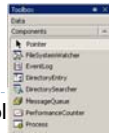
COMPSCI 280 S1 2011
Enterprise Software Development

Developing Windows Applications with Visual Studio 2010
The Process Control & Automation (Extra Reference Only)

Agenda & Reading

- Agenda:
  - The Process Component
- Recommended Reading:
  - Managing a Windows Process
    - http://msdn.microsoft.com/library/en-us/vbcon/html/vbwlkWalkthroughManagingNTProcesses.asp
- Namespace covered:
  - System.Diagnostics
- Hands-on Lab
  - Lecture17Lab

The Process Component



- The Process Control provides access to local and remote processes and enabl to start and stop local system processes
  - A process, in the simplest terms, is a running application.
- Using the Process component, you can obtain a list of the processes that are running, or you can start a new process.
  - The program must be registered in the system registry
- Properties:
  - The **StartInfo** property gets or sets the properties to pass to the Start method of the Process.
  - The **StartInfo.FileName** property gets or sets the application or document to start.
- When Start is called, the StartInfo is used to specify the process to start. The only necessary StartInfo member to set is the FileName property
  - The .exe extension is optional in the FileName member.
    - Notepad or Notepad.exe
  - The filename can be of any file type for which the extension has been associated with an application installed on the system.
    - .doc -> Microsoft Word
    - .txt -> Notepad (text editor)
- Note: Add a using directive:

```
using System.Diagnostics;
```


The Process Component (con't)

- Methods:
  - The **Start** method starts a process resource and associates it with a Process component.
  - The **CloseMainWindow** method closes a process that has a user interface by sending a close message to its main window.
  - The **Kill** method immediately stops the associated process.
  - The **GetProcesses** method creates an array of new Process components and associates them with existing process resources.
  - The **GetProcessesByName** method creates an array of new Process components and associates them with the existing process resources that all share the specified process name.

Example: processDemo

- To create a Process programmatically

```
Process notepadPro = new Process();
notepadPro.StartInfo.FileName = "calc.exe";
```

  - Alternatively, you can create a process at design time
    - Place the Process control on the form (It appears in the Component Tray)
    - Set the **StartInfo** | **FileName** property to the application name, e.g. "Notepad.exe"
- To create a process programmatically using StartInfo

```
ProcessStartInfo startInfo = new ProcessStartInfo("notepad.exe");
```

Tasks

- To start a process:

```
notepadPro.Start(); //or
Process proCal = Process.Start("calc.exe");
```

- To stop a process:

```
notepadPro.CloseMainWindow();
```

- To start a process by file names with known extensions

System will launch the appropriate application type

```
ProcessStartInfo startInfo = new ProcessStartInfo("..//readme.txt");
Process.Start(startInfo);
ProcessStartInfo startInfo = new ProcessStartInfo(txtFileName.Text);
Process.Start(startInfo);
```

Arguments: "C:\WINDOWS\Media\Windows XP Shutdown.wav"

- To get all the processes running on the computer

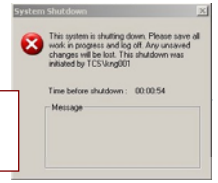
```
foreach (Process proc in Process.GetProcesses()) {
  txtLog.AppendText(proc.ToString() + Environment.NewLine);
}
```

- To get all the processes running on the computer by name

```
foreach (Process proc in Process.GetProcessesByName("calc")) {
  proc.CloseMainWindow();
}
```

Note: the name is "calc" only. (not calc.exe)

Tasks … (optional)



Help: Check for arguments:
Go to My Computer | Help | Help and Support Center
Search for "Command-line reference"

- To shut Down Computer
  - S: shuts down the local computer
  - F: forces all applications to close
  - T: sets the timer (shut down in 60 seconds)

```
Process myProcess = new Process();
myProcess.StartInfo.FileName = @"C:\WINDOWS\system32\shutdown.exe";
myProcess.StartInfo.Arguments = "-s -f -t 60";
myProcess.Start();
```

- To add a scheduled Task
  - /create : create a new scheduled task
  - /tn: Specifies a name for the task.
  - /sc: Specifies the schedule type. onlogon: The task runs whenever a user (any user) logs on.
  - /tr: Specifies the program or command that the task runs

```
ProcessStartInfo startInfo = new ProcessStartInfo("schtasks.exe");
startInfo.Arguments = @"/create /tn PlayMovie /sc onlogon /tr C:\01.avi" ;
Process.Start(startInfo);
```

- You can use the Process component to accomplish most Windows process management tasks quickly and easily. You can use this component to work with processes on either local or remote computers. On a local computer, you can start and stop a process and also query it for specific types of information. On a remote computer, you cannot start or stop a process, but you can query existing processes for information.
- If the path of the scheduled task contains a space, you have to enclose the path portion of the task (not including arguments or switches) between backslash (\) and quotation marks (") character combinations. Refer to http://support.microsoft.com/default.aspx?scid=kb;en-us;823093