TRIBHUVAN UNIVERSITY

Institute of Engineering

Pulchowk Campus

Department of Electronics and Computer Engineering

Minor Project

On

"Network Monitor"

Submitted

By

Baibhav L. R. [27105]	**E-mail: baibhavr@gmail.com**	**Ph: 9841592453**
Ricky Shakya [27135]	**E-mail: shakya_ricky@yahoo.com**	**Ph: 9841478810**
Vivek Gyawali [27144]	**E-mail: leovivu@hotmail.com**	**Ph: 9841413967**

Submitted

To

Department of Electronics and Computer Engineering

Pulchowk Campus

25th February 2008

Acknowledgement

The best part of writing any report is to acknowledge those who have contributed for the success of the project.

This project is carried out as per requirement of minor project in our sixth semester. It would have impossible to proceed and to achieve the design of our project without the help of technical staffs, teachers and friends.

We acknowledge out sincere thanks to Ass. Professor **Er. Jaya Ram Timilsina**, Deputy HOD, Electronics and Computer Department. We cannot miss to thank **Er. Bikash Shrestha** and **Er. Deepen Chapagain**, Department of Electronics and Computer Engineering, for their timely assistance and encouragement.

Our sincere thank also goes to **Er. Babu Ram Dawadi**.
Last but not the least we would like to thank all our friends who helped us directly and indirectly in the completion of this project.

The Authors

Abstract

This project is about the software that can monitor keyboard activities and mouse event of the client computers connected to the server. The prime person in the server could monitor the activities of the client computer users.

This project, Network Monitor, aims to empower the network administrator with the capability of monitoring the client activities. This project describes the use of a system that constantly monitors a computer network for slow or failing systems and that notifies the network administrator. It can be useful to determine sources of errors in computer systems, to study how users interact with systems, and is sometimes used to measure employee productivity on certain clerical tasks.

Table of Contents

Contents	**Page number**
Introduction	4
Objectives	9
Literature review	10
System Development	11
Methodology	16
System Block Diagram	20
Output	21
Scope of the project	23
Conclusion	24
Future Development	25
References	26
The Authors	27

Introduction

One of the fundamental jobs of a network administrator is networking monitoring. Networking monitoring is the process of checking the client computers for the various activities that leads to decrease in the performance of the organization.

This software, Network Monitor, aims to empower the network administrator with the capabilities of monitoring the keyboard activities and mouse event of the client computers connected to the server. The prime person in the server could monitor the activities of the client computer users.

The monitoring is done by examining the keyboard and mouse activities of the client computers within an organization.

The Server module needs to be installed on the computer which acts as server in the network. Only the network administrator or other authorized personnel should have the access to this server computer where server module is installed. The Client module needs to be installed on the computer which activities we want to monitor. All mouse movements and keyboard signals are transferred from the local computer directly to the remote computer, i.e. server where server module of this software is installed, over the network via LAN (by TCP/IP).

TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) is a routable protocol, and the IP part of TCP/IP provides the routing capability. In a routable protocol, all messages contain not only the address of the destination station, but the address of a destination network as well. This allows TCP/IP messages to be sent to multiple networks within an organization or around the world, hence its use in the worldwide Internet (see Internet address). Every client and server in a TCP/IP network requires an IP address, which is either permanently assigned or dynamically assigned at startup.

Keystroke logging (often called keylogging) is a method of capturing and recording user keystrokes. Keylogging can be useful to determine sources of errors in computer systems, to study how users interact with systems, and is sometimes used to measure employee productivity on certain clerical tasks. Such systems are also highly useful for law enforcement and espionage—for instance, providing a means to obtain passwords or encryption keys and thus bypassing other security measures. Keyloggers are widely available on the Internet.

Types of keystroke loggers

1) **Local Machine software Keyloggers** – Are software programs that are designed to work on the target computer's operating system. From a technical perspective they can be categorized into three categories:

* Kernel based: This method is most difficult both to write, and combat. Such keyloggers reside at the kernel level and are thus practically invisible. They almost always subvert the OS kernel and gain unauthorized access to the hardware which makes them very powerful. A keylogger using this method can act as a keyboard driver for example, and thus gain access to any information typed on the keyboard as it goes to the Operating System.

* Hook based: Such keyloggers hook the keyboard with functions provided by the OS. The OS warns them any time a key is pressed and it records it.

* Creative Methods: Here the coder uses functions like GetAsyncKeyState, GetForegroundWindow, etc. These are the easiest to write, but as they require polling the state of each key several times per second, they can cause a noticeable increase in CPU usage and can miss the occasional key.

2) **Remote Access software Keyloggers** – Are local software keyloggers programmed with an added feature to transmit recorded data out of the target computer and make the data available to the monitor at a remote location. Remote communication is facilitated by one of four methods:

* Data is uploaded to a website or an ftp account.

* Data is periodically emailed to a pre-defined email address.

* Data is wirelessly transmitted by means of an attached hardware system.

* It allows the monitor to log into the local machine via the internet or Ethernet and view the logs stored on the target machine itself.

3) **Hardware Keyloggers** - are used for keystroke logging by means of a hardware circuit that is attached somewhere in between the computer keyboard and the computer. It logs all keyboard activity to its internal memory which can be accessed by typing in a series of pre-defined characters. A hardware keylogger has an advantage over a software solution; because it is not dependent on the computers operating system it will not interfere with any program running on the target machine and hence cannot be detected by any software.

4) **Remote Access Hardware Keyloggers** – Or otherwise know as Wireless Hardware Keyloggers work in much the same way as regular hardware keyloggers. Except they have the ability to be controlled and monitored remotely by means of a wireless communication standard.

5) **Wireless Keylogger sniffers** - Collect packets of data being transferred from a wireless keyboard and its a receiver and then attempts to crack the encryption key being used to secure wireless communications between the two devices.

6) **Acoustic Keylogger** - This concept is based on analyzing a recording of the sound created by someone typing on a computer. Each character on the keyboard makes a subtely different acoustic signature when stroked. Using statistical methods similar to decryption, it is then possible to identify which keystroke signature relates to which keyboard character. This is done by analyzing the repetition frequency of similar acoustic keystroke signatures, the timings between different keyboard strokes and other context information such as the probable language in which the user is

writing. As with decryption, a fairly long recording (1000 or more keystrokes) is required so that the statistics are meaningful.

This project can be categorized as the hook based local machine software key logger.

Hook

Hooks tend to slow down the system because they increase the amount of processing the system must perform for each message.

Hook Chains

The system supports many different types of hooks; each type provides access to a different aspect of its message-handling mechanism. For example, an application can use the WH_MOUSE Hook to monitor the message traffic for mouse messages.

The system maintains a separate hook chain for each type of hook. A hook chain is a list of pointers to special, application-defined callback functions called hook procedures. When a message occurs that is associated with a particular type of hook, the system passes the message to each hook procedure referenced in the hook chain, one after the other. The action a hook procedure can take depends on the type of hook involved. The hook procedures for some types of hooks can only monitor messages; others can modify messages or stop their progress through the chain, preventing them from reaching the next hook procedure or the destination window.

Hook Procedures

A global hook monitors messages for all threads in the same desktop as the calling thread. A thread-specific hook monitors messages for only an individual thread. A global hook procedure can be called in the context of any application in the same desktop as the calling thread, so the procedure must be in a separate DLL module. A thread-specific hook procedure is called only in the context of the associated thread.

Objectives

The objectives of our project are as follows:

- To monitor computers in network
- Search for suspicious and malicious activities
- Keep records of individual behavior
- If it’s deployed in home, parents can use this system to monitor their kids

Literature Review

Most of the computer network today present is based on server/client architecture. One computer will act as a server controlling all the others computer (client) in the network. It is very much essential to monitor the activities of the user in the organization as the efficiency of any organization depends upon the activities of the workers.

Our software, Network Monitor, comes to play to ease the task of the network administrator in monitoring the user activities. The client module is installed on the client's computer in the network. It monitors the keyboard strokes and mouse movement continuously and sends the data to the server over the LAN connection. The network administrator or any authorized user can then view the data send over to the server by the client module.

System Development

Development Tools

For the design of this project we used the Microsoft Visual Studio 2005. The visual studio brings many tweaks and improvements to make the whole development process easier and increase the productivity, so we will use this development environment.

The language used is the C#; this is a relatively new language that was unveiled to the world when Microsoft released the first version of its .NET framework. Since then its popularity has rocketed, and it has arguably become the language of choice for both Windows and Web developers who use .NET. Despite its simplicity, C# has retained the power of C++, and there is now no reason not to move into C#.

Microsoft Visual Studio

Microsoft Visual Studio is the main Integrated Development Environment (IDE) from Microsoft. It can be used to develop console and GUI applications along with Windows Forms applications, web sites, web applications, and web services in both native codes as well as managed code for all platforms supported by Microsoft Windows, Windows Mobile, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It allows plug-ins to be added that enhance the functionality at almost every level - including adding support for source control systems to adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle.

Visual Studio supports languages by means of language services, which allow any programming language to be supported (to varying degrees) by the code editor and debugger, provided a language-specific service has been authored. In-built languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), and C# (via Visual C#). Support for other languages such as F#, Python, and Ruby among others has been made available via language services which are to be installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS.

Visual Studio Features

Features

Code editor - The Visual Studio code editor showing IntelliSense suggestions and a docked Task List window. The Visual Studio code editor showing IntelliSense suggestions and a docked Task List window.

Visual Studio, like any other IDE, includes a code editor that supports syntax highlighting and code completion using IntelliSense for not only variables, functions and methods but also language constructs like loops and queries. Visual Studio features background compilation.

Microsoft Visual Studio Debugger

Visual Studio includes a debugger that works both as a source-level debugger as well as machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. Multi-threaded programs are also supported. The debugger can be configured to be launched when an application running outside the Visual Studio environment, crashes.

Data tooltips in Visual Studio

The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses. It can either step into functions to debug inside it, or step over it, i.e., the execution of the function body isn't available for manual inspection.

Designer

Visual Studio includes a host of visual designers to aid in the development of applications. These tools include:

Visual Studio 2005 in Designer view

Visual Studio 2005 in Designer view

The WPF Designer in Visual Studio 2008

The WPF Designer in Visual Studio 2008

Visual Studio 2005 in Class Designer view

Visual Studio 2005 in Class Designer view

WinForms Designer

The WinForms designer is used to build GUI applications using WinForms. It includes a palette of UI widgets and controls (including buttons, progress bars, labels, layout containers and other controls) that can be dragged and dropped on a form surface. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form.

Open Tabs Browser

The open tabs browser is used to list all open tabs and switch between them. It is invoked using CTRL+TAB.

Properties Editor

The Properties Editor tool is used to edit properties in a GUI pane inside Visual Studio. It lists all available properties for all objects including classes, forms, web pages and other items.

Open Tab Browser and Properties Editor

Object Browser

The Object Browser is a namespace and class library browser for Microsoft .NET. It can be used to browse the namespaces in managed assemblies.

Solution Explorer

In Visual Studio parlance, a solution is a set of code files and other resources that are used to build an application. The files in a solution are arranged hierarchically.

Data Explorer

Data Explorer is used to manage databases on Microsoft SQL Server instances. It allows creation and alteration of database tables (either by issuing T-SQL commands or using the Data designer.

Server Explorer

The Server Explorer tool is used to manage database connections on an accessible computer.

Microsoft Visual C#

Microsoft Visual C# is Microsoft's implementation of the C# language that targets the .NET Framework, along with the language services that lets the Visual Studio IDE support C #projects. While the language services are a part of Visual Studio, the compiler is available separately as a part of the .NET Framework. The Visual C# 2008 compiler supports version 3.0 of the C# language specifications. Visual C# supports the Visual Studio Class designer, Forms designer, and Data designer among others.

Methodology

The project has been developed in the client-server model. And the passage of data is in TCP. In the client computer, the hook procedure detects the keyboard and mouse activities. The information is sent to the server computer through the port 5000. In the server computer, the information sent is received from the port 5000. Thus the network administrator can view the activities in the client computers.

Here to monitor the keyboard activities we used hook procedures. This allows you to tap keyboard and mouse and/or to detect their activities.

Global hook

Hook*:* A hook is a point in the system message-handling mechanism where an application can install a subroutine to monitor the message traffic in the system and process certain types of messages before they reach the target window procedure.

Available hooks:

WH_CALLWNDPROC
WH_CALLWNDPROCRET
WH_CBT
WH_DEBUG
WH_FOREGROUNDIDLE
WH_GETMESSAGE
WH_JOURNALPLAYBACK
WH_JOURNALRECORD
WH_KEYBOARD
WH_KEYBOARD_LL
WH_MOUSE
WH_MOUSE_LL
WH_MSGFILTER
WH_SHELL
WH_SYSMSGFILTER

WH_KEYBOARD_LL Hook

The WH_KEYBOARD_LL hook enables us to monitor keyboard input events about to be posted in a thread input queue.

The LowLevelKeyboardProc hook procedure is an application-defined or library-defined callback function used with the SetWindowsHookEx function. The system calls this function every time a new keyboard input event is about to be posted into a thread input queue. The keyboard input can come from the local keyboard driver or from calls to the keybd_event function. If the input comes from a call to keybd_event, the input was "injected". However, the WH_KEYBOARD_LL hook is not injected into another process. Instead, the context switches back to the process that installed the hook and it is called in its original context. Then the context switches back to the application that generated the event.

The HOOKPROC type defines a pointer to this callback function. LowLevelKeyboardProc is a placeholder for the application-defined or library-defined function name.

WH_KEYBOARD Hook

The WH_KEYBOARD hook enables an application to monitor message traffic for WM_KEYDOWN and WM_KEYUP messages about to be returned by the GetMessage or PeekMessage function. You can use the WH_KEYBOARD hook to monitor keyboard input posted to a message queue.

KeyboardProc Function

The KeyboardProc hook procedure is an application-defined or library-defined callback function used with the SetWindowsHookEx function. The system calls this function whenever an application calls the GetMessage or PeekMessage function and there is a keyboard message (WM_KEYUP or WM_KEYDOWN) to be processed.

The HOOKPROC type defines a pointer to this callback function. KeyboardProc is a placeholder for the application-defined or library-defined function name.

WH_MOUSE_LL Hook

The WH_MOUSE_LL hook enables us to monitor mouse input events about to be posted in a thread input queue.

LowLevelMouseProc Function

The LowLevelMouseProc hook procedure is an application-defined or library-defined callback function used with the SetWindowsHookEx function. The system call this function every time a new mouse input event is about to be posted into a thread input queue. The mouse input can come from the local mouse driver or from calls to the mouse_event function. If the input comes from a call to mouse_event, the input was "injected". However, the WH_MOUSE_LL hook is not injected into another process. Instead, the context switches back to the process that installed the hook and it is called in its original context. Then the context switches back to the application that generated the event.

The HOOKPROC type defines a pointer to this callback function. LowLevelMouseProc is a placeholder for the application-defined or library-defined function name.

WH_MOUSE Hook

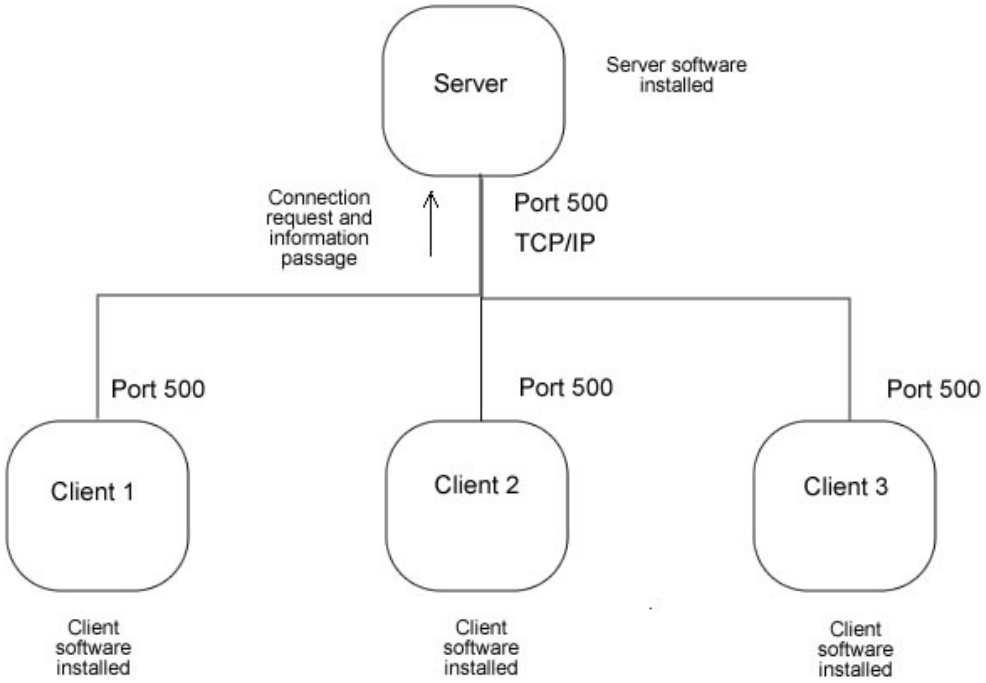
The WH_MOUSE hook enables us to monitor mouse messages about to be returned by the GetMessage or PeekMessage function. You can use the WH_MOUSE hook to monitor mouse input posted to a message queue.

MouseProc Function

The MouseProc hook procedure is an application-defined or library-defined callback function used with the SetWindowsHookEx function. The system calls this function whenever an application calls the GetMessage or PeekMessage function and there is a mouse message to be processed.

The HOOKPROC type defines a pointer to this callback function. MouseProc is a placeholder for the application-defined or library-defined function name.

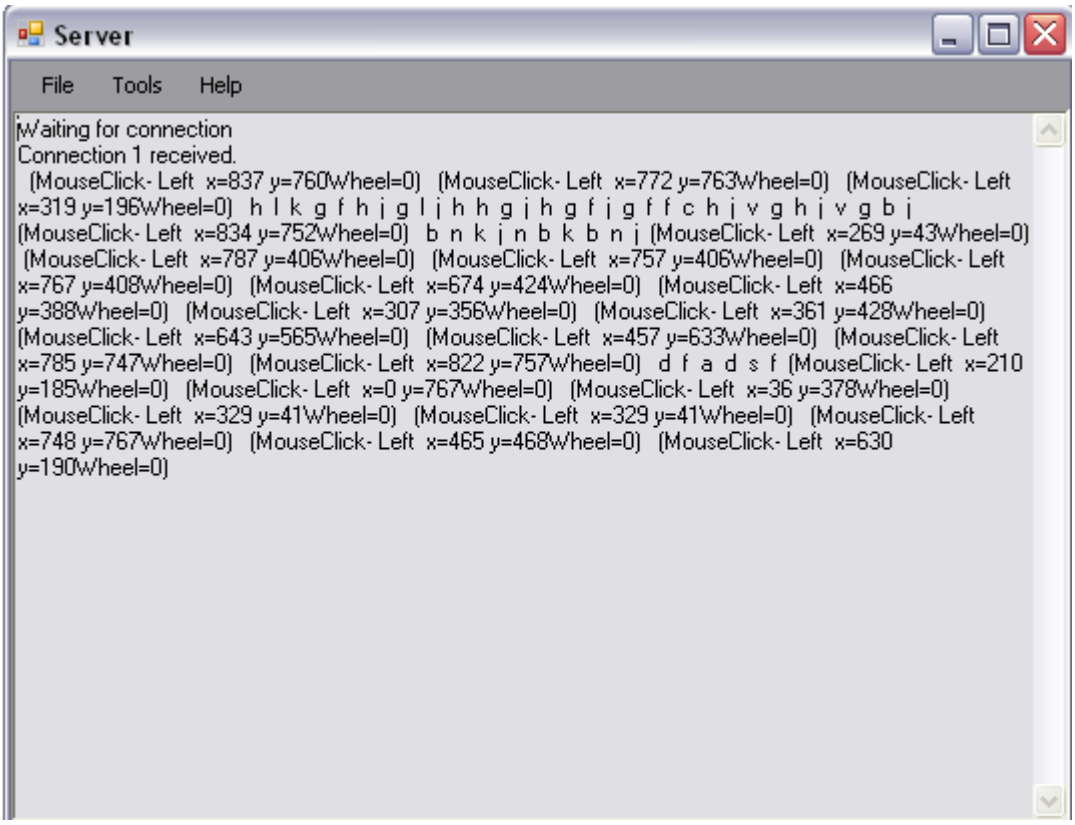
System Block Diagram



Output

Client

Server


Server
File
Tools
Help
Waiting for connection
Connection 1 received.
(MouseClick- Left x=837 y=760Wheel=0) (MouseClick- Left x=772 y=763Wheel=0) (MouseClick- Left x=319 y=196Wheel=0) h l k g f h j g l j h h g j h g f j g f f c h j v g h j v g b j (MouseClick- Left x=834 y=752Wheel=0) b n k j n b k b n j (MouseClick- Left x=269 y=43Wheel=0) (MouseClick- Left x=787 y=406Wheel=0) (MouseClick- Left x=757 y=406Wheel=0) (MouseClick- Left x=767 y=408Wheel=0) (MouseClick- Left x=674 y=424Wheel=0) (MouseClick- Left x=466 y=388Wheel=0) (MouseClick- Left x=307 y=356Wheel=0) (MouseClick- Left x=361 y=428Wheel=0) (MouseClick- Left x=643 y=565Wheel=0) (MouseClick- Left x=457 y=633Wheel=0) (MouseClick- Left x=785 y=747Wheel=0) (MouseClick- Left x=822 y=757Wheel=0) d f a d s f (MouseClick- Left x=210 y=185Wheel=0) (MouseClick- Left x=0 y=767Wheel=0) (MouseClick- Left x=36 y=378Wheel=0) (MouseClick- Left x=329 y=41Wheel=0) (MouseClick- Left x=329 y=41Wheel=0) (MouseClick- Left x=748 y=767Wheel=0) (MouseClick- Left x=465 y=468Wheel=0) (MouseClick- Left x=630 y=190Wheel=0)

Scope of the Project

This system is designed to help monitor the activities in the network so it can be used anywhere where there is an involvement of two or more computers connected together. It can be even used at home where parents can set their computer as a server and monitor the children activities. Besides monitoring for the security purpose or specious behavior, this system can be used to measure the efficiency of the workers based on their activities.

Keystroke logging can be achieved by both hardware and software means. Hardware key loggers are commercially available devices which come in three types: inline devices that are attached to the keyboard cable, devices which can be installed inside standard keyboards, and actual replacement keyboards that contain the key logger already built-in. The inline devices have the advantage of being able to be installed instantly. However, while they may go unnoticed for quite some time, they are easily detected visually upon closer inspection. Of the three devices available, the most difficult to install is also the most difficult to detect. The device that installs inside a keyboard (presumably the keyboard the target has been using all along) requires soldering skill and extended access to the keyboard to be modified. However, once in place, this type of device is virtually undetectable unless specifically looked for.

They also have a legitimate use, that is, they are used to study how users interact with software.

Conclusion

Amidst the project time the project team has tried its best to strike a balance between what is possible and what is acceptable within the project deadline. By the end of the project we were able to present the software capable of monitoring the client activities (keyboard and mouse activities) from the server.

The project has helped us, the project members, to learn the latest technological development and has encouraged us to explore the new horizons. Besides it has also made us appreciate the notion of working in team thus consolidating our engineering studies. Due to short duration of the allocated time for the completion of the entire project is very limited. Despite this the project objectives have been achieved with development of Network Monitor.

Future Developments

Currently the software is designed only to display the keyboard activities and the mouse movements of client computer. However, if we continue this project in future, we would like to augment the Network Monitor with some intelligence capable of analyzing the data (keyboard activities and mouse movements).

Beside during the course of our development we came across the idea of developing program for amateur server administrator. So that he/she could view the client desktops and capture them to track their activities going on in client computer.

References

1. H. M. Deitel, P. J. Deitel, J. Listfield, T. R. Nieto, C. Yaeger, M. Zlatkina, "C# How to program", Pearson Publication
2. "Practical dot .NET 2.0 Networking Projects", Apress
3. "Professional C# 2005", Wiley Publishing
4. MSDN Library for Visual Studio 2005
5. http://en.wikipedia.org

The Authors

Baibhav Lal Rajbhandary
061BCT506
baibhavr@gmail.com

Ricky Shakya
061BCT538
shakya_ricky@yahoo.com

Vivek Gyawali
061BCT547
leovivu@hotmail.com