

Índice general

CAPÍTULO 1

Pasar de ADO a ADO.NET	1
1.1 Una nueva manera de acceder a los datos	2
1.1.1 El namespace System.Data	2
1.1.2 Proveedores de datos ADO.NET	4
1.2 Los objetos básicos de datos ADO.NET	5
1.3 Creando objetos básicos de datos ADO.NET con SqlClient	7
1.3.1 SqlDataReader con juegos de resultados múltiples	7
1.3.2 XmlReaders con consulta FOR XML AUTO	9
1.3.3 Rellenar un DataGridView con un DataReader	11
1.3.4 Devolver una sola fila de datos	12
1.3.5 Devolver un valor escalar	12
1.3.6 Ejecutar peticiones que no devuelven datos	13
1.4 Aplicar transacciones para actualizar las tablas múltiples ..	14
1.5 Utilizando clases de miembros OleDb, SqlXml, y Odbc ...	17
1.5.1 Substituir OleDb por objetos SqlClient	18
1.5.2 Cambiar SqlConnection y SqlCommand por SqlXmlCommand	19
1.5.3 Probando el proveedor de datos Odbc	20
1.6 Trabajando con datos DataReader y SqlResultSet tipificados	21
1.7 Objetos tipificados DataSet de ADO.NET	22
1.7.1 Añadir un juego de datos tipificado desde un servidor SQL, fuente de datos	25
1.8 Añadir una DataGridView y BindingNavigator Controls ..	29
1.9 Persistir y reabrir el juego de datos	31
1.10 Cambiar de un DataGridView a un Details Form	32
1.11 Añadir un control de vínculo de datos relacionado	33

CAPÍTULO 2

Las novedades de ADO.NET 2.0	37
2.1 Los objetos de formulario	37
2.1.1 Utilizar DbProviderFactories para crear proyectos con bases de datos agnósticas	38
2.1.2 Restablecer los esquemas de las tablas base	39

Bases de datos con Visual Basic

2.2	Comprobar las instancias de servidor SQL disponibles y los proveedores de datos ADO.NET 2.0	44
2.2.1	Entradas Batch en tablas de servidor SQL con el objeto <code>SqlBulkCopy</code>	46
2.2.2	Obtener las estadísticas de conexión del servidor SQL	50
2.3	Ejecutar comandos SQL de forma asincrónica	51
2.3.1	El modelo Polling	52
2.3.2	El modelo Callback	54
2.3.3	El modelo WaitAll	57
2.3.4	Crear tablas de datos independientes	60
2.4	Utilizar tipos Nullable que soporten valores DBNull	65
2.5	Utilizar objetos persistentes de formulario Windows de ADO.NET 2.0	68
2.5.1	Comparando los diseñadores de datos de ADO.NET 1.x y 2.0	69
2.5.2	ADO.NET 1.x	70
2.5.3	ADO.NET 2.0	70
2.6	Añadir los controles ADO.NET que faltan	71
2.7	Actualizar proyectos de 1.x con componentes de ADO.NET 2.0	72
2.7.1	Añadir subformularios multinivel	72
2.8	Diseñar y mostrar informes con el control ReportViewer	74

Capítulo 3

	Concretando proyectos reales	77
3.1	Establecer la arquitectura	78
3.2	Las arquitecturas referenciales	79
3.2.1	Windows Server System Reference Architecture	79
3.2.2	Designando aplicaciones y servicios	80
3.2.3	Arquitectura referencial para el desarrollo empresarial	80
3.3	Encontrar modelos para proyectos	80
3.3.1	Enterprise Solution Patterns Using Microsoft .NET	81
3.3.2	Data Patterns	82
3.3.3	Modelos de sistemas distribuidos	82
3.3.4	Modelos de integración	82
3.3.5	Utilizar librerías de bloques de aplicaciones	83
3.4	El bloque de aplicación Data Access (Data Access Application Block)	84
3.4.1	El archivo de configuración de datos	85
3.4.2	Código de restablecimiento de datos	86
3.4.3	Código de actualización de datos	87
3.5	El cliente DataAccessQuickStart	89

Índice general

3.6	Seguir las guías de diseño	91
3.6.1	La guía .NET Data Access Architecture Guide	91
3.6.2	Mejorando el rendimiento y la escalabilidad de la aplicación .NET	91
3.6.3	Diseñar componentes Data Tier y pasar datos por tier	92
3.7	Aplicar las directrices de diseño para la biblioteca de clas (Apply Class Library Design Guidelines)	97
3.7.1	Naming Guidelines	97
3.7.2	Class Member Usage Guidelines	98
3.8	Prepararse para la arquitectura orientada al servicio	98
3.8.1	El camino a la Arquitectura orientada al servicio (SOA)	99
3.8.2	Implementar SOA con servicios Web	100
3.8.3	Garantizar total interoperabilidad del servicio Web	101
3.8.4	Instalar y publicar el servicio Web DataSetWS	103
3.9	Use FxCop para validar el código del proyecto	105
3.10	Automatizar Test-Driven Development	106
3.11	Ejecutar Best Practices Analyzer para SQL Server 2000 ...	106
3.12	Aplicar Best Practices específicas a los proyectos de ADO.NET 2.0	107
3.12.1	Use cadenas de conexión idénticas para las conexiones de bases de datos Pool	107
3.12.2	Definir el tamaño del pool de conexión	108
3.12.3	Guardar cadenas de conexión en archivos de configuración	109
3.12.4	Encriptar cadenas de conexión que contienen nombres de usuario y contraseñas	109
3.12.5	Ejecutar el SQL Server Profiler para inspeccionar las consultas SQL y RPC	109
3.12.6	Evitar añadir instancias CommandBuilder en tiempo de ejecución	110
3.12.7	Sustituir las consultas SQL Batch por procedimientos almacenados	110
3.12.8	Definir valores por defecto en los parámetros que no son necesarios	112
3.12.9	Utilizar sp_executesql y parámetros con nombre para reutilizar los Cached Query Plans	113
3.12.10	Añadir columnas timestamp para el control de concurrencia	114
3.12.11	Verificar registros en test de concurrencia	115
3.12.12	Evitar SqlExceptions con las validaciones del cliente	115
Capítulo 4		
Programar TableAdapters, BindingSources y DataGridViews		117
4.1	Diseñar un formulario básico Customer-Orders-Order Details	117

Bases de datos con Visual Basic

- 4.1.1 Reducir el tamaño del DataSet 118
- 4.1.2 Crear el origen de datos y añadir los controles 119
- 4.1.3 Añadir métodos FillBy para cada tabla de datos 120
- 4.1.4 Modificar el código autogenerado para llenar los controles 122
- 4.1.5 Llenar el cuadro combinado con valores CustomerID 123
- 4.1.6 Limpiar la UI y el código 124
- 4.2 Añadir y dar formato a DataGridView 125**
 - 4.2.1 Dar formato a las columnas OrdersDataGridView 125
 - 4.2.2 Añadir y dar formato a una columna calculada en Order_DetailsDataGridView 127
 - 4.2.3 Añadir la columna Extended amount 127
 - 4.2.4 Calcular y mostrar el valor Extended 127
- 4.3 Proporcionar valores por defecto a los nuevo records ... 129**
 - 4.3.2 Añadir valores por defecto en los registros de Order Details 130
- 4.4 Manejar el evento DataErrors 131**
- 4.5 Entrada de datos Streamline Heads-Down 132**
- 4.6 Migrar el UI a un formulario tabular 134**
 - 4.6.1 Comprobar el proyecto OrdersByCustomersV3 135
 - 4.6.2 Fijar los valores por defecto que faltan al añadir filas con código . 136
 - 4.6.3 Editar un record DataGridView seleccionado en la segunda ficha . 137
- 4.7 Crear y vincular listas de consulta Lookup para valores de clave primaria 139**
 - 4.7.1 Crear un juego de datos lookup no tipificado y sus tablas de datos 139
 - 4.7.2 Rellenar el cuadro combinado cboCustomerID 141
 - 4.7.3 Sustituir los cuadros de texto de DataGridView por cuadros combinados 141
 - 4.7.4 Añadir código para poblar los cuadros combinados Employees y ShipVia 142
 - 4.7.5 Reemplazar los valores nulos por defecto en las filas nuevas 143
 - 4.7.6 Asociar cuadros combinados con cuadros de texto 144
- 4.8 Añadir un cuadro combinado que defina valores adicionales 146**
 - 4.8.1 Crear y vincular un DataView ordenado por ProductName 147
 - 4.8.2 Comprobar que no haya duplicados y actualizar la columna UnitPrice 148
- 4.9 Añadir filas a las tablas lookup para entradas de nuevos Customers 150**
 - 4.9.1 Añadir y vincular una BindingSource CustomerID 151
 - 4.9.2 Comprobar la existencia de duplicados con un DataRowView 152
- 4.10 Aplicar reglas de negocio a las ediciones 153**

Índice general

4.11 Guardar los cambios en las tablas base 154

4.11.1 Mantener la integridad referencial 154

4.11.2 Crear y comprobar la función UpdateBaseTables 155

4.11.3 Entender la generación de cambios en tablas y las instrucciones para la actualización de las tablas base 156

4.11.4 Añadir la función UpdateBaseTables 158

4.11.5 Operaciones previas de actualización 162

4.11.6 Invocar la función UpdateBaseTables 162

4.11.7 Comprobar los valores CustomerID del servidor para evitar duplicados 163

Capítulo 5

Añadir código para validar datos y gestionar la concurrencia 165

5.1 Validar las entradas de datos 167

5.1.1 Validar cuadros de texto 167

5.1.2 Validar controles DataGridViews 168

5.1.3 Capturar las violaciones de restricción de clave primera durante la entrada 170

5.1.4 Validar valores por defecto 171

5.2 Gestionar las transgresiones de concurrencia 173

5.2.1 Control de concurrencia y cambios de transacción en ADO.NET 2.0 174

5.2.2 Propiedades ocultas de conexión y transacción 175

5.2.3 La propiedad ContinueUpdateOnError 176

5.2.4 Estrategias de control de concurrencia 176

5.2.5 Los "vínculos perdidos" en la gestión de la concurrencia 177

5.2.6 Detectar los fallos de concurrencia en los registros hijo 178

5.2.7 Detectar otros conflictos potenciales de concurrencia 179

5.2.8 Permitir a los usuarios re-crear los pedidos borrados 181

5.3 Anticipar las transgresiones de restricción de clave primaria basada en valores 184

5.4 Manejar elegantemente los errores de concurrencia 187

5.4.1 Obtener datos actuales del servidor 188

5.4.2 Restablecer y comparar los valores de celda del servidor y el cliente 190

5.5 Trabajar con usuarios desconectados 193

5.5.1 Crear y gestionar juegos de datos offline 194

5.5.2 Activar el tratamiento de registros padre múltiples 195

Capítulo 6

La aplicación de técnicas avanzadas de los DataSets 199

6.1 Aplicar transacciones a las actualizaciones de DataSets 200

Bases de datos con Visual Basic

6.1.1	Simplificar el listado con System.Transactions	202
6.1.2	Listar SqlDataAdapter en una transacción implícita	203
6.1.3	Autolistar SqlDataAdapter en una transacción implícita	204
6.1.4	SQL Profiler para rastrear transacciones	205
6.1.5	Listar manualmente SqlDataAdapter en una transacción explícita	206
6.1.6	Definir las opciones TransactionScope y Transaction	207
6.2	Añadir relaciones a los SelectCommand de la tabla de datos	208
6.2.1	Añadir una relación a SelectCommand	209
6.2.2	Añadir las columnas adjuntadas con relaciones al DataGridView ..	210
6.2.3	Proporcionar los valores por defecto y columnas de sólo lectura ..	210
6.3	Mejorar el rendimiento reduciendo el tamaño de los juegos de datos	212
6.3.1	Limitar el número de filas devueltas por las consultas TOP n	213
6.3.2	Añadir clases Partial para TableAdapters	214
6.4	Trabajar con imágenes en DataGridViews	215
6.4.1	Añadir columnas Image a los DataGridViews	215
6.4.2	Manipular imágenes en DataGridView	216
6.4.3	Cambiar ImageLayout	216
6.4.4	Guardar una imagen seleccionada, mostrarla en un PictureBox y reemplazarla	218
6.4.5	Evitar crear imágenes desde los campos de objeto OLE en Access	220
6.5	Editar documentos XML con DataSets yDataGridViews ...	220
6.5.1	Adaptar un esquema XML existente para generar un DataSet	221
6.5.2	Esquemas para documentos XML de jerarquía anidada	222
6.5.3	Un ejemplo de esquema anidado	224
6.5.4	La ventana Propiedades de las columnas	227
6.5.5	Un esquema anidado con atributos	227
6.5.6	Ejemplo de esquema anidado y "envuelto" (wrapped)	228
6.5.7	Un ejemplo de esquema plano	230
6.5.8	Inferir un esquema XML para generar un juego de datos	233
6.5.9	Crear formularios de edición desde fuentes de datos XML	235
6.5.10	El proyecto de ejemplo EditNorthwindDS	235

Capítulo 7

Trabajar con las fuentes de datos y controles vinculados de ASP.NET 2.0 **239**

7.1	Las nuevas características de ASP.NET 2.0	240
7.1.1	El modelo de compilación de ASP.NET	242
7.1.2	Los nuevos controles (Data Controls) de ASP.NET 2.0	244
7.2	Los controles DataSource	245
7.3	El control DataList	247
7.3.1	SqlDataSources para controles vinculados	247

Índice general

7.3.2	Propiedades de control	251
7.3.3	Plantilla de datos vinculados y formateo de datos	252
7.3.4	Restricciones WHERE en el código fuente en los valores de controles vinculados	254
7.3.5	Editar ítems en listas de datos	257
7.4	El control FormView	259
7.4.1	Paginar la fuente de datos	259
7.4.2	Reemplazar los valores Null por texto específico de la columna	260
7.4.3	Editar, añadir y borrar registros	262
7.4.4	Añadir botones de comando	263
7.5	El control GridView	264
7.5.1	Convertir campos BoundFields en campos EditItemTemplate	266
7.5.2	Reemplazar cuadros de texto por listas desplegadas para la edición	267
Capítulo 8		
Aplicar técnicas avanzadas con ASP.NET 2.0		271
8.1	Validar entradas en controles vinculados a datos	271
8.2	Los controles de validación de ASP.NET 2.0	272
8.2.1	La nueva propiedad ValidationGroup	273
8.3	Otras propiedades de validación compartidas	273
8.4	Validar ediciones en GridView	274
8.4.1	Añadir un campo necesario de validación a un control GridView	275
8.5	Validar entradas CustomerID con un control RegularExpressionValidator	276
8.5.1	Comprobar los valores de EmployeeID con un control RangeValidator	278
8.5.2	Aplicar RangeValidator y RegularExpressionValidator a las entradas de datos	278
8.6	Impedir entradas ilógicas con un CompareValidator	279
8.6.1	Añadir un control CustomValidator	280
8.7	Escribir un mensaje para el control Validation Summary	283
8.8	Validar ediciones de ProductID en el servidor Web	284
8.8.1	Test para descubrir valores duplicados de ProductID en el cliente	284
8.9	Reemplazar SqlDataSources por ObjectDataSources	285
8.9.1	ObjectDataSources a partir de DataTables	286
8.9.2	Crear y asignar ObjectDataSources de un DataSet	287