

# PORTAL DESIGN AND IMPLEMENTATION

## A THESIS

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR  
THE AWARD OF THE DEGREE OF

### MASTER OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

TO

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY  
(DEEMED UNIVERSITY)  
PATIALA-147 001



SUPERVISOR  
**DR.RENU VIG**  
ASSISTANT PROFESSOR  
DEPARTMENT OF COMPUTER SCIENCE  
T.T.T.I. CHANDIGARH

SUBMITTED BY  
**M.S.PATIL**  
REGISTRATION NO.: ME(R)-136/99(1)



DEPARTMENT OF COMPUTER SCIENCE  
TECHNICAL TEACHERS' TRAINING INSTITUTE  
SECTOR-26, CHANDIGARH-160019.  
2000-2001

## CERTIFICATE

This is to certify that the thesis report entitled "PORTAL DESIGN AND IMPLEMENTATION" submitted by Mr. PATIL MANOJ SAKHARAM in the partial fulfillment of the requirement, for the award of Master of Engineering (Computer Science and Engineering) Degree of THAPER INSTITUTE OF ENGINEERING AND TECHNOLOGY, PATIALA, is a record of student's own study carried under my supervision and guidance.

Supervisor

Head of Department



(Dr. Renu Vig)  
Assistant Professor,  
Computer Science Department  
T.T.T.I., CHANDIGARH.



(Prof. V.P. Puri)  
Professor and Head,  
Computer Science Department  
T.T.T.I., CHANDIGARH.

Place: CHANDIGARH.  
Date : 31<sup>st</sup> January 2001.

## ACKNOWLEDGEMENT

I take opportunity to express thanks to those who helped me directly and indirectly during my thesis work.

First of all, I would like to express deep gratitude and sincere thanks to my guide Dr.Renu Vig (Assistant Professor and coordinator, Department of Computer Science, T.T.T.I.,Chandigarh) and Mr.Sandeep Vig (Spectra Technologies, Chandigarh) young, enthusiastic businessman, for their constant encouragement, guidance, and valuable help.

I am thankful to Principal Shri.S.Krishnamurty and Head of Department Prof.V.P.Puri for their encouragement and providing facilities to carry out the work.

I am also very much thankful to Mr.Ramakrishna (faculty) and Mr.Siddharth (Technical Engineer) for their time-to-time help in this work. I am thankful to other staff members Mrs.Sangeeta Gupta, Mrs.M.V.Laxmi, Mr.Negi, and others who helped me directly or indirectly.

Thanks are also due to G.P.S.Raghava, Scientist and Head, Institute of Microbial Technology, Sector-39A, Chandigarh, and Mr.Babu T., Scientific Officer/Engineer 'SB', National Informatics Center, Ministry of Information Technology, Sector 8-C, Chandigarh-160018 for their help and good wishes.

I can not forget the encouragement and constant moral support not only for this work but throughout 2 years duration of my M.E. program from Principal R.P.PATIL, College of Engineering, Malegaon(Bk), Tal-Baramati, Dist-Pune. And able management of my sponsoring organization, who foresee the growth of the organization along with staff members.

Last but not least, I am very much thankful to my beloved son - Sagar for bearing to stay apart from me for much time during this work, my wife for bearing with me and my parents, brother and sister for their affection.

  
(M.S.PATIL)

## ABSTRACT

### PORTAL DESIGN & IMPLEMENTATION

"Portal" is a collection of different Internet technologies that provide different types of services. The most commonly used services are chat, e-mail, newsgroup, discussion-group, bulletin board, downloads, notice board, services provided by organization, etc. These services are provided by different organizations at different level. The services can be categorized as private and public.

The purpose of this project is to design a simple portal for the host institute viz. Technical Teachers' Training Institute, Chandigarh and implement some of the services. The services may not exactly match with the same type of services as found on Internet portals. However, these will be customized to institute needs. Also, some of the services such as www will be available on Internet for public access but other services such as e-mail will be available only for institute staff members.

The services proposed to develop are:

1. Public access : www, newsletter
2. Private (members): chat, e-mail, notice board, ftp

The programming will be carried out using the Java language. Also, it involves server setup, DNS installation and registration. The server will be setup using either Microsoft Windows NT 4.0 server or Red Hat Linux server. For this purpose, the project thesis work will also include server installation, configuration and required administration.

## CONTENTS

Chapter No.	Title	Page No.
1	Introduction	1
1.1	Aim	3
1.2	Objectives	3
2	Survey – Portal Components	4
2.1	WWW	5
2.2	E-mail	7
2.3	Downloads / FTP	9
2.4	Chat	10
2.5	Notice Board	12
2.6	Newsletter Subscription	12
3	Planning	14
3.1	Hardware Considerations	14
3.2	Operating System and Server Considerations	16
3.3	Development Tool Considerations	19
3.4	Web Application Framework	21
3.5	Deployment Issues	24
3.6	Security Considerations	24
4	Design	27
4.1	WWW Service	27
4.2	E-mail	30
4.3	FTP	36
4.4	Chat	38
4.5	Notice Board	40
4.6	Domain Name Space Service	40
5	Implementation	43
5.1	WWW Configuration	44
5.2	FTP Configuration	45
5.3	Sendmail Configuration	46
5.4	Newsletter Subscription	48
5.5	Chat	48
5.6	Notice Board	48
5.7	DNS Configuration	49
Appendix A	RFC 1123 – Requirement for Internet Hosts (Protocols only)	53
	References	62

## Chapter – 1

### Introduction

“Portal” is a collection of Internet technologies which provide different types of services. The most commonly used services that we come across in day-to-day Internet use are chat, e-mail, discussion group, news group, notice board, newsletter, forums, etc.

These services are provided by different organizations at different level. We can broadly categorize these services as (1)private and (2)public.

Many organizations provide more or less such services as per the needs of the organization. But these services are available only for members of that organization or people concerned with these organizations. Only such authorized members are able to use these services. Though the services are available on the Internet but these are authenticated by means of username and password so that only members can use them. Due to this, non-members can not see or use these services.

Such types of services are very useful for organizations. Many situations can be listed where these services are useful. The added advantage of on-line contact make Internet extremely useful for geographically separated branch offices (and members) of an organization to come close together.

For example, branch office employee can chat to any other branch office or head office and exchange information immediately or can share information. Newsgroup provides the facility to submit/update daily news/report which become available to all other offices. When a manager wants to discuss some issue, such as whether to adopt new policy, members can be invited to join discussion group. Management can put notice on notice board that becomes immediately available to all subordinates on-line. And e-mail facilitates greatly individual support either to customer or to subordinates. It is the most important service from the view of customer. Suppose a customer has purchased a

product and he is facing installation or some other problem. The technical support is far from his place. Still he can get immediate support from on-line customer support services now-a-days provided by many companies, moreover he can talk to support personnel and also able to witness live demonstration if he has required equipment such as PC camera, microphone, and speaker set with required Internet connection.

Till now there was no such medium available which could bring close many members close to each other and share information simultaneously. The telephone is one-to-one communication, which does not facilitate simultaneous communication or information sharing among many members.

Private portal services provide opportunity to members of an organization to share information on-line simultaneously which is authenticated by login process, and is not available to non-members.

There are many organizations which provide portal services to public. Most of these organizations provide these services free of cost. But the intention is to attract large number of people to avail these free services so that people will visit their web-site and these organizations advertise their products on these sites. However, apart from the advertising their products, these organizations provide a single platform to large number of people all around the world with different motives to come together, join people of similar interest, share information among all, exchange personal messages via e-mail, and so on. Public portals also contain more or less same components as private portals. Thus, Internet is a media which can bring billions of people close together and share information on personal level as well as group wise and at the cheapest rate.

The other personal level services provided are on-line membership, subscription to newsletter, etc. Portals also provide advertisement of products free of cost, product display, and on-line purchase; provides fast and cheap way for customer to view the product, its features and purchase it. And to host organization, it provides fast and cheap medium to reach people worldwide.

Portal is not a simple web site generating static or dynamic html contents. Portal provides number of different on-line services. Due to its on-line nature, it is not like web-sites that are hosted on professional web servers on rental basis. To host portal, the organization must own web servers and customized services. Also, it requires constant human interaction for services apart from regular maintenance. Thus, generally portals are deployed by larger organizations only.

The implementation of WWW, FTP, service will be carried using the NT server. It requires study of NT server, Internet Information Server, Configuration of these services on this server. Also, E-mail server will be implemented using Red Hat Linux 6.0 server for this thesis work. The implementation of newsletter will be carried out using aliases on Linux mail server. For Chat and Notice Board, one of the Java web server software will be loaded on server machine and application program will be developed for these components. With this introduction, the aim and objectives of the thesis are defined as follows:

### **1.1 Aim:**

Portal Design and Implementation.

### **1.2 Objectives:**

- I. To install and configure Microsoft Internet Information Server on Windows NT 4.0 server for WWW service and FTP service.
- II. To install and configure Red Hat Linux 6.0 server as e-mail server and implement the subscription in terms of e-mail aliases.
- III. Write program using Java language for Chat and Notice Board functionality.
- IV. Install and configure Domain Name Space Service. Register the Internet domain name with Indian Domain Registration Services.

## Chapter – 2

### Survey - Portal Components

Initially, I studied the underlying technique of most commonly used portal components. Alongwith it, it was necessary to find out how it can be possible to deploy these components on the server. It means what type of environment is required for execution of the component, what programming APIs will be required, which technology offers secure, reliable, portable solution, etc. Not only it was necessary to study how to deploy a component on the server, but also it was necessary to consider the security to that component on the server, if server is kept behind firewall like proxy server whether the client request will pass through the firewall or not and so on.

So it was a big work. In the beginning, at the time of selection of project, it was known that it is not possible to try to develop even most common used portal components for a single person. So I reduced the problem to only few components and that also with minimal functionality.

In the following survey, the components developed are described. The components are not described with the point of view of the actual technique used, but all possible ways to develop the component using Java and related technologies only. Since, it is possible to develop any component using Microsoft APIs as well as Sun APIs (which are used prominently for web development). But since the aim of the project is development using Java, only possible solutions using Java are devised. And in that also, using Java, Java Servlets, Java Server Pages and not using other development tools like RMI, CORBA, Beans, etc.

## 2.1 WWW :

### World Wide Web (WWW):

The World Wide Web is a worldwide information service on the Internet. It is a network formed by joining of computer hosts throughout the world. The Web, as it is popularly known, uses special software called a Browser (client) and TCP/IP, HTTP and a Web Server to function.

TCP/IP is the communication protocol used by the Internet and is a must for the WWW to function. HyperText Transfer Protocol (HTTP) is the protocol used by the WWW service to make communication possible between a Web Server and a Web Browser. A Web Server is a special software, which runs on a computer and responds to requests made by other computers on the network. A Web Browser is simply an application program, which sends request to a Web Server and accepts a response to that request from the Web Server.

### HyperText Markup Language (HTML):

Traditional methods of providing information to those who need it, like printed documents bound together in the form of manuals, books and so on, does not really work for software information. This is because this information is very dynamic and keeps changing and re-changing all the time.

A change in information requires the contents of books, manuals, and so on, to change. This will in turn require complete reprinting of the books and manuals. Reprinting always results in time and cost overheads. Reprinting also results in out dated books and manuals being held in inventory, unable to be used.

To resolve this problem, information needs be stored in such manner that whenever the information changes, these changes can be incorporated with least cost and time. A successful method that allows this is to store the information in the form of computer

based files. These files could be stored at a central location. Once stored at a central location the files can be accessed when required for reference.

Since these files are stored at a central location on a computer, their access will also require a computer and some sort of network that connects these two computers together. Thus to provide information to the users:

- Files holding information must be created.
- These files must be stored at a central location on a computer.
- When required the users should be able to access these files using their desktop computers.
- A network link must be established between the desktop computer and the computer serving information at a central location.

This introduces **Client/Server** terminology wherein

- The desktop computer requesting for information is termed as the **Client**.
- The computer serving information at a central location is termed as the **Server**.

It also introduces new problem that these files should be able to be interpreted by all types of computers and computer platforms. Also, it should be easier for any person to be able to view this information without knowing the details of computer terminology and should be able to navigate the information easily like usual books or manuals. The solution to these problems were devised by introducing a new language called HyperText Markup language.

The language used to develop the information in the form of pages is called **HyperText Markup Language (HTML)** and the information pages are known as Web pages. HTML is the language interpreted by a Browser. Web Pages are also called HTML documents. HTML is a set of special codes that can be embedded in text to add formatting and linking information. HTML is specified as TAGS in an HTML documents.

## 2.2 E-mail

E-mail or electronic mail is the most widely used service on the Internet. Since it allows highly personal communication, instantly, and at the cheapest rate. Most of the people who do not have personal Internet connection at home, still use it to communicate using connection at their work places or friends home, etc. The messages are delivered instantly. Like chat or other services, the person at the other end of communication need not be on-line.

This service is so highly preferred that many networking organizations have installed the e-mail servers. Many of them provide this facility to access to non-members also. One reason for the widespread use of the Internet as the international computer network is that it's a flexible enough system to allow just about any type of computer or network to participate. Whether you have a Mac, PC, or more exotic type of computer, whether you connect by modem or from a network, and no matter what e-mail program you use, you can still send and receive e-mail over the Internet.

There are many different e-mail programs available and may confuse a user. Many programs depend upon the type of connection a user has to the Internet. However, the operations will be more or less same.

Many networking organizations provide this service to non-members also free of cost. However, the motive behind this strategy is that these organizations advertise their products on the page of e-mail program thereby achieving advertisement. This way the advertisement reaches millions of people free of cost.

The basic facilities (operations) provided by any e-mail program are:

1. Run the e-mail program.
2. Send mail
3. Read incoming mail
4. Reply to mail
5. Delete mail

6. Send the same mail to more than one recipient at a time.
7. Store address book (including personal information).

There are basically two types of e-mail services:

1. **Web mail :** This type of service is open to public. Anyone can join as a member, register on-line, and free of cost. The main advantage of this service is that the member can access his mailbox from any computer and from any location worldwide.
2. **POP mail :** This type of service is provided at organizational level to members only. Large organizations maintain their own mail server and allow their staff, members, and clients to have account. This type of service is primarily maintained by organization for customer support such as complaints, feedback, order, registration, etc.

### **2.3 Downloads/ftp**

FTP (File Transfer Protocol) is the Internet's standard method for moving text files, data files, and binary program files from one computer to another. (The binary files may be executable files, picture files, sound files, or any other which can not be described as text or data file. Sometimes the data file itself may be binary).

As well known on the Internet, the documents describing number of topics, program files, sound and picture files, movies, animation, and number of other file types are available as downloads. Some of these can be procured by paying the required charges and many are available as free of cost. The main consideration here is that the size of these files is so large that it takes from few minutes to hours for complete transfer. For this purpose, a special Internet protocol is available which makes this transfer faster. It is known as File Transfer Protocol (ftp). On Internet, it is also known as downloads, which means get a copy of material (computer files) from remote computer to your own computer.

The purpose of FTP can be very well described by following example. Consider, some educational institute running a Distance Education Program. The institute use to supply some printed materials, books, etc to students spread in the region. If we consider the traditional way of distribution, the institute first compiles the material, then get it printed, then binds it in useful manner, prepares packets, put addresses and postage stamps, and finally posts it. The money and time required in this process is large. Also, it requires labor and record maintenance of which students are sent packets, some may not get due to unavailability at home, some may get in some spoiled situation, some may get delayed, and so on. Instead of this, if the same material is made available on the institutes' ftp site, the students can be informed to download it at their convenience. It will take hardly 5 to 10 minutes to download the complete semester course material. Student at his convenience can download the material. Thus, ftp will save the labor, time, and money involved in traditional distribution.

There are two types of ftp connections available:

**Anonymous:** As its name implies, the service can be availed by any person in the world. The contents (matter) available in anonymous ftp site can be accessed by anyone and it is available free of cost.

**Authenticated :** The authenticated ftp site can not be accessed by anyone. Whenever a person tries to connect to ftp site, the server asks for valid username and password which is authenticated (validated) by server. Upon confirmation, that the person trying to connect is a valid user, he is able to see and download the contents.

## 2.4 Chat

Chat as is known to most of us, is an Internet service which enable people connected by net to share information on-line. But the way to share information is somewhat uneasy. It is by means of directly typing through keyboard, the messages are shared among all members. Thus, the chatter has to continuously keep on pressing keys from keyboard.

Chat rooms are usually unruly. It means the information flows freely (not necessarily confined to a particular topic). And chat rooms can hold maximum 23 people at AOL (America On-Line). Auditoriums are somewhat similar to chat rooms, where special live events are taking place, have virtually unlimited seating (connection), but everyone is placed in a row, so the chatting doesn't go interest encouraging.

**Featured Chat :** These are the services provided by some organization where the chat topics are predefined. A user can get a list of topics called RoomList. User has to select one of the topics to join the chat room. Once joined the chat room, he gets the messages from on-line members and his messages are delivered to all other. Selecting MembersList option can see the members on-line.

**Member Chat :** Member chat is similar to featured chat, except the topics entry in ChatRoom was monitored by the servicing organization. In case of Member chat, the topic entry is allowed to members with some rules. A client can create a ChatRoom of his own interest and people having interest on the topic can join for chat. In these types of chat rooms, there is no user control to allow or deny someone to join the chat room. However, there are options settings in the application program, using which one can ignore messages from a particular rude talking member.

**Private Chat :** Unlike above types of chat, private chat is completely private. There are no special rooms, or topics, etc. It is personal. Any two or more members who recognize each other can enter chat room and share information, may be personal, home, office, etc type of information. In this type of chat, there is complete control with users as to who can join them for chat. They can accept or reject someone for chat.

**Chat Preferences :** These are the options available on chat application which allows us to customize chat sessions as we like. For example, as mentioned above, we can not deny some rude chatter in a public ChatRoom, however, we can customize the options to ignore messages from such members to appear on our screen. Also, there are some other options available such as:

1. Ignore member
2. Notify me when new member arrive
3. Notify me when a member leaves chat room
4. Enable chat room sounds

## **2.5 Notice Board**

Notice Board is usual term to almost everybody. It is a platform in which authorized persons can place the matter that is available to remaining members for view. This service not only contains notices to be brought to people but also other matter of concern to all members. For example, the periodicals, news, research papers, etc can be hosted. However, the direct hosting is not allowed to any member.

The matter to be hosted on notice board is first moderated by moderator committee members and upon approval, it is hosted. Then it becomes available for view to all members. This is the medium, which brings the latest information to the notice of members immediately.

## **2.6 Subscription/Membership**

Traditionally, when we subscribe to something, such as a magazine or a newspaper, it regularly arrives our doorstep or in mailbox,- daily, weekly, or monthly. In the New World of Internet, we don't need to subscribe these paper magazines or newspapers because we just have to open the required Internet site and it shows the current contents, news, etc.

But, if suppose we don't know when the contents of site changes because of difference in times across the world, also if suppose we want to get information from many sites (or sources). Again, if we happen not were able to connect to Internet for some time or days, and we can not every time go on browsing the sites only for arrival of new contents. Here comes the subscription in picture again.

There are number of web sites which offer subscription for their newsletter, magazine, new information. Usually, on the home page of such web sites we see one item Subscription/Membership. When we click this link, it takes us to fill up the form. The subscription form most commonly contains the name, full name, country, state, city, zip/pin code, e-mail address, hobbies, etc. Upon submitting this form, we receive confirmation of subscription by e-mail.

Compared to traditional subscription to different sources, the Internet subscription offers many advantages. The first and foremost is that most of these subscription are free of cost. These are delivered as soon as new information is released. We don't have to wait. There are no chances of losing delivery, because it is delivered to our e-mail account. There is no limit to number of sites to which we can subscribe, otherwise due to financial constraints, we cannot subscribe lot of magazines/newsletters.

There are some disadvantages observed. Some of the sites by getting registered, sends lot of their products information, more than the actual information.

Internet Explorer comes with automatic subscription facility. There are many sites, which do not offer subscription. Internet Explorer can be customized to receive information from such sites whenever these sites hosts new contents. Internet Explorer always keeps on checking the contents of our preferred sites, and if it finds the new information, it gets that information for us automatically.

## **Chapter – 3**

### **Planning**

In the previous chapter, the basic information about portal components is described such as how the portal components available on the Internet works. While developing the portal, it is not sufficient to consider only about the working of components and development tools and environments. Considerations about their deployment is equally important. The overall considerations need to be accounted from hardware requirement, setup, ... to deployment, and security. For this purpose, it is necessary to examine the following issues for successful deployment:

1. Hardware considerations.
2. Operating System/Servers considerations.
3. Development tools and environment considerations.
4. Additional Server extensions requirement.
5. Deployment Issues
6. Security Considerations.

#### **3.1 Hardware Considerations:**

The portal under development is proposed for a technical education institute. The portal will not contain components for attracting more and more people for some commercial purpose. However, it will serve the purpose of information source about the institute, for higher technical education, the short term and long term training programmes conducted by the institute, and other services provided by the institute.

These considerations are necessary to estimate the number of visitors to the site. This will decide the bandwidth requirement and hardware requirement. The institute serves higher technical education and training programmes for teaching and non-teaching staff members from Engineering Colleges, Polytechnics, Industrial Training Institutes, other

technical education institutes, and large and small scale industries in India. Taking a trial estimate, on average 10 to 20 visitors are expected fetching information at any time from the server.

The Institute has leased 64Kbps dedicated Internet connection for Internet access as well as portal services provision. The existing bandwidth will suffice the need. There will be services such as e-mail and ftp. For this purpose, a separate web server and e-mail server is proposed. The trial estimated servers configuration include:

- i. Pentium-II/III processor @666MHz (currently available product in the market)
- ii. 128MB RAM
- iii. 10GB Hard disk drives.
- iv. 48x CD-ROM drives.
- v. 10/100Mbps Ethernet Card(s).
- vi. Other necessary parts.

### **3.2 Operating Systems & Internet Servers Considerations:**

Before actually implementing the portal on any Operating System or Internet Servers, just by trial and error method or by the suggestion of others, it is necessary to consider thoroughly about these two factors. Because these two factors will decide the facilities provided for portal components management and deployment of specific services. These are some of the operating systems which are more preferred over others. It does not necessarily due to the facilities provided by one over the other, but sometimes, the familiarity of Server Administrator with a operating system in use in past in the organization, and moving for Internet hosting using that systems makes administration task and understanding services, configuration, errors, and performance monitoring becomes easier. Also, it decides the possibility of future expansion of site and services.

For example, though it is possible to deploy portal services on IBM mainframes with its multitasking, multiprocessing environment, the familiarity with Microsoft windows from home PC to office workstations, makes choice of Microsoft's Windows NT operating system for small organizations. It is observed in general that the organizations particularly in professional web sites hosting on rental basis are using Windows operating system whereas ISP's use UNIX operating system. It is also found that some organizations use Windows NT for contents deployment and use UNIX servers as e-mail servers.

Let us start from static web pages hosting. Almost every operating system in industry provide web server software for web contents deployment. Of all, the most widely used operating systems are Windows NT, Unix, Linux, IBM MVS, AS/400, etc. However, the selection of the system is not only driven by the familiarity and ease of use, but also the size of organization itself, its purpose, requirement, and the investment possible.

The operating system selection plays important role. It aids not only web site (portal components) management but also provides user access control. Further, the selection is driven by the tools to control user access. The site may be used for Intranet as well as Internet hosting. In such case, only few contents will be available to the rest of the

world, while members may have direct logon facility on the server. The user and group management tools available makes choice.

Most users like graphical user interface for operations compared to text based prompts and command interface. The graphical user interface (GUI) makes all the options available on the dialog boxes. There is no need to remember the options and their syntax at the command interpreter. Further, the GUI provides What You See Is What You Get (WYSIWYG) interface, which makes its obvious choice since we confirm ourselves about how the contents will be presented to users. On the other hand, generally GUI itself occupies a substantial amount of memory compared to text interface.

After selection of the operating system, the next choice arises among web server softwares. The operating systems listed above provide their own web server softwares, as well as third party web server softwares are available in the market. The choice for particular software is driven by the ways of handling user requests, dynamics of contents generations, extensions possible to software and services, processing of requests, and security. User requests handling comes into picture when developing web applications which needs the user requests be handled by server applications and accordingly generation of contents. At the same time, every request must pass through operating system authentication process, otherwise access control will be lost.

Dynamics of contents generation refers the way dynamic contents can be generated. It may need processing of the user sent data, database integration (storage and retrieval of data using third layer of database), or passing request from one program to another, processing relevant information at each stage in the server, and finally sending the well formatted output to the user.

Possible extensions and services refers to the ease with which new services can be added or modifying existing services, and the possibility of extending server software with future requirement without or with little modifications to existing applications.

Ways of processing requests refers to how the server software handles each user request. Some server softwares generate separate process space for each user request which consumes huge memory and time resources. While some servers make use of existing one process space only for all requests and generate threads for each individual user request, thereby saving memory and time. For example, using CGI scripts server software creates separate process space for each user request, whereas Java servlets, JSP create only one process space for one application and generate threads for individual user requests.

Security is most important. Many server applications create ports for listening user requests and handling requests efficiently. However, it has been devised time to time that the malicious virus programs try to enter server using this technique. Also, some hackers try to gain entry inside server using well-known ports such as ftp, SMTP, etc.. So how much these ports are protected by server and operating system is an important consideration.

Beside the extensions, the extension services provided by operating system such as ftp, e-mail, etc and the interface provided by the server software makes the choice. Further, the ease of internationalization of contents generated by server applications and locale, support for plug-ins, drives choice among server softwares.

Till recently, Apache web server was the leading server software, but since it is available free of cost, recent developments are not included in it. The other two server softwares play major role. First is Sun Microsystems, JavaWebServer and second is Microsoft's Internet Information Server (IIS). Microsoft's IIS has been powerful from the beginning at every stage – from contents preparation, generation, hosting, etc. with ActiveX a powerful software plug-in. All provided by one manufacturer, compatibility and consistency among its all components and latest up-to-date revisions. However, the only drawback that the possibility of embedding malicious code inside ActiveX controls, which can destroy the client-side as well as server-side softwares, has led it to second preference as server software. And today, the most widely used software is JavaWebServer from Sun Microsystems.

### 3.3 Development Tools Considerations:

The choice of development tools many times depends upon the selection of operating system and web server software. It is necessary to consider whether the selected development tool can be used on the system. Also, whether it will be possible to utilize the tool with its full potential on the platform or not.

There are two major language tools available with their corresponding environment for execution available. These are Java programming language and Java web server as environment for execution from Sun Microsystems. The another is Microsoft's C/C++ and Visual Basic as development tools with its proprietary web server, viz. Internet Information Server (IIS).

As already noted, the power and flexibility offered by Microsoft's tools and products is more than others. Also, the speed of execution of components or services is also more than the Java. The speed of execution is more because of readily compiled code used for deployment compared to byte code provided by Java environment. But Microsoft's products are limited in use due to following factors.

1. The compiled code is available from Microsoft products, it is not portable. The compiled code is platform specific. It is limited in use on systems using Microsoft operating systems only. It can not be executed on other systems.
2. The compiled code can contain hostile code which can disrupt the system on which it is downloaded for execution. For instance, it is possible to embed hostile code with ActiveX controls.
3. Microsoft products are proprietary and there is no license for open source code further development and deployment, it restricts volunteer users, like Java.
4. Another factor which takes away choice to use Microsoft's development tools and environment is due to proprietary rights. The server API's are not available to other vendors for incorporating in their servers such that though a user may use other operating system and server, the Microsoft's tools can be used for development. There is no support for softwares compiled using Microsoft products to execute on

other systems. For example, web servers from Sun Microsystems do not have provision to execute server components or applications developed by users using Microsoft's tools. Whereas, Microsoft's operating systems provide ability to install Sun Microsystems web server tools and environment to be used on their Windows systems.

5. The last factor which encourages developers to move away Microsoft systems and towards other Unix compatible platforms is availability of free operating systems such as Linux for development purpose along with availability of huge open source.

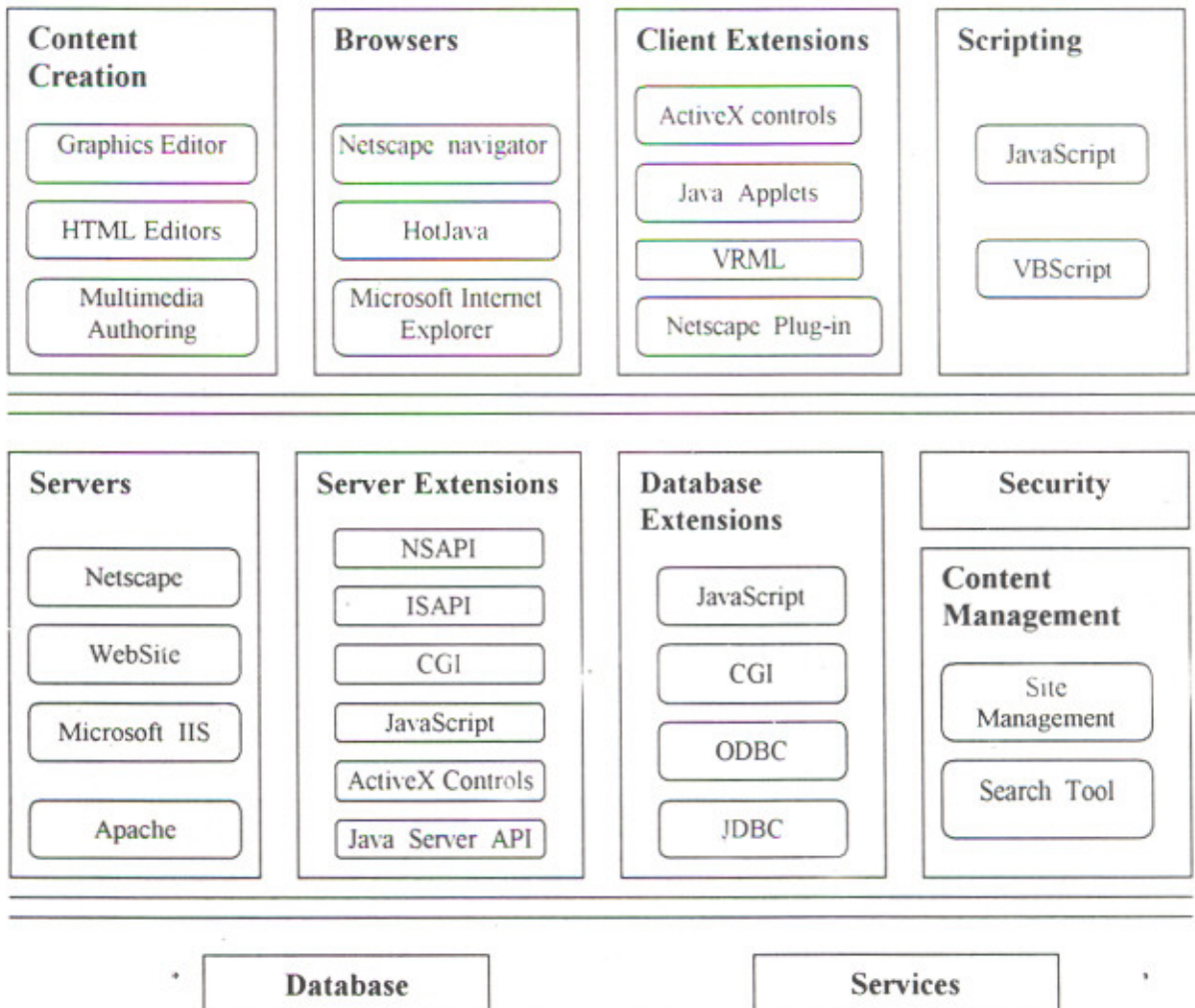
Above factors have driven mob of experts as well as new developers away from Microsoft systems, towards Unix/compatible systems with Java as development tool and environment from Sun Microsystems, which are also available for free trial evaluation.

Apart from development tools and environments which can be used to develop applications from scratch to specific customized needs, some ready to use packages are available for fast and easy preparation. One of such package is Net.commerce from IBM.

### 3.4 Web Application Framework

Using the Web as a development environment is a relatively new phenomenon. With the advent of Java, JavaScript, ActiveX, and other technologies, the idea of developing Web centric applications has many qualities.

The Web as a development environment can seem rather confusing. Since of the distributed nature of Web, a Web application can be composed of many parts, using a variety of technologies. Figure 3.1 shows the various parts that make up the application framework.



## **A. Client Side:**

The client side of the Web application framework consists of four building blocks:

- Browsers
- HTML (HyperText Markup Language)
- Client-Side Extensions
- Scripting Languages

### **A.1 Browsers:**

The most important component of a Web application is the browser itself. The browser is the window to the Web for the user and serves as the user interface for application. Browser technology is relatively simple, but the advent of nonstandard enhancements, such as Netscape layers and JavaScript, has made the selection of browser software a critical one to determine a Web development platform.

### **A.2 HTML:**

HTML is obviously one of the primary technologies upon which the Web is built. HTML is a markup language that is used to provide structure and formatting to a plain text file. Although the browser provides the window for displaying Web-based content to the user, the content itself comes in the form of HTML text. It doesn't matter whether to present static documents, returning a query result, providing a feedback form, or displaying a JavaScript based application. Regardless of the means of obtaining the data, it is ultimately converted into HTML tags for presentation.

### **A.3 Client Extensions:**

As the need for active Web pages increased, simply presenting the text information was not considered the best solution. Some extensions were third-party add-ons to the browser software to make it more powerful. However, there was also the need to work with executable content within the browser. Although the browser needed to support the technology, it didn't need to be tied to the browser to run. Three separate client-side extensions are emerged today. They have similarities, but each has a distinct identity. These are: 1)Java Applets 2)ActiveX controls 3)Netscape plug-ins

#### **A.4 Client Scripting Languages:**

The final pieces of software on the client side are the client scripting languages. JavaScript is the leading scripting language, but Microsoft is also promoting VBScript as an alternative.

### **B. Server Side:**

The server side of the Web application framework consists of the Web server itself, along with extensions to the server software. The server extensions can take various forms and can be employed with a variety of technologies.

#### **B.1 Servers:**

The Web server is provided with handling requests for HTML documents from the client and returning them for viewing. The server software is an application that runs on a TCP/IP enabled machine. Popular servers today include Netscape Enterprise Server (NES), Microsoft Internet Information Server (IIS), and Apache.

#### **B.2 Server Extensions:**

The Web server itself provides static HTML pages to the client when requested and performs a variety of other functions. However, several extensions to servers provide capabilities that the server itself doesn't support. These include 1) CGI, 2) server APIs, 3) JavaScript, and 4) Java.

### **3.5 Deployment Issues:**

Here the server-side deployment issues are considered. The Unix and Windows systems provide more or less the same level of security and easy of management.

The Unix systems works directly with TCP/IP ports creation, multi-threading, different levels of processes scheduling and priority. Microsoft systems do provide these supports. It creates its own Windows sockets and ports instead of directly supporting TCP/IP ports, supports multitasking and multithreading.

Both systems support other services such as ftp, gopher, remote login, support for extension to server softwares. However, there is difference in e-mail handling system. On Unix systems mails are handled using their own product Microsoft Exchange Server which supports POP3 as well as IMAP. On Unix systems, mails are handled using sendmail and client can be configured to use POP3 or IMAP for retrieval of mails.

Both server softwares support Java tools and environment. Thus, there is no significant difference between the two operating systems which can make selection choice of one over other. However, some of the educational institutes use Linux since it is available free of cost.

### **3.6 Security Considerations:**

Being on the Internet full time leaves the site open to potential problems with hackers and mischief. The very thing that made the Internet grow so large and fast made it such a dangerous place. Internet security is a complex topic. Merely by reading the books and making security arrangement does not claim the full security. Also, there may be some sites which might not have faced any problems. However, the minimum known precautions should be observed as a wise suggestion from experts.

There are two basic routes to take with Internet site security:

1. Using some basic Internet smarts and other tricks.

2. Setting up firewall like proxy server.

### **Basic Internet Smarts and Tricks:**

A fairly secure site can be hosted on Internet without doing anything spectacular, following some small tricks such as:

1. Disable services not actually offered by the site over the Internet.
2. Limit access to logging in as Administrator/Root to make it difficult for attackers outside the site to get root access.
3. Address Operating system and Internet security issues.
4. Periodically test for security.
5. Update own information about Internet security using different media as a source.

It can run a fairly secure site this way. However, it is not sufficient for sites who have a need to ensure that no one can get into their systems, such as organizations involved in financial transactions over the Internet.

### **Firewalls:**

The first requirement for setting firewalls is to remove any RAS based connection and implement an Ethernet or Router-based connection.

The second, it requires an extra computer to devote for firewall purpose.

A firewall is like a one-way mirror. It lets users see from outside of the site and doesn't let anyone see inside. It processes outside users requests on behalf protected web servers and let look to the rest of the Internet as if all the servers on that machine or as if it is actually processing all user requests.

Running a site with a firewall is a lot of extra work for the system administrator, but worthwhile for protection of web servers. Also, it needs to implement many extra security measures on each computer on the local network and prohibit the intranet users to install unknown third-party or not essentially required software/services.

It might be necessary to use firewall for one of the following reasons:

The data on the site is such that it absolutely need to make sure people can't get to. The type of data to be protected is that data people might actively try to get and the site can't afford for people to get it. For example, vital data such as financial data that can make it or break it for business may not be a good thing to keep on a computer that is accessible to the Internet without being protected by a firewall.

The site will be a target for intruders because of its nature. For example, a business in a highly competitive field facing problems with people stealing information from one another, make it necessary to install firewall.

As a practice, one may want to setup and maintain a firewall.

A firewall can be software, hardware, or a combination of the two. A firewall usually includes several software tools. For example, it might include separate proxy servers for e-mail, ftp, Gopher, Telnet, Web, and WAIS. The firewall can also filter certain outbound ICMP (Internet Control Message Protocol) packets so that server won't be capable of divulging network information.

The proxy server is used to mask the Intranet IP addresses on outbound packets so they look like all originated at the proxy server itself. Each of the client machines on Intranet must use the proxy server whenever they connect to the Internet for FTP, Telnet, Gopher, or the Web. The reason for doing this is to prevent outside detection of the structure of your network. Otherwise hackers monitoring outbound traffic would eventually be able to determine individual IP addresses and then use spoofing to feed those back to the server when they want to appear as a known client.

Another purpose of a firewall is to perform IP filtering of incoming packets. Suppose it is detected that some one is sending strange packets to the network, the IP and name of the site can be determined with WHOIS and proxy can be set to reject all the packets originating from that domain.

## Chapter – 4

### Design

#### 4.1 WWW Service :

##### Web Server:

Web pages are created using HTML syntax. These pages must be organized and stored at a central computer. The organization of web pages into directories and files stored on the HDD of a central computer is called 'Web Site' creation.

Computers, which store web pages in the form of folders and files and provide these files to be read, are called 'Servers'. They act like service providers that service the need for information.

The Server Computer runs special software called 'Web Server' software that allows:

- Web Site Management.
- Accept a client's request for information.
- Responds to a client's request by providing the page with the required information.

Web Server Software stores and manages web pages. When required, the Web Server accepts requests for these Web Pages, retrieves these web pages from its HDD, and sends the page back to the client who requested for it.

Some of the most popular software, which Servers run to allow them to respond to client requests for information, are Internet Information Server (IIS), Apache Web Server, Netscape Server, etc.

##### Web Client/Browser:

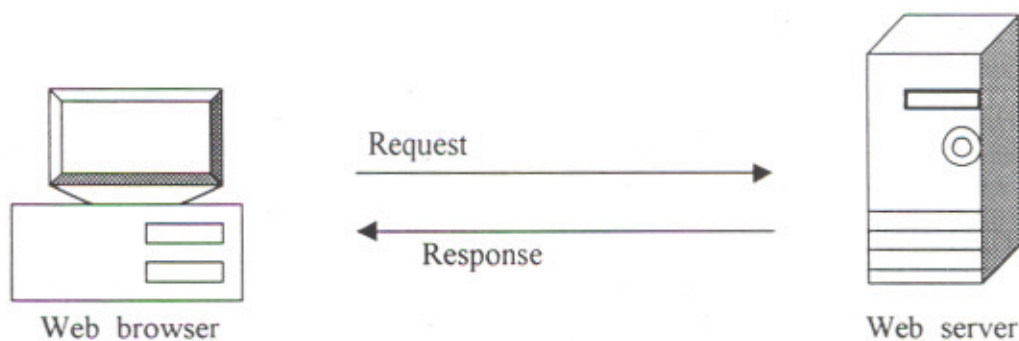
To access information stored in the form of web pages, users must connect to a Web Server. Once connected, an interface that displays the contents of the web page required.

Computers that offer the facility to read information stored in web pages are called '**Web Clients**'. Web Clients run special software called a '**Browser**' that allows them to:

- Connect to an appropriate Server.
- Query the Server for the information needed.
- Provide an interface to read the information returned by the Server.

#### Http request and response process:

Suppose we enter the following URL into address field of browser and press return key. `http://www.howstuffworks.com/web-server.htm`. And at next moment, no matter where in the world that URL lives, the page pops up on our screen. At the most basic level, the following diagram shows the steps that brought that page to our screen.



**Figure 2.1 Processing of http request by web server.**

The browser establishes connection with a Web server, requests a page and receives it. In detail, the basic steps followed are:

- The browser breaks the URL into 3 parts:
  1. The protocol ("http")
  2. The server name ("www.howstuffworks.com")
  3. The file name ("web-server.htm")
- The browser communicates with a name server to translate the server name "www.howstuffworks.com" into an **IP Address**, which it uses to connect to the server machine.

- The browser then formed a connection to the server at that IP address on port 80.
- Following the HTTP protocol, the browser sent a GET request to the server, asking for the file "http://www.howstuffworks.com/web-server.htm".
- The server then sends the HTML text for the requested Web page to the browser.
- The browser read the HTML tags and formatted the page onto your screen.

From the above process, we observe that the web server takes the filename sent in with the GET request from the browser, retrieves the requested file and sends it back to the browser. This is all good for web server that delivers standard "static" pages. "Static" pages are those that do not change unless the creator edits the page.

However, we know that web servers does not simply provide static information. It also provides dynamic information. For example:

- Any guest book allows us to enter a message in an HTML form, and the next time the guest book is viewed, the page will contain the new entry.
- Any search engine lets us enter keywords on an HTML form, and then it dynamically creates a page based on the keywords we enter.

In these cases, the web server is not simply "looking up a file." It is actually processing information and generating a page based on the specifics of the query. In almost all cases, the web server executes some programs to accomplish this.

There are many programming tools to generate these dynamic contents. For example, the server may use CGI scripts, Server-side Java scripts, or Active Server Pages, Java, Java Servlets, etc. These programs are executed on server to produce required results that are again formatted using HTML, so that it can be presented on the browser.

## 4.2 E-mail :

### Understanding Email Clients

Many users receive e-mail messages at our e-mail account. One may have e-mail account on work organization's private server or may be on free services provided by some web sites. To see the messages received, people use well-known stand-alone clients like Microsoft Outlook, Outlook Express, Eudora, etc. People who subscribe to free email services, use an email client that appears in a web page. No matter which type of client user uses, the email client generally does four things :

- It shows the list of all messages in the mailbox displaying the message headers. The header shows who sent the email and the subject of the mail and may also show the time and date of the message and the message size.
- It lets us to select a message header and read the body of the email message.
- It lets us create new messages and send them. Type the email address of the recipient and the subject for the message and then type the body of the message.
- Most email clients also lets add attachments to messages to send and save the attachments from messages we receive.

### Understanding a Simple Email Server

Machines on the Internet can run software applications that can act as servers. There are web servers, FTP servers, telnet servers and email servers running on millions of machines. These applications run all the time on the server machines and listen to specific ports waiting for people or programs to connect to the port.

- It would have a list of email accounts, with one account for each person who can receive email on the server. For example, the someone's account name may be xyz.
- The e-mail server maintains a text file for each account in the list. So the server would have a text file in its directory named XYZ.TXT for an account corresponding to account name xyz.
- When someone wants to send an email message to xyz person, the sender composes a text message in an email client, and indicates that the message should go to xyz.

- When the person presses the **Send** button, the email client would attach to the email server and pass to the server the name of the recipient (xyz), the name of the sender and the body of the message.
- The server would format those pieces of information and append them to the bottom of the XYZ.TXT file. The entry in the file might look like this:

There are several other pieces of information that the server might save into the file, like the time and date of receipt and a subject line, but overall it is a extremely simple process.

As other people send mail to xyz, the server would simply append those messages to the bottom of the file in the order that they arrive. The text file would accumulate a series of messages. When xyz want to look at his email, the email client would connect to the server machine. In the simplest possible system it would:

- Ask the server to send a copy of the XYZ.TXT file
- Ask the server to erase and reset the XYZ.TXT file
- Save the XYZ.TXT file his local machine
- Parse the file into the separate messages (using the word "From:" as the separator)
- Show him all of the message headers in a list
- When XYZ double-click on a message header, it would find that message in the text file and show him its body.

### **Understanding Real Email System:**

The real email system consists of two different servers running on a server machine. One is called the SMTP Server, where SMTP abbreviated for Simple Mail Transfer Protocol. The SMTP server handles outgoing mail. The other is a POP3 Server, where POP abbreviated for Post Office Protocol. The POP3 server handles incoming mail. The SMTP server listens on well-known port number 25, while POP3 listens on port 110.

Whenever we send a piece of email, the email client interacts with the SMTP server to do the sending. The SMTP server on the host may have conversations with other SMTP servers to actually deliver the email.

Let's assume that I (xyz) want to send a piece of email. My email ID is xyz and I have my account on abc.com. I want to send email to zyx@pqr.com. I am using a stand-alone email client like Outlook Express.

When I set up my account at abc, I configured Outlook Express (or pine on UNIX systems) the name of the mail server - mail.abc.com. When I compose a message and press the Send button, the sequence of events happens as follows:

- Outlook Express connects to the SMTP server at mail.abc.com using port 25.
- Outlook Express has a conversation with the SMTP server. The conversation is an extremely simple set of text commands and responses (as described below). Outlook express tells the SMTP server the address of the sender and the address of the recipient., as well as the body of the message.
- The SMTP server takes the "TO" part of address (for example, zyx@pqr.com) and breaks it into two parts. The first part is the recipient name (zyx) and the second part is the domain name (pqr.com). If the TO address had been another user at abc.com, the SMTP server would simply hand the message to the POP3 server for abc.com (using a small program called the **delivery agent**). Since the recipient is at another domain, SMTP needs to communicate with that domain.
- The SMTP server undergoes a conversation with a Domain Name Server to find out the IP address of the SMTP server for pqr.com. The DNS replies with the one or more IP addresses for the SMTP server(s) that Pqr operates.
- The SMTP server at abc.com connects with the SMTP server at Pqr using port 25. It has the same simple text conversation that email client had with the SMTP server for ABC, and gives the message to the Pqr server. The Pqr server recognizes that the domain name for zyx is at Pqr, so it hands the message to Pqr's POP3 server, which puts the message in zyx's mailbox.

If, for some reason, the SMTP server at ABC cannot connect with the SMTP server at PQR, then the message goes into a queue. The SMTP server on most machines uses a

program called **sendmail** to do the actual sending, so this queue is called the **sendmail queue**. Sendmail will periodically try to resend the messages in its queue. For example, it might retry every 15 minutes. After four hours it will usually send us a piece of mail so that we know there is some sort of problem. After five days, most sendmail configurations give up and return the mail to us undelivered. The conversation that an email client has with an SMTP server is likely as follows:

```
Email client: helo test
SMTP server: 250 mx1.pqr.com Hello abc.sample.com
                [220.57.69.37], pleased to meet you
Email client: mail from: test@sample.com
SMTP server: 250 2.1.0 test@sample.com... Sender ok
Email client: rcpt to: zyx@pqr.com
SMTP server: 250 2.1.5 zyx... Recipient ok
Email client: data
SMTP server: 354 Enter mail, end with "." on a line by itself
Email client: from: test@sample.com
                to:zyx@pqr.com
                subject: testing
                John, I am testing...
SMTP server: 250 2.0.0 e1NMajH24604 Message accepted for delivery
Email client: quit
SMTP server: 221 2.0.0 mx1.pqr.com closing connection
                Connection closed by foreign host.
```

What the email client says is followed by Email client, and what the SMTP server replies with is indicated followed by SMTP server. The email client introduces itself, indicates the from and to addresses, delivers the body of the message and then quits. We see that the SMTP server understands very simple text commands like HELO, MAIL, RCPT and DATA. The list of most common commands is listed below:

- HELO - Introduce yourself
- EHLO - Introduce yourself and request extended mode
- MAIL FROM: - Specify the sender
- RCPT TO: - Specify the recipient
- DATA - Specify the body of the message. To:, From: and Subject: should be the first 3 lines.
- RSET - Reset
- QUIT - quit the session
- HELP - get help on commands

- VRFY - verify an address
- EXPN - expand an address
- VERB - verbose

### Understanding POP3 Server

In the simplest implementations of POP3, the server really does maintain a collection of text files, one for each email account. When a message arrives, the POP3 server simply appends it to the bottom of the recipient's file.

When we check our email, our email client connects to the POP3 server using port 110. The POP3 server requires an account name and a password. Once we log in, the POP3 server opens our text file and allows us to access it. Like the SMTP server, the POP3 server understands a very simple set of text commands. Here are the most common commands:

- USER - enter your user ID
- PASS - enter your password
- QUIT - quit the POP3 server
- LIST - list the messages and their size
- RETR - retrieve a message number. Pass it a message number
- DELE - Delete a message. Pass it a message number.
- TOP - shows the top x lines of a message. Pass it a message number and the number of lines.

The email client connects to the POP3 server and issues a series of commands to bring copies of our email messages to our local machine. Generally it will then delete the messages from the server (unless configure the email client not to do so). Thus, the POP3 server simply acts as an interface between the email client and the text file containing our messages.

### Understanding Attachments

The email client allows us to add attachments to email messages to send, and also lets to save attachments from messages that you receive. Attachments might include word processing documents, spreadsheets, sound files, snapshots, pieces of software, etc. Usually an attachment is not text (if it was, we would simply include it in the body of

the message). Since email messages can contain only text information and since attachments are not text, there is a problem that needs to be solved.

In the early days of email, the problem was solved manually by using a program called **uuencode**. The uuencode program assumes that the file contains binary information. It extracts three bytes from the binary file and converts them to four text characters (that is, it takes 6 bits at a time, adds 32 to the value of the 6 bits and creates a text character). The uuencode produces an encoded version of the original binary file that contains only text characters. In the early days of email you would run uuencode yourself and paste the uuencoded file into your email message.

Here is typical output from the uuencode program:

```
begin 644 reports
M9W)E<" B<&P_(B O=F%R+VQO9R]H='1P9"]W96(V-C1F-
BYA8V-E<W,N;&]GM('P@8W5T("UF(#(@+60@(C\B('P@8W5T
("UF(#$@+60@(B8B(#X@<V5A<F-HM+61A=&$M)#$*?B]C;
W5N="UP86=E<R!\(' -O<G0@/B!S=&%T<RTD,0IC<
"@M?B]W96)S:71E+V-G:2UB:6XO<W5G9V5S="UD871A+V1A=
&$@<W5G9V5S="TDM,0IC<"!^+W=E8G-I=&408V=I+6)I;B
]W:&5R92UD871A+V1A=&$@=VAE<F4MM)#$*8W @?B]W96)S:7
1E+V-G:2UB:6XO96UA:6QE<BUD871A+V1A=&$@96UAL:6PM)#
*$?B]G971L;V<@/B!L;V=S+20Q"GXO=&]T86P@/B!T;W1A;"T
D,0IA
```

End

The recipient would then save the uuencoded portion of the message to a file and run **uudecode** on it to translate it back to binary. The word "reports" in the first line indicates uudecode what to name the output file.

Modern email clients are doing exactly the same thing, but they run uuencode and uudecode for us automatically. If we look at a raw email file that contains attachments, we would find that the attachment is represented in the same uuencoded text format shown above.

### 4.3 FTP

Early FTP programs were character based. The user typed DOS or UNIX commands to logon to the remote computers, browse directories, transfer files, or log-off. With the arrival of hypertext and WWW, new graphical interfaces have been developed to make FTP easier to use.

There are two methods of using FTP:

1. Establishing an account on the remote server.
2. Using anonymous FTP.

In the case user has account on a remote server, user can login. An anonymous FTP account allows user to connect to the remote host computer and download files. The user simply logon to the system with the username "anonymous" and password provided is usually user's e-mail address. Anonymous FTP server will normally restrict user's access to certain folders and files.

The main difficulty with either type of FTP is sourcing information. There are thousands of FTP sites on the Internet and it is difficult to know which one to look in for specific information. Also, after finding the correct site, user has to find out which directory the file is in.

Many FTP sites do not have their files listed in any particular order. For this purpose, Archie was developed. Archie is an information retrieval program which periodically connects to all the FTP sites on the Internet. It downloads lists of files and merges them in a database which can then be searched using an archie client program.

Files in any format can be accessed through FTP, for example, document files, multimedia files, or application files. FTP allows to transfer these files in ASCII or BINARY format. Any file that is not a text file is transferred in binary format.

FTP uses TCP as its transport protocol for all communications and data exchange between server and client. FTP uses two types of TCP connection to communicate with a host:

1. Control connection process
2. Data Transfer connection process.

The control connection process establishes a control connection or link between devices. The data transfer connection process is responsible for directing the flow of data. Control and data transfer connections can be in one of two states:

- i. passive open – waiting for a transmission.
- ii. active open - initiating a transmission.

The data transfer connection can not exist until control connection is established, and only if there is actual data to be transferred between the devices. Because FTP uses two types of connection, it uses two ports:

- i. Port 20 – for data transfer
- ii. Port 21 – for control connection.

## 4.4 Chat

The Chat application will contains the following functions:

- Chat room administration, where new rooms can be defined and existing ones removed.
- Chat room list, where a user can read the descriptions of all rooms and enter a room.
- Chat room, where multiple users can chat with each other.

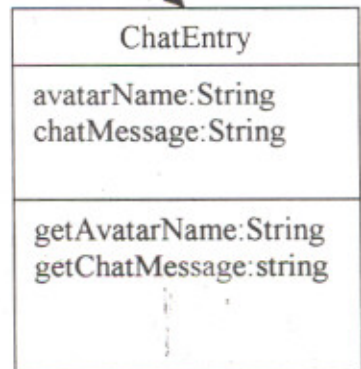
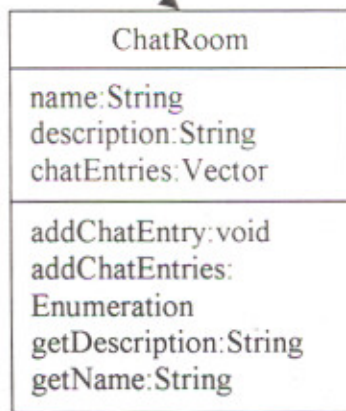
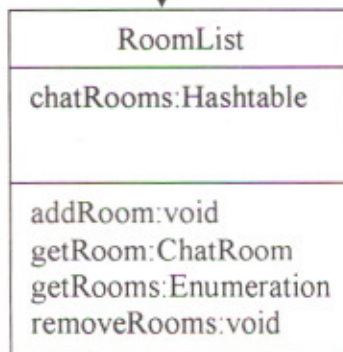
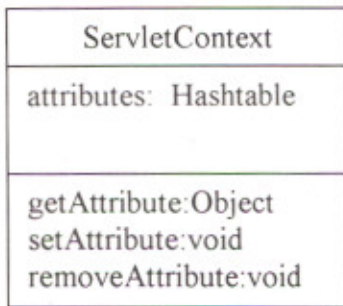
The application contains two types of objects: data objects and user interface objects.

Data Objects:

- **RoomList** : One reference of this clas holds references to all ChatRoom objects. This object is kept as a ServletContext attribute so it can be accessed by all the servlets that make up the application.
- **ChatRoom** : There is one Chat Room object for each room, containing information about the room and a collection of all ChatEntry objects for the room.
- **ChatEntry** : A ChatEntry object represents what someone says in a chat room. Each object contains the name of the person who said something and the actual message.

The application will have browser based interface, implemented by the following servlets:

- **ChatAdminServlet** : This servlet is loaded when the servlet engine is started and creates a RoomList object with all ChatRoom objects defined in an application configuration file. When the server is running, it is used to define new rooms and to remove existing ones.
- **ListRoomServlet** : This servlet lists all available rooms and lets the user read about them and enter a room.
- **ChatRoomServlet** : provides the user interface to a ChatRoom. Users can see what other people are saying and send their own messages.



## 4.5 Notice Board

In this application, the notice board is implemented using servlet. To store all the notices, a database is created using Oracle Database. The structure of the table is shown below. The NBAdmin Servlet is used to append new notice in the database.

Sr. No.	Attributes	Type	Description
1	REFNO		Reference Number
2	DISPLAYDATE	Date	Date on which notice appended in database for display
3	EXPIRYDATE	Date	Date after which the notice will not be available for users
4	SUBJECT		Subject of the Notice
5	AUDIANCE		Users for whom the notice is intended For ex. FACULTY/STAFF/STUDENTS
6	MESSAGE		Actual message to be displayed
7	AUTHORITY		Name of the Officer with whose instructions the Notice is displayed.
8	DESIGNATION		Designation of the Authority Officer
9	NOTE		Special Note for office record

## 4.6 Domain Name System

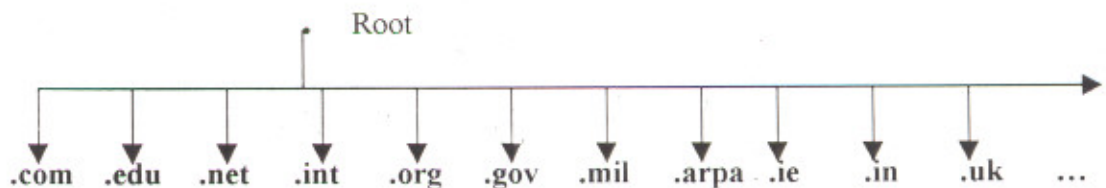
Although users prefer to use friendly names such as TTTIChandigarh, to refer to network computers, computer often use much less user friendly forms of address. Computers on Internet uses TCP/IP for communication and uses IP addresses such as 172.16.180.30 to recognize each other. DNS is the standard method for naming and locating resources on the Internet. It provides the service of mapping computer names-to-IP address.

DNS was developed to meet the requirements of the Network Information Center (NIC) in early 1980. The NIC was wholly responsible at this time for administering the ARPAnet, which was a predecessor of the current Internet. Before 1983, a static host name file was used to map computer names to IP addresses on the ARPAnet. This file

was small, stable, and could be manually updated every few days. However, with the introduction of Internet, the number of hosts started to grow exponentially, which became an impractical to maintain and update static file manually.

DNS was developed to replace static file mappings of host name-to-IP addresses and resolve the scalability problems. Unlike the flat name space that preceded it, DNS implemented a hierarchical name space called the Domain Name Space (DNS).

The DNS is a tree hierarchy that branches out from a root to all other nodes on the network. Each node in the tree, along with all the nodes below it, is called a **domain**. Each domain may contain hosts and other sub-domains. The name mapping database is divided among many DNS servers – each of which is responsible for resolving name queries for a single domain.



The topmost domain is called root domain. Below the root domain are several top-level domains most of which serve to organize the many sub-domains on the Internet into general categories. The top-level domains are divided into three main types:

1. Organizational domains
2. Geographical domains
3. In-addr.arpa domain.

Internet expansion led to the introduction of the top geographical domains, which provide a top-level domain to group sub-domains for every country in the world. Each country geographical domain has a two-letter name. For example, **.uk** for United Kingdom, **.in** for India, etc. The **in-addr.arpa** domain is reserved for reverse look-ups of IP-addresses to host name mapping.

The Internets' root and top-level domains are administered by the Internet Network Information Center (InterNIC). The InterNIC is consortium of organizations that provides network information services for the Internet. One of the Internics' most important function is to provide a registration service for Internet domain names that ensure unique name. Below the top-level domains, the InterNIC delegates responsibility for managing the domain name space to other organizations. The delegated organizations can further divide their areas of the domain name space and delegates responsibility for these subdivisions.

Areas of the domain name space that can be administered as separate entities are referred to as **zones**. A zone may consists of a single domain or a domain and a number of sub-domains. A single domain can be split into multiple zones, one of which is designated as the primary zone, while the others are designated as the secondary zones.

To identify a particular host or domain within the structure of domain name space, the name is written from left-to-right with the name followed by the names of each of the domains between it and the root. A dot separates one part of a domain name from another giving rise to the address format familiar to Internet and e-mail users.

The parts of the domain name that are separated by dots are also known as nodes. Each node is limited to 63 characters long. Any domain name that includes the complete chain, from the host through all superior domains to the top of the DNS tree is called a Fully Qualified Domain Name (FQDN). For example, ttichandigarh.ac.in.. The final dot that refers to the root domain is not necessary and is usually omitted. It represents the presence of the root domain implicitly.

## CHAPTER-5 IMPLEMENTATION

### Installation and Configuration of NT server 4.0

Following table lists the summary of key information required during Microsoft NT Server 4.0 installation. Also, the installation sequence asks whether to install Microsoft Internet Information Server. The required checkbox options be selected to install IIS with FTP as additional service installed.

Sr. No.	Particulars of Information	Value decided
1	File System Type	NTFS
2	Installation Drive and directory	C:\WINNT
3	Department Name	COMPUTER
4	Organization Name	TTTI Chandigarh
5	CD key	
6	Installation mode	Per User
7	Computer Name	WEBSERVER1
8	Domain Name	TTTICHD
9	Administrator Password	
10	Protocol Information for TCP/IP	
	IP address	203.197.15.168
	Subnet Mask	255.255.255.240
	Default Gateway	
	DNS/WINS server IP address	

After installation, the WWW and FTP services need configuration as described below. By default, the setup program sets properties to default values. There are minor changes or additional information need to be included.

## 5.1 WWW service Configuration:

To customize the settings of the default web site, select menus from Start – Program – Microsoft Internet Server (Common) – Internet Service Manager. This opens Internet Service Manager window which can be used to customize all Internet Services viz., WWW, FTP, Gopher, NNTP. However, the window displays only the installed components. To customize the service, double-click the computer name. It opens the Properties dialog window for the selected service. For WWW service settings, double-click it.

The WWW service properties for TTTICHHD dialog box opens. The dialog box shows four tabbed boxes viz. Service, Directories, Logging, and Advanced, with currently selected Service tab box.

Using the Service tabbed box, we can set following properties :

TCP port	:	80 (default)
Connection Timeout	:	900 sec (default)
Maximum connections	:	100000 (default)
Anonymous logon	:	IUSR_TTTICHHD (default Guest user name for NT)
Password authentication	:	Allow Anonymous (default) Windows NT Challenge/Response

The Directories tabbed box displays default directories set for www service. We can include extra directories for virtual web sites. For the default web service, the IIS has created directory C:\InetPub\wwwroot under which we should put all the documents for the default web site.

Using Logging tabbed box, we can set the logging properties. Here we can enable the logging of events by server. We can select log file format, the log file creation frequency, log file directory, or log the events in SQL database server if available.

Using Advanced tabbed box, we can restrict access to our web site to particular IP address or deny particular IP addresses to access the site.

## 5.2 FTP service Configuration:

To customize the settings of the default FTP site, select menus from Start – Program – Microsoft Internet Server (Common) – Internet Service Manager. To customize the service, double-click the computer name of FTP service. It opens the Properties dialog window for the selected service.

The FTP service properties for TTTICHD dialog box opens. The dialog box shows five tabbed boxes viz. Service, Messages, Directories, Logging, and Advanced, with currently selected Service tab box.

Using the Service tabbed box, we can set following properties :

TCP port	:	21 (default)
Connection Timeout	:	900 sec (default)
Maximum connections	:	1000 (default)
Anonymous logon	:	IUSR_TTTICHD (default Guest user name for NT)

Using Messages tabbed box, we can include entry and exit messages which will be displayed on user screen when he connects to our ftp site and when user finishes access and terminate the connection.

The Directories tabbed box displays directories set for FTP service. The default directory as set by server setup program is C:\inetpub\ftproot. We can include other directories for ftp access and assign virtual identification to prevent intruder spoofing. Also, from this tabbed page, we can set the properties for the selected directory for read/write access.

Using Logging tabbed box, we can set the server logging properties. Server will log the events in the log file which are very much helpful for performance tracking, detection of user activities, etc. We can select log file format, the log file creation frequency, log file directory, or log the events in SQL database server if available.

Using Advanced tabbed box, we can restrict access to our web site to particular IP address or deny particular IP addresses to access the site.

### 5.3 Sendmail Configuration

The Simple Mail Transfer Protocol (SMTP) is the established standard for transferring mail over the Internet. The sendmail program provides the services needed to support SMTP connections for Linux. To understand the jobs that sendmail performs, it is necessary to know little about Internet Protocols.

Protocols are simply agreed-upon standards that software and hardware use to communicate. Protocols are simply layered with higher levels using the lower ones as building blocks. For example, the Internet Protocol (IP) sends packets of data back and forth without building an end-to-end connection used by some of the higher level protocols. The Transmission Control Protocol (TCP), which is built on top of IP, provides for connection-oriented services such as those used by SMTP and telnet. TCP/IP protocols provide the basic network services for the Internet. Higher-level protocols such as the FTP and SMTP are built on top of TCP/IP. The advantage of such layering is that programs that implement the SMTP or FTP protocols don't have to know anything about transporting packets on the network and making connections to other hosts. They can use the services provided by TCP/IP for that job.

SMTP defines how programs exchange email on the Internet. It doesn't matter whether the program exchanging the email is sendmail or something else. As long as both programs implement the SMTP protocol correctly, they can exchange mail.

Sendmail can act as a mail router, final delivery agent, and SMTP client and server. However, it does not do final delivery of mail. Sendmail is primarily a router that inspects the recipient addresses, and decides the best way to send it. Sendmail determines some of the information it needs on its own, such as the current time and the name of the host on which it is running, but most of the other information it gets from a configuration file called `sendmail.cf`. This file is configured depending on how the organization wants sendmail to handle the various kinds of mails.

## General Form of the Configuration File

Each line of the configuration file begins with a single command character that indicates the function and syntax of that line. Line beginning with a # are comments, and blank lines are ignored. Lines beginning with a space or tab are continuations of the preceding line, however, spanning statements across multiple lines should usually be avoided. Following table shows the command characters and their functions.

Command Character	Command Syntax	Command function
#	# comment	A comment line.
D	DX string	Defines a macro X to have the value string
D	D{Macroname} value	Defines long macro {Macroname} to have the value, value, then referenced later with \${Macroname}.
C	CX word1, word2, ...	Defines a class X as word1, word2, and so on.
F	FX/path/to/a/file	Defines a class X by reading it from a file.
H	H?mailerflag?name: template	Defines a mailer header.
O	OX option arguments	Sets option X. Most command line options can be set in sendmail.cf
P	Pclass=nn	Sets mail delivery precedence based on the class of the mail.
V	Vn	Tells V.8 sendmail the version level of the configuration
K	Kname class argument	Defines a key file (database map)
M	Mname,field1=value1, field2=value2, ...	Defines a mailer
S	Snn	Begins a new rule set.
R	Rlhs rhs comment	Defines a matching/rewriting rule.

A configuration file does three things:

1. First, it sets the environment for sendmail by telling it which options to set and the locations of the files and databases it uses.
2. Second, a configuration file defines the characteristics of the mailers (delivery agents or MTAs) that sendmail uses after it decides where to route a letter. All configuration files must define local and program mailers to handle delivery to users on the local host.
3. Third, a configuration file specifies rulesets that rewrite sender and recipient addresses and select mailers. All rulesets are user-defined, but some have special meaning to sendmail.

## 5.4 Newsletter Subscription

Newsletter subscription is achieved using simple aliases file on linux mail server. A newsletter, plaintext (ASCII text codes) file will be prepared by administrator and will be delivered to all recipients listed in aliases file by sendmail.

## 5.5 Chat

A simple chat application will be developed in which a member can chat with any of the group. Three main servlets are designed as follows:

Administration Servlet: This servlet will be used by administrator to add a new chat room or delete an existing chat room.

Membership Servlet: This servlet is again used by administrator to add or remove the members. The Login names and passwords of members are stored in database and simple text format authentication is used to validate a member.

ListRooms Servlet: This servlet will list the currently available chat room in one of which the member can chat.

Chat Servlet: The servlet which will carry out actual message distribution among members during chat session.

## 5.6 Notice Board

Notice Board application is also developed using Servlets. The information about notice will be stored in database. The details of table attributes and their types are listed in design section. It uses two servlets as follows:

Administration Servlet: This servlet is used by an administrator to host notice on the notice board. The administrator can set the expiry date for the notice. After expiry date, the notice will be available in database but it will not be available to view for user.

NoticeBoard Servlet: This servlet will list the notices available on the notice board to user. The user can read the notices.

## 5.7 DNS Configuration

For every web site, it is necessary to install DNS service on any of the server connected directly to Internet. This server can be a proxy server. It is mandatory that the organization should maintain two name servers so that in situations where one is unavailable, the another can handle the requests. The main name server is called as Primary Name Server and the another supporting is called as Secondary Name Server.

On Windows NT server, it is easy to install and configure DNS service. To install DNS service on NT 4.0 server, steps are as follows:

1. Select menu Start – Settings - Control Panel.
2. Double-click Network icon. This opens Network dialog box.
3. Select Service tabbed page. Click Add button on this page.
4. Select Microsoft DNS service and click OK.

The DNS service is installed. Now to configure, follow the steps:

1. Select menu Start – Program – Administrative Tools (Common) – DNS Manager. This opens Domain Name Service Manager.
2. Now from the list of servers, select the name of the server which need to be configured as Primary name server.
3. Select menu DNS-New zone. It starts the New Zone wizard.
4. Select option Primary or Secondary using corresponding radio button. Click Next button.
5. Enter the name of New Zone and filename for new zone.
6. Click Finish button.

And the DNS is configured as Primary/Secondary name server as per selection.

## Domain Name Registration

The domain name of the portal will be registered with Indian Domain Name Registration Services. The Registration Form filled in as proposed by the host institute authorities is available as follows.

# INDIAN DOMAIN REGISTRATION SERVICES

## FORM FOR INTERNET DOMAIN REGISTRATION/MODIFICATION

### 1. Domain Name Information

#### 1.1 Registration Type

(Mention N for new domain, M for modification in existing domain or D for deleting an existing domain)

Type\* : N

#### 1.2 Fully Qualified Domain Name

(Currently we allot the following sub-domains

- co.in - For Registered Companies/Trademarks/Banks
- firm.in - For Proprietary Concerns/Partnership Firms/Shops/Liaison Offices
- ac.in - For Academic Community
- res.in - For Research Institutes
- gov.in - For Government Organisations
- mil.in - For Military Establishments
- net.in - For Internet Service Providers
- in - For Internet Service Providers
- org.in - For Non-profit Organisations
- ind.in - For Individuals
- gen.in - For General/Miscellaneous Purpose

Mention the domain that you would like to register.

eg: your\_company\_name.co.in, etc)

Domain Name \* : ttichd.ac.in.

### 2. General Details:

#### 2.1 Name of Organisation/Institution/Individual

Name \* : TECHNICAL TEACHERS' TRAINING INSTITUTE

#### 2.2 Address of Organisation/Institution/Individual

Address Line 1 \* : SECTOR-26  
Address Line 1 \* :  
City \* : CHANDIGARH  
State \* : UNION TERRITORY  
Pin Code\* : 160019  
Country\* : INDIA

2.3 Administrative Contact (from your organisation only)

Name \* : DR.S.KRISHNAMURTY  
Designation : PRINCIPAL  
E-mail Address\* : OR  
Phone Number : (0172)792369  
Fax Number : (0172)791368

2.4 Billing Contact

Name \* : B.B.SHARMA  
Designation : ACCOUNTS OFFICER  
Address Line 1 \* : TECHNICAL TEACHERS' TRAINING INSTITUTE,  
Address Line 2 \* : SECTOR-26  
City \* : CHANDIGARH  
State \* : UNION TERRITORY  
Pin Code\* : 160019  
Country\* : INDIA  
E-mail Address\* :  
Phone Number : (0172)791349 Ext-390  
Fax Number : (0172)791368

2.5 Technical Contact

Name \* : DR.RENU VIG  
Designation : ASSISTANT PROFESSOR,  
COMPUTER SCIENCE DEPARTMENT  
Address Line 1 \* : TECHNICAL TEACHERS' TRAINING INSTITUTE,  
Address Line 2 \* : SECTOR-26  
City \* : CHANDIGARH  
State \* : UNION TERRITORY  
Pin Code\* : 160019  
Country\* : INDIA  
E-mail Address\* : OR renuvig@tttichd.ac.in  
Phone Number : (0172)791349 Ext-495  
Fax Number : (0172)791368

3. Description of your Organisation/Trademark/Domain

Description : TECHNICAL TEACHERS' TRAINING INSTITUTE  
ESTABLISHED BY MINISTRY OF HUMAN RESOURCES  
DEVELOPMENT IN 1967.

4. Name of your Internet Service Provider/Web Service Provider

ISP/WSP : PUNJAB COMMUNICATIONS LIMITED, MOHALI.

5. Name Servers

5.1 Primary Name Server

Host name (with domain name) : NS1 (TTTICHD)  
Host IP Address \* : 203.197.15.168

5.2 Secondary Name Servers

(Please duplicate the below fields if you have more than one secondary name server)

Host name (with domain name) : NS2 (CSD\_TTTICHD)  
Host IP Address \* : 203.197.15.162

## Appendix A

### RFC-1123 Requirement for Internet Hosts

The following information is derived from RFC 1123 an official specification for the Internet Community. This RFC is one of a pair that defines and discusses the requirements for Internet host software. The RFC-1123 covers the application and support protocols; its companion RFC-1122 covers the communication protocol layers; link layer, IP layer, and transport layer.

The official publication of this RFC contains discussion of specifications. However, here only the protocol specifications are listed for reference. (The sub-numbering used is copied as it is in RFC-1123 specification to keep consistency for reference).

#### **REMOTE LOGIN – TELNET PROTOCOL**

Telnet is the standard Internet application protocol for remote login. It provides the encoding rules to link a user's keyboard/display on a client ("user") system with a command interpreter on a remote server system. A subset of the Telnet protocol is also incorporated within other application protocols, e.g., FTP and SMTP.

Telnet uses a single TCP connection, and its normal data stream ("Network Virtual Terminal" or "NVT" mode) is 7-bit ASCII with escape sequences to embed control functions. Telnet also allows the negotiation of many optional modes and functions.

The primary Telnet specification is to be found in RFC-854 [TELNET:1], while the options are defined in many other RFCs.

- 3.2.1 **Option Negotiation** : Every Telnet implementation MUST include option negotiation and subnegotiation machinery [TELNET:2].  
A host MUST carefully follow the rules of RFC-854 to avoid option-negotiation loops. A host MUST refuse (i.e., reply WONT/DON'T to a DO/WILL) an unsupported option. Option negotiation SHOULD continue to function (even if all requests are refused) throughout the lifetime of a Telnet connection.  
If all option negotiations fail, a Telnet implementation MUST default to, and support, and NVT.
- 3.2.2 **Telnet Go-Ahead Function** : On a host that never sends the Telnet command Go Ahead (GA), the Telnet Server MUST attempt to negotiate the Suppress Go Ahead option (i.e., send "WIL Suppress Go Ahead"). A User or Server Telnet MUST always accept negotiation of the Suppress Go Ahead option.  
When it is driving a full-duplex terminal for which GA has no meaning, a User Telnet implementation MAY ignore GA commands.
- 3.2.3 **Control Functions** : The list of Telnet commands has been extended to include EOR (End-of-Record), with code 239 [TELNET:9].  
Both User and Server Telnets MAY support the control functions EOR, EC, EL, and Break, and MUST support AO, AYT, DM, IP, NOP, SB, and SE.

A host **MUST** be able to receive and ignore any Telnet control functions that it does not support.

3.2.4 **Telnet :** When it receives "urgent" TCP data, a User or Server Telnet **MUST** discard all data except Telnet commands until the DM (and end of urgent) is reached.

When it sends Telnet IP (Interrupt Process), a User Telnet **SHOULD** follow it by the Telnet "Synch" sequence, i.e., send as TCP urgent data the sequence "IAC IP IAC DM". The TCP urgent pointer points to the DM octet.

When it receives a Telnet IP command, a Server Telnet **MAY** send a Telnet "Synch" sequence back to the user, to flush the output stream. The choice ought to be consistent with the way the server operating system behaves when a local user interrupts a process.

When it receives a Telnet AO command, a Server Telnet **MUST** send a Telnet "Synch" sequence back to the user, to flush the output stream.

A User Telnet **SHOULD** have the capability of flushing output when it sends a Telnet IP.

3.2.5 **NVT Printer and Keyboard :** In NVT mode, a Telnet **SHOULD NOT** send characters with the high-order bit 1, and **MUST NOT** send it as parity bit. Implementations that pass the high-order bit to applications **SHOULD** negotiate binary mode.

3.2.6 **Telnet Command Structure :** Since options may appear at any point in the data stream, a Telnet escape character (known as IAC, with the value 255) to be sent as data **MUST** be doubled.

3.2.7 **Telnet Binary Option :** When the Binary option has been successfully negotiated, arbitrary 8-bit characters are allowed. However, the data stream **MUST** still be scanned for IAC characters, any embedded Telnet commands **MUST** be obeyed, and data bytes equal to IAC **MUST** be doubled. Other character processing (e.g., replacing CR by CR NUL or by CR LF) **MUST NOT** be done. In particular, there is no end-of-line convention in binary mode.

3.2.8 **Telnet Terminal-Type Option :** The Terminal-Type option **MUST** use the terminal type names officially defined in the Assigned Numbers RFC [INTRO:5], when they are available for the particular terminal. However, the receiver of a Terminal-Type option **MUST** accept any name.

## **FILE TRANSFER PROTOCOL**

The File Transfer Protocol FTP is the primary Internet standard for file transfer. The current specification is contained in RFC-959 [FTP:1].

FTP uses two separate simultaneous TCP connections for control and for data transfer. The FTP protocol includes many features, some of which are not commonly implemented. However, for every feature in FTP, there exists at least one

implementation. The minimum implementation defined in RFC-959 was too small, so a somewhat larger minimum implementation is defined in this RFC.

- 4.1.2.1 **LOCAL Type:** An FTP program must support TYPE I ("IMAGE" or binary type) as well as TYPE L 8 ("LOCAL" type with logical byte size 8). A machine whose memory is organized into m-bit words, where m is not a multiple of 8, MAY also support TYPE L m.
- 4.1.2.2 **Telnet Format Control:** A host that makes no distinction between TYPE N and TYPE T SHOULD implement TYPE T to be identical to TYPE N.
- 4.1.2.3 **Page Structure :** Implementation of page structure is NOT RECOMMENDED in general. However, if a host system does need to implement FTP for "random access" or "holey" files, it MUST use the defined page structure format rather than define a new private FTP format.
- 4.1.2.4 **Data Structure Transformation :** An FTP transformation between record-structure and file-structure SHOULD be invertible, to the extent possible while making the result useful on the target host.
- 4.1.2.5 **Data Connection Management :** A User-FTP that uses STREAM mode SHOULD send a PORT command to assign a non-default data port before each transfer command is issued.
- 4.1.2.6 **PASV Command :** A server-FTP MUST implement the PASV command. If multiple third-party transfers are to be executed during the same session, a new PASV command MUST be issued before each transfer command, to obtain a unique port pair.
- 4.1.2.7 **LIST and NLST Commands :** The data returned by an NLST command MUST contain only a simple list of legal pathnames, such that the server can use them directly as the arguments of subsequent data transfer commands for the individual files. The data returned by a LIST or NLST command SHOULD use an implied TYPE AN, unless the current type is EBCDIC, in which case an implied TYPE EN SHOULD be used.
- 4.1.2.8 **SITE Command :** A Server-FTP SHOULD use the SITE command for non-standard features, rather than invent new private commands or unstandardized extensions to existing commands.
- 4.1.2.9 **STOU Command :** The STOU command stores into a uniquely named file. When it receives an STOU command, a Server-FTP MUST return the actual file name in the "125 Transfer Starting" or the "150 Opening Data Connection" message that precedes the transfer.

4.1.2.10 **Telnet End-of-line Code** : Implementors MUST NOT assume any correspondence between READ boundaries on the control connection and the Telnet EOL sequence (CR LF).

4.1.2.11 **FTP Replies** : A Server-FTP MUST send only correctly formatted replies on the control connection. Note that RFC-959 contains no provision for a "spontaneous" reply message.

A server-FTP SHOULD use the reply codes defined in RFC-959 whenever they apply. However, a server-FTP MAY use a different reply code when needed, as long as the general rules of Section 4.2 are followed. When the implementers has a choice between 4xx and 5xx reply code, a Server-FTP SHOULD send 4xx (temporary failure) code when there is any reasonable possibility that a failed FTP will succeed a few hours later.

A User-FTP SHOULD generally use only the highest-order digit of a 3-digit reply code for making a procedural decision, to prevent difficulties when a Server-FTP uses non-standard reply codes.

A User-FTP MUST be able to handle multi-line replies. If the implementation imposes a limit on the number of lines and if this limit is exceeded, the User-FTP MUST recover, e.g., by ignoring the excess lines until the end of the multi-line reply is reached.

A User-FTP SHOULD NOT interpret a 421 reply code ("Service not available, closing control connection") specially, but SHOULD detect closing of the control connection by the server.

4.1.2.12 **Connections** : The words "and the port used" in the second paragraph of this section of RFC-959 are erroneous (historical), and they should be ignored

On a multihomed server host, the default data transfer port (L-1) MUST be associated with the same local IP address as the corresponding control connection to port L.

A user-FTP MUST NOT send any Telnet controls other than SYNCH and IP on an FTP control connection. In particular, it MUST NOT attempt to negotiate Telnet options on the control connection. However, a server-FTP MUST be capable of accepting and refusing Telnet negotiations (i.e. sending DON'T/WONT).

4.1.2.13 **Minimum Implementation**: The following commands and options MUST be supported by every server-FTP and user-FTP, except in cases where the underlying file system or operating system does not allow or support a particular command.

Type : ASCII Non-print, IMAGE, LOCAL 8

Mode : Stream

Structure : File, Record\*

Commands:

USER,	PASS,	ACCT,
PORT,	PASV,	
TYPE,	MODE,	STRU,

RETR,	STOR,	APPE,		
RNFR,	RNTO,	DELE,		
CWD,	CDUP,	RMD,	MKD,	PWD,
LIST,	NLST,			
SYST,	STAT,			
HELP,	NOOP,	QUIT.		

\* Record structure is REQUIRED only for hosts whose file systems support record structure.

## SIMPLE MAIL TRANSFER PROTOCOL

This section covers both RFC-821 and RFC-822.

### 5.2.1 The SMTP Model:

Mail is sent by a series of request/response transactions between a client, the "sender-SMTP," and a server, the "receiver-SMTP". These transactions pass (1) the message proper, which is composed of header and body, and (2) SMTP source and destination addresses, referred to as the "envelop".

The SMTP programs are analogous to Message Transfer Agents (MTAs) of X.400. There will be another level of protocol software, closer to the end user, that is responsible for composing and analyzing RFC-822 message headers; this component is known as the "User Agent" in X.400, and that term is used in this document. There is a clear logical distinction between the User Agent and the SMTP implementation, since they operate on different levels of protocol. Note, however, that this distinction is may not be exactly reflected the structure of typical implementations of Internet mail. Often there is a program known as the "mailer" that implement SMTP and also some of the User Agent functions; the rest of the User Agent functions are included in a user interface used for entering and reading mail.

The SMTP envelope is constructed at the originating site, typically by the User Agent when the message is first queued for the Sender-SMTP program. The envelop addresses may be derived from information in the message header, supplied by the user interface (e.g., to implement a bcc: request), or derived from local configuration information (e.g., expansion of a mailing list). The SMTP envelop cannot in general be re-derived from the header at a later stage in message delivery, so the envelop is transmitted separately from the message itself using the MAIL and RCPT commands of SMTP.

The text of RFC-821 suggests that the mail be to be delivered to an individual user at a host. With the advent of the domain system and of mail routing using mail-exchange (MX) resource records, implementors should now think of delivering mail at a user at a domain, which may or may not be a particular host. This DOES NOT change the fact that SMTP is a host-to-host mail exchange protocol.

**5.2.2 Canonicalization :** The domain names that a sender-SMTP sends in MAIL and RCPT commands have been "canonicalized," i.e., they must be fully-qualified

principal names or domain literals, not nicknames or domain abbreviations. A canonicalized name either identifies a host directly or is an MX name, it cannot be a CNAME.

**5.2.3 VERY and EXPN Commands :** A receiver-SMTP MUST implement VERY and SHOULD implement EXPN (this requirement overrides RFC-821). However, there MAY be configuration information to disable VRFY and EXPN in a particular installation; this might even allow EXPN to be disabled for selected lists. A new reply code is defined for VRFY command:

252 Cannot VRFY user (e.g., info is not local), but will take message for this user and attempt delivery.

**5.2.4 SEND, SOMI, and SAML Commands :** An SMTP MAY implement the commands to send a message to a user's terminal: SEND, SOMI, and SAML.

**5.2.5 HELO Command :** The sender-SMTP MUST ensure that the <domain> parameter in a HELO command is a valid principal host domain name for the client host. As a result, the receiver-SMTP will not have to perform MX resolution on this name in order to validate the HELO parameter.

The HELO receiver MAY verify that the HELO parameter really corresponds to the IP address of the sender. However, the receiver MUST NOT refuse to accept a message, even if the sender's HELO command fails verification.

**5.2.6 Mail Relay :** We distinguish three types of mail (store-and-) forwarding :

1. A simple forwarder or "mail exchanger" forwards a message using private knowledge about the recipient (Refer section 3.2 of RFC-821).
2. An SMTP mail "relay" forwards a message within an SMTP mail environment as the result of an explicit source route (as defined in section 3.6 of RFC-821). The SMTP relay function uses the "@..." form of source route from RFC-822.
3. A mail "gateway" passes a message between different environments. (Refer Section 5.3.7 The rules for mail gateways).

An Internet host that is forwarding a message but is not a gateway to a different mail environment (i.e., it falls under (1) or (2)) SHOULD NOT alter any existing header fields, although the host will add an appropriate Received: line as required in Section 5.2.8.

A Sender-SMTP SHOULD NOT send a RCPT TO: command containing an explicit source route using the "@..." address form. Thus, the relay function defined in section 3.6 of RFC-821 should not be used.

**5.2.7 RCPT Command :** A host that supports a receiver-SMTP MUST support the reserved mailbox "Postmaster". The received-SMTP MAY verify RCPT parameters as they arrive; however, RCPT responses MUST NOT be delayed beyond a reasonable time.

Therefore, a "250 OK" response to a RCPT does not necessarily imply that the delivery address(es) are valid. Errors found after message acceptance will be reported by mailing a notification message to an appropriate address.

5.2.8 **DATA Command** : Every receiver-SMTP (not just one that "accepts a message for relaying or final delivery" [SMTP:1] MUST insert a "Received:" line at the beginning of a message. In this line, called a "time stamp line" in RFC-821:

- The FROM field SHOULD contain both (1) the name of the source host as presented in the HELO command and (2) a domain literal containing the IP address of the source, determined from the TCP connection.
- The ID field MAY contain an "@" as suggested in RFC-822, but this is not required.
- The FOR field MAY contain a list of <path> entries when multiple RCPT commands have been given.

An Internet mail program MUST NOT change a Received: line that was previously added to the message header.

5.2.9 **Command Syntax** : The syntax shown in RFC-821 for the MAIL FROM: command omits the case of an empty path: "MAIL FROM: <>" (Refer RFC-821 Page 15). An empty reverse path MUST be supported.

5.2.10 **SMTP Replies** : A receiver-SMTP SHOULD send only the reply codes listed in section 4.2.2 of RFC-821 or in this document. A receiver-SMTP SHOULD use the text shown in RFC-821 whenever appropriate

A sender-SMTP MUST determine its actions only by the reply code, not by the text (except for 251 and 551 replies); any text, including no text at all, must be acceptable. The space (blank) following the reply code is considered part of the text. Whenever possible, a sender-SMTP SHOULD text only the first digit of the reply code, as specified in Appendix E of RFC-821.

5.2.11 **Transparency** : implementors MUST be sure that their mail system always add and delete periods to ensure message transparency.

5.2.12 **WKS Use in MX Processing** : RFC-974 [SMTP:3] recommended that the domain system be queried for WKS ("Well-Known Service") records, to verify that each proposed mail target does support SMTP. Later experience has shown that WKS is not widely supported, so the WKS step in MX processing SHOULD NOT be used.

5.2.13 **RFC-822 Message Specification** : The syntax shown for the Return-path line omits the possibility of a null return path, which is used to prevent looping of error notifications. The complete syntax is:

```
return = "Return-path" ":" route-addr
        / "Return-path" ":" "<" ">"
```

The set of optional header fields is hereby expanded to include the Content-Type field defined in RFC-1049 [SMTP:7]. This field "allows mail reading systems to automatically identify the type of a structured message body and to process it for display accordingly". [SMTP:7] A User Agent MAY support this field.

5.2.14 **RFC-822 Date and Time Specification :** The syntax for the date is hereby changed to:

date = 1\*2DIGIT month 2\*4DIGIT

All mail software SHOULD use 4-digit years in dates, to ease the transition to the next century.

There is a strong trend towards the use of numeric timezone indicators, and implementations SHOULD use numeric timezone names. However, all implementations MUST accept either notation. If timezone names are used, they MUST be exactly as defined in RFC-822.

The military time zones are specified incorrectly in RFC-822: they count the wrong way from UT (the signs are reserved). As a result, military time zones in RFC-822 headers carry no information.

Finally, note that there is a typo in the definition of "zone" in the syntax summary of appendix D; the correct definition occurs in Section 3 of RFC-822.

5.2.15 **RFC-822 Syntax Change :** The syntactic definition of "mailbox" in RFC-822 is hereby changed to:

mailbox = addr-spec ; simple address  
/ [phrase] route-addr ; name & addr-spec

That is, the phrase preceding a route address is now OPTIONAL. This change makes the following header field legal, for example:

From: [craig@nnsf.net](mailto:craig@nnsf.net)

5.2.16 **RFC-822 Local-part :** The basic mailbox address specification has the form: local-part@domain. Here "local-part", sometimes called the "left-hand side" of the address, is domain dependent.

A host that is forwarding the message but is not the destination host implied by the right-hand side "domain" MUST NOT interpret or modify the "local-part" of the address.

When mail is to be gatewayed from the Internet mail environment into a foreign mail environment, routing information for that foreign environment MAY be embedded within the "local-part" of the address. The gateway will then interpret this local part appropriately for the foreign mail environment.

5.2.17 **Domain Literals :** A mailer MUST be able to accept and parse an Internet domain literal whose content ("dtext"; refer RFC-822) is a dotted-decimal host address. This satisfies the requirement of Section 2.1 for the case of mail. An SMTP MUST accept and reorganize a domain literal for any of its own IP address.

5.2.18 **Common Address Formatting Errors :** Errors in formatting or parsing 822 addresses are unfortunately common. This section mentions only the most common errors. A User Agent **MUST** accept all valid RFC-822 address formats, and **MUST NOT** generate illegal address syntax.

- A common error is to leave out the semicolon after a group identifier.
- Some systems fail to fully-qualified domain names in messages they generate. The right-hand side of an "@" sign in a header address field **MUST** be a fully-qualified domain name.

For example, some systems fail to fully-qualify the From: address; this prevents a "reply" command in the user interface from automatically constructing a return address.

5.2.19 **Explicit Source Routes :** Internet host software **SHOULD NOT** create an RFC-822 header containing an address with an explicit source route, but **MUST** accept such headers for compatibility with earlier systems.

## References

1. Bruce Eckel, Thinking in Java, Addison-Wesley Publication.
2. Christopher L.T.Brown and Scott Zimmerman, Web Site Construction Kit for Windows NT, Sams.net Publication.
3. David Pitts and Bill Ball, Red Hat Linux 6.0, SAMS Techmedia Publication.
4. Dee-Ann LeBlanc, Running A Perfect Internet Site with Linux, Que Publication.
5. Herbert Schildt and Patrick Naughton, Java2 Complete Reference, Tata McGraw Hill Publication.
6. Internet Explorer Complete, BPB publication.
7. Ivan Bayross, HTML, DHTML, JavaScript, Perl CGI, BPB Publication.
8. James Goodwill, Developing Java Servlets, SAMS Techmedia Publication.
9. Jason Hunter, William Crawford, Java Servlet Programming, O'Reilly/SPD Publication.
10. Mark Minasi, Mastring Windows NT 4.0 Server, BPB Publication.
11. Richard Wagner and Allen Wyke, JavaScript, SAMS Techmedia Publication.
12. Wrox Author Team, Java Server Pages, Wrox Publication.
13. Wrox Author Team, Java Server Programming, Wrox Publication.

**RFC:** RFC-1123 – Requirement for Internet Hosts – Applications and Support.

### INTERNET SITES:

1. [java.sun.com](http://java.sun.com)
2. [www.borland.com](http://www.borland.com)
3. [www.google.com](http://www.google.com)
4. [www.howstuffworks.com](http://www.howstuffworks.com)
5. [www.microsoft.com](http://www.microsoft.com)