Session: 420219
Agenda Key: 44CA

eserver iSeries

iSeries Access in the .NET World

Brent Nelson - bmnelson@us.ibm.com
iSeries Access Development

ON DEMAND BUSINESS™

Agenda

- Overview of .NET
- ADO.NET
- iSeries Access in .NET Environment
- iSeries Access .NET Provider Class Notes
- Example programs

What is .NET?

Programming model for building XML Web services and applications

.NET definitions

.NET framework

- Underlying plumbing for .NET applications
- Common Language Runtime (CLR)
- Unified set of class libraries

CLR

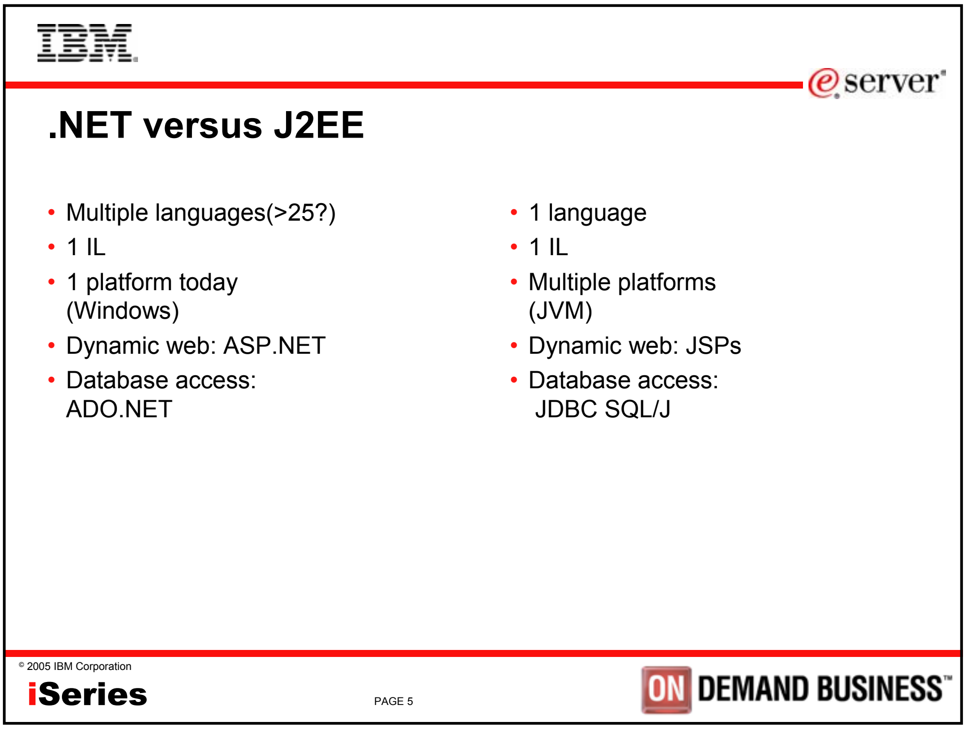
- Language integration, security handling, memory/ thread/process management, exception handling, …

ADO.NET

- .NET classes enabling access to databases

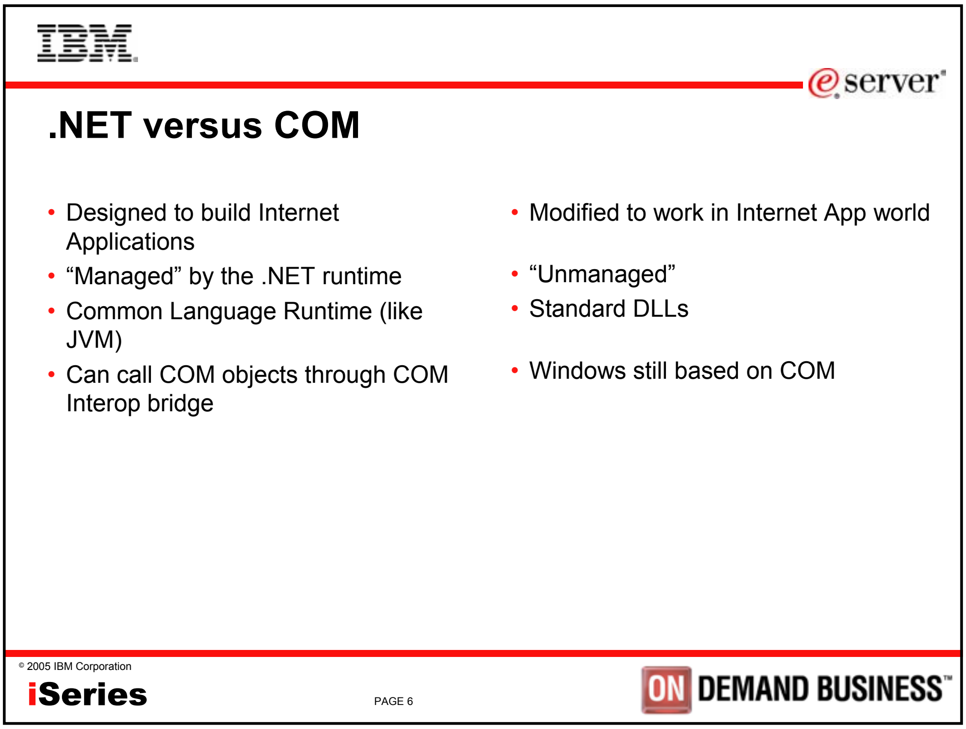
ASP.NET

- .NET classes to support development of Web-based applications and Web services



.NET versus J2EE

• Multiple languages(>25?)	• 1 language
• 1 IL	• 1 IL
• 1 platform today (Windows)	• Multiple platforms (JVM)
• Dynamic web: ASP.NET	• Dynamic web: JSPs
• Database access: ADO.NET	• Database access: JDBC SQL/J



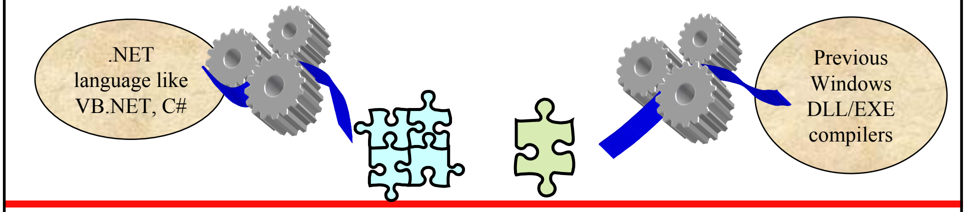
.NET versus COM

• Designed to build Internet Applications	• Modified to work in Internet App world
• “Managed” by the .NET runtime	• “Unmanaged”
• Common Language Runtime (like JVM)	• Standard DLLs
• Can call COM objects through COM Interop bridge	• Windows still based on COM

Assemblies versus Binaries

- Intermediate Language (IL) JIT compile
- Self describing metadata
- Referenced by namespace, version, and culture

- Platform specific byte codes
- Type libraries (IDL)
- Referenced by Registry



.NET language like VB.NET, C#

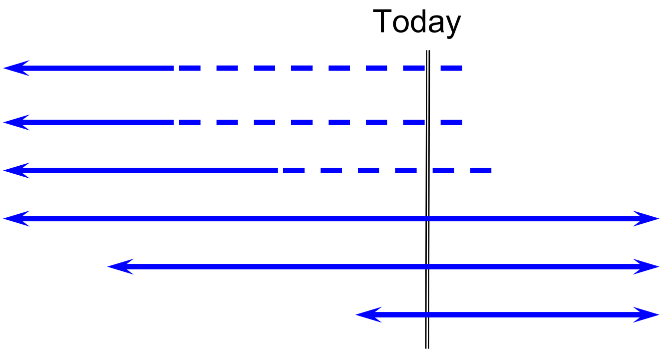
Previous Windows DLL/EXE compilers

Interesting Anecdotes

- Fixing “DLL hell” by allowing multiple versions
- Garbage collection
- Common Type System (CTS) but all the languages do not support all .NET data types
- COM still alive but calling it from .NET means bad performance

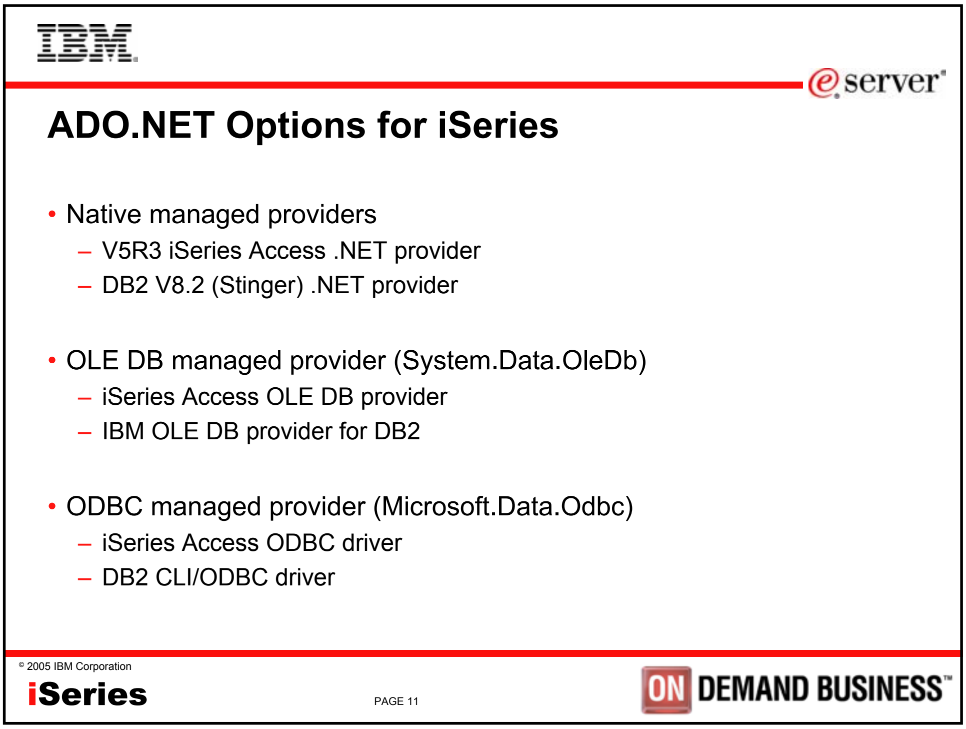
Microsoft Technology Roadmap

- DB-Library
- ESQL
- DAO
- ODBC
- ADO/OLE DB
- ADO.NET



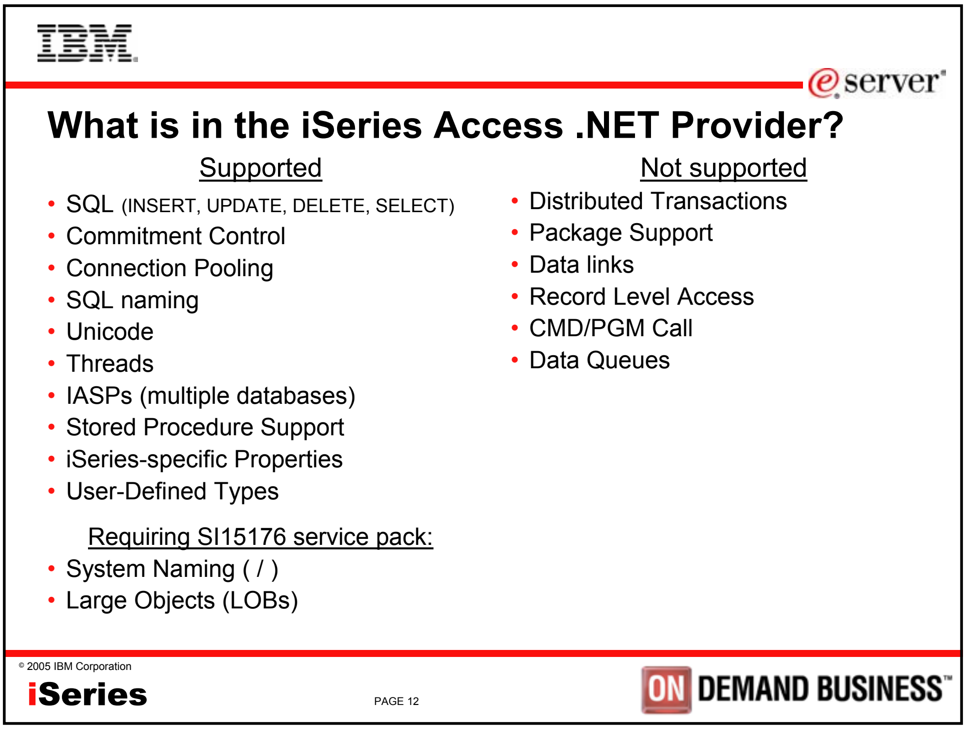
ADO Options for iSeries

- OLE DB providers
  - iSeries Access OLE DB Provider (IBMDA400, IBMDASQL, IBMDARLA)
  - IBM OLE DB Provider for DB2 (IBMDADB2)
- OLE DB provider for ODBC (MSDASQL)
  - iSeries Access ODBC driver
  - DB2 CLI/ODBC driver



ADO.NET Options for iSeries

- Native managed providers
  - V5R3 iSeries Access .NET provider
  - DB2 V8.2 (Stinger) .NET provider
- OLE DB managed provider (System.Data.OleDb)
  - iSeries Access OLE DB provider
  - IBM OLE DB provider for DB2
- ODBC managed provider (Microsoft.Data.Odbc)
  - iSeries Access ODBC driver
  - DB2 CLI/ODBC driver



What is in the iSeries Access .NET Provider?

Supported

- SQL (INSERT, UPDATE, DELETE, SELECT)
- Commitment Control
- Connection Pooling
- SQL naming
- Unicode
- Threads
- IASPs (multiple databases)
- Stored Procedure Support
- iSeries-specific Properties
- User-Defined Types

Requiring SI15176 service pack:

- System Naming (/)
- Large Objects (LOBs)

Not supported

- Distributed Transactions
- Package Support
- Data links
- Record Level Access
- CMD/PGM Call
- Data Queues

What is in the DB2 V8.2 .NET Provider?

Supported

- SQL (INSERT, UPDATE, DELETE, SELECT)
- Commitment Control
- Connection Pooling
- SQL Naming
- Unicode
- Threads
- IASPs (multiple databases)
- Stored Procedure Support
- Large Objects (LOBs)
- Distributed Transactions
- Accessing Data on Other DB2 Boxes
- Visual Studio .NET Add-ins

Not supported

- System Naming (/)
- Package Support
- Data links
- User-Defined Types
- Record Level Access
- CMD/PGM Call
- Data Queues

For more information…

- V5R3 iSeries Access .NET provider
  - http://www-1.ibm.com/servers/eserver/iseries/access/
  - NET Technical Reference - cwbmptch.hlp
    - Found in ..\IBM\Client Access\Mri2924 directory

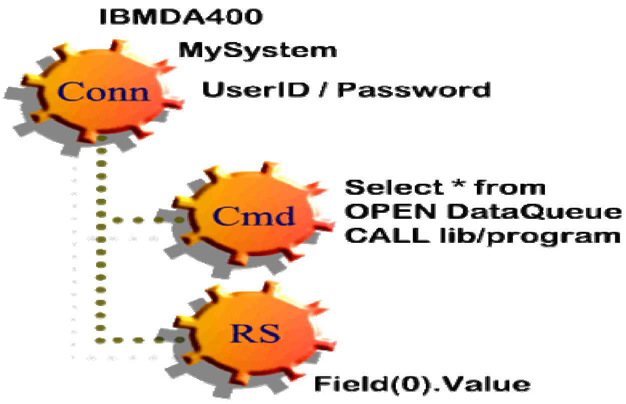
- DB2 V8.2 (Stinger) .NET provider
  - http://www-306.ibm.com/software/data/db2/udb/v82/
  - http://www7b.software.ibm.com/dmdd/downloads/dotnetbeta/

- Using iSeries Access through .NET
  - http://www-919.ibm.com/servers/eserver/iseries/developer/education/wp/db2_intnet/index.html
  - Info APAR II13341 Using IBMDA400:
    - http://www.ibm.com/servers/eserver/iseries/access/caiixe1.htm
    - Scroll to the OLE DB section look for “Using managed providers in the .NET framework for accessing DB2 data on AS/400 or iSeries servers”

For more information…

- See .NET Redbook coming out soon!
  - http://www.redbooks.ibm.com/
- MC Press Online Articles
  - ".NET Integration with DB2 UDB for iSeries"
    - http://www.mcpressonline.com/mc/.6b1993b7
  - "A Detailed Look at DB2 Stinger .NET CLR Routines"
    - http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0406evans/index.html

ADO



IBMDA400

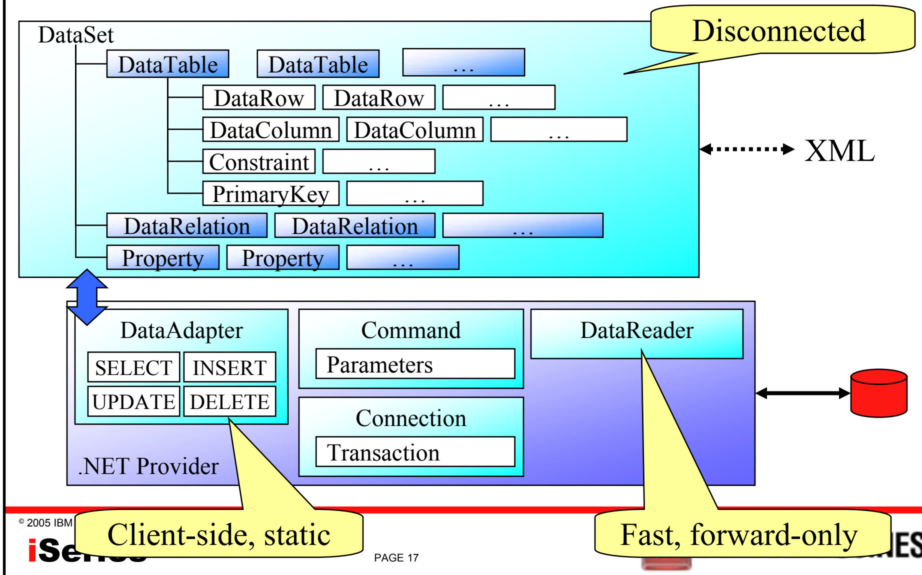
MySystem

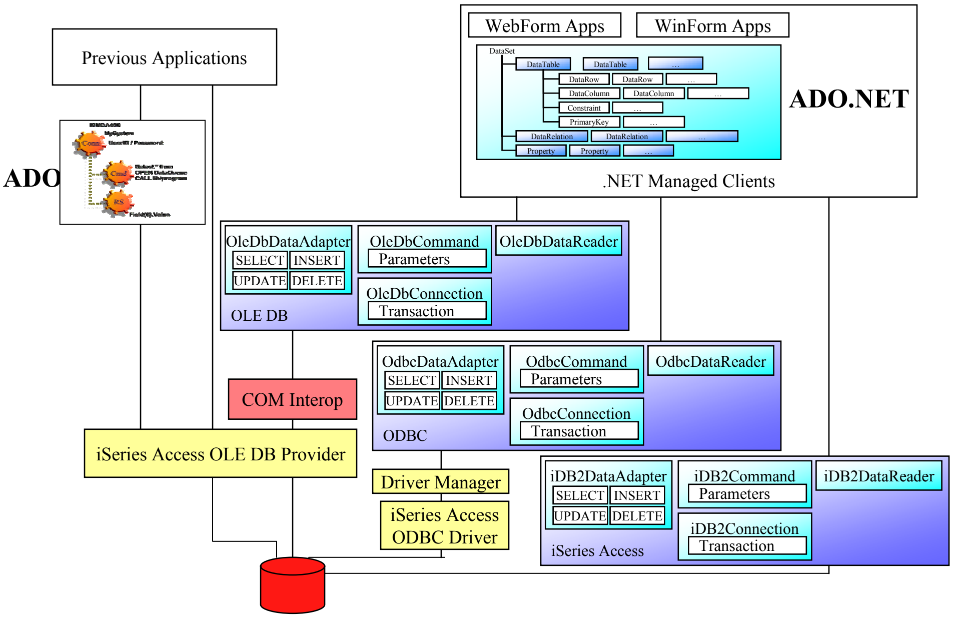
UserID / Password

Select * from
OPEN DataQueue
CALL lib/program

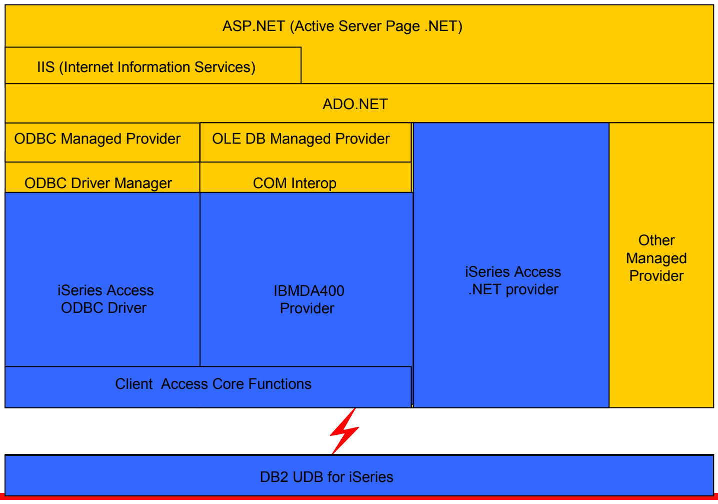
Field(0).Value

ADO.NET


DataSet
DataTable
DataTable
…
DataRow
DataRow
…
DataColumn
DataColumn
…
Constraint
…
PrimaryKey
…
DataRelation
DataRelation
…
Property
Property
…
Disconnected
XML
DataAdapter
SELECT
INSERT
UPDATE
DELETE
Command
Parameters
Connection
Transaction
DataReader
.NET Provider
Client-side, static
Fast, forward-only


Previous Applications
ADO
IBMDA400
MySystem
Conn
UserID / Password
Cmd
Select * from
OPEN DataQueue
CALL Mdprogram
RS
Field(0).Value
COM Interop
iSeries Access OLE DB Provider
WebForm Apps
WinForm Apps
DataSet
DataTable
DataTable
…
DataRow
DataRow
…
DataColumn
DataColumn
…
Constraint
…
PrimaryKey
…
DataRelation
DataRelation
…
Property
Property
…
ADO.NET
.NET Managed Clients
OleDbDataAdapter
SELECT
INSERT
UPDATE
DELETE
OleDbCommand
Parameters
OleDbDataReader
OleDbConnection
Transaction
OLE DB
OdbcDataAdapter
SELECT
INSERT
UPDATE
DELETE
OdbcCommand
Parameters
OdbcDataReader
OdbcConnection
Transaction
ODBC
Driver Manager
iSeries Access ODBC Driver
iDB2DataAdapter
SELECT
INSERT
UPDATE
DELETE
iDB2Command
Parameters
iDB2DataReader
iDB2Connection
Transaction
iSeries Access

iSeries Access in 3-Tier Environment



ASP.NET (Active Server Page .NET)

IIS (Internet Information Services)

ADO.NET

ODBC Managed Provider

OLE DB Managed Provider

ODBC Driver Manager

COM Interop

iSeries Access ODBC Driver

IBMDA400 Provider

iSeries Access .NET provider

Other Managed Provider

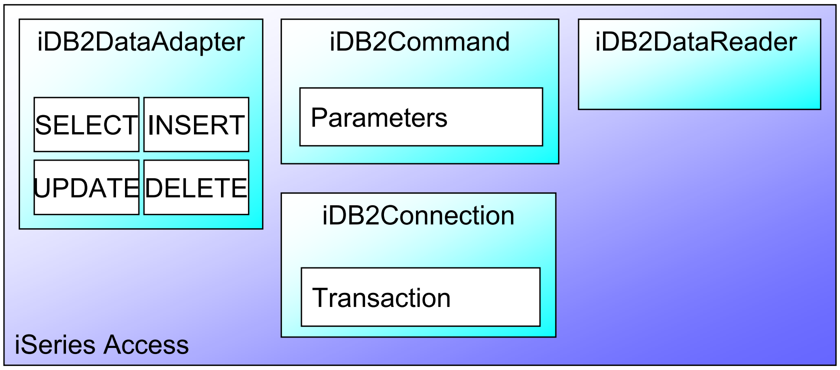
Client Access Core Functions

DB2 UDB for iSeries

iSeries Access .NET Provider Notes

- Install requires the .NET framework be on PC
  - Windows Server 2003 installs .NET framework by default
- Same requirements as iSeries Access OLE DB / ODBC to use
- Some support limited on pre-V5R2 servers

iSeries Access .NET Provider - Class Notes



Connections

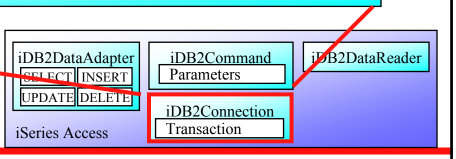
iDB2Connection

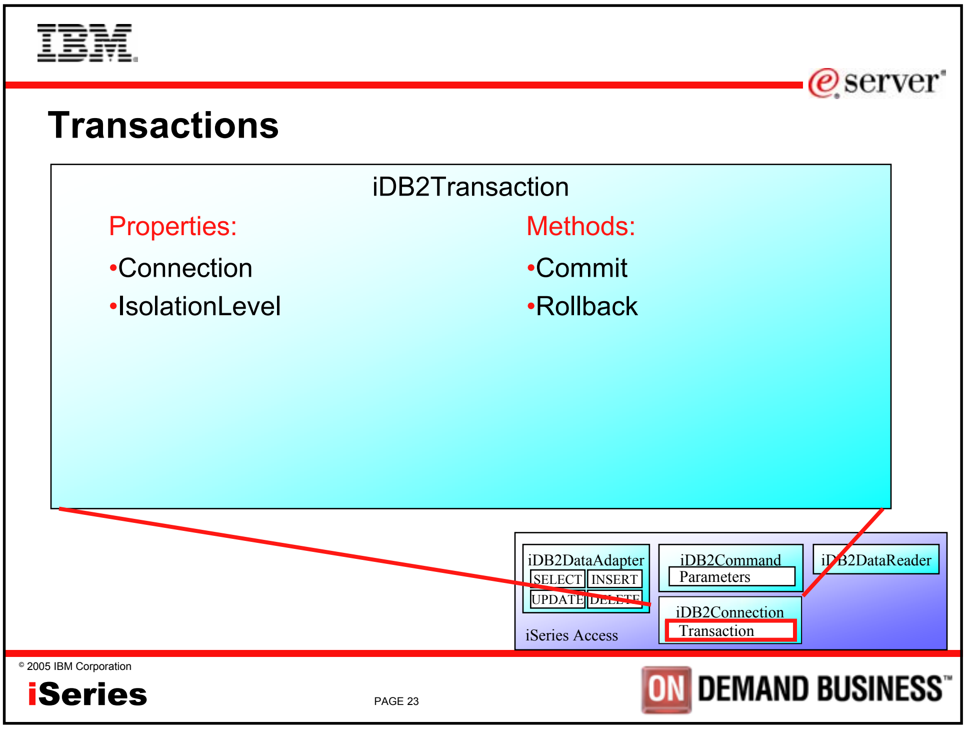
Properties:

- ConnectionString

Methods:

- BeginTransaction
- CreateCommand
- Open
- Close





Transactions

iDB2Transaction

Properties:

- Connection
- IsolationLevel

Methods:

- Commit
- Rollback

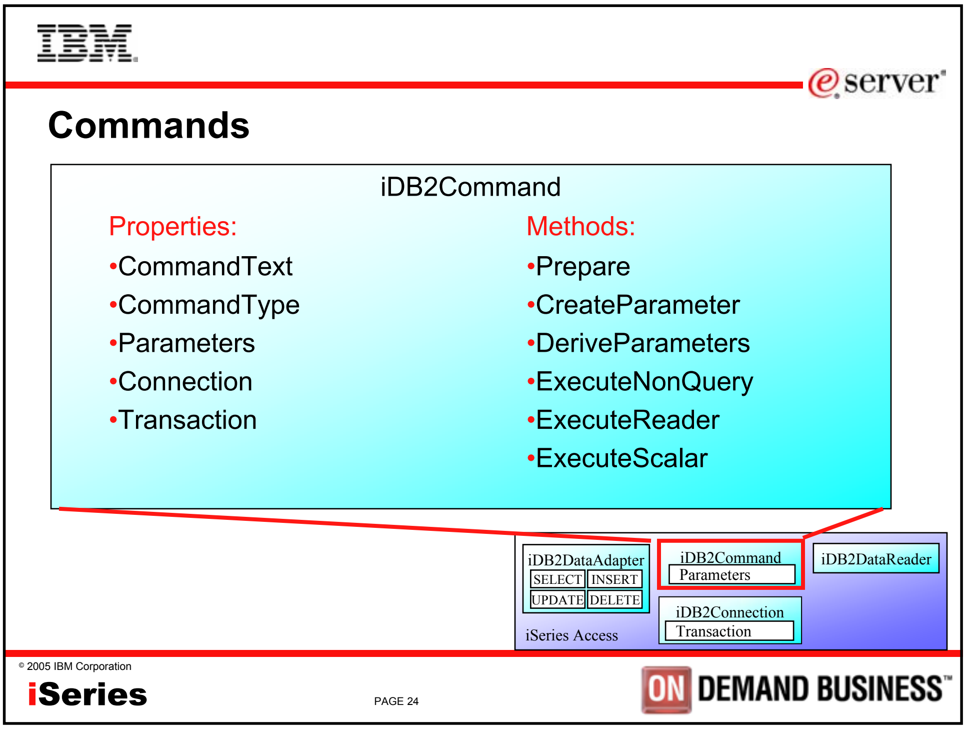
iDB2DataAdapter
SELECT INSERT
UPDATE DELETE

iDB2Command
Parameters

iDB2DataReader

iDB2Connection
Transaction

iSeries Access



Commands

iDB2Command

Properties:

- CommandText
- CommandType
- Parameters
- Connection
- Transaction

Methods:

- Prepare
- CreateParameter
- DeriveParameters
- ExecuteNonQuery
- ExecuteReader
- ExecuteScalar

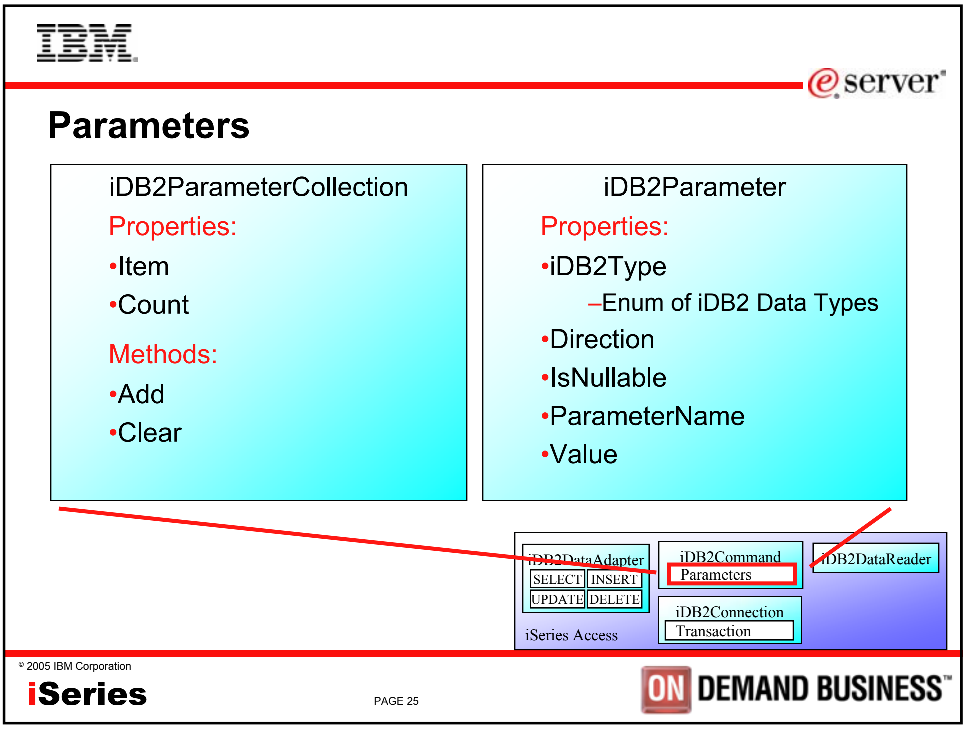
iDB2DataAdapter
SELECT INSERT
UPDATE DELETE

iDB2Command
Parameters

iDB2DataReader

iDB2Connection
Transaction

iSeries Access



Parameters

iDB2ParameterCollection

Properties:

- Item
- Count

Methods:

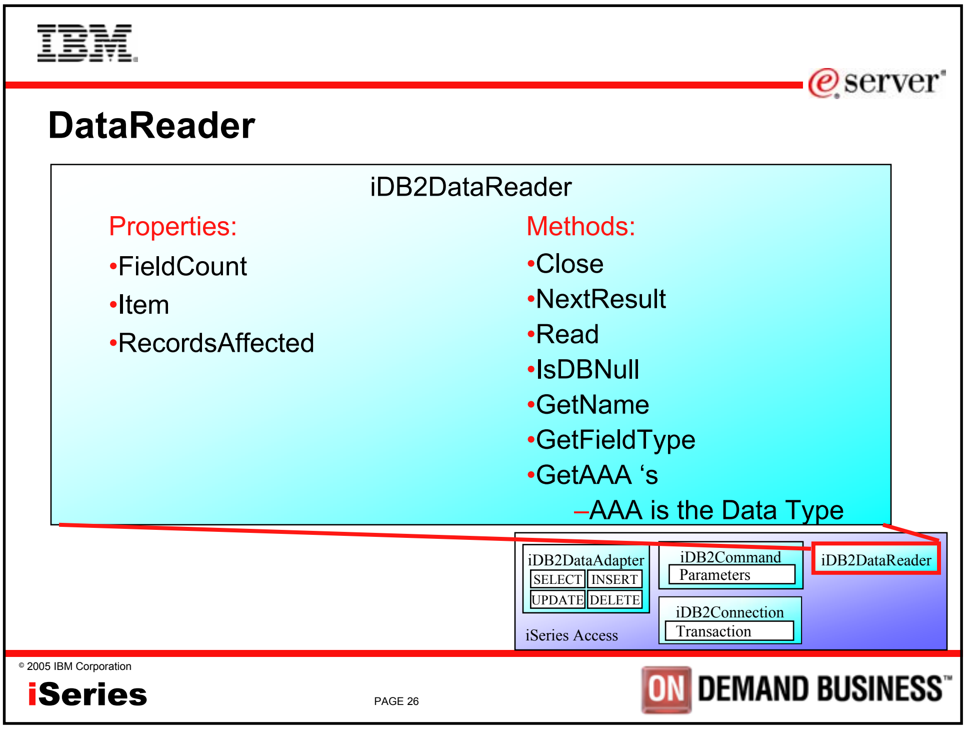
- Add
- Clear

iDB2Parameter

Properties:

- iDB2Type
  - Enum of iDB2 Data Types
- Direction
- IsNullable
- ParameterName
- Value

iDB2DataAdapter
SELECT
INSERT
UPDATE
DELETE
iDB2Command
Parameters
iDB2DataReader
iDB2Connection
Transaction
iSeries Access



DataReader

iDB2DataReader

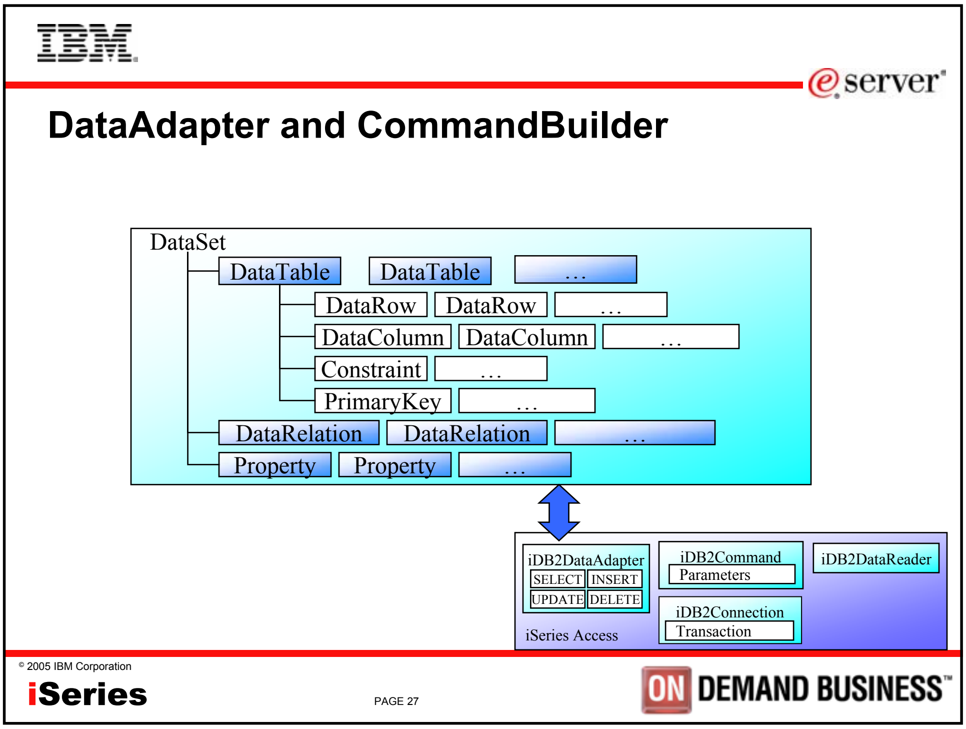
Properties:

- FieldCount
- Item
- RecordsAffected

Methods:

- Close
- NextResult
- Read
- IsDBNull
- GetName
- GetFieldType
- GetAAA ‘s
  - AAA is the Data Type

iDB2DataAdapter
SELECT
INSERT
UPDATE
DELETE
iDB2Command
Parameters
iDB2DataReader
iDB2Connection
Transaction
iSeries Access



DataAdapter and CommandBuilder

DataSet

DataTable | DataTable | ...

DataRow | DataRow | ...

DataColumn | DataColumn | ...

Constraint | ...

PrimaryKey | ...

DataRelation | DataRelation | ...

Property | Property | ...

iDB2DataAdapter

SELECT | INSERT

UPDATE | DELETE

iDB2Command

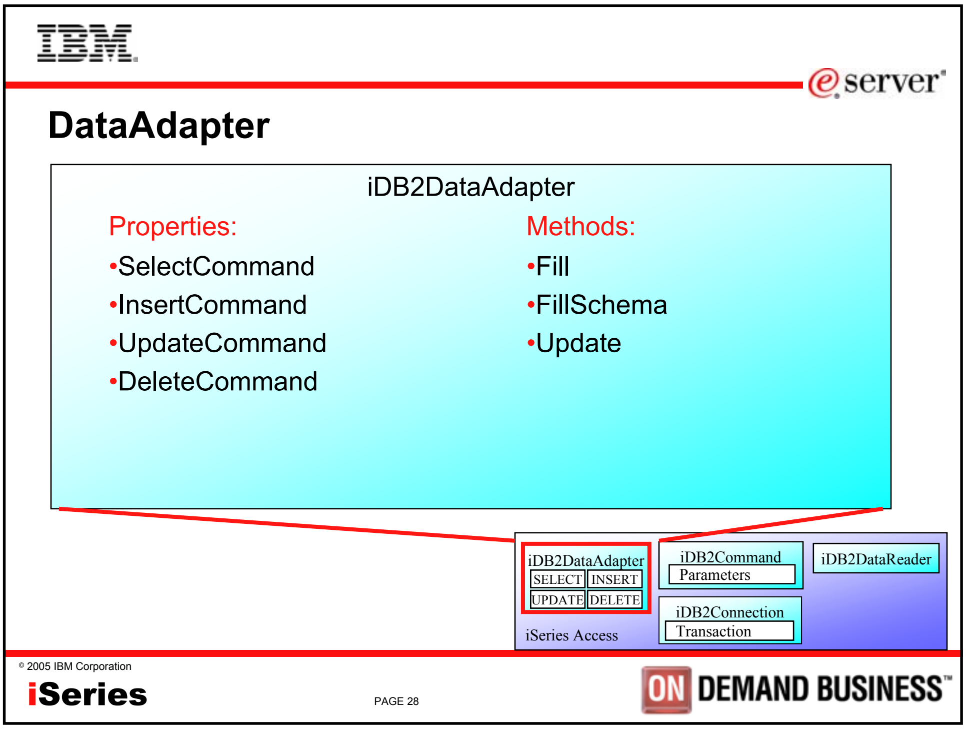
Parameters

iDB2DataReader

iDB2Connection

Transaction

iSeries Access



DataAdapter

iDB2DataAdapter

Properties:

- SelectCommand
- InsertCommand
- UpdateCommand
- DeleteCommand

Methods:

- Fill
- FillSchema
- Update

iDB2DataAdapter

SELECT | INSERT

UPDATE | DELETE

iDB2Command

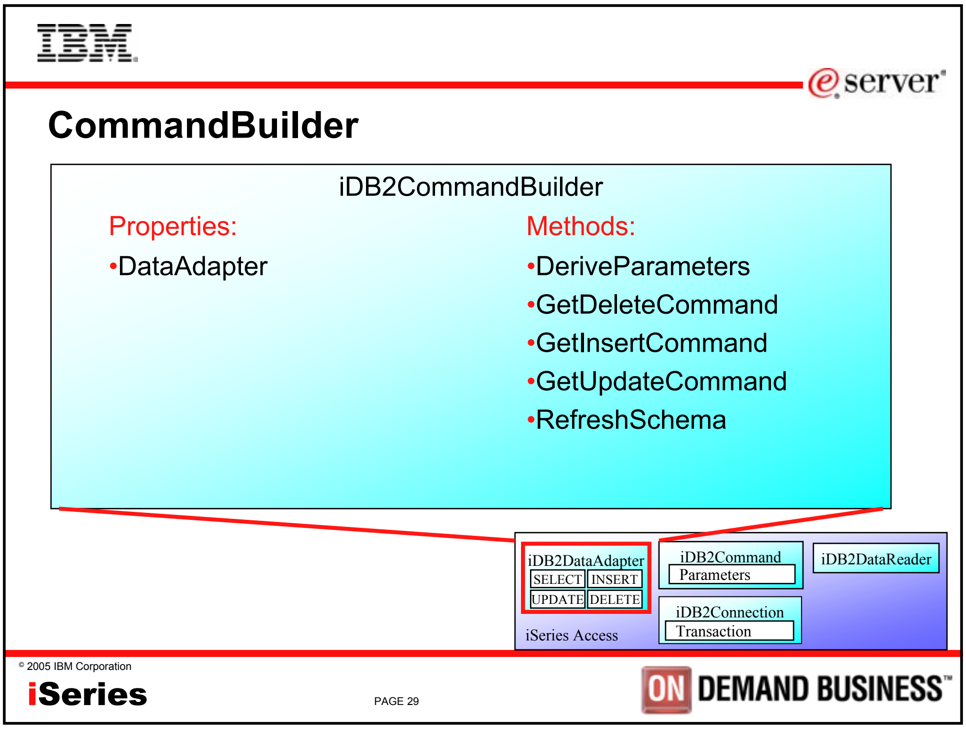
Parameters

iDB2DataReader

iDB2Connection

Transaction

iSeries Access



CommandBuilder

iDB2CommandBuilder

Properties:

- DataAdapter

Methods:

- DeriveParameters
- GetDeleteCommand
- GetInsertCommand
- GetUpdateCommand
- RefreshSchema

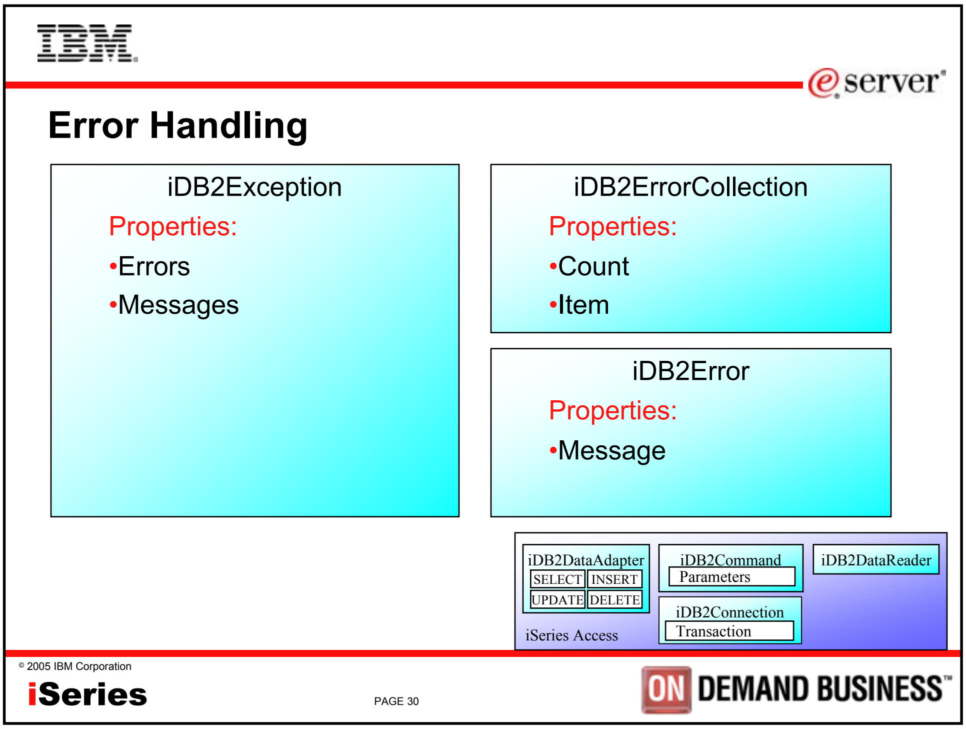
iDB2DataAdapter
SELECT INSERT
UPDATE DELETE

iDB2Command
Parameters

iDB2DataReader

iDB2Connection
Transaction

iSeries Access



Error Handling

iDB2Exception

Properties:

- Errors
- Messages

iDB2ErrorCollection

Properties:

- Count
- Item

iDB2Error

Properties:

- Message

iDB2DataAdapter
SELECT INSERT
UPDATE DELETE

iDB2Command
Parameters

iDB2DataReader

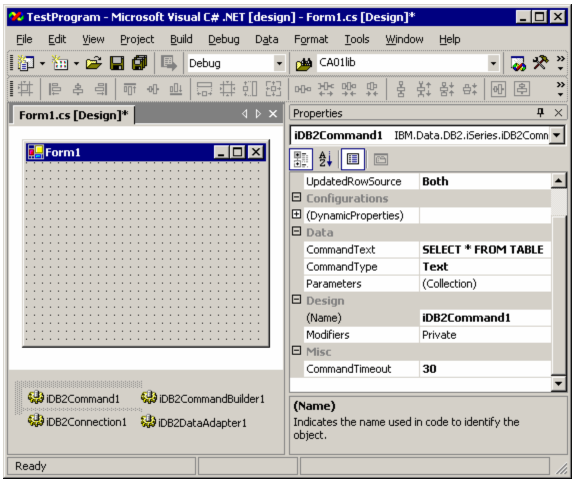
iDB2Connection
Transaction

iSeries Access

Coding Tips

- Use Parameters instead of literal values in SQL statements
  - INSERT INTO MYTABLE VALUES (?)
  - INSERT INTO MYTABLE VALUES (‘abcde’)
- Build SQL queries to retrieve only the data you need
- Reuse Command objects with the same CommandText
- With LOBs, use a DataReader or the Command’s ExecuteScalar method instead of a DataAdapter
- Call an object’s Close or Dispose method when finished with it
- Make sure to handle Exceptions

Visual Studio .NET Integration



iDB2Command1 IBM.Data.DB2.iSeries.iDB2Comm

UpdatedRowSource: Both

Configurations

(DynamicProperties)

Data

CommandText: SELECT * FROM TABLE

CommandType: Text

Parameters: (Collection)

Design

(Name): iDB2Command1

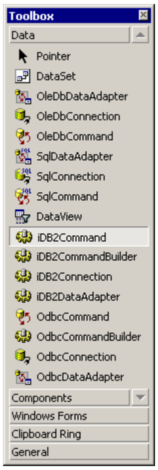
Modifiers: Private

Misc

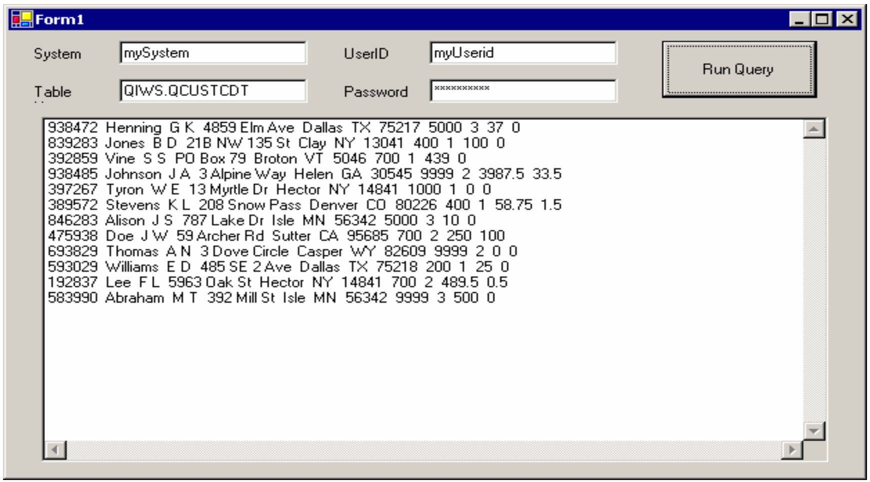
CommandTimeout: 30

(Name)
Indicates the name used in code to identify the object.

iDB2Command1 iDB2CommandBuilder1 iDB2Connection1 iDB2DataAdapter1



Example #1: Basic Class Usage



System mySystem

UserID myUserid

Table QIWS.QCUSTCDT

Password ***********

```
938472  Henning  G K  4859 Elm Ave  Dallas  TX  75217  5000  3  37  0
839283  Jones  B D  21B NW 135 St  Clay  NY  13041  400  1  100  0
392859  Vine  S S  PO Box 79  Broton  VT  5046  700  1  439  0
938485  Johnson  J A  3 Alpine Way  Helen  GA  30545  9999  2  3987.5  33.5
397267  Tyron  W E  13 Myrtle Dr  Hector  NY  14841  1000  1  0  0
389572  Stevens  K L  208 Snow Pass  Denver  CO  80226  400  1  58.75  1.5
846283  Alison  J S  787 Lake Dr  Isle  MN  56342  5000  3  10  0
475938  Doe  J W  59 Archer Rd  Sutter  CA  95685  700  2  250  100
693829  Thomas  A N  3 Dove Circle  Casper  WY  82609  9999  2  0  0
593029  Williams  E D  485 SE 2 Ave  Dallas  TX  75218  200  1  25  0
192837  Lee  F L  5963 Oak St  Hector  NY  14841  700  2  489.5  0.5
583990  Abraham  M T  392 Mill St  Isle  MN  56342  9999  3  500  0
```

Example #1: Basic Class Usage - Code

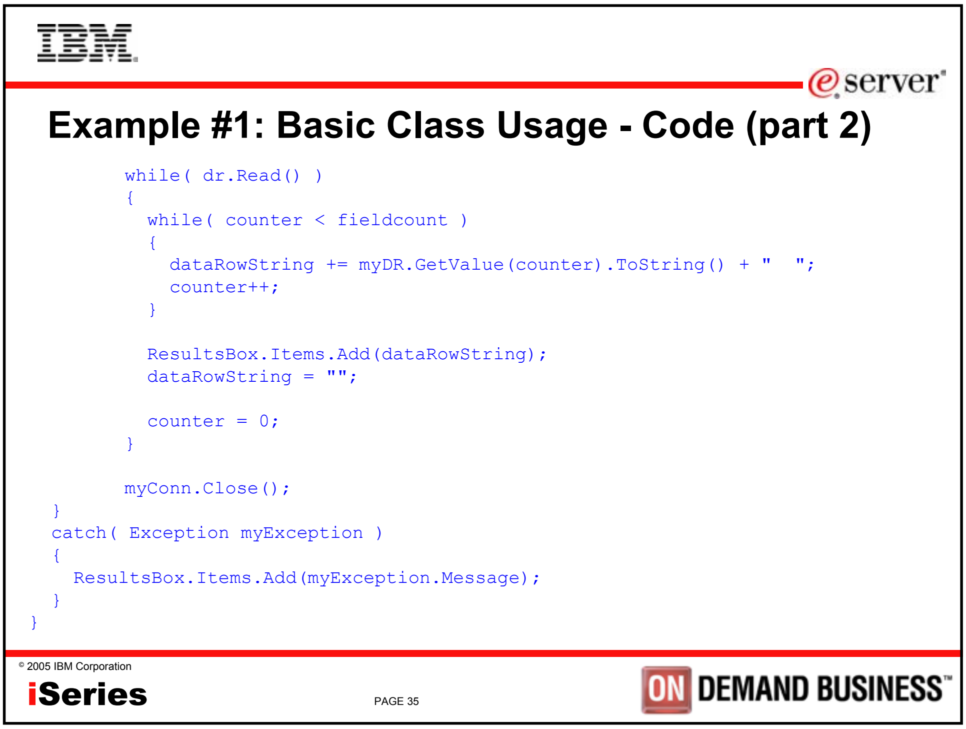
```
private void RunQueryButton_Click(object sender, System.EventArgs e)
{
  ResultsBox.Items.Clear();
  iDB2Connection myConn = new iDB2Connection();
  myConn.ConnectionString = "DataSource=" + SystemBox.Text +
                            ";UserID=" + UseridBox.Text +
                            ";Password=" + PasswordBox.Text;

  try
  {
    myConn.Open();

    iDB2Command myCmd = new iDB2Command(TableBox.Text,myConn);
    myCmd.CommandType = CommandType.TableDirect;
    iDB2DataReader myDR = myCmd.ExecuteReader();

    String dataRowString = "";

    int fieldcount = myDR.FieldCount, counter = 0;
```



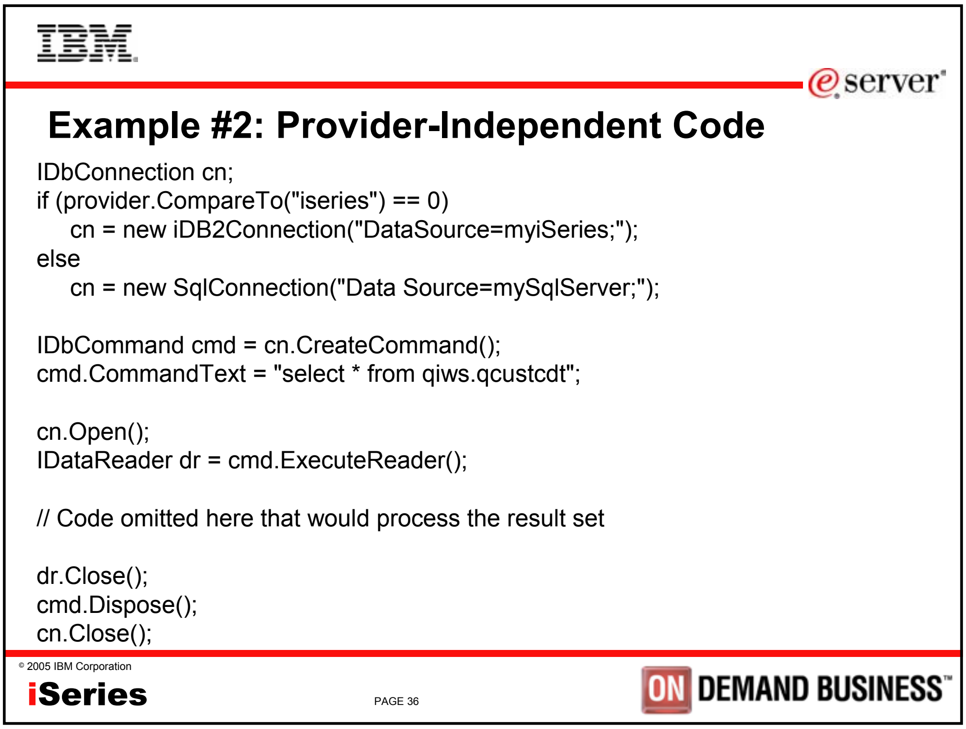
Example #1: Basic Class Usage - Code (part 2)

```
        while( dr.Read() )
        {
          while( counter < fieldcount )
          {
            dataRowString += myDR.GetValue(counter).ToString() + "  ";
            counter++;
          }

          ResultsBox.Items.Add(dataRowString);
          dataRowString = "";

          counter = 0;
        }

        myConn.Close();
  }
  catch( Exception myException )
  {
    ResultsBox.Items.Add(myException.Message);
  }
}
```



Example #2: Provider-Independent Code

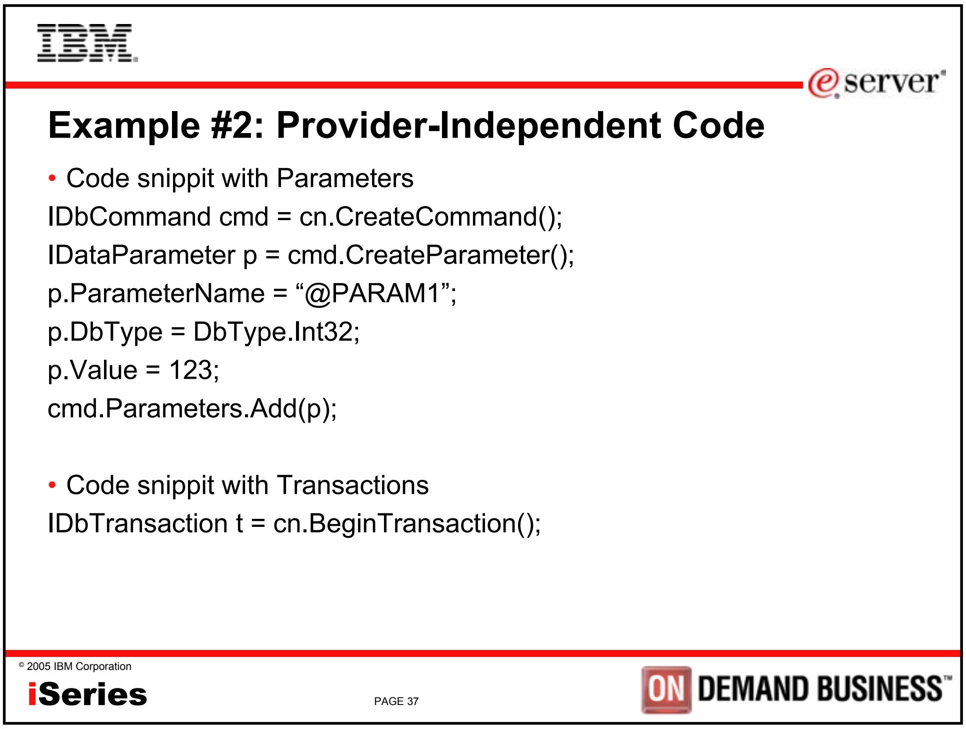
```
IDbConnection cn;
if (provider.CompareTo("iseries") == 0)
    cn = new iDB2Connection("DataSource=myiSeries;");
else
    cn = new SqlConnection("Data Source=mySqlServer;");

IDbCommand cmd = cn.CreateCommand();
cmd.CommandText = "select * from qiws.qcustcdt";

cn.Open();
IDataReader dr = cmd.ExecuteReader();

// Code omitted here that would process the result set

dr.Close();
cmd.Dispose();
cn.Close();
```



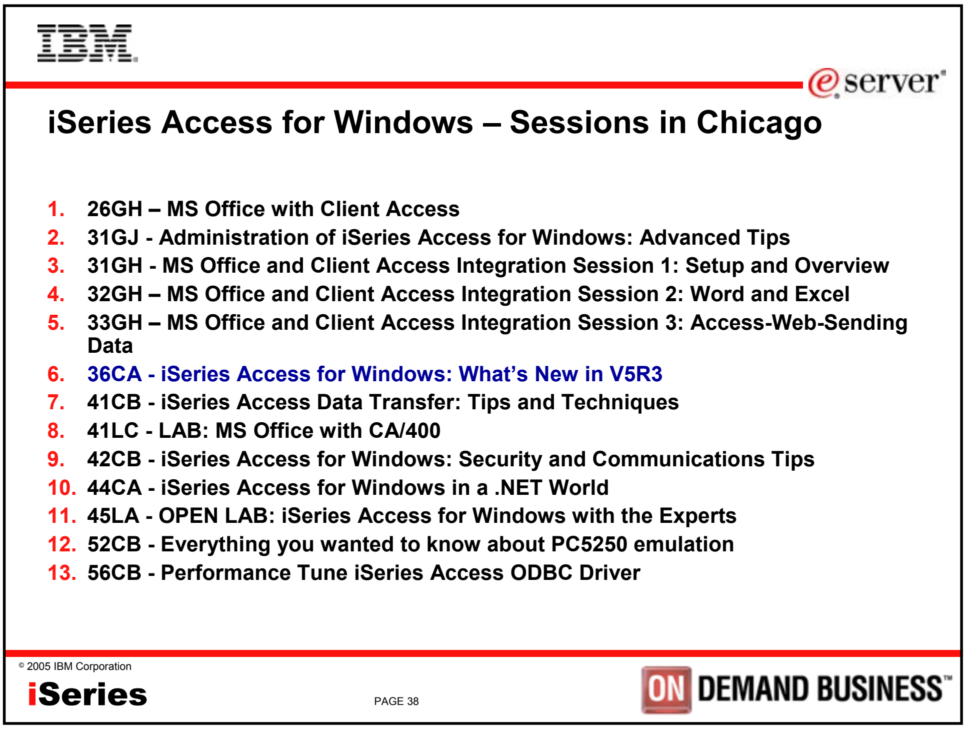
Example #2: Provider-Independent Code

- Code snippit with Parameters

```
IDbCommand cmd = cn.CreateCommand();
IDataParameter p = cmd.CreateParameter();
p.ParameterName = “@PARAM1”;
p.DbType = DbType.Int32;
p.Value = 123;
cmd.Parameters.Add(p);
```

- Code snippit with Transactions

```
IDbTransaction t = cn.BeginTransaction();
```



iSeries Access for Windows – Sessions in Chicago

1. **26GH – MS Office with Client Access**
2. **31GJ - Administration of iSeries Access for Windows: Advanced Tips**
3. **31GH - MS Office and Client Access Integration Session 1: Setup and Overview**
4. **32GH – MS Office and Client Access Integration Session 2: Word and Excel**
5. **33GH – MS Office and Client Access Integration Session 3: Access-Web-Sending Data**
6. **36CA - iSeries Access for Windows: What’s New in V5R3**
7. **41CB - iSeries Access Data Transfer: Tips and Techniques**
8. **41LC - LAB: MS Office with CA/400**
9. **42CB - iSeries Access for Windows: Security and Communications Tips**
10. **44CA - iSeries Access for Windows in a .NET World**
11. **45LA - OPEN LAB: iSeries Access for Windows with the Experts**
12. **52CB - Everything you wanted to know about PC5250 emulation**
13. **56CB - Performance Tune iSeries Access ODBC Driver**

Session Title: iSeries Access in the .NET World

Session ID: 420219

Agenda Key: 31CU

Speaker: Brent Nelson

Sample Code Disclaimer

This material contains IBM copyrighted sample programming source code for your consideration. This sample code has not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function. IBM provides no program services for this material. This material is provided "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO YOU. IN NO EVENT WILL IBM BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS MATERIAL INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM OR OTHERWISE, EVEN IF EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Trademarks and Disclaimers

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	e-business on demand	OS/400
AS/400e	IBM	i5/OS
eServer	IBM (logo)	
eserver	iSeries	

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
Other company, product or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore,
no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.