

Programming in C# Jump Start

Jerry Nixon | Microsoft Developer Evangelist

Daren May | President & Co-founder, Crank211

08 | Advanced C#, Part 4

Jerry Nixon | Microsoft Developer Evangelist
Daren May | President & Co-founder, Crank211

Module Overview

- Accessing a Database
- Using LINQ

Using LINQ to Objects and XML

MVA

Microsoft
Virtual
Academy

What is a database?

- A database is a data store
- Data stored in a database is structured
- Data stored in a database may be related
 - Referential integrity
- A database provides a way to access/query data
 - Optimized by indexes

What databases can we use?

- Windows Azure SQL Database
- Local Network SQL Server
- Local Machine SQL Server Express
- Application SQL LocalDB
- Application SQL CE
- Other providers: Oracle, SQLite, MySql, DB2, Jet
 - ADO.Net implements a provider model enabling many (if not all) databases. Database implementation is abstracted.

Types of access to a database

- Low-level
 - Manual queries
 - SqlDataReader
- Object Relationship Models (ORM)
 - Conceptual Modelling
 - Entity Framework, Nhibernate, CSLA, Dapper

DEMO



Database Access

What is Language Integrated Query (033)

- LINQ is a general-purpose Query Language
- LINQ is an *integrated* part of the .Net languages
- LINQ is Type Safe and has Intellisense
- LINQ includes operators like Traversal, Filter, and Projection
- LINQ can be optimized with compiler versions
- LINQ can be invoked using its Query Syntax
- LINQ can be invoked using its Method Syntax

```
var data = Enumerable.Range(1, 50);

var method = // IEnumerable<string>
data.Where(x => x % 2 == 0)
.Select(x => x.ToString());

var query = // IEnumerable<string>
from d in data
where d % 2 == 0
select d.ToString();
```

```
var data = Enumerable.Range(1, 50);
```

```
var projection =  
    from d in data  
    select new  
    {  
        Even = (d % 2 == 0),  
        Odd = !(d % 2 == 0),  
        Value = d,  
    };
```

```
var letters = new[]{"A", "C", "B", "E", "Q"};
```

```
var sortAsc =  
    from d in data  
    orderby d ascending  
    select d;
```

```
var sortDesc =  
    data.OrderByDescending(x => x);
```

```
var values = new[] { "A", "B", "A", "C", "A", "D" };

var distinct = values.Distinct();

var first = values.First();
var firstOr = values.FirstOrDefault();

var last = values.Last();

var page = values.Skip(2).Take(2);
```

```
// aggregates
```

```
var numbers = Enumerable.Range(1, 50);
```

```
var any = numbers.Any(x => x % 2 == 0);
```

```
var count = numbers.Count(x => x % 2 == 0);
```

```
var sum = numbers.Sum();
```

```
var max = numbers.Max();
```

```
var min = numbers.Min();
```

```
var avg = numbers.Average();
```

```
var dictionary1 = new Dictionary<string, string>()
{
    {"1", "B"}, {"2", "A"}, {"3", "B"}, {"4", "A"},
};

var dictionary2 = new Dictionary<string, string>()
{
    {"5", "B"}, {"6", "A"}, {"7", "B"}, {"8", "A"},
};

var join =
    from d1 in dictionary1
    join d2 in dictionary2 on d1.Value equals d2.Value
    select new
    {
        Key1 = d1.Key,
        Key2 = d2.Key,
        Value = d1.Value
    };
```

```
var dictionary = new Dictionary<string, string>()
{
    {"1", "B"}, {"2", "A"}, {"3", "B"}, {"4", "A"},
};

var group = // IEnumerable<string, IEnumerable<string>>
from d1 in dictionary
group d1 by d1.Value into g
select new
{
    Key = g.Key,
    Members = g,
};
```


What is LINQ to XML? (034)

- One of the most common data formats is XML
- LINQ to XML provides LINQ syntax over an XML document
 - Read XML Documents
 - Create XML Document
- **LINQ to XML versus XSLT**
 - XSLT is a script for creating XML from XML
 - XSLT cannot create .Net objects
 - XSLT cannot reference .Net objects

DEMO



LINQ

Module Recap

- Accessing a Database
- Using LINQ



©2013 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.