

HANSER

Inhaltsverzeichnis

Fabrice Marguerie, Steve Eichert, Jim Wooley

LINQ im Einsatz

Übersetzt aus dem Englischen von Walter Doberenz

ISBN: 978-3-446-41429-7

Weitere Informationen oder Bestellungen unter

<http://www.hanser.de/978-3-446-41429-7>

sowie im Buchhandel.



Inhaltsverzeichnis

Geleitwort	15
Vorwort	17
Danksagung	19
Über dieses Buch	21

Teil I: Einführung

1 Einführung in LINQ	27
1.1 Was ist LINQ?	28
1.1.1 Ein Überblick	28
1.1.2 LINQ als Werkzeugsammlung	29
1.1.3 LINQ als Spracherweiterung	30
1.2 Warum brauchen wir LINQ?	32
1.2.1 Allgemeine Probleme	33
1.2.2 Zur Fehlanpassung zwischen den Paradigmen	35
1.2.3 LINQ als Retter in der Not	40
1.3 Entwurfsziele und Ursprünge von LINQ	42
1.3.1 Die Ziele des LINQ-Projekts	42
1.3.2 Ein historischer Rückblick	43
1.4 Die ersten Schritte mit LINQ to Objects	45
1.4.1 Was wir für den Start brauchen	45
1.4.2 Hello LINQ to Objects	47

1.5	Erste Schritte mit LINQ to XML	51
1.5.1	Warum brauchen wir LINQ to XML?	51
1.5.2	Hello LINQ to XML	53
1.6	Erste Schritte mit LINQ to SQL	58
1.6.1	Überblick über die LINQ to SQL-Features	58
1.6.2	Hello LINQ to SQL	59
1.6.3	Ein genauerer Blick auf LINQ to SQL	63
1.7	Zusammenfassung	63
2	Spracherweiterungen in C# und VB	65
2.1	Die neuen Spracherweiterungen erkunden	65
2.1.1	Eine Liste der laufenden Prozesse generieren	66
2.1.2	Gruppieren der Ergebnisse in einer Klasse	68
2.2	Implizit typisierte lokale Variablen	69
2.2.1	Die Syntax	69
2.2.2	Beispiel mit implizit typisierten lokalen Variablen	70
2.3	Objekt- und Collection-Initialisierer	72
2.3.1	Warum wir Objektinitialisierer brauchen	72
2.3.2	Collection-Initialisierer	73
2.3.3	Verbesserung unseres Beispiels mit Objektinitialisierer	74
2.4	Lambda-Ausdrücke	76
2.4.1	Eine Auffrischung zu Delegates	76
2.4.2	Anonyme Methoden	78
2.4.3	Einführung in die Lambda-Ausdrücke	78
2.5	Erweiterungsmethoden	84
2.5.1	Beispiel für das Erzeugen einer Erweiterungsmethode	84
2.5.2	Weitere Beispiele zu den LINQ Standard-Abfrageoperatoren	87
2.5.3	Erweiterungsmethoden im Einsatz	89
2.5.4	Einsatzbeschränkungen	90
2.6	Anonyme Typen	93
2.6.1	Anonyme Typen zum Gruppieren von Daten	93
2.6.2	Namenlose Typen sind trotzdem Typen	93
2.6.3	Verbesserung unseres Beispiels durch anonyme Typen	94
2.6.4	Einschränkungen	95
2.7	Zusammenfassung	97

3	Der LINQ-Baukasten	101
3.1	Wie .NET mit LINQ erweitert wird	101
3.1.1	Die grundlegenden Spracherweiterungen	101
3.1.2	Die Schlüsselemente der LINQ-Foundation	103
3.2	Einführung von Sequenzen	103
3.2.1	IEnumerable<T>	104
3.2.2	Mehr über Iteratoren	105
3.2.3	Verzögerte Ausführung von Abfragen	107
3.3	Einführung in die Abfrageoperatoren	111
3.3.1	Was macht eigentlich ein Abfrageoperator?	111
3.3.2	Die Standard-Abfrageoperatoren	114
3.4	Einführung in Abfrageausdrücke	115
3.4.1	Was ist ein Abfrageausdruck?	115
3.4.2	Schreiben von Abfrageausdrücken	116
3.4.3	Standard-Abfrageoperatoren versus Abfrageausdrücke	118
3.4.4	Einschränkungen	120
3.5	Einführung in die Expression Trees	122
3.5.1	Die Rückkehr der Lambda-Ausdrücke	122
3.5.2	Was sind Expression-Trees?	123
3.5.3	IQueryable und verzögertes Ausführen von Abfragen	125
3.6	LINQ-DLLs und -Namensräume	126
3.7	Zusammenfassung	128

Teil II: LINQ to Objects

4	Arbeiten mit LINQ to Objects	131
4.1	Einführung in unsere Beispielanwendung	131
4.1.1	Unsere Ziele	132
4.1.2	Features	132
4.1.3	Die Geschäfts-Entitäten	133
4.1.4	Datenbankschema	134
4.1.5	Beispieldaten	134
4.2	LINQ für In-Memory-Collections	136
4.2.1	Was können wir abfragen?	137
4.2.2	Unterstützte Operationen	141
4.3	LINQ mit ASP.NET und Windows Forms	142
4.3.1	Datenbindung für Webanwendungen	142

4.3.2	Datenbindung für Windows Forms-Applikationen	149
4.4	Die wichtigsten Standard-Abfrageoperatoren	154
4.4.1	Der Einschränkungs-Operator Where	155
4.4.2	Verwendung von Projektions-Operatoren	156
4.4.3	Anwendung von Distinct	159
4.4.4	Anwenden von Konvertierungsoperatoren	160
4.4.5	Verwendung von Aggregat-Operatoren	161
4.5	Sichten auf einen Objektgraphen im Speicher	162
4.5.1	Sortierung	162
4.5.2	Eingebettete Abfragen	164
4.5.3	Gruppieren	167
4.5.4	Verwenden von Joins	168
4.5.5	Partitionieren	174
4.6	Zusammenfassung	179
5	Jenseits der Standardabfragen	181
5.1	Allgemeine Szenarien	181
5.1.1	Abfrage nichtgenerischer Collections	182
5.1.2	Gruppieren nach Mehrfachkriterien	184
5.1.3	Dynamische Abfragen	187
5.1.4	LINQ to Text Files	197
5.2	Entwurfsmuster	200
5.2.1	Funktionale Konstruktion – ein Pattern	200
5.2.2	Das ForEach-Pattern	203
5.3	Betrachtungen zur Performance	205
5.3.1	Streaming-Strategie	206
5.3.2	Vorsicht bei sofortiger Ausführung	208
5.3.3	Verschlechtert LINQ to Objects die Performance?	210
5.3.4	Gedanken zum Overhead von LINQ to Objects	213
5.3.5	Performance versus Kürze – ein Dilemma?	216
5.4	Zusammenfassung	217

Teil III: LINQ to SQL

6	Der Einstieg in LINQ to SQL	221
6.1	Start mit LINQ to SQL	222
6.1.1	Zuordnung (Mapping) der Objekte	225

6.1.2	Einrichten des DataContext	227
6.2	Lesen von Daten mit LINQ to SQL	228
6.3	Verfeinern unserer LINQ-Abfragen	232
6.3.1	Filtern	232
6.3.2	Sortieren und gruppieren	234
6.3.3	Aggregation	236
6.3.4	Joining	237
6.4	Arbeiten mit Objektbäumen	240
6.5	Wann werden meine Daten geladen?	243
6.5.1	Träges Laden (lazy loading)	243
6.5.2	Sofortiges Laden von Detaildaten	245
6.6	Aktualisieren der Daten	247
6.7	Zusammenfassung	249
7	Details zu LINQ to SQL	251
7.1	Mapping von Objekten mit relationalen Daten	251
7.1.1	Inline-Attribute	252
7.1.2	Mapping mit externen XML-Dateien	257
7.1.3	Das SqlMetal-Tool	259
7.1.4	Der LINQ to SQL-Designer	262
7.2	Übersetzen von Abfrageausdrücken nach SQL	264
7.2.1	IQueryable	264
7.2.2	Ausdrucksbäume	266
7.3	Der Lebenszyklus der Entitäten	269
7.3.1	Verfolgen von Änderungen	270
7.3.2	Änderungen durchführen	271
7.3.3	Arbeiten mit verbindungslosen Daten	274
7.4	Zusammenfassung	276
8	Erweiterte LINQ to SQL-Features	277
8.1	Umgang mit simultanen Änderungen	277
8.1.1	Pessimistische Konkurrenz	278
8.1.2	Optimistische Konkurrenz	278
8.1.3	Behandlung von Konkurrenz-Ausnahmen	281
8.1.4	Auflösen von Konflikten mittels Transaktionen	284
8.2	Fortgeschrittene Datenbankfeatures	286
8.2.1	SQL pass-through: Rückgabe von Objekten aus SQL Abfragen	287
8.2.2	Arbeiten mit gespeicherten Prozeduren	288

8.2.3	Benutzerdefinierte Funktionen	296
8.3	Verbesserungen auf Geschäftsebene	300
8.3.1	Kompilierte Abfragen	300
8.3.2	Partielle Klassen für unsere Geschäftslogik	301
8.3.3	Vorzüge partieller Methoden	303
8.3.4	Vererbung von Objekten	305
8.4	Ein kurzer Abstecher nach LINQ to Entities	310
8.5	Zusammenfassung	313

Teil IV: LINQ to XML

9	Einführung in LINQ to XML	317
9.1	Was ist die LINQ to XML-API?	317
9.2	Wozu brauchen wir noch eine API?	318
9.3	Die Entwurfsprinzipien von LINQ to XML	319
9.3.1	Schlüsselkonzept: Funktionale Konstruktion	321
9.3.2	Schlüsselkonzept: kontext-freie XML-Erzeugung	322
9.3.3	Schlüsselkonzept: Vereinfachte Namen	322
9.4	Die LINQ to XML-Klassenhierarchie	325
9.5	XML mit LINQ verarbeiten	327
9.5.1	XML laden	328
9.5.2	XML parsen	330
9.5.3	XML erzeugen	331
9.5.4	XML mit den Visual Basic XML-Literalen erzeugen	335
9.5.5	XML-Dokumente erstellen	337
9.5.6	Content in XML einfügen	340
9.5.7	Content aus XML-Dokumenten löschen	342
9.5.8	Inhalte ändern	343
9.5.9	Die Verwendung von Attributen	345
9.5.10	XML speichern	346
9.6	Zusammenfassung	346
10	XML abfragen und transformieren	349
10.1	Die LINQ to XML-Achsen-Methoden	350
10.1.1	Element	352
10.1.2	Attribute	352
10.1.3	Elements	354

10.1.4	Descendants (Nachkommen)	355
10.1.5	Ancestors (Vorfahren)	357
10.1.6	Weitere Achsenmethoden	359
10.1.7	Visual Basic XML-Achseigenschaften	360
10.2	Die Standard-Abfrageoperatoren	363
10.2.1	Projektionen mit Select	365
10.2.2	Filtern mit Where	366
10.2.3	Sortieren und Gruppieren	368
10.3	LINQ to XML-Objekte mit XPath abfragen	371
10.4	XML transformieren	373
10.4.1	Transformieren mit LINQ to XML	373
10.4.2	LINQ to XML-Objekte mit XSLT transformieren	376
10.5	Zusammenfassung	378
11	Diverse LINQ to XML-Szenarien	379
11.1	Objekte aus XML-Daten erzeugen	379
11.1.1	Aufgabenstellung	380
11.1.2	Implementierung	382
11.2	XML aus Objektbäumen erzeugen	385
11.2.1	Aufgabenstellung	385
11.2.2	Implementierung	386
11.3	XML aus Datenbankinformationen erzeugen	391
11.3.1	Aufgabenstellung	391
11.3.2	Implementierung	393
11.4	Filtern/Mischen von relationalen und XML-Daten	397
11.4.1	Aufgabenstellung	398
11.4.2	Implementierung	398
11.5	XML-Daten in relationale Daten umwandeln	402
11.5.1	Aufgabenstellung	403
11.5.2	Implementierung	404
11.6	Textdateien nach XML transformieren	417
11.6.1	Aufgabenstellung	417
11.6.2	Implementierung	418
11.7	Zusammenfassung	420

Teil V: Diverse LINQ-Techniken

12 LINQ erweitern	425
12.1 Die Erweiterungsmechanismen von LINQ	426
12.1.1 LINQ-Flavours als LINQ-Implementierungen	426
12.1.2 Vorschläge für kundenspezifische LINQ-Erweiterungen	428
12.2 Erstellen benutzerdefinierter Abfrageoperatoren	429
12.2.1 Verbesserungen der Standard-Abfrageoperatoren	429
12.2.2 Hilfs- oder bereichsspezifische Abfrageoperatoren	432
12.3 Neuimplementierungen der Basis-Abfrageausdrücke	436
12.3.1 Wiederholung: Der Übersetzungsmechanismus für Abfragen	437
12.3.2 Spezifizierung des Musters für Abfrageausdrücke	438
12.3.3 Beispiel 1: Verfolgen der Ausführung von Standard-Abfrageoperatoren ...	440
12.3.4 Einschränkung: Kollision von Abfrageausdrücken	442
12.3.5 Beispiel 2: Nichtgenerische, domänenspezifische Operatoren	443
12.3.6 Beispiel 3: Non-Sequence Operator	445
12.4 Abfragen eines Webdienstes: LINQ to Amazon	447
12.4.1 Einführung in LINQ to Amazon	447
12.4.2 Anforderungen	449
12.4.3 Implementierung	450
12.5 IQueryable und IQueryProvider: LINQ to Amazon erweiterte Version	458
12.5.1 Die IQueryable- und IQueryProvider- Schnittstellen	458
12.5.2 Implementierung	462
12.5.3 Was genau passiert	463
12.6 Zusammenfassung	464
13 LINQ auf allen Ebenen	465
13.1 Überblick über die LinqBooks-Applikation	465
13.1.1 Features	465
13.1.2 Überblick über die Benutzerschnittstelle	466
13.1.3 Das Datenmodell	468
13.2 LINQ to SQL und die Datenzugriffsschicht	469
13.2.1 Auffrischkurs zur traditionellen 3-Schichten-Architektur	469
13.2.2 Brauchen wir eine extra Datenzugriffsschicht oder genügt LINQ to SQL?	471
13.2.3 Beispielanwendungen von LINQ to SQL in LinqBooks	476
13.3 Einsatz von LINQ to XML	483
13.3.1 Datenimport von Amazon	483
13.3.2 Generieren von RSS-Feeds	485

13.4	Einsatz von LINQ to DataSet	486
13.5	Einsatz von LINQ to Objects	489
13.6	Erweiterbarkeit	490
13.6.1	Benutzerspezifische Abfrageoperatoren	490
13.6.2	Erzeugen und Verwenden benutzerspezifischer LINQ-Provider	491
13.7	Ein Blick in die Zukunft	491
13.7.1	Kundenspezifische LINQ-Flavours	492
13.7.2	LINQ to XSD – das typisierte LINQ to XML	493
13.7.3	PLINQ: LINQ und Parallel Computing	494
13.7.4	LINQ to Entities – LINQ für das ADO.NET Entity Framework	494
13.8	Zusammenfassung	495
14	Arbeiten mit LINQ to DataSet	497
14.1	Übersicht zu LINQ to DataSet	497
14.2	Kleiner Auffrischkurs zu DataSets	498
14.2.1	Anwendungsfälle und Features	498
14.2.2	Was sind DataSets?	499
14.2.3	Änderungen in .NET 3.5 für LINQ to DataSet	503
14.3	Abfrage untypisierter DataSets	503
14.3.1	Daten in DataSets laden	504
14.3.2	Abfragen von DataSets ohne LINQ	508
14.3.3	Abfragen untypisierter DataSets mit LINQ to DataSet	509
14.4	Abfrage typisierter DataSets	514
14.4.1	Ein typisiertes DataSet erzeugen	515
14.4.2	Daten in ein typisiertes DataSet laden	518
14.4.3	Abfragen typisierter DataSets mit LINQ to DataSet	520
14.5	Binden der Abfrageergebnisse an die Steuerelemente	523
14.5.1	Verwendung von CopyToDataTable	523
14.5.2	Zwei-Wege-Datenbindung mit AsDataView	526
14.6	Verwendung der Abfrageoperatoren	528
14.6.1	Field<T>- und SetField<T>-Operatoren für DataRow	528
14.7	Set-Operatoren und DataRow-Vergleich	529
14.8	Zusammenfassung	531

Anhang

A Die Standard-Abfrageoperatoren	535
B Weitere Informationen	539
Index	543