eProcurement System
Application Environment and Setup

Table Content

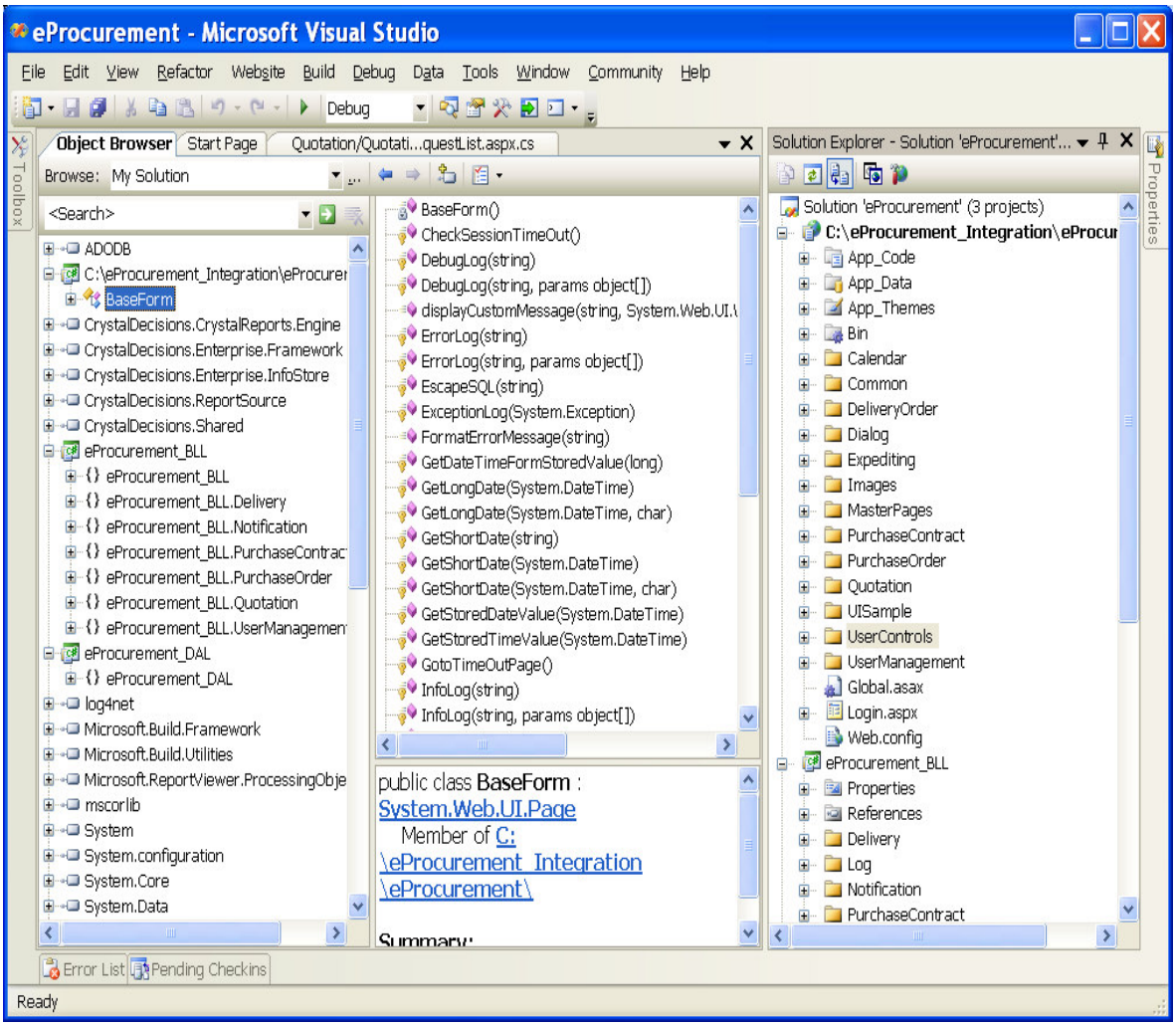
eProcurement Application Information and Setup

1 Application Development Environment

eProcurement System is a web application which developed using Microsoft .NET Platform. The eProcurement .NET project uses following versions of IDE for development:

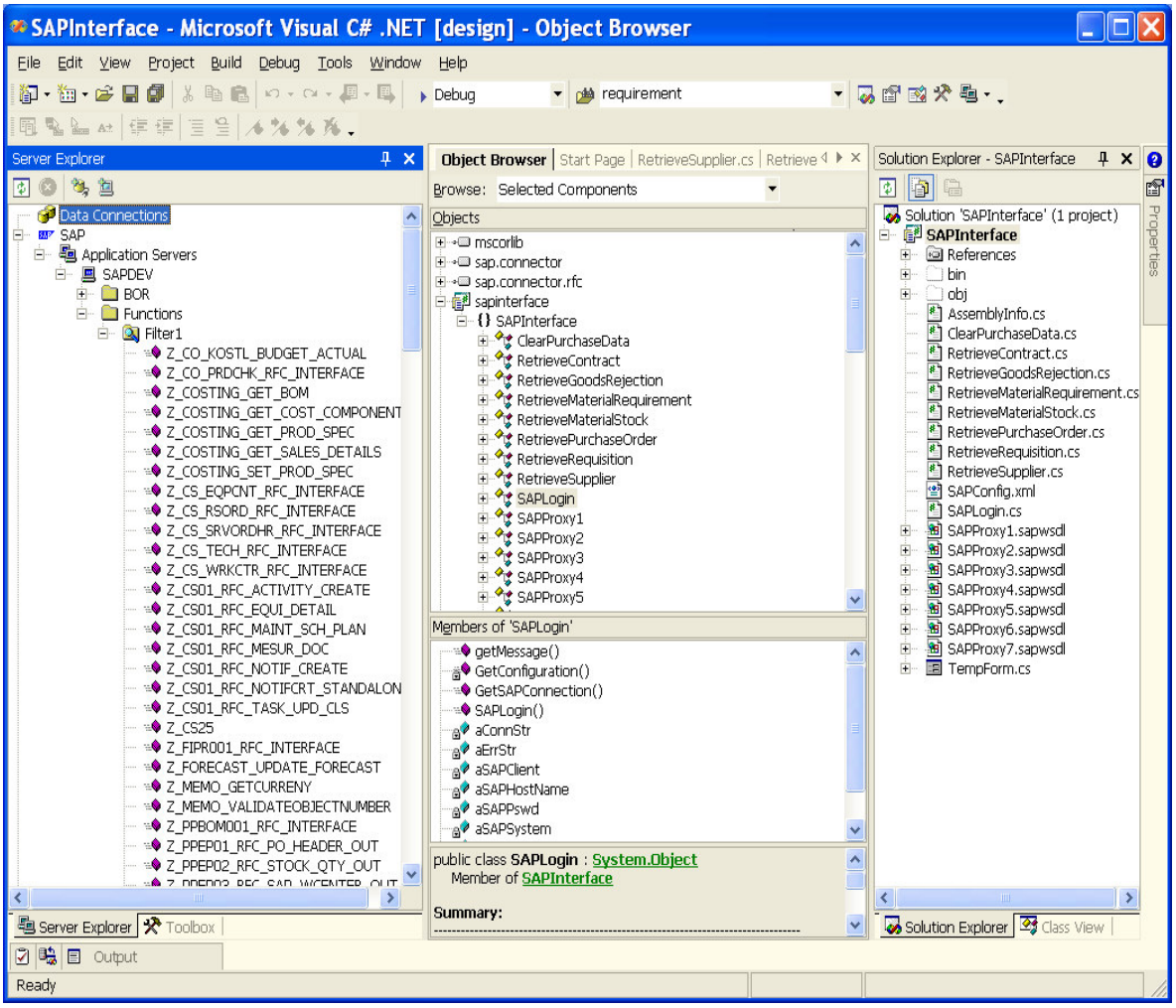
1.1 Visual Studio 2005

The version 2005 of visual studio .NET is being used to design and develop the codes for eProcurement Web Site i.e. web pages (.aspx) and code behind web pages (C#). The classes for business logic, data access layer, entity model layer and eProcurement interface layer is developed using C# programming language.



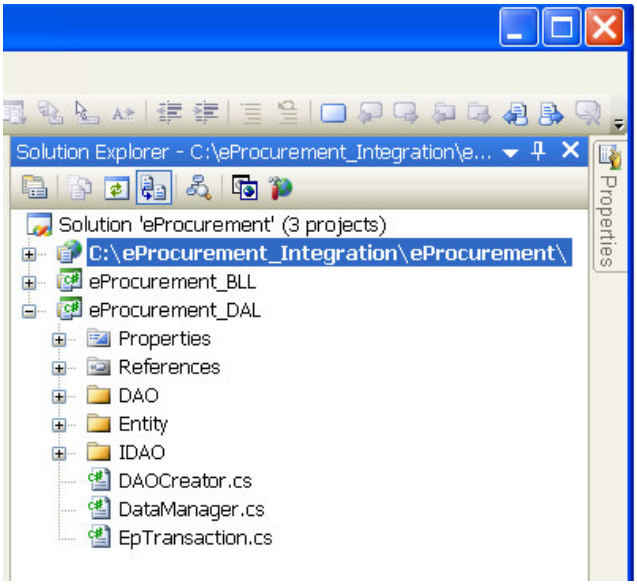
1.2 Visual Studio 2003

The version 2003 of visual studio .NET is being used to design and develop the SAP client interface, as SAP .NET connector provided by SAP works with .NET 2003 version.



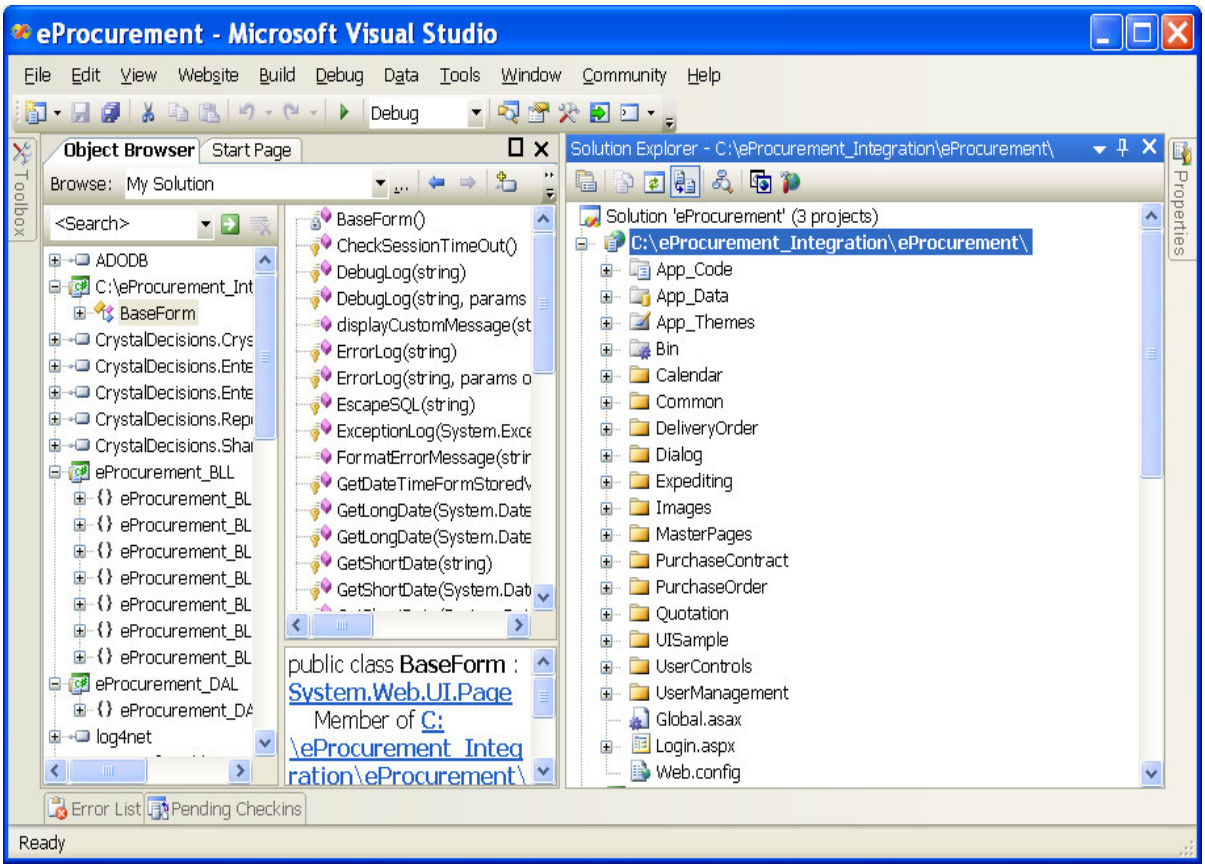
2 Application Layering

The eProcurement application architecture is made up of following layers:



2.1 Presentation Layer

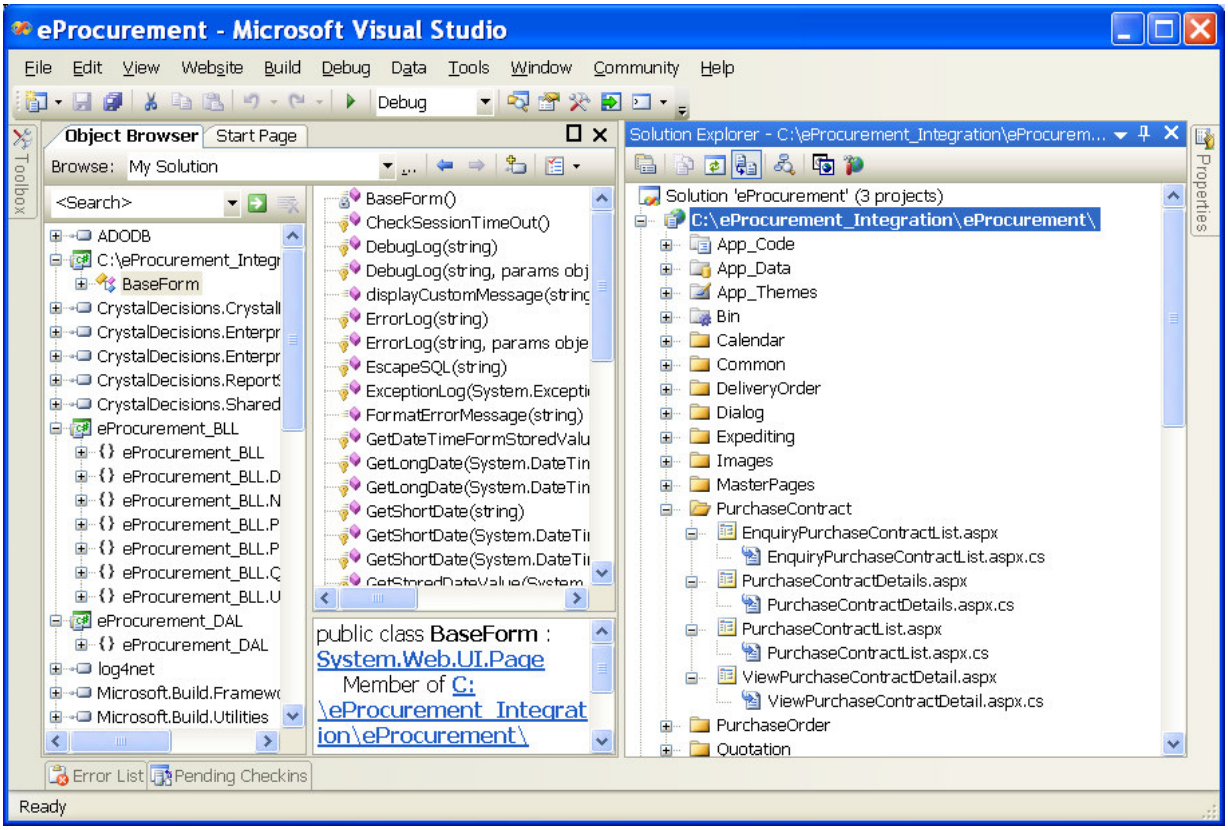
This is a UI Layer which is developed using web pages (.aspx) and web controls from .NET WEB control libraries. Please refer to the below screen capture for the project related to eProcurement presentation layer (web pages).

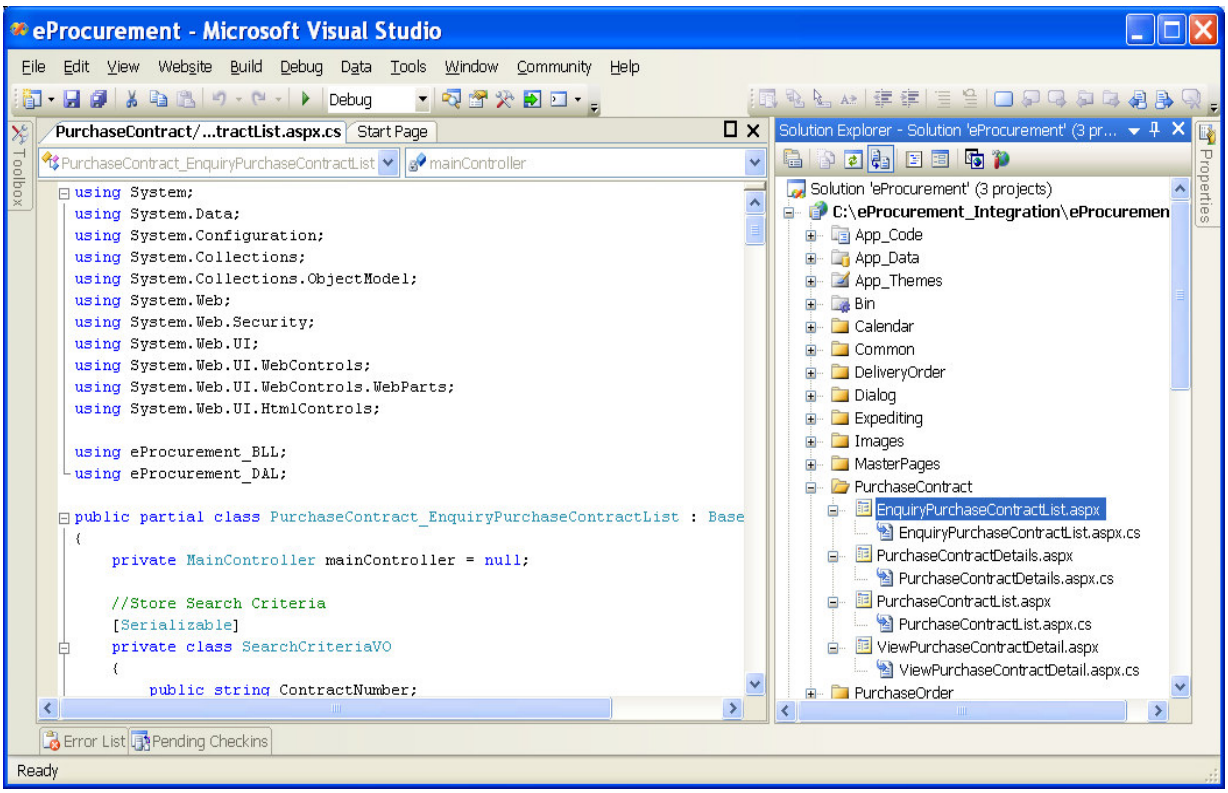


2.2 Presentation View Controller

This view controller layer is the code behind the web pages (.aspx) is developed using .NET libraries in C#. This layer communicates with the business logic layer to send the messages to the Data Access layer to create the instances of the entities from eProcurement in order to display the requested information by the client browser through internet.

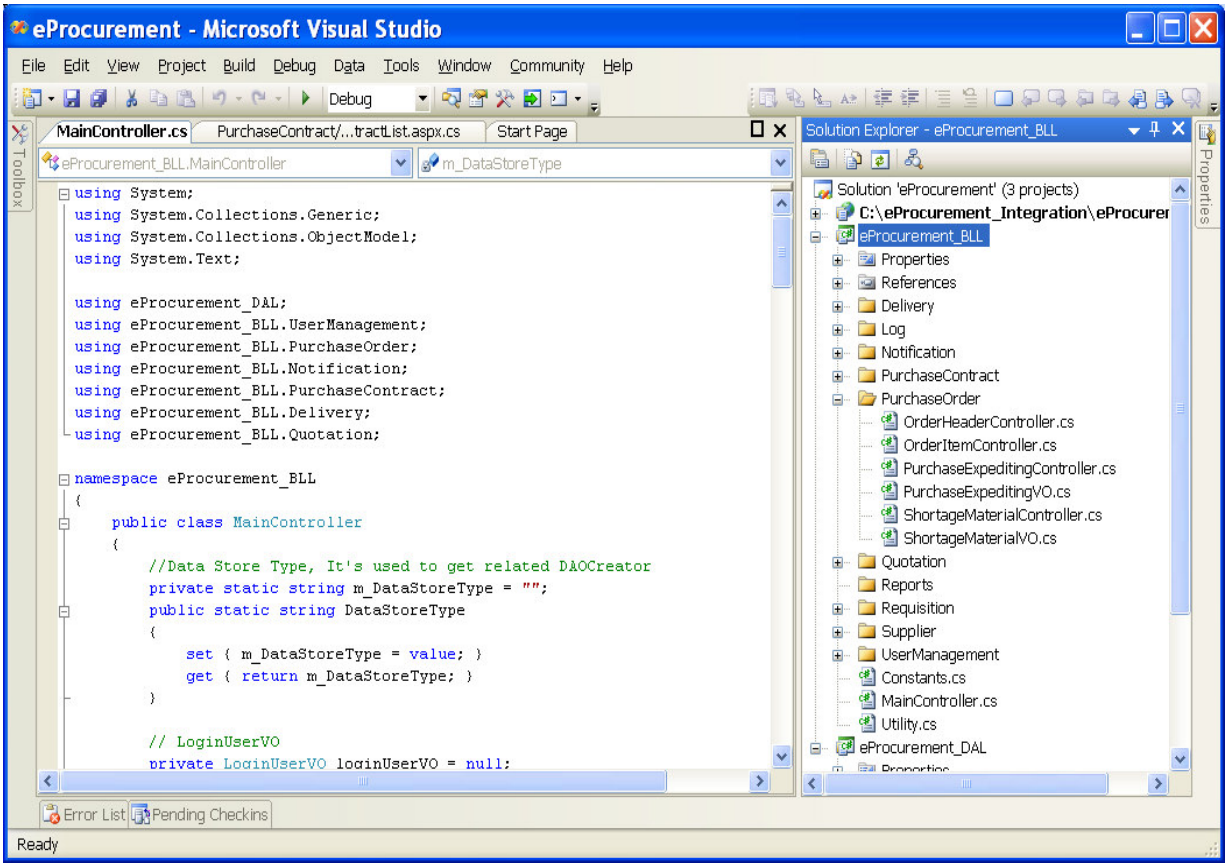
Below is the sample screen capture for the view controller the C# code behind for individual web page.


eProcurement - Microsoft Visual Studio
File Edit View Website Build Debug Data Tools Window Community Help
Debug
Object Browser
Start Page
Browse: My Solution
<Search>
ADODB
C:\eProcurement_Integr
BaseForm
CrystalDecisions.Crystall
CrystalDecisions.Enterpr
CrystalDecisions.Enterpr
CrystalDecisions.Report
CrystalDecisions.Shared
eProcurement_BLL
eProcurement_BLL
eProcurement_BLL.D
eProcurement_BLL.N
eProcurement_BLL.P
eProcurement_BLL.P
eProcurement_BLL.Q
eProcurement_BLL.U
eProcurement_DAL
eProcurement_DAL
log4net
Microsoft.Build.Framew
Microsoft.Build.Utilities
BaseForm()
CheckSessionTimeOut()
DebugLog(string)
DebugLog(string, params obj
displayCustomMessage(string
ErrorLog(string)
ErrorLog(string, params obje
EscapeSQL(string)
ExceptionLog(System.Excepti
FormatErrorMessage(string)
GetDateTimeFormStoredValu
GetLongDate(System.DateTin
GetLongDate(System.DateTin
GetShortDate(string)
GetShortDate(System.DateTi
GetShortDate(System.DateTi
public class BaseForm :
System.Web.UI.Page
Member of C:
\eProcurement_Integrat
ion\eProcurement\
Solution Explorer - C:\eProcurement_Integration\eProcurem...
Solution 'eProcurement' (3 projects)
C:\eProcurement_Integration\eProcurement\
App_Code
App_Data
App_Themes
Bin
Calendar
Common
DeliveryOrder
Dialog
Expediting
Images
MasterPages
PurchaseContract
EnquiryPurchaseContractList.aspx
EnquiryPurchaseContractList.aspx.cs
PurchaseContractDetails.aspx
PurchaseContractDetails.aspx.cs
PurchaseContractList.aspx
PurchaseContractList.aspx.cs
ViewPurchaseContractDetail.aspx
ViewPurchaseContractDetail.aspx.cs
PurchaseOrder
Quotation
Error List
Pending Checkins
Ready


eProcurement - Microsoft Visual Studio
File Edit View Website Build Debug Data Tools Window Community Help
Debug
PurchaseContract/...tractList.aspx.cs
Start Page
PurchaseContract_EnquiryPurchaseContractList
mainController
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Collections.ObjectModel;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using eProcurement_BLL;
using eProcurement_DAL;
public partial class PurchaseContract_EnquiryPurchaseContractList : Base
{
private MainController mainController = null;
//Store Search Criteria
[Serializable]
private class SearchCriteriaVO
{
public string ContractNumber;
Solution Explorer - Solution 'eProcurement' (3 pr...
Solution 'eProcurement' (3 projects)
C:\eProcurement_Integration\eProcuremen
App_Code
App_Data
App_Themes
Bin
Calendar
Common
DeliveryOrder
Dialog
Expediting
Images
MasterPages
PurchaseContract
EnquiryPurchaseContractList.aspx
EnquiryPurchaseContractList.aspx.cs
PurchaseContractDetails.aspx
PurchaseContractDetails.aspx.cs
PurchaseContractList.aspx
PurchaseContractList.aspx.cs
ViewPurchaseContractDetail.aspx
ViewPurchaseContractDetail.aspx.cs
PurchaseOrder
Error List
Pending Checkins
Ready

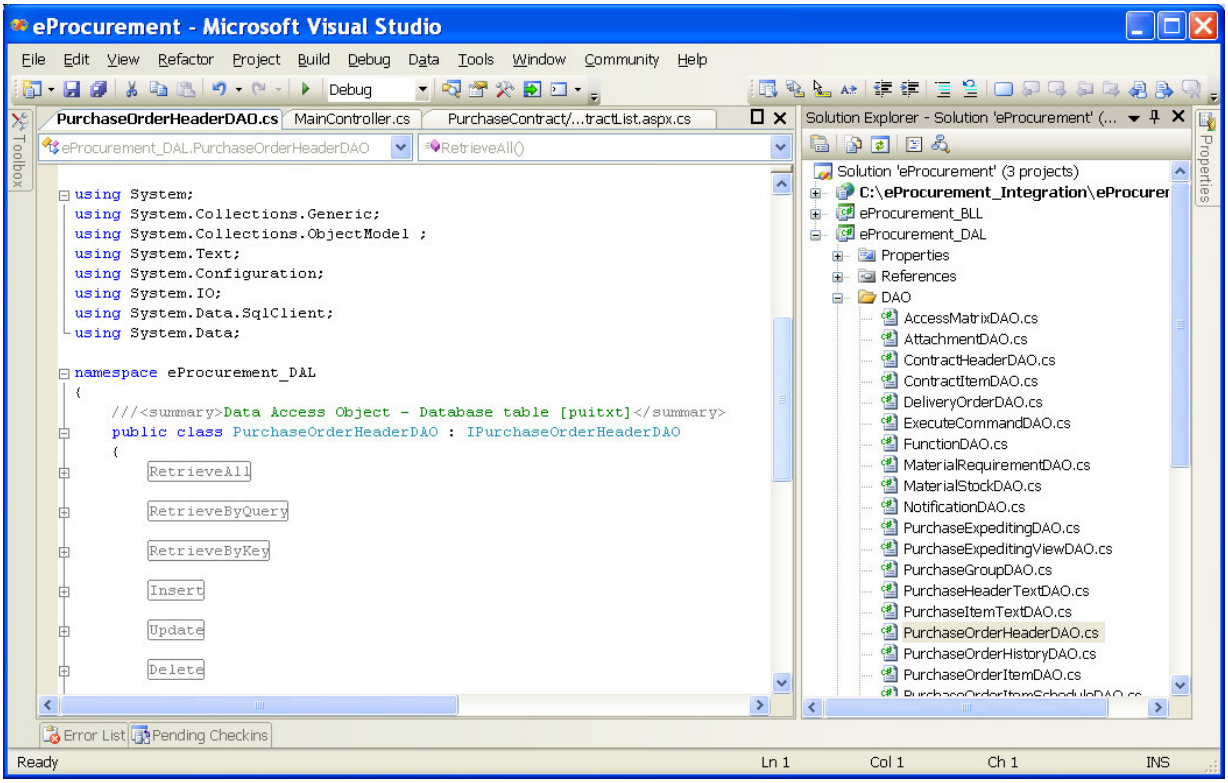
2.3 Business Logic Layer

The business logic layer which is also referred to controller layer is developed using .NET C# and its libraries. This layer is mainly governing the controlling of the application and manages the request and provides an bridge between the view controller components and data access layer for any requested information. Below is the sample screen capture of the Business Logic Layer → example Main Controller



2.4 Data Access Layer

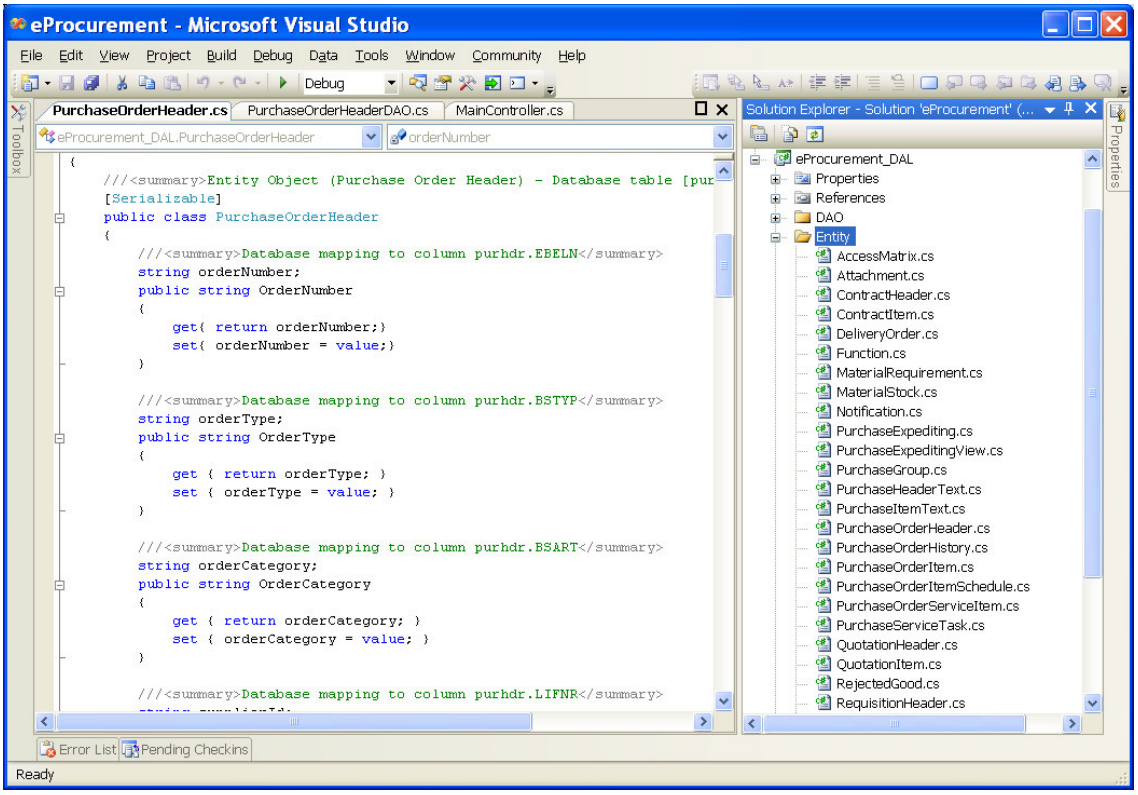
The data access layer which is also referred to DAO (Data Access object) layer developed using .NET C# and its libraries. The main feature of this DAO is the Data Manager class which controls the database transaction and manages all the DAO instance request of retrieval, insert, update and delete through transaction commit and rollback. This is a customized Transaction control which manages data insert, update and delete to the multiple tables within one transaction. This layer consists of various DAO classes built for managing the various table updates. Below is the screen capture for the data access layer showing the DAO classes → example →Purchase OrderHeaderDAO.cs



2.5 Entity Model

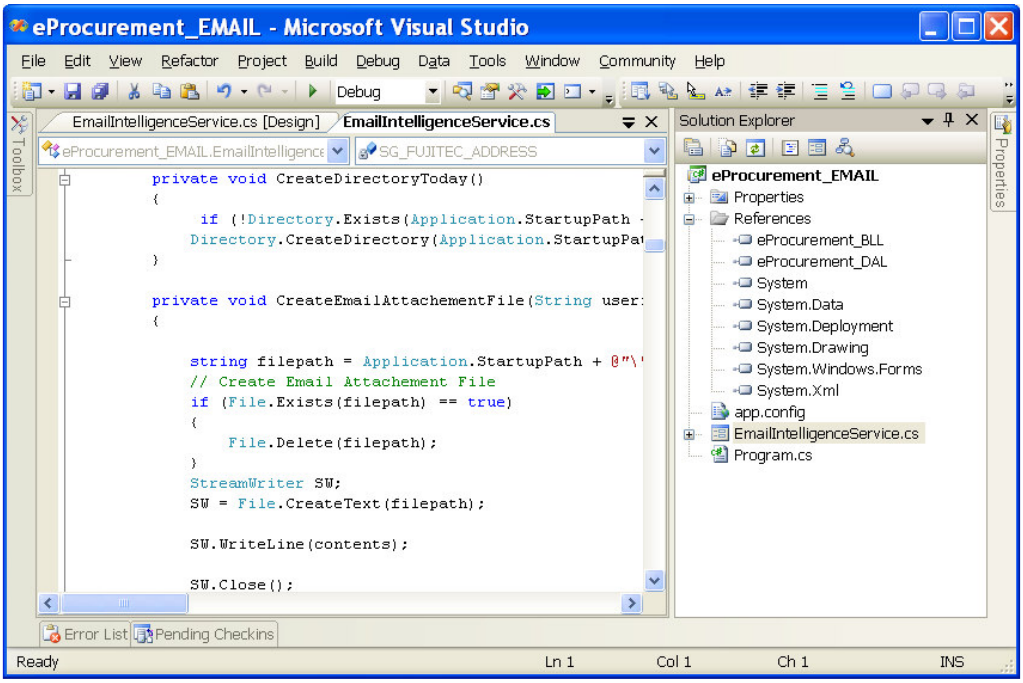
This layer consists of classes defining the individual entities residing with the eProcurement Application. These entity classes are developed using C# and mostly contains the attributes of the entities and setter and getter methods i.e. set and get property value. For Example the entities in eProcurement are: Supplier, Purchase Order, Purchase Contract, Purchase Requisition, Material Stock, and Request for Quotation (RFQ), Quotation, and User etc.

Below is the screen capture showing the Entity models defined in the eProcurement System.



2.6 Email Notification

This application is deployed to process all the notification created during synchronization and updates through web site about eProcurement information such as new purchase orders, contracts, amendments, expediting of purchase orders etc. This application will consolidate and send the individual recipient a mail with attachment about the notification of any changes or new additions.

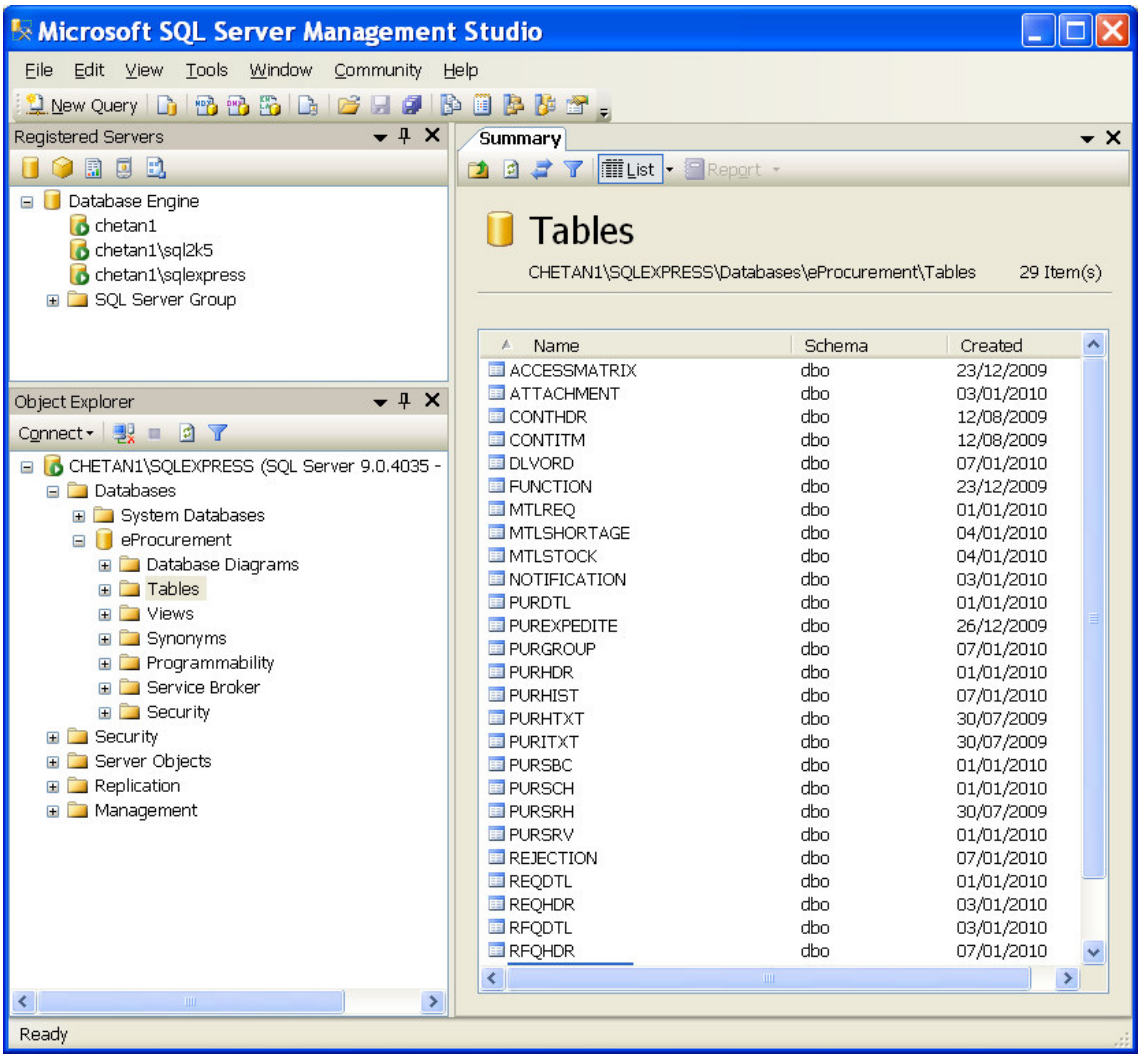


3 Application Database

The eProcurement database is designed in Microsoft SQL Server 2005. The name of the database is **"eProcurement"**. The database files will be available under database folder of the application in the CD. The files are MDF and LDF. The administrator needs to copy the MDF file into the SQL application server by creating appropriate folder **"eProcurement"** at the root folder example **C:\eprocurement.**

Invoke the enterprise SQL Manager from the SQL server and attach the database by giving appropriate path as mentioned in above paragraph.

Below is the screen capture showing the SQL Server 2005 Environment with eProcurement Database Loaded.

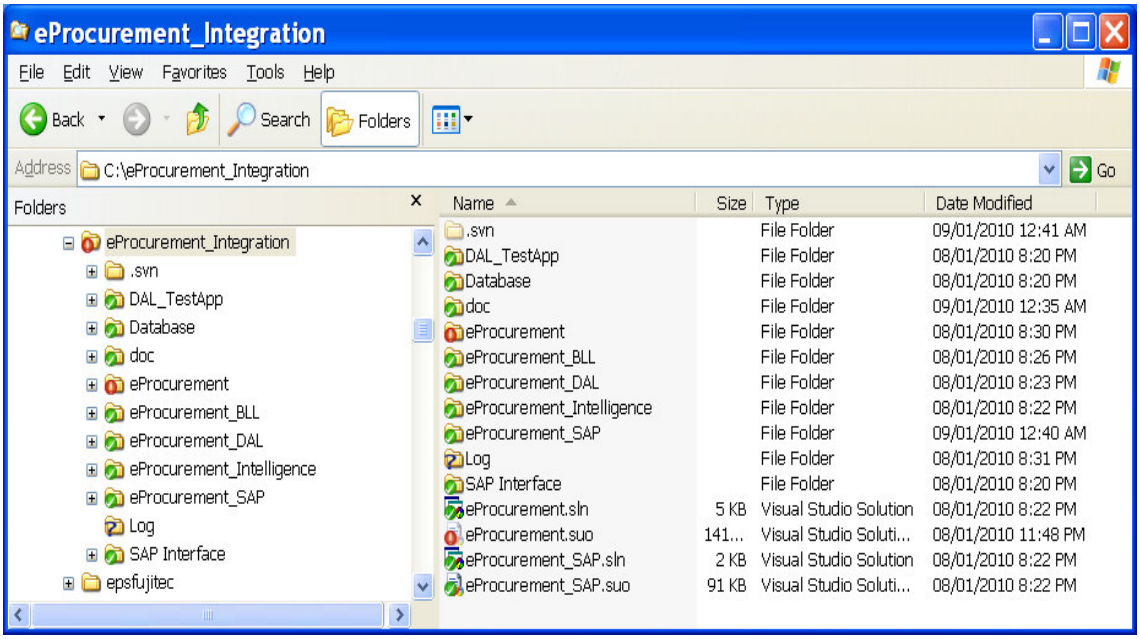


4 Application Project Structure

The eProcurement application consists of one project solution with following projects for various layers discussed above

[1] eProcurement (Web site) → Presentation
[2] eProcurement_BLL → Business Logic Layer
[3] eProcurement_DAL → Data Access Layer & Model Entities
[4] eProcurement_Intelligence → Creation of Notification and sending Emails
[5] eProcurement_SAP → Interface Layer of eProcurement Application
[6] SAP Inetrface → SAP client Interface middle layer to communicate to remote SAP R/3 System

Folder Structure

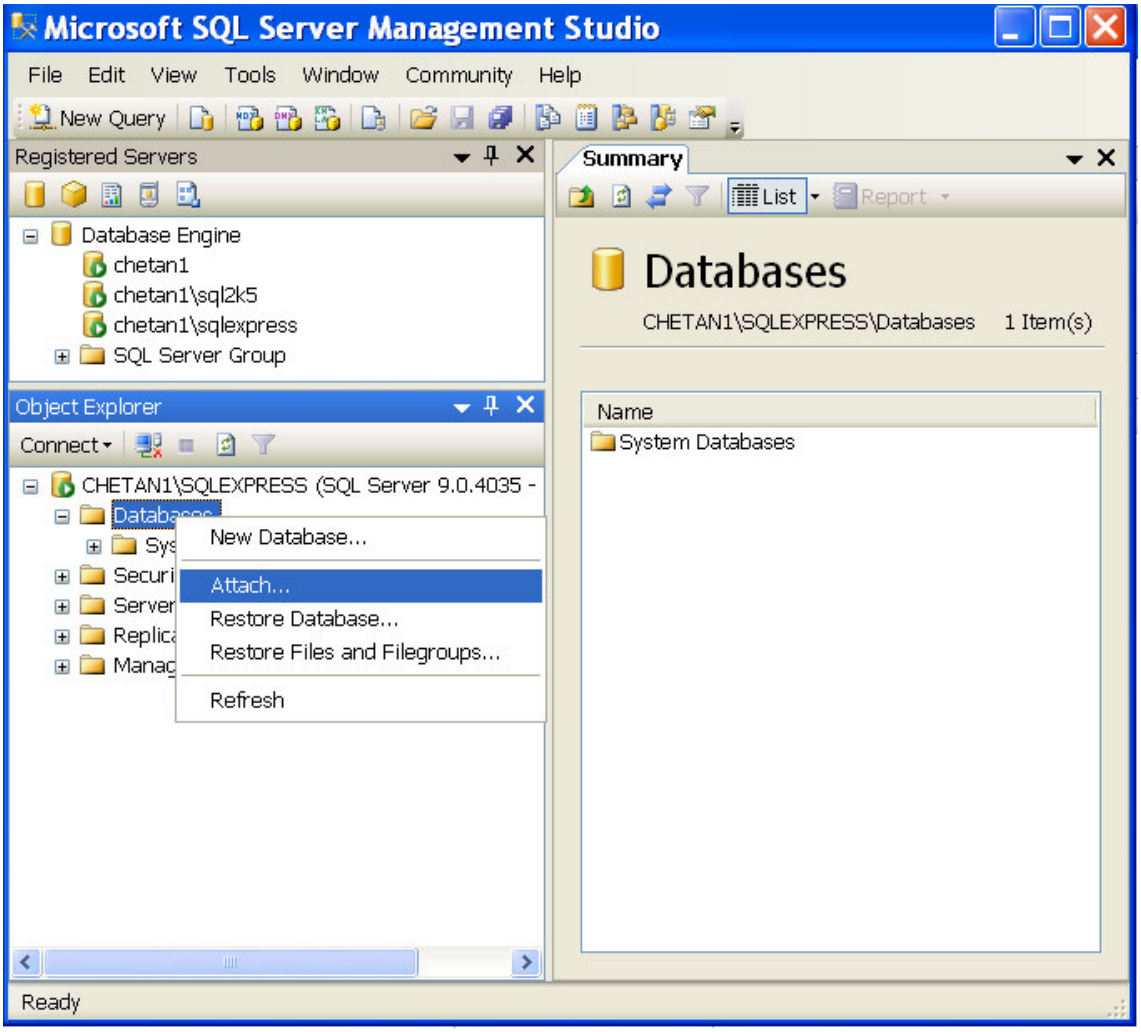


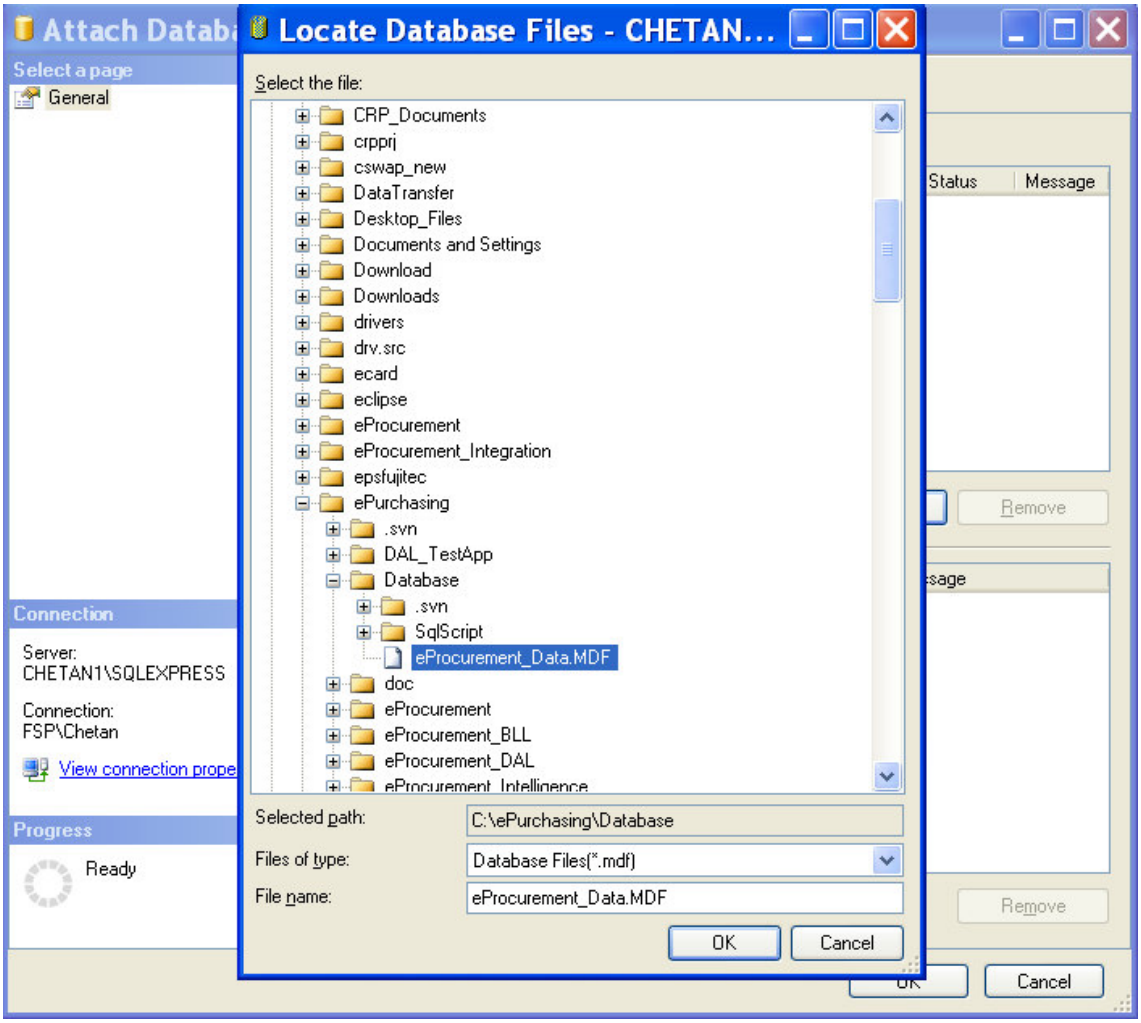
Folders	Description
eProcurement	Presentation → Web pages and scripts, images, utilities, code behind of Web pages (classes) etc
eProcurement_BLL	Business Logic Layer → Controllers classes
eProcurement_DAL	Data Access layer → DAO, IDAO, Entity classes
eProcurement_SAP	Client Side Interface for synchronization consists of all interface controller classes
SAP_Interface	SAP Client Interface → Remote Function calls to access the SAP R/3 System, Consists of SAP Connection, SAP .NET Connector Libraries and Client interface classes
eProcurement_Intelligence	Email Notification Application related classes

5 Installation Procedure

5.1 Database Installation

The eProcurement database is designed in Microsoft SQL Server 2005. The name of the database is **"eProcurement"**. The database files will be available under database folder of the application in the CD. The files are MDF and LDF. The administrator needs to copy the MDF file into the SQL application server by creating appropriate folder **"eProcurement"** at the root folder example **C:\eprocurement.**





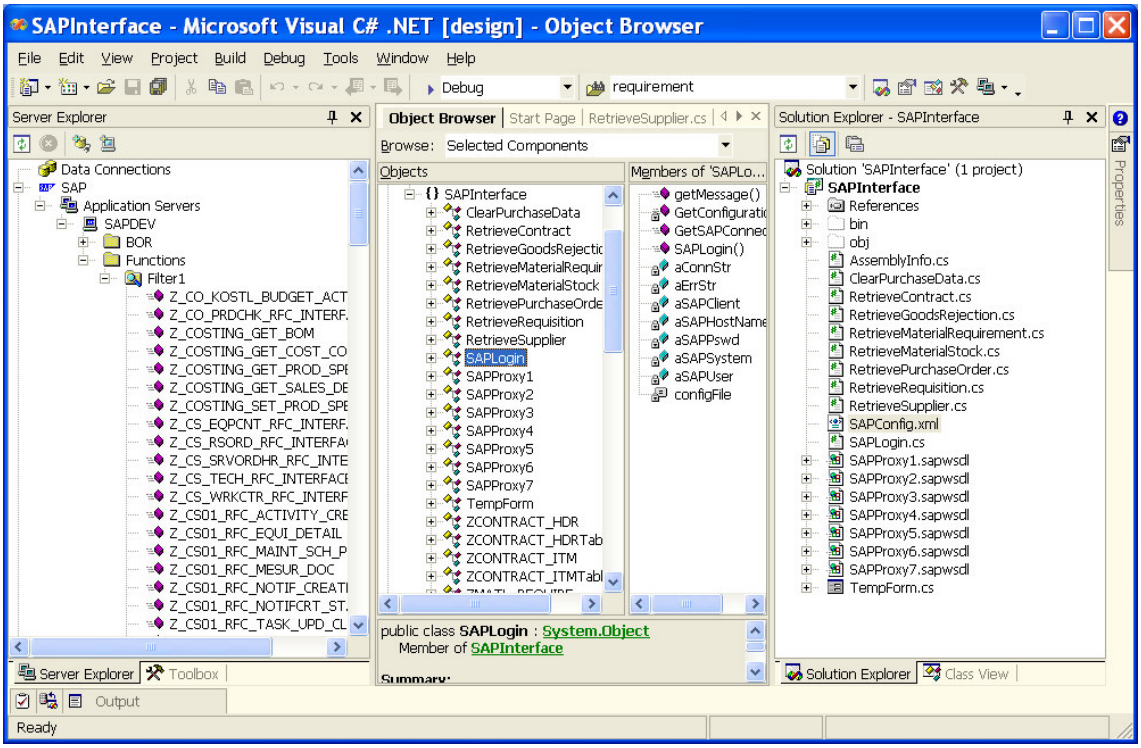
5.2 Web Application Deployment

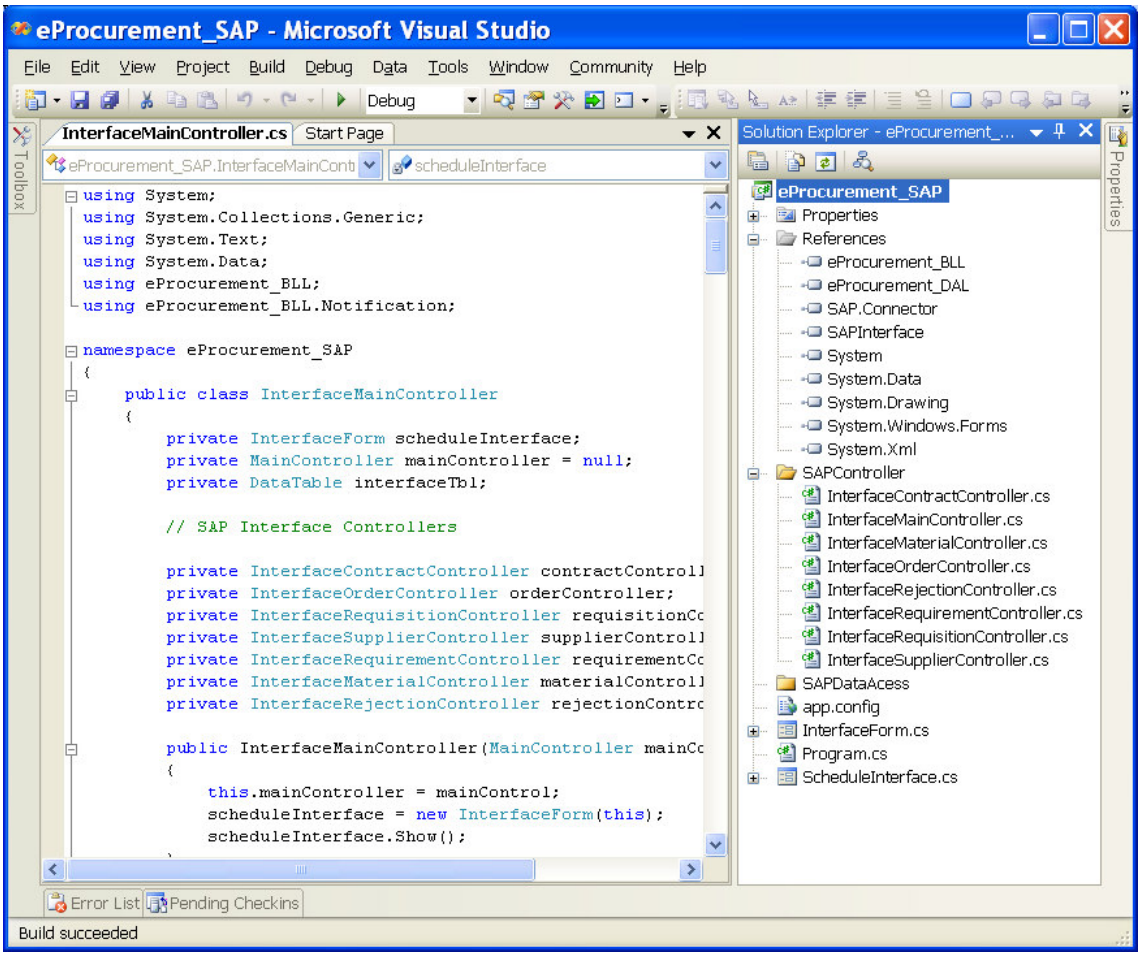
To install the eProcurement system, open the solution file into visual studio 2005 IDE and execute the project publishing option. It will request for the IIS server to publish the eProcurement web application.

5.3 eProcurement Interface

This is a windows application which is required to be deployed on the windows server within the company's network (Local Area Network). This application can be either schedule as a job or can be executed as normal windows application to execute the interface between SAP and eProcurement system to synchronize the relevant information.

Below is the screen capture for the eProcurement → SAP Client Interface and eProcurement_SAP projects respectively.


SAPInterface - Microsoft Visual C# .NET [design] - Object Browser
File Edit View Project Build Debug Tools Window Help
Debug
requirement
Server Explorer
Object Browser | Start Page | RetrieveSupplier.cs
Browse: Selected Components
Objects
Members of 'SAPLo...
Solution Explorer - SAPInterface
Solution 'SAPInterface' (1 project)
SAPInterface
References
bin
obj
AssemblyInfo.cs
ClearPurchaseData.cs
RetrieveContract.cs
RetrieveGoodsRejection.cs
RetrieveMaterialRequirement.cs
RetrieveMaterialStock.cs
RetrievePurchaseOrder.cs
RetrieveRequisition.cs
RetrieveSupplier.cs
SAPConfig.xml
SAPLogin.cs
SAPProxy1.sapwsdl
SAPProxy2.sapwsdl
SAPProxy3.sapwsdl
SAPProxy4.sapwsdl
SAPProxy5.sapwsdl
SAPProxy6.sapwsdl
SAPProxy7.sapwsdl
TempForm.cs
public class SAPLogin : System.Object
Member of SAPInterface
Summary:
Server Explorer
Toolbox
Output
Solution Explorer
Class View
Ready


eProcurement_SAP - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
Debug
InterfaceMainController.cs
Start Page
Solution Explorer - eProcurement_...
eProcurement_SAP
Properties
References
eProcurement_BLL
eProcurement_DAL
SAP.Connector
SAPInterface
System
System.Data
System.Drawing
System.Windows.Forms
System.Xml
SAPController
InterfaceContractController.cs
InterfaceMainController.cs
InterfaceMaterialController.cs
InterfaceOrderController.cs
InterfaceRejectionController.cs
InterfaceRequirementController.cs
InterfaceRequisitionController.cs
InterfaceSupplierController.cs
SAPDataAcess
app.config
InterfaceForm.cs
Program.cs
ScheduleInterface.cs
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using eProcurement_BLL;
using eProcurement_BLL.Notification;
namespace eProcurement_SAP
{
public class InterfaceMainController
{
private InterfaceForm scheduleInterface;
private MainController mainController = null;
private DataTable interfaceTbl;
// SAP Interface Controllers
private InterfaceContractController contractControll
private InterfaceOrderController orderController;
private InterfaceRequisitionController requisitionCc
private InterfaceSupplierController supplierControll
private InterfaceRequirementController requirementCc
private InterfaceMaterialController materialControll
private InterfaceRejectionController rejectionContrc
public InterfaceMainController(MainController mainCc
{
this.mainController = mainControl;
scheduleInterface = new InterfaceForm(this);
scheduleInterface.Show();
Error List
Pending Checkins
Build succeeded

5.3.1 SAP Connection Configuration

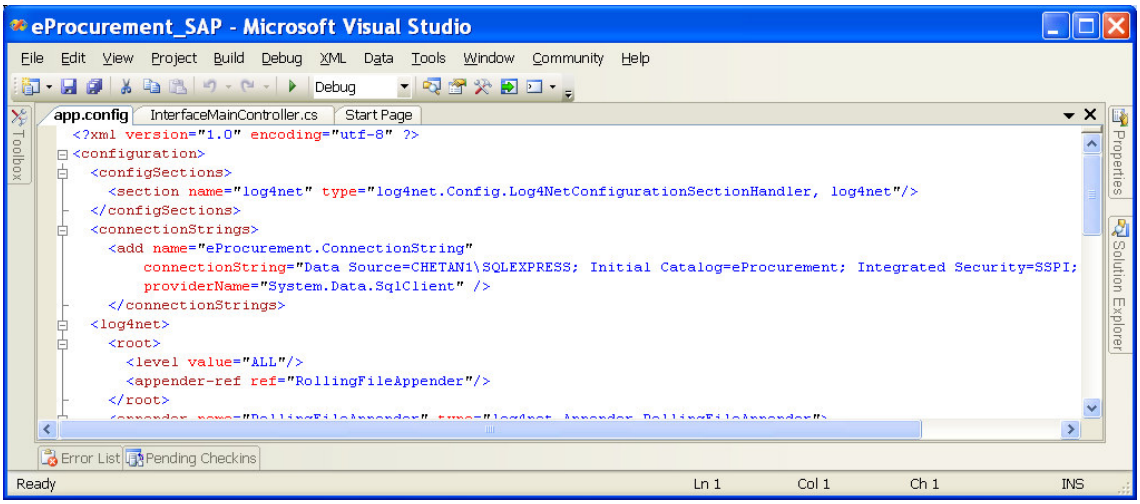
During the deployment following config.xml file needs to configured and copied into the application folder on the server. This config.xml file contains details about the SAP connectivity i.e. ***SAP Hostname, SAP system Number (or ID), SAP Client ID, SAP User Profile and SAP User Password***.

```
<?xml version="1.0" encoding="utf-8" ?>
<SAPCONNECTION>
 <SAPHOST>SAPPR3</SAPHOST>
 <SAPSYSTEM>00</SAPSYSTEM>
 <SAPCLIENT>330</SAPCLIENT>
 <SAPUSER>SAPADMIN</SAPUSER>
 <SAPPASSWORD>XXXXXXXX</SAPPASSWORD>
</SAPCONNECTION>
```

5.3.2 eProcurement Database Connection Configuration

In the eProcurement interface application, under the application folder there is a file with name ***app.config***. Open this file and set the appropriate name of the Data Source equivalent to → *SQL Server Instance Name* and database as ***"eProcurement"*** under the tag "***connectionstring***". Once done, please save the file.

Important Note: As this an interface to the SAP R/3 System, this application will not execute as this requires connecting to SAP R/3 system.



5.4 Email Notification

This is a windows application which is required to be deployed on the windows server within the company's network (LAN). This application can be scheduled to run at specified interval of day. This application will consolidate all the notification messages by recipient and notification type and send as attachment to the recipient's mail id using the company's SMTP gateway server