



## Education: [Online courses](#)

### Accessing XML on the Client

#### Table of Contents

- [The XML Enabler: Delivering XML to Everyone](#)
  - [XML Enabler Architecture](#)
  - [Implementation](#)
  - [Configuring the XML Enabler](#)
- [A Sample Document](#)
- [Using the XML Enabler with Microsoft Internet Explorer 5.0](#)
- [Using the XML Enabler with Dynamic HTML-Enabled Browsers](#)
- [Using the XML Enabler with Netscape Navigator 2.02 for OS/2](#)
- [Using the XML Enabler with Other Browsers](#)
- [Summary](#)
- [For More Information](#)
  - [XML Technologies](#)
  - [Specifications](#)
  - [Browsers](#)

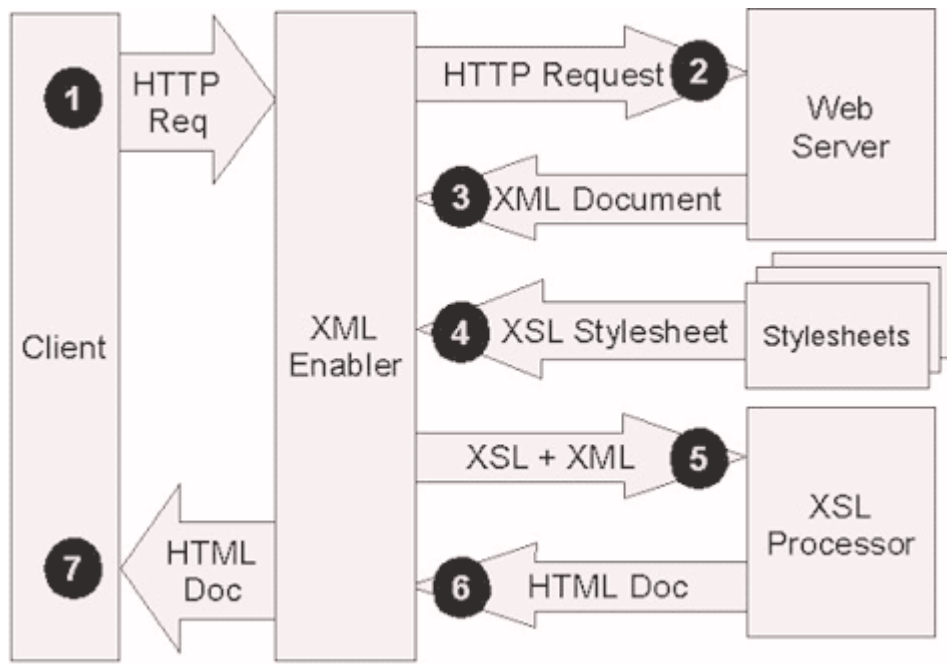
### The XML Enabler: Delivering XML to Everyone

The [XML Enabler](#), available on IBM's alphaWorks site, allows you to convert XML-tagged data into HTML, using different stylesheets for different browsers. We'll discuss the XML Enabler briefly, then we'll show how to view XML-tagged data on any browser you choose.

#### XML Enabler Architecture

The architecture of the XML Enabler is simple: when a browser attempts to load an XML document, the Enabler looks at the type of browser that requested the document. The Enabler is set up to use a different XSL stylesheet, depending on the browser type. It retrieves the requested XML document, processes it with the appropriate stylesheet, and returns the results to the browser. This allows you to generate different views of an XML document, each of which takes advantage of as many browser features (table support, Dynamic HTML support, frame support, etc.) as possible.

Here's a diagram that shows how the XML Enabler works:



The XML Enabler goes through the following steps as it processes a client's request for an XML document:

1. The HTTP request goes from the client to the XML Enabler. Included with this request is a `User-Agent` field that identifies the browser type.
2. The XML Enabler retrieves the URL of the XML document from the HTTP request, then opens that URL to request the document.
3. The requested XML document is returned.
4. Based on the value found in the `User-Agent` field, the XML Enabler selects an XSL stylesheet.
5. The XML Enabler sends both the XML document and the XSL stylesheet to the XSL processor. The processor transforms the XML document according to the rules in the XSL stylesheet.
6. When the transformation is complete, the XSL processor sends the HTML document back to the XML Enabler. (Note: Technically, the output of the stylesheet doesn't have to be an HTML document. It could be some other flavor of XML.)
7. The XML Enabler sends the transformed document back to the requesting client. The client displays the HTML, having no idea that those tags began life in an XML document.

## Implementation

The browser is implemented as a Java servlet. It is built upon the [XML4J Parser](#) from IBM, and the [Lotus XSL Processor](#). All of the code is written in Java.

## Configuring the XML Enabler

The XML Enabler has three configuration parameters:

1. The default stylesheet
2. Some number of browser/stylesheet pairs
3. The order in which browsers should be checked

These parameters can be edited using the [TaskGuide Viewer](#), an XML-based wizard tool. Alternately, the Java properties file can be edited directly.

The browsers are specified by a search string. The XML Enabler looks in the `User-Agent` field to determine which browser requested the XML document. As soon as it finds a match, the XSL stylesheet associated with that search string is used. For example, if the browser is [Opera](#), the `User-Agent` field will contain the word `Opera`. If no matches are found, the default stylesheet is used.

Here's a table showing the `User-Agent` fields for several browsers. All browsers are running on Windows NT unless otherwise stated. Also note that the values here may change depending on your system configuration.

Browser	User-Agent
Charlotte Version 2.1 (running on VM)	Charlotte/2.1.0 VM_ESA/2.2.0 CMS/13
HandWeb Version 2.0.2 (running on an IBM WorkPad/Palm III)	Mozilla/3.0 (compatible; HandHTTP 1.1)
HotJava Version 1.1.4	HotJava/1.1.4 FCS
Lynx	Lynx/2.8.1dev.16 libwww-FM/2.14FM
Microsoft Internet Explorer Version 4.01	Mozilla/4.0 (compatible; MSIE 4.01; Windows NT; compat)
Microsoft Internet Explorer Version 5 Beta 2	Mozilla/4.0 (compatible; MSIE 5.0b2; Windows NT)
Netscape Navigator Version 2.02 (running on OS/2)	Mozilla/2.02 (OS/2; I)
Netscape Navigator Version 4.07	Mozilla/4.07 [en] (WinNT; U ;Nav)
Opera Version 3.20	Mozilla/3.0 (compatible; Opera/3.0; Windows 95/NT4) 3.2
Pocket Internet Explorer (running on Windows CE)	Microsoft Pocket Internet Explorer/0.6

Notice that some browsers with very different functions may contain the same string in their `User-Agent` fields. For example, all versions of the Netscape, Internet Explorer, and Opera browsers contain the phrase `Mozilla`. For this reason, the order in which the search strings should be checked is important. Searching for `MSIE` before `Mozilla` will distinguish between Internet Explorer and Netscape; searching for those strings in the reverse order will not.

These parameters can be edited directly in a Java properties file, or they can be edited using the [TaskGuide Viewer](#), an XML-based wizard tool.

## A Sample Document

Before we demonstrate viewing XML on various clients, we'll look at a sample XML document. Here's the text of `newsArticle.xml`, the document we'll try to view:

```

<?xml version="1.0"?>
<!DOCTYPE NewsArticle SYSTEM "NewsArticle.dtd">
<NewsArticle>

  <Head>
    <Title>
      IBM Releases Open Software to Improve Security,
      Performance & Reliability of Internet E-Mail Systems
    </Title>
    <Subtitle>
      Free Secure Mailer Code Could Ensure Security of Systems
      Transferring Billions of E-Mails Daily
    </Subtitle>
    <Date>
      12/14/98
    </Date>
    <Summary>
      IBM is making available through alphaWorks free open source
      software designed to improve the security, reliability and performance
      of e-mail delivery services, a crucial component of the Internet's
      infrastructure. IBM's Secure Mailer could replace Sendmail -- 20-year-old,
      freely available mail delivery software that processes more than
      three-quarters of the Internet's e-mail traffic today.
    </Summary>
    <Category topic="CorporateAndFinancial"/>
  </Head>

  <Body>
    <Paragraph>
      <Text>
        Yorktown Heights, New York, December 14, 1998 --
        IBM today announced it is making available open source software
        designed to improve the security, reliability and performance of
        e-mail delivery services, a crucial component of the Internet's
        infrastructure. Called Secure Mailer, the new software could
        replace e-mail delivery software that processes more than
        three-quarters of the Internet's
        e-mail traffic today.
      </Text>
    </Paragraph>

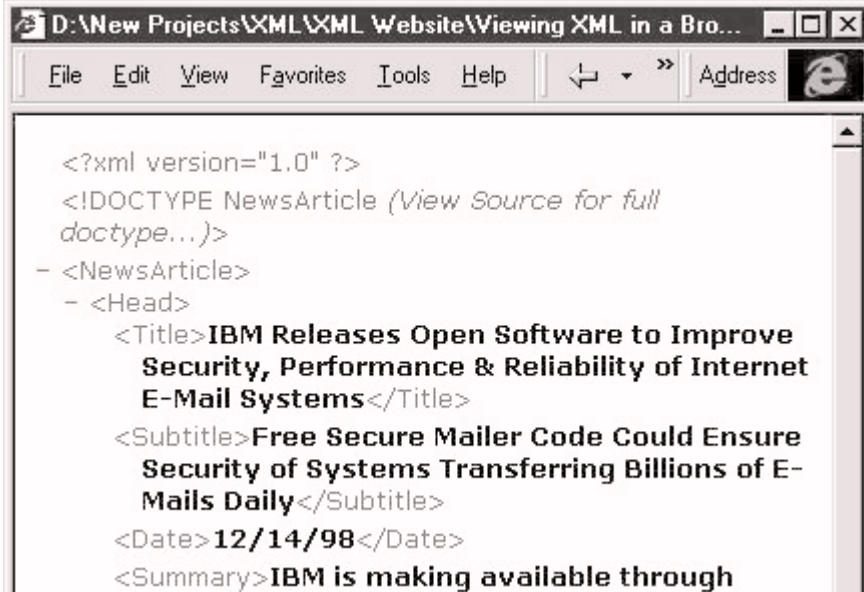
    <Paragraph>
      <Text>
        Secure Mailer is available for download beginning today
        from IBM's alphaWorks website at www.ibm.com/alphaworks. Secure
        Mailer is open-source software, so anyone can freely copy, use,
        modify and distribute it.
      </Text>
      <A href="http://www.ibm.com/alphaworks" title="alphaWorks" />
    </Paragraph>
    .
    .
    .
  </Body>
</NewsArticle>

```

Our sample document is a news item, marked up with various categories, summaries, and other pieces of information. Among other things, the use of XML tags to describe this data makes the job of a search engine easier. Searching for all articles in which the `<Category>` is `CorporateAndFinancial` and the `<Summary>` contains the words `open source` is far more likely to produce useful matches than a simple search of HTML tags. Tagging this data with XML could also enable a smart news reader to organize and display an article or set of articles in various ways.

### Using the XML Enabler with Microsoft Internet Explorer 5.0

The latest beta of Microsoft's Internet Explorer 5.0 (which we'll call IE5 from now on) has the most advanced XML capabilities of any mainstream browser available today. The simplest way to exploit IE5's XML capabilities is to view the XML document directly. Here's what you'll see:



Well, that's not terribly exciting, is it? Fortunately, Microsoft has added support for the emerging Extensible Stylesheet Language (XSL) standard. Using an XSL stylesheet, we can tell the browser how each XML tag should be rendered. Here's an excerpt from a stylesheet based on the December 17, 1998 version of the XSL standard:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">

  <xsl:template match="NewsArticle">
    <HTML>
      <HEAD>
        <TITLE>
          <xsl:apply-templates select="Head/Title"/>
        </TITLE>
      </HEAD>
      <BODY>
        <xsl:apply-templates select="//Category"/>
        <H1>
          <xsl:apply-templates select="Head/Title"/>
        </H1>
        <xsl:apply-templates select="//Summary"/>
        <xsl:apply-templates select="//Paragraph/Text"/>
      </BODY>
    </HTML>
  </xsl:template>

  <xsl:template match="Head/Title">
    <xsl:process-children/>
  </xsl:template>

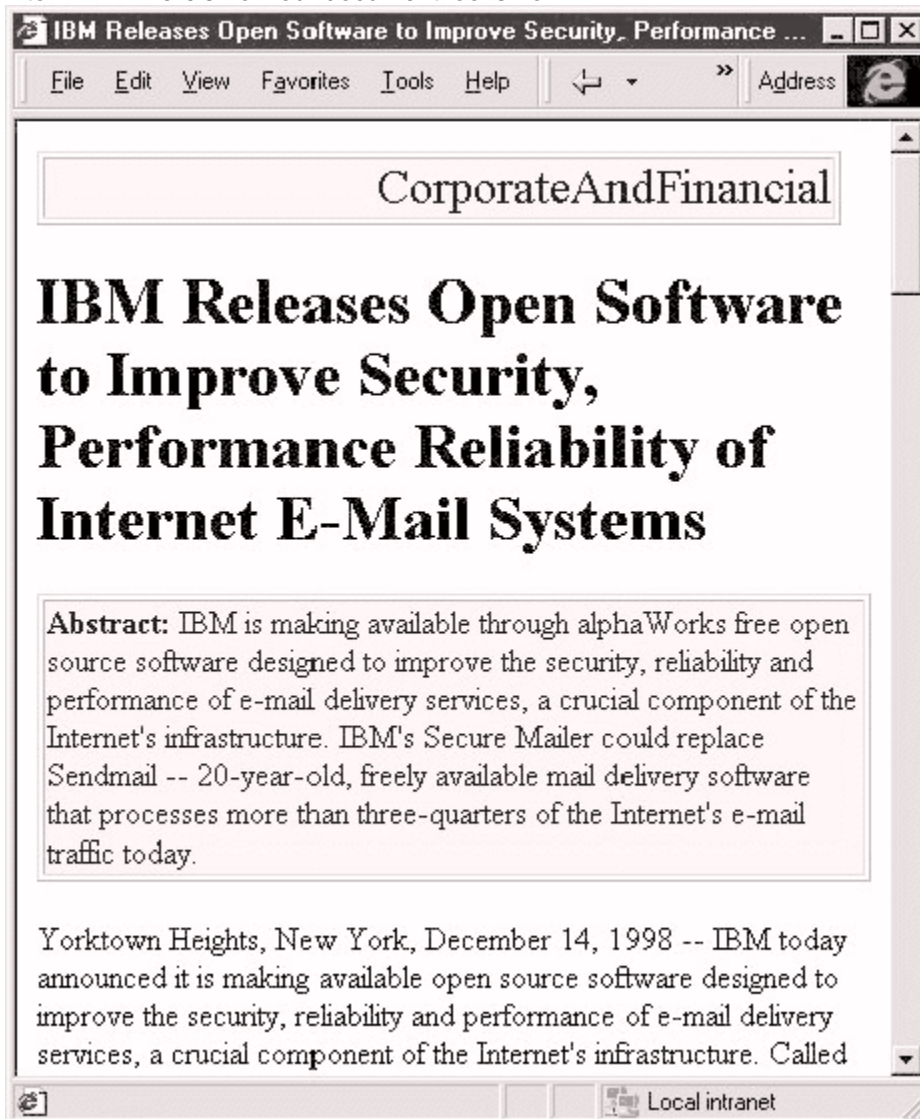
  ...
```

To associate the stylesheet with our XML document, we can add a *processing instruction* to the top of the XML document:

```
<?xml version="1.0"?>
```

```
<?xml:stylesheet type="text/xsl" href="ie5.xsl" ?>
<!DOCTYPE NewsArticle SYSTEM "NewsArticle.dtd">
<NewsArticle>
...
```

Now when we view the XML document in IE5, the browser uses our stylesheet to convert the XML tags into HTML. Here's how our document looks now:



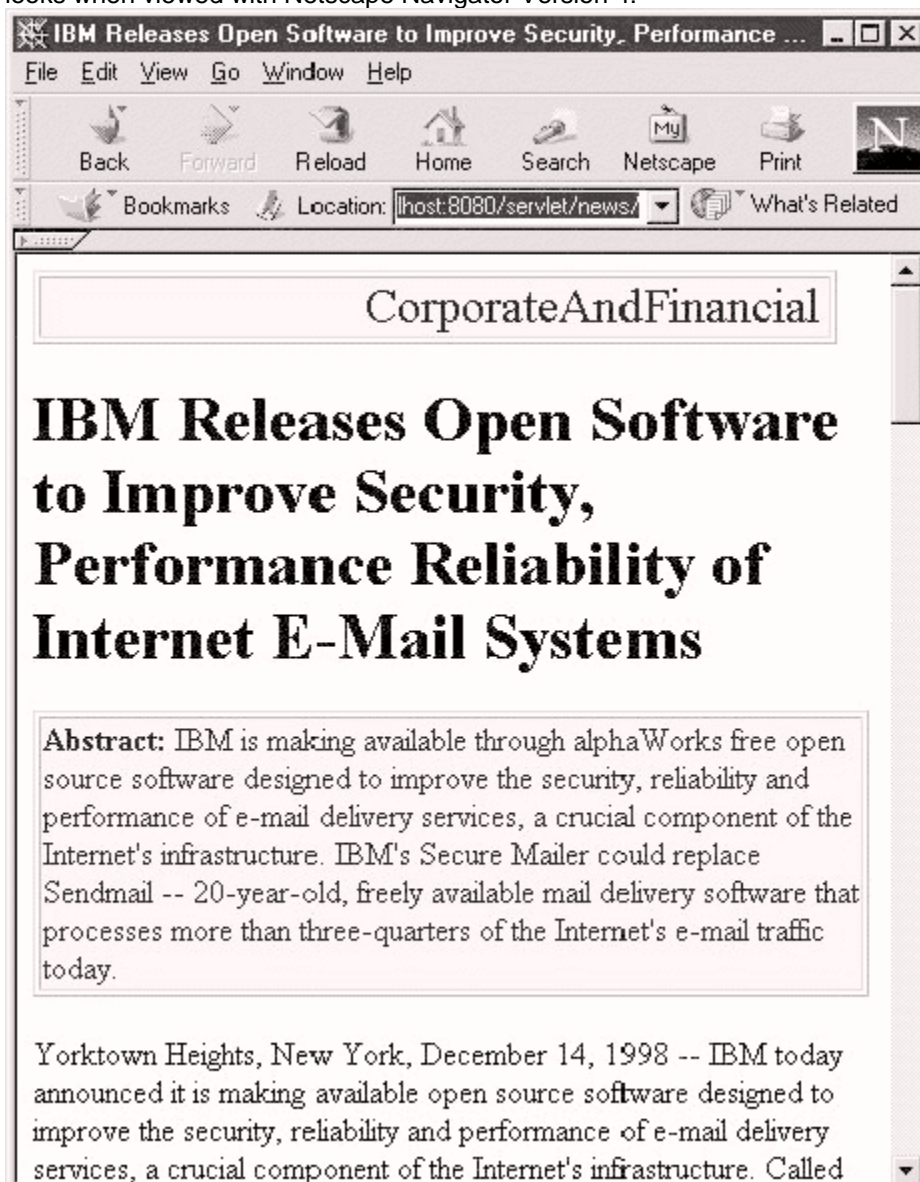


Notice that we've made a number of formatting enhancements to the original document. The `<Category>` element is now in a right-justified table at the top of the document, and the `<Summary>` element appears in a table just below the heading. Although we added a processing instruction to the document, we did not change the XML-tagged data at all. The stylesheet allowed us to generate a very different view of the document.

This is just a small sample of what XML and XSL can do; we could have used Dynamic HTML to add more effects and enhancements to the page. Furthermore, because the stylesheet is processed on the client, the client can use Dynamic HTML to generate new views of the data without going back to the server. This reduces the load on the server and the network.

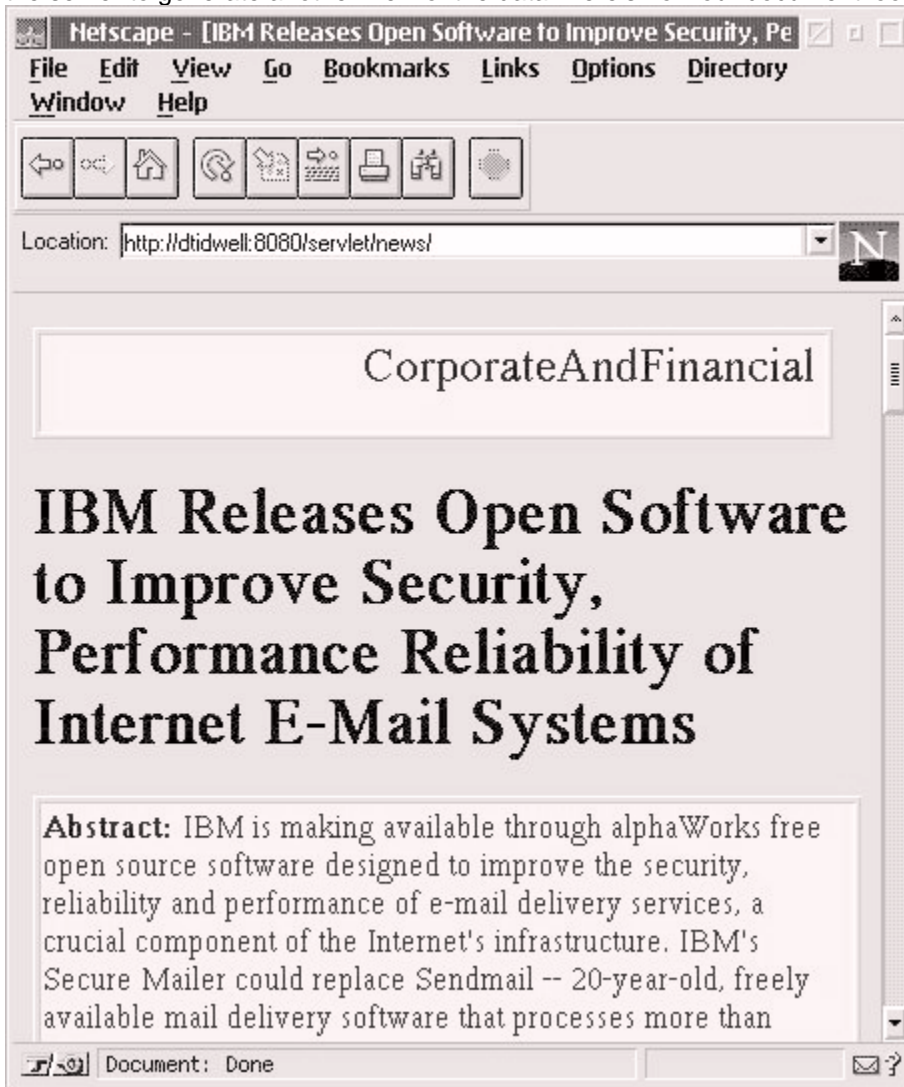
### Using the XML Enabler with Dynamic HTML-Enabled Browsers

If you're using Dynamic HTML-enabled browsers (Netscape Navigator Version 4 and Internet Explorer 4), you can use the XML Enabler with stylesheets that generate Dynamic HTML effects. This allows you to generate different views of the same data without re-accessing the server. Here's how our document looks when viewed with Netscape Navigator Version 4:



## Using the XML Enabler with Netscape Navigator 2.02 for OS/2

Netscape Navigator 2.02 for OS/2 has no support for Dynamic HTML. As a result, the best we can do is have an XSL stylesheet that generates a single view of the data. That view can contain a link back to the server to generate another view of the data. Here's how our document looks on OS/2:





## Using the XML Enabler with Other Browsers

For other browsers, our approach has been to put the browser into one of three categories:

1. Browsers with native XML and XSL support
2. Browsers with Dynamic HTML support
3. Anything else

In general, a stylesheet that works for one browser in these categories will work for any other browser in the same category. (The main exception to this comes from the differences in the way Netscape and Microsoft implemented Dynamic HTML.) This allows you to view your XML-tagged data in any browser, using a stylesheet that takes full advantage of the browser's capabilities.

Here's a look at our data from the Lynx browser, an MS-DOS browser with no support for graphics, frames, or tables. Although this is a very low-function browser, it still displays our XML-tagged data in a useful way.

```

MS-DOS Prompt - lynx http://localhost:8080/servlet/news/
IBM Releases Open Software to Improve Security, Performance Reliab (p1 of 4)
CorporateAndFinancial
IBM Releases Open Software to Improve Security, Performance Reliability of
Internet E-Mail Systems

Abstract: IBM is making available through alphaWorks free open source
software designed to improve the security, reliability and performance
of e-mail delivery services, a crucial component of the Internet's
infrastructure. IBM's Secure Mailer could replace Sendmail --
20-year-old, freely available mail delivery software that processes
more than three-quarters of the Internet's e-mail traffic today.

Yorktown Heights, New York, December 14, 1998 -- IBM today announced
it is making available open source software designed to improve the
security, reliability and performance of e-mail delivery services, a
crucial component of the Internet's infrastructure. Called Secure
Mailer, the new software could replace e-mail delivery software that
processes more than three-quarters of the Internet's e-mail traffic
today.

Developed by IBM researcher Wietse Venema, Secure Mailer is far more
robust and flexible than similar messaging components, called Mail
Transfer Agents (MTA). According to Venema, the majority of e-mail --
more than a billion messages sent daily worldwide -- is processed by
MTA technology that originated in the early eighties and was not
designed with today's Internet traffic and security needs in mind.

Secure Mailer is available for download beginning today from IBM's
alphaWorks website at www.ibm.com/alphaworks. Secure Mailer is
open-source software, so anyone can freely copy, use, modify and
distribute it.

"By offering Secure Mailer free without licensing restrictions, IBM is
helping build a stronger base for secure e-business," said Jeff Jaffe,
general manager for IBM's IT Security. "This is an important step
because MTAs with poor security are one of the most common ways for
intruders to invade a company's network."

-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

```

## Summary

IBM offers a variety of solutions that allow you to begin working with XML technology **today**. Using the XML Enabler, you can serve XML-tagged data to any type of client, without having to change your existing systems or browsers. If your users have a modern browser from Microsoft or Netscape, you can create XSL stylesheets that use their latest features. At the same time, you can support users with lower-function browsers. Best of all, the XML Enabler lets you do these things in a way that's easy to support and expand as your environment changes.

## For More Information

If you'd like to learn more about the technologies discussed in this article, try these links:

### XML Technologies

#### [alphaWorks](#)

This is IBM's source for cutting-edge Internet research and technology.

#### [The XML Enabler from IBM](#)

The XML Enabler is available free of charge.

#### [The XML for Java Parser](#)

Download the XML4J parser.

#### [The Lotus XSL Processor](#)

The latest version of the Lotus XSL Processor is available free of charge.

#### [The TaskGuide Viewer](#)

Download the TaskGuide Viewer.

### Specifications

#### [XML](#)

The official recommendation of the World Wide Web Consortium (W3C).

#### [XSL](#)

The latest version of the W3C's XSL specification.

### Browsers

#### [HandWeb](#)

The HandWeb browser for the PalmPilot supports both online and offline browsing.

#### [HotJava](#)

Sun's all-Java browser is available from the JavaSoft web site.

#### [The Lynx Browser](#)

Get more details on the Lynx browser for MS-DOS.

#### [Microsoft's Internet Explorer](#)

Download the latest version of IE.

#### [Netscape Navigator](#)

Download the latest version of Netscape Navigator.

#### [Opera Browser](#)

The Opera Browser is designed to be smaller and lighter than IE or Netscape Navigator.

#### [Pocket Internet Explorer](#)

Get more information about Pocket Internet Explorer, Microsoft's browser for Windows CE devices.