SYBASE

APPLICATION DEVELOPMENT WITH SQL ANYWHERE FOR .NET

Bill Hillis – hillis@ianywhere.com
Senior Manager, Software Engineering
iAnywhere Solutions

SYBASE TechWave 2005
USER TRAINING & SOLUTIONS CONFERENCE

Agenda

- Introduction to .NET
- ASA and ADO.NET
- Visual Studio Integration
- What's new in "Jasper"

.NET: Definition

.NET technology enables the creation and use of XML-based applications, processes, and Web sites as services that share and combine information and functionality with each other by design, on any platform or smart device, to provide tailored solutions for organizations and individual people.

.NET is a comprehensive family of products, built on industry and Internet standards, that provide for each aspect of developing (tools), managing (servers), using (building block services and smart clients) and experiencing (rich user experiences) Web services.

http://www.microsoft.com/net/basics/faq.asp

.NET Framework

- Infrastructure for the overall .NET platform
- Common Language Runtime (CLR)
  - Managed, protected application execution environment
  - C#, Visual Basic.NET, C++, J#, …
- Common Class Libraries
  - Windows Forms, ADO.NET, ASP.NET,…

- .NET Compact Framework
  - Subset of .NET Framework for smart devices
  - Part of Visual Studio.NET 2003
  - Included on device with CE.NET (CE 4.1), add-on for older devices

.NET: Managed Code

- Code is written in desired language (C++, C#, VB.NET, Pascal, etc.)
- Compiled into Microsoft Intermediate Language (MSIL)
- At runtime Common Language Runtime (CLR) compiles the MSIL code and executes it

.NET Terms

Namespace

- "A logical naming scheme for grouping related types"
- Analogous to Java packages
- iAnywhere.Data.AsaClient.AsaConnection ➔ iAnywhere.Data.AsaClient is the namespace

Assembly

- "A collection of one or more files that are versioned and deployed as a unit"
- DLLs in .NET-land
- Unlike DLLs, .NET assemblies also include version control/security information

GAC (Global Assembly Cache)

- "A machine-wide code cache that stores assemblies specifically installed to be shared by many applications on the computer"

What is ADO.NET?

- Microsoft's latest data access API
  - ODBC, DAO, RDO, OLE DB, ADO
- Implements System.Data namespace
- "Data providers" manage access to data stores
- Providers from Microsoft:
  - System.Data.OleDb
  - System.Data.Odbc
  - System.Data.SQLClient
  - System.Data.OracleClient

"Managed provider" ➔ "Data provider"

ADO.NET Interfaces

Each data provider implements the following classes:

- Connection – connects to data source
- Command – executes commands
- DataReader – forward-only, read-only access to data
- DataAdapter – fills DataSet and handles updates

- Parameter – parameter to a Command object
- Transaction – provides commit/rollback functionality
- Error, Exception – collect error/warning messages

Agenda

- Introduction to .NET
- ASA and ADO.NET
- Visual Studio Integration
- What's new in "Jasper"

ASA Interfaces

ODBC
ESQL
OLEDB
Open Client
JDBC
Perl
PHP
…

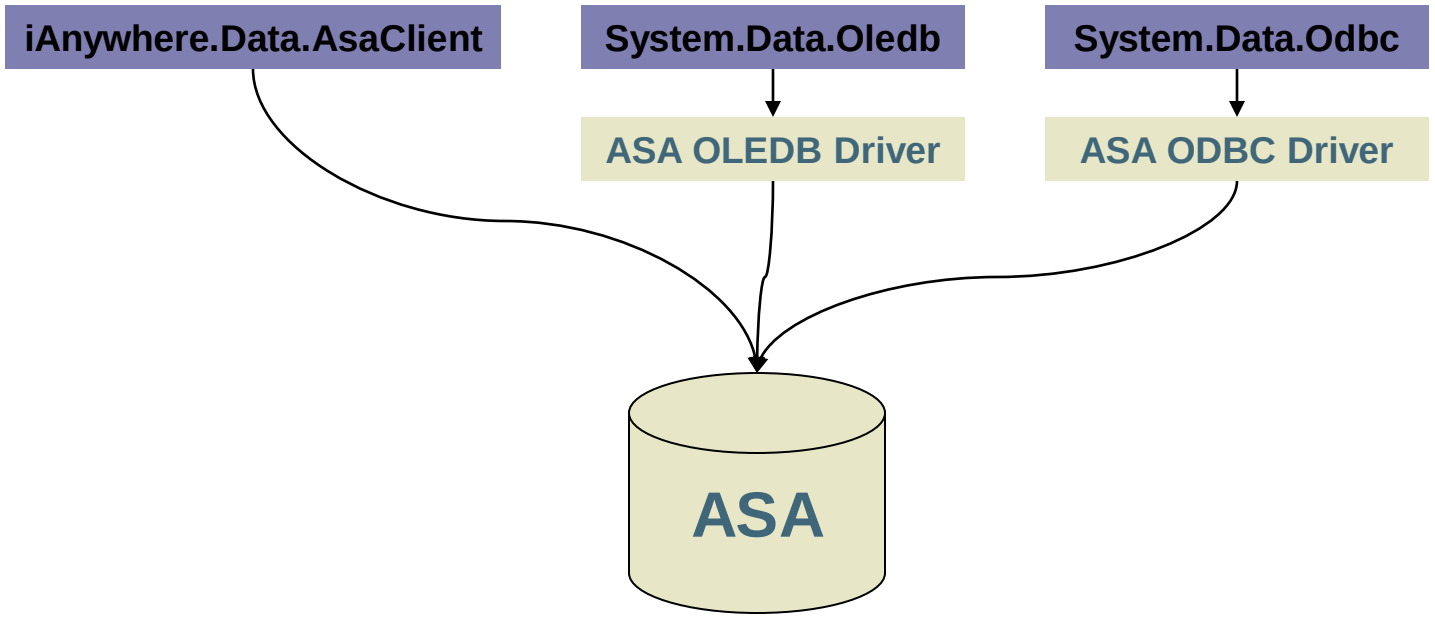
ADO.NET

ADO.NET Data Providers For ASA

OLEDB

ODBC

AsaClient



SYBASE®

ASA Data Provider

- Implements iAnywhere.Data.AsaClient namespace
  - AsaConnection
  - AsaCommand
  - AsaDataReader
  - AsaDataAdapter
  - etc.
- Supports Windows (.NET framework) and Windows CE (.NET compact framework)
- Available since 8.0.2.4122 (March 2003)

Example using DataReader (C#)

```
AsaConnection conn =
        new AsaConnection( “dsn=ASA 9.0 Sample” );
conn.Open();
AsaCommand cmd = new AsaCommand(
        “select emp_lname from employee”, conn );
AsaDataReader reader = cmd.ExecuteReader();
while( reader.Read() ) {
    str = reader.GetString( 0 );
    Console.WriteLine( str );
}
reader.Close();
conn.Close();
```

Example using DataAdapter (VB.NET)

```
Dim conn As New
        iAnywhere.Data.AsaClient.AsaConnection()
conn.ConnectionString = “dsn=ASA 9.0 Sample"
conn.Open()

Dim ds As New DataSet()
Dim da As New AsaDataAdapter
                ("select * from employee", conn)
da.Fill(ds, "Employees")

DGEmployees.DataSource = ds
DGEmployees.DataMember = "Employees"
```

Using the ASA Data Provider

Use Visual Studio.NET (VS.NET 2003 preferred)

Reference the provider in your project (required)

- Project menu, Add Reference
- In the .NET tab, find iAnywhere.Data.AsaClient.dll
- If the provider is not listed, find it in %ASANY9%\win32

Reference provider in your code (optional)

- Allows you to use ASA provider classes without namespace prefix
- C#: `using iAnywhere.Data.AsaClient`
- VB.NET: `Imports iAnywhere.Data.AsaClient`

ASA ADO.NET Data Provider

Classes

- AsaConnection
- AsaError
- AsaException

- AsaCommand
- AsaParameter
- AsaDataReader
- AsaDataAdapter

- AsaCommandBuilder
- AsaErrorCollection
- AsaInfoMessageEventArgs
- AsaParameterCollection

Classes (continued)

- AsaPermission
- AsaPermissionAttribute
- AsaRowUpdatedEventArgs
- AsaRowUpdatingEventArgs
- AsaTransaction

Enumerations

- AsaDbType

Delegates

- AsaInfoMessageEventHandler
- AsaRowUpdatedEventHandler
- AsaRowUpdatingEventHandler

SYBASE®

ADO.NET Application Tasks

Connecting

Error handling

Executing SQL

Retrieving data

Transactions

Disconnected result sets

Connecting

AsaConnection

- Represents a connection to an ASA database
- Uses normal ASA connection strings
- Optional use of event handlers (InfoMessage, StateChange)

Connection Example

```
using iAnywhere.Data.AsaClient;

private AsaConnection myConn;

myConn = new iAnywhere.Data.AsaClient.AsaConnection();

myConn.ConnectionString =
        "Data Source=ASA 9.0 Sample;UID=DBA;PWD=SQL”;
myConn.Open();
…
myConn.Close();
```

Connection Pooling

ADO.NET spec dictates that connection pooling be enabled by default

- Even on Windows CE!
- Disable in connection string: POOLING=false

ASA engine may not auto-stop with pooling enabled

- Connection object must be destroyed first

```
myConn.ConnectionString =
        "POOLING=FALSE;Data Source=ASA 9.0 Sample;UID=DBA;PWD=SQL";
```

Error Handling

AsaException

- Thrown by a failed statement (try/catch)
- Message parameter contains error message
- Errors property is a collection of ASAError objects

AsaError

- More detailed ASA-specific error information
- Message, NativeError, Source, SqlState

Errors and Exceptions Example

```
try {
        myConn = new AsaConnection(
                "Data Source=ASA 9.0 Sample” );
        myConn.Open();
} catch( AsaException ex ) {
        MessageBox.Show(
                ex.Errors[0].Source + " : " +
                ex.Errors[0].Message + " (" +
                ex.Errors[0].NativeError.ToString() + ")",
                "Failed to connect" );
}
```

Demo:

Executing SQL

AsaCommand

- Represents a SQL statement or stored procedure that is executed against an Adaptive Server Anywhere database
- Parameter property is a collection AsaParameter objects

Multiple methods to execute your SQL

- ExecuteNonQuery (returns a row count)
- ExecuteReader (returns result set – DataReader)
- ExecuteScalar (returns a single value – column 1, row 1)

Stored procedures

- Use the name of the procedure as the statement (no “call” or “exec”)
- Set the CommandType property to StoredProcedure

AsaCommand Example

```
AsaConnection myConn;
AsaCommand myCmd;
int num_depts;

myConn = new AsaConnection(
                    “ENG=asademo9;UID=DBA;PWD=SQL";
myConn.Open();

myCmd = new AsaCommand(
                    "select count(*) from department", myConn );
num_depts = (int) myCmd.ExecuteScalar();
```

Retrieving Data

AsaDataReader

- Read-only, forward-only result set from a query or stored procedure (rows are fetched as needed)
- GetXXX methods to get column value as specific data types
- Read method moves to next row

AsaDataAdapter

- Used to fill a DataSet; fetches all rows and closes cursor
- More on this later…

DataReader Example

```
AsaCommand cmd = new AsaCommand("select * from department",
    myConn);
AsaDataReader reader;
reader = cmd.ExecuteReader();

while( reader.Read() ) {
        int dept_id = reader.GetInt32(0);
        string dept_name = reader.GetString(1);
        MessageBox.Show( "dept_id: " + dept_id +
                                "\ndept_name: " + dept_name );
}
reader.Close();
```

DataReader Example – BLOBs

```
AsaCommand cmd = new AsaCommand(
        "select bitmap from images where id = 1", myConn);
AsaDataReader reader = cmd.ExecuteReader();

if( reader.Read() ) {
   // get the length of the BLOB by passing a NULL buffer
   long len = reader.GetBytes(0, 0, null, 0, 0);
   byte bitmap[] = new byte[len];
   // get the BLOB
   reader.GetBytes(0, 0, bitmap, 0, (int)len);
}
reader.Close();
```

Demo:

Transactions

AsaTransaction

- Represents a SQL transaction
- Returned by ASAConnection.BeginTransaction()
- Commit, Rollback methods
- IsolationLevel property

Autocommit

Autocommit is on by default in ADO.NET

- Disable by explicitly using an AsaTransaction

```
myTran = myConn.BeginTransaction();
myCmd = new AsaCommmand(
    “call sp_update_some_data()”, myConn, myTran );
…
myTran.Commit();
```

Disconnected Result Sets

ADO.NET DataSet (System.Data.DataSet)

- Disconnected data access
- In-memory cache of data retrieved from database
- A collection of DataTables which consist of:
  - DataRow (data)
  - DataColumn (schema)
  - DataRelation (relate DataTables via DataColumns)
- Can read/write data/schema as XML documents
- Works with data providers to load and modify data using the provider's DataAdapter

AsaDataAdapter

AsaDataAdapter

- Represents a set of commands and a database connection used to fill a DataSet and to update a database
- Fill method fetches all rows and closes cursor
- Update method applies changes to DataSet to database (beware of ConcurrencyException!)
- SelectCommand, InsertCommand, UpdateCommand, DeleteCommand properties

AsaCommandBuilder

- Attached to an AsaDataAdapter
- Given a SELECT statement, generates corresponding INSERT/UPDATE/DELETE statements

DataAdapter Example

```
AsaDataAdapter da;
AsaCommandBuilder cb;
DataSet ds;

da = new AsaDataAdapter("select * from product", conn );
cb = new AsaCommandBuilder(da);

ds = new DataSet();
da.Fill(ds, "product");

. . .

da.Update(ds, "product“ );
```

SYBASE®

Demo:

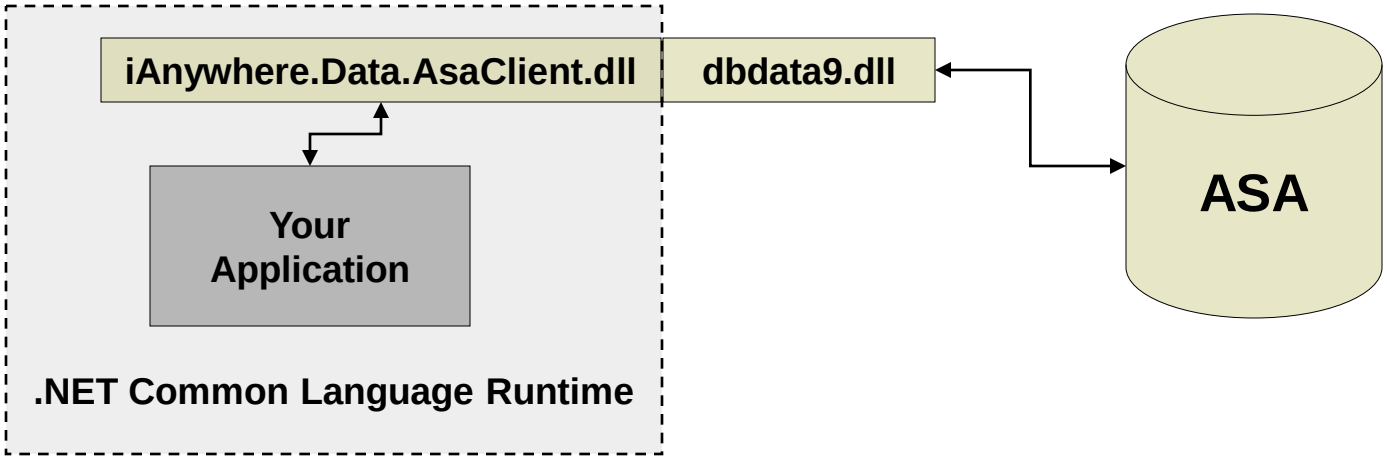
Application Deployment

ASA ADO.NET Provider has two files

- iAnywhere.Data.AsaClient.dll (managed code)
- dbdata9.dll (native code)

Both files must be deployed

- Version of files (i.e. build number) must match
- iAnywhere.Data.AsaClient.dll will throw error if versions don't match



SYBASE®

Updating the ASA Provider

At compile time, .NET compilers use strong name of referenced assemblies

- Strong name includes both name AND version
- Microsoft's attempt to eliminate "DLL hell"

At run time, .NET looks for assemblies based on strong name

An application compiled with iAnywhere.Data.AsaClient version 9.0.1.1234 will only run with version 9.0.1.1234 UNLESS you have a publisher policy file in place

Publisher Policy Files

- Policy files redirect one version of an assembly to another
- Installed into GAC
- ASA EBFs install policy files, for example:
  - Application built against 9.0.1.1000
  - EBF applied to machine; upgrade to 9.0.1.1883
    - EBF installs policy file
    - Requests for 9.0.0.0 – 9.0.1.1883 redirected to 9.0.1.1883
  - %ASANY9%\win32\iAnywhere.Data.AsaClient.dll.config

- Security is built-in to policy files
  - Policy files cannot be compiled without private key assembly was signed with
  - Only iAnywhere can create policy files for iAnywhere assemblies

Example Policy File

```
<configuration>
    <runtime>
        <assemblyBinding xmlns="urn:schemas-microsoft-
  com:asm.v1">
            <dependentAssembly>
                <assemblyIdentity
                    name="iAnywhere.Data.AsaClient"
                    publicKeyToken="f222fc4333e0d400"
                />
                <bindingRedirect
                    oldVersion="9.0.1.0-9.0.1.1883"
                    newVersion="9.0.1.1883"
                />
            </dependentAssembly>
        </assemblyBinding>
    </runtime>
</configuration>
```

SYBASE®

Application Deployment: Win32

Files can go anywhere in the path or program directory, but typically in <%ASANY9%>\win32

iAnywhere.Data.AsaClient.dll

policy.iAnywhere.Data.AsaClient.dll

- Register with gacutil.exe (shipped with .NET)

dbdata9.dll

dblgen9.dll

- No registration required

Application Deployment: Windows CE

One iAnywhere.Data.AsaClient.dll for all CE platforms

- Deploy to the Windows or application directory
- Visual Studio.NET will deploy automatically
- *Make sure to use the CE version of the DLLs!*

Separate dbdata9.dll for each CE platform

- In %ASANY9%\ce\xxx
- Can go in Windows directory or your application's directory on the device

Policy files are not supported by .NET Compact Framework

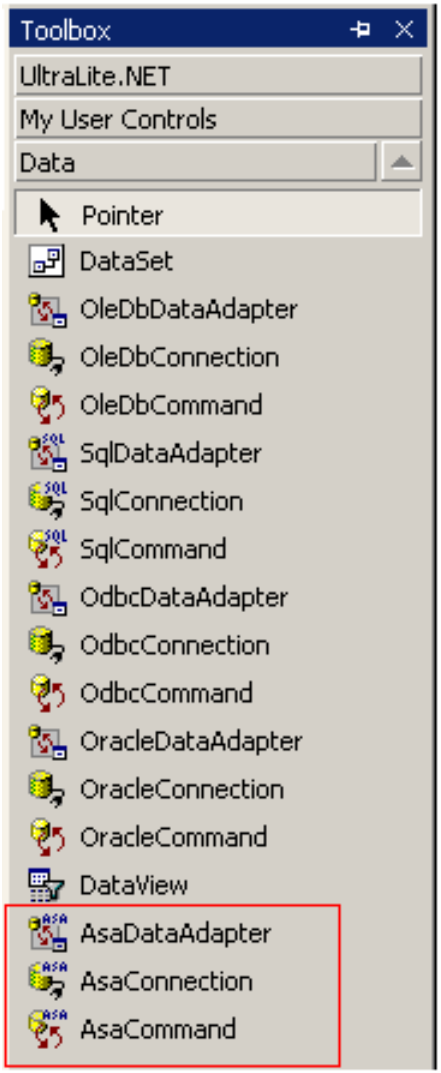
- .NET will automatically use newest version of iAnywhere.Data.AsaClient.dll that it finds

Agenda

- Introduction to .NET
- ASA and ADO.NET
- Visual Studio Integration
- What's new in "Jasper"

Visual Studio Integration

- Added in 9.0.2 (November 2004)
- Described in white paper
  - www.ianywhere.com/downloads /whitepapers/integrate_vs.pdf
- Adds 3 icons to the toolbar
- Enables developers to be more productive



SYBASE®

Visual Studio Integration

Demo

Agenda

- Introduction to .NET
- ASA and ADO.NET
- Visual Studio Integration
- What's new in "Jasper"

What's New in "Jasper"

- Namespace is renamed
- iAnywhere.Data.SQLAnywhere
  - SAConnection
  - SACommand
  - SADataReader
  - SADataAdapter
- More integration with Visual Studio .NET
  - SQL Anywhere Explorer
- ADO.NET 2.0

SQL Anywhere Explorer

Demo

ADO.NET 2.0

- Releasing with Visual Studio 2005 in November
- New features
  - Provider factories
  - Data source enumeration
  - Connection string builder
  - Metadata schemas
  - Asynchronous commands
  - Snapshot isolation level

Summary

Does SQL Anywhere Studio support .NET?

YES!

iAnywhere at TechWave 2005

Ask the iAnywhere Experts on the Technology Boardwalk (exhibit hall)

- Drop in during exhibit hall hours and have all your questions answered by our technical experts!
- Appointments outside of exhibit hall hours are also available to speak one-on-one with our Senior Engineers. Ask questions or get your yearly technical review – ask us for details!

TechWave ToGo Channel

- TechWave ToGo, an AvantGo channel providing up-to-date information about TechWave classes, events, maps and more –now available via your handheld device!
- www.ianywhere.com/techwavetogo

iAnywhere Developer Community - A one-stop source for technical information!

Access to newsgroups,new betas and code samples

- Monthly technical newsletters
- Technical whitepapers,tips and online product documentation
- Current webcast,class,conference and seminar listings
- Excellent resources for commonly asked questions
- All available express bug fixes and patches
- Network with thousands of industry experts

http://www.ianywhere.com/developer/

SQL Anywhere ‘Jasper’ Release

Learn more about **'Jasper', the upcoming SQL Anywhere release,** loaded with features focused on:

- Enhanced data management including performance, data protection, and developer productivity
- Innovative data movement including manageability, flexibility and performance, and messaging

Attend the following sessions:

SQL Anywhere 'Jasper' New Feature Overview
Session SQL512 will be held **Monday, August 22nd, 1:30pm**

MobiLink 'Jasper' New Feature Overview
Session SQL515 will be held **Wednesday, August 24th, 1:30pm**

... **and** remember to look for sneak peeks in other sessions and morning education courses!

Register for the Jasper Beta program:
www.ianywhere.com/jasper

SYBASE®