# NETWORK MONITORING AND SHARING FTP

By

**M.J.Vinu**

**Reg. No: 71202702013**

**Of**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE.**

**A PROJECT REPORT**

**Submitted to the**

**FACULTY OF SCIENCE AND HUMANITIES**

*In partial fulfillment of the requirements*

*for the award of the degree*

*of*

**MASTER OF SCIENCE**

**IN**

**APPLIED SCIENCE - COMPUTER TECHNOLOGY**

**JUNE, 2004.**

## BONAFIDE CERTIFICATE

Certified that this project report titled "**NETWORK MONITORING AND SHARING FTP**" is the bonafide work of Mr. **M.J.VINU** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROJECT GUIDE**

**HEAD OF THE DEPARTMENT**

The candidate with university registration no: 71202702013 was examined by as in the project vice-voce examination held on..17|06|2004..

R. Dinesh

**INTERNAL EXAMINER**

17/6/04

**EXTERNAL EXAMINER**

# ABSTRACT

The project "Network monitoring and sharing FTP" has networking as its base. FTP, is the name of the application and the protocol used for moving files between two hosts on a TCP/IP network.

File Transfer Protocol, usually called FTP, is a utility for managing files across machines without having to establish a remote session. FTP enables to transfer files back and forth, manage directories, and access electronic mail. FTP is not designed to enable access to another machine to execute programs, but it is the best utility for file transfers.

Therefore, it is essential that the correct file type be specified, since using the wrong format can cause some files to be transferred imperfectly, causing programs not to run, documents to be scrambled, etc. FTP requires login to the server just as most machine require a login : user name and password for an account that exist on the server .In this class, a specific account for the class is setup and FTP client can be used to login and move whatever is necessary.

Underneath the many different interfaces available, FTP actually does some very complex processing. The complexity is required because FTP works across different operating systems, each of which organize their file systems and file storage uniquely.

# ACKNOWLEDGEMENT

A project work is a product of experience and it goes a long way in shaping up a person in one's respective profession, with great gratitude we would like to acknowledge the immense help of all those who contributed with their valuable suggestions and timely assistance to complete this work.

I here by take this opportunity to express my sincere gratitude to the management of Kumaraguru College of Technology, Coimbatore for giving me an opportunity to study in this esteemed institution.

I deem it a great privilege to express my profuse thanks to our beloved principal **Dr. K. K. Padmanabhan, PhD** for providing all facilities during the course of study.

My sincere thanks to our Head of Department of Computer Science and Engineering **Dr. S. Thangasamy, PhD** who has been a support for his constant encouragement and unending enthusiasm to help me finish this project work.

I would like to express our thanks to **Mr. Dinesh Ranganathan, M.S,** Project internal guide, for his words of encouragement were the sparkles that has which helped me a lot to proceed my work.

I sincerely appreciate the guidance, and support given to me by our lecturers, throughout the completion of the project. Finally I wish to extend my sincere thanks to all my friends who took a lot of effort and support to complete this project.

# CONTENTS

| | PAGE NO |
|---|---|
| ABSTRACT | iii |
| LIST OF FIGURES | vi |
| Chapter 1 INTRODUCTION | 1 |
| 1.1 ABOUT THE PROJECT | 1 |
| 1.2 ABOUT THE ORGANIZATION | 4 |
| 1.3 GOALS OF THE PROJECT | 6 |
| Chapter 2 SYSTEM STUDY | 7 |
| 2.1 HARDWARE REQUIREMENTS | 7 |
| 2.2 SOFTWARE REQUIREMENTS | 7 |
| 2.3 SOFTWARE OVERVIEW | 8 |
| Chapter 3 SYSTEM DESIGN | 15 |
| 3.1 BASIC DESIGN CONCEPTS | 15 |
| 3.2 DFD NOTATIONS | 20 |
| Chapter 4 TESTING AND IMPLEMENTATION | 26 |
| 4.1 TESTING PROCESS | 26 |
| 4.2 IMPLEMENTATION | 29 |
| 4.3 MAINTANANCE | 30 |
| 4.4 FUTURE ENHANCEMENTS | 31 |
| Chapter 5 CONCLUSIONS | 32 |
| APPENDICES | 33 |
| REFERENCES | 48 |

# LIST OF FIGURES

**FIGURE NO** **FIGURE DESCRIPTION**

3.2.1 Server Log
3.2.2 Online User
3.2.3 Configuration
3.2.4 Statistics
3.2.5 Security

# CHAPTER 1

# INTRODUCTION

## 1.1 ABOUT THE PROJECT

FTP server is a multithreaded FTP server for windows 98/NT/XP.It comes with an easy to use interface and can be accessed from the system tray. The server handles all basic FTP commands and offers easy user account management and support for virtual directories.

### Installation

Installing FTP server is simple. Create a shortcut on the Start Menu on the desktop and the FTP Server will be ready to use.

### Description of the available modules:

### SERVER LOG

This screen can be used to keep an eye on all the things that are going, on the server. It shows detailed information about all communication between the server and the connected FTP clients.

## ONLINE USERS

This displays a list of users that are currently online. This includes the IP address and login time.

## CONFIGURATION

All the settings of the FTP Server can be configured here.

## FTP PORT

This is the socket port on which new connections are accepted. By default the port used is 21.

## MAXIMUM USERS

It is the maximum number of users that can simultaneously be connected.

## CONNECTION TIMEOUT (IN MINUTES)

When a client has been left idle for a specific time it will be automatically disconnected. This is called connection timeout.

## WELCOME MESSAGE

This is the Text that will be displayed when a clients connects to the server.

## GOODBYE MESSAGE

This is the Text which will be displayed when a client disconnects from the server.

## VIEW LOG

Opens the log file with Notepad and shows the time when the user has logged in.

## CLEAR LOG

*It clears the contents of the log file and shows the time when the user has* logged out.

## STATISTICS

This shows a few interesting statistics of the server like the number of *uploads, downloads and the number of kilobytes sent or received.*

## SECURITY

The security page makes it possible to block certain IP addresses to connect to the FTP server. Add an IP address to the list; it will be kicked off immediately *after it tries to connect. When selecting the block all the IP addresses except the* option (except the ones added to the list), all the IP addresses will be blocked.

## 1.2. ABOUT THE ORGANIZATION

Hindustan Software Ltd., established in 1995, has been actively involved in the development of Customized Software for various clients since its inception and also has its presence strongly felt in the areas of Training & Consulting. Market Leadership through customer satisfaction, a commitment to excellence & high growth rate, these have characterized HSL in its rapid climb in the Information Technology Industry.

Our software development strives to develop innovative software's that meets customer needs. A team of developers with exclusive educational background and excellent computing skills exert their maximum efforts and endeavor to satisfy the specialized requirements of out esteemed clients. Out expertise and vast exposure qualifies us in developing software's in Information Systems, Web Based Applications, Internet Solutions, Web Page Designing and Hosting. Our team consists of expertise in JAVA, VB, Oracle and all Internet Application Development Tools.

Hindustan has trained more than 1000 professionals so far in the area of JAVA & VC++, almost all the students have launched in a very good career path both in India & Abroad. The curriculum matches latest trends and make the students to compete in the IT Industry.

The training division gives us an opportunity to meet variety of candidates with various skill sets. Our database consists of details regarding more than 1000 professionals. So we are fit to assist all sort of IT Human resource consultancies and IT companies in India and Abroad.

Excellence through "TEAM WORK" -The Philosophy of Total Quality Management is inculcated in every HSL employee through intensive training program. Our investment on a highly skilled and motivated manpower constitutes towards stabilizing HSL as truly quality conscious company, continually striving to bring the finest solutions to the discerning customers.

Our wide contacts within the IT industry allow us to locate & select Merit Students to be sent to prospective interviews to leading organizations having opening either in India or Abroad. We take this opportunity to inform the student community that most of our previous students have been immensely benefited from this service which comes free of cost.

## 1.3. GOALS OF THE PROJECT

- promote sharing of files
- encourage indirect use of remote computers
- shield user from variations in file storage
- transfer data reliably and efficiently

# CHAPTER 2

# SYSTEM STUDY

## 2.1 HARDWARE REQUIREMENTS

| | | |
|---|---|---|
| PROCESSOR | : | PENTIUM II OR ABOVE |
| MEMORY | : | 32 MB SDRAM |
| HARD DISK | : | 2 GB OR ABOVE |
| FLOPPY DISK | : | 1.44MB |
| MONITOR | : | 14" COLOR MONITOR |
| KEYBOARD | : | 104 WIN KEYBOARD |
| CLOCK SPEED | : | 450 MHz |
| MOUSE | : | LOGITECH SERIAL MOUSE |

## 2.2 SOFTWARE REQUIREMENTS

| | | |
|---|---|---|
| OPERATING SYSTEM | : | Windows 98 and above |
| LANGUAGE | : | VC++ 6 |

## 2.3 SOFTWARE OVERVIEW

The software used in this project is **"VISUAL C++"**. Visual C++ is much more than a compiler. It is a complete application development environment that, when used as intended, lets exploit the object oriented nature of C++ to create professional Windows applications. In order to take advantage of these features, we need to understand the C++ program.

The Microsoft Foundation Class (MFC) hierarchy encapsulates the user interface portion of the Windows API, and makes it significantly easier to create Windows applications in an object oriented way. This hierarchy is available for and compatible with all versions of Windows. This code in MFC is extremely portable.

The MFC library is an attempt by Microsoft to provide C++ programmers with an object oriented programming interface to windows. Microsoft created the MFC library with the goal of facilitating and simplifying the process of programming for Microsoft windows. Thus, VISUAL C++ is a visual development tool that makes use of the MFC library to make the development under windows environment faster and reliable.

MFC is provided as a dynamic link library (DLL) so the application has access to the classes in MFC .To create a windows application in VC++, a good starting place is the design of the user interface. First, decide what the user should be able to do with the program and then pick a set of user interface objects accordingly. The Windows user interface has a number of standard controls, such as buttons, menus, scroll bars, and lists that are already familiar to Windows users. With this in mind, the programmer must choose a set of controls and decide how they should be arranged on screen.

The next step is to implement the code. When creating a program for any Windows platform, the programmer has two choices: C or C++. With C, the programmer codes at the level of the Windows Application Program Interface (API). This interface consists of a collection of hundreds of C functions described in the Window's API Reference books. For Window's NT, the API is typically referred to as the "Win32 API," to distinguish it from the original 16-bit API of lower-level Windows products like Windows 3.1.

Microsoft also provides a C++ library that sits on top of any of the Windows APIs and makes the programmer's job easier. Called the Microsoft Foundation Class library (MFC), this library's primary advantage is efficiency. It greatly reduces the amount of code that must be written to create a Windows program. It also provides all the advantages normally found in C++ programming, such as inheritance and encapsulation. MFC is portable, so that, for example, code created under Windows 3.1 can move to Windows NT or Windows 95 very easily. MFC is therefore the preferred method for developing Windows applications and will be used throughout the process.

When we use MFC, we write code that creates the necessary user interface controls and customizes their appearance. We also write code that responds when the user manipulates these controls. For example, if the user clicks a button, we want to have code in place that responds appropriately. It is this sort of event-handling code that will form the bulk of any application. Once *the application responds correctly to all of the available controls, it is finished.*

The creation of a Windows program is a straightforward process when using MFC. The Visual C++ application development environment is specifically *tuned to MFC, so by learning MFC and Visual C++ together we can significantly* increase our power as an application developer.

## WINDOWS VOCABULARY

The vocabulary used to talk about user interface features and software development in Windows is basic but unique.

Windows applications use several standard user controls:

- Static text labels
- Push buttons
- List boxes
- Combo boxes (a more advanced form of list)
- Radio boxes
- Check boxes
- Editable text areas (single and multi-line)
- Scroll bars

These controls can be created either in code or through a "resource editor" that can create dialogs and the controls inside them. Windows supports several types of application windows.

A typical application will live inside a "frame window". A frame window is a fully featured main window that the user can re-size, minimize, and maximize to fill the screen, and so on. Windows also supports two types of dialog boxes: modal and modeless. A modal dialog box, once on the screen, blocks input to the rest of the application until it is answered. A modeless dialog box can appear at the same time as the application and seems to "float above" it to keep from being overlaid. Most simple Windows applications use a Single Document Interface, or SDI, frame. The Clock, PIF editor, and Notepad are examples of SDI applications. Windows also provides an organizing scheme called the Multiple Document Interface or MDI for more complicated applications.

The MDI system allows the user to view multiple documents at the same time within a single instance of an application.

For example, a text editor might allow the user to open multiple files simultaneously. When implemented with MDI, the application presents a large application window that can hold multiple sub-windows, each containing a document. The single main menu is held by the main application window and it applies to the top-most window held within the MDI frame. Individual windows can be iconified or expanded as desired within the MDI frame, or the entire MDI frame can be minimized into a single icon on the desktop. The MDI interface gives the impression of a second desktop out on the desktop, and it goes a long way towards organizing and removing window clutter.

Each application that we create will use its own unique set of controls, its own menu structure, and its own dialog boxes. A great deal of the effort that goes into creating any good application interface lies in the choice and organization of these interface objects. Visual C++, along with its resource editors, makes the creation and customization of these interface objects extremely easy.

## EVENT-DRIVEN SOFTWARE AND VOCABULARY

All window-based GUIs contain the same basic elements and all operate in the same way. On screen the user sees a group of windows, each of which contains controls, icons, objects and such that are manipulated with the mouse or the keyboard. The interface objects seen by the user are the same from system to system: push buttons, scroll bars, icons, dialog boxes, pull down menus, etc. These interface objects all work the same way, although some have minor differences in their "look and feel." For example, scroll bars look slightly different as you move from Windows to the Mac to Motif, but they all do the same thing.

From a programmer's standpoint, the systems are all similar in concept, although they differ radically in their specifics. To create a GUI program, the programmer first puts all of the needed user interface controls into a window. For example, if the programmer is trying to create a simple program such as a Fahrenheit to Celsius converter, then the programmer selects user interface objects appropriate to the task and displays them on screen. In this example, the programmer might let the user enter a temperature in an editable text area, display the converted temperature in another un-editable text area, and let the user exit the program by clicking on a push-button labeled "quit".

As the user manipulates the application's controls, the program must respond appropriately. The responses are determined by the user's actions on the different controls using the mouse and the keyboard. Each user interface object on the screen will respond to events differently.

For example, if the user clicks the Quit button, the button must update the screen appropriately, highlighting itself as necessary. Then the program must respond by quitting. Normally the button manages its appearance itself, and the program in some way receives a message from the button that says, "The quit button was pressed. Do something about it." The program responds by exiting.

Windows follows this same general pattern. In a typical application you will create a main window and place inside it different user interface controls. These controls are often referred to as child windows-each control is like a smaller and more specialized sub-window inside the main application window. As the application programmer, we manipulate the controls by sending messages via function calls, and they respond to user actions by sending messages back to our code.

In an event-driven interface, the application paints several (or many) user interface objects such as buttons, text areas, and menus onto the screen. Now the application waits-typically in a piece of code called an event loop-for the user to do something.

The user can do anything to any of the objects on screen using either the mouse or the keyboard. The user might click one of the buttons, for example. The mouse click is called an event. Event driven systems define events for user actions such as mouse clicks and keystrokes, as well as for system activities such as screen updating.

At the lowest level of abstraction, we have to respond to each event in a fair amount of detail. This is the case when we are writing normal C code directly to the API. In such a scenario, we receive the mouse-click event in some sort of structure. Code in our event loop looks at different fields in the structure, determines which user interface object was affected, perhaps highlights the object in some way to give the user visual feedback, and then performs the appropriate action for that object and event. When there are many objects on the screen the application becomes very large. It can take quite a bit of code simply to figure out which object was clicked and what to do about it.

Fortunately, we can work at a much higher level of abstraction. In MFC, almost all these low-level implementation details are handled for us. If we want to place a user interface object on the screen, we create it with two lines of code. If the user clicks on a button, the button does everything needed to update its appearance on the screen and then calls a pre-arranged function in your program. This function contains the code that implements the appropriate action for the button. MFC handles all the details for us. We create the button and tell it about a specific handler function, and it calls our function when the user presses it.

# CHAPTER 3

# SYSTEM DESIGN

## 3.1 Basic Design Concepts

### FTP SERVER

FTP server allows Internet users access to files and directories. It is one of the oldest of the Internet protocols, which is broadly supported by software vendors. Many types of FTP servers are available at most sites which can be easily accessed through the client software, web-browsers, or via e-mail remote access.

FTP server software is less common although server software is also available from many sources. An FTP server makes specified files available and may even allow uploading. FTP requires a login to the server just as most machines require a login: username and password for an account that exists on the server. Access to the ftp server can be restricted by defining users and assigning passwords.

Here a specific account for the user is created so that the user is able to log in using the ftp client, and then move whatever is necessary. The server can be configured to define the server's time-out value and specify how many users can be connected to the FTP server simultaneously.

FTP does some complex processing. This complexity is required because FTP works across different operating systems, each of which organize their file systems and file storage uniquely. FTP must be able to translate from one to another in a transparent way.

Only authorized users may have access to the FTP server. For example, an anonymous user may define. This is generally the username invoked when accessing any public FTP site over the Internet. Once an anonymous user has been defined, connection attempts for anonymous are accepted. The use of the anonymous login identifier is referred to as anonymous FTP.

A user may have permission to see all local drives, directories, and files or be restricted to a certain directory and its subdirectories. For example, users with read-only permissions can retrieve files, but not send them to the FTP server. In addition, read-only users cannot create or rename directories or files. With many FTP server applications, a log is created which contains information about client operation, such as login, logout, and files accessed activity.

## CONFIGURING THE SERVER

To configure the server and to run it securely

1. After the server is installed, a NT user account is to be created.

2. This new user is given user account permissions limited to what the server requires. The permissions are then gradually increased wherever this is necessary.

## SERVER SECURITY

Any time the server is run the computer is exposed to outside users. Running a server as a service can increase the severity of this risk because a service has a potential to expose files and programs on the computer and network to outside users.

The following is the list of things to be kept in mind while configuring the server security:

Storing the login name and password may be convenient but not secure. The password may be available to anyone who uses the server machine.

## CONNECTING TO A SERVER

Connect to the server from the administrator Interface. Generally, if it is not already connected, the connect dialog box will open automatically.

To connect to a local server

In the administrator Interface go to FILE>CONNECT TO FTP SERVER.
The connect dialog box appears.
Enter the user name if not entered before
Enter the password.
Select your Local Host.
Click Connect.

## COMMON CONNECTION PROBLEMS

If there are problems in connecting, make sure that:

Check whether the username and password are correct. (As these are case sensitive).

Check whether host (IP address) and port are correct.

Check whether the service is running.

Check if the network connection is good.

FTP server can display messages in four situations:

Successful connection

Login

Maximum number of connections

Exit

## CONNECTION MESSAGE

The connection message appears when a user first connects, but before a user logs on.

## LOGIN MESSAGE

Login messages may be applied at the user or user option level. Users will automatically inherit the message applied to the user option level.

## MAXIMUM CONNECTIONS MESSAGE

A site can be configured to allow a specified number of maximum simultaneous connections. If this option is chosen then a message can be specified when the maximum number of connections are exceeded.

## EXIT MESSAGE

The server can send an exit message when the client closes the session. *This can be done by using a Quit or a Close command.*

## 3.2 DFD NOTATIONS

### 3.2.1 Data Flow Diagram

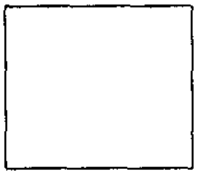
A Data Flow Diagram (DFD) is a graphical technique that depicts information flow and the transformation that is applied as data moves from input to output. The data flow diagram may be used to represent a system at any level of abstraction. So the DFD may be partitioned into levels that represent increasing information flow. Data Flow Design is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources/destinations.

The purpose of data flow diagrams is to provide a semantic bridge between users and systems developers. The diagrams are:

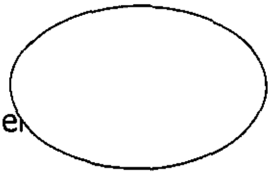
- Graphical, eliminating thousands of words;
- Logical representations, modeling WHAT a system does, rather than physical models showing HOW it does it;
- Hierarchical, showing systems at any level of detail; and
- Jargon less, allowing user understanding and reviewing.

The goal of data flow design is to have a commonly understood model of a system. The diagrams are the basis of structured systems analysis. Data flow diagrams are supported by other techniques of structured systems analysis such as data structure diagrams, data dictionaries, and procedure-representing techniques such as decision tables, decision trees, and structured English.

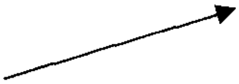
The basic symbols used to draw a DFD are the following.



Represents the information that resided outside the bounds of the system to be developed.

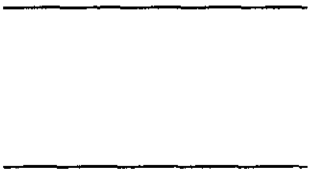


A transformation of information that resided within the bounds of the system to be model



A data object, the arrowhead indicates the direction of

Information flow



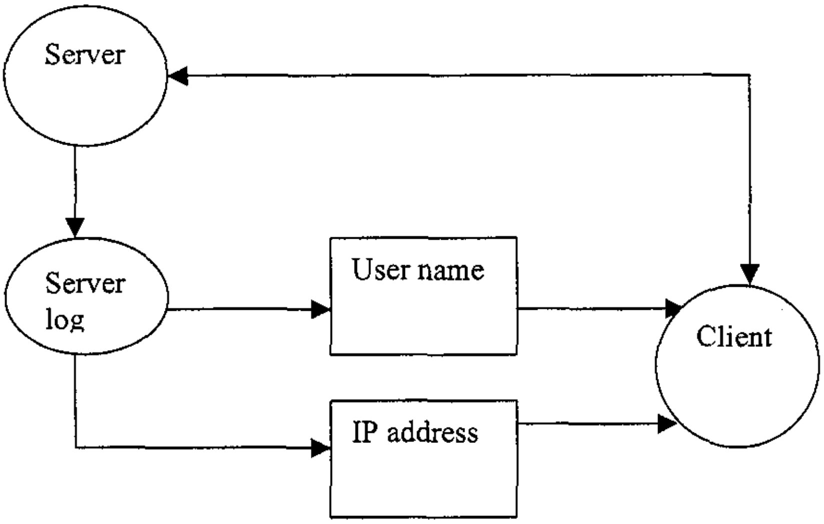
A repository of data that is stored for use by one or more

processes. A data is represented inside this symbol.

Each of the bubbles that represent a process or transform may be refined or layered to depict information in more detail. Each such refinement is called a level. The level 0 DFD represent the fundamental system model, which can be refined to more detail.

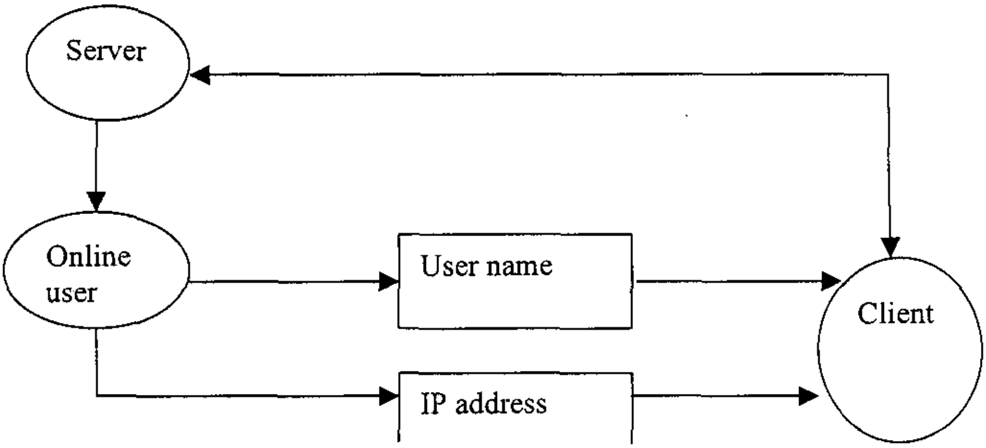
## Server Log

Figure 3.2.1



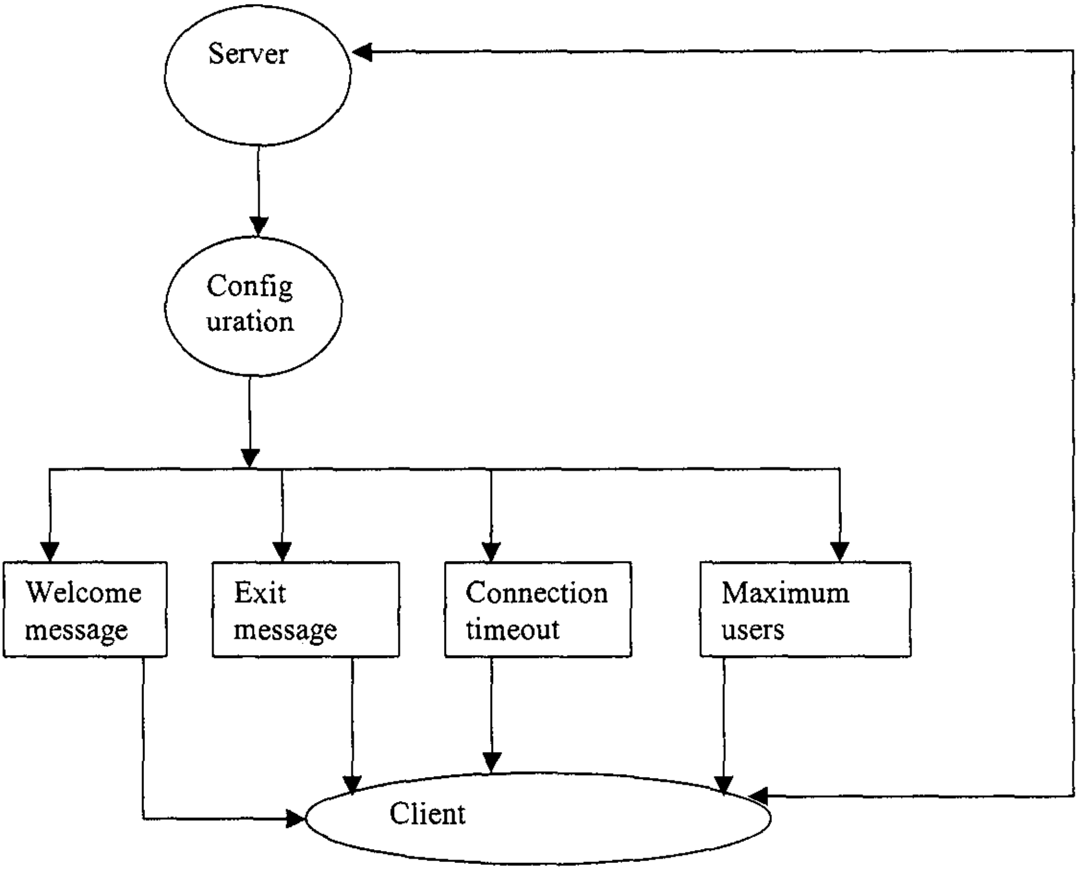
## Online User

Figure 3.2.2



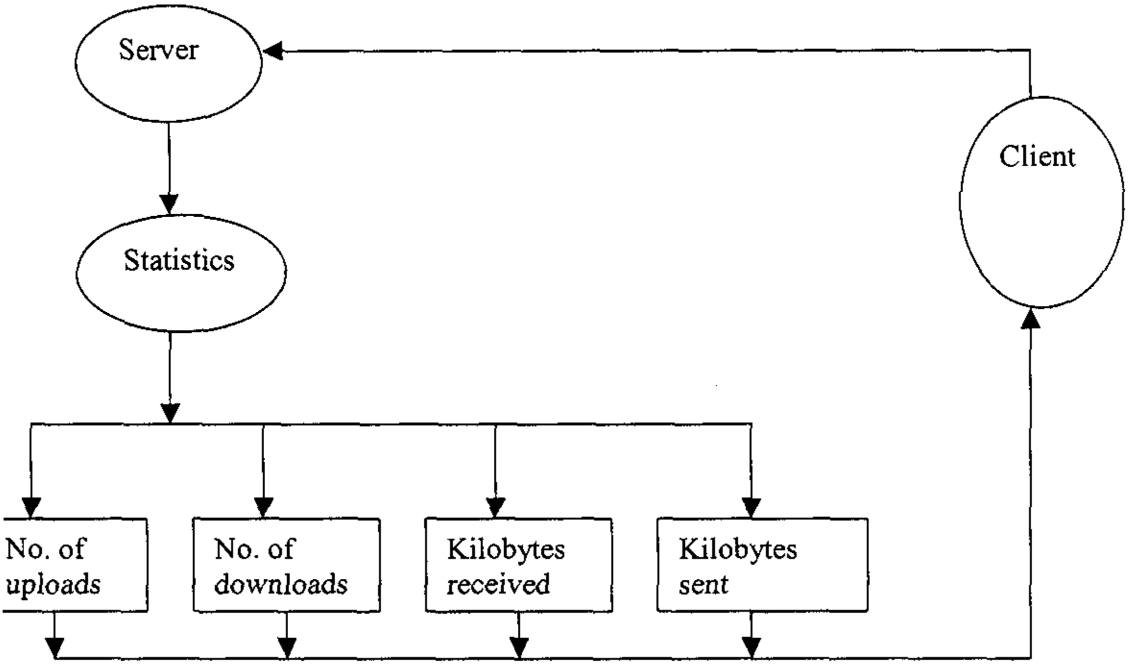
## Configuration

Figure 3.2.3



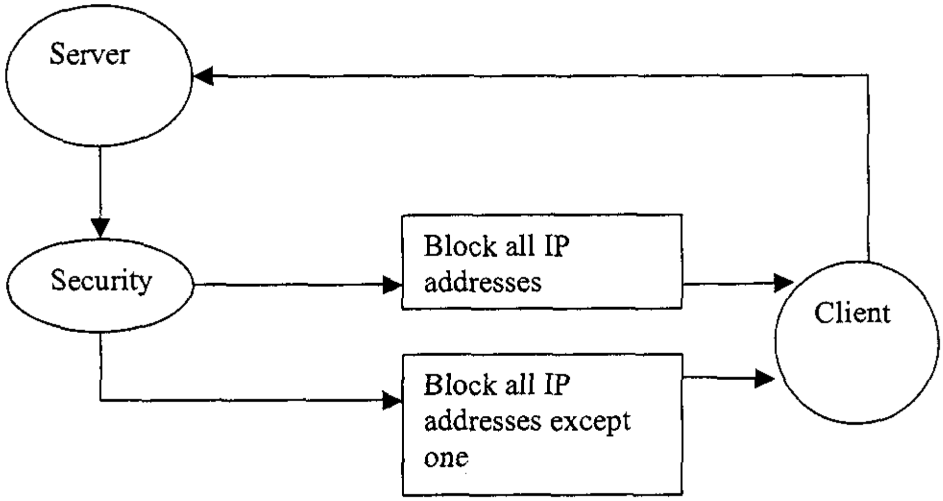
### Statistics

Figure 3.2.4



## Security

Figure 3.2.5



# CHAPTER 4

# *TESTING AND IMPLEMENTATION*

## 4.1 TESTING PROCESS

The development of the software system involves a series of production activities where opportunities of infections due to human fallibility are enormous. Hence software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The success or failure of the software testing deeply lies in the testing strategies followed and the test cases taken in to consideration. The software has been tested using the following strategies.

### 4.1.1 Unit Testing

The unit testing is done to all independent paths by ensuring that all statements in each program are executed at least once. Data is provided in such a manner that for each data, the conditional statements in the program take a new path and hence all paths are executed. Erroneous data is provided in order to check if the error message and the error routines are executing properly.

### 4.1.2 Validation Testing

Validation can be defined in many ways, but a simple definition is that validation that succeeds when the system functions in a manner that can be reasonably expected by the customer. This strategy was used to identify whether the initial proposed system is working as it was intended. Normally, a validation testing is done using a series of black box testing, which demonstrates conformity with requirements.

### 4.1.3 Integration Testing

Integration Testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover error associated with interfacing. The objective is to take unit-tested modules and build the program structure that has been dictated by the design.

Top-down integration is an incremental approach to the construction of the program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module.

Bottom-up integration begins construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from bottom-up, processing required for modules subordinate to a given level is always available and the need for the stubs is eliminated.

### 4.1.4 Alpha Testing

A third person who just has the knowledge and the working capacity of the system conducts the alpha test at the developer's site. The developer 'looks over the shoulder' of the user and records the errors and usage problems. The user in turn gives general discomforts, which may be mended to make the system little better in a way of efficiency and user-friendly.

### 4.1.5 User Acceptance Testing

Acceptance testing involves planning and execution of the functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements. It is not unusual for two sets of acceptance tests to be run those developed by the Quality Assurance group and those developed by the customer.

## 4.2 IMPLEMENTATION

Implementation includes all those activities that take place to creating a domain name, uploading the files to the website and other administrative activities. A proper implementation is essential to provide a reliable system to meet the requirements of the customers and the company. An improper installation may affect the success of the website and the administrative system.

### 4.2.1 IMPLEMENTATION PROCEDURE

**Implementation Plan**

The implementation plan includes a description of all the activities that must occur to launch the website and to put it into operation. It identifies the personnel responsible for the activities and prepares a time chart for launching the website. The implementation plan consists of the following steps.

- List all files that have to be uploaded.
- Identify all data required to build new files during the upload.
- List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current files, errors in data translation, missing data etc.

The implementation will start after all the modules of the website are integrated and tested together.

## 4.3 MAINTENANCE

Maintenance is the enigma of the system development. It holds the software industry capture, trying up programming resources. Analysts and programmers spend far more time maintaining programs than they do writing them.

The first maintenance activity occurs since it is to assume that software testing will uncover all the errors in a large software system. The process of the coding, the diagnosis and the correction of one or more errors are called corrective maintenance.

The second activity that contributes to the definition of maintenance occurs since rapid change is encountered in every aspect of computing. Therefore, adaptive maintenance modifies software to properly interface with the changing environment.

## 4.4 FUTURE ENHANCEMENTS

Enhancement is always an important and necessary activity in the life of the software product developed. Any system developed should always provide room to accept any changes and further inclusions.

This may occur so as to suit the new trend and additional features required by the customer. The future enhancements are to be carried out in such a way that the application moves in the positive direction or it adds more functionality, acceptability and usability to the existing system.

The future enhancements done to any application are endless and can suit the needs of the changing environment. Hence this multi portal website is not an exception to it. There are many enhancements possible.

- Project can be extended for WAN and wireless LANs also
- It can be extended to include video conferencing facilities
- It can be extended to include some security features like authentication etc
- It can be extended to run over the internet
- Text-based chat windows and file-transfer features can also be included

# CHAPTER 5

# CONCLUSIONS

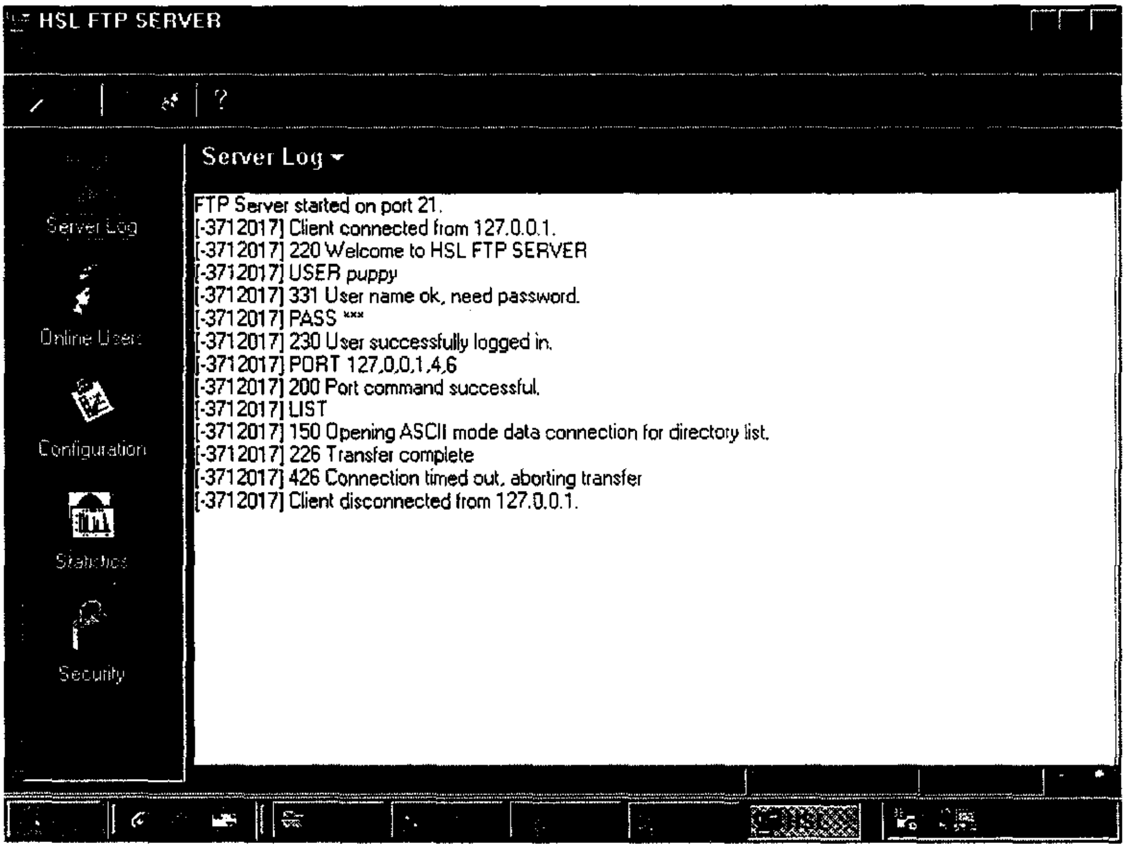
The project was successfully designed and developed as per the Company requirements and specifications.

The system was thoroughly tested with extensive set of files and found to transfer correctly. Security is maintained and severed validations and checks have been performed during data transferring.

*It also provides a user-friendly environment and reduces user difficulties.* The main advantage is that the user can utilize the services offered by this system with ease.

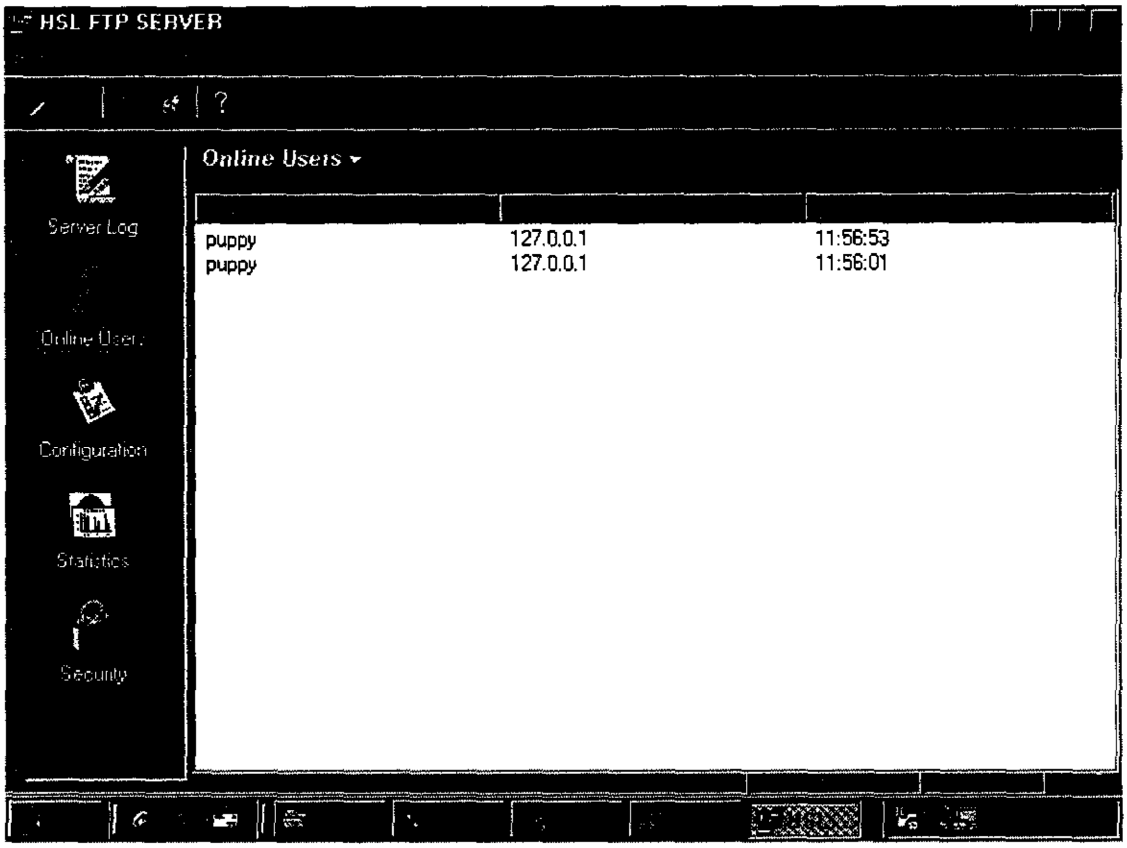
# APPENDICES

## SERVER LOG



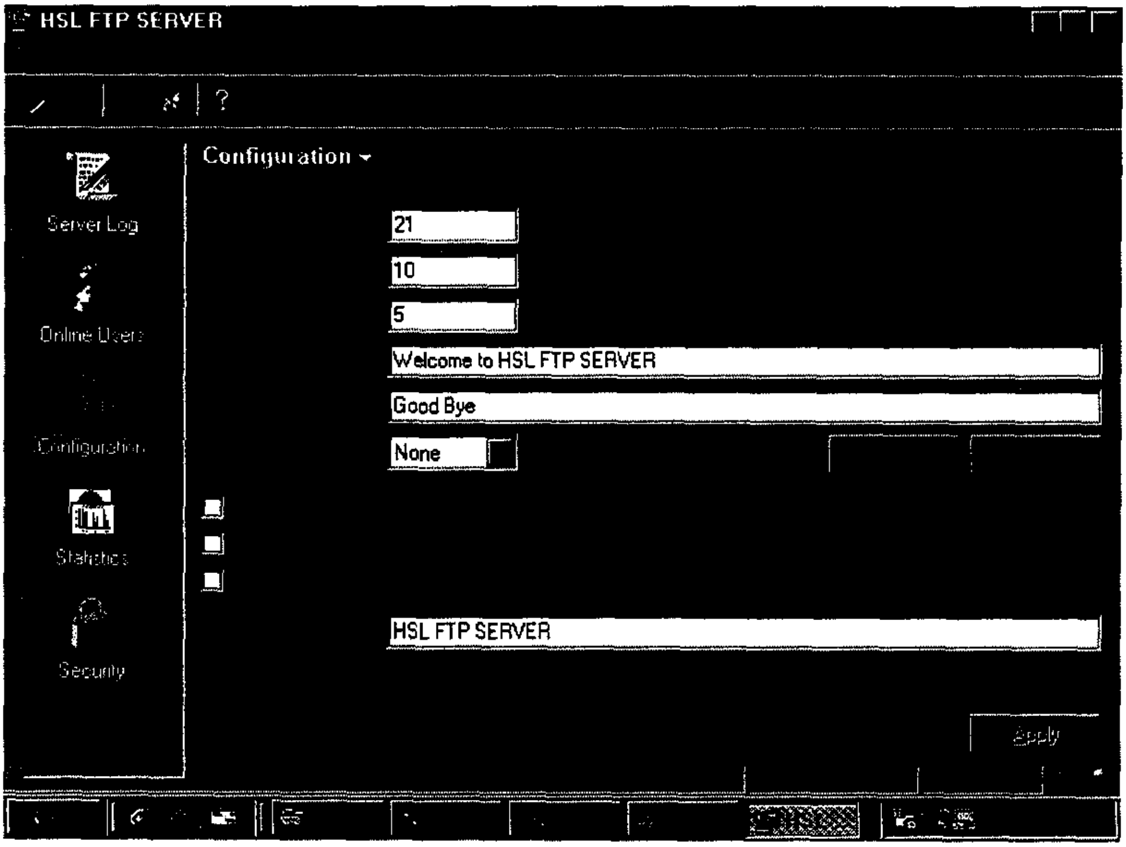
```
FTP Server started on port 21.
[-3712017] Client connected from 127.0.0.1.
[-3712017] 220 Welcome to HSL FTP SERVER
[-3712017] USER puppy
[-3712017] 331 User name ok, need password.
[-3712017] PASS ***
[-3712017] 230 User successfully logged in.
[-3712017] PORT 127,0,0,1,4,6
[-3712017] 200 Port command successful.
[-3712017] LIST
[-3712017] 150 Opening ASCII mode data connection for directory list.
[-3712017] 226 Transfer complete
[-3712017] 426 Connection timed out, aborting transfer
[-3712017] Client disconnected from 127.0.0.1.
```

## ONLINE USER



```
puppy          127.0.0.1          11:56:53
puppy          127.0.0.1          11:56:01
```

## CONFIGURATION



21

10

5

Welcome to HSL FTP SERVER

Good Bye

None

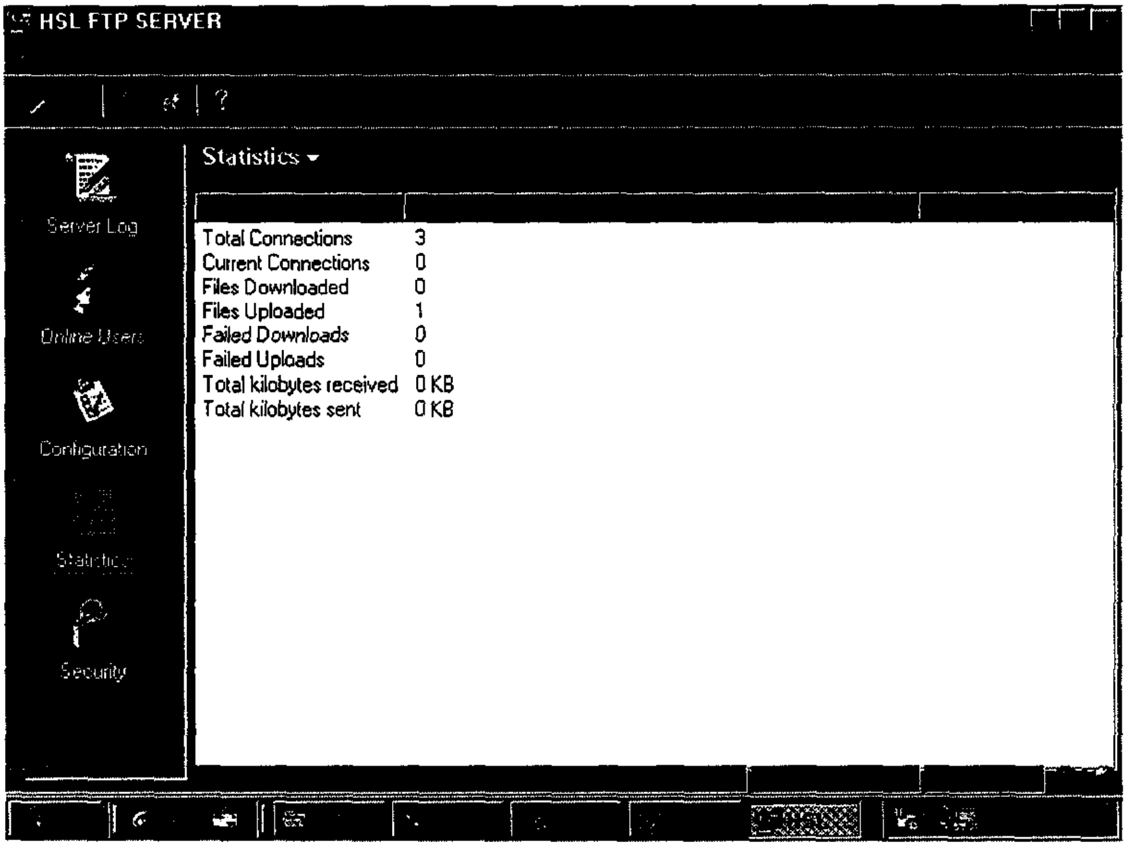
□

□

□

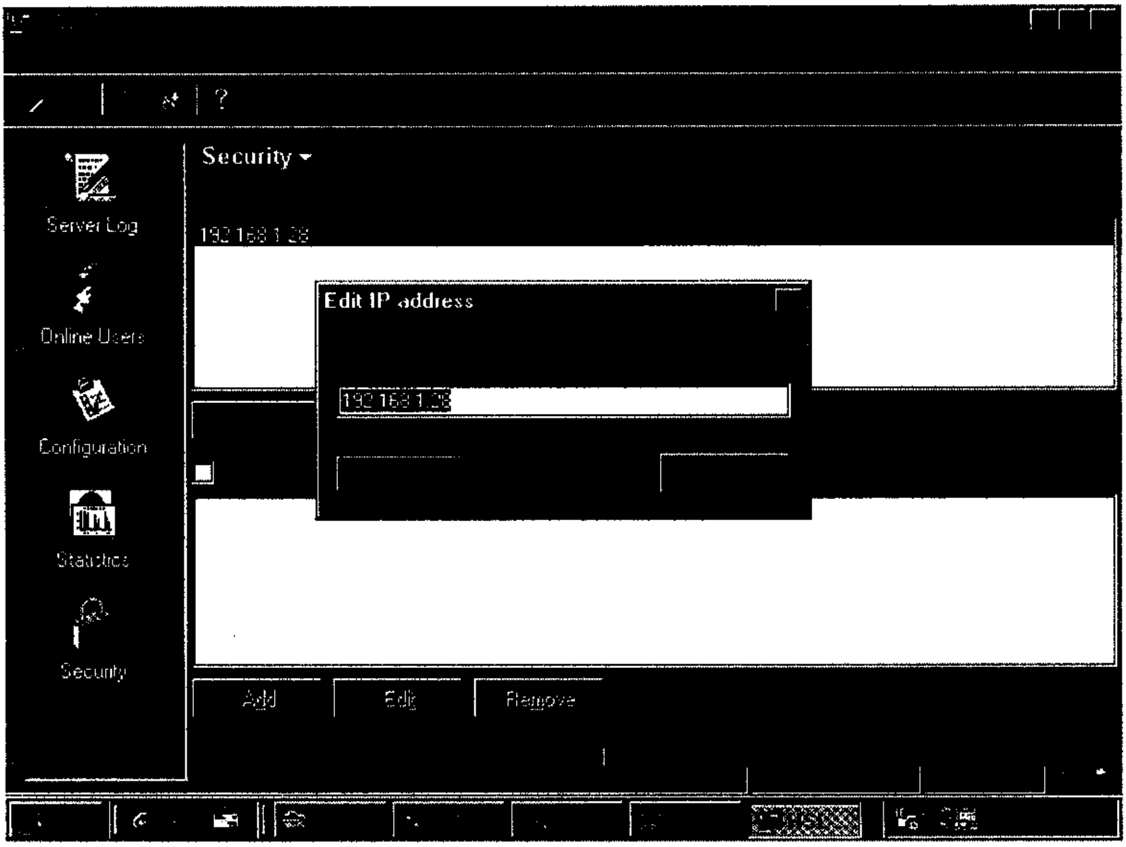
HSL FTP SERVER

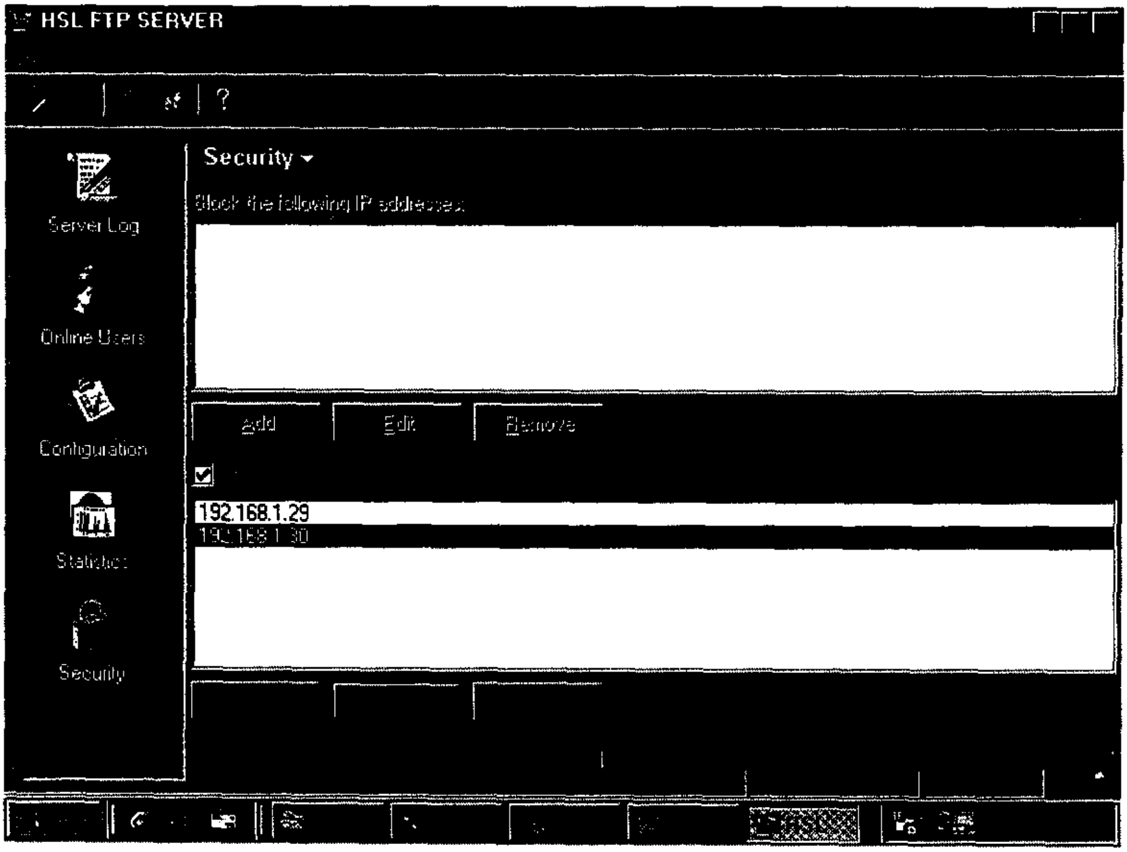
## STATISTICS



```
Total Connections        3
Current Connections      0
Files Downloaded         0
Files Uploaded           1
Failed Downloads         0
Failed Uploads           0
Total kilobytes received 0 KB
Total kilobytes sent     0 KB
```

## SECURITY


Security
Server Log
Online Users
Configuration
Statistics
Security
192.168.1.28
Edit IP address
192.168.1.28
Add
Edit
Remove



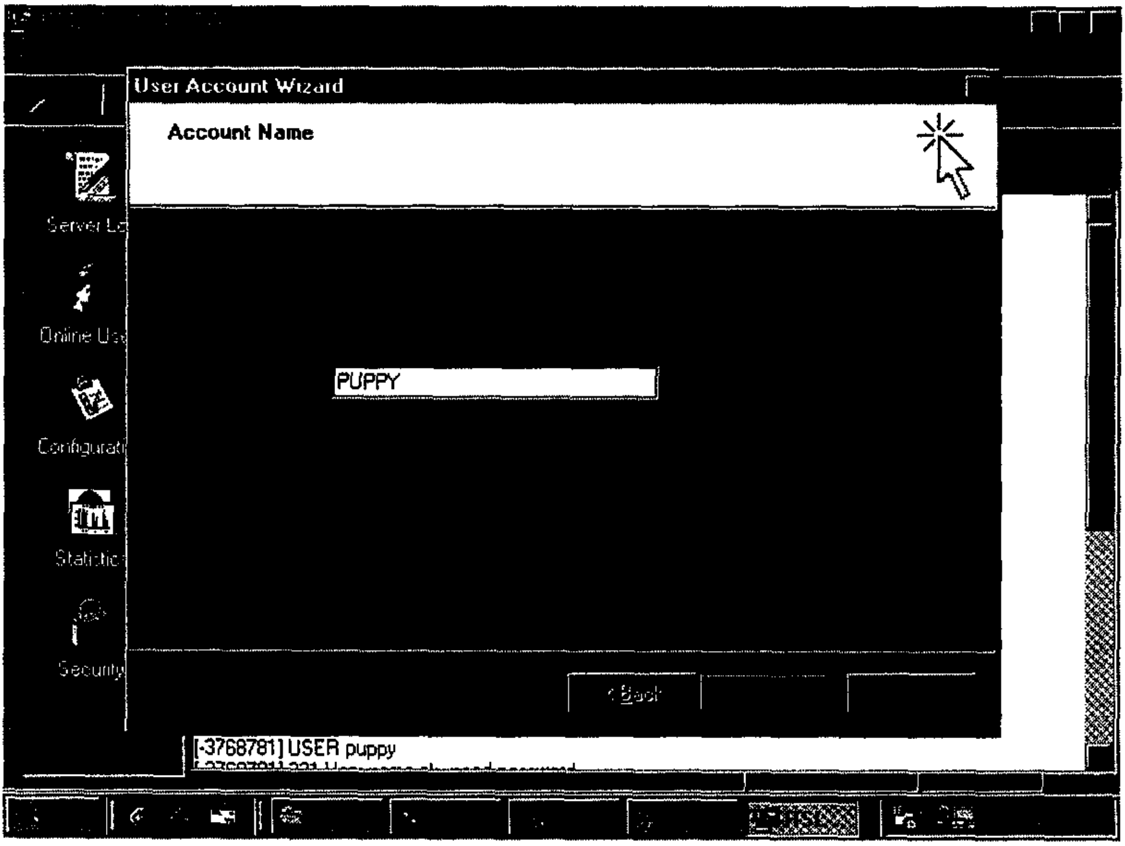
Block the following IP addresses:

☑

192.168.1.29
192.168.1.30

## USER ACCOUNT WIZARD

### ACCOUNT NAME

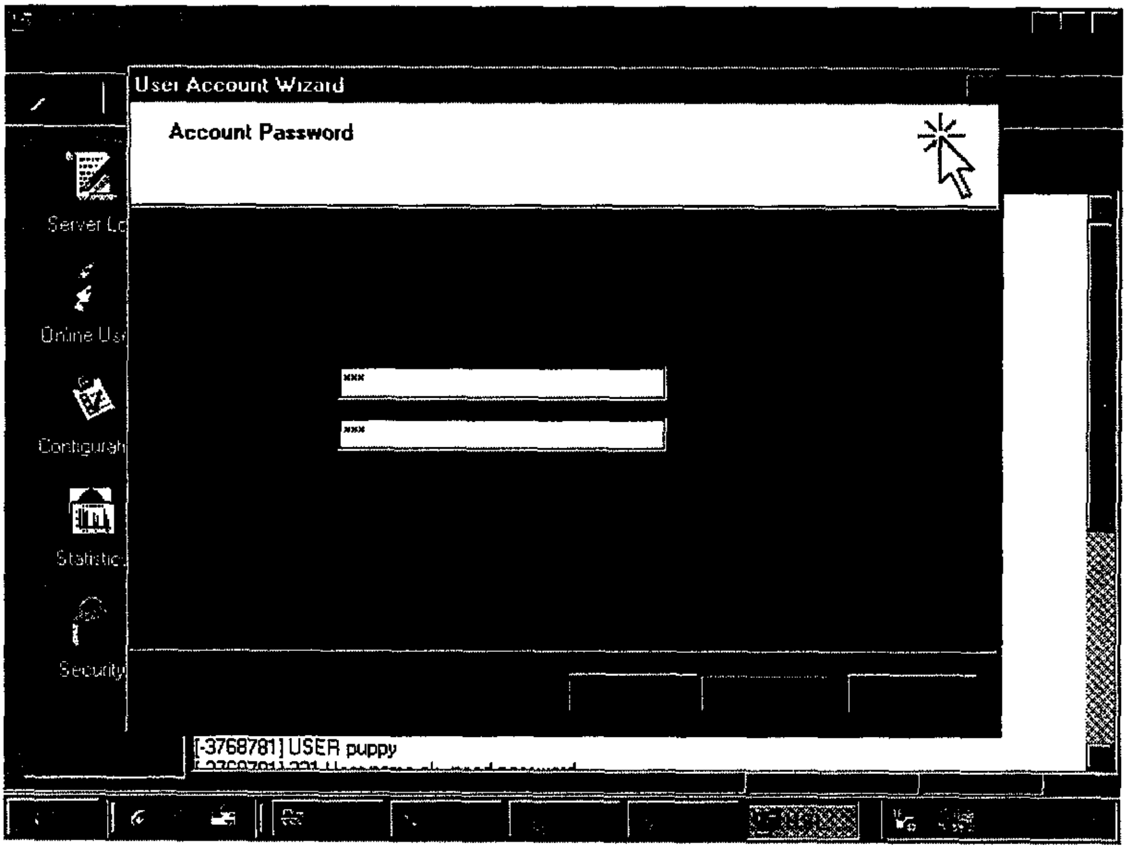


PUPPY

[-3768781] USER puppy

## ACCOUNT PASSWORD

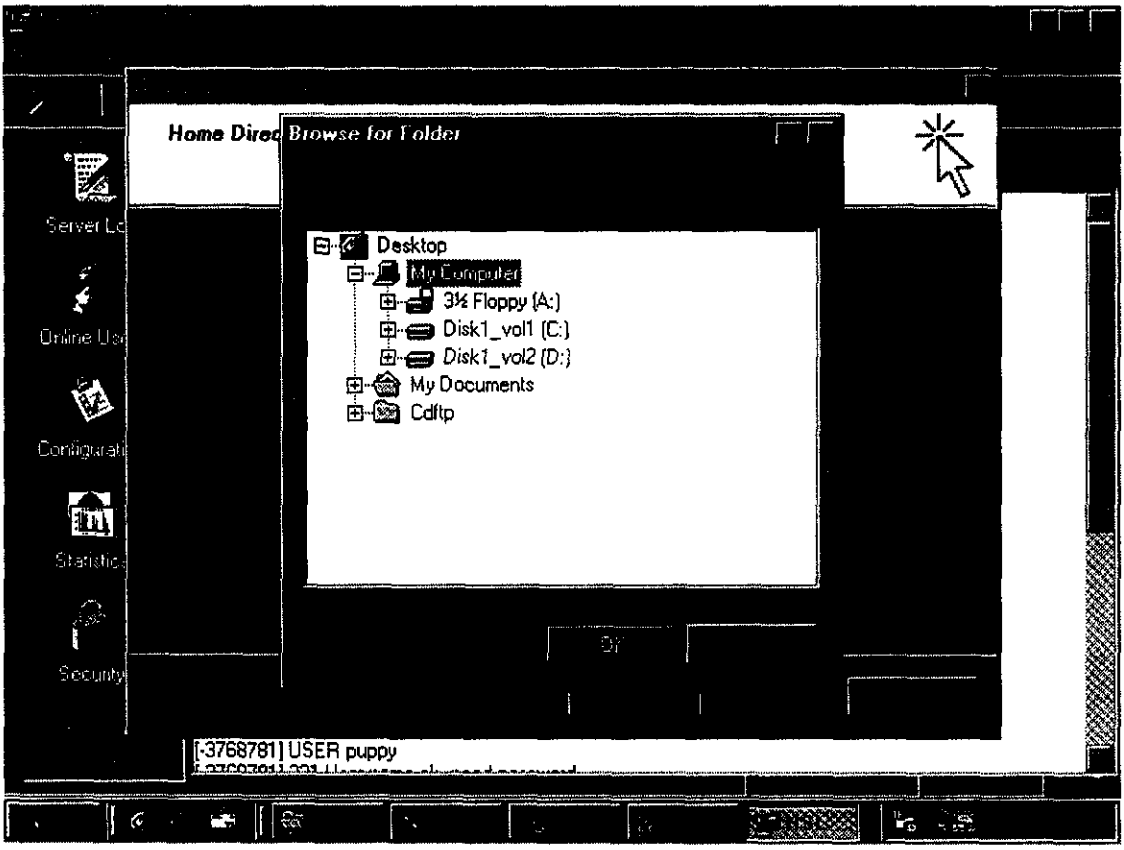
```
[-3768781] USER puppy
```



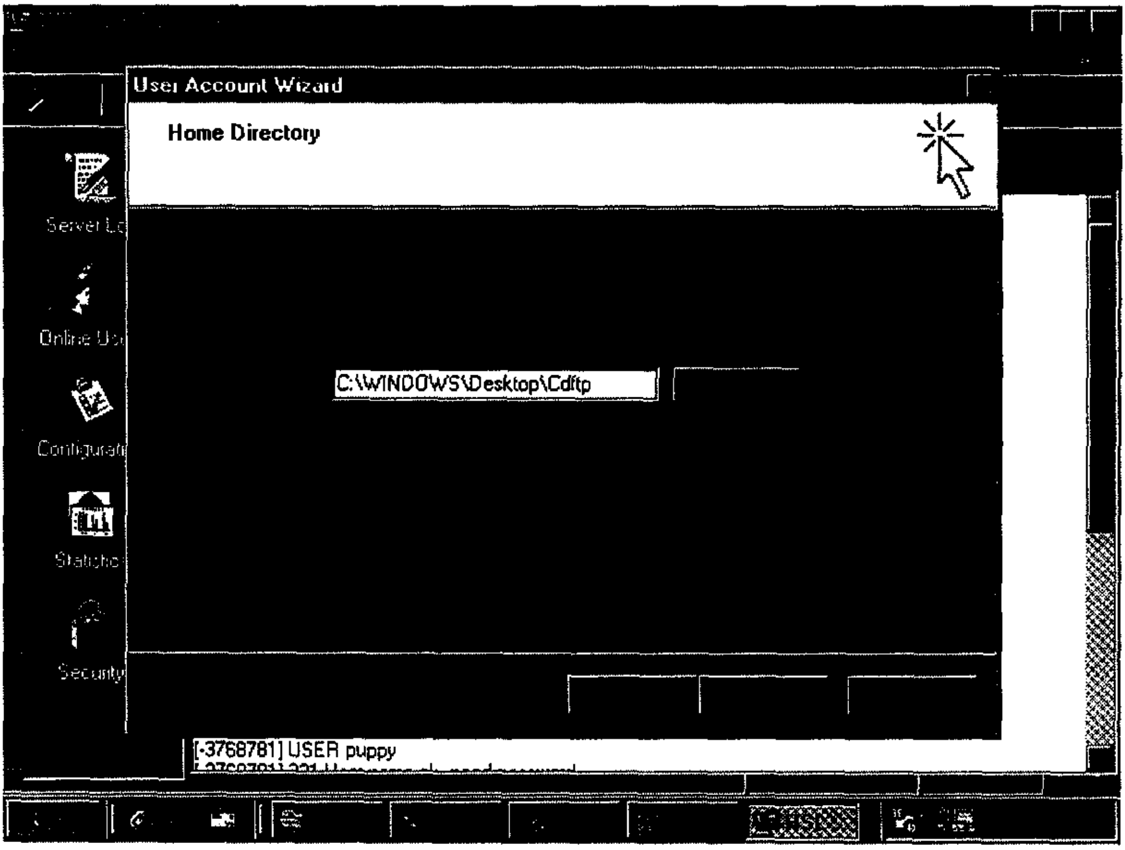
```
***
***
```

```
[-3768781] USER puppy
```

## HOME DIRECTORY



```
[-3768781] USER puppy
```



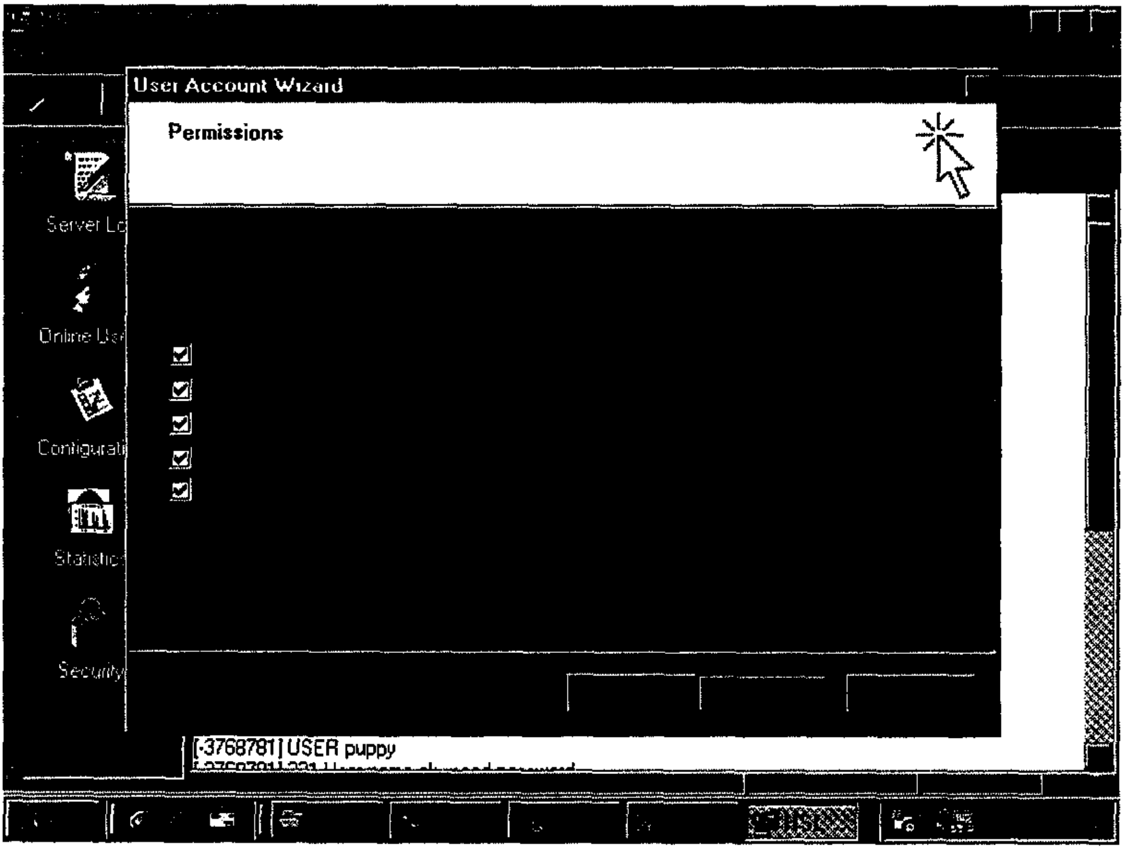
User Account Wizard

Home Directory

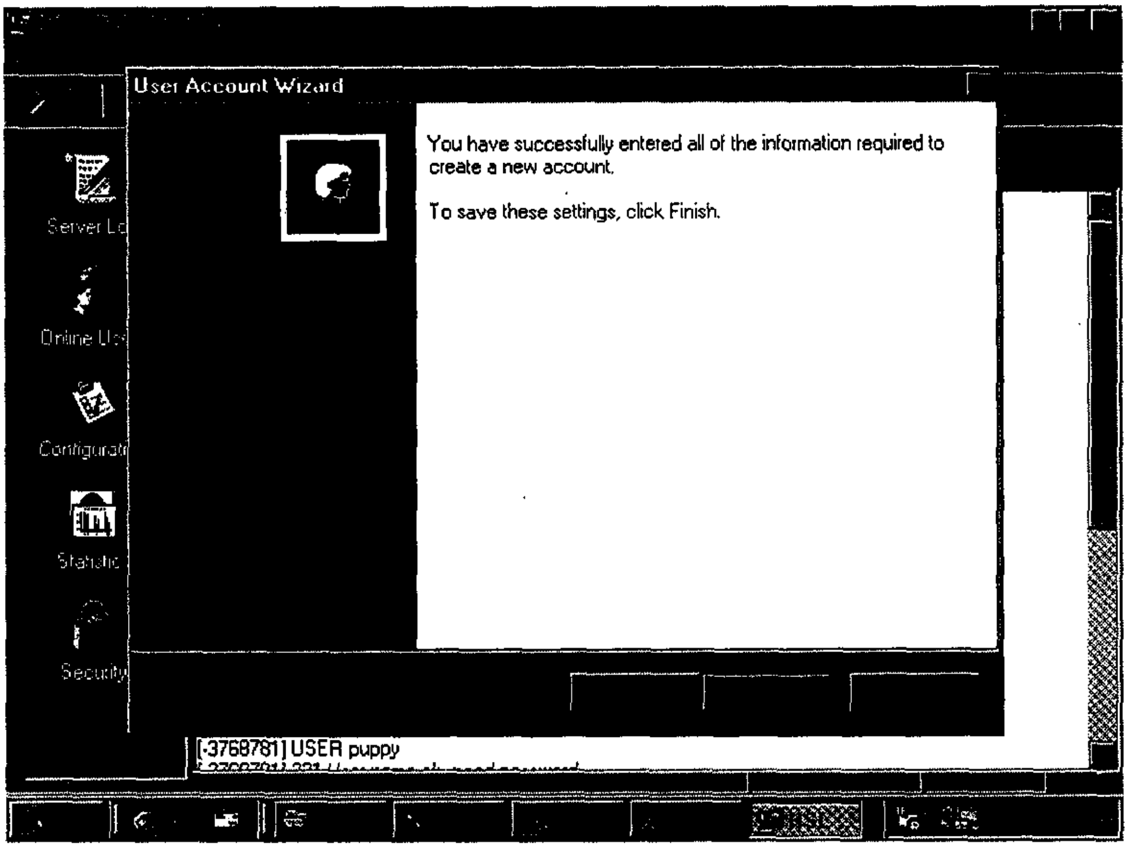
C:\WINDOWS\Desktop\Cdftp

```
[-3768781] USER puppy
```

## PERMISSIONS



```
[-3768781] USER puppy
```

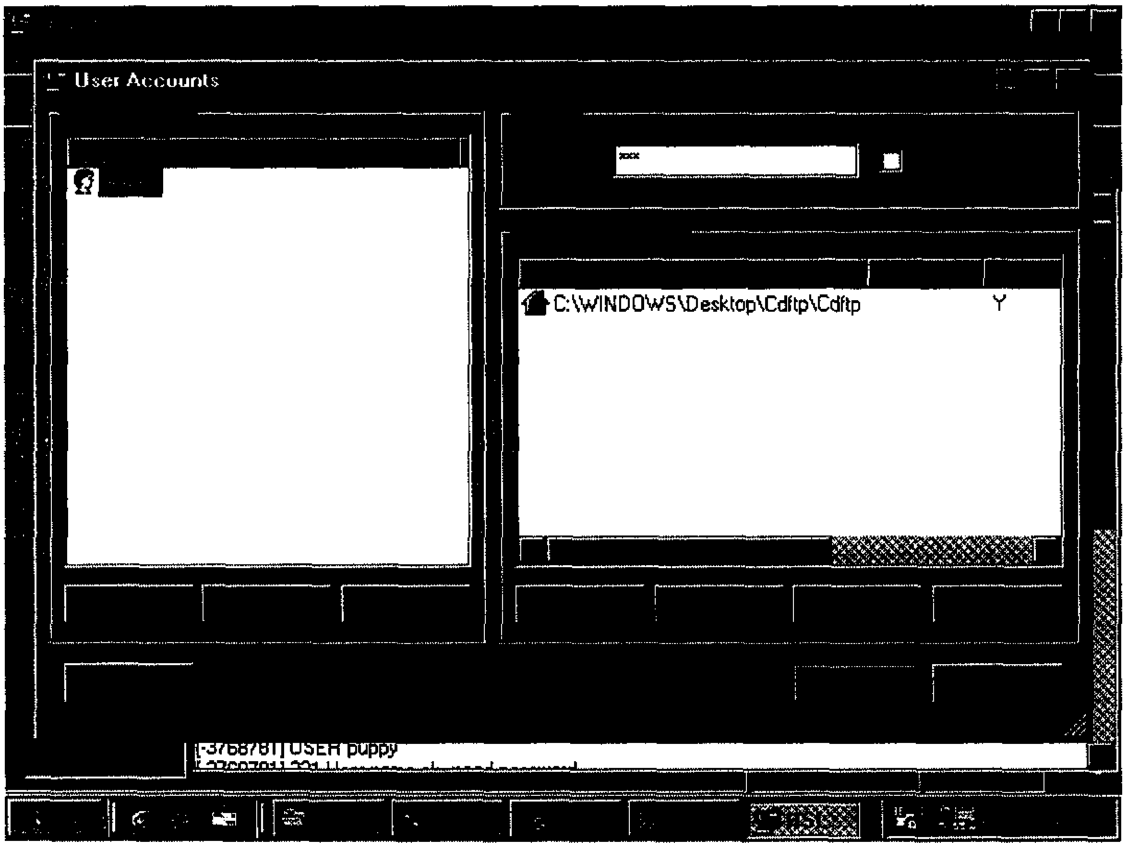


You have successfully entered all of the information required to create a new account.

To save these settings, click Finish.

```
[-3768781] USER puppy
```

## USER ACCOUNT



xxx

C:\WINDOWS\Desktop\Cdftp\Cdftp Y

```
[-3768781] USER puppy
```

# FTP USER COMMANDS

| FTP COMMAND | DESCRIPTION |
|---|---|
| | |
| Ascii | Switch to ASCII transfer mode |
| Binary | Switch to binary transfer maode |
| Cd | Change directory on the server |
| Close | Terminate the connection |
| Del | Delete a file on the server |
| Dir | Display the server directory |
| Get | Fetch a file from the server |
| Hash | Display a pound character for each block transmitted |
| Help | Display help |
| Lcd | Change directory on the client |
| Mget | Fetch several files from the server |
| Mput | Send several files to the server |
| Open | Connect to a server |
| Put | Send a file to the server |
| Pwd | Display the current server directory |
| Quote | Supply an FTP command directly |
| Quit | Terminate the FTP session |

# REFERENCES

## BOOKS:

- David J. Kruglinski, George Shepard and Scot Wingo "Programming Microsoft Visual C++"
- Davis Chapman " Teach yourself Visual C++ 6 in 21 days"
- Visual C++ MFC Reference Library
- Jeffery A. Hoffer, Joey F. George and Joseph S. Valacich "Modern System Analysis & Design"
- Bob Hughes and Mike Kotterell "Software Project Management"

## WEBSITES:

RFC-1889 RTP: A Transport Protocol for Real-Time Applications

http://www.cs.columbia.edu/~hgs/rtp/