Eclipse Intranet Application

1. Statement of Need

Electronic Commerce Link, Inc. (*ec link*) is a Cincinnati-based Information Technology firm. In the past several years *ec link* has grown to be a multi-faceted company, although its primary source of revenue has remained constant. The Managed Desktop (ec|MD) service the company provides is responsible for maintaining more than 15 small-to-medium sized business networks. In general what the service provides is transparent setup and maintenance of clients and servers in a business network environment. This network must be reliable, accessible, scalable, redundant, and backed-up. *ec link* also provides services to prevent and relieve virus problems and keep software versions up-to-date.

Another main part of the ec|MD service was meant to be an Intranet site, named Eclipse, that each company had set up for them upon initial enrollment in this service. Although this was a single Intranet site it would appear private to each company using it. Almost from the initial launch of Eclipse there was need for a reworking of the code and interface. There was also the need to add many features; and short of a complete code rewrite, these features could not be implemented. As a result, Eclipse has not been used extensively by any ec|MD customers.

Eclipse 2 is the venture to create a new and improved multi-company Intranet site that is both dynamic and scalable. It not only includes many of the features missing from Eclipse, but also is completely rewritten to provide a framework for easily adding new feature modules, to implement user extensibility and incorporate an innovative, intuitive and professional interface. Furthermore, Eclipse 2 is built using the most current web

development tools and languages such as ASP, ADSI, XML and Flash for enhanced speed, interactivity, and customizability. The release of Eclipse 2 will incorporate the following features and features modules:

- Feature module based infrastructure
- Interface overhaul
- Personal start page
- Search functionality
- Conversion of Perl scripts to ASP
- Company news, *ec link* news and national news
- Local Weather
- *ec link* help desk integration
- Knowledge base application
- Quick links
- In/Out Board
- Company Directory
- Active Directory support

2. Review of Literature

Eclipse 1 problems, now recognized by the company, led to a conversation I had with the president of *ec link,* Bill Nadler. In this conversation I learned not only about problems such as unprofessional appearance, lack of content, lack of features and using Microsoft Access as the backend database, but also that the company's needs have grown and the Intranet site was never designed to meet those. These presented further problems with the site, in that it didn't allow room for modulization for easy upgrading, invisible administration and maintenance, user extensibility, pyramid marketing and it definitely did not offer us a competitive advantage. Therefore, the interview with Bill yielded both a redevelopment initiative and a solid development plan to ensure Eclipse's success.

3. Description of Solution

Eclipse 2 is an Intranet application that incorporates a slew of portal-like features as well as access to sensitive corporate data. It was developed with two different types of users in mind: general users and administrators, or users who can modify data within Eclipse. However, since both types of users are perceived as having a low to moderate level of computer knowledge, Eclipse had to be designed in a way that made it easy to use, navigate, understand and help the user along in the process. Using the latest technologies, neutral color schemes and recognizable icons and graphics made this hard task a reality.

3.1 User Profile

There are two major account classifications in Eclipse 2: users and administrators. Administrators can be further divided into news administrators, help desk administrators and location administrators. Each administrative account also retains the privileges of a user account.

The users of Eclipse 2 will be ec|MD users and generally have a novice to intermediate knowledge of technology and the Internet. Although a majority of our users may be overwhelmed when they open the new Eclipse web site, it has been developed in such a way that is intuitive, with help available at virtually any point along the way. *ec link* foresees that any problems a user may have regarding Eclipse can be answer by these help files or a location administrator.

Eclipse 2 administrators are considered to be “trained” users. While their level of computer literacy is usually higher than average users, *ec link* does not consider administrators to be expert users. Usually, administrators are corporate managers who

have been delegated to maintain or monitor a specific area within Eclipse. These users have been trained only through help files, tutorials, the Eclipse 2 user's manual or infrequent calls to our help desk. While news administrators perform news maintenance only, help desk administrators monitor help desk tickets only and location administrators monitor and maintain user accounts and eCapture only, a user can be also be delegated more than one administrative role.

3.2 Design Protocols

A lot of thought was put into the interface of Eclipse 2 and how the user would interact with the application. Since the first version of Eclipse lacked an interesting and easy to navigate interface and a good help system, *ec link* knew design components would be crucial in the development in Eclipse 2. The next 5 sections focus on the interface and interactivity of Eclipse 2.

3.2.1 Organizational Scheme

Eclipse 2 runs on many different servers. There is one central Web server that is used to provide client Web servers with Database information. These files are written in the ASP and XML programming languages and depend on the use of a custom developed component for all database connectivity. A component was used in order to encapsulate the business logic and separate it from the presentation code. A central Database server is used to warehouse general client information, Eclipse data, and company and user preferences. Finally, *ec link* customers are required to run a Web server at their location. This server allows Eclipse to edit client domain and user information as needed. The use of three servers not only provides greater speed and scalability, but also allows *ec link*

easy access for maintenance and upgrades. The following diagram shows how the servers work together to provide the Eclipse experience.

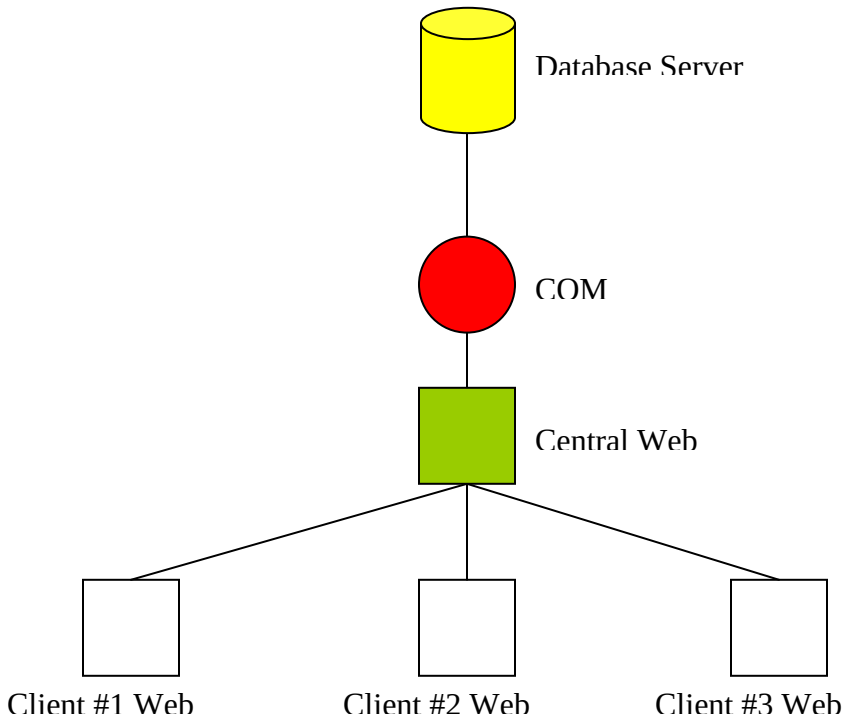


Figure 1. Server Setup

3.2.2 Interface Navigation

When the user first opens the browser he/she is presented with the Eclipse logon screen (see Figure 2). From here he/she enters his/her network username and password. After a successful logon, they are now able to view their personal start page. This start page is organized into three columns and includes news, weather, quick links, search, and allows more features to be added easily. The login process and section breakdown is illustrated in Figures 2 and 3.

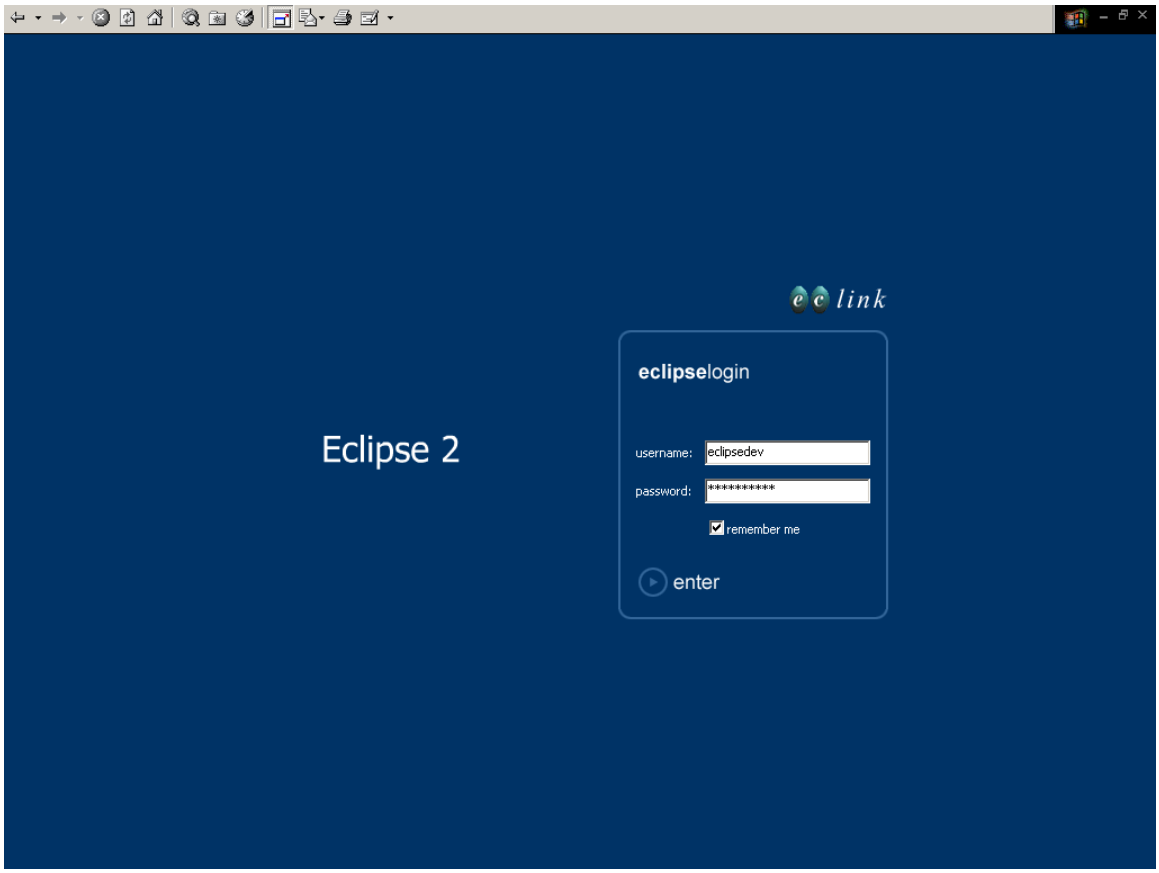


Figure 2. Login Screen

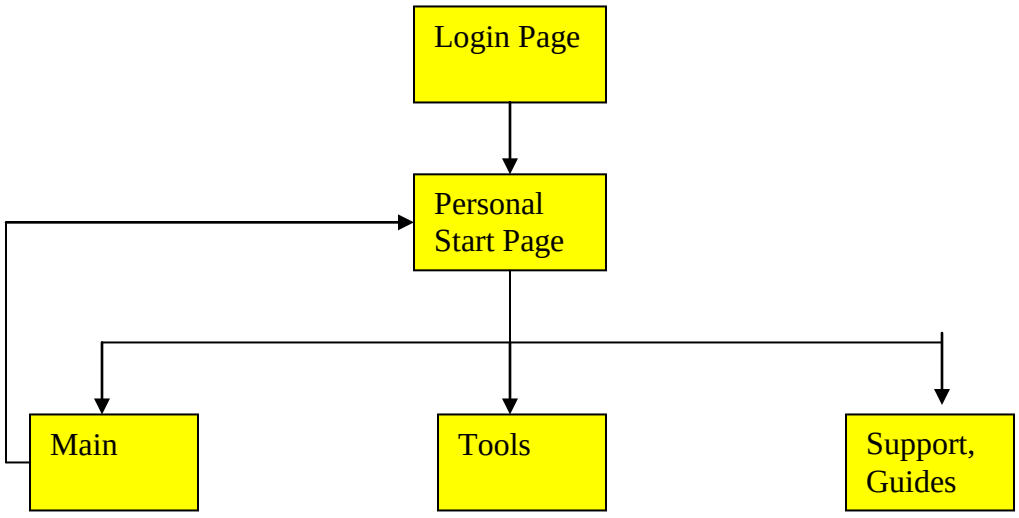


Figure 3. Post-Login Options

3.2.3 Graphic Symbols

Since Eclipse is such a large-scale application, a lot of the interface relies on text. This was chosen over graphics to create a less cluttered interface as well as faster loading times. However a few universal graphic symbols did make their way into the design.


This symbol is used in representing that help is available


This symbol designates the action of logging out.


This symbol is used in representing an administrative tool.

Figure 4. Symbols

3.2.4 Color Scheme

Eclipse 2 uses a color scheme of different shades of blues, white, black, and orange. This allows Eclipse 2 to retain a professional look while removing itself from the all-too prevalent genre of boring-looking business sites. Furthermore, the color scheme uses Web-safe colors kept within a Cascading Style Sheet for easy customization and reconfiguration.

3.2.5 Help

Not only has *ec link* placed specific help into Eclipse for one-click access to the user, *ec link* now offers access to a state-of-the-art help desk via phone or the built-in Eclipse integration. Furthermore, a user manual will be constructed and placed online for the user. *ec link* is also considering a series of Flash or Quicktime streaming tutorial movies.

4. Deliverables

This section outlines the projects specific deliverables and describes how the final project should run once all these deliverables have been accomplished.

4.1 Feature module based infrastructure

Eclipse 2 was been rewritten in such a way so that it provides an infrastructure that is easily updateable by *ec link* developers. By first creating a generic framework to build on, developers can simply "plug-in" features after they are developed using minimal time and effort. This makes the Eclipse 2 system a much more efficient system to develop for as well as allowing user-customizable interfaces with little effort.

4.2 Interface overhaul

The first version of Eclipse not only lacked in functionality, it looked horrible. Eclipse 2 sports a fresh look based mainly on Cascading Style Sheets so that even colors and fonts may be customized by the user.

4.3 Personal Start Page

Eclipse 2 incorporates a new portal feel as the default Web page. Called the "Start Page", users can customize links, news and the features available to them so when they first log on, they see only the information they want. Furthermore, each feature has integrated help so that a user can get information on using any of he various tools in Eclipse 2.

4.4 Search functionality

While the original version of Eclipse allowed the user to search their files, it often broke or yielded no results. The old method for supporting Microsoft Index Server has been replaced with a robust ASP approach that allows the user to not only search for company files, but also search through the Knowledge base files and the World Wide Web.

4.5 Conversion of Perl scripts to ASP

To make Eclipse 2 as maintainable as possible, many lines of source code have been optimized. Included in this optimization was the conversion of Perl scripts to ASP files. This allows all Web developers at *ec link* the ability to understand and modify Eclipse 2 if necessary.

4.6 Company news, *ec link* news and national news

A lot of effort went into the redesign of internal news and the integration of national news. Eclipse 2 now offers users and news administrators more control over their internal news articles including enhanced article expiration features, HTML-savvy articles, article archives and XML based database retrieval for custom formatting and speed increases. Eclipse 2 also incorporates national news onto the start page and allows the user to pick from several different news feeds.

4.7 Local Weather

Eclipse 2 allows the user to take a quick look at the conditions outside and even offers a link to glance at a 10 day forecast provided by weather.com.

4.8 *ec link* help desk integration

The *ec link* help desk plays a significant role in maintaining our networks and our relationships between companies. It only makes sense to offer the ability not only to submit help desk tickets but also monitor all outstanding tickets for a user. Administrators can also view all outstanding tickets for the company.

4.9 Knowledge base application

Our custom knowledge base application, named eCapture, has been tightly integrated with Eclipse 2. Developed as a solution to one of *ec link*'s needs, eCapture

allows users to share information within an organization. Information can either be actual files or links to files on the Intranet or Internet. eCapture takes full advantage of Active Directory under Windows 2000 or Active Directory Services under Windows NT, by offering the user access only to the folders and files allowed by his/her user account.

4.10 Quick links

Quick links are simply custom links that a user can setup for himself/herself, an administrator can setup for the company, or *ec link* can setup for all of its customers. Quick links appear as a feature module on the start page. Quick links allow you to have access to these custom links from any workstation on the network after logging on with a unique username and password combination.

4.11 In/Out Board

The In/Out status board was developed as a way to track and schedule employees who are constantly in the field. This feature not only allows co-workers and administrators to know where an employee is in case they need to reach them, but also allows a quick view of the statuses of all employees.

4.12 Company Directory

Based directly on Active Directory, the company directory displays information about each employee within an organization. This information can be updated by the user or by an administrator via any of the tools that will be created for them.

4.13 Active Directory support

Active Directory (AD) support offers robust categorization of data as well as a framework for NT security. Eclipse 2 has been tightly integrated with AD to enable the smallest amount of extra work for System Administrators and less reliance on databases.

Using AD allows Eclipse 2 to make decisions, based on NT security, without explicitly coding the privileges into ASP files, traditionally by using 3rd party components.

5. Design and Development

The following section discuss budget, timeline, software and hardware used in the proposal and development of this project.

5.1 Budget

Product	Description	Quantity	Cost
Hewlett Packard NetServer e800	Dual 800Mhz Pentium 3 machines with 512 MB RAM and 18 GB Ultra Wide SCSI drives	3	$16,855.00
Windows 2000 Server	Operating System running on all NetServers	3	$2,997.00
Microsoft SQL Server 2000	Program used for data storage, chosen because of its support for XML	1	$1,489.00
Microsoft Visual Basic Enterprise 6.0	Used to develop custom COM components for the intranet site and application modules	1	$1,299.00
Adobe Photoshop 6.0	Used to develop all of the custom graphics for the Intranet site	1	$609.00
Edit Plus 2.01	Code Editor used to create the entire web site (enhanced notepad)	1	$30.00
Microsoft XML 3.0	Used initially to grab HTML and ASP pages	1	$0.00
Microsoft IE 5.5	Required install for the components from the XML 3.0 release to run	1	$0.00
ADSI 2.5	Required for ADSI (Active Directory) support	1	$0.00
			$23,279.00

5.2 Timeline

Senior Design I

September 25	Review and understanding of current site
October 2/9/16	Redesign of database layouts and structures
October 23/30	Start production of new site
November 6/13	Implementation of News Headlines/Weather Features
November 20/27	Implementation of eCapture Feature
	Senior Design I presentation preperations
December 4	Submission of proposal and presentation

Senior Design II

January 1	Research and planning of quick links feature
January 8	Implementation of quick links feature
January 15	Internal beta test / code fixes or enhancements
January 22	Research and planning of in/out status board feature
January 29	Implementation of in/out status board feature
February 5	Internal beta test / code fixes or enhancements
February 14	Rough draft of Senior Design II design freeze
February 12/19	Research & planning of Active Directory support
February 26	Implementation of Active Directory support
	Senior Design II presentation preparations
March 5/12	Implementation of Active Directory support
March 5/12	Submission of final design freeze presentation

Senior Design III

March 26	Internal beta test / code fixes or enhancements
April 2	Research and planning of Company Directory
April 9	Implementation of Company Directory
April 16	Internal beta test / code fixes or enhancements
April 23	Research and planning of feature module customization
April 30/May 7	Implementation of feature module customization
May 14	Internal beta test / code fixes or enhancements
May 21	Senior Design III presentation preparations
May 28	Senior Design III presentation preparations
June 4	Submission of final report and presentation

5.3 Software

The software that was used to develop this Intranet application were simply a text-editor named Edit Plus and a graphics manipulation tool called Photoshop. However, many pieces of software are used to both host the application and perform data manipulation. SQL Server 7 was used as the database server software because of its robust and scalable nature. The web server software chosen was Internet Information Server 5.0. This server software has support for 3rd party components, Microsoft technology expansion such as XML, custom components written with Visual Basic and Active Server Pages.

5.4 Hardware

In the development of Eclipse, three servers were needed. One server acted solely as the database server. The other two acted as central and client servers. The central

server is an *ec link* based server that does all common routines. Common routines are routines that are generic enough to be written and use for every company within *ec link's* domain. The client server acts as one of many of our customer's server. This server handles all local authorizations and also performs many company-specific tasks such as company directory listings, the viewing of backup logs and eCapture management.

6. Proof of Design

The development of Eclipse 2 was not a small task. Many different new technologies were both learned and implemented including XML and Active Directory. As I walk through the project in a series of screen shots I will discuss each module in the page and what technologies were used to make them work.

6.1 Logon Screen

The logon screen is the first page the user will see when the user opens a browser window. This page has a rather simple layout that accepts only a username and password for verification (Figure 2), however the backend to this is one of the most complicated in the application. Using Active Directory support (ADSI), when the user clicks the login button the page first checks the username/password combination against the local domain. If this combination is valid, it then requests validation from the central server based on a database in which *ec link* stores account information. If the central server returns valid as well, the user's Session variables all filled with and amalgamation of data from the database and from the local domain server. Only then is the user granted access to Eclipse and presented with their personalized start page.

6.2 Personal Start Page

Quite possibly the most complicated section of Eclipse, the personalized start page not only shows the user's personal links and in/out status, it also displays *ec link* news, company news, global news, weather information, company links and allows the user to get stock quotes, search eCapture, the Internet, and even their local files.

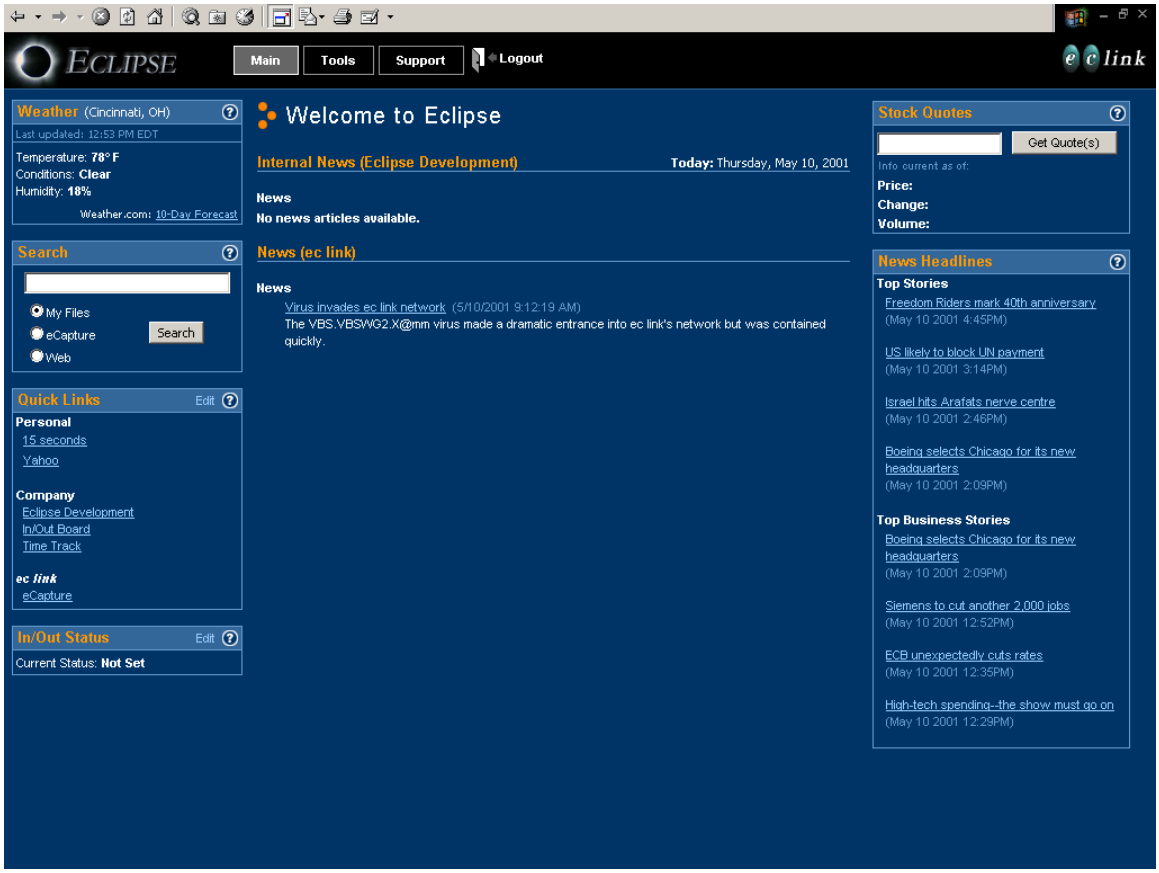


Figure 5. Personal Start Page

A lot of different routines are being used in this page to gather all the information into one place. Furthermore, 2 different databases are being queried and the results combined.

Finally all the data is formatted against a style sheet to make the appearance of a seamless application.

The weather module here was created using the XML object ServerXMLHTTP, to parse weather information out of a website. This object takes a URL as input and returns the XML, or in this case the HTML, for the URL specified. Using VBScript this data is then parsed to retrieve the information displayed on the start page. Using an Internet Explorer-specific object, the IFRAME, allows us to grab this information in the background so that the user can read news or do other tasks as the data is loading. Weather information is determined by the company's zip code and is specific to each company. This allows the user to view up-to-date weather information pre-customized for him or her.

The search module here is based on Windows 2000 Indexing Services technologies. When a user searches for files located in "My Files" or "eCapture", a query is built and run against the local domain's Indexing Services databases. A "My Files" search allows the user to search for all files he or she has permission to access within their local domain while a "eCapture" search looks for knowledge base files matching the specified. The last option "Web" simply conducts a search against the Google search engine.

Quick links are a quick and easy way for *ec link*, the company or the user to set up favorites that can be access through any station they can logon to Eclipse from. While the user can't modify these links from this page, an "Edit" link is provided that takes them to the appropriate administration pages. All the links here all pulled from a central server database table. A record marked as global has been designated as an *ec link* quick

link and is pushed to all users. Links that are stored in the database have a minimal data requirement storing only URL and link name information.

The In/Out status is a tie in to the In/Out feature module. The start page allows the user to see what their current status is and again an “Edit” link is provided to enable them to change their status. One of the more complicated features in Eclipse, the In/Out status module must query two separate databases to pull any information. This was a hard task to accomplish, but by using LEFT OUTER JOIN SQL queries, the data can be easily retrieved for immediate viewing. Below is the query I use to retrieve the In/Out status data.

```
SELECT u.FirstName + ' ' + IsNull(u.MiddleInitial + ' ', '') + u.LastName
   [FullName], u.UserID, IsNull(u.WorkPhone, '') [WorkPhone], s.Status,
   i.ReturnDateTime, i.LastUpdated, i.InOutID,
  CASE WHEN i.Annotations IS NULL THEN 0 ELSE 1 END [HasAnnotations]
FROM ECMD.dbo.Users [u]
LEFT OUTER JOIN Eclipse.dbo.InOut [i] ON i.UserID = u.UserID AND
  i.ActivateDateTime <= GetDate() AND IsNull(i.ReturnDateTime, GetDate()) >=
  GetDate()
LEFT OUTER JOIN Eclipse.dbo.InOutStatusCodes [s] ON i.StatusID =
s.StatusID AND
  i.ActivateDateTime <= GetDate() AND IsNull(i.ReturnDateTime, GetDate()) >=
  GetDate()
WHERE u.CompanyID = " & CoID & " " sLocation & sGroup
  AND u.Active = 1 "_
  AND u.isEclipseUser = 1
ORDER BY u.LastName, u.FirstName
```

Internal news and *ec link* news combined create the News and News Administration sections of Eclipse. Using ASP to query the database server and XML to format data retrieved by the custom news component, information is returned that is formatted against an XSLT (eXtensible Stylesheet Language: Tranformations) server side. The output of this formatting is simple HTML and is then plugged into the start

page with little effort. An example of the XML output produced by the component would look like this:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="news.xsl"?>
<news>
 <feed type="external">
  <category name="News">
   <article id="646">
    <title>Scheduled Network Maintenance</title>
    <date>2/26/01 1:40:35 PM</date>
    <priority>1</priority>
    <leadin>A notice about upcoming…</leadin>
   </article>
   <article id="632">
    <title>Customer Survey results are in</title>
    <date>1/5/01 12:56:55 PM</date>
    <priority>0</priority>
    <leadin>See how your fellow peers have rated…</leadin>
   </article>
  </category>
  <category name="Focus 2001">
   <article id="638">
    <title>What kind of day will you have</title>
    <date>3/1/01 9:14:07 AM</date>
    <priority>0</priority>
    <leadin>Take a break, pause and think! A bright…</leadin>
   </article>
  </category>
 </feed>
 <feed type="internal">
  <category name="News">
   <article id="648">
    <title>The Month Behind Us...</title>
    <date>3/2/01 9:32:12 AM</date>
    <priority>0</priority>
    <leadin>A month long tally of where we have been…</leadin>
   </article>
   <article id="635">
    <title>Email heads the list again...</title>
    <date>3/2/01 9:29:14 AM</date>
    <priority>0</priority>
    <leadin>Email problems still! Are they going…</leadin>
   </article>
  </category>
```

```
  </feed>
</news>
```

A new feature to Eclipse and one that was not included in the project deliverables is the ability to get stock quotes. This simple module uses a similar XML parsing method to the weather module to check stock prices against the finance.yahoo.com website. The stock quote module uses a similar IFRAME model for loading the data in the background as well.

The final module on the start page is the News Headlines. Using a database coupled with XML and XSLT, the correct headlines all downloading from moreover.com, formatted into HTML and then placed into the website. Since a sample XML file for the Internal news was shown above and it is similar to the XML file schema used here, a sample XSLT document is shown. Note that XSLT documents contain both HTML and XSLT language elements.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

<xsl:template match="/">
  <html>
  <head>
    <title>News</title>
    <style>
      body              {font-family:Arial; font-size:11px; color:#ffffff;}
      .en-story a:link    {font-family:Arial; font-size:11px; color:#6699cc;}
      .en-story a:visited {font-family:Arial; font-size:11px; color:#336699;}
      .en-story a:hover   {font-family:Arial; font-size:11px; color:#ffffff;}
      .en-source          {font-family:Arial; font-size:10px; color:#ffffff;}
      .en-date            {font-family:Arial; font-size:10px; color:#ffffff;}
    </style>
  </head>

  <body bgcolor="#003366">
  <xsl:for-each select="moreovernews/article">
```

```
  <xsl:choose>
    <xsl:when test="position() &lt;= 5">
      <div class="en-story">
          <a href="{url}" target="_blank">
            <xsl:value-of select="headline_text"/>
          </a>
        </div>
      <span class="en-date"><xsl:value-of select="harvest_time"/></span>
      <span class="en-source"> (<xsl:value-of select="source"/>)</span><br/>
      <br/>
    </xsl:when>
  </xsl:choose>
 </xsl:for-each>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

6.3 eCapture

One of the most useful tools, not to mention most used tools in *ec link's* version of Eclipse, is eCapture (See Figure 6). This is the feature module that took the most time to develop with close to 4000 lines of code behind it at and a separate database. The database for eCapture was developing using Microsoft Access. The reason for this is that the eCapture database is not hosted on the main database server. Rather, it is hosted on each client's server, and since all our clients have Access it was an obvious choice. The eCapture database consists of only a few tables that log uploads and eCapture maintenance. It is useful for determining what new articles have been added and which users are active contributors.

eCapture is organized in a file hierarchy and an unlimited of folders and files, called categories and articles, can be added to the root folder. For this reason eCapture was developed using a similar IFRAME routine to both the weather and news headline modules. This allows each folder to be loaded one

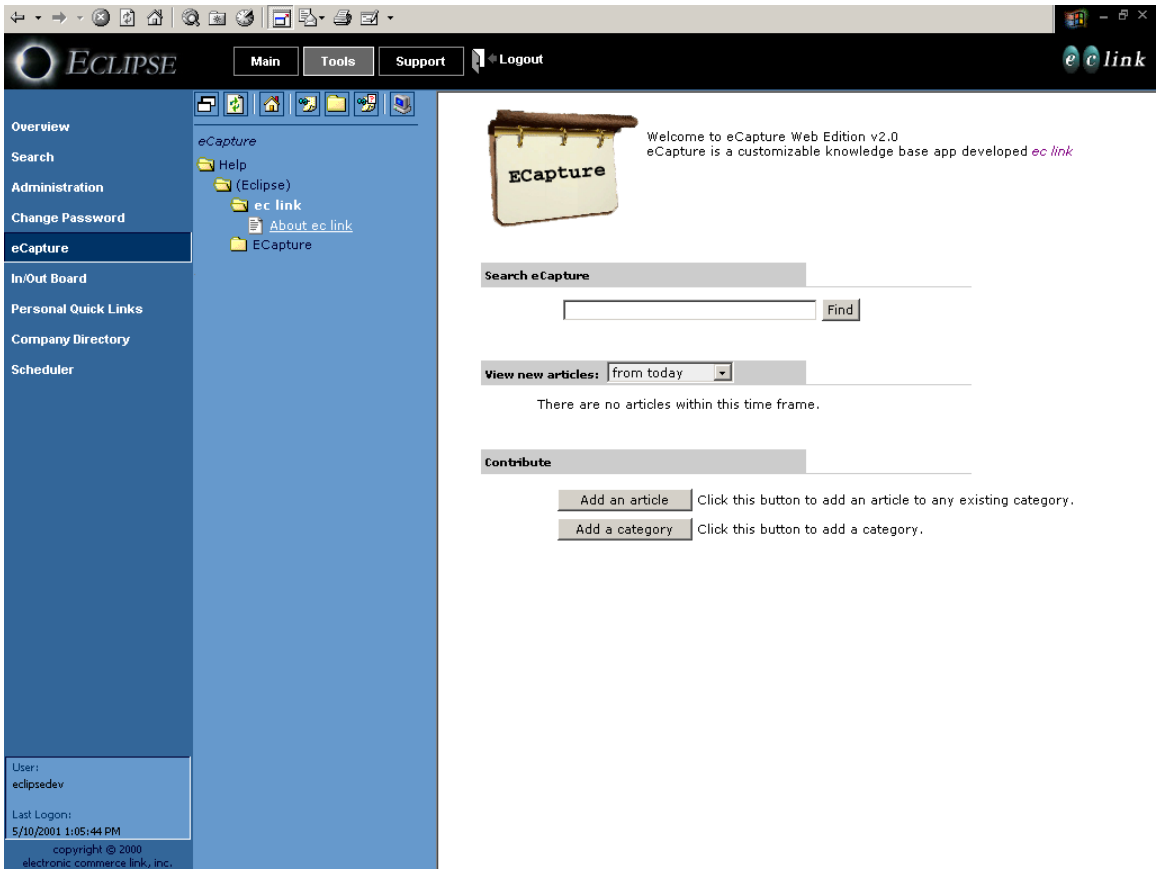


Figure 6. eCapture Main Screen

at a time instead of all at once, which greatly improves the performance and load times for the application. Since, eCapture is based on folders and files, it only made sense to create physical files and folders for storage instead of using a relational (read not hierarchical) database. Therefore when a file is added using a 3rd party upload component, or a category is created the ASP FileSystemObject (FSO) is instantiated to create the desired result. In fact, the FSO is the backbone of eCapture. Performing routines such as the creation, modification and deletion of both files and folders it is easy to see why. An example of how to create a new folder is shown in Figure 7.

To create a new category in eCapture First the user would click on the "Add Category" button. The user is then presented with a wizard, which walks through the

process of adding a category. This includes both choosing a location for the new folder as well as naming the folder. Once both steps have been completed, the user clicks the "Add Category" button under step 3 and the physical folder is created. An example of creating a folder using the FSO object within an ASP is shown here following Figure 7.

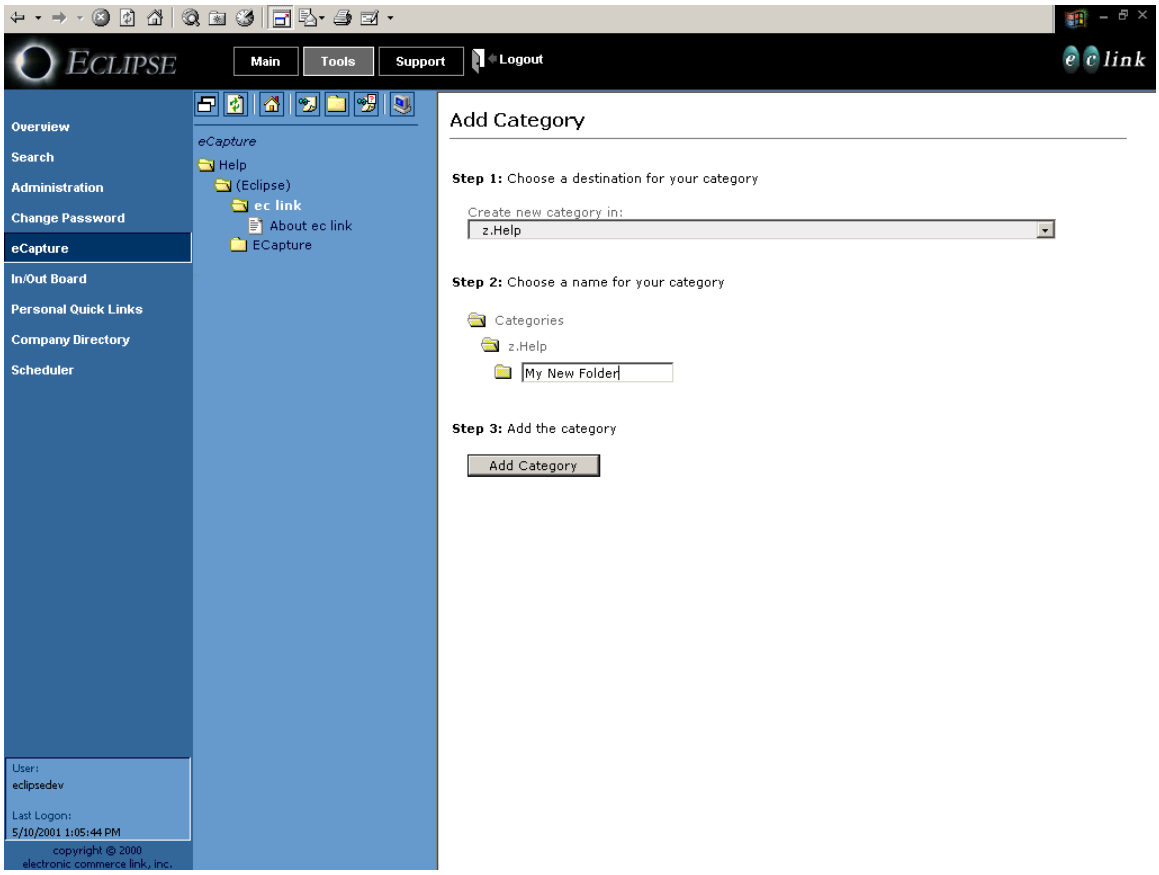


Figure 7. Adding a category (folder) in eCapture

```
<%
strNewFolderPath = Server.MapPath("\ecapture") & "\z.Help\My New Folder"

Set objFSO = CreateObject("Scripting.FileSystemObject")
If  Not objFSO.FolderExists(strNewFolderPath) Then
  objFolder = objFSO.CreateFolder strNewFolderPath
End If
%>
```

6.4 Help Desk Integration

Eclipse 2 offers help desk integration. What this means is that a feature has been built in to the Eclipse application to allow users to both submit and review help desk tickets. It also provides a uniform way for *ec link* to process these tickets. Furthermore, this feature directly ties into the help desk software solution *ec link* uses called Remedy. Using an API provided by the software, it allows the help desk feature to directly create tickets, which can then be reviewed by *ec link* and put in the support queue. The user interface for submitting a ticket is shown below in Figure 8.

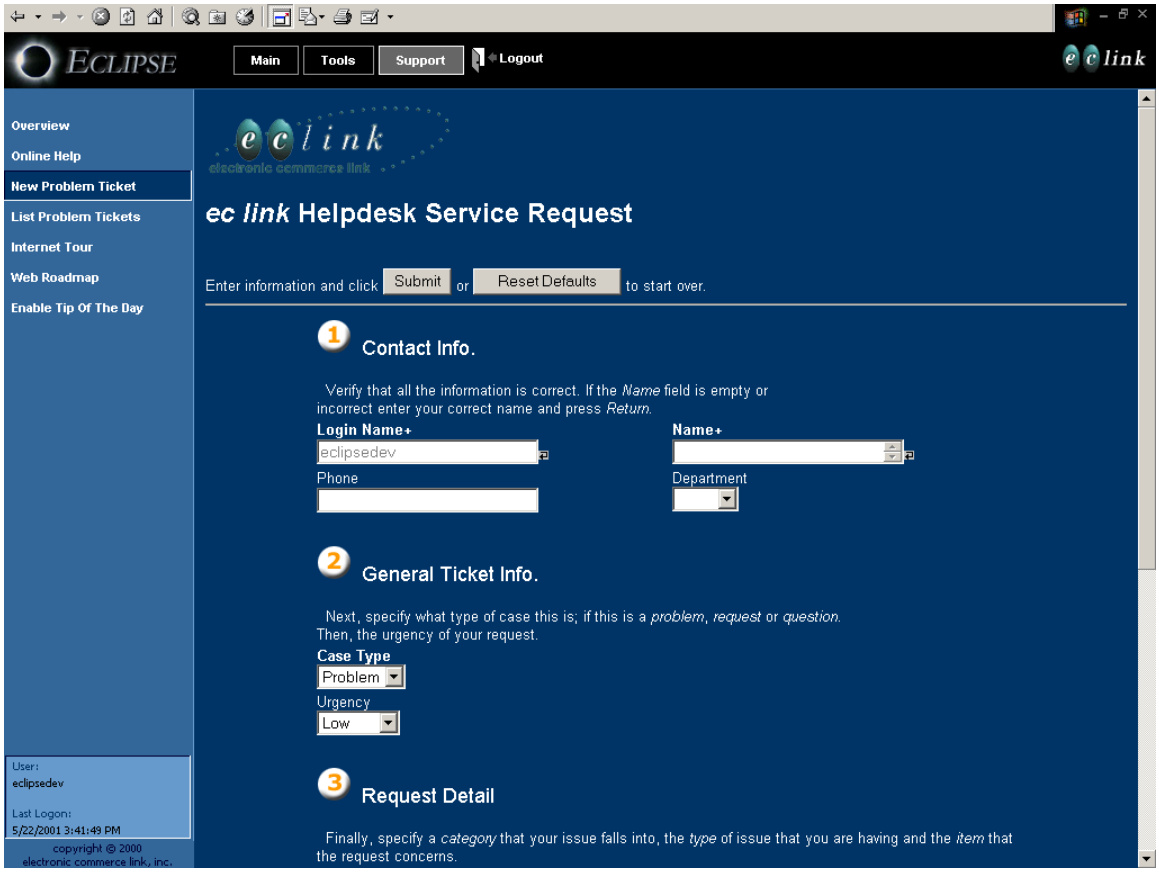
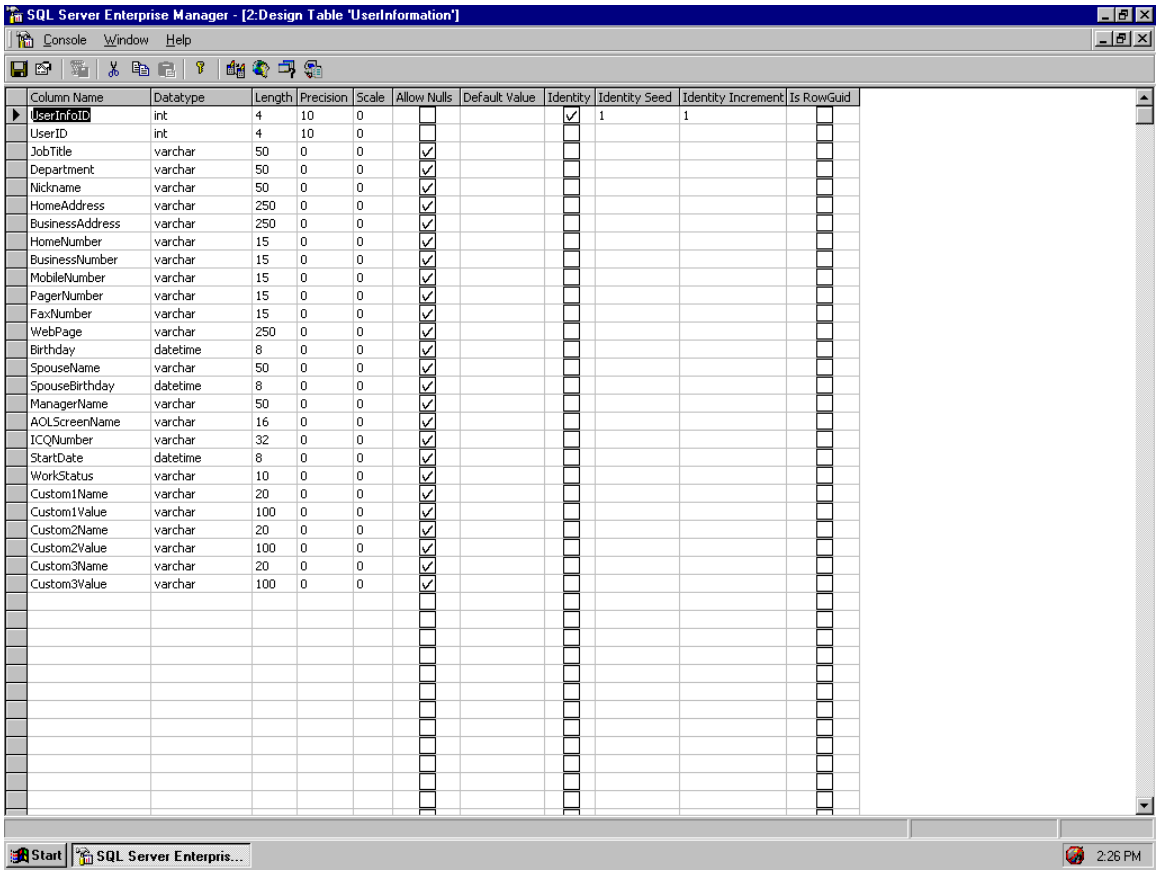


Figure 8. Help Desk Ticket Form

6.5 Company Directory

A company directory is a standard application in the business environment. However, web enabling that directory and allowing each user to update their information is not. Since Eclipse was implemented based on Active Directory support the task of a company directory should have been easy. After all, Active Directory has an extensive schema for information such as this. Unfortunately however, Windows 2000 is the only OS that can take full advantage of this information, so yet another table was created in the Eclipse database. This table includes standard personnel information and the table is shown below in Figure 9.



Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value	Identity	Identity Seed	Identity Increment	Is RowGuid
UserInfoID	int	4	10	0	☐		☑	1	1	☐
UserID	int	4	10	0	☐		☐			☐
JobTitle	varchar	50	0	0	☑		☐			☐
Department	varchar	50	0	0	☑		☐			☐
Nickname	varchar	50	0	0	☑		☐			☐
HomeAddress	varchar	250	0	0	☑		☐			☐
BusinessAddress	varchar	250	0	0	☑		☐			☐
HomeNumber	varchar	15	0	0	☑		☐			☐
BusinessNumber	varchar	15	0	0	☑		☐			☐
MobileNumber	varchar	15	0	0	☑		☐			☐
PagerNumber	varchar	15	0	0	☑		☐			☐
FaxNumber	varchar	15	0	0	☑		☐			☐
WebPage	varchar	250	0	0	☑		☐			☐
Birthday	datetime	8	0	0	☑		☐			☐
SpouseName	varchar	50	0	0	☑		☐			☐
SpouseBirthday	datetime	8	0	0	☑		☐			☐
ManagerName	varchar	50	0	0	☑		☐			☐
AOLScreenName	varchar	16	0	0	☑		☐			☐
ICQNumber	varchar	32	0	0	☑		☐			☐
StartDate	datetime	8	0	0	☑		☐			☐
WorkStatus	varchar	10	0	0	☑		☐			☐
Custom1Name	varchar	20	0	0	☑		☐			☐
Custom1Value	varchar	100	0	0	☑		☐			☐
Custom2Name	varchar	20	0	0	☑		☐			☐
Custom2Value	varchar	100	0	0	☑		☐			☐
Custom3Name	varchar	20	0	0	☑		☐			☐
Custom3Value	varchar	100	0	0	☑		☐			☐

Figure 9. Company Directory Table

Note that there are 6 different fields in the database that allow the user or administrator to enter different custom information about each employee. After the table was created the directory was a rather simple module to write. Based on simple SQL queries and ASP scripts information from the database is shown to whomever request it. The ability to edit the information is provided to the specific users and company administrators.

7. Conclusions and Recommendations

Eclipse 2 is an internal *ec link* project focused on creating a robust Intranet site for all its clients. This free service will deliver all of the features outlined in section II which include: module based infrastructure, an interface overhaul, a personal start page, search functionality, conversion of Perl scripts to ASP, company news, *ec link* news and national news, local weather, *ec link* help desk integration, a knowledge base application, quick links, an in/out board, a company directory and Active Directory support. By learning and implementing new technologies such as ASP and XML, Eclipse will have both a solid and professional infrastructure. Furthermore, the Eclipse 2 project will not only further my education but also allow me to showcase my programming knowledge and analytic thinking skills to the IET faculty.

Future enhancements should include the integration of eCapture and Active Directory so that folder and file permission apply. Currently everyone has access to all folders and files within eCapture. News headlines customization should be allowed so that users can see the headlines that are of interest to them. One enhancement, which would lend a great performance boost to Eclipse across slow networks, is the implementation of the SOAP protocol for data transmission. This would allow all the

web pages to be located remotely instead of on a central server and would allow minimal data to be transferred back and forth.