Language Integrated Query: An introduction

Damien Guard (BSc, MBCS)

https://damieng.com

Guernsey Software Developers Forum

29 January 2008

What is LINQ?

- Language Integrated Query
- Make query part of the language
- Component of .NET Framework 3.5
- Now shiping with Visual Studio 2008

Query without LINQ

- Objects using loops and conditions

```
foreach(Customer c in customers)
  if (c.Region == "UK") ...
```

- Databases using SQL

```
SELECT * FROM Customers WHERE Region='UK'
```

- XML using XPath/XQuery

```
//Customers/Customer[@Region='UK']
```

ADO without LINQ

```
SqlConnection con = new SqlConnection(...);
con.Open();
SqlCommand cmd = new SqlCommand(
  @"SELECT * FROM Customers WHERE c.Region = @Region", con);
cmd.Parameters.AddWithValue("@Region", "UK");
DataReader dr = cmd.ExecuteReader();
while (dr.Read()) {
   string name = dr.GetString(dr.GetOrdinal("Name"));
   string phone = dr.GetString(dr.GetOrdinal("Phone"));
   DateTime date = dr.GetDateTime(3);
}
dr.Close();
con.Close();
```

Query with LINQ

C#

```
var myCustomers = from c in customers
    where c.Region == "UK"
    select c;
```

VB. NET

```
Dim myCustomers = From c In customers _
    Where c.Region = "UK" _
    Select c
```

More LINQ queries

C#

```
var goodCusts = (from c in db.Customers
     where c.PostCode.StartsWith("GY")
     orderby c.Sales descending
     select c).Skip(10).Take(10);
```

VB. NET

```
Dim goodCusts = (From c In db.Customers _
     Where c.PostCode.StartsWith("GY") _
     Order By c.Sales Descending _
     Select c).Skip(1).Take(10)
```

Advantages

- **Unified data access** - Single syntax to learn and remember
- **Strongly typed** - Catch errors during compilation
- **IntelliSense** - Prompt for syntax and attributes
- **Bindable result sets** - In some providers

LINQ to Objects

C#

```
int[] nums = new int[] {0,4,2,6,3,8,3,1};
double average = nums.Take(6).Average();
var above = from n in nums
            where n > average
            select n;
```

VB. NET

```
Dim nums() As Integer = {0,4,2,6,3,8,3,1}
Double average = nums.Take(6).Average()
Dim above = From n In nums _
            Where n > average _
            Select n
```

LINQ to Objects

- Query any IEnumerable source
  Includes arrays, List, Dictionary…
- Many useful operators available
  Sum, Max, Min, Distinct, Intersect, Union
- Expose your own data with
  IEnumerable or IQueryable
- Create operators using extension methods

LINQ operators

Aggregate	Conversion	Ordering	Partitioning	Sets
Aggregate	Cast	OrderBy	Skip	Concat
Average	OfType	ThenBy	SkipWhile	Distinct
Count	ToArray	Descending	Take	Except
Max	ToDictionary	Reverse	TakeWhile	Intersect
Min	ToList			Union
Sum	ToLookup			

LINQ to SQL

- Object-relational mapping
  Records become strongly-typed objects
- Data context is the controller mechanism
- Facilitates update, delete & insert
- Translates LINQ queries behind the scenes
- Type, parameter and injection safe

Database mapping

- VS 2008 designer or SQLMetal command
- Map tables & fields to classes & properties
- Generates partial classes with attributes
- Each record becomes an object
- Data context represents the database
- Utilise tables, views or stored procedures

Modifying objects

- Update

  Set object properties

- Delete

  `context.Table.DeleteOnSubmit(object)`

- Insert

  `context.Table.InsertOnSubmit(object)`

- Commit changes back

  `context.SubmitChanges()`

  Transactional - all or nothing

Demo of LINQ to SQL

Additional providers

- Relational data

  NHibernate, MySQL, Oracle, PostgreSQL

- Web services

  RDF, Flickr, Amazon, WebQueries

- Custom

  LDAP, Google Desktop, SharePoint, TerraServer maps

Future developments

- Blinq

  Scaffold web UI for list/view/update pages
- PLINQ

  Parallel query processing over many CPUs
- SyncLINQ & Continuous LINQ

  Updated results via INotifyCollectionChanged

Limitations

LINQ

- Only defines query, not update or context

LINQ to SQL

- Mapping is set at compile-time
- Can not mix mapped and unmapped properties in a single query
- Microsoft SQL Server 2000 or later only

.NET features used

.NET Framework 2.0

- Partial classes (mapping)

.NET Framework 3.5

- Anonymous types (shaping)
- Extension method (query operators)
- Type inference (var keyword)
- Lambda expressions (query syntax)

Alternatives for .NET

- NHibernate
- Castle MonoRail/ActiveRecord
- SubSonic
- Code generation tool+templates
  CodeSmith, MyGeneration, LLBLGen/Pro +
  NetTiers, DooDads, roll your own…

More information

- Official site - https://msdn.microsoh.com/linq/
- Tutorials - https://weblogs.asp.net/scottgu/
- Screencasts - https://tinyurl.com/yusch
- This presentation & cheat sheet https://damieng.com/blog/tag/linq

Questions & answers