

# Programming with Microsoft Visual Basic.NET

## Lesson 7

I135-1-A @ Peter Lo 2010

1

## What have we learnt last week?

- Connect Database using ADO.NET through ODBC
- Introduction to SQL Connection
- Development of a Simple Order System
- Crystal Report



I135-1-A @ Peter Lo 2010

2

# Deployment

Deploy your application to users

I135-1-A @ Peter Lo 2010

3

## Deploying with an MSI File Overview

- Windows Installer creates .msi files
- Proven technology but no panacea
- Best option for traditional deployment
- Extended to support needs of .NET
- Visual Studio .NET 2003:
  - Uses Windows Installer 2.0
  - Setup and Deployment project templates

I135-1-A @ Peter Lo 2010

4

## Deploying with an MSI File Deploying the .NET Framework

- Framework is not installed automatically
- Distribute using Dotnetfx.exe
- Cannot include in a Visual Studio .NET Setup Project
- Deployment options
  - Manual installation
  - Bootstrapper
  - Electronic Software Distribution
  - Active Directory or SMS

## Deploying with an MSI File

Customization Option	Description
Setup Project Properties	✓ Author, product name, description, bootstrapper, MSI name, etc.
Project Output	✓ The assemblies and other files you are deploying
Launch Conditions	✓ Specify conditions that must exist prior to installation
Program Menu and Desktop Shortcuts	✓ Add items to the Start menu or desktop
User Interface Editor	✓ Add/remove/customize dialog boxes used in the Setup wizard
Custom Actions	✓ Launch scripts or execute external programs during installation

## Publishing an Application with Click Once Deployment

- After an application is completely debugged and working properly, you can deploy the project
- Deploying a project means placing an executable version of the program on your hard disk, on a Web server, or on a network server
- When programming using Visual Basic 2008, you can create a deployed program by using ClickOnce Deployment
- The deployed version of the program you create can be installed and executed on any computer that has the .NET framework installed

## Smart Device Development

**Develop Application for Smart Phone and  
Pocket PC**

## .NET Device Programming

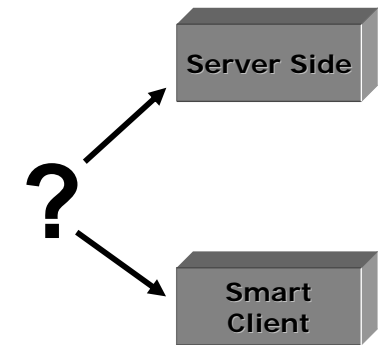


I135-1-A @ Peter Lo 2010

9

## Server Side or Smart Client?

- Server side
  - Broad device support
  - Server side logic
  - No client installation
  - Browser or messaging UI
  - Online only
- Smart Client
  - Target smart clients
  - Client side logic & data
  - UI flexibility
  - Performance
  - Offline & Online



I135-1-A @ Peter Lo 2010

10

## Connected

- Web based solutions
  - Easiest to maintain
  - Easiest to support multiple devices
  - Simple single technology - ASP .Net Mobile Extensions
  - Can not operate outside of connectivity infrastructure

I135-1-A @ Peter Lo 2010

11

## Partially Connected

- Solutions that exchange information when connected to hosts
  - Easy to update in the field
  - Requires bandwidth in balance with data exchanged
  - Requires some device specific development
  - Allows for easiest mixes of technology
  - Most common mobile solution due to limited wireless global infrastructure

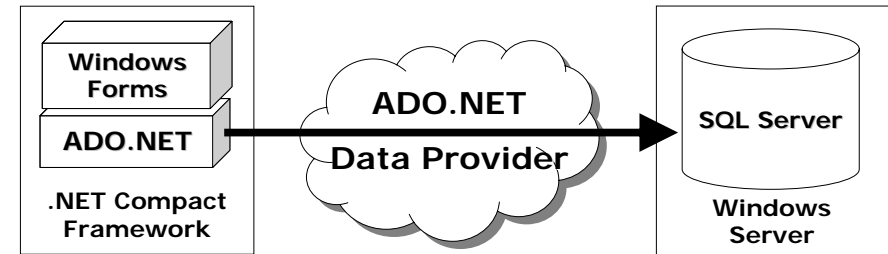
I135-1-A @ Peter Lo 2010

12

## Disconnected

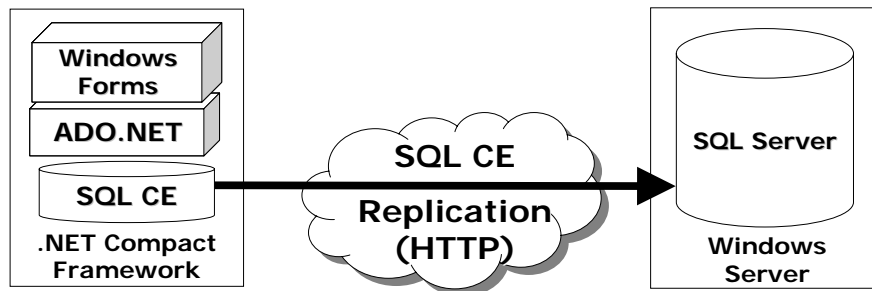
- Solutions that do not connect to the host
  - Less security risks
  - Most gain physical access to device to exchange information or update
  - Requires some device specific programming
  - Simple single technology (Compact Framework)

## Data Access - Direct Database



- Pros
  - Rich query into a large data source
  - Access to *entire* data source; device queries for and uses what's relevant
- Cons
  - Must be constantly connected
  - Not scalable on server; classic "two tier" design
  - No synchronization model for typical dynamic data

## Data Access - SQLCE Replication



- Pros
  - Very robust local store
  - Complex SQL querying
  - Performance synchronization
  - Utilizes HTTP
- Cons
  - Server configuration required
  - Engine footprint on device