Language Integrated Query: An introduction

Damien Guard (BSc, MBCS)  Guernsey Software Developer Forum
damien@envytech.co.uk
What is LINQ?

- Language Integrated Query
- Make query a part of the language
- Component of .NET Framework 3.5
- Now shipping with Visual Studio 2008
Query without LINQ

- Objects using loops and conditions
  ```csharp
  foreach(Customer c in customers)
  if (c.Region == "UK") ...
  ```

- Databases using SQL
  ```sql
  SELECT * FROM Customers WHERE Region='UK'
  ```

- XML using XPath/XQuery
  ```xml
  //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
Architecture

- C#
- VB.NET
- Others

.NET Language Integrated Query (LINQ)

LINQ data source providers
- LINQ to Objects
- LINQ to Datasets
- LINQ to SQL
- LINQ to Entities
- LINQ to XML

ADO.NET support for LINQ
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<T> source
  Includes arrays, List<T>, Dictionary...
- Many useful operators available
  Sum, Max, Min, Distinct, Intersect, Union
- Expose your own data with
  IEnumerable<T> or IQueryable<T>
- Create operators using extension methods
## LINQ operators

<table>
<thead>
<tr>
<th>Aggregate</th>
<th>Conversion</th>
<th>Ordering</th>
<th>Partitioning</th>
<th>Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate</td>
<td>Cast</td>
<td>OrderBy</td>
<td>Skip</td>
<td>Concat</td>
</tr>
<tr>
<td>Average</td>
<td>OfType</td>
<td>ThenBy</td>
<td>SkipWhile</td>
<td>Distinct</td>
</tr>
<tr>
<td>Count</td>
<td>ToArray</td>
<td>Descending</td>
<td>Take</td>
<td>Except</td>
</tr>
<tr>
<td>Max</td>
<td>ToDictionary</td>
<td>Reverse</td>
<td>TakeWhile</td>
<td>Intersect</td>
</tr>
<tr>
<td>Min</td>
<td>ToList</td>
<td></td>
<td></td>
<td>Union</td>
</tr>
<tr>
<td>Sum</td>
<td>ToLookup</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ToSequence</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

and many others
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, 2005 only
.NET features used

.NET Framework 2.0
- Partial classes (mapping)

.NET Framework 3.5
- Anonymous types (shaping)
- Extension methods (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 - msdn.microsoft.com/linq/
- Tutorials - weblogs.asp.net/scottgu/
- Screencasts - tinyurl.com/yuscft
- This presentation & cheat sheet damieng.com/blog/tag/linq
Questions & answers