

# **Module 11: Upgrading to Visual Basic .NET**

# Overview

- Deciding Whether to Upgrade
- Options for Upgrading
- Recommendations
- Performing the Upgrade

## ◆ Deciding Whether to Upgrade

- Advantages Gained
- Cost Incurred
- Ease of Upgrade

# Advantages Gained

- Scalability
- Performance
- Deployment
- Access to rich set of base classes
- Better debugging
- Solves DLL conflicts
- Maintenance

# Cost Incurred

- Time to upgrade may trade-off against future maintenance time
- May require redesign, as well as upgrading and recoding
- Financial costs can be spread by upgrading an application section by section

# Ease of Upgrade

- Modularity of code
- Project types
- Control types
- Language constructs

## ◆ Options for Upgrading

- Complete Rewrite
- Complete Upgrade
- Partial Upgrade

# Complete Rewrite

## ■ Use if:

- Upgrading is impractical
- Performance is essential

## ■ Advantages

- Best performance
- Best scalability
- Cleanest design
- Reduced code base
- Uses all new features

## ■ Disadvantages

- Labor intensive
- Steep learning curve
- Wasted investment in existing code
- Introduction of errors

# Complete Upgrade

- Not as elegant as a rewrite
- Use if time or resources are limited
  
- Advantages
  - Improved performance
  - Improved scalability
  - Preserved investment in existing code
- Disadvantages
  - Some sections may not upgrade
  - Not best performance

# Partial Upgrade

- Most likely option
- COM interoperability is only a problem if large number of client server calls
  
- Advantages
  - Improved performance
  - Improved scalability
  - Preserves investment in existing code
  - Quick upgrade, and retain non-upgradeable code
- Disadvantages
  - Use of COM interoperability adds overhead
  - Difficult to maintain
  - Difficult to deploy

# Recommendations

- **Web client server**
  - Complete upgrade
  - ASP to ASP .NET and Web Forms, COM components to .NET components, and ADO to ADO .NET
- **Traditional N-tier applications**
  - Partial upgrade
  - Leave client in Visual Basic 6.0
- **Enterprise legacy applications**
  - Complete rewrite
  - Encapsulate legacy system in Web Service
- **Stand-alone Windows-based applications**
  - Little benefit to upgrading

# ◆ Performing the Upgrade

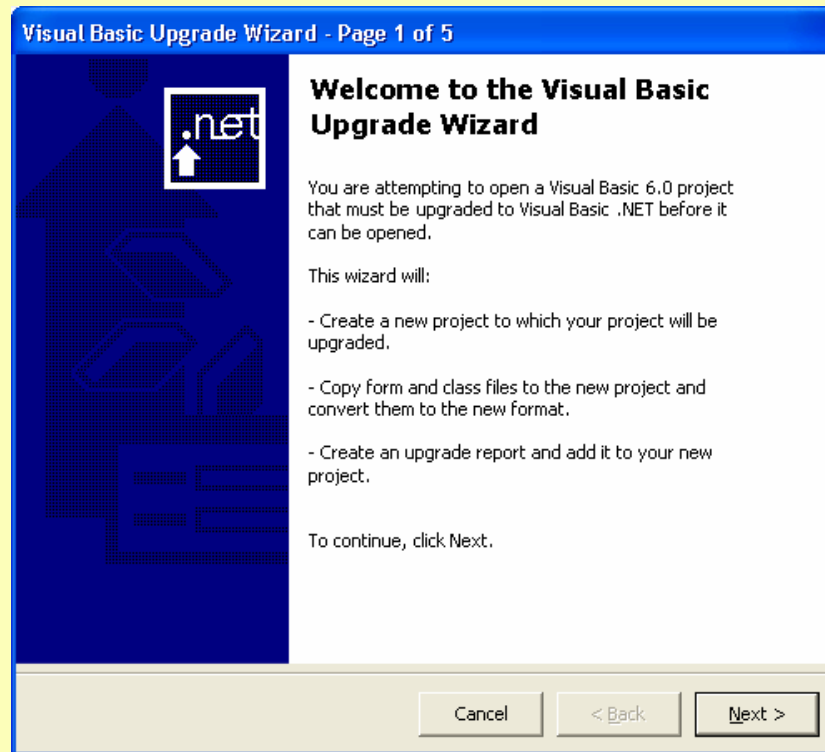
- Preparing for the Upgrade
- Using the Upgrade Wizard
- Results of the Upgrade Wizard
- Completing the Upgrade

# Preparing for the Upgrade

- Early binding
- Null propagation
- Date variables
- Constants
- Data access

# Using the Upgrade Wizard

- Open a Visual Basic 6.0–based project in Visual Basic .NET



# Results of the Upgrade Wizard

- Language changes

- Code upgraded to be syntactically correct in Visual Basic .NET

- Form changes

- Most controls will upgrade

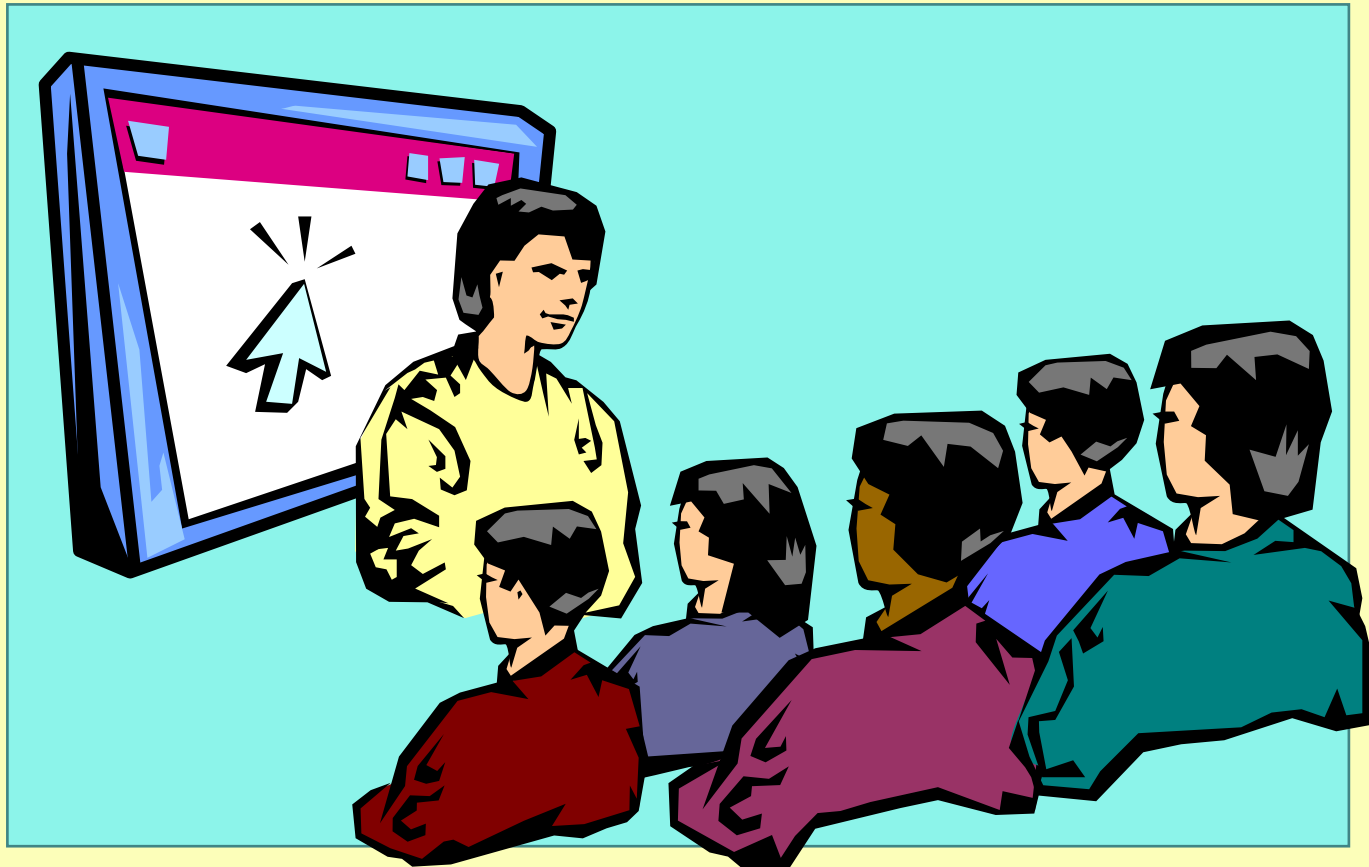
- Other changes

- Other functionality will be upgraded to similar objects

# Completing the Upgrade

- Upgrade Report
- Upgrade comments
- Task List entries
- Testing
- Other tasks

# Demonstration: Using the Upgrade Wizard



# Review

- Deciding Whether to Upgrade
- Options for Upgrading
- Recommendations
- Performing the Upgrade

# Course Evaluation

