
Contents at a Glance

Part I LINQ Essentials

1	Introducing LINQ	3
2	Using LINQ to Objects	31
3	Handling LINQ to SQL with Visual Studio	67

Part II LINQ to Any Data Source

4	Working with ADO.NET Through LINQ to DataSet	111
5	Programming Objects with LINQ to Entities	119
6	Programming Objects with LINQ to XML	149
7	Automatically Generating LINQ to SQL Code with SqlMetal	171

Part III Extending LINQ

8	Digging Into Expression Trees and Lambdas	187
9	Constructing New Code with Extension Methods	209
10	Building a Custom LINQ Provider—Introducing LINQ to Twitter	215
11	Designing Applications with LINQ	235
12	Concurrent Programming with Parallel LINQ (PLINQ)	271
A	Standard Query Operator Reference	285
	Index	347

Contents

Foreword	xvii
Acknowledgments	xix
Introduction	xxi

Part I LINQ Essentials

1 Introducing LINQ	3
Getting Started	4
Example Data	4
Viewing Assembly Contents and Intermediate Language	5
Auto-Implemented Properties	5
Using Auto-Implemented Properties	5
A Few More Tips	6
In Case You're Curious	6
Partial Methods	8
How Are Partial Methods Used?	9
A Few More Tips	10
In Case You're Curious	10
Extension Methods	12
Creating Extension Methods	12
Using Extension Methods	13
Extension Method Precedence	13
In Case You're Curious	14
Object Initializers	17
Using Object Initializers	17
In Case You're Curious	18
Collection Initializers	20
Implementing Collection Initializers	20
Collection Initializers for Dictionary Collections	21
Lambda Expressions	22
How to Use a Lambda Expression	22
Replacing Anonymous Methods and Delegates	23
A Few Tips on Lambda Expressions	24
Implicitly Typed Local Variables	24
Query Syntax	25
A First Look at a Very Simple Query	25
Using Sequences from a Query	25
Anonymous Types	26
A Simple Anonymous Type	26
Anonymous Types Are Strongly Typed	26
In Case You're Curious	27
Summary	29

2	Using LINQ to Objects	31
	LINQ to Objects Essentials	32
	LINQ to Objects Benefits	32
	The Example Collections	34
	Implementing Projections	35
	Collection Object Projections	36
	LINQ to Objects Queries Return IEnumerable<T>	36
	Selecting a Single Field or Property	37
	Projecting into Anonymous Types	37
	Projecting into Custom Types	39
	Filtering Data	40
	Calculating Intermediate Values	41
	Sorting Query Results	42
	Sorting a Single Property	42
	Sorting Multiple Properties	43
	Managing Sort Direction	43
	Grouping Sets	44
	Grouping by a Single Property	44
	Grouping by Multiple Properties	45
	Accessing Grouped Objects	47
	Joining Objects	48
	Joining Objects	48
	Performing Left Joins	49
	Performing Group Joins	50
	Joining with Composite Keys	51
	Performing Select Many Joins	54
	Flattening Object Hierarchies	54
	Performing Cross-Joins	56
	Querying Non-IEnumerable<T> Collections	57
	A Practical Example of LINQ to Objects	59
	Summary	64
3	Handling LINQ to SQL with Visual Studio	67
	LINQ to SQL Capabilities	68
	Introducing the Example Database	68
	Understanding the DataContext	70
	Creating a DataContext with the VS 2008 LINQ to SQL Designer	70
	Database Connection Management	73
	Object Mapping	76
	Object Tracking	83
	Defining Entity Relationships	84
	Adding Multiple Entities to a DataContext	84
	Demonstrating Entity Associations	85
	Handling Multiple Relationships Between Two Entities	87
	Examining Association Attributes	88
	A Closer Look at Queries	89
	Querying with the DataContext	90
	Examining Generated SQL	91
	Examining Query Results	92

Inserting, Updating, and Deleting	97
Inserting New Data	97
Updating Existing Data	98
Deleting Existing Data	98
Attaching Objects	99
Implementing Stored Procedures and Functions	101
Stored Procedures	101
Functions	102
Executing Raw SQL	103
Entity Inheritance	104
Summary	107

Part II LINQ to Any Data Source

4 Working with ADO.NET Through LINQ to DataSet	111
Setting Up the DataSet	112
Querying a DataSet	113
Querying DataSet Tables	113
Reading DataRow Results	113
Modifying a DataSet	114
Working with Strongly Typed DataSets	115
Creating a Strongly Typed DataSet	115
Querying a Strongly Typed DataSet	115
Special DataSet Operations	117
Creating DataViews from LINQ to DataSet Queries	117
Copying LINQ to DataSet Queries to DataTables	118
Summary	118
5 Programming Objects with LINQ to Entities	119
Introduction to Entities	120
The ADO.NET Entity Framework (AEF) Architecture	121
Creating an Entity Data Model (EDM)	122
Inside the Entity Data Model (EDM)	126
Accessing the EDM with theObjectContext	126
Examining EDM Entity Types	132
Top-Down EDM Design	134
What Is the Top-Down Design Approach?	135
Setting Up an EDM Using Top-Down Design	135
Creating New Entities Yourself	135
Mapping Entities to the Database	142
Querying the EDM with LINQ to Entities	147
Summary	147
6 Programming Objects with LINQ to XML	149
The LINQ to XML Objects	150
Creating XML Documents	151
Functional Construction	151
Saving XML Documents	152

Retrieving XML from Various Sources	153
Creating XML from Strings	153
Reading XML from Streams	154
Reading XML from Files	154
Reading XML from URIs	155
Reading XML from XmlReaders	155
Working with LINQ to XML Namespaces	156
Creating and Using a Namespace	156
Creating Namespace Prefixes	156
Working with Documents	157
Adding Declarations, Text, and Processing Instructions	158
Including Declarations	158
Defining CDATA Text	159
Handling Processing Instructions	160
Validating XML	160
Queries and Axes	162
Basic LINQ to XML Queries	162
LINQ to XML Axis Methods	163
Axis Method Filters	166
Manipulating XML	166
Adding New Elements	167
Modifying Existing Elements	167
Removing Elements	168
Summary	169
7 Automatically Generating LINQ to SQL Code with SqlMetal	171
Using SqlMetal.exe	172
Database Connection Options	172
Working with DBML Files	174
SqlMetal Source Code Options	177
Implementing External Mapping Files	178
Generating an External Mapping File with SQLMetal.exe	179
Using the External Mapping File in Your Code	181
Summary	184
Part III Extending LINQ	
8 Digging Into Expression Trees and Lambdas	187
Working with Lambda Expressions	188
More Lambda Examples	188
Lambdas and the Func Delegates	190
Building Expression Trees	192
Expression Tree and Lambda Expression Conversions	192
Expression Tree Internals	193
Dynamic LINQ Queries with Expression Trees	197
The Where Or Problem	198
A Dynamic Query Building Algorithm	199
Executing the Dynamic Query	205
Summary	207

9	Constructing New Code with Extension Methods	209
	Relating Query Expressions to Extension Methods	210
	Matching Extension Methods	210
	How Query Expressions Translate	211
	Extension Method Composability	211
	Building a Custom Extension Method	213
	Summary	214
10	Building a Custom LINQ Provider—Introducing LINQ to Twitter	215
	Introducing LINQ to Twitter	216
	Overview of the LINQ Provider Development Process	218
	.NET Framework Interfaces	218
	Implementation Types	219
	Data Source Communication	219
	Additional Information Resources	221
	Implementing LINQ Provider Interfaces	222
	Understanding the IQueryable<T> Interface	222
	Understanding the IQueryProvider Interface	223
	Understanding IOrderedQueryable<T>	224
	Implementing IQueryable<T>	224
	Implementing IQueryProvider	226
	Building TwitterContext	227
	TwitterContext Instantiation	227
	Exposing IQueryable<T>	228
	Execution—The Bird’s Eye View	229
	Transforming the Query	230
	Communicating with Twitter	232
	Summary	233
11	Designing Applications with LINQ	235
	Introduction to the ASP.NET LinqDataSource	236
	N-Layer Architecture with LINQ	241
	Creating the Data Access Layer (DAL)	242
	Creating the Business Logic Layer (BLL)	243
	Creating the User Interface Layer (UI)	247
	Handling Data Concurrency	254
	Implementing Pessimistic Concurrency with LINQ	255
	Optimistic Concurrency	255
	Working with Deferred Execution	258
	Working with Deferred Loading	259
	What Is Good about Deferred Loading?	260
	What Could Be Problematic about Deferred Loading?	260
	Resolving Deferred Loading Problems	262
	Summary	269
12	Concurrent Programming with Parallel LINQ (PLINQ)	271
	System Requirements and Setup for PLINQ	272
	Running PLINQ Queries	273
	Examining PLINQ Types	273
	Parallelizing a Query	273

Two Threads Are Not Always Better than One	274
Choosing What to Parallelize	275
Setting the Degree of Parallelism	275
Ordering PLINQ Results	276
Additional PLINQ Methods	276
PLINQ Performance Testing	277
Analyzing Performance	277
Considering the Impact of Hardware	280
Handling PLINQ Exceptions	282
Summary	283
A Standard Query Operator Reference	285
Setting Up the Examples	286
Setting Up LINQ to Objects	286
Setting Up LINQ to SQL	289
Organization of the Examples	289
Alphabetical Standard Query Operator Reference	292
Aggregate	292
All	293
Any	294
AsEnumerable	295
AsQueryable	295
Average	296
Cast	297
Concat	297
Contains	298
Count	299
DefaultIfEmpty	300
Distinct	301
ElementAt	302
ElementAtOrDefault	302
Empty	303
Except	303
First	304
FirstOrDefault	305
GroupBy	306
GroupJoin	312
Intersect	314
Join	315
Last	317
LastOrDefault	317
LongCount	318
Max	319
Min	320
OfType	321
OrderBy	321
OrderByDescending	322
Range	323
Repeat	324

Reverse	325
Select	325
SelectMany	326
SequenceEqual	329
Single	330
SingleOrDefault	331
Skip	332
SkipWhile	332
Take	333
TakeWhile	334
ThenBy	335
ThenByDescending	336
ToArray	337
ToDictionary	337
ToList	340
ToLookup	340
Union	343
Where	344
Index	347

territory exposed by the LINQ architecture. Before they can fully understand LINQ, most developers are going to need to become acquainted with several new concepts. The basics of LINQ are easy to understand, but gaining a deep understanding of the conceptual framework behind LINQ requires some work.

This book provides developers with a clear, easy to read map that details the LINQ architecture and supplies us with a well thought out overview of the terrain. Joe Mayo is not only an excellent engineer, but also an excellent writer with an organized and very logical mind. He is the ideal guide to the new and exciting world of LINQ development.

Joe starts out this excellent book with an explanation of the fundamental syntactical elements introduced in C# 3.0 that form the foundation for LINQ. Building on these principles, he goes on to explain how to write query expressions, explaining the syntax in clear, well-written, and logically structured prose. From there, he goes on to cover all the key features of LINQ, include LINQ to SQL, LINQ to XML, and LINQ to DataSets.

The latter half of the book contains an excellent and highly valuable chapter on creating LINQ providers. Near the end of the text, Joe gifts us with an excellent chapter on how to design LINQ applications. This guide to many important LINQ best practices will no doubt become one of the most utilized and favorite sections of this very useful book.

I have read several LINQ books. This text is the best laid out, and the most thoughtful and thorough examination of this subject that I have seen. The book that you hold in your hands is a superior guide to a new world of programming. LINQ is both a practical tool that will help us get our work done more quickly, and also a beautifully designed tool that yields many wonders for those who take the time to study it in-depth. This book is an excellent guide to that wondrous new world, written by an expert working at the peak of his powers. We are all lucky to have such a great new guide to this exciting territory.

—Charlie Calvert
C# Community Program Manager, Microsoft