

Secure Lan Communicator a .Net Project

Dara Ramesh Kumar

Student(M.Tech) , CSE,
Gokul Institute of Technology and Science
Visakhapatnam, India.

K.R.Koteeswa Rao

Asst. Prof, CSE,
Gokul Institute of Technology and Science
Visakhapatnam, India.

ABSTRACT:

A LAN communicator application is developed that will facilitate secure chatting and file sharing over the network. This Secure LAN Communicator consists of two Modules, Server Module and Client Module. Server module will keep record of all the users, their login name, passwords. It will listen to Clients requests. Client module from other computer will first connect to the Server by providing user name and password. If client is authenticated the Server will send back list of all of the users that are online or offline. Client will select any of the online users to chat with or any of the offline users to send him/her an offline messages. All message sent will first go to the Server and then serv

The LAN Chat creates only one chat room that is why it has very clear interface. The LAN Chat's interface is one window with list of chat participants, chat messages and outgoing message. And you can also set to your status online, away or disconnected in this window.

After the user type's text into the input region and hits Return, the text is transmitted to the server. The server echoes back everything that is sent by the client. The client displays everything received from the server in the output region. When multiple clients connect to one server, we have a simple chat system.

I INTRODUCTION:

Secure LAN communicator is an easy-to-use LAN messaging application for effective intra-office communication. It does not require a server and is very easy to install. Secure LAN communicator correctly identifies and works under Windows NT/2000/XP/Vista limited user accounts (without administrative privileges). Secure LAN communicator comes with a variety of handy features such as message notification alarms, personal or group messaging, file transfer and an intuitive interface. Our LAN chat utility offers strong encryption options for all incoming and outgoing messages, guaranteeing no unauthorized person ever reads personal correspondence. The program is very stable when running under any Windows operating system and on large or small TCP/IP networks.

This application can be used for contacting individual users or for group notifications. All chat messages are logged, so there is no risk of losing a message. All correspondence can be looked up at any time. There are convenient sound alarms for incoming messages. The program is very straightforward and requires no special training. It is ready to be used right after installation is completed. Configuring the program can be done by individual users without requesting any assistance from the IT staff or system administrators. All settings can be saved and easily transferred to another computer if necessary. Software Solutions is an IT solution provider for a dynamic environment where business and technology strategies converge. Their approach focuses on new ways of business combining IT innovation and adoption while also leveraging an organization's current IT assets. Their work with large global corporations and new products or services and to implement prudent business and technology strategies in today's environment.

1.1 Xxxxxxx's RANGE OF EXPERTISE INCLUDES:

- Software Development Services.
- Engineering Services.
- Systems Integration.
- Customer Relationship Management.
- Product Development.
- Electronic Commerce.
- Consulting.
- IT Outsourcing.

We apply technology with innovation and responsibility to achieve two broad objectives:

- Effectively address the business issues our customers face today.
- Generate new opportunities that will help them stay ahead in the future.

1.2 THIS APPROACH RESTS ON:

- A strategy where we architect, integrate and manage technology services and solutions - we call it AIM for success.
- A robust offshore development methodology and reduced demand on customer resources.
- A focus on the use of reusable frameworks to provide cost and times benefits.

They combine the best people, processes and technology to achieve excellent results - consistency. We offer customers the advantages of: They understand the importance of timing, of getting there before the competition.

A rich portfolio of reusable, modular frameworks helps jump-start projects. Tried and tested methodology ensures that we follow a predictable, low - risk path to achieve results. Our track record is testimony to complex projects delivered within and even before schedule.

II SYSTEM STUDY:

2.1.Problem Identification:

A LAN communicator application is developed that will facilitate secure chatting and file sharing over the network. This Secure LAN Communicator consists of two Modules, Server Module and Client Module. Server module will keep record of all the users, their login name, passwords. It will listen to Clients requests. Client module from other computer will first connect to the Server by providing user name and password.

If client is authenticated the Server will send back list of all of the users that are online or offline. Client will select any of the online users to chat with or any of the offline users to send him/her an offline messages. All message sent will first go to the Server and then server will send that message to the destination client.

The LAN Chat creates only one chat room that is why it has very clear interface. The LAN Chat's interface is one window with list of chat participants, chat messages and outgoing message. And you can also set to your status online, away or disconnected in this window.

2.2.Existing System:

The existing system has been maintained manually. The system, which has been maintained manually, had been complex and complicated. In the existing system client is not authenticated. The existing system we can use browser means, global connection secure Data will be loss. The existing system chat with any one we can chat the available in chat room.

2.3.Proposed System:

Considering the anomalies in the existing system computerization of the whole activity is being suggested after initial analysis by the organization. The proposed system keeps the Username and password of the application server that is in running and stopped status. The user who wants to use the system should have login and password. Each activity carried out by him can be tracked later by using log files. Hence this helps out in error detection and correction. The proposed system Chat creates only one chat room that is why it has very clear interface and secured.

2.4.Features of Proposed System:

Instant messaging with LAN chat ability Create chat rooms with multiple users' conversation. Secure LAN messaging - Save your privacy All message exchanges are protected by the encryption algorithm Group Broadcast messages notify all users or specified user groups about an event.

III SYSTEM SPECIFICATION:

3.1 Hardware Specification

Hardware Requirements:

PIV 2.8 GHz Processor and Above

RAM 1 GB and Above

HDD 40 GB Hard Disk Space and Above

3.2 Software Specification

Software Requirements:

WINDOWS OS (XP)

Visual Studio .Net 2008 Enterprise Edition

Internet Information Server 6.0 (IIS)

Visual Studio .Net Framework (Minimal for Deployment) version 3.5

Microsoft Visual C# .Net

3.3 About Software:

The Microsoft .NET Framework is a software technology that is available with several Microsoft Windows operating systems. It includes a large library of pre-coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform.

The pre-coded solutions that form the framework's Base Class Library cover a large range of programming needs in a number of areas, including user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. The class library is used by programmers, who combine it with their own code to produce applications.

Programs written for the .NET Framework execute in a software environment that manages the program's runtime requirements. Also part of the .NET Framework, this runtime environment is known as the Common Language Runtime (CLR). The CLR provides the appearance of an application virtual machine so that programmers need not consider the capabilities of the specific CPU that will execute the program. The CLR also provides other important services such as security, memory management, and exception handling. The class library and the CLR together compose the .NET Framework.

3.4 Common Runtime Engine :

The Common Language Runtime (CLR) is the virtual machine component of the .NET framework. All .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling. The design of the .NET Framework allows it to theoretically be platform agnostic, and thus cross-platform compatible. That is, a program written to use the framework should run without change on any type of system for which the framework is implemented. Microsoft's commercial implementations of the framework cover Windows, Windows CE, and the Xbox 360.

In addition, Microsoft submits the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language), the C# language, and the C++/CLI language to both ECMA and the ISO, making them available as open standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms.

Architecture:

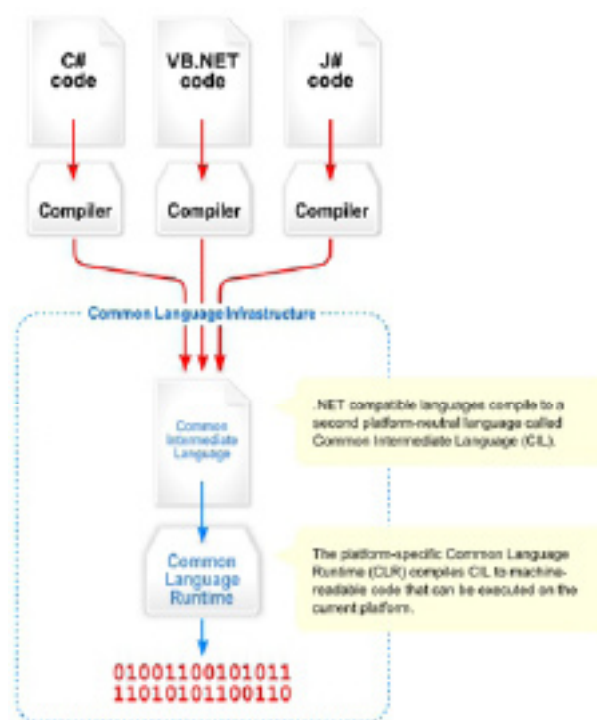


Fig 1 Visual overview of the Common Language Infrastructure (CLI)

3.5 Common Language Infrastructure:

The core aspects of the .NET framework lie within the Common Language Infrastructure, or CLI. The purpose of the CLI is to provide a language-neutral platform for application development and execution, including functions for exception handling, garbage collection, security, and interoperability. Microsoft's implementation of the CLI is called the Common Language Runtime or CLR.

The GC used by .NET Framework is actually generational. Objects are assigned a generation; newly created objects belong to Generation 0. The objects that survive a garbage collection are tagged as Generation 1, and the Generation 1 objects that survive another collection are Generation 2 objects. The .NET Framework uses up to Generation 2 objects.

Higher generation objects are garbage collected less frequently than lower generation objects. This helps increase the efficiency of garbage collection, as older objects tend to have a larger lifetime than newer objects. Thus, by removing older (and thus more likely to survive a collection) objects from the scope of a collection run, fewer objects need to be checked and compacted.

3.6 Versions:

Microsoft started development on the .NET Framework in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2000 the first beta versions of .NET 1.0 were released.



Fig 2 The .NET Framework stack.

Version	Version Number	Release Date
1.0	1.0.3705.0	2002-01-05
1.1	1.1.4322.573	2003-04-01
2.0	2.0.50727.42	2005-11-07
3.0	3.0.4506.30	2006-11-06
3.5	3.5.21022.8	2007-11-09
4.0	4.0.30319.1	2010-04-12

IV Client Application Development:

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on.

Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers. Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

4.1 Server Application Development:

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server.

This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server. The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

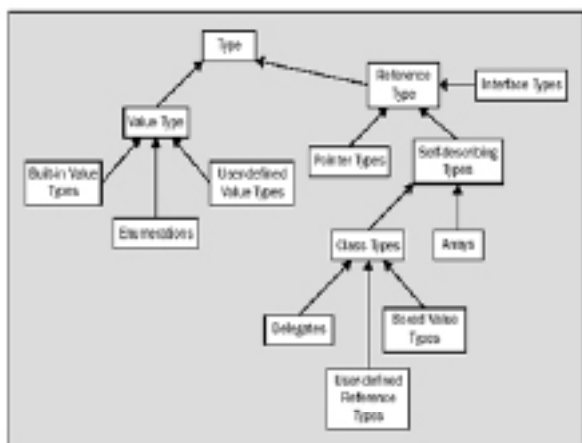


Fig 4 Common Language Specification (CLS)

4.2 Common Language Specification (CLS):

The Common Language Specification works with the Common Type System to ensure language interoperability. The CLS is a set of minimum standards that all compilers targeting .NET must support. Since IL is a very rich language, writers of most compilers will prefer to restrict the capabilities of a given compiler to only support a subset of the facilities offered by IL and the CTS. That is fine, as long as the compiler supports everything that is defined in the CLS.

It is perfectly acceptable to write non-CLS-compliant code. However, if you do, the compiled IL code isn't guaranteed to be fully language-interoperable.

4.3 Security:

.NET can really excel in terms of complementing the security mechanisms provided by Windows because it can offer code-based security, whereas Windows only really offers role-based security. Role-based security is based on the identity of the account under which the process is running, in other words, who owns and is running the process. Code-based security on the other hand is based on what the code actually does and on how much the code is trusted. Thanks to the strong type safety of IL, the CLR is able to inspect code before running it in order to determine required security permissions.

.NET also offers a mechanism by which code can indicate in advance what security permissions it will require to run.

The importance of code-based security is that it reduces the risks associated with running code of dubious origin (such as code that you've downloaded from the Internet). For example, even if code is running under the administrator account, it is possible to use code-based security to indicate that that code should still not be permitted to perform certain types of operation that the administrator account would normally be allowed to do, such as read or write to environment variables, read or write to the registry, or to access the .NET reflection features.

V CONCLUSION:

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in windows forms and C#.NET Network based application and no some extent Windows Application, but also about all chatting procedure related with "Secure LAN Communicator with Chat Application Using Windows". It also provides knowledge about the latest technology used in developing Network enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

VI REFERENCE :

- FOR .NET INSTALLATION

www.support.mircosoft.com

- FOR DEPLOYMENT AND PACKING ON SERVER

www.developer.com

www.15seconds.com

- FOR SQL

www.msdn.microsoft.com

- FOR ASP.NET