

## *INDEX*

### **S.NO**

### **CONTENTS**

1. APPLE REMOTE DESKTOP(ARP)
2. APPLE SECURITY UPDATES
3. FAQ
4. ORACLE
5. NEWLY DISCOVERED VIRUSES
6. TOP FIVE VIRUSES
7. VPN
8. RPD;D RPD; D MIRFS;
9. HARDWARE COMPONENTS
10. NEW LINUX SERVER ANDDESKTOP
11. ACTIVE SERVER PAGE
12. ASP.NET
13. ABBREVIATION

## *Apple Remote Desktop (ARD)*

### **Remote Assistance**

- ? Full Screen mode: You can now control and observe remote computers in full screen mode, allowing you to view the remote screen on the entire screen of your display.
- ? Control and observe remote computers with multiple monitors: Administrators can now control and observe remote Mac OS X systems which have two or more displays. The displays are shown in one virtual window.
- ? Support for mouse scroll wheel and right-click when controlling remote computers.
- ? Improved control and observation of remote computers that are running applications in full screen.
- ? Improved opt-out for Guest Access option.
- ? Improved support for third party VNC Viewers and VNC servers.
- ? AppleVNC Server no longer relies on presence of MDNS process.

### **Remote Administration**

- ? Multi-line output for Send UNIX Command: You can now view all the results from the Send UNIX Command.
- ? The Install Package command now detects whether a package requires a restart and will optionally restart the client computers after the package installation has completed.

- ? Send UNIX Command no longer dependent on client side.  
Observe privilege.

## **Asset Management**

- ? Improved printing of hardware and software reports.
- ? Improved remote data collection.
- ? You can now drag results from reports to the Finder. The items will then be copied from the client to the administrator's computer.

## **Software Distribution**

- ? Improved file copy for files that reside on networked home directories.

## **Other Enhancements**

- ? Client authentication using Active Directory: Clients can now be authenticated against directory services groups in Active Directory.
- ? Improved client authentication using two additional directory services groups, ard\_manage and ard\_interact. Members of ard\_manage groups will be able to perform all commands except for control and observe. members of the ard\_interact group will be able to perform all commands under the interact menu.
- ? Enhanced column sorting in the Administration application.

- ? Improved support for managing computers that have dynamic DNS enabled.
- ? Administration application now correctly remembers its settings if it unexpectedly quits.
- ? The ordering of network scanners is now remembered.
- ? Includes security fix present in Security update 2004-10-27.

### *Apple Security Updates*

<b>Name and Information link</b>	<b>Released</b>	<b>Release Date</b>
	Mac OS X 10.4.1	
Security Update 2005 ? 2006	Mac OS X Server 10.4.1 Mac OS X 10.3.9	08 June 2005
	Mac OS X Server 10.3.9	
Quick Time 7.0.1	Quick Time 7	31 May 2005
	Keynote 2.0	
Keynote 2.0.2 update	Keynote 2.0.1	25 May 2005
	Mac OS X 10.4	16 May 2005 ( Client )
Mac OS X 10.4.1 update	Mac OS X 10.4 Server	19 May 2005 ( Server )
	Mac OS X 10.2.8 or later	
tunes 4.8	Windows XP	09 May 2005
	Windows 2000	
Security Update 2005	Mac OS X 10.3.9 Mac OS X Server 10.3.9	03 May 2005
Security Update 2004? 2005	iSync 1.5	19 Apr 2005
Mac OS X 10.3.9	Mac OS X 10.3 through	21 Mar 2005

Update	10.3.8	
Security Update	Mac OS X 10.3.8	21 Mar 2005
2003 ? 2005	Mac OS Z Server 10.3.8	
Security Update	Java 1.4.2	22 Feb 2005
2002 ? 2005		
Mac OS X 10.3.8 Update	Mac OS X 10.3 through 10.3.7	09 Feb 2005
	Mac OS X 10.3.7	
Security Update	Mac OS X Server 10.3.7	25 Jan 2005
2001 ? 2005	Mac OS X 10.2.8	
	Mac OS X Server 10.2.8	

B Priya,  
K P Keerthana,  
III BCA.

## *FAQ*

### **What is a firewall?**

- When your computer is connected to the Internet, it receives traffic from a wide range of sources, most of it benign. Your instant messaging client alerts you that a friend has signed on; your mail client finds new mail waiting for you and downloads it; a weather site refreshes its rainfall map by telling your web browser to reload a page. All of this traffic is handled invisibly by your computer, which is listening to a large number of "ports." A port is a specific connection point through which applications on your computer connect to the Internet. And a hacker only needs one open port through which to mount an attack.
- A firewall is a piece of software that monitors all incoming network traffic and allows in only the connections that are known and trusted.

- You could manually grant or restrict access to each of the 65,535 ports available under the Internet Protocol. Every time you add a new program that requires Internet access, you would need to determine which port(s) it uses, and reconfigure your computer accordingly. There are better ways to spend your time.
- Firewall software takes on this burden for you, allowing access to the ports you need open, and closing off those you don't. It also makes your computer "invisible" on the Internet; if hackers can't find you, they will have a hard time attacking you.
- More advanced firewall software also monitors outgoing traffic. This is crucial since malicious code spreads by accessing the Internet and pushing copies of itself to other computers (often those of your friends and family!). Outbound protection can keep even brand-new Trojan horses and spy-ware from doing their damaging work. The ultimate protection is program-level control, so that only those applications that you trust are allowed to access the Internet.

### **Why do I need a firewall?**

- Without a firewall, your computer is operating under an "open door" policy. Bank account information. Passwords. Credit card numbers. Documents and photos that you don't want to share with the world. They are all available to anyone with bad intentions and basic computer skills. Hackers can get in, take what they want, and even leave open a "back door" so they use it to attack other computers.
- Every minute that your computer is connected to the Internet, either through a dial-up (modem) connection or through a broadband (DSL or cable) service, it is at risk.

### **What Is a Computer Virus?**

- A computer virus is a program that has the unique ability to replicate. Like biological viruses, computer viruses can spread quickly and are often difficult to eradicate. They can attach themselves to just about any type of file and are spread as files that are copied and sent from individual to individual.
- Besides replication, some computer viruses have something else in common: a "damage routine", i.e. a piece of software, that can deliver

the virus payload. While payloads may only display messages or images, they can also destroy files, reformat your hard drive, or cause other kinds of damage. If the virus doesn't contain a damage routine, it can still cause trouble by taking up storage space and memory, and downgrading the overall performance of your computer.

### **Do I really need an Anti-virus?**

- Virtually, no one who uses Windows-based computers is immune from viruses. Every time your computer handles a new file, chances are that it could be infected. In particular, that is the case when you open attachments in your emails, when you download a program from the Internet or when you copy a file from one computer to another. Unless your computer is completely isolated from the outside world, which would make it pretty useless, the risk of infections are quite high.
- There are 60,000 to 70,000 active viruses and worms, with 10,000 new ones appearing every year. Nearly all are designed to infect Windows PCs. That means that Mac users don't need anti-virus protections because their machines can not run any program written for Windows, including viruses.
- At the moment, it's safe to say that a Mac user would need an anti-virus only when a Windows emulator like Virtual PC is used. That would be required to protect the Windows environment only, because Windows-viruses can not run and damage OS X.

### **What is Ad-ware?**

- "Ad-ware" or "advertising-supported software" is any software application in which advertisements are displayed while the program is running. These applications include additional code that displays the ads in pop-up windows or through a bar that appears on a computer screen. Ad-ware helps recover programming development costs, and helps to hold down the price of the application for the user (even making it free of charge)--and, of course, it can give programmers a profit, which helps to motivate them to write, maintain, and upgrade valuable software.

- Some ad-ware is also "shareware", in that users are given the option to pay for a "registered" or "licensed" copy, which typically does away with the advertisements.
- Some ad-ware programs have been criticized for occasionally including code that tracks a user's personal information and passes it on to third parties, without the user's authorization or knowledge. This practice has been dubbed "spy-ware".

### **Is Ad-ware dangerous?**

- While Ad-ware may be a great concept, the downside is that very often the advertising companies also install additional tracking software on your system, which is continuously "calling home", using your Internet connection and reports statistical data to the "mother ship". While according to the privacy policies of the companies, there will be no sensitive or identifying data collected from your system and you shall remain anonymous, it still remains the fact, that you have a "live" server sitting on your PC that is sending information about you and your surfing habits to a remote location.
- It is safe to say that not all ad-ware products are spy-ware, but the vast majority are. When ad-ware becomes spy-ware your privacy is at risk.

### **What is Spyware?**

- Strictly speaking, "spy-ware" is computer software that gathers information about a computer user without the user's knowledge or informed consent, and then transmits this information to an organization that expects to be able to profit from it in some way. Data-collecting programs installed with the user's knowledge are not, technically speaking, spy-ware, if the user fully understands what data is being collected and with whom it is being shared.
- More broadly, the term spy-ware is applied to a wide range of related "mal-ware" products which are not spy-ware in the strict sense. These products perform many different functions, including the delivery of unrequited advertising (pop-ups in particular), harvesting private information, re-routing page requests to illegally claim commercial site referral fees, hijacking the web browser homepage and installing stealth phone dialers.

## **What is a Trojan Horse?**

- A Trojan Horse is a particular category of spy-ware. A Trojan is a program that infects your computer and allows a hacker to take control of your machine behind your back. A Trojan infection can allow total remote access to your computer by a third party.
- Unlike virus and worms, Trojans do not replicate themselves so to get infected you must, one way or another, have downloaded the program onto your computer. This most commonly occurs when you download a program that pretends to be one thing while it is actually another. Hence the origin of the "Trojan" name.
- The most common types of Trojans include such functions as the ability to steal all passwords cached or not (this is done using key logging technology), run files, do serious damage to your machine and do pretty much whatever the intruder wants.
- However these days most Trojans are being used to turn your computer into a Zombie. In other words the hacker will be able to use your computer and a bunch of other infected computers like yours and turn them into an army of zombies to attack a more important target. This way the intruder can hide his or her own trace and use your computer as a front. This means the victim's firewall will show your Internet address in the log as the attacker's address and you may end up with several complaint letters to your ISP and even lose your account as a result.
- So just because you think you have nothing important on your computer doesn't mean your computer itself and your bandwidth is not valuable to hackers. In fact they are quite valuable and worth the hacker's time.

## **How can Spyware end up in my computer?**

- Spyware is normally installed through either one of two common methods. The first is to hide a spy-ware component within an otherwise apparently useful program. Often, the containing program is made available for download free of charge, so as to encourage wide uptake of the spy-ware component. The second common method is to take advantage of security flaws in Internet Explorer. Sometimes they

arrive as an automatic download from a website you are surfing. Spyware can also be installed on a computer by a virus or an e-mail Trojan program, but this is not common.

- The HTTP cookie (e.g. a packet of information) is a well-known mechanism for storing information about an Internet user on their own computer, often used to assign website visitors an individual identification number for subsequent recognition. However, the existence of cookies and their use is generally not concealed from users, who can also disallow access to cookie information. Nevertheless, to the extent that a Web site uses a cookie identifier to build a profile about the user, who does not know what information is added to this profile, the cookie mechanism could be considered a form of spy-ware. For example, a search engine website could assign a user an individual ID the first time he visits and store all search terms in a database with this ID as a key on all subsequent visits (until the cookie expires or is deleted). This data could be used to select advertisements to display to that user, or could?legally or illegally?be transmitted to third parties.
- Another cause is granting permission for web-based applications to integrate into your system. These "browser helper" object embeds itself as part of your web browser.
- Spyware is usually installed by some stealthy means. If you read the user agreements for the software you download and install, references (sometimes vague) are cited for allowing the issuing company of the software to record your Internet usage and website surfing. Some software vendors allow you to buy the same product without this overhead.

### **What are the consequences of Spyware activity?**

- Unprotected Windows-based computers, particularly those used by children or credulous adults, can rapidly accumulate a great many spy-ware components?several hundred individual instances is common. The consequences of a moderate to severe spy-ware infection (privacy issues aside) generally include a substantial loss of system performance (over 50% in severe cases), and major stability

issues (crashes and hangs). Difficulty connecting to the Internet is another common symptom.

- Spyware infection is now (as of 2004) responsible for more visits to professional computer repairers than any other single cause. In more than half of these cases, the user is unaware of the spy-ware problem and initially assumes that the system performance, stability, and/or connectivity issues are related to hardware, Windows installation problems, or a virus.
- Some spy-ware products have additional consequences. Dialers attempt to connect directly to a particular telephone number rather than to the user's own Internet Service Provider: where the number in question is overseas, this can result in massive telephone bills which the user has no choice but to pay.
- Much worse than your computer performances, is the violation of your privacy. The very essence of spy-ware is to collect as much information as possible to identify your behaviors for marketing purposes. The websites you visit and the time of the day when you browse the Internet are only a couple of examples. As a matter of fact, there is no limit to the type of information that spy-ware can be programmed to transmit.
- Particularly worrisome spy-ware programs are the so called "key-loggers". They log keystrokes and mouse clicks on the computer where they are installed and write them to a file. Usually they have the option of encrypting and decrypting the log files and the option of sending the file to a destination across the Internet. Software key-loggers, as opposed to hardware's, are practically impossible to track once installed. However, key logging can be effectively prevented by adopting good security practices. While key-loggers have many perfectly legal applications, their common utilization in espionage speaks volumes about the implications on the target's privacy.

### **Do I need an Anti-Spyware?**

- If you are very careful not to use your computer in any way that could allow your identification or if you don't care about your privacy, you certainly don't need it. Millions of people are using advertising

supported "spyware" products and could not care less about the privacy hype; in fact, some "Spyware" programs are among the most popular downloads on the Internet.

- On the other side, if you use your own identity while browsing the Internet, shop on-line using your credit card, use Internet banking or have at least one email account that could identify you, just to mention but a few examples, it really comes down to how much you value your privacy.

## Contents

1. Introduction to links and anchors
  1. Visiting a linked resource
  2. Other link relationships
  3. Specifying anchors and links
  4. Link titles
  5. Internationalization and links
2. The A element
  1. Syntax of anchor names
  2. Nested links are illegal
  3. Anchors with the id attribute
  4. Unavailable and unidentifiable resources
3. Document relationships: the LINK element
  1. Forward and reverse links
  2. Links and external style sheets
  3. Links and search engines
4. Path information: the BASE element
  1. Resolving relative URIs

## Introduction to links and anchors

HTML offers many of the conventional publishing idioms for rich text and structured documents, but what separates it from most other markup languages is its features for hypertext and interactive documents. This section introduces the *link* (or hyperlink, or Web link), the basic hypertext construct. A link is a connection from one Web resource to

another. Although a simple concept, the link has been one of the primary forces driving the success of the Web.

A *link* has two ends -- called *anchors* -- and a direction. The link starts at the "source" anchor and points to the "destination" anchor, which may be any Web resource (e.g., an image, a video clip, a sound bite, a program, an HTML document, an element within an HTML document, etc.).

## Visiting a linked resource

The default behavior associated with a link is the retrieval of another Web resource. This behavior is commonly and implicitly obtained by selecting the link (e.g., by clicking, through keyboard input, etc.).

The following HTML excerpt contains two links, one whose destination anchor is an HTML document named "chapter2.html" and the other whose destination anchor is a GIF image in the file "forest.gif":

```
<BODY>
...some text...
<P>You'll find a lot more in <A href="chapter2.html">chapter two</A>.
See also this <A href="../images/forest.gif">map of the enchanted
forest.</A>
</BODY>
```

By activating these links (by clicking with the mouse, through keyboard input, voice commands, etc.), users may visit these resources. Note that the href attribute in each source anchor specifies the address of the destination anchor with a URI.

The destination anchor of a link may be an element within an HTML document. The destination anchor must be given an anchor name and any URI addressing this anchor must include the name as its fragment identifier.

Destination anchors in HTML documents may be specified either by the A element (naming it with the name attribute), or by any other element (naming with the id attribute).

Thus, for example, an author might create a table of contents whose entries link to header elements H2, H3, etc., in the same document. Using the A element to create destination anchors, we would write:

```
<H1>Table of Contents</H1>
<P><A her="#section1">Introduction</A><BR>
<A her="#section2">Some background</A><BR>
<A her="#section2.1">On a more personal note</A><BR>
...the rest of the table of contents...
...the document body...
<H2><A name="section1">Introduction</A></H2>
...section 1...
<H2><A name="section2">Some background</A></H2>
...section 2...
<H3><A name="section2.1">On a more personal note</A></H3>
...section 2.1...
```

**We may achieve the same effect by making the header elements themselves the anchors:**

```
<H1>Table of Contents</H1>
<P><A her="#section1">Introduction</A><BR>
<A her="#section2">Some background</A><BR>
<A her="#section2.1">On a more personal note</A><BR>
...the rest of the table of contents...
...the document body...
<H2 id="section1">Introduction</H2>
...section 1...
<H2 id="section2">Some background</H2>
...section 2...
<H3 id="section2.1">On a more personal note</H3>
...section 2.1...
```

## **Other link relationships**

By far the most common *use* of a link is to retrieve another Web resource, as illustrated in the previous examples. However, authors may insert links in their documents that express other relationships between resources than simply "activate this link to visit that related

resource". Links that express other types of relationships have one or more link types specified in their source anchor.

The *roles* of a link defi

By far the most common *use* of a link is to retrieve another Web resource, as illustrated in the previous examples. However, authors may insert links in their documents that express other relationships between resources than simply "activate this link to visit that related resource". Links that express other types of relationships have one or more link types specified in their source anchor.

ned by A or LINK are specified via the rel and rev attributes.

For instance, links defined by the LINK element may describe the position of a document within a series of documents. In the following excerpt, links within the document entitled "Chapter 5" point to the previous and next chapters:

```
<HEAD>
...other head information...
<TITLE>Chapter 5</TITLE>
<LINK rel="prev" href="chapter4.html">
<LINK rel="next" href="chapter6.html">
</HEAD>
```

The link type of the first link is "prev" and that of the second is "next" (two of several recognized link types). Links specified by LINK are **not** rendered with the document's contents, although user agents may render them in other ways (e.g., as navigation tools).

Even if they are not used for navigation, these links may be interpreted in interesting ways. For example, a user agent that prints a series of HTML documents as a single document may use this link information as the basis of forming a coherent linear document. Further information is given below on using links for the benefit of search engines.

## Specifying anchors and links

Although several HTML elements and attributes create links to other resources (e.g., the IMG element, the FORM element, etc.), this chapter discusses links and anchors created by the LINK and A elements. The LINK element may only appear in the head of a document. The A element may only appear in the body.

When the A element's href attribute is set, the element defines a source anchor for a link that may be activated by the user to retrieve a Web resource. The source anchor is the location of the A instance and the destination anchor is the Web resource.

The retrieved resource may be handled by the user agent in several ways: by opening a new HTML document in the same user agent window, opening a new HTML document in a different window, starting a new program to handle the resource, etc. Since the A element has content (text, images, etc.), user agents may render this content in such a way as to indicate the presence of a link (e.g., by underlining the content).

When the name or id attributes of the A element are set, the element defines an anchor that may be the destination of other links.

Authors may set the name and href attributes simultaneously in the same A instance.

The LINK element defines a relationship between the current document and another resource. Although LINK has no content, the relationships it defines may be rendered by some user agents.

## **Link titles**

The title attribute may be set for both A and LINK to add information about the nature of a link. This information may be spoken by a user agent, rendered as a tool tip, cause a change in cursor image, etc.

Thus, we may augment a previous example by supplying a title for each link:

```
<BODY>
...some text...
<P>You'll find a lot more in <A href="chapter2.html"
    title="Go to chapter two">chapter two</A>.
```

```

<A href="./chapter2.html"
    title="Get chapter two.">chapter two</A>.
See also this <A href="../images/forest.gif"
    title="GIF image of enchanted forest">map of
the enchanted forest.</A>
</BODY>

```

## Internationalization and links

Since links may point to documents encoded with different character encodings, the A and LINK elements support the charset attribute. This attribute allows authors to advise user agents about the encoding of data at the other end of the link.

The hreflang attribute provides user agents with information about the language of a resource at the end of a link, just as the lang attribute provides information about the language of an element's content or attribute values.

Armed with this additional knowledge, user agents should be able to avoid presenting "garbage" to the user. Instead, they may either locate resources necessary for the correct presentation of the document or, if they cannot locate the resources, they should at least warn the user that the document will be unreadable and explain the cause.

## The A element

```

<!ELEMENT A - - (%inline:)* -(A)    -- anchor -->
<!ATTLIST A
  %attrs;                -- %coreattrs, %i18n, %events --
  charset  %Charset;     #IMPLIED -- char encoding of linked resource --
  type     %ContentType; #IMPLIED -- advisory content type --
  name     CDATA         #IMPLIED -- named link end --
  href     %URI;         #IMPLIED -- URI for linked resource --
  hreflang %LanguageCode; #IMPLIED -- language code --
  rel      %LinkTypes;   #IMPLIED -- forward link types --
  rev      %LinkTypes;   #IMPLIED -- reverse link types --
  accesskey %Character;  #IMPLIED -- accessibility key character --
  shape    %Shape;      rect    -- for use with client-side image maps --
  coords   %Coords;     #IMPLIED -- for use with client-side image maps --
  tabindex NUMBER       #IMPLIED -- position in tabbing order --
  onfocus %Script;     #IMPLIED -- the element got the focus --
  onblur   %Script;     #IMPLIED -- the element lost the focus --
>

```

***Start tag: required, End tag: required***

### ***Attribute definitions***

**name = *cdata* [CS]**

This attribute names the current anchor so that it may be the destination of another link. The value of this attribute must be a unique anchor name. The scope of this name is the current document. Note that this attribute shares the same name space as the id attribute.

**href = *uri* [CT]**

This attribute specifies the location of a Web resource, thus defining a link between the current element (the source anchor) and the destination anchor defined by this attribute.

**hreflang = *langcode* [CI]**

This attribute specifies the base language of the resource designated by href and may only be used when href is specified.

**type = *content-type* [CI]**

This attribute gives an advisory hint as to the content type of the content available at the link target address. It allows user agents to opt to use a fallback mechanism rather than fetch the content if they are advised that they will get content in a content type they do not support.

Authors who use this attribute take responsibility to manage the risk that it may become inconsistent with the content available at the link target address.

For the current list of registered content types, please consult [MIMETYPES].

**rel = *link-types* [CI]**

This attribute describes the relationship from the current document to the anchor specified by the href attribute. The value of this attribute is a space-separated list of link types.

**rev = *link-types* [CI]**

This attribute is used to describe a reverse link from the anchor specified by the href attribute to the current document. The value of this attribute is a space-separated list of link types.

**charset = *charset* [CI]**

This attribute specifies the character encoding of the resource designated by the link. Please consult the section on character encodings for more details.

## *Attributes defined elsewhere*

- id, class (document-wide identifiers)
- lang (language information), dir (text direction)
- title (element title)
- style (inline style information )
- shape and coords (image maps)
- onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events )
- target (target frame information)
- tabindex (tabbing navigation)
- accesskey (access keys)

## **Each A element defines an anchor**

1. The A element's content defines the position of the anchor.
2. The name attribute names the anchor so that it may be the destination of zero or more links (see also anchors with id).
3. The href attribute makes this anchor the source anchor of exactly one link.

Authors may also create an A element that specifies no anchors, i.e., that doesn't specify href, name, or id. Values for these attributes may be set at a later time through scripts.

In the example that follows, the A element defines a link. The source anchor is the text "W3C Web site" and the destination anchor is "<http://www.w3.org/>":

For more information about W3C, please consult the  
<[A href="http://www.w3.org/"](http://www.w3.org/)>W3C Web site</A>.

This link designates the home page of the World Wide Web Consortium. When a user activates this link in a user agent, the user agent will retrieve the resource, in this case, an HTML document.

User agents generally render links in such a way as to make them obvious to users (underlining, reverse video, etc.). The exact rendering depends on the user agent. Rendering may vary according to whether the user has already

visited the link or not. A possible visual rendering of the previous link might be:

For more information about W3C, please consult the W3C Web site.

**To tell user agents explicitly what the character encoding of the destination page is, set the charset attribute:**

For more information about W3C, please consult the  
<A href="http://www.w3.org/" charset="ISO-8859-1">W3C Web  
site</A>

**Suppose we define an anchor named "anchor-one" in the file "one.html".**

*...text before the anchor...*  
<A name="anchor-one">This is the location of anchor one.</A>  
*...text after the anchor...*

This creates an anchor around the text "This is the location of anchor one.". Usually, the contents of A are not rendered in any special way when A defines an anchor only.

Having defined the anchor, we may link to it from the same or another document. URIs that designate anchors contain a "#" character followed by the anchor name (the fragment identifier). Here are some examples of such URIs:

- An absolute URI: `http://www.mycompany.com/one.html#anchor-one`
- A relative URI: `./one.html#anchor-one` or `one.html#anchor-one`
- When the link is defined in the same document: `#anchor-one`

Thus, a link defined in the file "two.html" in the same directory as "one.html" would refer to the anchor as follows:

*...text before the link...*  
For more information, please consult <A href="./one.html#anchor-one">  
anchor one</A>.  
*...text after the link...*

**The A element in the following example specifies a link (with href) and creates a named anchor (with name) simultaneously:**

```
I just returned from vacation! Here's a  
<A name="anchor-two"  
  href="http://www.somecompany.com/People/Ian/vacation/family.png">  
photo of my family at the lake.</A>.
```

This example contains a link to a different type of Web resource (a PNG image). Activating the link should cause the image resource to be retrieved from the Web (and possibly displayed if the system has been configured to do so).

***Note.** User agents should be able to find anchors created by empty A elements, but some fail to do so. For example, some user agents may not find the "empty-anchor" in the following HTML fragment:*

```
<A name="empty-anchor"></A>  
<EM>...some HTML...</EM>  
<A href="#empty-anchor">Link to empty anchor</A>
```

## **Syntax of anchor names**

An anchor name is the value of either the name or id attribute when used in the context of anchors. Anchor names must observe the following rules:

- **Uniqueness:** Anchor names must be unique within a document. Anchor names that differ only in case may not appear in the same document.
- **String matching:** Comparisons between fragment identifiers and anchor names must be done by exact (case-sensitive) match.

Thus, the following example is correct with respect to string matching and must be considered a match by user agents:

```
<P><A href="#xxx">...</A>  
...more document...  
<P><A name="xxx">...</A>
```

## ILLEGAL EXAMPLE:

The following example is illegal with respect to uniqueness since the two names are the same except for case:

```
<P><A name="xxx">...</A>
<P><A name="XXX">...</A>
```

Although the following excerpt is legal HTML, the behavior of the user agent is not defined; some user agents may (incorrectly) consider this a match and others may not.

```
<P><A href="#xxx">...</A>
...more document...
<P><A name="XXX">...</A>
```

Anchor names should be restricted to ASCII characters. Please consult the appendix for more information about non-ASCII characters in URI attribute values.

## Nested links are illegal

Links and anchors defined by the A element must not be nested; an A element must not contain any other A elements.

Since the DTD defines the LINK element to be empty, LINK elements may not be nested either.

## Anchors with the id attribute

The id attribute may be used to create an anchor at the start tag of any element (including the A element).

This example illustrates the use of the id attribute to position an anchor in an H2 element. The anchor is linked to via the A element.

```
You may read more about this in <A href="#section2">Section Two</A>.
...later in the document
<H2 id="section2">Section Two</H2>
...later in the document
<P>Please refer to <A href="#section2">Section Two</A> above
for more details.
```

The following example names a destination anchor with the id attribute:

I just returned from vacation! Here's a  
<A id="anchor-two">photo of my family at the lake.</A>.

The id and name attributes share the same name space. This means that they cannot both define an anchor with the same name in the same document. It is permissible to use both attributes to specify an element's unique identifier for the following elements: A, APPLET, FORM, FRAME, IFRAME, IMG, and MAP. When both attributes are used on a single element, their values must be identical.

### **ILLEGAL EXAMPLE:**

The following excerpt is illegal HTML since these attributes declare the same name twice in the same document.

```
<A href="#a1">...</A>
...
<H1 id="a1">
...pages and pages...
<A name="a1"></A>
```

The following example illustrates that id and name must be the same when both appear in an element's start tag:

```
<P><A name="a1" id="a1" href="#a1">...</A>
```

Because of its specification in the HTML DTD, the name attribute may contain character references. Thus, the value D&#xfc;rst is a valid name attribute value, as is D&uuml;rst . The id attribute, on the other hand, may not contain character references.

***Use id or name?*** Authors should consider the following issues when deciding whether to use id or name for an anchor name:

- *The id attribute can act as more than just an anchor name (e.g., style sheet selector, processing identifier, etc.).*
- *Some older user agents don't support anchors created with the id attribute.*
- *The name attribute allows richer anchor names (with entities).*

## Unavailable and unidentifiable resources

A reference to an unavailable or unidentifiable resource is an error. Although user agents may vary in how they handle such an error, we recommend the following behavior:

- If a user agent cannot locate a linked resource, it should alert the user.
- If a user agent cannot identify the type of a linked resource, it should still attempt to process it. It should alert the user and may allow the user to intervene and identify the document type.

## Document relationships: the LINK element

```
<!ELEMENT LINK - O EMPTY          -- a media-independent link -->
<!ATTLIST LINK
  %attrs:                -- %coreattrs, %i18n, %events --
  charset  %Charset:     #IMPLIED -- char encoding of linked resource --
  href     %URI:         #IMPLIED -- URI for linked resource --
  hreflang %LanguageCode; #IMPLIED -- language code --
  type     %ContentType; #IMPLIED -- advisory content type --
  rel      %LinkTypes:   #IMPLIED -- forward link types --
  rev      %LinkTypes:   #IMPLIED -- reverse link types --
  media    %MediaDesc;   #IMPLIED -- for rendering on these media --
>
```

*Start tag: required, End tag: forbidden*

### *Attributes defined elsewhere*

- id, class (document-wide identifiers)
- lang (language information), dir (text direction)
- title (element title)
- style (inline style information )
- onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events )
- href, hreflang, type, rel, rev (links and anchors)
- target (target frame information)
- media (header style information)
- charset(character encodings)

This element defines a link. Unlike A, it may only appear in the HEAD section of a document, although it may appear any number of times. Although LINK has no content, it conveys relationship information that may be rendered by user agents in a variety of ways (e.g., a toolbar with a drop-down menu of links).

of a document. The current document is "Chapter2.html". The rel attribute specifies the relationship of the linked document with the current document. The values "Index This example illustrates how several LINK definitions may appear in the HEAD section ", "Next", and "Prev" are explained in the section on link types.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Chapter 2</TITLE>
  <LINK rel="Index" href="../index.html">
  <LINK rel="Next" href="Chapter3.html">
  <LINK rel="Prev" href="Chapter1.html">
</HEAD>
...the rest of the document...
```

## Forward and reverse links

The rel and rev attributes play complementary roles -- the rel attribute specifies a forward link and the rev attribute specifies a reverse link.

Consider two documents A and B.

Document A:   <LINK href="docB" rel="foo">

Has exactly the same meaning as:

Document B:   <LINK href="docA" rev="foo">

Both attributes may be specified simultaneously.

## Links and external style sheets

When the LINK element links an external style sheet to a document, the type attribute specifies the style sheet language and the media attribute specifies the intended rendering medium or media. User agents may save time by retrieving from the network only those style sheets that apply to the current device.

Media types are further discussed in the section on style sheets.

## Links and search engines

**Authors may use the LINK element to provide a variety of information to search engines, including:**

- Links to alternate versions of a document, written in another human language.
- Links to alternate versions of a document, designed for different media, for instance a version especially suited for printing.
- Links to the starting page of a collection of documents.

The examples below illustrate how language information, media types, and link types may be combined to improve document handling by search engines.

In the following example, we use the hreflang attribute to tell search engines where to find Dutch, Portuguese, and Arabic versions of a document. Note the use of the charset attribute for the Arabic manual. Note also the use of the lang attribute to indicate that the value of the title attribute for the LINK element designating the French manual is in French.

```
<HEAD>
<TITLE>The manual in English</TITLE>
<LINK title="The manual in Dutch"
      type="text/html"
      rel="alternate"
      hreflang="nl"
      href="http://someplace.com/manual/dutch.html">
<LINK title="The manual in Portuguese"
      type="text/html"
      rel="alternate"
      hreflang="pt"
```

```

    href="http://someplace.com/manual/portuguese.html">
<LINK title="The manual in Arabic"
  type="text/html"
  rel="alternate"
  charset="ISO-8859-6"
  hreflang="ar"
  href="http://someplace.com/manual/arabic.html">
<LINK lang="fr" title="La documentation en Fran&ccedil;ais"
  type="text/html"
  rel="alternate"
  hreflang="fr"
  href="http://someplace.com/manual/french.html">
</HEAD>

```

In the following example, we tell search engines where to find the printed version of a manual.

```

<HEAD>
<TITLE>Reference manual</TITLE>
<LINK media="print" title="The manual in postscript"
  type="application/postscript"
  rel="alternate"
  href="http://someplace.com/manual/postscript.ps">
</HEAD>

```

In the following example, we tell search engines where to find the front page of a collection of documents.

```

<HEAD>
<TITLE>Reference manual -- Page 5</TITLE>
<LINK rel="Start" title="The first page of the manual"
  type="text/html"
  href="http://someplace.com/manual/start.html">
</HEAD>

```

Further information is given in the notes in the appendix on [helping search engines index your Web site](#).

## Path information: the BASE element

```
<!ELEMENT BASE - O EMPTY -- document base URI -->
```

```
<!ATTLIST BASE
  href    %URI:    #REQUIRED -- URI that acts as base URI --
>
```

*Start tag: required, End tag: forbidden*

### *Attribute definitions*

**href = uri [CT]**

This attribute specifies an absolute URI that acts as the base URI for resolving relative URIs.

### *Attributes defined elsewhere*

- **target (target frame information)**

In HTML, links and references to external images, applets, form-processing programs, style sheets, etc. are always specified by a URI. Relative URIs are resolved according to a base URI, which may come from a variety of sources. The BASE element allows authors to specify a document's base URI explicitly.

When present, the BASE element must appear in the HEAD section of an HTML document, before any element that refers to an external source. The path information specified by the BASE element only affects URIs in the document where the element appears.

For example, given the following BASE declaration and A declaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Our Products</TITLE>
  <BASE href="http://www.aviary.com/products/intro.html">
</HEAD>

<BODY>
  <P>Have you seen our <A href="../cages/birds.gif">Bird Cages</A>?
</BODY>
</HTML>
```

the relative URI "../cages/birds.gif" would resolve to:

http://www.aviary.com/cages/birds.gif

## Resolving relative URIs

User agents must calculate the base URI for resolving relative URIs according to [RFC1808], section 3. The following describes how [RFC1808] applies specifically to HTML.

### User agents must calculate the base URI according to the following precedences (highest priority to lowest):

1. The base URI is set by the BASE element.
2. The base URI is given by meta data discovered during a protocol interaction, such as an HTTP header (see [RFC2616]).
3. By default, the base URI is that of the current document. Not all HTML documents have a base URI (e.g., a valid HTML document may appear in an email and may not be designated by a URI). Such HTML documents are considered erroneous if they contain relative URIs and rely on a default base URI.

Additionally, the OBJECT and APPLET elements define attributes that take precedence over the value set by the BASE element. Please consult the definitions of these elements for more information about URI issues specific to them.

***Note.** For versions of HTTP that define a Link header, user agents should handle these headers exactly as LINK elements in the document. HTTP 1.1 as defined by [RFC2616] does not include a Link header field (refer to section 19.6.3).*

**By,  
L.Manjula,  
Final B.C.A.**

## *ORACLE*

- Oracle is SQL is divided in to two types.
  - . DDL
  - . DML

. DCL

- Oracle is an RDBMS but DCL not DBMS

### **DDL - Definition language.**

DDL is one which control the creation and removal of database objects.

- . DML - Data manipulation language.
- DML is a one which performs operations upon the data in database objects.
- . DCL - Data controlled language.
- DCL is one which controls the effects made by the DML statement on a database objects.

### **Naming rules for oracle sum:**

The name of database object must 1 to 30 characters long, except for database names which are limited to 8 characters.

- . The name must not contain a quotation mark.
- . Upper and lower case letters are equalent
  - . The name must begin in the alphabet and may contain the characters A to Z, 0 to 9 and \_ (under scour)
- . Oracle reserve may not be used.

### **In SQL two types of**

- . Query is extense of SQL
  - . Text (spool -) LST
- These two are operating system.

S.Karthikeyan,  
II B.SC.(cs) -?B?.

### *Newly Discovered Viruses*

Take a look at the latest virus threats including viruses, Trojans, and worms.

#### **26 June 2005 -**

- Bagle.BK  
Also known as Email-Worm.Win32.Bagle.bq (Kaspersky), Win32/Bagle.BK!Worm
- Glieder.AO  
Also known as W32/Bagle.dldr (McAfee) , Email-Worm.Win32.Bagle.bq (Kaspersky), Win32/Glieder.AO!Trojan

- 23 June 2005

### **Alcan.D**

Also known as Worm.Win32.VB.an

(Kaspersky),

Win32/Alcan.C!Worm, W32/Alcan.worm!p2p (McAfee)

- 22 June 2005

### **Qoologic Family**

- 20 June 2005

### **Mytob.EO**

Also known as W32/Mytob.HF (Norman), W32/Mytob-AS (Sophos),

W32/Mytob.ch@MM (McAfee), Net-Worm.Win32.Mytob.bk (Kaspersky),

Win32.Mytob.EO!ZIP, W32.Mytob.ED@mm (Symantec),

W32/Mytob.FK@mm (F-Secure), Win32/Mytob.57470!Worm

### **Mytob.EN**

Also known as WORM\_MYTOB.EI (Trend), W32/Mytob-AT (Sophos),

W32.Mytob.EC@mm (Symantec), W32/Mytob.FI@mm (F-Secure), Net-

Worm.Win32.Mytob.bi (Kaspersky), Win32/Mytob.53248!Worm,

W32/Mytob.cg@MM (McAfee), W32/Mytob.HD (Norman)

### **Mytob.EP**

Also known as Net-Worm.Win32.Mytob.bi (Kaspersky),

W32.Mytob.EB@mm (Symantec), WORM\_MYTOB.DZ (Trend),

Win32/Mytob.41727!Worm, W32/Agobot-Fam (Sophos), W32/Mytob.HE

(Norman), W32/Mytob.ce@MM (McAfee), W32/Mytob.FF@mm (F-Secure)

### **Mytob.EQ**

Also known as W32/Mytob.FJ@mm (F-Secure), Net-

Worm.Win32.Mytob.bi (Kaspersky), W32.Mytob.EE@mm (Symantec),

WORM\_MYTOB.EP (Trend), Win32/Mytob.40448!Worm, W32/Mytob-EP

(Sophos), W32/Mytob.HC (Norman), W32/Mytob.ci@MM (McAfee)

### **Mytob.ER**

Also known as W32/Mytob.cj@MM (McAfee), W32/Mytob.HG (Norman),

W32/MyDoom-Gen (Sophos), Win32/Mytob.31561!Worm,

WORM\_MYTOB.EH (Trend), W32.Mytob.EG@mm (Symantec), Net-Worm.Win32.Mytob.bi (Kaspersky), W32/Mytob.FL@mm

## **19 June 2005**

### **Sinteri.E**

Also known as Trojan.Abwiz (Symantec), Win32/Sinteri.E!Trojan, Trojan-Proxy.Win32.Lager.i (Kaspersky)

### **Cuebot Family**

Also known as Troj/Bdoor-HU (Sophos), BKDR\_SDBOT.GAA (Trend), TROJ\_SMALL.AOH (Trend), Backdoor.Win32.Small.fb (Kaspersky), W32/Backdoor.BEI (F-Secure), WORM\_CUEBOT.B (Trend), W32/SillyWorm.Z (F-Secure), W32/Cuebot.B (Norman), W32/Cuebot.A (F-Secure), Net-Worm.Win32.Small.b (Kaspersky), Win32/Cuebot.A!Worm, W32.Wallz (Symantec), Net-Worm.Win32.Small.c (Kaspersky), Win32.Cuebot.B, BKDR\_SDBOT.LA (Trend), Win32.Cuebot.A, W32/Cuebot-A (Sophos), Win32.Cuebot.C, W32/Cuebot-C (Sophos), Win32/Hwbot.B!Worm, W32/Cuebot-B (Sophos), Win32.Cuebot.E, Win32.Cuebot.D, Win32/Hwbot.6694!Trojan, W32/Backdoor.BAE (F-Secure), Backdoor.Win32.Small.eo (Kaspersky)

### **Lastad Family**

Also known as Win32.Lastad.E, Win32.Lastad.C, Win32.Lastad.D, Trojan-Downloader.Win32.Lastad.i (Kaspersky), Win32/Ndw!Trojan, W32/DLoader.BWP (Norman), Trojan-Downloader.Win32.Lastad.d (Kaspersky), TROJ\_DLOADER.MF (Trend), Adware-Ndware (McAfee), W32/Downloader.EW (F-Secure), TrojanDownloader.Win32.Small.rn (Pest Patrol), W32/Downloader.NZ (F-Secure), Troj/Dloader-MF (Sophos), Win32.Lastad.A, Win32.Lastad.B, W32/DLoader.AFM (Norman), Trojan-Downloader.Win32.Lastad.f (Kaspersky), TROJ\_NDWARE.A (Trend), Trojan-Dropper.Win32.Agent.jl, Win32/Lastad.G!Trojan, Trojan-Downloader.Win32.Lastad.h (Kaspersky), Win32/Lastad.H!Trojan, Win32.Lastad.G, Trojan-Downloader.Win32.Lastad.p (Kaspersky), Trojan-Downloader.Win32.Lastad.n (Kaspersky), Win32/Lastad.36352!Trojan, Win32.Lastad.F

## **17 June 2005**

### **Mytob.FI**

Also known as W32/Mytob.GB@mm (F-Secure), Net-Worm.Win32.Mytob.bi (Kaspersky), W32/Mytob.dd@MM (McAfee), W32/Mytob-AT (Sophos), WORM\_MYTOB.FY (Trend), Win32/Mytob.FI!Worm , W32.Mytob.EV@mm (Symantec)

### **Mytob.EM**

Also known as Net-Worm.Win32.Mytob.bj (Kaspersky), W32.Mytob.EA@mm (Symantec), W32/Mytob.HB (Norman), W32/Mytob-AS (Sophos), W32/Mytob.cf@MM (McAfee), WORM\_MYTOB.EB (Trend), W32/Mytob.FG@mm (F-Secure), Win32/Mytob.55934!Worm

## *TOP FIVE VIRUSES.*

21 Mar 2004 **Win32.Netsky.P**

01 Mar 2004 **Win32.Netsky.D**

19 Nov 2004 **Win32.Efewe.B**

06 Apr 2005 **Win32.Mytob.X**

09 May 2005 **Win32.Mytob.CH**

## *Extensible Markup Language*

### **Get Its Name:**

XML is an acronym. It stands for EXtensible Markup Language.  
It is also TLA - [Three Letter Acronym].

### **What Does It Do:**

XML provides a standard approach for describing, capturing, processing & publishing information. It is more efficient & beneficial than HTML.

### **Markup:**

Extra information consisted of instructions to control the layout & appearance of the word themselves. Such information is collectively known as Markup.

### **Industrial Based XML:**

In industrial based XML push technologies (CDF-Channel Definition Format) Electronic Commerce (OTP-Open Trading Protocol) Mathematic (MML-Mathematical Markup Language).Any language based on XML consists of set of element types that have been given certain names & meanings.

### **Set of Element Types:**

It defines types of documents & referred to DTD-Document Type Definition.

### **About Hypertext:**

XML does not predefine the elements so it use hypertext.XML detegrates the task of capturing hypertext information to a subsidiary standard known as XLL ? EXtensible Markup Language.

XML based on truly International Standards

- ? style="FONT: 7pt 'Times New Roman'"> XML is derived from SGML(ISO 8879)
- ? style="FONT: 7pt 'Times New Roman'"> XSL is derived from DSSSL(ISO/IEC10179)
- ? style="FONT: 7pt 'Times New Roman'"> XLL is derived from hytime(ISO/IEC10744)
- ? style="FONT: 7pt 'Times New Roman'"> XML is Unicode(ISO/IEC 10646)

It is directly supported in programming language such as Java & in OS such as Windows NT & AIX.XML is the frame work for making markup language.

XML not in competition, but HTML may well one day become a XML application.

**By,  
L.Manjula,  
FINAL B.C.A.**

### *VPN*

What are the technologies available for tackling the problem of security in mobile networks?

VPN is the most reliable technology you tackle the problem of security in mobile networks. VPNs are already mobile (Nokia offers IP Sec ? VPN clients on many of its symbian based devices). Mobile VPNs provide a high level of security for mobile Business users who wish to connect to their corporate network. While conceptually similar similar to a remoteaccess VPN, the mobility of the remote device, diversity of the underlying network infrastructure, and limited resources of a mobile phone being in new challenges.

Mobile VPN is only one part in a corporate security package. Personal firewall, anti-virus, and local data encryption are other essential components in an enterprise. More importantly, if users store sensitive information on devices, then they might also need file encryption on the device. Nokia has a security function on the Nokia 9500 communicator and it offers SMS-activated security. VPN is the most secure technology for mobile networks.

**By,  
J.RAMYAH,  
III ? B.Sc, (CS) ?B?.**

## *rp;d rp;d Mirfs;*

Syntax ,y;yhj Software Nfl;Nld;.  
Code Nl ,y;yhj Language Nfl;Nld;.  
Loop Na ,y;yhj Condition Nfl;Nld;.  
Error Nu ,y;yhj Program Nfl;Nld;.  
Parameter ,y;yhj Functions Nfl;Nld;.  
Keyword ,y;yhj Statement Nfl;Nld;.  
Operators ,y;yhj Calculation Nfl;Nld;.  
Mouse Nr ,y;yhj Movement Nfl;Nld;.  
nrhd;dij Nfl;Fk; Computer Nfl;Nld;.  
Booting nra;ahky; OS Nfl;Nld;.  
Compile nra;ahky; Compiler Nfl;Nld;.  
Run Nd nra;ahky; Answer Nfl;Nld;.  
Declare nra;ahky; Variable Nfl;Nld;.  
Define Nd nra;ahky; Value Nfl;Nld;.  
jhNd nra;Ak; Processor Nfl;Nld;.  
rp;d rp;d Comment Line Nfl;Nld;.  
rPf;fp uk; MUk; Error ia Nfl;Nld;.  
Balagurusamy apd; mepit Nfl;Nld;.  
Return nra;ahky; Function Nfl;Nld;.  
Dennis Ritchie d; Ntfk; Nfl;Nld;.  
Charless Babbage apd; jpwikia Nfl;Nld;.  
Minimize nra;ahky; rp;d;djh Nfl;Nld;.

Maximize nra;ahky; nghpajhf Nfl;Nld;.  
 nrhd;dhy; Save thFk; Memory Nfl;Nld;.  
 Pixels ,y;yhj Picture Nfl;Nld;.  
 Stack Nf ,y;yhj Storage Nfl;Nld;.  
 Toolbar ,y;yhj Window Nfl;Nld;.  
 Icons ,y;yhj Desktop Nfl;Nld;.  
 Printer ,y;yhj Output Nfl;Nld;.  
 Scanner ,y;yhj Scanning Nfl;Nld;.  
 ghh;j;jhy; Open MFk; Computer Nfl;Nld;.  
 ,j;jid Nfl;Lk; fpilf;ftpy;iy>  
 ,jpNy vJTk; elf;ftpy;iy>  
 Computer, Computer, Computer Ntz;lhk;>  
 vd;W T.V, T.V, T.V ghh;j;Njd;.

By  
**S.Anitha,**  
**Final B.C.A**

***HARDWARE COMPONENTS***

<b>SNO</b>	<b>PRODUCT</b>	<b>DESCRIPTION</b>	<b>COMPANY</b>	<b>PRICE</b>
1	CPU (Processor)	Pentium IV 3.0 GHz Athlon 643200 2.0 GHz	Intel AMD	8,500 13,600
2	Motherboard	Intel 845GVSZR ? chipset for PIV Intel 865GBF Original	Giga byte Intel	4,600
3	RAM	128MB (SD) for ? PIV 256MB (SD) for ? PIV	Alpha Alpha	1,350 2,400

4	Mouse	Optical OEM Scroll mouse	Microsoft Samsung	700 300
5	Keyboard	Wireless keyboard Normal keyboard	Logitech Mercury	4,500 250
6	Monitor	35.6 cm ( 14? ) Mono 48.3 cm ( 19? ) Flat Monitor	Microtek Samsung	3,000 19,500
7	CD/DVD Drive	52X CD ROM Combo Drive	LG/Samsung Samsung	750 3,500
8	CD Writer	52X CD Writer ( Internal ) 4X DVD Writer ( Internal )	Samsung Pioneer	1,650 10,000
9	Floppy Drive	1.44 MB FDD	Samsung	325
10	ZIP Drive	100 MB ZIP Drive ( Internal ) 250 MB ZIP Drive ( External )	Iomega Iomega	3,600 6,800
11	Graphic Card	32 MB 128 MB	Sys ATI	1,400 3,000
12	Modem	56.6 Kbps ( Internal ) 56.6 Kbps Data / Fax ( External )	DAX US Robotics	480 6,300
13	LAN Card	10 / 100 Mbps 10 / 100 Mbps	DAX Intel	370 2,000
14	UPS	500 va ( 15 ? 20 mins backup ) 1000 va ( Line interactive )	Nexus Wipro	2,100 12,300
15	Speaker	160 W ( PS22 ) 980 W with Woofer	Vintron Creative	500 1,500
16	Sound Card	Sound card	Intex	500
17	Switch	8 Port 10/100 24 Port 10/100	Surecom D ? Link	1,500 6,700
18	Router	805 1,751	Cisco Cisco	37,000 49,000
19	Printer	DJ3745 LJ1160	HP HP	2,350 15,700
20	Scanner	SJ2400C SJ4670C	HP HP	3,200 16,500

### *Software Components*

<b>COMPANY</b>	<b>PRODUCT</b>	<b>PRICE ( INR )</b>
Microsoft	Windows XP Home OEM	3,700
	Windows 2K / XP Pro. OEM	5,900
	Office Small Business 2003 OEM	10,500
	Windows XP Home Full Pack	5,550
	Visual Studio .NET Pro 2003 Full Pack	
	Visual Studio .NET Enterprise Developer 2003	28,500
	SQL Server 2000 ( 5 Clients ) Full Pack	69,000
		50,000

Adobe	Adobe Photoshop CS 2	31,000
	In Design CS2	35,260
	Acrobat 7.0 Standard	14,050
	Premier 6.5	30,250
COREL	Corel Draw graphic Suite 12.0	19,000
Macromedia	Flash MX 2004 Standard	21,500
	Flash MX 2004 Professional	29,650
	Freehand MX 2004	15,500
	Director MX 2004	49,900
	Studio MX 2004	38,200
Symantec	Norton Antivirus 2005 for desktop	1,575
	Norton personal Firewall 2005	2,025
	PC Anywhere for 11.0 Host and Remote	7,29

By,  
**M. Praveen Raj Kamal**  
 III B.Sc ( CS ) ?B?

### *NEW LINUX SERVER AND DESKTOP*

Onward Novell software has launched its next generation. Novel open enterprise server and Linux desktop in India. Novell Linux desktop 9 comes with a ready-to-use end-user operating system, office applications and productivity tools that support users with capabilities they are used to.

Novell open enterprise server includes a suite of core networking services including the directory, identity management, file, print and provisioning services. It helps the IT staff install, configure, support, monitor, and manage their network infrastructure, regardless of the mix of Netware or Linux servers.

L.DHAMODHARAN,  
 II ? B.SC.,(cs)

### *Active Server Page*

#### **Life Cycle:-**

When the first user access the application starters & application object is created. When the second user wants to access they use the already application.

The applications are storing state information into a session. The user starts the session object is created.

The session is continued to live as long as the user continue to access the page in the application.

When all user sessions terminate for a particular application. The application is terminate & application object destroyed.

### **ASP Object:-**

- ❖ Application
- ❖ Session
- ❖ Request
- ❖ Response
- ❖ Server

### **Application Object:-**

It represent application.

It provide the memory to stored the shared data.

### **Session:-**

Sesion object represent a single user session.

The session will created when the user first visit a page in application.

### **Request:-**

It represent the visitors request.

Request object is created when the user request the page.

### **Response:-**

It represent how to send information back to the browser.

### **Server:-**

It represent web server.

It provide URL and HTML encoding facilities encoding the data to URL.

### **ASP Web application Components:-**

- ❖ Regular HTML pages.
- ❖ Active Server Page[ASP].
- ❖ Global . asa.
- ❖ Ancillary resources.

### **Virtual Root:-**

ASP application to place the files up ypur application into a special directory called Virtual Directory [or] Virtual Root [or] Application Root.

## **Running your application:-**

Embedding Server-side scripts.

Embedding with <script> tags.

Embedding with <%-----%>

Embedding directives & variables.

Eg:-

<%= variablename%>

<%=directive=value%>

by,  
**Final B.C.A,**  
**T.Suresh.**

## **ASP.net**

### **Introduction to ASP.net: -**

ASP.net is more than next version of Active Server Pages(ASP). ASP.net is largely Syntax Compatible with Asp. ASP.net is different from ASP in two major ways.

- ? ASP.net offers several programmatic enhancements over ASP.
- ? ASP.net offers change in programming fundamentals, in the way you write the code for creating these pages.

ASP.net is a compiled .NET based Environment.

### **Features of ASP.net: -**

- ? Compiled ASP.net scripts
- ? Caching

- ? ADO+, an easier to use ADO.
- ? Post back forms
- ? Server Controls

## **Advantages of ASP.net: -**

### ? **Enhanced Performance:**

ASP.net is compiled common language runtime code

Running on the server. ASP.net can take advantage of early binding, just-in-time compilation, native optimization & catching services, right out-of-the-box.

### ? **World-class tool support:-**

A rich toolbox & designer in the visual studio integrated development environment compliment the ASP.net framework.

### ? **Power & Flexibility:-**

Power & flexibility of that entire platform is made available to web application developers. ASP.net is also language independent.

### ? **Simplicity:-**

ASP.net makes it easy to perform common tasks from simple form submission & client authentication to deployment & site configuration.

### ? **Manageability:-**

ASP.net employs a text based, hierarchical configuration system which simplifies applying settings to your server environment & web applications.

### ? **Scalability & Availability:-**

ASP.net has been designed with scalability in mind with features specifically tailored to improve performance in clustered & multi-processor environment.

? **Customizability & Extensibility:-**

ASP.net delivers a well-factored architecture that allows developers to plug-in their code at the appropriate level.

? **Security:-**

With built in windows authentication & per application configuration. You can be assured your applications and secure.

? **ASP & ASP.net can coexist:-**

ASP & ASP.net can run side-by-side on an internet information server, web server without interference, there is no chance of corrupting an existing asp application simply by installing ASP. Net. The ASP. Net runtime will process only those files with an. aspx file name extension.

? **ASP.net Applications: -**

ASP. Net defines an application as the sum of all files, pages, handlers, modules & executable code that can be invoked or run in the scope of the given virtual directory on a web application server.

ASP. Net application on a web server is executed within a unique. ASP. Net automatically assigns one of these instances to process each incoming http request that is received by the application.

? **Lifetime of an ASP.net application: -**

ASP. Net applications created the first time request is made to the server before that no ASP. Net code is executing. First request pool of http application instances is created & application-on start event is fixed. If last instance exist the application on End event is fixed.

By,  
B.Priya,

*ABBREVIATION*

ZIF	-Zero Insertion Force
XNS	-Xerox Network System
WORM	-Write Ones Read Many
WOSA	-Windows Open Services Architecture
MAIL	-Manufacturers Association of Information Technology
MIDI	- Musical Instrument Digital Interface
MIME	- Multipurpose Internet Mail Extensions
NBCD	- Natural Binary Coded Decimal
SPARC	- Standards Planning And Requirements Committee
EDVAC	- Electronic Discrete Variable Automatic Computer
FFT	- Fast Fourier Transformation
IBM	-International Business Machine Corporation