The twist, however, is that a query expression can be used to interact with numerous types of data—

even data that has nothing to do with a relational database. Strictly speaking, “LINQ” is the term used to

describe this overall approach to data access. However, based on where you are applying your LINQ

queries, you will encounter various terms, such as the following:

• *LINQ to Objects:* This term refers to the act of applying LINQ queries to arrays and

collections.

• *LINQ to XML:* This term refers to the act of using LINQ to manipulate and query

XML documents.

• *LINQ to DataSet:* This term refers to the act of applying LINQ queries to ADO.NET

DataSet objects.

• *LINQ to Entities:* This aspect of LINQ allows you to make use of LINQ queries

within the ADO.NET Entity Framework (EF) API.

• *Parallel LINQ* (a.k.a. *PLINQ*): This allows for parallel processing of data returned

from a LINQ query.

To be sure, Microsoft seems quite dedicated to integrating LINQ support deeply within the .NET

programming environment. Today, LINQ is an integral part of the .NET base class libraries, managed

languages, and Visual Studio itself.



