Spring 5-1-2005

IT Asset Management System on .NET Environment @ BM InfoTech, New York

Prabhu S. Sundaram
Dakota State University

IT Asset Management System on .Net Environment

@ BM InfoTech, New York

By: Prabhu S Sundaram

A project submitted in partial fulfillment of the requirements for the

Master of Science in Information Systems

Dakota State University
2005

DAKOTA STATE
dsu
UNIVERSITY

MSIS

PROJECT APPROVAL FORM

Student Name: Prabhu S Sundaram

Expected Graduation Date: May 2005

Master's Project Title: IT Asset Management System at BM InfoTech, New York

Date Project Plan Approved: May 2004

Date Project Coordinator Notified and Grade Submitted: May 10, 2005

Approvals/Signatures:

Student: [illegible] Date: 5/10/05

Faculty supervisor: [illegible] Date: 5/12/05

Committee member: [illegible] Date: 5/13/05

Committee member: ____________ Date: ____________

Copies to:
Original Attached to Written Report
Copies to: Advisor, Graduate Coordinator, and Student

Abstract

IT Asset Management (ITAM) is the process of managing an identified IT asset throughout its lifecycle of procurement, installation, support, and retirement and disposal. This paper was prepared to explain the design and development of the ITAM system which is an internal web application that was completed for BM InfoTech Inc.

The need for an IT Asset Management (ITAM) System, the technology which is enabling it, IT Asset Management data model, General ITAM tool characteristics are discussed in detail. The .Net Framework Technology was used in the development. The comparison between various web technologies[1] and the reason why the .Net Framework Technology has been chosen is discussed in detail. The Rapid Application Development Tool Visual Studio .Net is used for the development of the application. The code generated by the tool is included in the appendix B.

Acknowledgements

I would like to thank Dr. Zehai Zhou, my project advisor, for providing me an opportunity to pursue my Curriculum Practical Training at BM InfoTech, New York. I would like to express my gratitude to Mr. Anand, Technical Architect, BM InfoTech for making me to understand the concepts and his continuous support in this project development. I would like to thank Dr. Omar El-Gayar, MSIS Program Coordinator and Dr. Haomin Wang for their continuous encouragement for making me to complete my project.

Table of Contents

List of Tables

List of Figures

1. Introduction

In brief, ITAM (IT Asset Management) is the process for keeping track of technology assets (hardware, software, etc.)--Knowing where they are, who owns them, and how they are used and maintained. It is a combination of policies, procedures, and enabling technology. IT asset classes are composed of categories such as desktops, servers, network and telephony equipment, mobile phones, pagers, and PDAs[2].

ITAM is accomplished by centralizing all manually or electronically gathered information into an IT Asset Management Database (AMDB). The benefits resulting from a successful ITAM process implementation are significant and can be quantified for forward savings based on industry data and on the organization's own experiences with not having the right information available at the right time.

This report gives an overall view of the business concepts of an IT Asset Management System. ITAM System is part of a Support Module under any CRM application. The report carries the entire technical environment on how the project was designed and implemented.

The report consists of the following sections –

- Necessity of this ITAM System development
- Deliverables

- Project Life Cycle

- Future Work and enhancements

Why IT Asset Management is important

In today's complex business environment, there is constant pressure on IT organizations to understand their cost envelope in order to better manage costs and to justify their expenditures and resulting service pricing. An ITAM program is critical in helping IT improve sourcing decisions, in positioning the IT organization as a business enabler rather than a cost center, and in running IT as a business.

The company BM InfoTech, NY has various assets at different locations. They couldn't keep track of their assets. So there was a necessity for the development of an IT Asset Management System.

Technology

The technologies that were used for the IT Asset Management System are:

- Programming Languages: C#, ASP.NET, ADO.NET, XML, Web Services
- Database: SQL Server 2000
- Graphic Tools: Fireworks 5.0
- Web Development Tool: DreamWeaver 4.0
- Web server: IIS 5.0
- Third-Party Tools: Concept Draw 2.0

Development Environment

Microsoft .NET allow programmers to develop applications in language independent manner. It has been specifically designed with Web Services in mind. Visual Studio.Net is the IDE (Integrated Development Environment) for .NET based application. The professional version of Visual Studio.NET contains the .NET Framework SDK (Software Development Kit) and the main .NET languages VB (Visual Basic), C++, C# and Java script (Jscript) .NET. All the XML capabilities that were formally available to the programmer through the Microsoft XML (MSXML) parsers are now encapsulated in the system.xml namespace of classes. Appendix lists the classes that are used in the project. This project uses C# as the main language to develop the XML Web Services along with ASP.NET Web forms.

Deliverables

There is a list of deliverables pertaining to this ITAM project at BM InfoTech, NY.

- A working model of the Asset Management Application.

The company should be able to add, update and delete assets. The application should list all the details pertaining to a particular asset.

- A good documentation explaining the functionality of the system.

In order to facilitate the description of the model, all the screen shots and the source code are included in the appendix.

2. Requirements Gathering and Plan of Action

This section discusses about:

- The company and the department for which the project was developed
- Significance of the project work
- Business Requirements and application users

Company

The company BM InfoTech is located at New York. It provides resources for the Fortune 500 Clients specialized in the areas of Client Server Technologies, System Administration, and ERP applications. The following sections give a brief introduction to the significance of the project and the Business Requirements that was devised for development of the project.

Significance of the project

BM InfoTech (www.bminfotech.com) currently has around 75 consultants working on various projects for clients like Verizon, Sun Microsystems, and Delta Airlines etc. The company has offices at various places like New York, Sioux Falls, Sioux City, New Hampshire. The offices have different assets based on the size, operations and requirements. The company provides laptops for the consultants and also issues licensed software (CD's and product manual's) for the need of cross training or updating knowledge of the consults on the current technologies. But keeping track of those assets has always been a major problem for the Consultancy due to the frequent updates in this process. Thus the project (IT Asset management System) to keep track of those million dollar assets is highly significant in order to have a high ROI (Return on Investment).

Business Requirements and Users of the project

The following listed below are the business requirements which were framed for this project:

- The accounts Manager must able to identify the current assets of the company in a time-efficient manner (Ex: What is the total asset of the entire company at any point of time? Answer: Total Assets: 5 Million)

- The Company must able to find the depreciation costs each year (Ex: Say each year Sioux Falls Location's current assets depreciates by 5 million to 3.8 million)

- The Company must be able get the value of the assets in a particular location (Ex: what is the current total asset in Sioux Falls Location? Answer: 1 million)

- The Company must be able to add assets as on when needed. (Ex: If accounts department wants to add 10 computers which they have recently purchased.)

- The Company must be able to delete assets when they are sold or discarded (Ex: It might be either sold or was thrown away)

The **people** who would be using the system are:

1. Accounts Manager

2. IT Systems Engineer (For the system maintenance)

The below listed are the **current assets** of the Company's Total Assets on various Locations (varies frequently*)

1. Laptops – 25

2. Desktops – 50

3. Landline phones – 10

4. Cell phones - 10

5. Licensed Software's – 50

6. Printers - 25

Table 2-1: Assets in Specific Locations

Location	Current Assets					
	Laptops	Computers	Land phones	Cell phones	Licensed software's	Printers
New York	10	20	5	3	30	15
Iowa	10	20	3	1	10	5
South Dakota	5	10	2	1	10	5
Total	**25**	**50**	**10**	**5**	**50**	**25**

(Note: though close to the real Projections, actually these are the sample figures)

The following would be **Database Entities** of the system:

According to the requirements, the main possible entities are:

a. Assets (**Primary Key** # Asset ID)

b. Asset Categories (AssetCategory ID)

c. Depreciation (Depreciation ID)

d. Status (Status ID)

e. Maintenance (Maintenance ID)

f. Departments (Department ID)

g. Employees (Employee ID)

h. Vendors (Vendor ID)

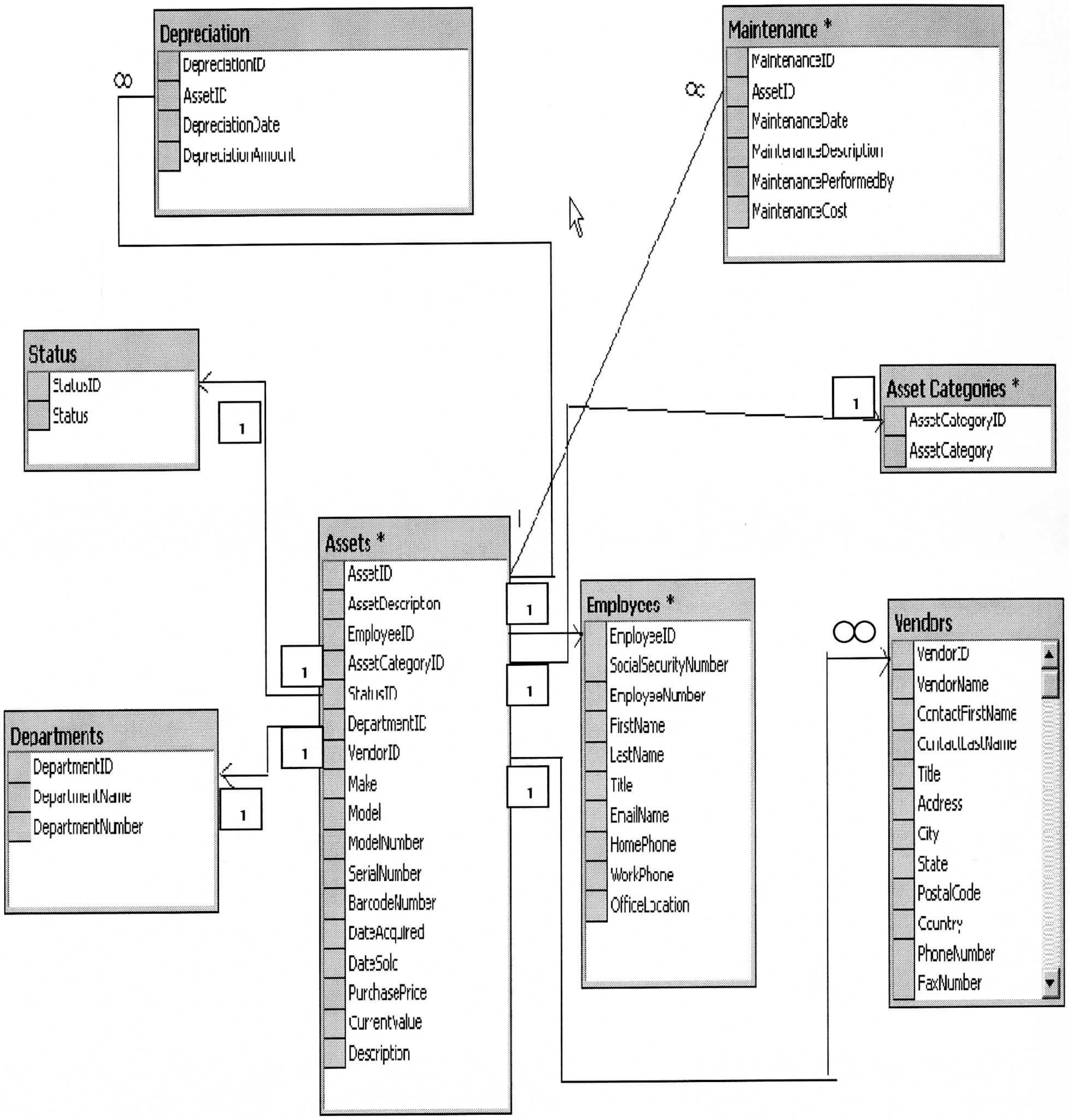
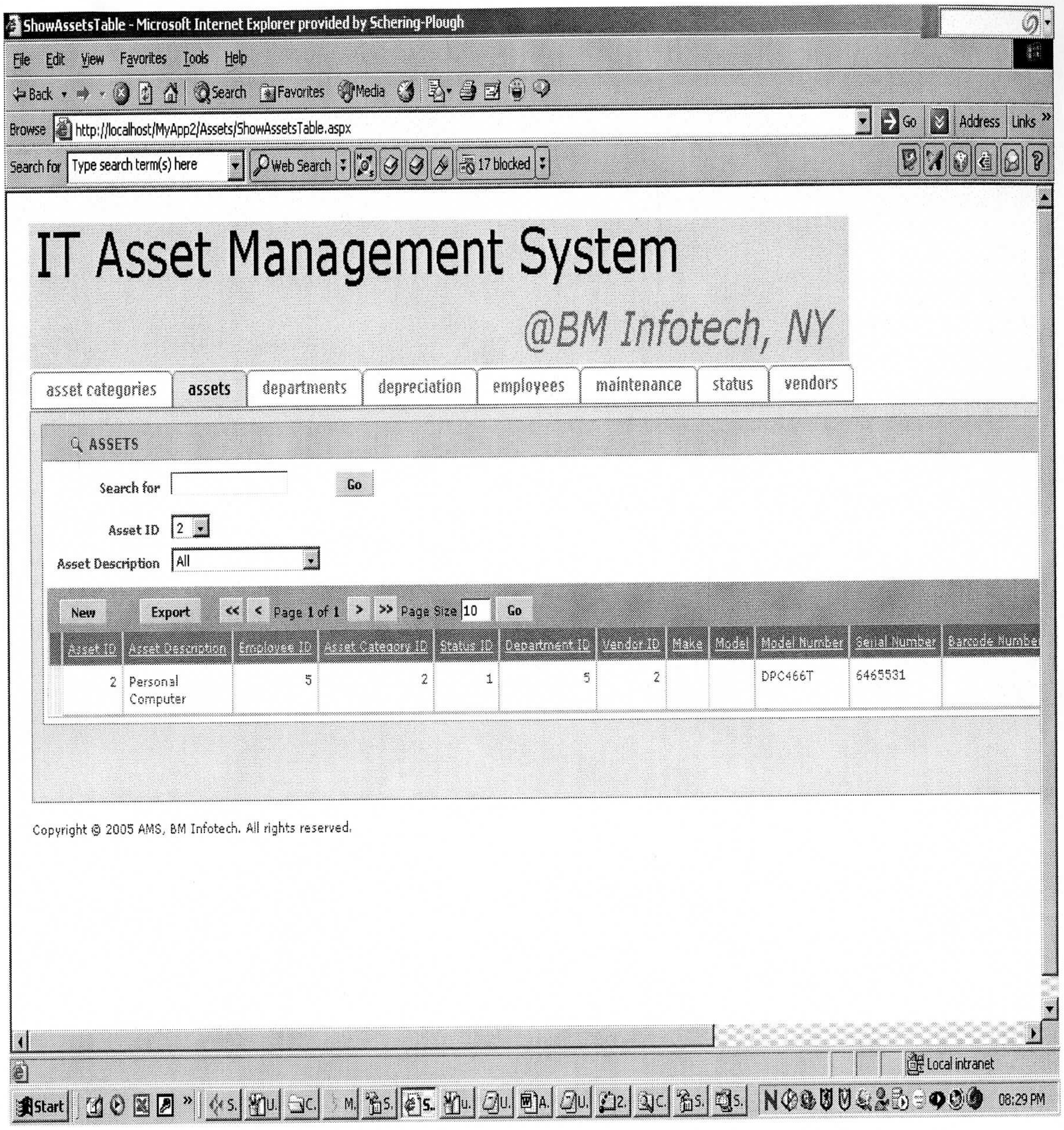


Fig 2-1: Based on the Business Requirements, the ER Diagram was created for the IT Asset Management Application


ShowAssetsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
Browse http://localhost/MyApp2/Assets/ShowAssetsTable.aspx
Go Address Links
Search for Type search term(s) here Web Search 17 blocked
Local intranet
Start
08:29 PM

IT Asset Management System

@BM Infotech, NY

asset categories | assets | departments | depreciation | employees | maintenance | status | vendors

ASSETS

Search for: Go

Asset ID: 2

Asset Description: All

New Export << < Page 1 of 1 > >> Page Size 10 Go

Asset ID	Asset Description	Employee ID	Asset Category ID	Status ID	Department ID	Vendor ID	Make	Model	Model Number	Serial Number	Barcode Numbe
2	Personal Computer	5	2	1	5	2			DPC466T	6465531	

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig 2-2. Shown is the screenshot of the web application developed using this ER diagram. Front-end web application system of AMS (Asset Management System)

3. Overall Structure

The following was discussed in this section:

- The overall structure of the web application including the .Net Framework.
- Necessary explanation of the technologies to understand the working of the application.

The .NET Framework Application

- The application to be presented consists of the following sections:
    - Microsoft ASP.NET Http handler architecture for processing requests
    - A company request class for controlling authorization checks and HTML rendering
    - XSLT documents for XML to HTML transforms
    - XML returning stored procedures
    - The physical database layer, in addition to the (forms) authentication mechanism.
- Unlike the prior ASP applications, Visual Studio Developer is like a RAD tool and handles the database connection objects and there isn't much code on the client side. Also the .Net Environment's FCL (Foundation Class Libraries) provides many system classes that handle the input/output and SQL part of the application. The challenge was on data transformation through XML and XSLT part, which largely

reduces the data transaction speed and thereby reducing the network load on the server. The below section highlights the necessary steps required for building this transformation.

UML Diagrams

Use Case Diagram

A use case is initiated by a user with a particular goal in mind, and completes successfully when the goal is satisfied. It can be further defined as a set of interactions between external actors and the system under consideration. An actor may be a class of users, roles users can play, or other systems. A primary actor is one having a goal requiring the assistance system. A secondary actor is one from which the system needs assistance.

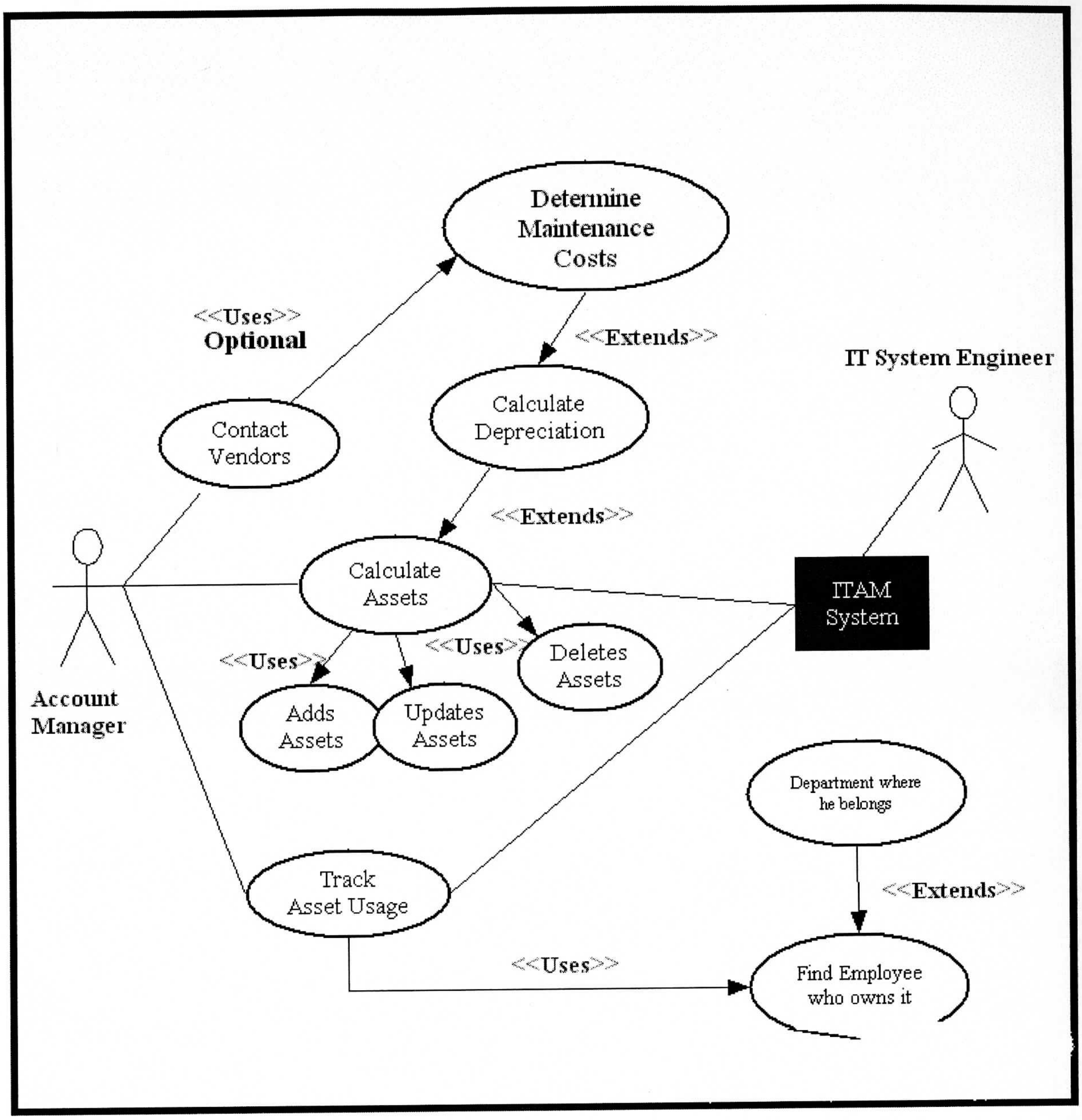


Fig 3-1: UML Use Case Diagram

Use Case Diagram Description

Primary User: Account Manager

Secondary User: IT System Engineer

System was created for the following usage:

- The System should help to keep track of the assets and able to include all the other external data in determining it.

Class Diagram

Step 1: Presentation Layer

The driving force behind the application design was the presentation layer. The premise of the developed solution was to present, using an HTML interface, the Asset Details information from the Asset database. This supports two types of users, the administrative user (System IT Engineer) and generic user (Account Manager).

Table 2-2: Presentation flow

Page Description	**Operation**	**XML Stored Procedure Style Sheet**
Companys for Location	GetCustomersByCountry	xml_customer_cty customerlistcty.xslt
Companys for Asset	GetCompaniesByUser	xml_customer_user customerlistuser.xslt
Company Detail	GetCompanyById	xml_company_id companyheader.xslt
Asset Listing	GetCompanyAsset	xml_company_Assets companyAssets.xslt
Asset Details	GetCompanyAssetDetails	xml_asset_details companyassetdetails.xslt
Company Summary	GetCompanySummById	xml_company_summary companysummary.xslt

The application creates each HTML page through the use of an XML-returning stored procedure, which is then transformed into HTML using an XSLT document. Table 1 outlines each page, the name of the operation to support the XML retrieval, the name of the associated XML returning stored procedures, and finally the corresponding XSLT document from which the HTML is generated.

Step 2: XML Data Layer

As previously stated, the application makes use of SQL Server 2000 XML functionality, each operation in Table 1 has a corresponding XML-returning stored procedure. In designing these stored procedures, the structure of the resultant XML required consideration. Several techniques could be used, as this application demonstrates. To retrieve XML results directly from the database, the FOR XML clause of the SELECT statement is used; with one of either three modes RAW, AUTO, and EXPLICIT.

RAW mode means each row in the resulting XML has the generic row identifier.

AUTO mode returns query results in a simple nested XML tree. Each table in the FROM clause, listed in the SELECT statement, is represented as an XML element of the same name. The SELECT columns are then mapped as attributes of the elements. The hierarchy is determined based on the order of the tables identified by the columns of the SELECT statement. To structure the XML identifiers column, alias names should be used.

EXPLICIT mode specifies the shape of the XML tree, the query specifying all the information to produce what is known as a universal tree. This means that the query must, in addition to specifying the required data, specify all the meta data. Other important benefits of EXPLICIT mode are that columns can be individually mapped to either attributes or sub-elements and one can generate sibling hierarchies.

In writing XML queries, one must remember that the XML identifier names are case sensitive, important when performing the XSLT transforms. In addition within the XSLT documents we also has to remember to prefix the XML identifiers with @ symbols for attributes.

Step 3: Modifying XML Hierarchy with Views

The first challenge in formulating the XML came when the XML hierarchy required flattening; information for a single hierarchal layer is derived from more than a single table. To achieve this, the XML query can use views rather than underlying tables. The view, being seen as a single table, flattens the resulting XML hierarchy. This was a requirement for presenting company asset information. In this case, I have to present the company information, with each company element containing asset information:

SELECT CMP_DETAILS.CMP_ID CompanyId, CMP_DETAILS.CMP_NAME CompanyName, CMP_DETAILS.CMP_DESC CompanyDescription, CMP_DETAILS.CMP_LOCATION CompanyLocation FROM CMP_DETAILS

Company INNER JOIN dbo.view_assets ASSET_DETAILS ON

CMP_DETAILS.CMP_ID = ASSET_DETAILS.CMP_ID WHERE

CMP_DETAILS.CMP_ID = @CMP_ID

FOR XML AUTO, ELEMENTS

The VIEW in the query is simple INNER JOIN query:

SELECT CMP_DETAILS.*, ASSET_DETAILS.*

FROM CMP_DETAILS

INNER JOIN ON ASSET_DETAILS CMP_DETAILS.CMP_ID =

ASSET_DETAILS.CMP_ID

In the generated XML, the view is treated as a single hierarchy; namely

CMP_DETAILS.

Step 4: FOR XML EXPLICIT Queries

For queries requiring a more complex XML structure, the FOR XML EXPLICIT clause can be used. EXPLICIT mode is used for the company details and asset summary pages. As stated, the structure of the XML hierarchy is determined by a generated universal table, thus the use of views are only relevant for query simplification. One of the main tasks in using EXPLICT mode is ensuring that the Tag and Parent properties are correctly set. Hence, in this case a UNION ALL is used to return the required Company and then Asset information. The final ORDER BY is important to correctly relate the assets with the appropriate company. The CAST function is used to ensure correct data type is returned from the query: numeric rather than a float. As we see, EXPLICIT mode, although verbose, offer greater flexibility in the generation of the resulting universal tree. To get a better understanding of this concept of a universal table, run the previous query without the FOR XML EXPLICIT clause. The output will be the universal table.

Step 5: Data Access Layer

As in most applications, access to the database is controlled though a data access layer; the CompanyAssets class. The purpose of this class is to abstract the stored procedure calls and present XML back to the Http application. The chosen return data type is an XPathDocument, rather than an XmlDocument. The reasons are twofold: The resultant XML obtained from SQL Server is a document fragment (may not have a root node) and hence cannot be loaded directly into an XmlDocument, and secondly an XpathDocument offers the best performance for performing transforms.

Step 6: Obtaining and Returning XML

The structure of each method call with the class is identical: format a SQLCommand and its parameters, set its active connection, execute the XML reader, and return the constructed XPath document. As only the SQLCommand settings differentiate the data access method calls, the XML processing is managed by a private method:

```
private XPathDocument CommandToXPath(SqlCommand assetCom)

{

   // setup the local objects

   SqlConnection assetCon = null;

   XmlReader xmlReader = null;

   XPathDocument xpathDoc = null;

   // set base command options

   assetCom.CommandType = CommandType.StoredProcedure;

   assetCom.CommandTimeout = 15;

   // now execute the command

   try

   {

     // setup the database connection
```

```
  assetCon = new SqlConnection(dbConnectionString);

  assetCon.Open();

  assetCom.Connection = assetCon;

  // execute the command and place into an Xpath document

  xmlReader = assetCom.ExecuteXmlReader();

  xpathDoc = new XPathDocument(xmlReader, XmlSpace.Preserve);

}

catch (Exception ex)

{

  throw new ApplicationException

       ("Cannot Execute SQL Command: " + ex.Message, ex);

}

finally

{

   // dispose of open objects
```

```
        if (xmlReader != null) xmlReader.Close();

        if (assetCon != null) assetCon.Close();

    }

    return xpathDoc;

}
```

The ExecuteXmlReader method of the SqlCommand class returns an XmlReader representing the returned XML. This reader is used to construct the XPath document. Prior to making this method call, the calling method will create a new SqlCommand, specify which stored procedure to execute, and define the appropriate parameters collection.

Step 7: Class Methods

As we can again see from Table 1, the class methods correspond 1-to-1 with an XML-returning stored procedures and a Web application-presenting function. Figure 2 outlines the public and private methods for the CompanyAssets class. The class methods are all simple get methods, and return an XPath document representation of the required XML.

Table 3-1 : CompanyAssets Class diagram

CompanyAssets
+GetCompanyAssets(): XpathDocument +GetCompanyByLocation(in Company: string): XpathDocument +GetCompanyByName(in CompanyName: string): XpathDocument +GetCompanyById(in companyID:string): XpathDocument +GetCompanyAssets(in companyID:string, in assetId: int): XpathDocument -GetDbConnection(): SqlConnection -CommandToXPath(in sqlCommand: SqlCommand): XpathDocument

The only work of the class methods is to define the stored procedure and associated parameters. No authorization checks are performed, that was done in an earlier step.

Consider the method to return the details of the company asset:

```
public XPathDocument GetCompanyAssetDetails

   (string CMP_ID, int AST_ID)

{

   // the command object to return the XML Document Fragment

   SqlCommand AssetCom = new SqlCommand("xml_asset_details");


   // the stored procedure parameters


   SqlParameter companyParam = new SqlParameter


      ("@p_CMP_ID", SqlDbType.NChar, 5);
```

```
    companyParam.Direction = ParameterDirection.Input;

    companyParam.Value = CMP_ID;

    assetCom.Parameters.Add(companyParam);

    SqlParameter assetParam = new SqlParameter("@asset", SqlDbType.Int);

    assetParam.Direction = ParameterDirection.Input;

    assetParam.Value = AST_ID;

    assetCom.Parameters.Add(assetParam);

    // return the XPath document

    return CommandToXPath(assetCom);

}
```

As we watch the above code, all the XML work is accomplished in the **CommandToXPath** method.

4. Project Life Cycle

The project life cycle of the IT Asset Management System includes the sub sections titled planning, analysis, design and execution.

Asset Management System

Planning & Analysis

The company was having a difficult time in maintaining its assets. Asset Management System would be useful for the Accounts Department and also the company could increase the ROI (Return on Investment).

UML Design for IT Asset Management System

A good UML design in the planning and the development phase is the backbone and success for any software project. In .NET, UML has a special emphasis on the standards and quality of the code involved in the projects. The Class Diagrams was used to convert the design ideas to the system code for the planning and development phases.

Class Diagram for AMS

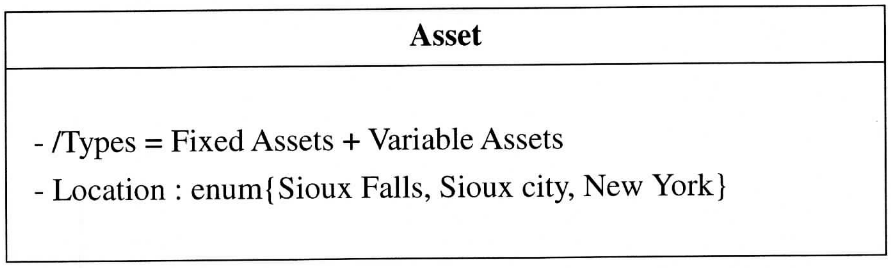
Attributes correspond to the variables declared within a class (but not variables declared within the body of a function or sub procedure).

Derived Attribute

- Types

Attribute with enumerated values

- Location


Asset
- /Types = Fixed Assets + Variable Assets
- Location : enum{Sioux Falls, Sioux city, New York}

Derived Code:

```
Class Asset

{

            private String Types;

            private String Location;

            private string Asset_Types

            {

            return Fixed Assets + Variable Assets;

            }

}
```

Visibility

- Public members (attributes or operations) **can be referenced directly by any class** in this or any other model *package (*more on packages later).

- Private members **can only be referenced in the same class** where they're declared.
- Protected members **can be referenced in the same class** or in any descendants of that class (more on inheritance later).
- Any classes in the same UML package can reference package scope members only.

Operations correspond to functions or sub procedures in C#.

For eg: public void walk(CompassBearing direction, float distance) + serialize(inout pBag : PropertyBag)

Asset
- Total Value +Location : String = Sioux City +Asset_Types
+Add(Asset: Single) -Update(Asset: Single) -Delete(Asset: Single)

```
Class Asset

{

}

Class Asset

{
```

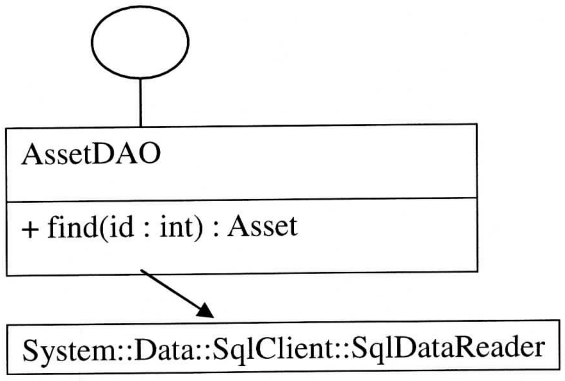
```
          Public types;

          Private Asset Value;

}

Class Asset

{

Private float value = 0;

Public string Location;

Public string Type;

Public Void AddAsset(String Type)

{

}

}

AddAsset()

{

          private static int numberOfAsset = 0;

          private string Type;

          private Asset(string Type)

          {

                    this.Type = Type;
```

```
                numberOfAsset++;

        }

        public static Person createAsset(string Type)

        {

                return new Asset(Type);

        }

        public string getType()

        {

                return this.Type;

        }

        public static int getNumberOfAsset()

        {

                return numberOfAsset;

        }

}
```

Dependencies – C#

To show that the 2 classes are related to each other, the DataAccessObject uses the System.Data.SqlClient.SqlDataReader class.

DataAccessObject



```
public Customer find(int id)
        {
        SqlConnection cn = new SqlConnection(CONNECT_STRING);
        cn.Open();
        SqlCommand cmd =
        new SqlCommand("SELECT * FROM Customers WHERE id = "
        + id.ToString(), cn);
        SqlDataReader reader = cmd.ExecuteReader();
        Customer c;
        while (reader.Read())
        {
        c = new Customer(reader.GetInt32(0), reader.GetString(1));
        }
        cn.Close();
        return c; }
```

The above code conversions are to show how UML is necessary for the deployment of the System. Other UML diagrams could be used as well to describe the functional actions in detail.

Implementation

The design of the application was based on the business requirements, database design, technologies used for the application, implementation methodologies used for coding, Xml-Html transformations. All these different sections were handled during the implementation phase of the project. The coding was done using the IDE and a RAD Tool Visual Studio .Net 2003. Unlike ASP, Web controls and the Code behind files of this VS tool assists the development work a lot. The Web.Config handles the configurations of the IIS server and other IP settings for the mail server etc. Finally the company assets could now be maintained in an efficient way. Since the application is built on the .Net Framework, it could use the future enhancements like the .Net Mobile application Framework.

5. Future Work and Enhancements

The application was earlier planned to have a number of additional features. In addition to storing basic information about each asset, the repository can include data integrated from a number of sources to provide updated inventory information, purchasing details, and financial costs, as well as support and maintenance contract details. Also the application could be enhanced to track the assets in terms of the employee, vendors etc., by having a separate project under the solution. So, for example if a consultant borrows or holds a particular asset and after a certain time frame he would be sent an email asked to return the asset (say a Laptop) back to the accounts department or to get their permission. So, future enhancements would be like calculating the depreciation costs for a particular asset, maintaining of the license agreements and the level of priority given to that particular asset as well.

Similarly I need to include the date in the employee table to identify when the asset was taken the employee in order to track efficiently.

6. Summary and Conclusions

Thus the benefits of the organization with the help of the ITAM system include:

- Accurate hardware, software license, and configuration information for purchasing decisions, and IT projects.
- Identification and subsequent redeployment, exchange, or credit for software licenses and hardware.
- Reduced risk of noncompliance with software licensing restrictions.
- Service contract quantities based on hard data can be compared with the vendor's estimate and used as the basis for successful vendor negotiations, resulting in savings.
- Reduction of unplanned expenses, such as lease and rental penalties for the late return of equipment.
- Availability of IT asset information to the service desk and technical support organizations.
- Reduction of administrative costs through the adoption of process-based roles and responsibilities and enabling technology.

This report has shown the application that was developed in BM InfoTech. The report also includes XML Web Service in general and the building blocks of any XML Web Service: SOAP, WSDL, and UDDI under Appendix C. Finally it shows how to build an XML Web Service in .NET Environment. The Web Service attributes, which is placed in top of the class, is an optional attributes that can have the service name and description. The lessons learnt while implementing the system include the Web based Client which involves more programming than the console based as we are using ASP.NET Web Components. While building the Web based client user friendliness factor was considered by giving the user better control over the transformed XML document. Also proper planning and execution of the project according to the plan is very important ie. Project Management plays an important role. Keeping a good focus on the sequence of the project would help the completion of project on time. It was a nice experience trying to enforce the business concepts in the technical environment.

References

1 Comparison between different Enterprise web Technologies like .NET

and J2EE Available online at

http://www.computerworld.com/developmenttopics/development/story/0,10801,77399,00.html

2 HP Leveraging HP IT Service Management to enable IT Asset

Management. (Online), Feb 2005: Available online at

http://cp.jupiterweb.com/index.php/846_itsm/a75bc21a892838411e5f73c29c8e5e45

3 The involvement of XML on web services.

Available online at http://www.xmlme.com/XmlWebServices.aspx

4 Deitel, H.M. & Deitel, P.J, J.Listfield.(2002). C# How to program. New

Jersey: ASP.NET (C#) [pp 895 – 1105]

5 Guerrero, F.G & Rogas, C.E. SQL SERVER 2000 Programming BY

EXAMPLE. USA : Transfering Data to & from SQL Server [617-665];

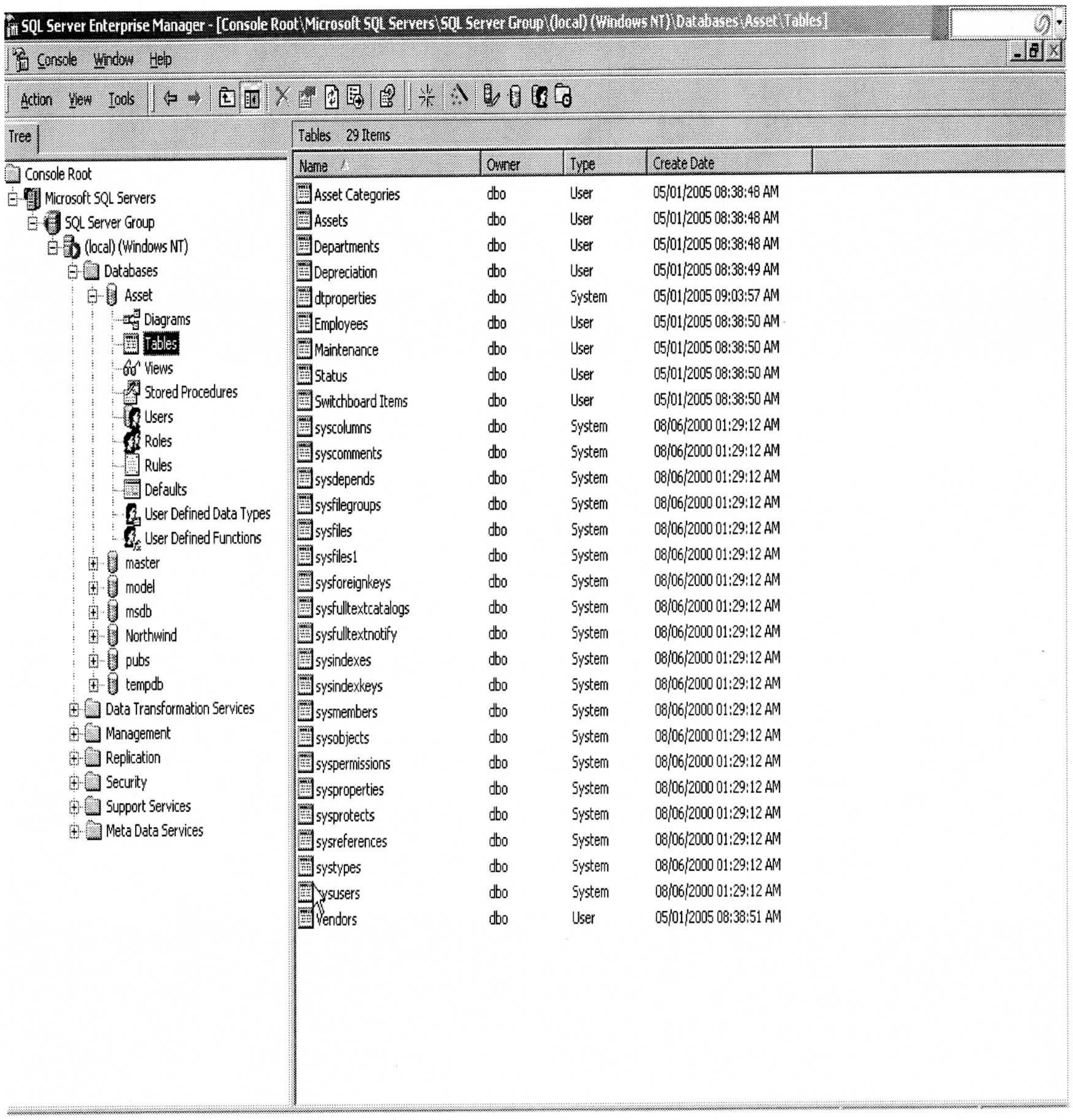
Querying and modifying Data [121-159]

6 Web Projects with Source Control and Versioning:

Available online at
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dvvstechart/html/vetchWebProjectsSourceControlIntegrationInVisualStudioNET.asp

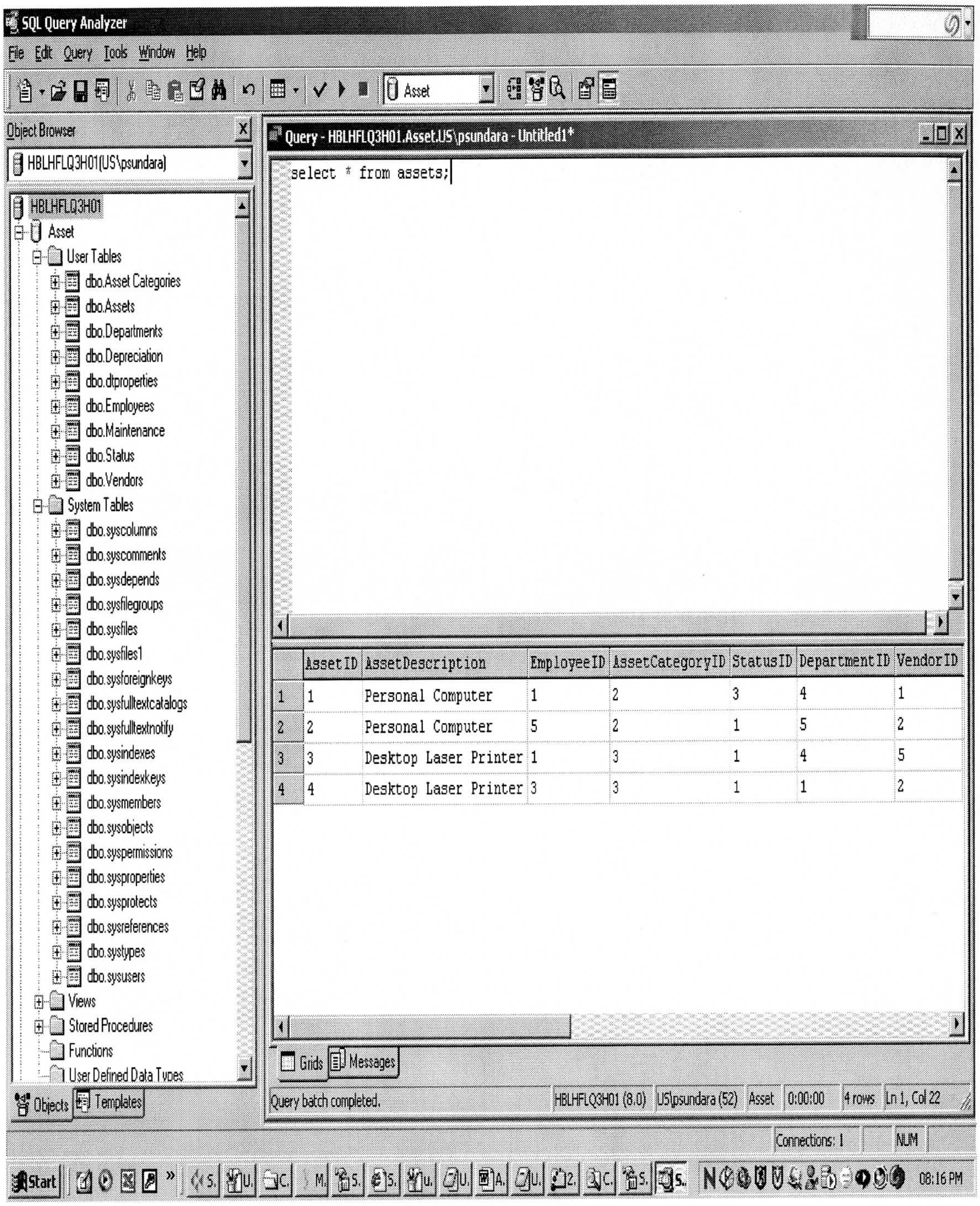
Appendix A Screen Shots

The below screenshots will walkthrough the application with an example. Also the sample .screenshots of the application includes the database and the client side interactions.



Name	Owner	Type	Create Date
Asset Categories	dbo	User	05/01/2005 08:38:48 AM
Assets	dbo	User	05/01/2005 08:38:48 AM
Departments	dbo	User	05/01/2005 08:38:48 AM
Depreciation	dbo	User	05/01/2005 08:38:49 AM
dtproperties	dbo	System	05/01/2005 09:03:57 AM
Employees	dbo	User	05/01/2005 08:38:50 AM
Maintenance	dbo	User	05/01/2005 08:38:50 AM
Status	dbo	User	05/01/2005 08:38:50 AM
Switchboard Items	dbo	User	05/01/2005 08:38:50 AM
syscolumns	dbo	System	08/06/2000 01:29:12 AM
syscomments	dbo	System	08/06/2000 01:29:12 AM
sysdepends	dbo	System	08/06/2000 01:29:12 AM
sysfilegroups	dbo	System	08/06/2000 01:29:12 AM
sysfiles	dbo	System	08/06/2000 01:29:12 AM
sysfiles1	dbo	System	08/06/2000 01:29:12 AM
sysforeignkeys	dbo	System	08/06/2000 01:29:12 AM
sysfulltextcatalogs	dbo	System	08/06/2000 01:29:12 AM
sysfulltextnotify	dbo	System	08/06/2000 01:29:12 AM
sysindexes	dbo	System	08/06/2000 01:29:12 AM
sysindexkeys	dbo	System	08/06/2000 01:29:12 AM
sysmembers	dbo	System	08/06/2000 01:29:12 AM
sysobjects	dbo	System	08/06/2000 01:29:12 AM
syspermissions	dbo	System	08/06/2000 01:29:12 AM
sysproperties	dbo	System	08/06/2000 01:29:12 AM
sysprotects	dbo	System	08/06/2000 01:29:12 AM
sysreferences	dbo	System	08/06/2000 01:29:12 AM
systypes	dbo	System	08/06/2000 01:29:12 AM
sysusers	dbo	System	08/06/2000 01:29:12 AM
Vendors	dbo	User	05/01/2005 08:38:51 AM

Fig A-1: Shows the Asset database and its tables.

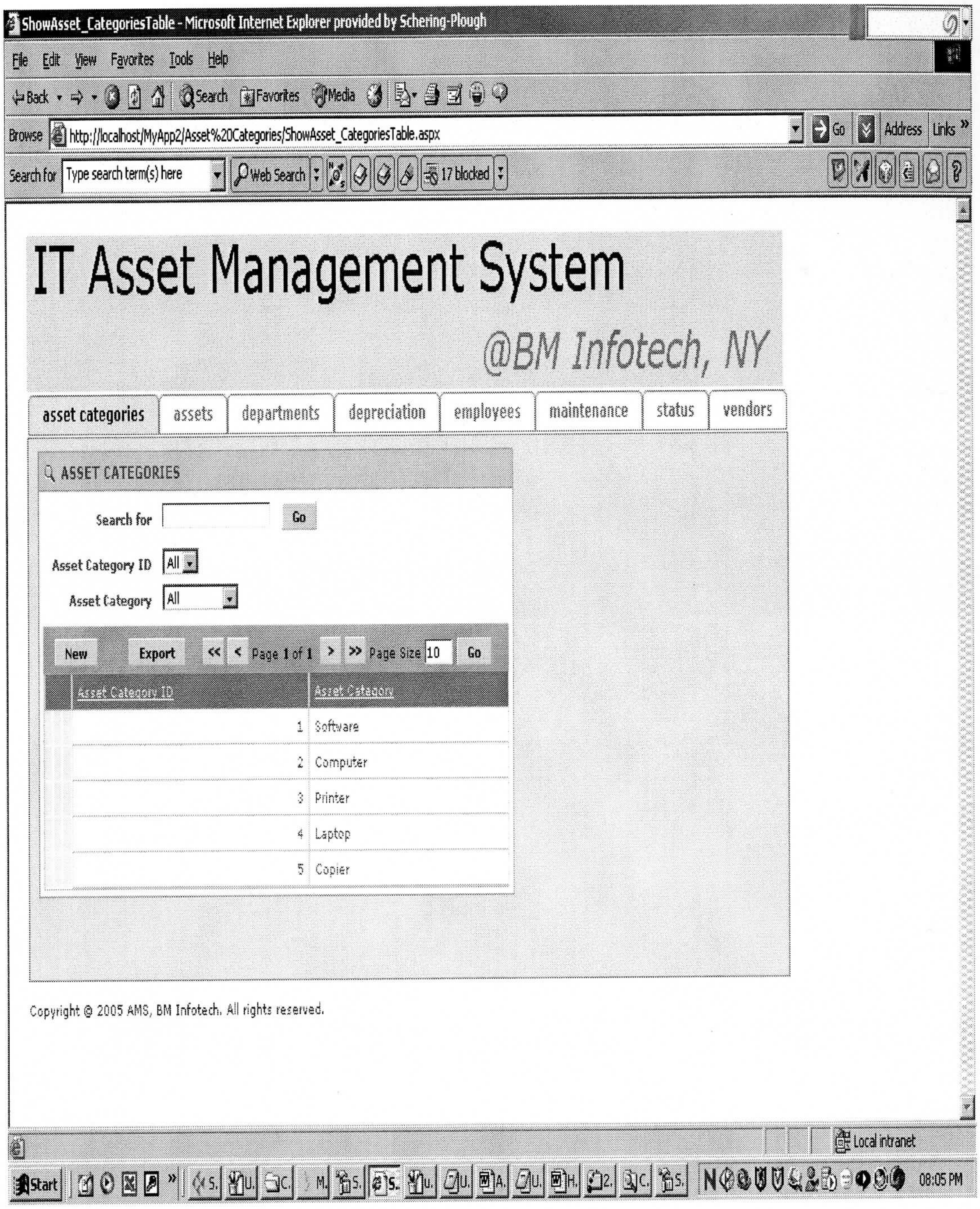


```
select * from assets;
```

	AssetID	AssetDescription	EmployeeID	AssetCategoryID	StatusID	DepartmentID	VendorID
1	1	Personal Computer	1	2	3	4	1
2	2	Personal Computer	5	2	1	5	2
3	3	Desktop Laser Printer	1	3	1	4	5
4	4	Desktop Laser Printer	3	3	1	1	2

```
Query batch completed.    HBLHFLQ3H01 (8.0)    US\psundara (52)    Asset    0:00:00    4 rows    Ln 1, Col 22
```

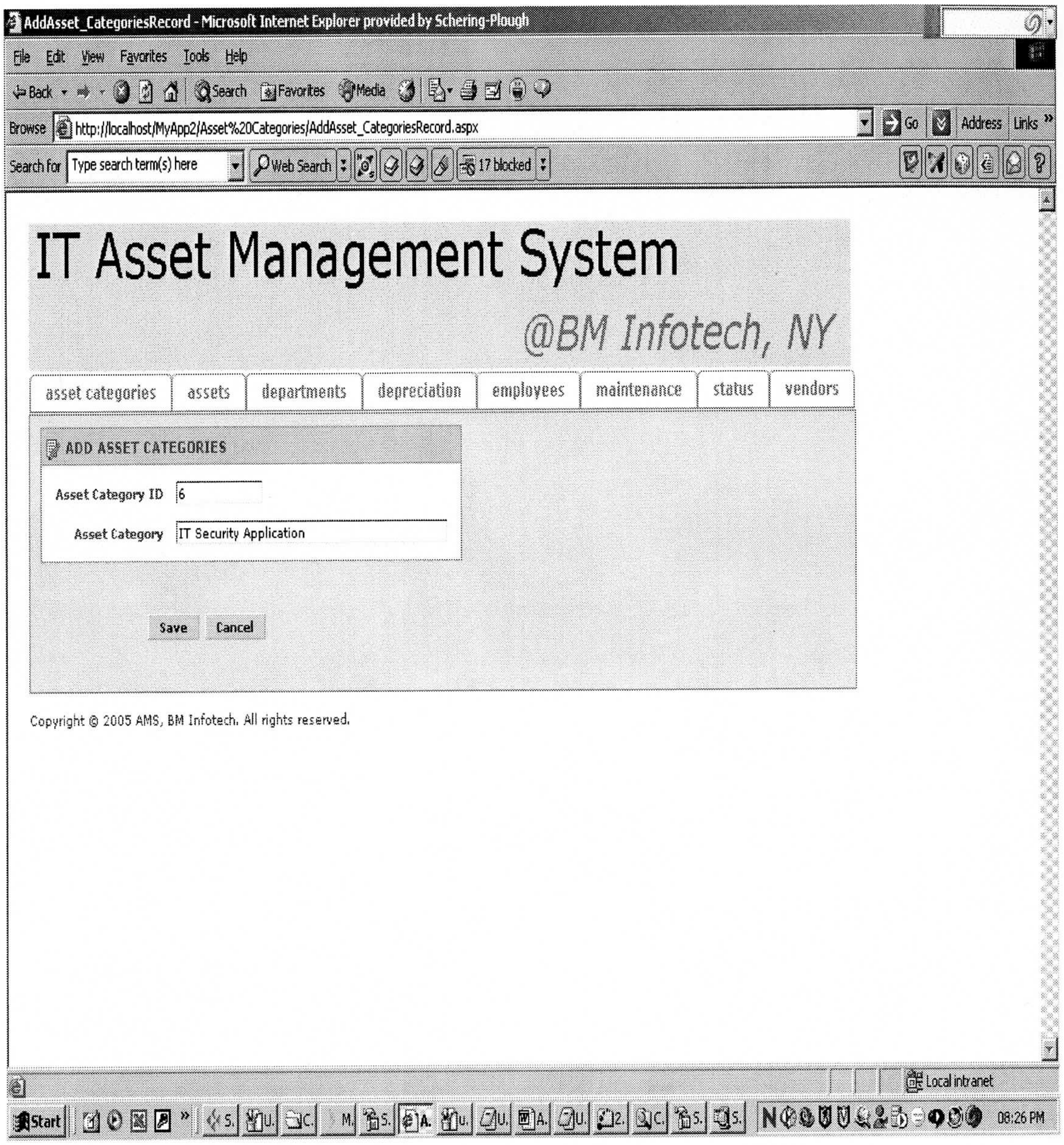
Fig A-2: Shows the values of the 'Assets' Table details


ShowAsset_CategoriesTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
http://localhost/MyApp2/Asset%20Categories/ShowAsset_CategoriesTable.aspx
IT Asset Management System
@BM Infotech, NY
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
ASSET CATEGORIES
Search for
Go
Asset Category ID All
Asset Category All
New
Export
Page 1 of 1
Page Size 10
Local intranet
08:05 PM

Asset Category ID	Asset Category
1	Software
2	Computer
3	Printer
4	Laptop
5	Copier

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-3: Shows all the values of the Asset Categories table.

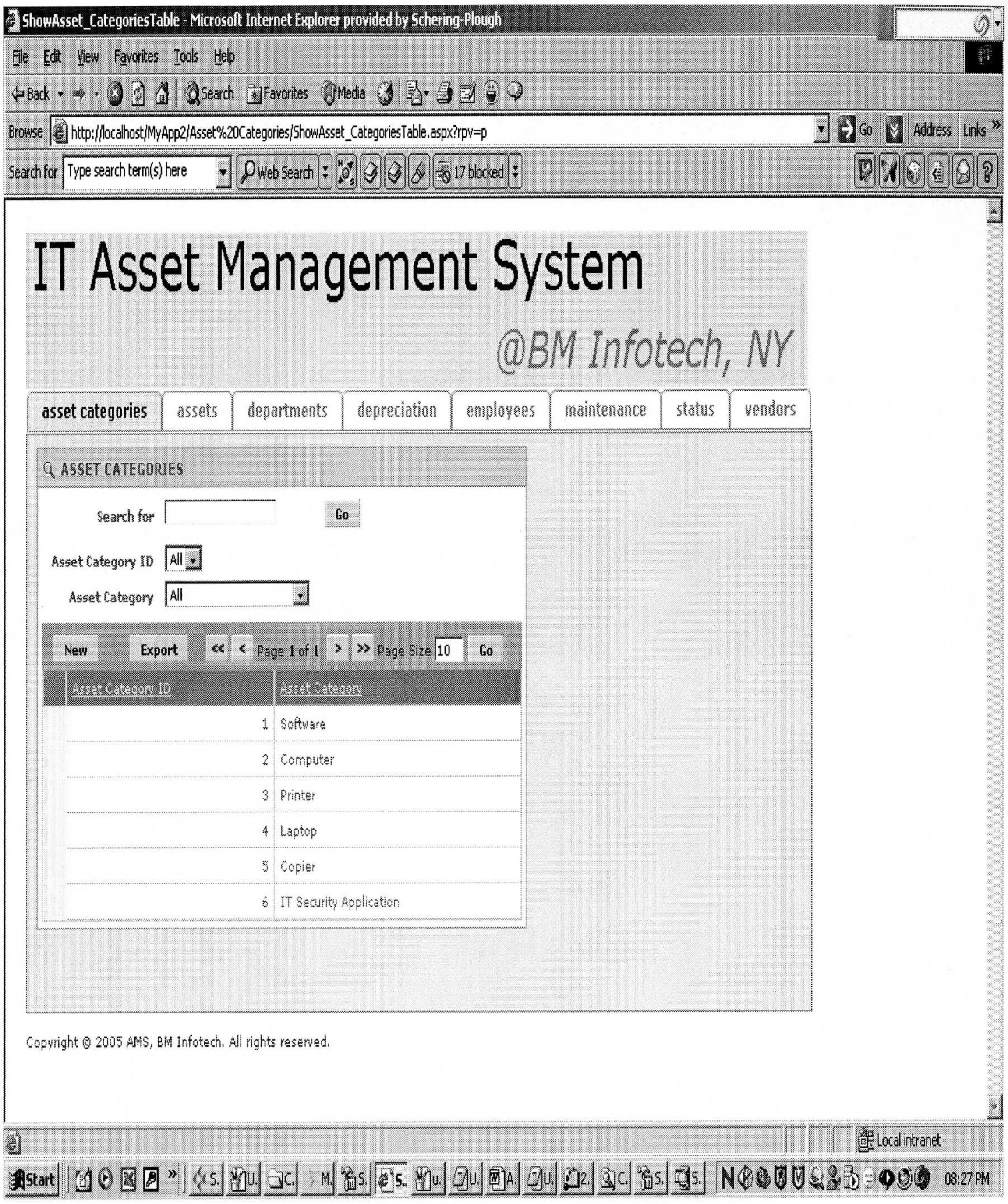

AddAsset_CategoriesRecord - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
http://localhost/MyApp2/Asset%20Categories/AddAsset_CategoriesRecord.aspx
Go Address Links
Search for Type search term(s) here Web Search 17 blocked
IT Asset Management System
@BM Infotech, NY
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
ADD ASSET CATEGORIES
Save
Cancel
Local intranet
Start
08:26 PM

Asset Category ID 6

Asset Category IT Security Application

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-3: Adding a new asset category 'IT Security Application' in the Asset Categories table



ShowAsset_CategoriesTable - Microsoft Internet Explorer provided by Schering-Plough

http://localhost/MyApp2/Asset%20Categories/ShowAsset_CategoriesTable.aspx?rpv=p

IT Asset Management System

@BM Infotech, NY

Asset Category ID: All

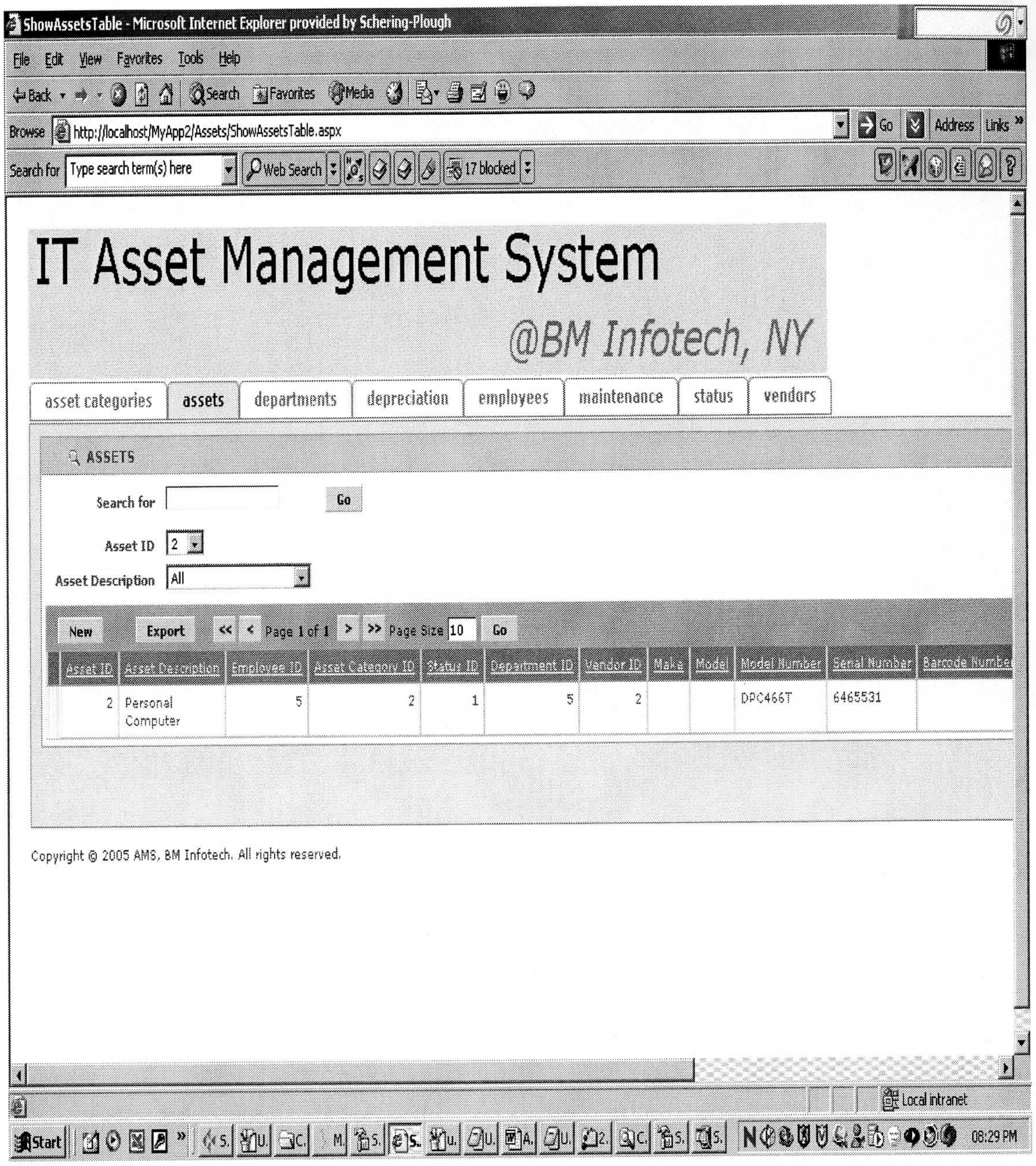
Asset Category: All

Page Size: 10

Asset Category ID	Asset Category
1	Software
2	Computer
3	Printer
4	Laptop
5	Copier
6	IT Security Application

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-5: Shows the 'asset categories' Table values after adding the new asset 'IT Security Application'.


ShowAssetsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/MyApp2/Assets/ShowAssetsTable.aspx
Search for Type search term(s) here Web Search 17 blocked
asset categories assets departments depreciation employees maintenance status vendors
Local intranet
Start 08:29 PM

IT Asset Management System

@BM Infotech, NY

ASSETS

Search for: Go

Asset ID: 2

Asset Description: All

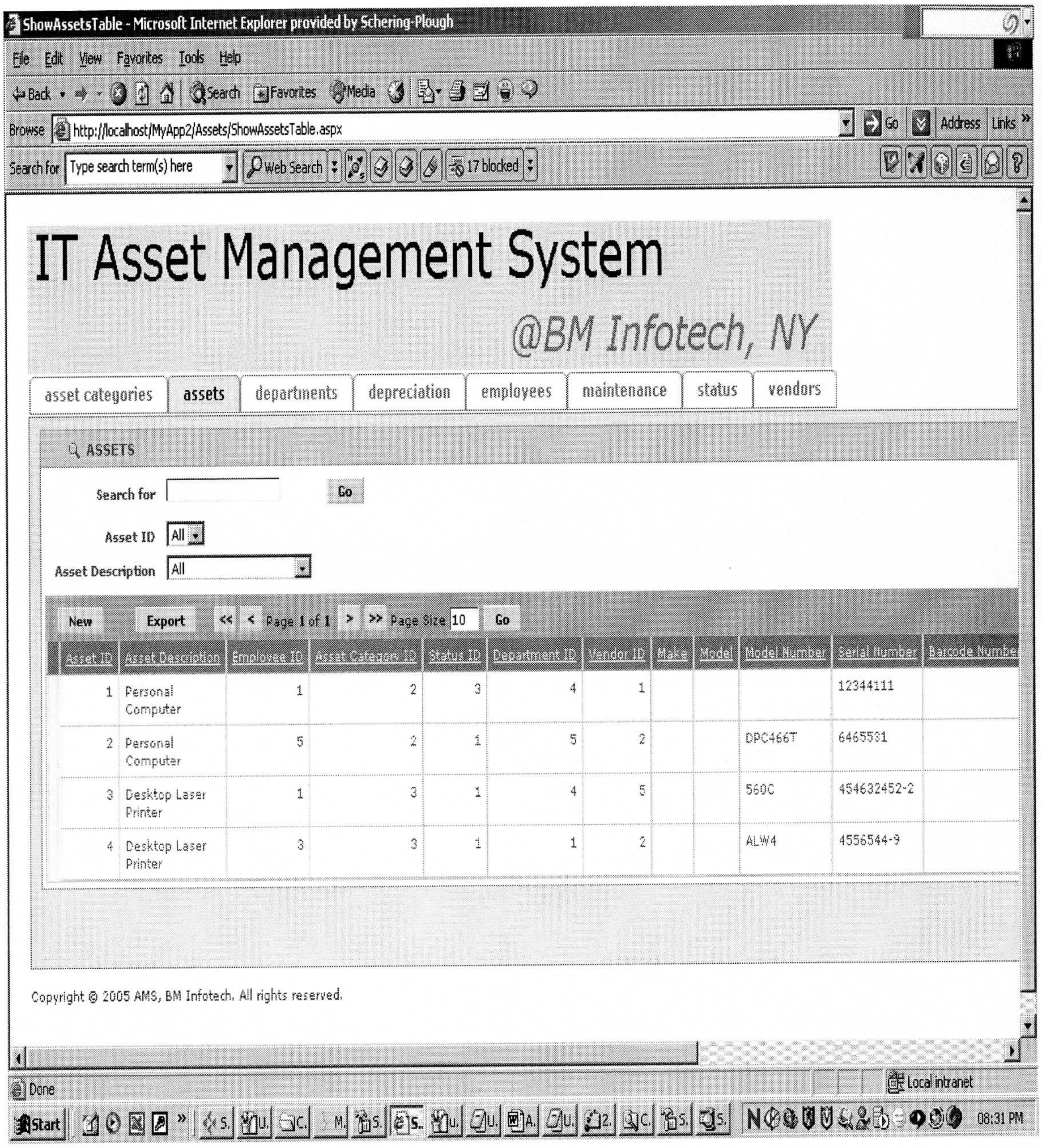
New Export << < Page 1 of 1 > >> Page Size 10 Go

Asset ID	Asset Description	Employee ID	Asset Category ID	Status ID	Department ID	Vendor ID	Make	Model	Model Number	Serial Number	Barcode Numbe
2	Personal Computer	5	2	1	5	2			DPC466T	6465531	

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-6: Can see the value with a particular Asset ID. The value shown is for Asset ID = 2

]


ShowAssetsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
Browse http://localhost/MyApp2/Assets/ShowAssetsTable.aspx
Go Address Links
Search for Type search term(s) here Web Search 17 blocked

IT Asset Management System

@BM Infotech, NY

asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
ASSETS

Search for [ ] Go

Asset ID All

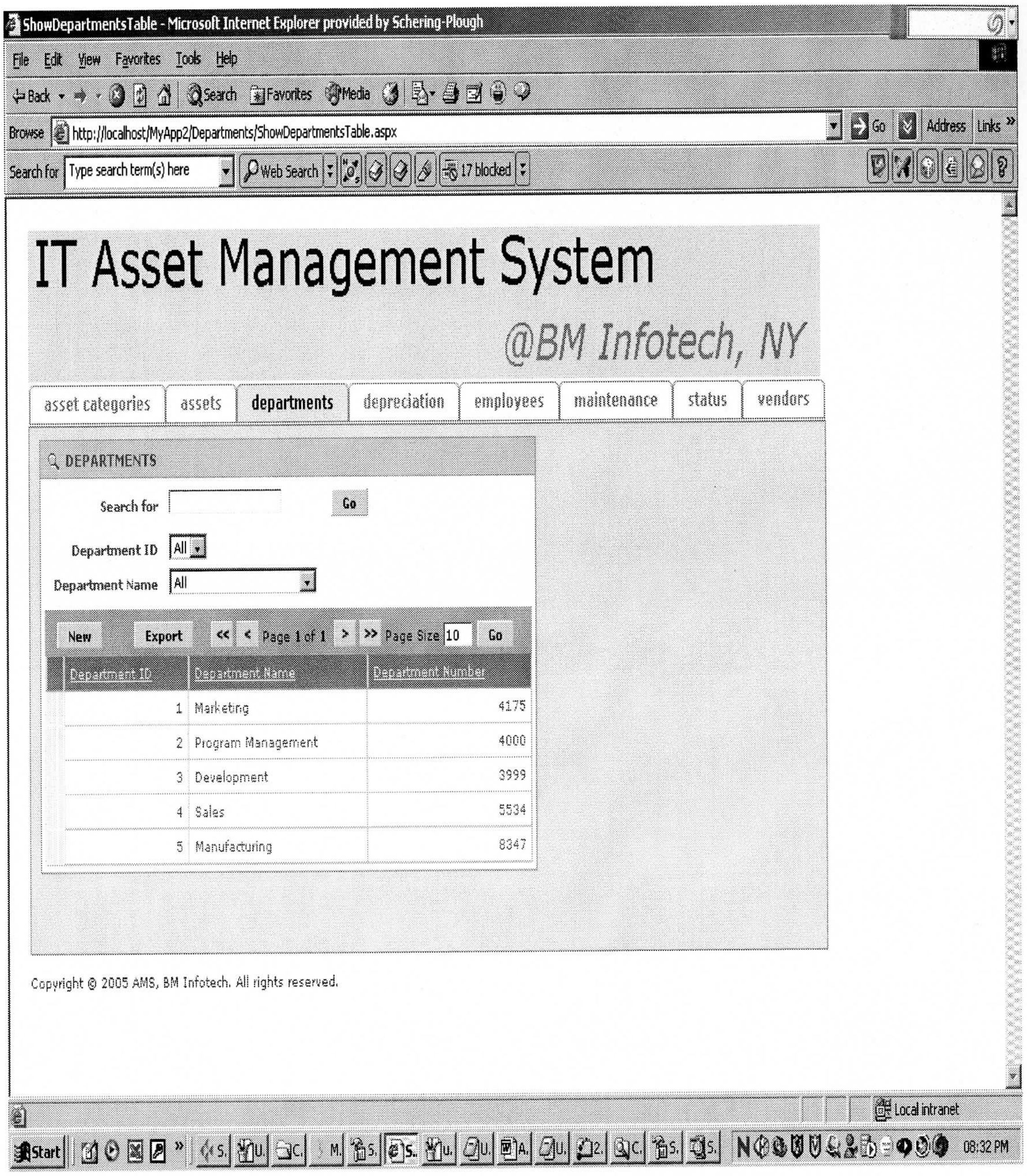
Asset Description All

New
Export
<< < Page 1 of 1 > >> Page Size 10 Go

Asset ID	Asset Description	Employee ID	Asset Category ID	Status ID	Department ID	Vendor ID	Make	Model	Model Number	Serial Number	Barcode Numbe
1	Personal Computer	1	2	3	4	1				12344111	
2	Personal Computer	5	2	1	5	2			DPC466T	6465531	
3	Desktop Laser Printer	1	3	1	4	5			560C	454632452-2	
4	Desktop Laser Printer	3	3	1	1	2			ALW4	4556544-9	

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-7: Shows the Asset Table values


ShowDepartmentsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
http://localhost/MyApp2/Departments/ShowDepartmentsTable.aspx
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
New
Export
Page 1 of 1
Page Size 10
Go
Local intranet
08:32 PM

IT Asset Management System

@BM Infotech, NY

DEPARTMENTS

Search for

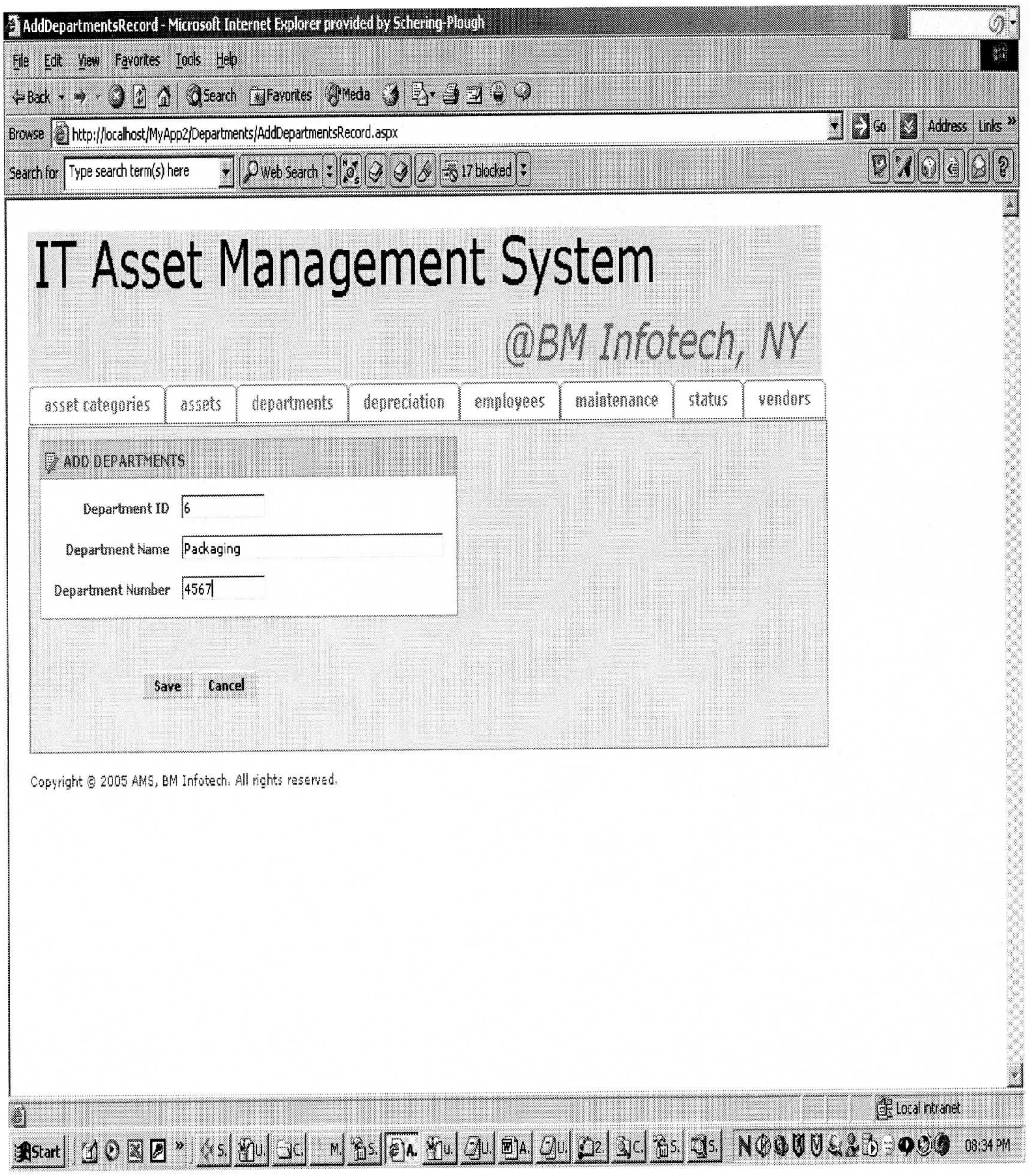
Department ID: All

Department Name: All

Department ID	Department Name	Department Number
1	Marketing	4175
2	Program Management	4000
3	Development	3999
4	Sales	5534
5	Manufacturing	8347

Copyright © 2005 AMS, BM Infotech. All rights reserved.

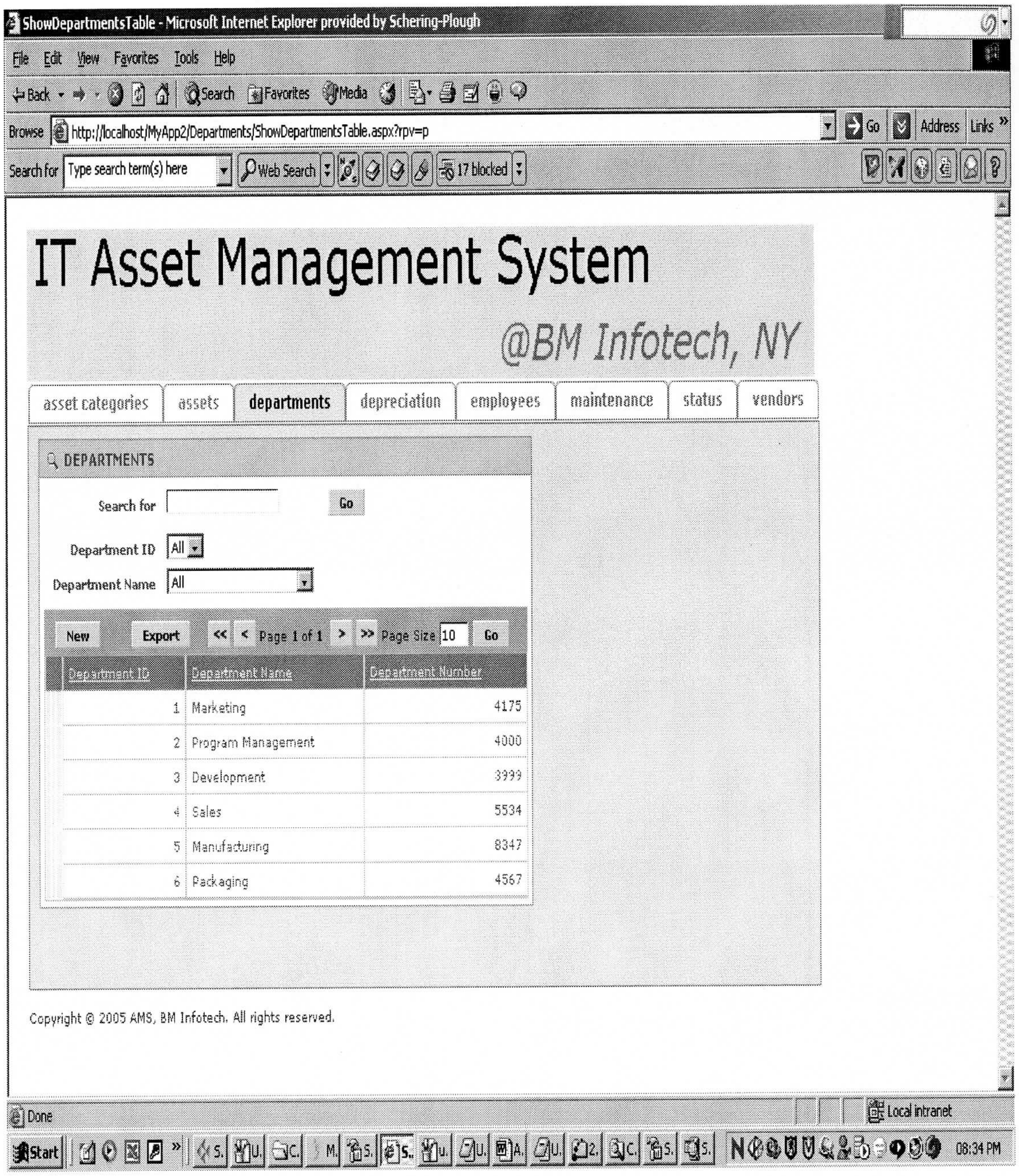
Fig A-8: Shows the Current Values of the 'Departments' table


AddDepartmentsRecord - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
http://localhost/MyApp2/Departments/AddDepartmentsRecord.aspx
IT Asset Management System
@BM Infotech, NY
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
ADD DEPARTMENTS
Save
Cancel

Department ID	6
Department Name	Packaging
Department Number	4567

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-9: Shows adding a new department 'Packaging'


ShowDepartmentsTable - Microsoft Internet Explorer provided by Schering-Plough

http://localhost/MyApp2/Departments/ShowDepartmentsTable.aspx?rpv=p

IT Asset Management System

@BM Infotech, NY

asset categories | assets | departments | depreciation | employees | maintenance | status | vendors

DEPARTMENTS

Search for: Go

Department ID: All

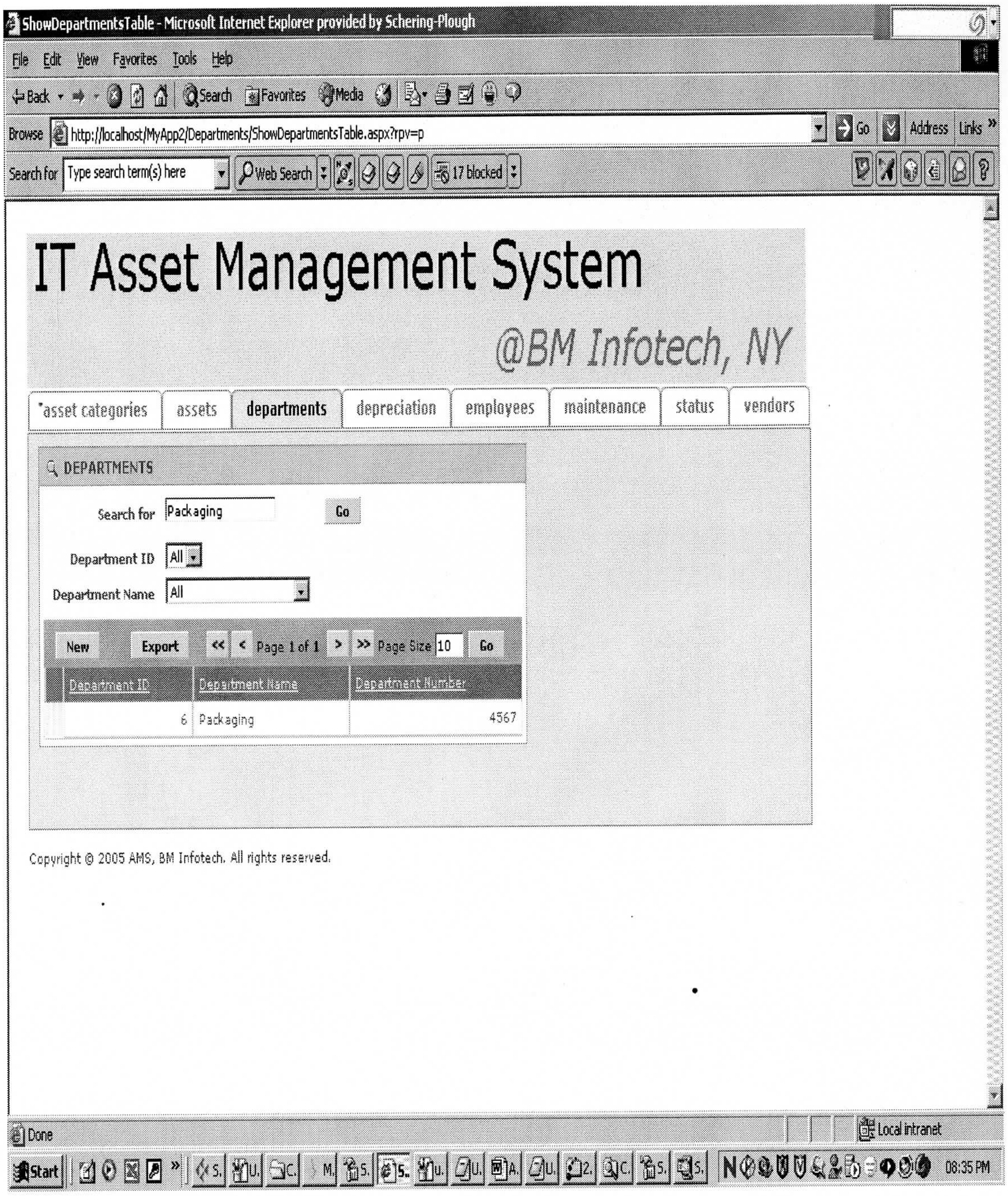
Department Name: All

New | Export | << < Page 1 of 1 > >> | Page Size 10 | Go

Department ID	Department Name	Department Number
1	Marketing	4175
2	Program Management	4000
3	Development	3999
4	Sales	5534
5	Manufacturing	8347
6	Packaging	4567

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-10: Shows the added department 'Packaging'



ShowDepartmentsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
http://localhost/MyApp2/Departments/ShowDepartmentsTable.aspx?rpv=p
IT Asset Management System
@BM Infotech, NY
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
DEPARTMENTS
New
Export
Page 1 of 1
Page Size
Go
Done
Local intranet
08:35 PM

Search for: Packaging

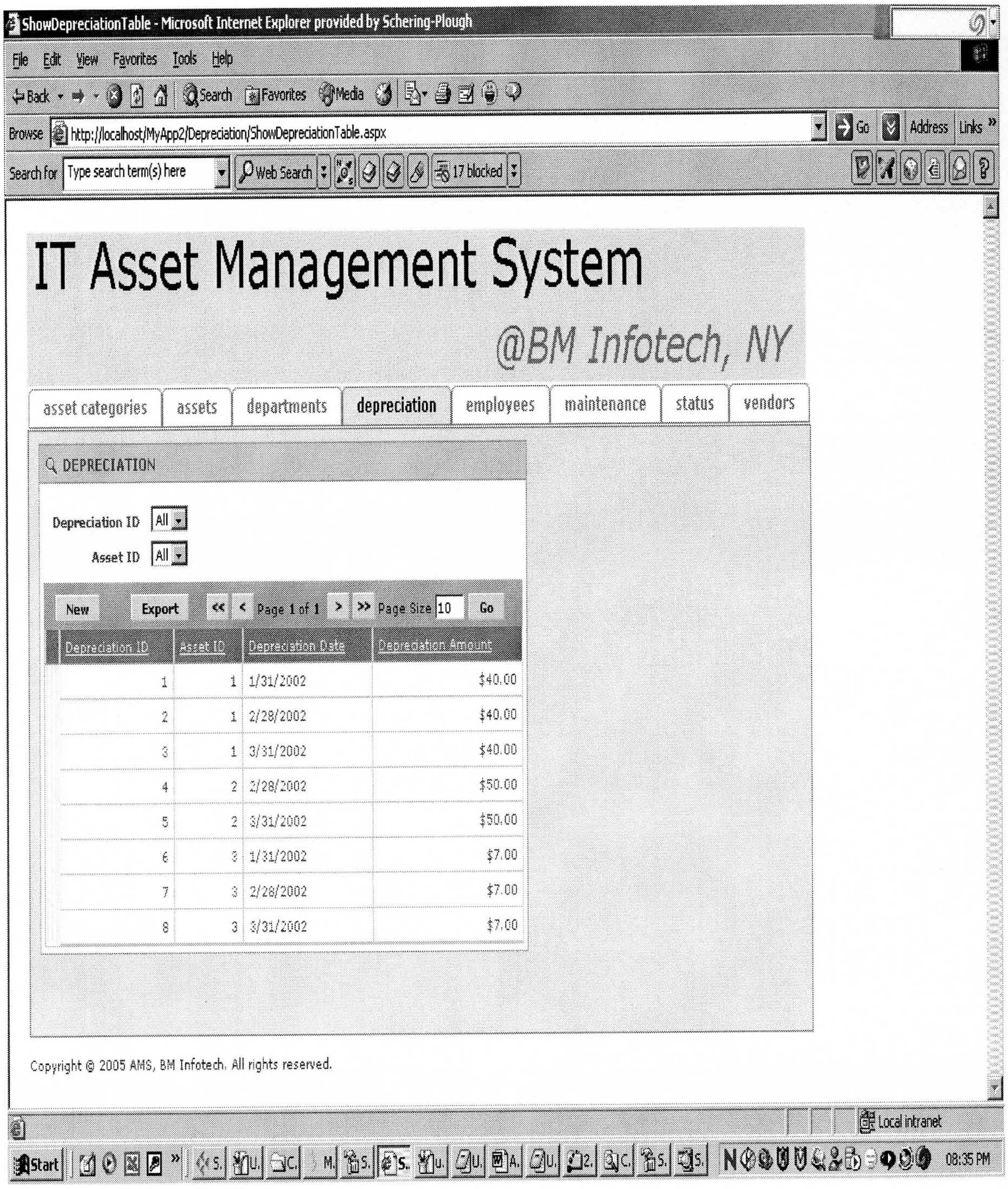
Department ID: All

Department Name: All

Department ID	Department Name	Department Number
6	Packaging	4567

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-11: Can search for a particular department 'Packaging'


ShowDepreciationTable - Microsoft Internet Explorer provided by Schering-Plough
http://localhost/MyApp2/Depreciation/ShowDepreciationTable.aspx
IT Asset Management System
@BM Infotech, NY
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
DEPRECIATION
New
Export
Page 1 of 1
Page Size 10
Go

Depreciation ID All

Asset ID All

Depreciation ID	Asset ID	Depreciation Date	Depreciation Amount
1	1	1/31/2002	$40.00
2	1	2/28/2002	$40.00
3	1	3/31/2002	$40.00
4	2	2/28/2002	$50.00
5	2	3/31/2002	$50.00
6	3	1/31/2002	$7.00
7	3	2/28/2002	$7.00
8	3	3/31/2002	$7.00

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-12: Shows all the values of the Depreciation Tables

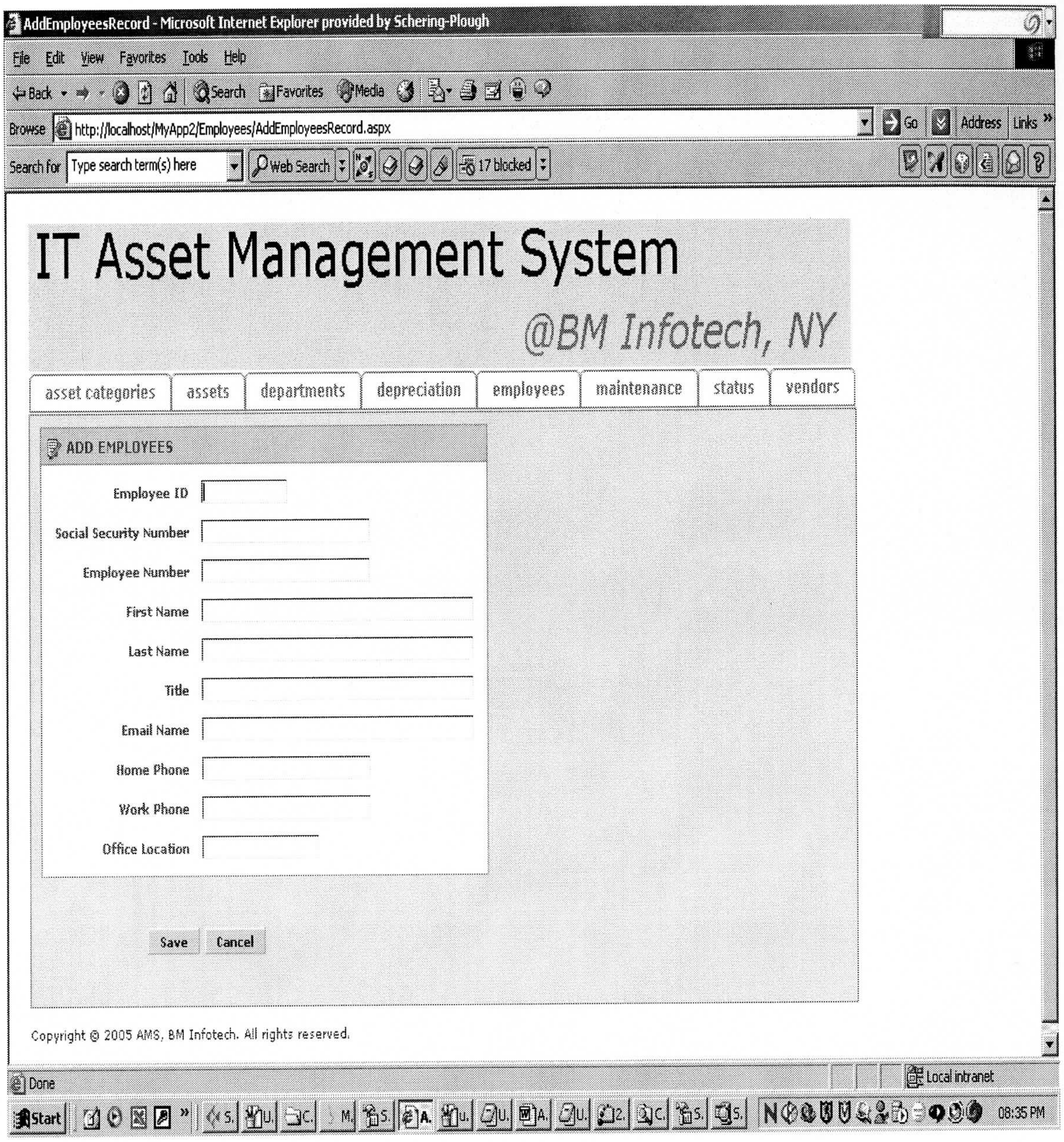
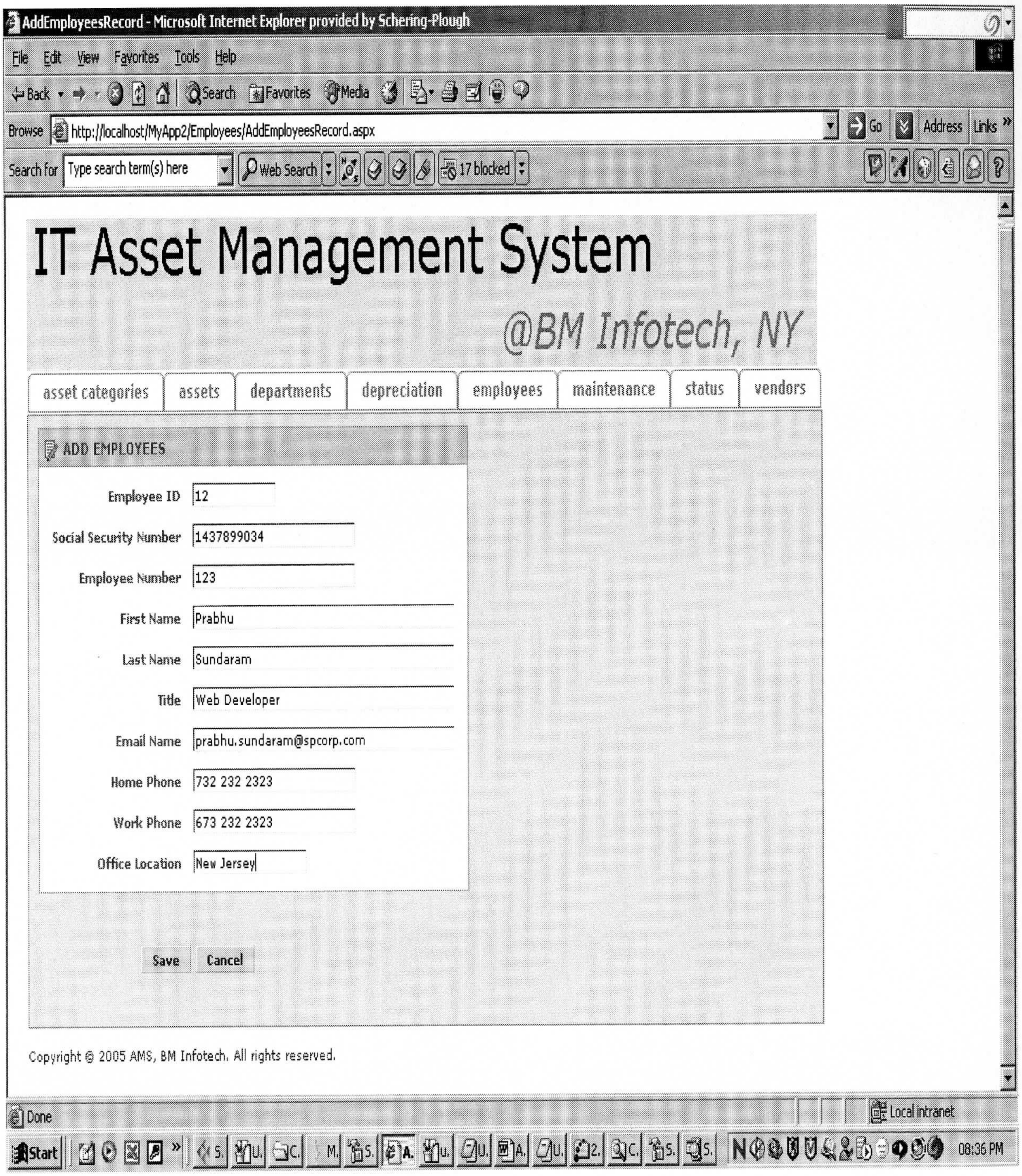

AddEmployeesRecord - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
http://localhost/MyApp2/Employees/AddEmployeesRecord.aspx
IT Asset Management System
@BM Infotech, NY
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
ADD EMPLOYEES
Employee ID
Social Security Number
Employee Number
First Name
Last Name
Title
Email Name
Home Phone
Work Phone
Office Location
Save
Cancel
Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A- 13: Adding a new employee to the Employees Table


AddEmployeesRecord - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
Browse http://localhost/MyApp2/Employees/AddEmployeesRecord.aspx Go Address Links
Search for Type search term(s) here Web Search 17 blocked
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
ADD EMPLOYEES
Save
Cancel
Done
Local intranet
Start
08:36 PM

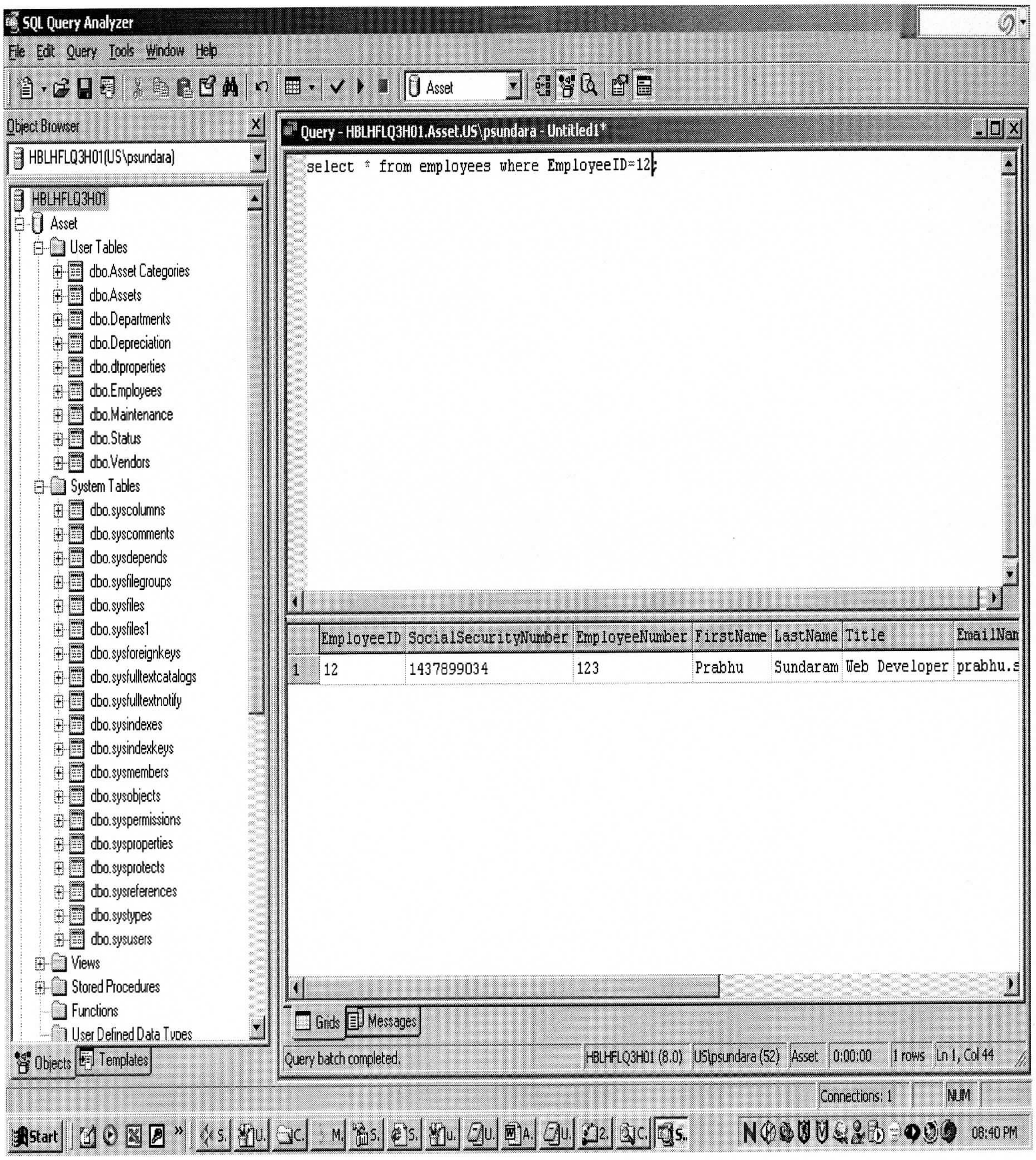
IT Asset Management System

@BM Infotech, NY

Employee ID	12
Social Security Number	1437899034
Employee Number	123
First Name	Prabhu
Last Name	Sundaram
Title	Web Developer
Email Name	prabhu.sundaram@spcorp.com
Home Phone	732 232 2323
Work Phone	673 232 2323
Office Location	New Jersey

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A- 14: Adding an employee named 'Prabhu'

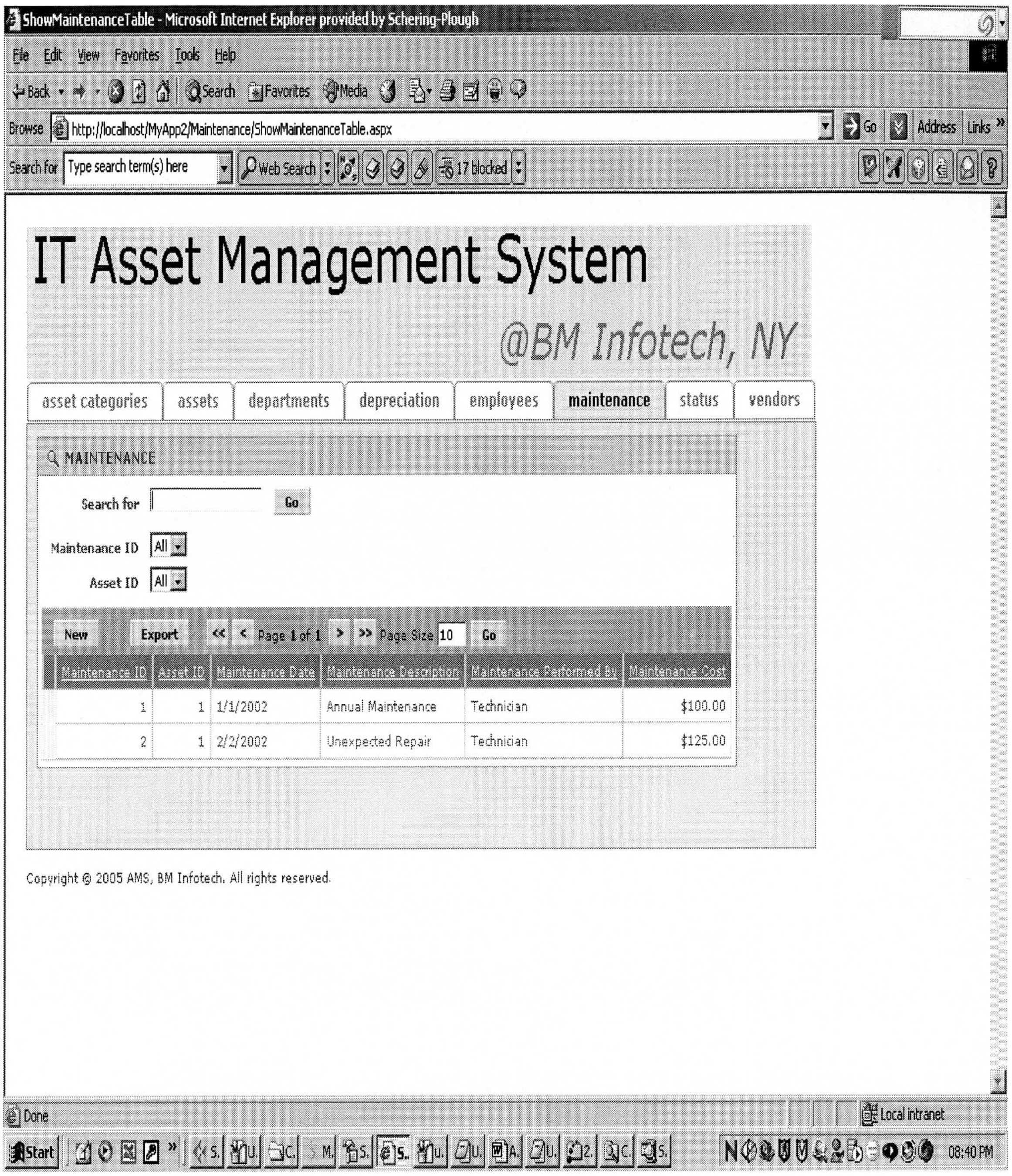


```
select * from employees where EmployeeID=12;
```

```
  EmployeeID SocialSecurityNumber EmployeeNumber FirstName LastName Title         EmailNan
1 12         1437899034           123            Prabhu    Sundaram Web Developer prabhu.s
```

```
Query batch completed.   HBLHFLQ3H01 (8.0)   US\psundara (52)   Asset   0:00:00   1 rows   Ln 1, Col 44
```

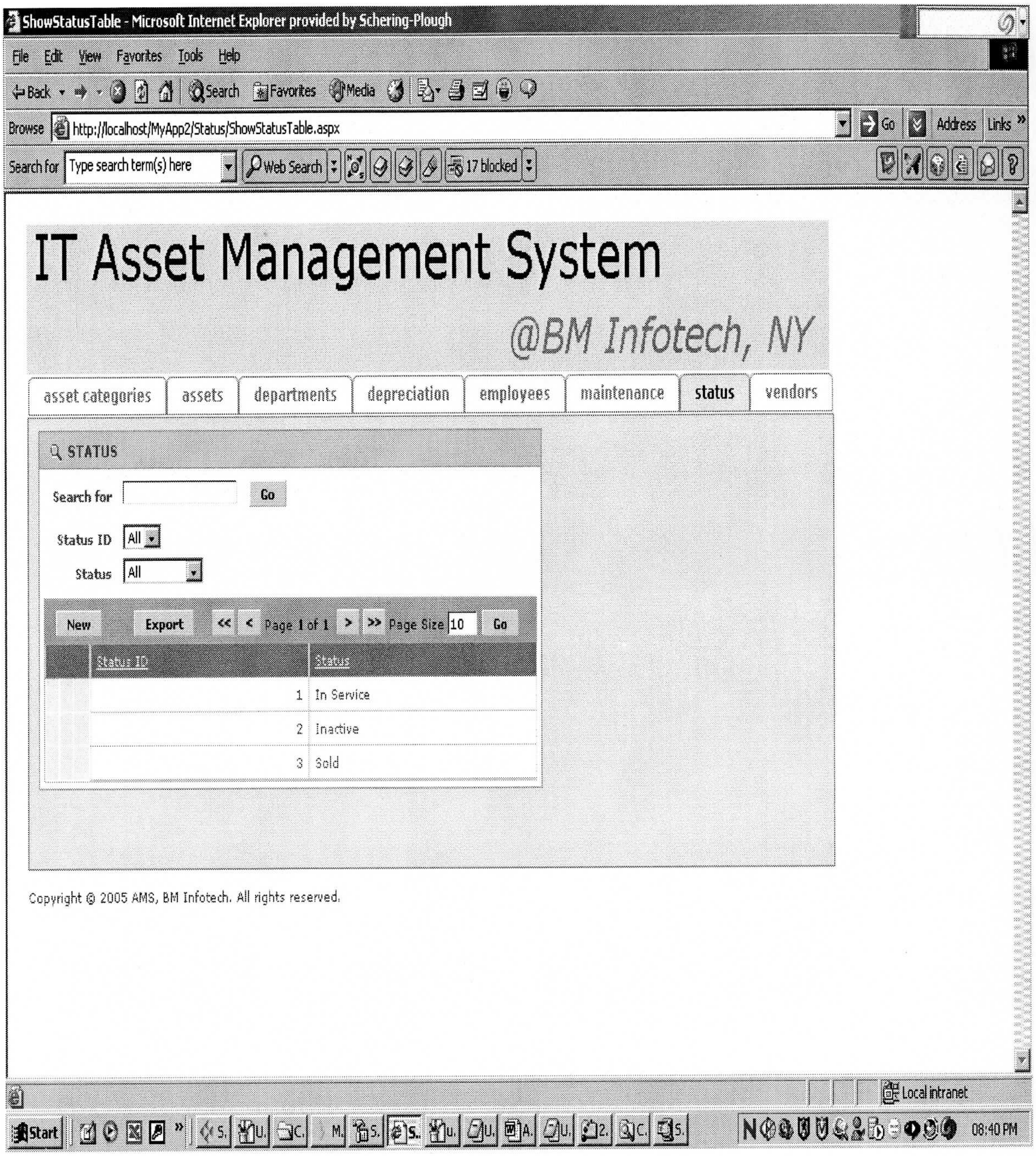
Fig A-15: Retrieving the Record from the database for EmployeeID ‘12’



Maintenance ID	Asset ID	Maintenance Date	Maintenance Description	Maintenance Performed By	Maintenance Cost
1	1	1/1/2002	Annual Maintenance	Technician	$100.00
2	1	2/2/2002	Unexpected Repair	Technician	$125.00

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A- 16: Showing the records of the Maintenance table

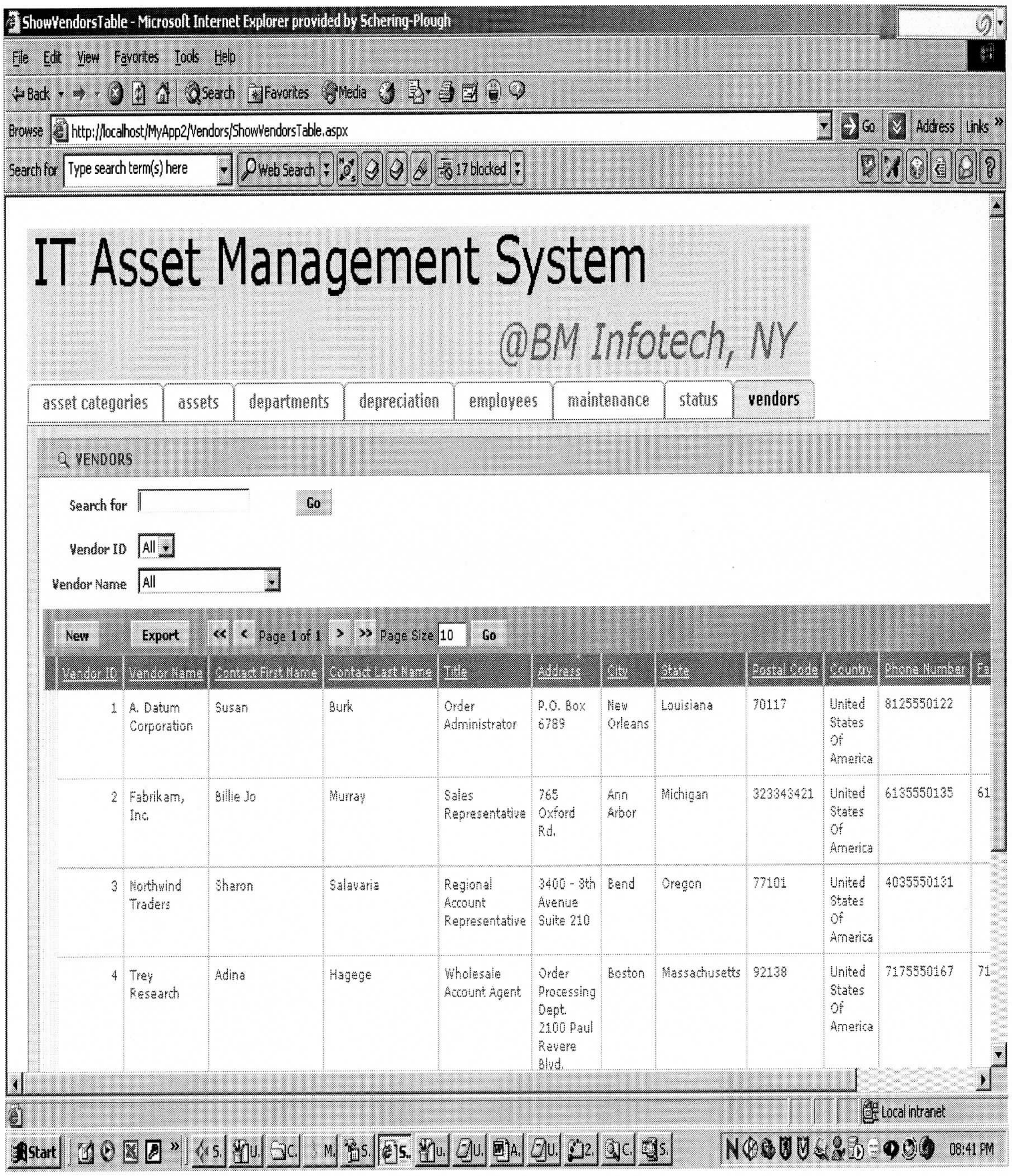


Status ID	All
Status	All

Status ID	Status
1	In Service
2	Inactive
3	Sold

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A- 17: Showing the records of the Status table

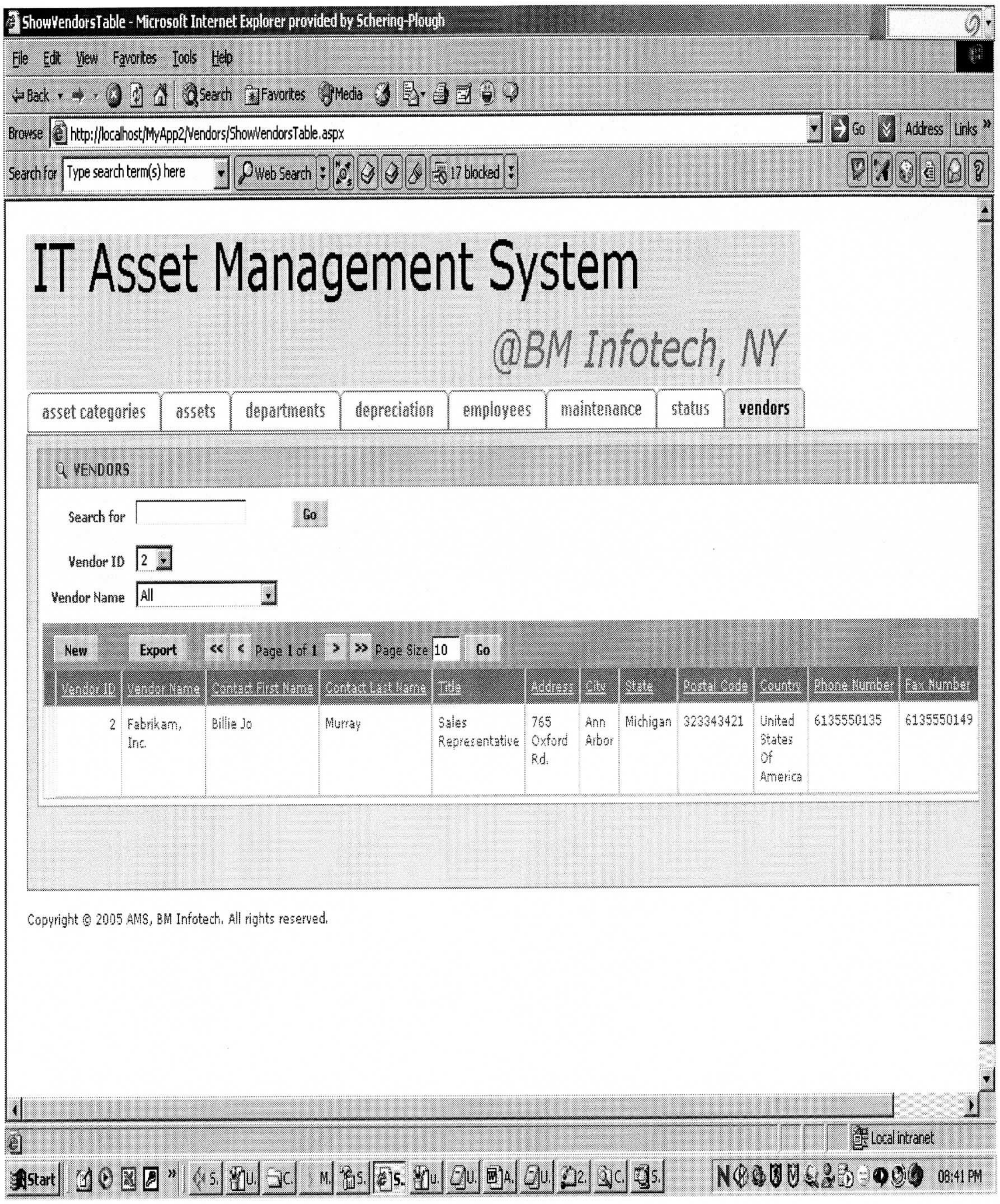

ShowVendorsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/MyApp2/Vendors/ShowVendorsTable.aspx
Go Links
Search for Type search term(s) here Web Search 17 blocked
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
VENDORS
Search for
Go
Vendor ID All
Vendor Name All
New
Export
Page 1 of 1
Page Size 10
Go
Local intranet
Start
08:41 PM

IT Asset Management System

@BM Infotech, NY

Vendor ID	Vendor Name	Contact First Name	Contact Last Name	Title	Address	City	State	Postal Code	Country	Phone Number	Fa
1	A. Datum Corporation	Susan	Burk	Order Administrator	P.O. Box 6789	New Orleans	Louisiana	70117	United States Of America	8125550122	
2	Fabrikam, Inc.	Billie Jo	Murray	Sales Representative	765 Oxford Rd.	Ann Arbor	Michigan	323343421	United States Of America	6135550135	61
3	Northwind Traders	Sharon	Salavaria	Regional Account Representative	3400 - 8th Avenue Suite 210	Bend	Oregon	77101	United States Of America	4035550131	
4	Trey Research	Adina	Hagege	Wholesale Account Agent	Order Processing Dept. 2100 Paul Revere Blvd.	Boston	Massachusetts	92138	United States Of America	7175550167	71

Fig A-18: Showing the records of the Vendors table


ShowVendorsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
Browse http://localhost/MyApp2/Vendors/ShowVendorsTable.aspx
Go Address Links
Search for Type search term(s) here Web Search 17 blocked
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
VENDORS
New
Export
Page 1 of 1
Page Size 10
Go
Local intranet
Start
08:41 PM

IT Asset Management System

@BM Infotech, NY

Search for

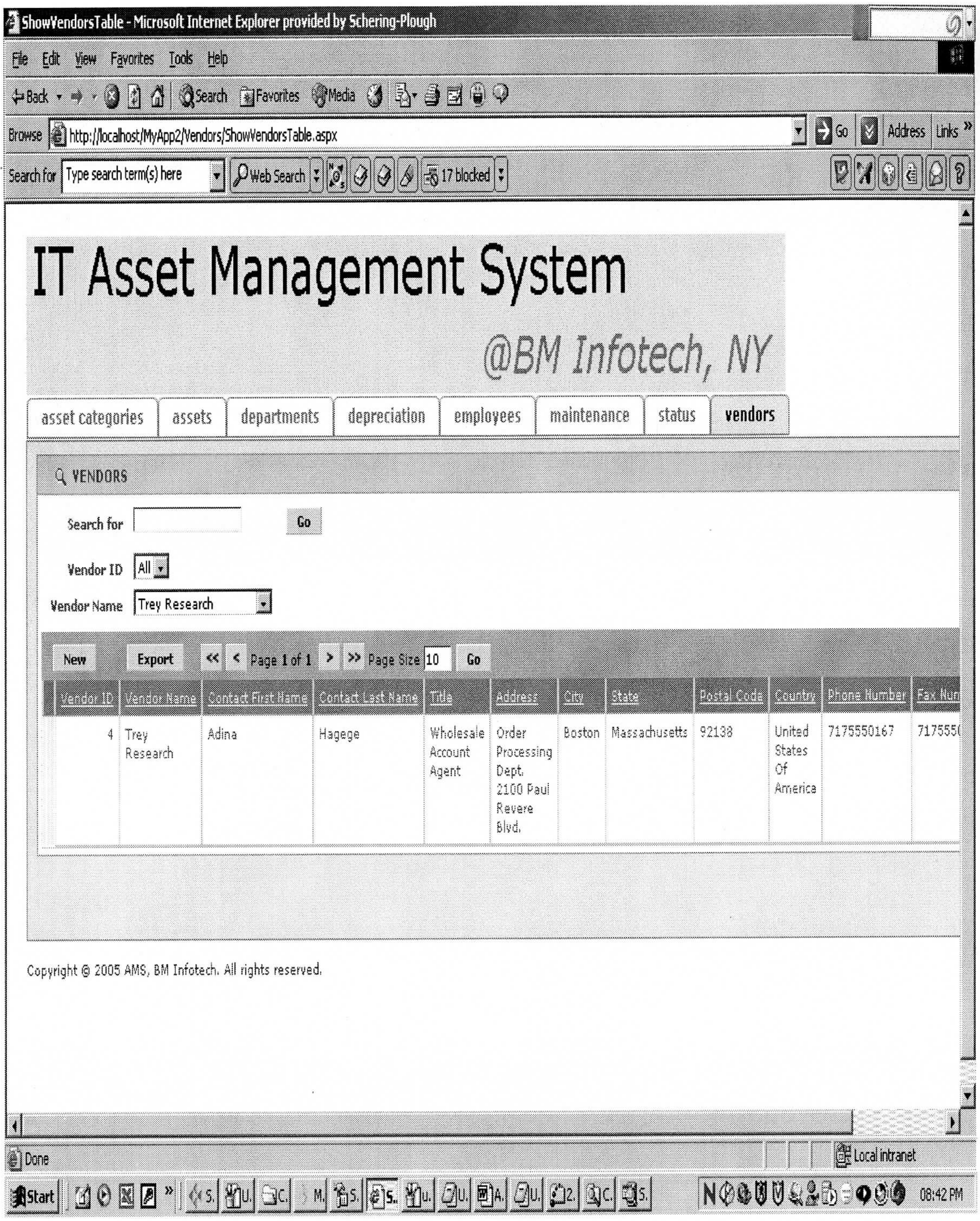
Vendor ID 2

Vendor Name All

Vendor ID	Vendor Name	Contact First Name	Contact Last Name	Title	Address	City	State	Postal Code	Country	Phone Number	Fax Number
2	Fabrikam, Inc.	Billie Jo	Murray	Sales Representative	765 Oxford Rd.	Ann Arbor	Michigan	323343421	United States Of America	6135550135	6135550149

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-19: Showing the records of the Vendors table through 'Vendor ID'


ShowVendorsTable - Microsoft Internet Explorer provided by Schering-Plough
File Edit View Favorites Tools Help
Back Search Favorites Media
Browse http://localhost/MyApp2/Vendors/ShowVendorsTable.aspx
Go Address Links
Search for Type search term(s) here Web Search 17 blocked
IT Asset Management System
@BM Infotech, NY
asset categories
assets
departments
depreciation
employees
maintenance
status
vendors
VENDORS
Go
New
Export
Page 1 of 1
Go
Done
Local intranet
Start
08:42 PM

Search for

Vendor ID All

Vendor Name Trey Research

Page Size 10

Vendor ID	Vendor Name	Contact First Name	Contact Last Name	Title	Address	City	State	Postal Code	Country	Phone Number	Fax Nun
4	Trey Research	Adina	Hagege	Wholesale Account Agent	Order Processing Dept. 2100 Paul Revere Blvd.	Boston	Massachusetts	92138	United States Of America	7175550167	717555

Copyright © 2005 AMS, BM Infotech. All rights reserved.

Fig A-20: Showing the records of the Vendors table through 'Vendor Name'

Appendix B Program Codes

Program # 1

Name: AddAsset.cs (C# Code for adding assets to the database)

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;

namespace MyApp2
{
public AddAssetsRecordGen()
   {
      base.Init += new System.EventHandler(this.Page_Init);
      base.Load += new System.EventHandler(this.Page_Load);


   }

   private void Page_Load(object sender, System.EventArgs e)
   {
      if (!this.IsPostBack)
      {
         this.DataBind();
      }
   }
// web control variable properties code
}
}
```

Program # 2
Name: AddAssets.aspx (ASP.NET Code for Adding Assets)

```
<%@ Register Tagprefix="MyApp2" TagName="Button"
Src="../Shared/Button.ascx" %>

<%@ Register Tagprefix="MyApp2" TagName="Menu" Src="../Menu
Panels/Menu.ascx" %>
<%@ Register Tagprefix="MyApp2" TagName="Footer" Src="../Header &
Footer/Footer.ascx" %>
<%@ Register Tagprefix="Panel"
Namespace="MyApp2.PanelControls.AddAssetsRecord" Assembly="MyApp2" %>
<%@ Register Tagprefix="Custom" Namespace="MyApp2" Assembly="MyApp2"
%>
<%@ Page Language="cs" AutoEventWireup="false"
Codebehind="AddAssetsRecord.gen.aspx.cs" Inherits="MyApp2.AddAssetsRecord"
%>
<%@ Register Tagprefix="MyApp2" TagName="Header" Src="../Header &
Footer/Header.ascx" %>

<HTML id="Html1" runat="server">
<HEAD id="Head1" runat="server">
   <title>AddAssetsRecord</title>
   <meta name='GENERATOR' content='Microsoft Visual Studio.NET 7.0'>
   <meta name='CODE_LANGUAGE' content='Visual Basic 7.0'>
   <meta name='vs_defaultClientScript' content='JavaScript'>
   <meta name='vs_targetSchema'
content='http://schemas.microsoft.com/intellisense/ie5'>
  </HEAD>
  <body id="Body1" runat="server" bottomMargin='0' leftMargin='0' topMargin='0'
rightMargin='0'>
   <form id='Form1' method='post' runat='server'>
<BaseClasses:BasePageSettings id="PageSettings" runat="server"
></BaseClasses:BasePageSettings>
<link rel="stylesheet" rev="stylesheet" type="text/css" href="../Styles/Style.css">
<table cellspacing="0" cellpadding="0" border="0" class="pageBackground">
```

```
<tr>
<td class="pageAlignment">
<table cellspacing="0" cellpadding="0" border="0" class="borderTable">
     <tr>
   <td class="pageBorderTL"><img src="../Images/space.gif" height="1" width="1"
alt=""></td>
      <td class="pageBorderT"><img src="../Images/space.gif" height="1" width="1"
alt=""></td>
      <td class="pageBorderTR"><img src="../Images/space.gif" height="1"
width="1" alt=""></td>
     </tr>
     <tr>
      <td class="pageBorderL"><img src="../Images/space.gif" height="1" width="1"
alt=""></td>
      <td class="pageBorderC">
         <table cellspacing="0" cellpadding="0" border="0" class="master_table">
          <tr>
           <td>
            <table cellspacing="0" cellpadding="0" border="0" width="100%">
                    <tr>
                     <td>
                           <MyApp2:Header runat="server" id="V_Header">
</MyApp2:Header>
                     </td>
                    </tr>
            </table>
            <table cellspacing="0" cellpadding="0" border="0" class="master_table">
                    <tr>
                     <td>
                     </td>
                     <td>
                       <MyApp2:Menu runat="server" id="V_Menu">
</MyApp2:Menu>
                     </td>
                    </tr>
                    <tr>
           <td>
                  </td>
```

```
                        <td>
                    <table cellspacing="0" cellpadding="0" border="0"
class="master_table">
                        <tr>
                        <td class="page_yellow">
                                    <Panel:CustomRecordControl runat="server"
id="V_AssetsRecord" TableName="MyApp2.AssetsAccess, MyApp2"
VirtualId="Assets">

<!-- Begin Record Panel.html -->

<%=
SystemUtils.GenerateEnterKeyCaptureBeginTag(FindControl("V_CancelButton:V_B
utton")) %>
<%=
SystemUtils.GenerateEnterKeyCaptureBeginTag(FindControl("V_OKButton:V_Butt
on")) %>
<%=
SystemUtils.GenerateEnterKeyCaptureBeginTag(FindControl("V_SaveButton:V_But
ton")) %>
 <table class="dialog_view" cellpadding="0" cellspacing="0" border="0">
  <tr>
   <td class="dialog_header">
   <table cellpadding="0" cellspacing="0" width="100%" border="0">
    <tr>
     <td class="dialogHeaderEdgeL"><img src="../Images/space.gif"></td>
     <td class="dialogHeaderIcon"><BaseClasses:Image
ImageUrl="../Images/icon_edit.gif" runat="server" id="V_AssetsDialogIcon">

</BaseClasses:Image></td>
     <td class="dialog_header_text" valign="middle">
      <BaseClasses:Literal runat="server" id="V_AssetsDialogTitle" Text="Add
Assets" HtmlEncodeWhiteSpace="True">
</BaseClasses:Literal>
</td>
     <td class="dialogHeaderEdgeR"><img src="../Images/space.gif"></td>
    </tr>
   </table>
  </td>
 </tr>
 <tr>
  <td>
  <table cellpadding="0" cellspacing="0" border="0" width="100%">
   <tr>
    <td class="dialog_body">
      <table cellpadding="0" cellspacing="3" border="0">

       <tr>
        <td class="field_label_on_side">
         <BaseClasses:Literal runat="server" id="V_AssetIDLabel" Text="Asset ID">
                        </BaseClasses:Literal>
        </td>
        <td class="dialog_field_value">
         <BaseClasses:FieldValueTextBox runat="server" id="V_AssetID"
Field="Assets_.AssetID" Columns="14" CssClass="field_input">

          </BaseClasses:FieldValueTextBox>&nbsp;<BaseClasses:RequiredFieldValid
ator runat="server" id="V_AssetIDRequiredFieldValidator"
ControlToValidate="V_AssetID" Enabled="True" ErrorMessage="AssetID is
required." Text="*">                        </BaseClasses:RequiredFieldValidator>
                        <BaseClasses:TextBoxMaxLengthValidator runat="server"
id="V_AssetIDTextBoxMaxLengthValidator" Visible="False"
ControlToValidate="V_AssetID" ErrorMessage="The Asset ID value is too long.">
                        </BaseClasses:TextBoxMaxLengthValidator>
                        <BaseClasses:FieldValueValidator runat="server"
id="V_AssetIDFieldValueValidator" ControlToValidate="V_AssetID"
ErrorMessage="The Asset ID value you entered is invalid.  It may be invalid for a
number of reasons.  You may have entered a value that is longer than what is allowed,
the value may contain invalid characters or the value might be out of range of allowed
values.  Please review the value and correct it and submit the changes again.">
                  </BaseClasses:FieldValueValidator>
                        <BaseClasses:RecordControlCustomValidator runat="server"
id="V_AssetIDRecordControlCustomValidator" ControlToValidate="V_AssetID"
Enabled="False" Visible="False" ErrorMessage="The Asset ID value you entered is
invalid.  It may be invalid for a number of reasons.  You may have entered a value
that is longer than what is allowed, the value may contain invalid characters or the
```

```
value might be out of range of allowed values.  Please review the value and correct it
and submit the changes again.">
          </BaseClasses:RecordControlCustomValidator>
                              <BaseClasses:CustomValidator runat="server"
id="V_AssetIDCustomValidator" ControlToValidate="V_AssetID" Enabled="False"
Visible="False" ErrorMessage="The Asset ID value you entered is invalid.  It may be
invalid for a number of reasons.  You may have entered a value that is longer than
what is allowed, the value may contain invalid characters or the value might be out of
range of allowed values.  Please review the value and correct it and submit the
changes again.">                    </BaseClasses:CustomValidator>
        </td>
       </tr>
       <tr>
        <td class="field_label_on_side">
         <BaseClasses:Literal runat="server" id="V_AssetDescriptionLabel"
Text="Asset Description">
        </BaseClasses:Literal>
        </td>
        <td class="dialog_field_value">
         <BaseClasses:FieldValueTextBox runat="server" id="V_AssetDescription"
Field="Assets_.AssetDescription" Columns="60" CssClass="field_input" Rows="5"
TextMode="MultiLine">
                              </BaseClasses:FieldValueTextBox>&nbsp;
                              <BaseClasses:RequiredFieldValidator runat="server"
id="V_AssetDescriptionRequiredFieldValidator"
ControlToValidate="V_AssetDescription" ErrorMessage="A value for Asset
Description is required." Enabled="False">
          </BaseClasses:RequiredFieldValidator>
                              <BaseClasses:TextBoxMaxLengthValidator runat="server"
id="V_AssetDescriptionTextBoxMaxLengthValidator"
ControlToValidate="V_AssetDescription" ErrorMessage="The Asset Description
value is too long.">                    </BaseClasses:TextBoxMaxLengthValidator>
                              <BaseClasses:FieldValueValidator runat="server"
id="V_AssetDescriptionFieldValueValidator"
ControlToValidate="V_AssetDescription" ErrorMessage="The Asset Description
value you entered is invalid.  It may be invalid for a number of reasons.  You may
have entered a value that is longer than what is allowed, the value may contain invalid
characters or the value might be out of range of allowed values.  Please review the
```

```
value and correct it and submit the changes again.">
        </BaseClasses:FieldValueValidator>
                        <BaseClasses:RecordControlCustomValidator runat="server"
id="V_AssetDescriptionRecordControlCustomValidator"
ControlToValidate="V_AssetDescription" Enabled="False" Visible="False"
ErrorMessage="The Asset Description value you entered is invalid.  It may be invalid
for a number of reasons.  You may have entered a value that is longer than what is
allowed, the value may contain invalid characters or the value might be out of range
of allowed values.  Please review the value and correct it and submit the changes
again.">                        </BaseClasses:RecordControlCustomValidator>
                        <BaseClasses:CustomValidator runat="server"
id="V_AssetDescriptionCustomValidator" ControlToValidate="V_AssetDescription"
Enabled="False" Visible="False" ErrorMessage="The Asset Description value you
entered is invalid.  It may be invalid for a number of reasons.  You may have entered
a value that is longer than what is allowed, the value may contain invalid characters or
the value might be out of range of allowed values.  Please review the value and
correct it and submit the changes again.">
        </BaseClasses:CustomValidator>
      </td>
     </tr>
</table>
    <BaseClasses:ValidationSummary id="ValidationSummary1"
runat="server"></BaseClasses:ValidationSummary>
  </form>
 </body>
</HTML>
```

Program # 3

Name: Asset.xml (XML Code for generating Assets Table)

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
 - <ComponentDefinition xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 - <Self>
     <Name>AddAsset_CategoriesRecord</Name>
     <Type>Page</Type>
     <GenerateDirectory>Asset Categories</GenerateDirectory>
     <Layout>Asset Categories\AddAsset_CategoriesRecord.html</Layout>
     <DisplayName>AddAsset_CategoriesRecord</DisplayName>
     <Description>Standard record panel.</Description>
     <CheckSum>659939543-5608</CheckSum>
   </Self>
 - <ComposedOf>
   - <ContainedComponent>
       <Name>Asset_CategoriesDialogIcon</Name>
       <Type>Image</Type>
       <ImageUrl>../Images/icon_edit.gif</ImageUrl>
     </ContainedComponent>
   - <ContainedComponent>
       <Name>Asset_CategoriesDialogTitle</Name>
       <Type>Html</Type>
     - <PassthroughAttributes>
       - <Attribute>
           <Name>HtmlEncodeWhiteSpace</Name>
           <Value>True</Value>
         </Attribute>
       - <Attribute>
           <Name>Text</Name>
           <Value>Add Asset Categories</Value>
         </Attribute>
       </PassthroughAttributes>
     </ContainedComponent>
   - <ContainedComponent>
       <Name>Asset_CategoriesRecord</Name>
       <Type>Record</Type>
      <Table>Asset Categories</Table>
    - <PassthroughAttributes>
      - <Attribute>
          <Name>InternalUse:DataSource</Name>
          <Value>AddRecord</Value>
        </Attribute>
      - <Attribute>
          <Name>VirtualId</Name>
          <Value>0</Value>
        </Attribute>
      </PassthroughAttributes>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>AssetCategory</Name>
      <Type>FieldValue</Type>
      <Field>AssetCategory</Field>
      <FieldValueStyle>TextBox</FieldValueStyle>
      <Table>Asset Categories</Table>
    - <PassthroughAttributes>
      - <Attribute>
          <Name>CssClass</Name>
          <Value>field_input</Value>
        </Attribute>
      - <Attribute>
          <Name>InternalUse:DataSource</Name>
          <Value>Parent</Value>
        </Attribute>
      - <Attribute>

              <Name>V_AssetCategoryRequiredFieldValidator:
              Enabled</Name>
          <Value>False</Value>
        </Attribute>
      </PassthroughAttributes>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>AssetCategoryID</Name>
      <Type>FieldValue</Type>
    <Field>**AssetCategoryID**</Field>
    <FieldValueStyle>**TextBox**</FieldValueStyle>
    <Table>**Asset Categories**</Table>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>**CssClass**</Name>
        <Value>**field_input**</Value>
      </Attribute>
    - <Attribute>
        <Name>**InternalUse:DataSource**</Name>
        <Value>**Parent**</Value>
      </Attribute>
    - <Attribute>

            <Name>**V_AssetCategoryIDRequiredFieldValidato
            r:Enabled**</Name>
        <Value>**True**</Value>
      </Attribute>
    - <Attribute>

            <Name>**V_AssetCategoryIDRequiredFieldValidato
            r:ErrorMessage**</Name>
        <Value>**AssetCategoryID is required.**</Value>
      </Attribute>
    - <Attribute>

            <Name>**V_AssetCategoryIDRequiredFieldValidato
            r:Text**</Name>
        <Value>*</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>**AssetCategoryIDLabel**</Name>
    <Type>**Literal**</Type>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>**InternalUse:Field**</Name>
```

```
            <Value>AssetCategoryID</Value>
          </Attribute>
        - <Attribute>
            <Name>InternalUse:Table</Name>
            <Value>Asset Categories</Value>
          </Attribute>
        - <Attribute>
            <Name>Text</Name>
            <Value>%ISD_DEFAULT%Asset
                Categories%AssetCategoryID</Value>
          </Attribute>
        </PassthroughAttributes>
      </ContainedComponent>
    - <ContainedComponent>
        <Name>AssetCategoryLabel</Name>
        <Type>Literal</Type>
      - <PassthroughAttributes>
        - <Attribute>
            <Name>InternalUse:Field</Name>
            <Value>AssetCategory</Value>
          </Attribute>
        - <Attribute>
            <Name>InternalUse:Table</Name>
            <Value>Asset Categories</Value>
          </Attribute>
        - <Attribute>
            <Name>Text</Name>
            <Value>%ISD_DEFAULT%Asset
                Categories%AssetCategory</Value>
          </Attribute>
        </PassthroughAttributes>
      </ContainedComponent>
    - <ContainedComponent>
        <Name>CancelButton</Name>
        <Type>Button</Type>
        <Layout>..\Shared\Button.html</Layout>
      - <PassthroughAttributes>
        - <Attribute>
```

```
            <Name>P_Button-CausesValidation</Name>
            <Value>False</Value>
          </Attribute>
        - <Attribute>
            <Name>P_Button-RedirectURL</Name>
            <Value>Back</Value>
          </Attribute>
        - <Attribute>
            <Name>P_Button-Text</Name>
            <Value>Btn:Cancel</Value>
            <Type>ResourceReference</Type>
          </Attribute>
      </PassthroughAttributes>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>Footer</Name>
      <Type>Footer</Type>
      <Layout>..\Header & Footer\Footer.html</Layout>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>Header</Name>
      <Type>Header</Type>
      <Layout>..\Header & Footer\Header.html</Layout>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>Menu</Name>
      <Type>Menu</Type>
      <Layout>..\Menu Panels\Menu.html</Layout>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>SaveButton</Name>
      <Type>Button</Type>
      <Layout>..\Shared\Button.html</Layout>
    - <PassthroughAttributes>
        - <Attribute>
            <Name>P_Button-CausesValidation</Name>
            <Value>True</Value>
          </Attribute>
        - <Attribute>
              <Name>P_Button-CommandName</Name>
              <Value>UpdateData</Value>
          </Attribute>
        - <Attribute>
              <Name>P_Button-Consumers</Name>
              <Value>page</Value>
          </Attribute>
        - <Attribute>
              <Name>P_Button-RedirectArgument</Name>
              <Value>updatedata</Value>
          </Attribute>
        - <Attribute>
              <Name>P_Button-RedirectURL</Name>
              <Value>Back</Value>
          </Attribute>
        - <Attribute>
              <Name>P_Button-Text</Name>
              <Value>Btn:Save</Value>
              <Type>ResourceReference</Type>
          </Attribute>
        </PassthroughAttributes>
      </ContainedComponent>
    </ComposedOf>
</ComponentDefinition>
```

Program # 4

Name: Departments.xml (XML Data Access for the Departments)

```
 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- - <ColumnDefinition>
    - <Column InternalName="0" Priority="1" ColumnNum="0">
        <columnName>DepartmentID</columnName>
        <columnUIName>%ISD_DEFAULT%</columnUIName>
        <columnType>Number</columnType>
        <columnDBType>int</columnDBType>
        <columnLengthSet>10.0</columnLengthSet>
        <columnDefault />
        <columnDBDefault />
        <columnIndex>N</columnIndex>
        <columnUnique>N</columnUnique>
        <columnFunction />
        <columnDBFormat />
        <columnPK>N</columnPK>
        <columnPermanent>N</columnPermanent>
        <columnComputed>N</columnComputed>
        <columnIdentity>N</columnIdentity>
        <columnReadOnly>N</columnReadOnly>
        <columnRequired>Y</columnRequired>
        <columnNotNull>Y</columnNotNull>

            <columnVisibleWidth>%ISD_DEFAULT%</columnVisibleW
            idth>
        <columnTableAliasName />
      </Column>
    - <Column InternalName="1" Priority="2" ColumnNum="1">
        <columnName>DepartmentName</columnName>
        <columnUIName>%ISD_DEFAULT%</columnUIName>
        <columnType>String</columnType>
        <columnDBType>nvarchar</columnDBType>
        <columnLengthSet>50</columnLengthSet>
        <columnDefault />
        <columnDBDefault />
    <columnIndex>N</columnIndex>
    <columnUnique>N</columnUnique>
    <columnFunction />
    <columnDBFormat />
    <columnPK>N</columnPK>
    <columnPermanent>N</columnPermanent>
    <columnComputed>N</columnComputed>
    <columnIdentity>N</columnIdentity>
    <columnReadOnly>N</columnReadOnly>
    <columnRequired>N</columnRequired>
    <columnNotNull>N</columnNotNull>

   <columnVisibleWidth>%ISD_DEFAULT%</columnVisibleWidth
    <columnTableAliasName />
  </Column>
- <Column InternalName="2" Priority="3" ColumnNum="2">
    <columnName>DepartmentNumber</columnName>
    <columnUIName>%ISD_DEFAULT%</columnUIName>
    <columnType>Number</columnType>
    <columnDBType>int</columnDBType>
    <columnLengthSet>10.0</columnLengthSet>
    <columnDefault />
    <columnDBDefault />
    <columnIndex>N</columnIndex>
    <columnUnique>N</columnUnique>
    <columnFunction />
    <columnDBFormat />
    <columnPK>N</columnPK>
    <columnPermanent>N</columnPermanent>
    <columnComputed>N</columnComputed>
    <columnIdentity>N</columnIdentity>
    <columnReadOnly>N</columnReadOnly>
    <columnRequired>N</columnRequired>
    <columnNotNull>N</columnNotNull>

        <columnVisibleWidth>%ISD_DEFAULT%</columnVisibleW
        idth>
    <columnTableAliasName />
```

```
        </Column>
    </ColumnDefinition>
    <TableName>Departments</TableName>
    <Version>1</Version>
    <Owner />
    <TableCodeName>Departments</TableCodeName>
    <TableAliasName>Departments_</TableAliasName>
    <ConnectionName>DatabaseAsset1</ConnectionName>
    <canCreateRecords Source="Database">Y</canCreateRecords>
    <canEditRecords Source="Database">N</canEditRecords>
    <canDeleteRecords Source="Database">N</canDeleteRecords>
    <canViewRecords Source="Database">N</canViewRecords>
    <AppShortName>MyApp2</AppShortName>
    <TableStoredProcPrefix>pMyApp2Departments</TableStoredProcPrefix>
</XMLDefinition>
```

Program # 5

Name: Departments.cs (C# code for adding of Departments Record)

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;

namespace MyApp2
{
public class AddDepartmentsRecord : AddDepartmentsRecordGen
{
#region Overview: The life cycle of a Page and its Controls
   // The Life Cycle of a Page and its Controls
   //
   // In ASP.NET, when a page is executed, it goes through a number of
   // events.  You can customize the behavior of the page by handling
   // each one of these events.  Here is the basic order of events:
   //
   // 1. Init event - Initialize settings needed during the lifetime
   // of the incoming web request. We recommend you use the Load
   // event preferably.
   //
   // 2. Load - Perform actions common to all requests. This is the
   // usual event that developers can use to do their processing.
   //
   // 3. PreRender - Perform any updates before the output is rendered.
   // This is another common event that developers can use to do their
   // processing. This event is the last one in the sequence of events
   // prior to the page being displayed to the end user.
   //
   // There are some additional events such as SaveViewState, Render,
   // Dispose and Unload that occur during the lifecycle of the page.
#endregion
} // End class AddDepartmentsRecord

} // End namespace MyApp2

namespace MyApp2.PanelControls.AddDepartmentsRecord
{
[System.Web.UI.ToolboxDataAttribute("<{0}:CustomTableControl
runat=server></{0}:CustomTableControl>")]

[System.ComponentModel.DesignerAttribute("System.Web.UI.Design.ReadWriteControlDes
igner, System.Design")]
   [System.Web.UI.PersistChildrenAttribute(true)]
 [System.Web.UI.ToolboxDataAttribute("<{0}:CustomRecordControl
runat=server></{0}:CustomRecordControl>")]

[System.ComponentModel.DesignerAttribute("System.Web.UI.Design.ReadWriteControlDes
igner, System.Design")]
   [System.Web.UI.PersistChildrenAttribute(true)]
   public class CustomRecordControl
   {
      public CustomRecordControl()
      {


      }
} // End class CustomRecordControl

} // End namespace MyApp2.PanelControls.AddDepartmentsRecord
```

Program # 6

Purpose: Depreciation.xml (XML generated for showing the Depreciation Table)

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
 - <ComponentDefinition xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
    <Name>ShowDepreciationTable</Name>
    <Type>Page</Type>
    <GenerateDirectory>Depreciation</GenerateDirectory>
    <Layout>Depreciation\ShowDepreciationTable.html</Layout>
    <DisplayName>ShowDepreciationTable</DisplayName>
    <Description>Table view to show or edit multiple
       records.</Description>
    <CheckSum>-1530362054-9813</CheckSum>
  - <EventBindings>
    - <Event>
        <Producer>DepreciationExportButton.V_Button</Producer>
        <Consumer>DepreciationTable</Consumer>
      </Event>
    - <Event>
        <Producer>DepreciationIDFilter</Producer>
        <Consumer>DepreciationTable</Consumer>
      </Event>
    - <Event>
        <Producer>DepreciationTable</Producer>
        <Consumer>DepreciationIDFilter</Consumer>
      </Event>
    - <Event>
        <Producer>AssetIDFilter</Producer>
        <Consumer>DepreciationTable</Consumer>
      </Event>
    - <Event>
        <Producer>DepreciationTable</Producer>
        <Consumer>AssetIDFilter</Consumer>
      </Event>
    - <Event>
    <Producer>DepreciationPagination.V_PageSize</Producer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>

       <Producer>DepreciationPagination.V_CurrentPage</Pro
       ducer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>

       <Producer>DepreciationPagination.V_TotalPages</Prod
       ucer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>
    <Producer>DepreciationPagination.V_FirstPage</Producer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>

       <Producer>DepreciationPagination.V_PreviousPage</Pr
       oducer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>
    <Producer>DepreciationPagination.V_NextPage</Producer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>
    <Producer>DepreciationPagination.V_LastPage</Producer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>
    <Producer>DepreciationPagination.V_FirstItem</Producer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>
```

```
    <Producer>DepreciationPagination.V_LastItem</Producer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>
       <Producer>DepreciationPagination.V_TotalItems</Prod
       ucer>
    <Consumer>DepreciationTable</Consumer>
  </Event>
- <Event>
    <Producer>DepreciationTable</Producer>
       <Consumer>DepreciationPagination.V_PageSize</Consu
       mer>
  </Event>
- <Event>
    <Producer>DepreciationTable</Producer>
       <Consumer>DepreciationPagination.V_CurrentPage</C
       onsumer>
  </Event>
- <Event>
    <Producer>DepreciationTable</Producer>
       <Consumer>DepreciationPagination.V_TotalPages</Con
       sumer>
  </Event>
- <Event>
    <Producer>DepreciationTable</Producer>
       <Consumer>DepreciationPagination.V_FirstPage</Cons
       umer>
  </Event>
- <Event>
    <Producer>DepreciationTable</Producer>
       <Consumer>DepreciationPagination.V_PreviousPage</C
       onsumer>
```

```
      </Event>
    - <Event>
        <Producer>DepreciationTable</Producer>

            <Consumer>DepreciationPagination.V_NextPage</Cons
            umer>
      </Event>
    - <Event>
        <Producer>DepreciationTable</Producer>

            <Consumer>DepreciationPagination.V_LastPage</Cons
            umer>
      </Event>
    - <Event>
        <Producer>DepreciationTable</Producer>

            <Consumer>DepreciationPagination.V_FirstItem</Cons
            umer>
      </Event>
    - <Event>
        <Producer>DepreciationTable</Producer>

            <Consumer>DepreciationPagination.V_LastItem</Consu
            mer>
      </Event>
    - <Event>
        <Producer>DepreciationTable</Producer>

            <Consumer>DepreciationPagination.V_TotalItems</Con
            sumer>
      </Event>
    </EventBindings>
  </Self>
- <ComposedOf>
  - <ContainedComponent>
      <Name>AssetID</Name>
      <Type>FieldValue</Type>
      <Field>AssetID</Field>
    <FieldValueStyle>Literal</FieldValueStyle>
    <Table>Depreciation</Table>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>InternalUse:DataSource</Name>
        <Value>Parent</Value>
      </Attribute>
    - <Attribute>
        <Name>NullValueText</Name>
        <Value>&nbsp;</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>AssetIDFilter</Name>
    <Type>FieldFilter</Type>
    <FieldValueStyle>DropDownList</FieldValueStyle>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>Field</Name>
        <Value>AssetID</Value>
      </Attribute>
    - <Attribute>
        <Name>InternalUse:Table</Name>
        <Value>Depreciation</Value>
      </Attribute>
    - <Attribute>
        <Name>Operator</Name>
        <Value>=</Value>
      </Attribute>
    - <Attribute>

          <Name>V_AssetIDFilterDropDownList:CssClass</
          Name>
        <Value>Filter_Input</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
```

```
- <ContainedComponent>
    <Name>AssetIDLabel</Name>
    <Type>Literal</Type>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>InternalUse:Field</Name>
        <Value>AssetID</Value>
      </Attribute>
    - <Attribute>
        <Name>InternalUse:Table</Name>
        <Value>Depreciation</Value>
      </Attribute>
    - <Attribute>
        <Name>Text</Name>

           <Value>%ISD_DEFAULT%Depreciation%AssetI
           D</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>AssetIDLabel1</Name>
    <Type>FieldSort</Type>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>Field</Name>
        <Value>AssetID</Value>
      </Attribute>
    - <Attribute>
        <Name>Table</Name>
        <Value>Depreciation</Value>
      </Attribute>
    - <Attribute>
        <Name>Text</Name>

           <Value>%ISD_DEFAULT%Depreciation%AssetI
           D</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>DepreciationAmount</Name>
    <Type>FieldValue</Type>
    <Field>DepreciationAmount</Field>
    <FieldValueStyle>Literal</FieldValueStyle>
    <Table>Depreciation</Table>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>InternalUse:DataSource</Name>
        <Value>Parent</Value>
      </Attribute>
    - <Attribute>
        <Name>NullValueText</Name>
        <Value>&nbsp;</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>DepreciationAmountLabel</Name>
    <Type>FieldSort</Type>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>Field</Name>
        <Value>DepreciationAmount</Value>
      </Attribute>
    - <Attribute>
        <Name>Table</Name>
        <Value>Depreciation</Value>
      </Attribute>
    - <Attribute>
        <Name>Text</Name>

           <Value>%ISD_DEFAULT%Depreciation%Depre
           ciationAmount</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>DepreciationDate</Name>
    <Type>FieldValue</Type>
    <Field>DepreciationDate</Field>
    <FieldValueStyle>Literal</FieldValueStyle>
    <Table>Depreciation</Table>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>InternalUse:DataSource</Name>
        <Value>Parent</Value>
      </Attribute>
    - <Attribute>
        <Name>NullValueText</Name>
        <Value>&nbsp;</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>DepreciationDateLabel</Name>
    <Type>FieldSort</Type>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>Field</Name>
        <Value>DepreciationDate</Value>
      </Attribute>
    - <Attribute>
        <Name>Table</Name>
        <Value>Depreciation</Value>
      </Attribute>
    - <Attribute>
        <Name>Text</Name>

            <Value>%ISD_DEFAULT%Depreciation%Depre
            ciationDate</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
```

```
- <ContainedComponent>
    <Name>DepreciationExportButton</Name>
    <Type>Button</Type>
    <Layout>..\Shared\Button.html</Layout>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>P_Button-CausesValidation</Name>
        <Value>False</Value>
      </Attribute>
    - <Attribute>
        <Name>P_Button-CommandName</Name>
        <Value>ExportData</Value>
      </Attribute>
    - <Attribute>
        <Name>P_Button-Text</Name>
        <Value>Btn:Export</Value>
        <Type>ResourceReference</Type>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>DepreciationID</Name>
    <Type>FieldValue</Type>
    <Field>DepreciationID</Field>
    <FieldValueStyle>Literal</FieldValueStyle>
    <Table>Depreciation</Table>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>InternalUse:DataSource</Name>
        <Value>Parent</Value>
      </Attribute>
    - <Attribute>
        <Name>NullValueText</Name>
        <Value>&nbsp;</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
```

```
    <Name>DepreciationIDFilter</Name>
    <Type>FieldFilter</Type>
    <FieldValueStyle>DropDownList</FieldValueStyle>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>Field</Name>
        <Value>DepreciationID</Value>
      </Attribute>
    - <Attribute>
        <Name>InternalUse:Table</Name>
        <Value>Depreciation</Value>
      </Attribute>
    - <Attribute>
        <Name>Operator</Name>
        <Value>=</Value>
      </Attribute>
    - <Attribute>

          <Name>V_DepreciationIDFilterDropDownList:Cs
          sClass</Name>
        <Value>Filter_Input</Value>
      </Attribute>
    </PassthroughAttributes>
  </ContainedComponent>
- <ContainedComponent>
    <Name>DepreciationIDLabel</Name>
    <Type>Literal</Type>
  - <PassthroughAttributes>
    - <Attribute>
        <Name>InternalUse:Field</Name>
        <Value>DepreciationID</Value>
      </Attribute>
    - <Attribute>
        <Name>InternalUse:Table</Name>
        <Value>Depreciation</Value>
      </Attribute>
    - <Attribute>
        <Name>Text</Name>
```

```
              <Value>%ISD_DEFAULT%Depreciation%Depre
              ciationID</Value>
          </Attribute>
        </PassthroughAttributes>
      </ContainedComponent>
    - <ContainedComponent>
        <Name>DepreciationIDLabel1</Name>
        <Type>FieldSort</Type>
      - <PassthroughAttributes>
        - <Attribute>
            <Name>Field</Name>
            <Value>DepreciationID</Value>
          </Attribute>
        - <Attribute>
            <Name>Table</Name>
            <Value>Depreciation</Value>
          </Attribute>
        - <Attribute>
            <Name>Text</Name>

              <Value>%ISD_DEFAULT%Depreciation%Depre
              ciationID</Value>
          </Attribute>
        </PassthroughAttributes>
      </ContainedComponent>
    - <ContainedComponent>
        <Name>DepreciationNewButton</Name>
        <Type>Button</Type>
        <Layout>..\Shared\Button.html</Layout>
      - <PassthroughAttributes>
        - <Attribute>
            <Name>P_Button-CausesValidation</Name>
            <Value>False</Value>
          </Attribute>
        - <Attribute>
            <Name>P_Button-CommandName</Name>
            <Value>Redirect</Value>
        </Attribute>
      - <Attribute>
          <Name>P_Button-RedirectArgument</Name>
          <Value>ID</Value>
        </Attribute>
      - <Attribute>
          <Name>P_Button-RedirectURL</Name>

              <Value>../Depreciation/AddDepreciationRecord.as
              px</Value>
        </Attribute>
      - <Attribute>
          <Name>P_Button-Text</Name>
          <Value>Btn:New</Value>
          <Type>ResourceReference</Type>
        </Attribute>
      </PassthroughAttributes>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>DepreciationPagination</Name>
      <Type>Pagination</Type>
      <Layout>..\Shared\Pagination.html</Layout>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>DepreciationTable</Name>
      <Type>Table</Type>
      <Table>Depreciation</Table>
    - <EventBindings>
      - <Event>
          <Producer>DepreciationIDLabel1</Producer>
          <Consumer>DepreciationTable</Consumer>
        </Event>
      - <Event>
          <Producer>DepreciationTable</Producer>
          <Consumer>DepreciationIDLabel1</Consumer>
        </Event>
      - <Event>
          <Producer>AssetIDLabel1</Producer>
```

```
            <Consumer>**DepreciationTable**</Consumer>
         </Event>
       - <Event>
            <Producer>**DepreciationTable**</Producer>
            <Consumer>**AssetIDLabel1**</Consumer>
         </Event>
       - <Event>
            <Producer>**DepreciationDateLabel**</Producer>
            <Consumer>**DepreciationTable**</Consumer>
         </Event>
       - <Event>
            <Producer>**DepreciationTable**</Producer>
            <Consumer>**DepreciationDateLabel**</Consumer>
         </Event>
       - <Event>
            <Producer>**DepreciationAmountLabel**</Producer>
            <Consumer>**DepreciationTable**</Consumer>
         </Event>
       - <Event>
            <Producer>**DepreciationTable**</Producer>
            <Consumer>**DepreciationAmountLabel**</Consumer>
         </Event>
      </EventBindings>
    - <PassthroughAttributes>
       - <Attribute>
            <Name>**InternalUse:DataSource**</Name>
            <Value>**TableOrQuery**</Value>
         </Attribute>
       - <Attribute>
            <Name>**PageSize**</Name>
            <Value>**10**</Value>
         </Attribute>
      </PassthroughAttributes>
   </ContainedComponent>
 - <ContainedComponent>
      <Name>**DepreciationTableIcon**</Name>
      <Type>**Image**</Type>
      <ImageUrl>**../Images/icon_view.gif**</ImageUrl>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>DepreciationTableTitle</Name>
      <Type>Html</Type>
    - <PassthroughAttributes>
      - <Attribute>
          <Name>HtmlEncodeWhiteSpace</Name>
          <Value>True</Value>
        </Attribute>
      - <Attribute>
          <Name>Text</Name>
          <Value>Depreciation</Value>
        </Attribute>
      </PassthroughAttributes>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>Footer</Name>
      <Type>Footer</Type>
      <Layout>..\Header & Footer\Footer.html</Layout>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>Header</Name>
      <Type>Header</Type>
      <Layout>..\Header & Footer\Header.html</Layout>
    </ContainedComponent>
  - <ContainedComponent>
      <Name>Menu</Name>
      <Type>Menu</Type>
      <Layout>..\Menu Panels\Menu.html</Layout>
    - <PassthroughAttributes>
      - <Attribute>
          <Name>HiliteSettings</Name>
          <Value>Menu4MenuItem</Value>
        </Attribute>
      </PassthroughAttributes>
    </ContainedComponent>
  </ComposedOf>
</ComponentDefinition>
```

Program # 7
Name: Header.ascx

```
<%@ Register Tagprefix="Panel" Namespace="MyApp2.PanelControls.Header"
Assembly="MyApp2" %>

<%@ Control Language="cs" TargetSchema='http://schemas.microsoft.com/intellisense/ie5'
AutoEventWireup="false" CodeBehind="Header.gen.ascx.cs" Inherits="MyApp2.Header"
%>
<%@ Register Tagprefix="Custom" Namespace="MyApp2" Assembly="MyApp2" %>
<table cellspacing="0" cellpadding="0" border="0" class="master_table">
 <tr>
 <td class="logoBG">
 <BaseClasses:Image ImageUrl="../Images/Logo.gif" runat="server" id="V_Logo">
</BaseClasses:Image>
 </td>
 </tr>
</table>
```

Program # 8
Name: Header.xml

```
 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <ComponentDefinition xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >

    <Version>1.4</Version>
  - <Self>
      <Name>Header</Name>
      <Type>Header</Type>
      <GenerateDirectory>Header & Footer</GenerateDirectory>
      <Layout>Header & Footer\Header.html</Layout>
      <DisplayName>Header</DisplayName>
      <Description>(Description)</Description>
    </Self>
  - <ComposedOf>
    - <ContainedComponent>
```

```
            <Name>Logo</Name>
            <Type>Image</Type>
            <ImageUrl>../Images/Logo.gif</ImageUrl>
        </ContainedComponent>
    </ComposedOf>
</ComponentDefinition>
```

Stored Procedure # 1

Purpose: To add department record into the Departments Table

```
if exists (select * from sysobjects where id = object_id('pMyApp2DepartmentsAdd')
and sysstat & 0xf = 4) drop procedure pMyApp2DepartmentsAdd
GO

CREATE PROCEDURE pMyApp2DepartmentsAdd
   @p_DepartmentID int,
   @p_DepartmentName nvarchar(50),
   @p_DepartmentNumber int
AS
BEGIN
   INSERT
   INTO [Departments]
      (
         [DepartmentID],
         [DepartmentName],
         [DepartmentNumber]
      )
   VALUES
      (
         @p_DepartmentID,
         @p_DepartmentName,
         @p_DepartmentNumber
      )

END
```

Stored Procedure # 2

Purpose: Exporting the Employees data to a CSV file

```
if exists (select * from sysobjects where id = object_id('pMyApp2EmployeesExport')
and sysstat & 0xf = 4) drop procedure pMyApp2EmployeesExport
GO

CREATE PROCEDURE pMyApp2EmployeesExport
      @p_title_str nvarchar(4000),
      @p_select_str nvarchar(4000),
      @p_join_str nvarchar(4000),
      @p_where_str nvarchar(4000),
      @p_num_exported int output
   AS
   DECLARE
      @l_title_str nvarchar(4000),
      @l_select_str nvarchar(4000),
      @l_from_str nvarchar(4000),
      @l_join_str nvarchar(4000),
      @l_where_str nvarchar(4000),
      @l_query_select nvarchar(4000),
      @l_query_union nvarchar(4000),
      @l_query_from nvarchar(4000)
   BEGIN
      -- Set up the title string from the column names.  Excel
      -- will complain if the first column value is ID. So wrap
      -- the value with "".
      SET @l_title_str = @p_title_str + char(13)
      IF @p_title_str IS NULL
         BEGIN
         SET @l_title_str =
            N'"EmployeeID"' + ',' +
            N'"SocialSecurityNumber"' + ',' +
            N'"EmployeeNumber"' + ',' +
            N'"FirstName"' + ',' +
            N'"LastName"' + ',' +
            N'"Title"' + ',' +
            N'"EmailName"' + ',' +
            N'"HomePhone"' + ',' +
            N'"WorkPhone"' + ',' +
            N'"OfficeLocation"' + ' ';
         END
      ELSE IF SUBSTRING(@l_title_str, 1, 2) = 'ID'
         SET @l_title_str =
            '"' +
            SUBSTRING(@l_title_str, 1, PATINDEX('%,%', @l_title_str)-1) +
            '"' +
            SUBSTRING(@l_title_str, PATINDEX('%,%', @l_title_str),
LEN(@l_title_str));

      -- Set up the select string
      SET @l_select_str = @p_select_str
      IF @p_select_str IS NULL
         BEGIN
         SET @l_select_str =
            N'IsNULL(Convert(nvarchar, Employees_.[EmployeeID]), '''') + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[SocialSecurityNumber], ''''),
N''"'', N''""'') + N''"''  + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[EmployeeNumber], ''''), N''"'',
N''""'') + N''"''  + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[FirstName], ''''), N''"'', N''""'') +
N''"''  + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[LastName], ''''), N''"'', N''""'') +
N''"''  + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[Title], ''''), N''"'', N''""'') + N''"''
+ '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[EmailName], ''''), N''"'', N''""'')
+ N''"''  + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[HomePhone], ''''), N''"'', N''""'')
+ N''"''  + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[WorkPhone], ''''), N''"'', N''""'')
+ N''"''  + '','' +' +
            N'N''"'' + REPLACE(IsNULL(Employees_.[OfficeLocation], ''''), N''"'',
N''""'') + N''"''  + '' '''';
         END
```

```
    -- Set up the from string (with table alias) and the join string
    SET @l_from_str = '[Employees] Employees_';

    SET @l_join_str = @p_join_str
    if @p_join_str is null
       SET @l_join_str = ' ';

    -- Set up the where string
    SET @l_where_str = ' ';
    IF @p_where_str IS NOT NULL
       SET @l_where_str = @l_where_str + 'WHERE ' + @p_where_str;

    -- Construct the query string.  Append the result set with the title.
    SET @l_query_select =
          'SELECT '''
    SET @l_query_union =
          ''' UNION ALL ' +
          'SELECT '
    SET @l_query_from =
          ' FROM ' + @l_from_str + ' ' + @l_join_str + ' ' +
          @l_where_str;

    -- Run the query
    EXECUTE (@l_query_select + @l_title_str + @l_query_union +
@l_select_str+ @l_query_from)

    -- Return the total number of rows of the query
    SELECT @p_num_exported = @@rowcount
  END
```

Stored Procedure # 3

Purpose: This stored procedure will return query result based on the passed in select, search and ORDER BY clauses.

```
-----------------------------------------------------------------------------------------------

if exists (select * from sysobjects where id = object_id('pMyApp2AssetsDrillDown')
and sysstat & 0xf = 4) drop procedure pMyApp2AssetsDrillDown
GO

CREATE PROCEDURE pMyApp2AssetsDrillDown
      @p_select_str nvarchar(4000),
      @p_is_distinct int,
      @p_select_str_b nvarchar(4000),
      @p_join_str nvarchar(4000),
      @p_where_str nvarchar(4000),
      @p_sort_str nvarchar(4000),
      @p_page_number int,
      @p_batch_size int
AS
DECLARE
   @l_createtemp_select nvarchar(4000),
   @l_createtemp_into nvarchar(4000),
   @l_createtemp_from nvarchar(4000),
   @l_createtemp_where nvarchar(4000),
   @l_temp_insert nvarchar(4000),
   @l_temp_select nvarchar(4000),
   @l_temp_from nvarchar(4000),
   @l_final_sort nvarchar(4000),
   @l_query_select nvarchar(4000),
   @l_query_from nvarchar(4000),
   @l_query_where nvarchar(4000),
   @l_from_str nvarchar(4000),
   @l_join_str nvarchar(4000),
   @l_sort_str nvarchar(4000),
   @l_where_str nvarchar(4000),
   @l_count_query nvarchar(4000),
   @l_end_gen_row_num integer,
   @l_start_gen_row_num integer
```

```
BEGIN
    -- Set up the from string as the base table.
    SET @l_from_str = '[Assets] Assets_'

    -- Set up the join string.
    SET @l_join_str = @p_join_str
    IF @p_join_str is null
        SET @l_join_str = ' '

    -- Set up the where string.
    SET @l_where_str = ' '
        IF @p_where_str is not null
        SET @l_where_str = 'WHERE ' + @p_where_str

    -- Get the total count of rows the query will return
    IF @p_page_number > 0 and @p_batch_size >= 0
    BEGIN
        IF @p_is_distinct = 0
        BEGIN
            SET @l_count_query =
                'SELECT count(*) FROM ( SELECT ' + @p_select_str +
                'FROM ' + @l_from_str + ' ' + @l_join_str + ' ' +
                @l_where_str + ' ) countAlias'

        END
        ELSE
        BEGIN
            SET @l_count_query =
                'SELECT COUNT(*) FROM ( SELECT DISTINCT ' + @p_select_str + ', 1
As __One  ' +
                'FROM ' + @l_from_str + ' ' + @l_join_str + ' ' +
                @l_where_str + ' ) pass1 '

        END
    END
    ELSE
    BEGIN
        SET @l_count_query = ' '
```

```
END

-- Get the list.
IF @p_page_number > 0 AND @p_batch_size > 0
BEGIN
   -- If the caller did not pass a sort string, use a default value
   IF @p_sort_str IS NULL OR LTRIM(RTRIM(@p_sort_str)) = ''
      SET @l_sort_str = 'ORDER BY 1 '

   ELSE
      SET @l_sort_str = 'ORDER BY ' + @p_sort_str
   -- Calculate the rows to be included in the list
   -- before geting the list.
   SET @l_end_gen_row_num = @p_page_number * @p_batch_size;
   SET @l_start_gen_row_num = @l_end_gen_row_num - (@p_batch_size-1);

   -- Create a temporary table to keep the numbering
   -- of the rows returned.  It contains the necessary colums
   -- from base table plus an identity field used for numbering

   SELECT 1 AS IS_SECONDARY_TEMP_T_GETLIST_COL INTO
#IS_SECONDARY_TEMP_T_GETLIST

   SET @l_createtemp_select =
      'SELECT Identity(int,1,1) AS IS_ROWNUM_COL, ' +
      @p_select_str
   SET @l_createtemp_into =
      ' INTO #IS_TEMP_T_GETLIST '
   SET @l_createtemp_from =
      ' FROM [Assets] AS [Assets_], #IS_SECONDARY_TEMP_T_GETLIST'
   SET @l_createtemp_where =
      ' WHERE 1=2 '
   -- Now copy column data into temporary table.
   SET @l_temp_insert =
      'INSERT INTO #IS_TEMP_T_GETLIST ( ' +
      @p_select_str +
      ' ) '
```

```
    IF @p_is_distinct = 0 OR @p_sort_str IS NULL OR
LTRIM(RTRIM(@p_sort_str)) = ''
    BEGIN
      IF @p_is_distinct = 0
      BEGIN
        SET @l_temp_select =
          'SELECT '
      END
      ELSE
      BEGIN
        SET @l_temp_select =
          'SELECT DISTINCT '
      END
    SET @l_temp_select =
      @l_temp_select +
      'TOP ' + convert(varchar, @l_end_gen_row_num) + ' ' +
      @p_select_str
    END
    ELSE
    BEGIN
        -- Need to construct query differently when sorting by expanded DFKA,
        -- So get the TOP DISTINCT values after selecting, joining, and sorting
ALL the values
        SET @l_temp_select =
          'SELECT __ReturnCol FROM ( ' +
          'SELECT ' +
          'DISTINCT ' +
          'TOP ' + convert(varchar, @l_end_gen_row_num) + ' ' +
          @p_select_str +
          ' As __ReturnCol, ' +
          @p_select_str_b
        -- Close and alias the outer FROM clause after the inner ORDER BY clause
        SET @l_sort_str =
          @l_sort_str +
          ' ) pass1 '
    END
    SET @l_temp_from =
      ' FROM ' + @l_from_str + ' ' + @l_join_str
```

```
        -- Construct the main query
        SET @l_query_select = 'SELECT '
        SET @l_query_from =
           'FROM #IS_TEMP_T_GETLIST ' +
           'AS [Assets_]' +
           ' WHERE IS_ROWNUM_COL >= '+ convert(varchar,
@l_start_gen_row_num)

        SET @l_final_sort = 'ORDER BY IS_ROWNUM_COL Asc '

        -- Run all the queries as a batch so the temp tables won't lose scope
        EXECUTE (@l_count_query + '     ' + @l_createtemp_select +
@l_createtemp_into + @l_createtemp_from + @l_createtemp_where + '     ' +
@l_temp_insert + @l_temp_select + @l_temp_from + ' ' + @l_where_str + ' ' +
@l_sort_str + '     ' + @l_query_select + @p_select_str + @l_query_from +
@l_query_where + @l_final_sort)

    END
    ELSE
    BEGIN
        -- If page number and batch size are not valid numbers
        -- return the empty result set
        SET @l_query_select = 'SELECT '
        SET @l_query_from =
           ' FROM [Assets] Assets_ ' +
           'WHERE 1=2;'
        EXECUTE (@l_query_select + @p_select_str + @l_query_from);
    END

END
```

Stored Procedure # 4

Purpose: Runs a SQL function against a column and returns the result back giving the current page number and batch size. SQL functions can include sum, avg, max, etc.

```
if exists (select * from sysobjects where id =
object_id('pMyApp2Asset_CategoriesGetStats') and sysstat & 0xf = 4) drop
procedure pMyApp2Asset_CategoriesGetStats
GO

CREATE PROCEDURE pMyApp2Asset_CategoriesGetStats
      @p_select_str nvarchar(4000),
      @p_join_str nvarchar(4000),
      @p_where_str nvarchar(4000),
      @p_sort_str nvarchar(4000),
      @p_page_number integer,
      @p_batch_size integer
   AS
   DECLARE
      @l_query nvarchar(4000),
      @l_from_str nvarchar(4000),
      @l_join_str nvarchar(4000),
      @l_sort_str nvarchar(4000),
      @l_where_str nvarchar(4000),
      @l_count_query nvarchar(4000),
      @l_end_gen_row_num integer,
      @l_start_gen_row_num integer,
      @l_select_str nvarchar(4000),
      @l_stat_col nvarchar(4000),
      @l_insert_to_temp nvarchar(4000),
      @l_create_temp nvarchar(4000),
      @l_stat_col_alias nvarchar(20)
   BEGIN

      -- Extract the col only that we need to run statistics on.
      -- First extract the content in the function call.
```

```
    SET @l_stat_col = @p_select_str
    SET @l_stat_col = SUBSTRING(@l_stat_col,
            PATINDEX('%(%', @l_stat_col) + 1,
            PATINDEX('%)%', @l_stat_col) - PATINDEX('%(%', @l_stat_col) - 1)

    -- Then extract the column from the distinct clause.
    SET @l_stat_col = LTRIM(RTRIM(@l_stat_col))
    IF PATINDEX('%DISTINCT %', UPPER(@l_stat_col)) = 1
       SET @l_stat_col = SUBSTRING(@l_stat_col, PATINDEX('% %',
@l_stat_col) + 1, LEN(@l_stat_col))

    -- Get the select column name without alias.
    SET @l_select_str = SUBSTRING(@l_stat_col, PATINDEX('%.%',
@l_stat_col) + 1, LEN(@l_stat_col))

    -- Set up the from string.
    SET @l_from_str = '[Asset Categories] Asset_Categories_'

    -- Set up the join string.
    SET @l_join_str = @p_join_str
    IF @p_join_str IS NULL
       SET @l_join_str = ' '

    -- Set up the search string.
    SET @l_where_str = ' '
    IF @p_where_str IS NOT NULL
       SET @l_where_str = @l_where_str + 'WHERE ' + @p_where_str

    -- Get the list.
    IF @p_page_number > 0 AND @p_batch_size > 0
    BEGIN
       -- If the caller did not pass a sort string,
       -- use a default value.
       IF @p_sort_str IS NOT NULL
          SET @l_sort_str = 'ORDER BY ' + @p_sort_str;
       ELSE
          SET @l_sort_str = ' '
```

```
-- Calculate the rows to be included in the list
-- before geting the list.
SET @l_end_gen_row_num = @p_page_number * @p_batch_size
SET @l_start_gen_row_num = @l_end_gen_row_num - (@p_batch_size-1)

-- Creating a temporary table to keep the numbering
-- of the rows returned.  It keeps the same primary
-- key columns as the base table besides an identity
-- field which keeps the numbering.
Select 1 As tempCol Into #IS_TEMP_TABLE_SECONDARY;

-- Create the temporary table from the
SET @l_create_temp =
   'SELECT Identity(int, 1, 1) As IS_ROWNUM_COL, ' +
   @l_stat_col +
   ' INTO #IS_TEMP_FROM' +
   ' FROM ' + @l_from_str + ' ' +
   ' CROSS JOIN #IS_TEMP_TABLE_SECONDARY' +
   ' WHERE 1=2'
-- Insert records into the temporary table from the
-- base table
SET @l_insert_to_temp =
   'INSERT INTO #IS_TEMP_FROM ' +
   '(' + @l_select_str + ') ' +
   'SELECT ' + @l_stat_col +
   ' FROM ' + @l_from_str + ' ' + @l_join_str + ' ' +
   @l_where_str + ' ' +
   @l_sort_str

-- Construct the query for the current page
SET @l_query =
   'SELECT ' + @p_select_str + ' ' +
   'FROM ( ' +
   'SELECT ' + @l_select_str + ', IS_ROWNUM_COL ' +
   'FROM #IS_TEMP_FROM ' +
   'WHERE IS_ROWNUM_COL >= ' + convert(varchar,
@l_start_gen_row_num) +
            ' AND IS_ROWNUM_COL <= ' + convert(varchar,
@l_end_gen_row_num) +
            ') ' + 'Asset_Categories_'

          -- Run the query and get the result for the current page
          EXECUTE (@l_create_temp + '     ' + @l_insert_to_temp + '     ' + @l_query
+ ' ');

        END
        ELSE
        -- Return the empty result if page number or batch size
        -- has invalid number
        BEGIN
          SET @l_query =
            'SELECT 1 from ' + '[Asset Categories] Asset_Categories_ ' +
            'WHERE 1=2;'
          EXECUTE (@l_query)
        END
      END

------------------------------------------------------------------------------------------------
```

Appendix C Glossary

The Glossary is useful since these below technologies are being used for the project, which needs a small description.

ASP.NET

The key to .NET technology for developing XML Web Services is ASP.NET. ASP.NET provides framework for developing Web based applications in which the user interacts with a service using a browser. Although the main emphasis of ASP.NET is to build conventional web applications, Microsoft has added support for building XML Web Services as well.

ASP.NET vs. Traditional ASP

ASP.NET provides a true language neutral execution framework for web application to use. This means that whether we use C#, VB, Jscript .NET, C++ or Perl the code will compile to IL (Intermediate Languages) and then executed by .NET Framework. This also means that ASP.NET will take full advantages of .NET Framework based classes through and compiled languages. However, compared to the traditional ASP, the choice of languages is limited with VB script and Java script.

Another major difference between traditional ASP and ASP.NET is that traditional

ASP pages are parsed each time they are requested, which will make the ASP execution slow. On the other hand, when we execute an ASP.NET page for the first time it gets compiled into a binary .dll (dynamic library link) and then get executed. All further requests are served by this compiled code, making execution faster.

ASP.NET and XML

The relation between ASP.NET and XML can be summarized but not limited to the following points:

- ASP.NET uses XML extensively to represent Web Controls and Configuration setting. It relies on special XML document web.config for configuration purposes.
- ASP.NET provides Web Controls that can deal with XML data and display it as our requirement, which can be either tabular or style sheet-based.
- The creation of XML Web Services in .NET platform is done through ASP.NET. Since it is a huge process to see how we can create a Web Service in .NET and a client, I have shown this in the appendix.
  - *"**XML Web Services** are a category of software components that provide functionality over the network*" – a general definition to Web Services.

However, if we define Web Services in terms of functionality, then XML Web Services let applications share data, and more powerfully invoke capabilities from other application without regards to how those applications were built, what operating system or platform they run on, and what device are used to access them. While XML Web Services remain independent of each other, they can loosely link themselves into a collaborating group that performs a particular task.

XML Web Services are invoked by means of industry-standard protocols includes SOAP, WSDL, UDDI and XML. They are defined through public standard organization namely World Wide Web Consortium (W3C). We will define each of these underling technologies, which are the building blocks of any XML Web Services –

- **Simple Object Access Protocol** (SOAP) is the communication protocol for XML Web Services, which provides a standard way of packaging messages. A SOAP message is composed of an envelope that contains the body of the message and any header information used to describe the message, which is an XML document that follows specific XML schema. The latest version of SOAP is 1.1. However, in 2001, a working group draft of SOAP 1.2 was published (www.w3.org/TR/2001/WD-soap12-20010709) by the XML Protocol Working Group.)

- **Web Service Description Language** (WSDL) is an XML-based language layered on top of the schema that describes a Web Service. It provides the information necessary for a client to interact with the Web Service. WSDL specifies what a request message must contain and what the response message will look like in unambiguous notation. WSDL is extensible and can be used to describe practically any network service, including SOAP over HTTP and even protocols that are not XML-based, such as DCOM over UDP.

- **Universal Description, Discovery, and Integration (UDDI)** provides a central directory service for publishing technical information about Web Services. In other words, it is the yellow pages of Web Services. UDDI is the result of an industry initiative backed by a significant number of technology companies, including Microsoft, IBM, SAP, and Ariba.

All the above protocols and standards we discussed so far are based on the Extensible Markup Language. Indeed, *XML is the backbone of any Web Service*[3]. "XML is a vehicle for information that brings usable data to the desktop and is a universal data format that does for data what HTML does for web content- it provides the necessary markup".