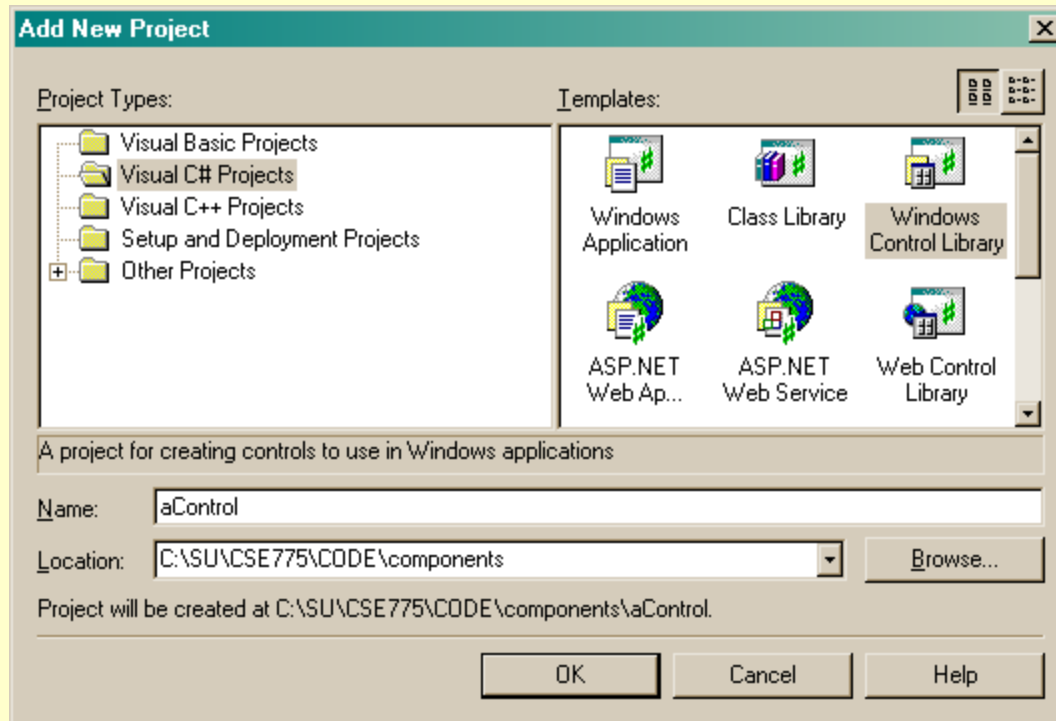

C# Controls

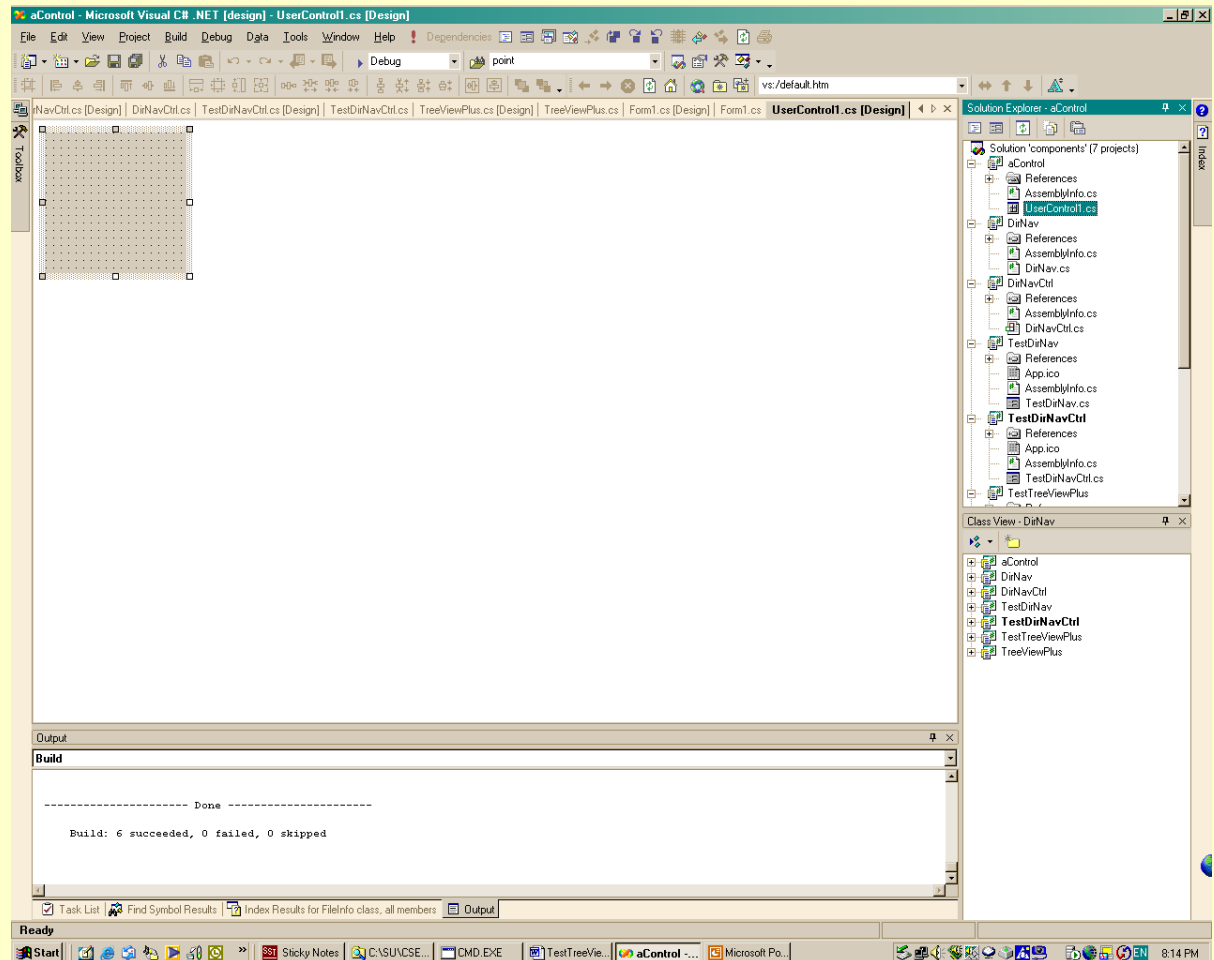
Jim Fawcett
CSE775 – Distributed Objects
Spring 2003

Creating Control Project



That Gives you a UserControl

- Is both a form and a control.
- Intended to become a composite control.
- Like a Form that can be hosted by another form.
- You pull existing controls from toolbox to add functionality.
- Or you can provide your own custom interface and functionality via an override of OnPaint.



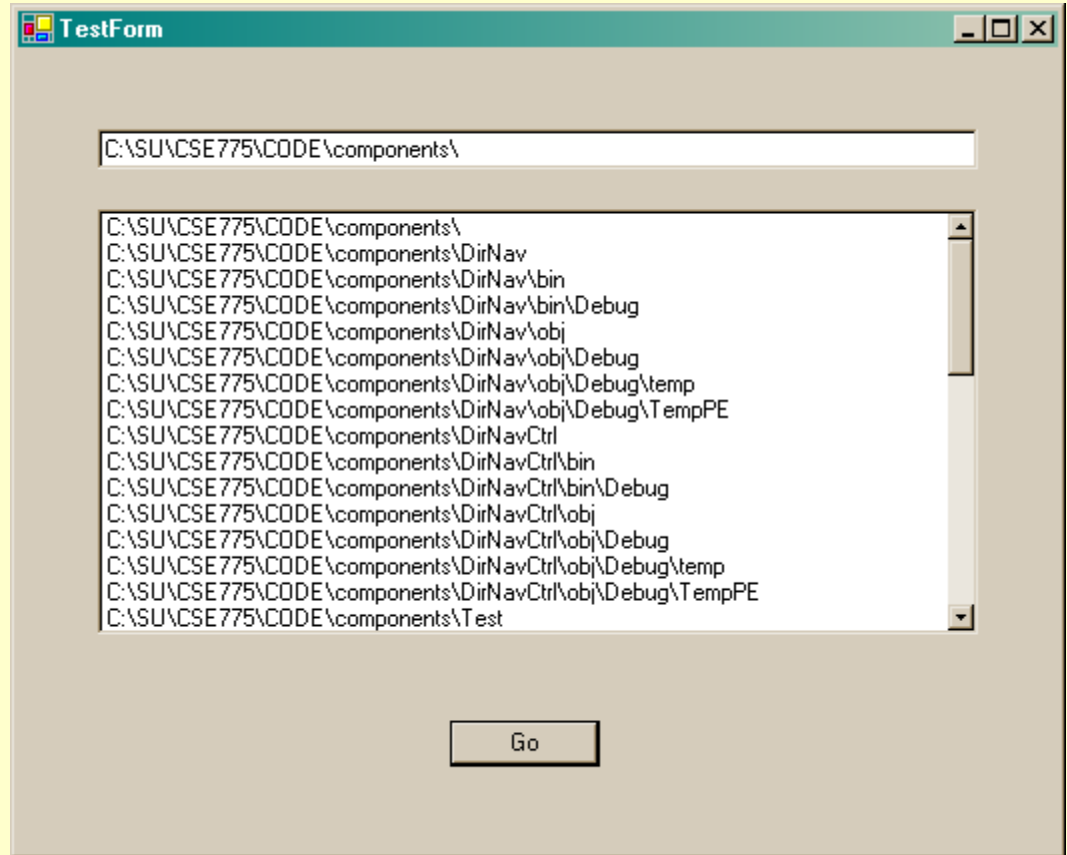
Variations

- You can modify this by changing the class declaration:
 - Derive from `ComponentModel.Component`.
 - That allows library to be hosted in toolbox and be shown in component designer.
 - Derive from `Control` instead of `UserControl`.
 - That takes away the designer view, so you would only want to do this for controls that have no visible view or a very simple view – see `DirNavCtrl`.
 - You still get the ability to be hosted by the toolbox and by a Form
 - Derive from an existing control, like `TextBox` or `TreeView`.
 - You will get whatever designer view was provided for the base control.
 - For a control like the `TreeView` you don't want to populate the control in the class constructor. That can have unexpected effects in the designer.
 - Instead, you provide an initialization function that the hosting form calls in its `Form_Load` event handler.

DirNav Component

DirNav Library

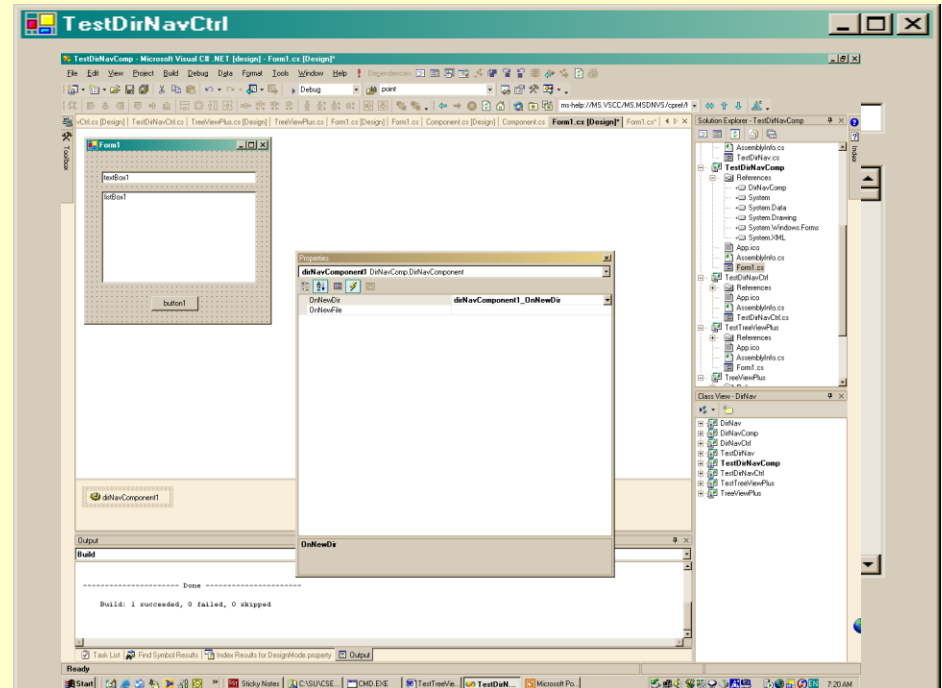
- Simple C# Library
- No base class
- To build this component:
 - Create C# Library project
 - Add go(string path) function with code to recursively walk directory tree
 - Add delegates and events for NewDir and NewFile
- Can't be hosted in toolbox.



DirNavComp Component

DirNavComp Component

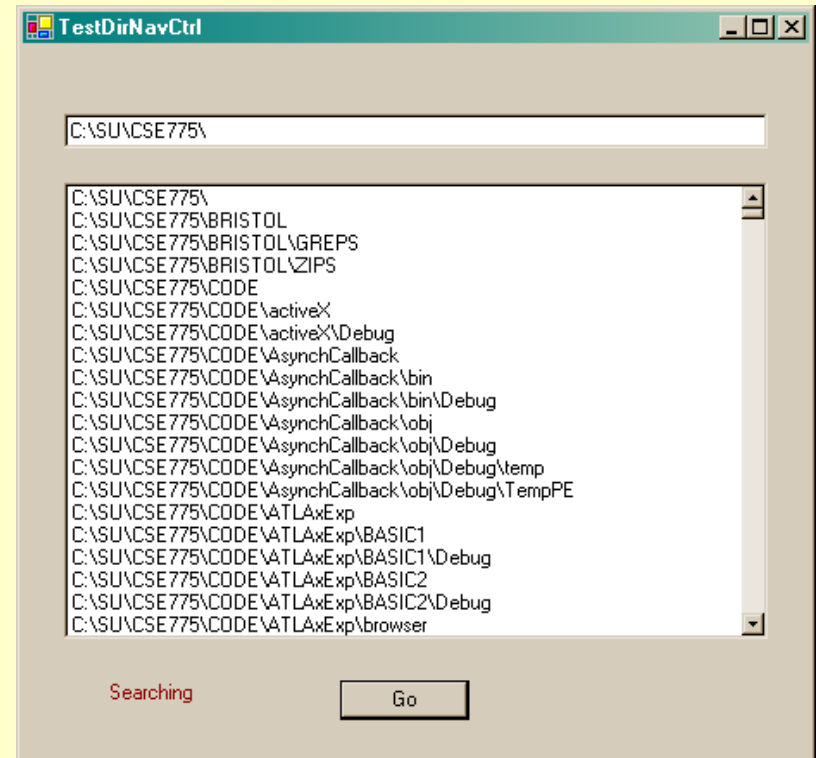
- Navigates directory tree
- Derives from ComponentModel.Component
- To build this control:
 - Create UserControl project
 - Change base class to ComponentModel.Component
 - Repeat DirNav construction
- Can be hosted in toolbox
- Can use property sheet to wire-up event handlers for NewDir and NewFile events in Form hosting control.



DirNavCtrl Control

DirNavCtrl Control

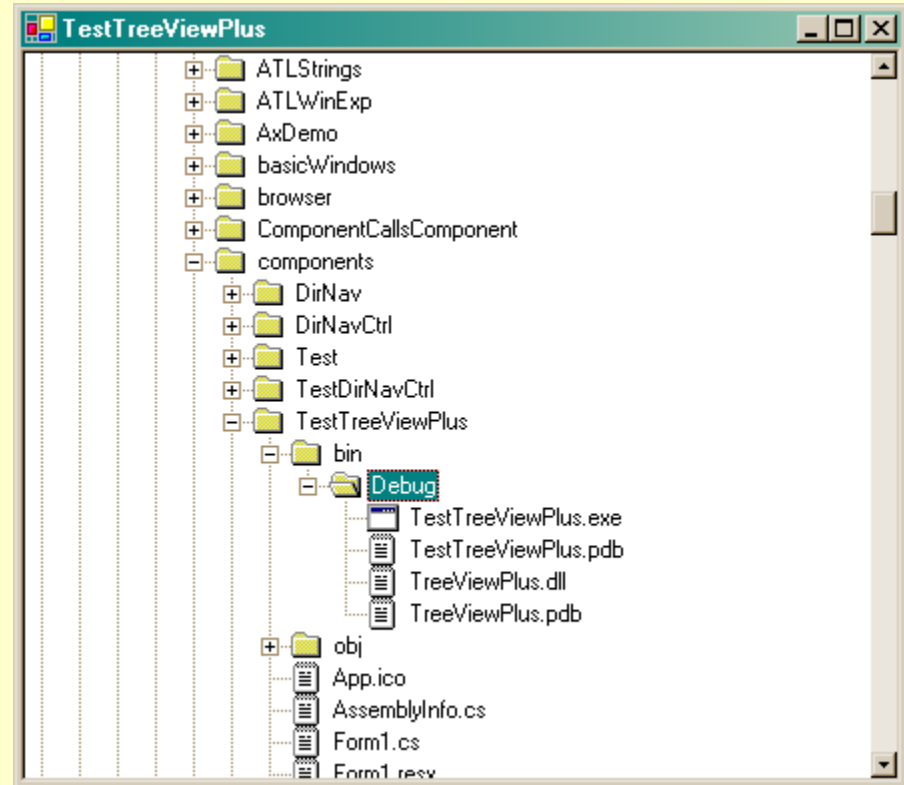
- Navigates directory tree
- Derives from Control
- To build this control:
 - Create UserControl project
 - Change base class to control
 - Repeat DirNav construction
 - Override OnPaint and include drawing code.
- Can be hosted in toolbox
- Can use property sheet to wire-up event handlers for NewDir and NewFile events in Form hosting control.



Controls Derived from Existing Controls

TreeViewPlus

- Derived from TreeView
- To Build this control:
 - Create UserControl project
 - Change base from UserControl to TreeView.
 - Rip out designer code
 - Add code to populate tree with directories and files.
 - Add code to expand to current directory



.Net vs COM Components

.Net

- Library
 - Interface provided directly by class
 - Class factory is operator new using component metadata
- Control
 - Supports hosting in toolbox and on forms.
 - Supports events via delegates
 - Supports properties directly.
- Remoting through binary and standard SOAP channels
 - Simple RPC model
- Can use COM through Run-time Callable Wrapper (RCW).

COM

- Simple ATL component
 - Interface derived from Iunknown
 - Class factory provided by ATL class
- ActiveX Control
 - Supports hosting in toolbox and on forms.
 - Supportst events via connection points
 - Supports properties through IDispatch.
- Remoting through COM specific RPC Channels
 - More complex RPC model
- Can use .Net through COM Callable Wrapper (CCW)

End of Presentation