- IgnoreNonSpace - Specifies that SqlString comparisons must ignore nonspace combining characters, such as diacritics. The Unicode Standard defines combining characters as characters that are combined with base characters to produce a new character. Non-space combining characters do not use character space by themselves when rendered. For more information about non-space combining characters, see the Unicode Standard at http://www.unicode.org.
- IgnoreKanaType - Specifies that SqlString comparisons must ignore the Kana type. Kana type refers to Japanese hiragana and katakana characters that represent phonetic sounds in the Japanese language. Hiragana is used for native Japanese expressions and words, while katakana is used for words borrowed from other languages, such as "computer" or "Internet". A phonetic sound can be expressed in both hiragana and katakana. If this value is selected, the hiragana character for one sound is considered equal to the katakana character for the same sound.
- IgnoreWidth - Specifies that SqlString comparisons must ignore the character width. For example, Japanese katakana characters can be written as full-width or half-width and, if this value is selected, the katakana characters written as full-width are considered equal to the same characters written in half-width.
- BinarySort - Specifies that sorts should be based on a characters numeric value instead of its alphabetical value.
- BinarySort2 - Performs a binary sort.

## The `StorageState` enumeration contains the following members:

- Buffer - Buffer size.
- Stream - Stream.
- UnmanagedBuffer - Unmanaged buffer.


# The `System.Data.SqlTypes` Namespace

→ Exceptions

# Exceptions

The following are the exceptions in the `System.Data.SqlTypes` namespace.


## Exceptions

- SqlAlreadyFilledException - The SqlAlreadyFilledException class is not intended for use as a stand-alone component, but as a class from which other classes derive standard functionality.
- SqlNotFilledException - The SqlNotFilledException class is not intended for use as a stand-alone component, but as a class from which other classes derive standard functionality.
- SqlNullValueException - The exception that is thrown when the Value property of a System.Data.SqlTypes structure is set to null.
- SqlTruncateException - The exception that is thrown when you set a value into a System.Data.SqlTypes structure would truncate that value.
- SqlTypeException - The base exception class for the System.Data.SqlTypes.

# What Next?


In *C# Programming IV-#*, we look at specific classes within specific namespaces such as the System namespace, the System.Data.SqlTypes namespace, etc.

# References

*Click the book titles below to read more about these books on Amazon.com's website.*


- Introducing Microsoft LINQ, Paolo Pialorsi and Marco Russo, Microsoft Press, ISBN:9780735623910
- LINQ Pocket Reference, Joseph Albahari and Ben Albahari, O'Reilly Press, ISBN:9780596519247
- Inside C#, Tom Archer and Andrew Whitechapel, Microsoft Press, ISBN:0735616485
- C# 4.0 In a Nutshell, O'Reilly Press, Joseph Albahari and Ben Albahari, ISBN:9780596800956
- The Object Primer, Scott W. Ambler, Cambridge Press, ISBN:0521540186
- CLR via C#, Jeffrey Richter, Microsoft Press, ISBN:9780735621633

## Support sheepsqueezers.com

If you found this information helpful, please consider supporting sheepsqueezers.com. There are several ways to support our site:

- ☐ Buy me a cup of coffee by clicking on the following link and donate to my PayPal account: Buy Me A Cup Of Coffee?.
- ☐ Visit my Amazon.com Wish list at the following link and purchase an item: http://amzn.com/w/3OBK1K4EIWIR6

Please let me know if this document was useful by e-mailing me at `comments@sheepsqueezers.com`.