

Language Integrated Query (LINQ)

TOP NEW FEATURES

- Write queries against in-memory collections and other queryable sources in any .NET language using LINQ.
- Write data access code directly against a Microsoft SQL Server database by using LINQ to SQL.
- Write data access code for XML data by using LINQ to XML.
- Write common data access code across different data providers and storage schemas by using LINQ to Entities.
- Write rich queries against data in a Microsoft DataSet by using LINQ to DataSet

Microsoft® SQL Server™ 2008 LINQ is an exciting evolution in data programming that enables developers to build queries over data directly from within any .NET-based programming language. LINQ to Entities, LINQ to DataSet, LINQ to SQL, and LINQ to XML are key implementations of this powerful technology.

<http://www.microsoft.com/sqlserver/2008/en/us/linq.aspx>

SIMPLICITY

Simplify your data access code and enhance the maintainability of your applications by using LINQ. Express traversal, filter, and projection operations declaratively by using any .NET-based programming language with LINQ's set of general purpose *query operators*.

Query data in any .NET programming language

Write queries by using LINQ in the programming language of your choice instead of learning the SQL dialect native to your data source.

```
var customers = from c in
    Customers
    where c.City == "London"
    select c;
```

.NET language support

Use strongly typed objects in your data access code

Query against strongly typed data objects easily. Write data access code in the same object-oriented manner used for the rest of your application.

PRODUCTIVITY

Increase productivity and reduce run-time errors in your applications by using strongly typed objects instead of embedded SQL query syntax.

Optimize development efforts

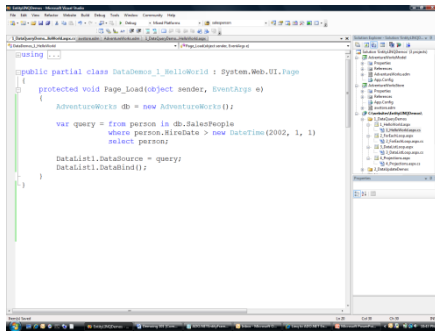
Become more productive and optimize the overall application development effort by using a single consistent query language for all aspects of the application.

Reduce bugs and errors

Work with strongly typed CLR objects that reduce run-time errors in applications. Identify query-related coding errors at compile time and reduce the debugging effort by using strongly typed variables.

Be more productive with Microsoft Visual Studio

Maximize developer efficiency when writing code that includes strongly typed data objects and take full advantage of the productivity enhancing capabilities of Visual Studio®, such as the object browser and IntelliSense®.



LINQ in Visual Studio

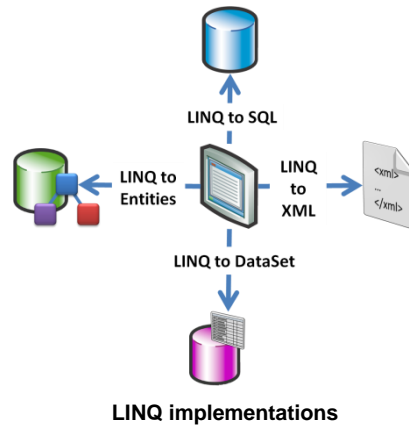
FLEXIBILITY

Access data in a wide range of data stores while using consistent LINQ syntax. Easily adapt to the needs of a particular application scenario by taking advantage of several data source-specific implementations of LINQ to query various types of data. **Use LINQ with any data source**

Use whichever implementation of LINQ is designed for your scenario:

- LINQ to SQL for objects mapped directly to Microsoft SQL Server database schemas
- LINQ to XML for XML data
- LINQ to Entities for entity objects mapped to ADO.NET Data Providers

- LINQ to DataSet to work with existing DataSet functionality



LINQ to Entities: Design flexible mappings

Build enterprise-grade applications using strongly typed objects mapped to different data sources including SQL Server and third-party databases. Use a common set of business objects that may differ from, and evolve independently of, your database storage schema, using flexible mapping support in the ADO.NET Entity Framework. Write one set of code to access data from a variety of data sources and schemas that can be interchanged or evolved without changing the application.

New in Visual Studio 2008 SP1: Connect to SQL Server 2008 and use the new SLQ 2008 data types such as Date, DateTime2, DateTimeOffset, Time, and FileStream.

LINQ to SQL: Optimize SQL Server client development

Rapidly build client applications directly against Microsoft SQL Server storage schemas. Work with strongly typed objects that are mapped directly to the underlying database schema.

New in Visual Studio 2008 SP1: Connect to SQL Server 2008 and use the new SLQ 2008 Data types such as Date, DateTime2, DateTimeOffset, Time, and FileStream.

LINQ to XML: Work with XML data in a familiar way

Use LINQ to XML to query strongly typed XML objects that represent XML data. LINQ to XML provides a comprehensive in-memory XML programming API that enables you to work with XML data in a familiar way.

For more information about LINQ, visit <http://msdn.microsoft.com/en-us/data/cc299380.aspx>. For the latest developer news and more information on Microsoft's broad range of resources for developers, including support programs, events, training, and the MSDN Library Online, visit MSDN Online at <http://msdn.microsoft.com/>.

This data sheet is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. (Use this only if fictitious content appears.)

0308 Part No. 098-00000 (color)
Part No. 098-00000 (black and white)