ISO ANALYZER AND REPORTING SYSTEM

A Paper
Submitted to the Graduate Faculty of
the
North Dakota State University
of Agriculture and Applied Science

By

Vishnu Vardhana Reddy Yarram

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Program:
Computer Science

January 2013

Fargo, North Dakota

North Dakota State University
Graduate School

Title

ISO ANALYZER AND REPORTING SYSTEM

By

Vishnu Vardhana Reddy Yarram

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Kendall Nygard

Chair

Dr. Saeed Salem

Dr. Chao You

Approved by Department Chair:

Jan 3rd 2013

Date

Dr. Brian M. Slator

Department Chair

ABSTRACT

This paper describes the development of a GUI-based application program called the ISO Analyzer and Reporting System. The Independent System Operator (ISO) Analyzer and Reporting System facilitates the study of electricity markets by empowering its users with the ability to query and to generate reports from market and operational data by allowing them to analyze data over a period of time, which is helpful to draw useful conclusions. The system downloads publicly available files on energy load and pricing data from an independent system operator's website and stores data from the files in the form of SQL server database tables for querying and generating reports. In addition, ISO reports can dynamically create certain custom queries to extract information from a database, based on the user's selection. ISO maintenance can clean the database as a regular activity. This system pertains to the New York Independent System Operator (NYISO).

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Kendall Nygard for his constant guidance and support. This paper would not have been possible without him. I would also like to thank all the supervisory committee members for their time and consideration.

DEDICATION

Dedicated to my lovely wife Kiranmai, my parents, sister, brother and brother-in-law who have always loved me unconditionally and supported me no matter what.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Studying electricity markets, such as analyzing energy usage patterns; forecasting energy needs; or finding an optimal solution for energy generation, distribution, consumption, pricing, etc., require analyzing the relevant historic and recent data. For a simple simulation or a simple study, one could work with hypothetical data. In order to observe real-world scenarios and to find an efficient way to manipulate and analyze data, it is desirable to work with real time data which would lead us to make informed decisions about the energy supply.

Various independent system operators (ISOs), administering electricity in different regions, have made their market and operation data available to the public. The New York Independent System Operator (NYISO) publishes data online which is available for everyone [1]. NYISO maintains different types of market and operation data on load, such as load forecast for the current day and next day, real-time load, etc. Similarly, it maintains different data for pricing, such as day-ahead, hour-ahead, and real-time pricing information. Energy load and pricing data are uploaded online to NYISO's website on a daily basis in different file formats, such as Excel, PDF, zip, and comma-separated value (CSV) files [2]. For each day, there is a new file containing data with the timestamps from the start to the end of a day. NYISO has data collection from 5-10 years back or more, depending on the type of data. Given that the data with which to work is already available, we need a system that can manipulate and query the data to extract useful information. Thus, there was an idea of a graphical user interface (GUI-) based solution that enables users to selectively download CSV files from the web and to upload file contents into SQL server database tables.

Imagine a simple scenario of having to look through 10 individual files manually to collect some needed information. Having the data in the files stored as records in database tables provides an efficient way to manipulate and analyze the data for information. Based on the generalized capabilities desired for a system as described above, we will look into system, design, and functional requirements for the ISO Analyzer and Reporting System.

The ISO Analyzer and Reporting System uses a SQL server database, which is a relational database, version 2008 R2, to create databases and tables, and to run queries. The ISO Analyzer and Reporting System is developed and builds on .Net framework using Microsoft Visual Studio C#, version 2010. These Technologies are discussed in Chapter 2. In order for the program to successfully run on a machine, this software must be installed.

Given that the system requirements are all met, upon running the ISO Analyzer and Reporting System, users can interact with the system using the provided GUI. Below are some of the high-level user interface design requirements for the ISO Analyzer and Reporting System.

1. Users shall be able to select or deselect load or pricing data files to download for a selected date range.
2. Users shall be able to upload data from the downloaded files into SQL server tables.
3. Users shall be able to select (a) database table(s) from which to query.
4. Users shall be able to generate the reports using data in the database table(s).
5. Users shall be informed of any errors in their selection. For example, the system should notify users if there are no files available for download based on their selection.
6. Users shall be notified when a download operation or an upload operation has completed.
7. Users shall be provided with helpful hints to navigate through the system.

Next, we will look into the desired functionalities for the ISO Analyzer and Reporting System. The core system functionalities are downloading files from the NYISO website, uploading data in the files into SQL server tables, and querying the tables to obtain the desired information and to generate reports. Below is a list of high-level functional requirements for the system.

1. Users shall be able to select (a) file(s) for a certain date or a range of dates to download.
2. The selected files shall be downloaded to the user's local machine.
3. The ISO Analyzer and Reporting System shall create SQL server database tables and store data from the downloaded files in the tables.
4. Based on the user's selection, the ISO Analyzer and Reporting System shall dynamically create certain custom queries for a zonal load commitment forecast; integrated real-time actual load under load data; and day-ahead market zonal and integrated, real-time market zonal under pricing data.
5. The result of the query shall be in CSV format, which can be opened in Excel for further manipulation.
6. Based on the user's selection, the ISO Analyzer and Reporting System shall dynamically create reports for the zonal load commitment forecast; integrated, real-time actual load under load data; and day-ahead market zonal and integrated, real-time market zonal under pricing data.
7. The reports shall be a graphical representation in a new window.
8. Users shall be notified if there is an error. For example, the system should display an error if the data are not available for the user's selection.

Chapter 2 contains the Literature Review for the various technologies and methods used in developing the ISO Analyzer and Reporting System. In Chapter 3, the ISO Analyzer and Reporting System is discussed in detail. Discussed are its interface design, use cases, and system functionalities. Chapter 4 demonstrates the use of ISO Analyzer and reporting system based on a use case and the results that were achieved. Chapter 5 explains the limitations and potential future work involving the system. The References provided in the last section conclude this paper.

2. LITERATURE REVIEW

The ISO Analyzer and Reporting System provides a graphical user interface (GUI) to interact with the NYISO website to download selected load and pricing data and to upload them into SQL server tables. Also, utilizing the GUI users can interact with SQL server database to run queries and to generate reports. This GUI is also clears data from the database for the selected date range. The technology background relevant to the above process and the development of the ISO Analyzer and Reporting System are discussed in this chapter.

2.1. Microsoft Visual Studio and C#.NET

Microsoft Visual Studio is an integrated development environment (IDE) that provides a platform for .NET application development. It aids developers in producing quality code by providing built-in tools and controls, such as buttons, text boxes, labels, etc., and features, such as intelligence. It also provides a debugging capability.

C# is a programming language based on an object-oriented design and was mainly created to be used in .NET development [3]. Development in .NET supports using different programming languages, such as J#, Visual Basic, C++, C#, etc. [4]. C# is made possible by compiling programs written in different languages to a common Microsoft Intermediate Language (IL) [5].

IL does not contain any machine codes native to the computer. Another compilation is needed to convert IL to machine code using a just-in-time compiler (JITer) [6]. Any .NET development requires using the .NET Framework, which provides a large collection of built-in reusable classes known as .NET Framework Classes [7]. Common Language Runtime (CLR) is the core of .NET technology. CLR is the runtime execution environment that is responsible for thread execution, memory allocation, just-in-time compilation, etc. [8]. Also, ASP.NET and XML support allows easier data importing and exporting, and ADO.NET support allows the capability to work with databases [9].

The ISO Analyzer and Reporting System is created using the 2010 version of Microsoft Visual Studio. The database it works with is a SQL server database, which we will discuss next.

2.2. SQL Server and Structured Query Language

The SQL server is a popular relational database management system developed by Microsoft. The SQL in SQL server stands for Structured Query Language. As a database, the primary function is to store and retrieve data as requested by the users and other software applications. The SQL server can be downloaded from the Microsoft website and installed on a computer as a server. The ISO Analyzer and Reporting System uses SQL Server 2008 R2 to create and store databases and tables.

SQL is a query language to manipulate and query data in a relational database. SQL queries can contain statements, clauses, expressions, and predicates necessary to produce the desired result [11] [12]. For example, in the SQL statement below we have SELECT, GROUP BY,and HAVING clauses; *SUM (salary) < 5000*is the predicate to filter the result, and 5000is the expression.

```
SELECT name, SUM (salary)
FROM Employee GROUP BY EmpName HAVING SUM (salary) <5000
```

2.3. New York Independent System Operator

The New York Independent System Operator (NYISO) administers New York state's energy needs and operates the state's electricity grid and wholesale electricity markets. It covers nearly 11,000 miles of high-voltage transmission lines from 500 generators. The load, transmission, and electricity cost is monitored and recorded 24 hours a day, 7 days a week. Utility companies obtain electricity to distribute to their consumers through NYISO's wholesale electricity market. The minimum purchase necessary is one megawatt, and it can supply power to nearly 1,000 homes. There are 11 energy pricing zones in NYISO as shown in Figure 1 [10].

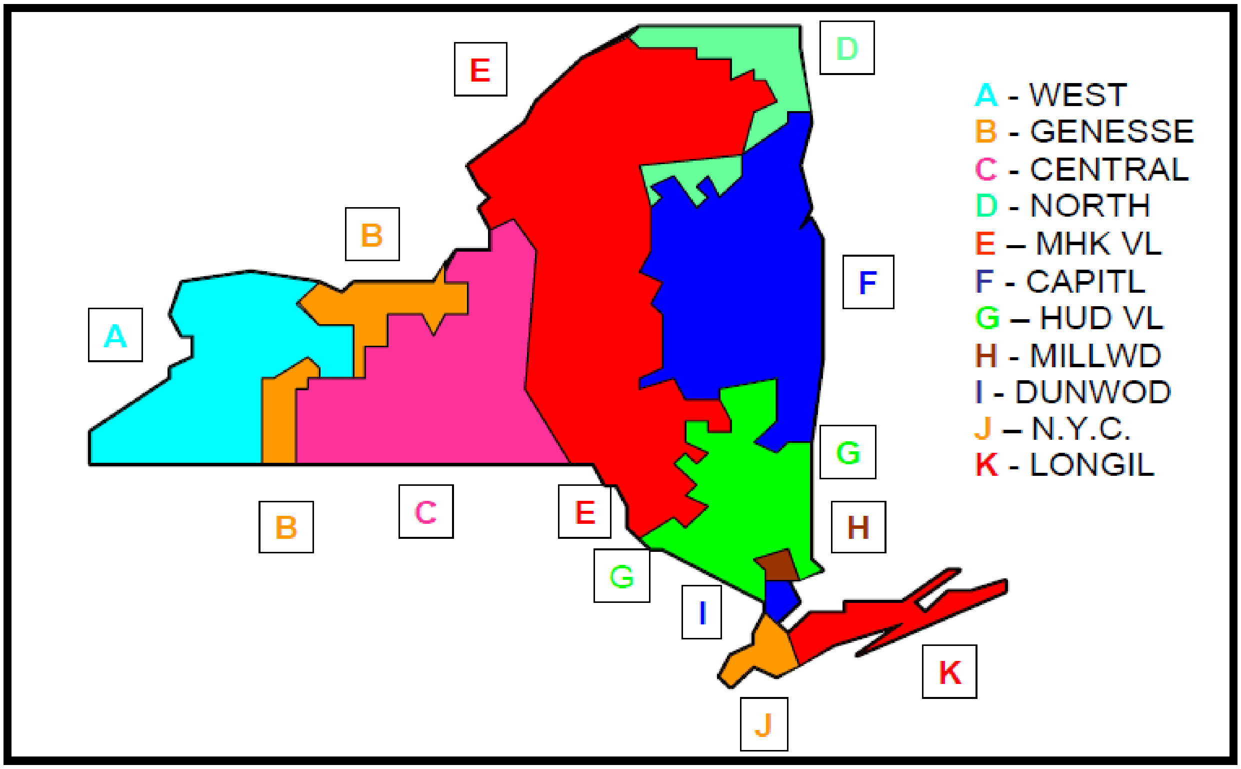


Figure 1. LBMP Zones

Some of NYISO's key terminologies are discussed below.

2.3.1. Locational-Based Marginal Price

One of the components of wholesale electricity prices is the energy cost known as the Locational Based Marginal Price (LBMP). LBMP can be calculated by the cost to produce energy plus the cost to transport that energy[11].

2.3.2. Day-Ahead and Real-Time Electricity Markets

NYISO operates a day-ahead and a real-time market [10]. Day-ahead market data provide the next day's operating schedule for generators. By providing an energy forecast in combination with the cost of electricity forecast, the generators are financially motivated to operate as scheduled. However, there can be certain factors that influence energy generation, such as system availability, condition, etc., which can alter projected load, so NYISO runs a real-time market to account for the changes and to maintain financial balance.

In the next chapter, we will discuss the ISO Analyzer and Reporting System in detail. We will look into the system functionalities and the methods used to implement those functionalities.

3. ISO ANALYZER AND REPORTING SYSTEM

NYISO offers online access to its electricity market and operation data to the general public and its market participants. The electricity market can be a day-ahead market, real-time market, hour-ahead market, etc. These markets capture energy pricing and energy schedules for generators based on the market type. This data is monitored and recorded throughout the 24-hour period, every day according to timestamps, and are grouped by energy pricing zones or some other criteria. The data are made available in different file formats, such as Excel, zip, or comma-separated values (CSV) format.

The ISO Analyzer and Reporting System allow users to interact with the NYISO website through its graphical user interface. Users can select the files for the energy markets and download them to their local machines. The system then stores the data in the files as tables in Pricing Data or Load Data SQL Server databases which can be queried to obtain the desired information and also to generate reports. The result is in CSV format for queries and graphical representation for reports. The ISO Analyzer and Reporting System also have a database maintenance piece for regular database maintenance. Figure 2 explains how the user is interacting with the ISO Analyzer and Reporting System and with the NYISO website. It illustrates the basic idea of the ISO Analyzer and Reporting System.

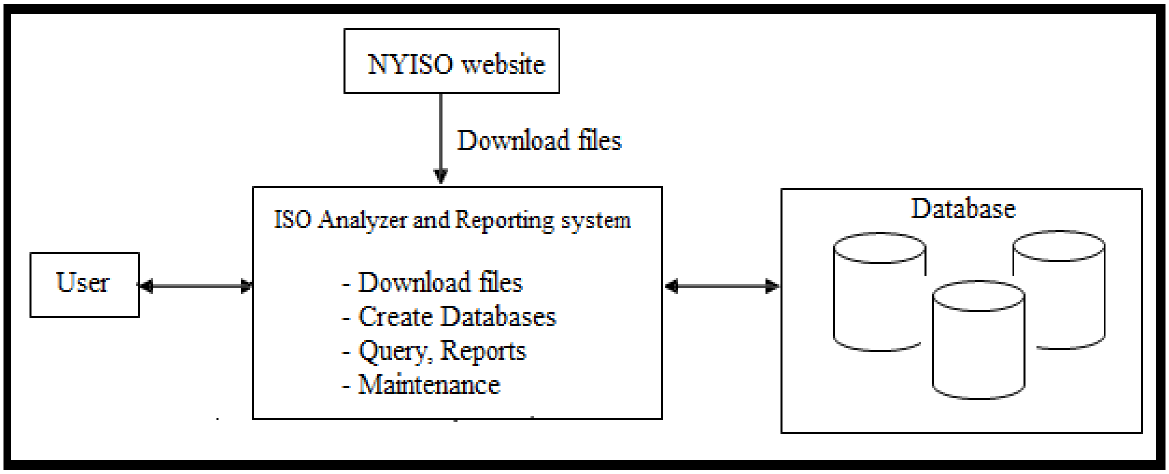


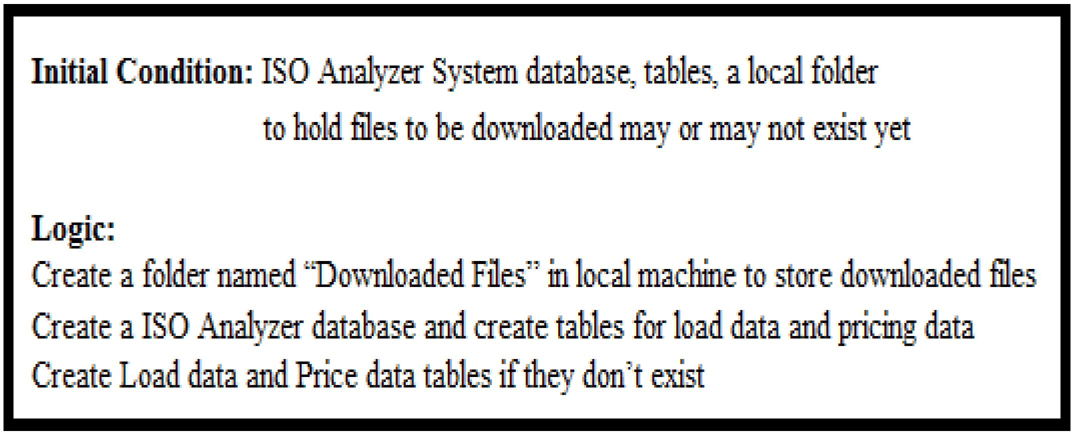
Figure 2. ISO Analyzer and Reporting System Overview

The system can be divided into seven different parts based on its functionalities. We will look at the core functionalities and discuss what methods were used to implement them.

1. Initialization
2. File Selection
3. Download
4. SQL Server Upload
5. Query
6. Reports
7. Maintenance

3.1. Initialization

First, we are running an SQL script to create an ISO analyzer database as well as ISO load data and pricing data tables. Upon running the ISO Analyzer and Reporting System, a user can select and download certain types of files under the load data tab and the pricing data tab. To store data from these files, SQL server database tables are already available in the database for each file type. For example, the NY Load Forecast file may have certain columns for a selected date range; the data for this table are stored in the dbo.isolf table in the database. More details will be discussed in the download files section. Figure 3 describes the logic during initialization.



Initial Condition: ISO Analyzer System database, tables, a local folder
to hold files to be downloaded may or may not exist yet

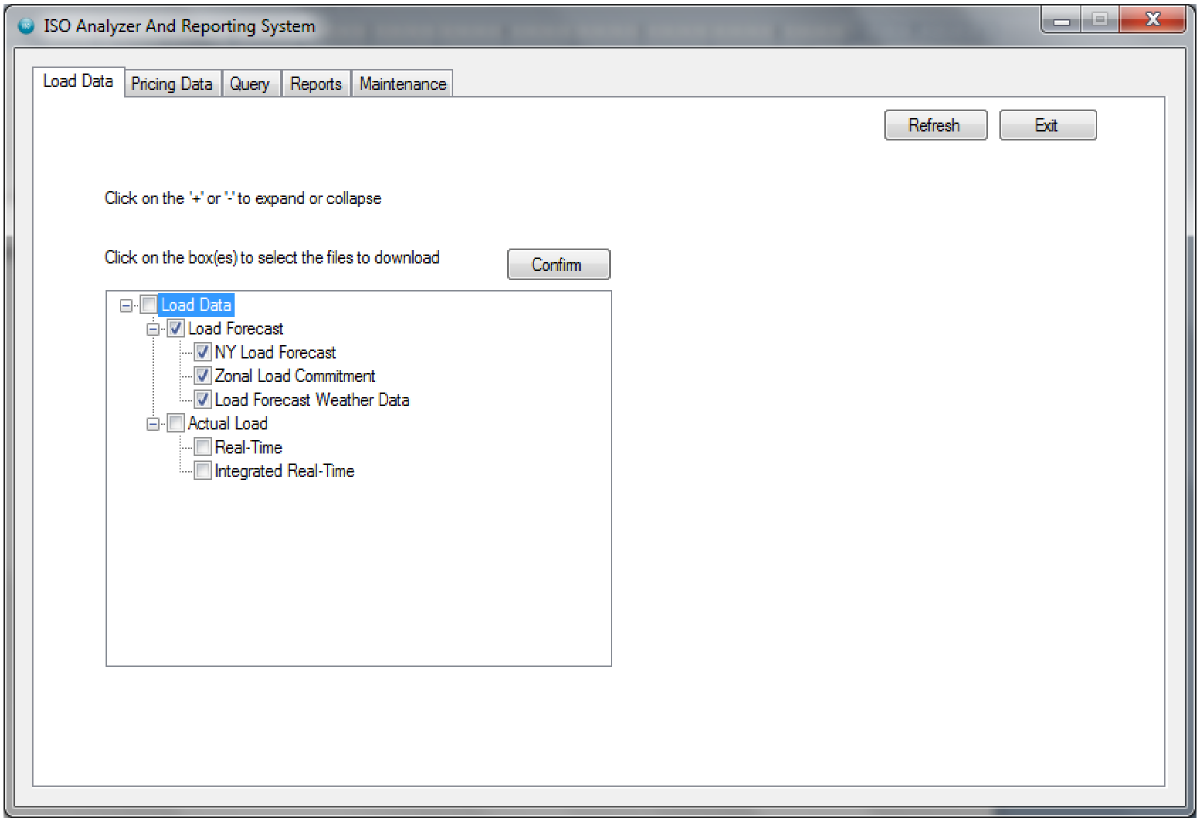
Logic:
Create a folder named "Downloaded Files" in local machine to store downloaded files
Create a ISO Analyzer database and create tables for load data and pricing data
Create Load data and Price data tables if they don't exist

Figure 3. Initialization

Figure 4 shows a screenshot the ISO Analyzer and Reporting System with the Load Data tab selected. Users can click the checkboxes and press the Confirm button to select the files they want to download for querying. Here, a user wants to download the Load Forecast data.

There are various options like NY Load Forecast, Zonal Load Commitment and Load Forecast Weather Data for the user to select from and then retrieve the results. At any point of time the user can click on the refresh button on the right top corner of the interface to see the refreshed view of results instead of clicking on confirm multiple times. The interface provides an option of Real-time and Integrated Real-time which are both classified under Actual Load.

The interface provides detailed and tabbed menus for user to have easy navigation from one tab to another. There are five tabs that this interface provides that can be used to generate multiple reports and download them.



Click on the '+' or '-' to expand or collapse

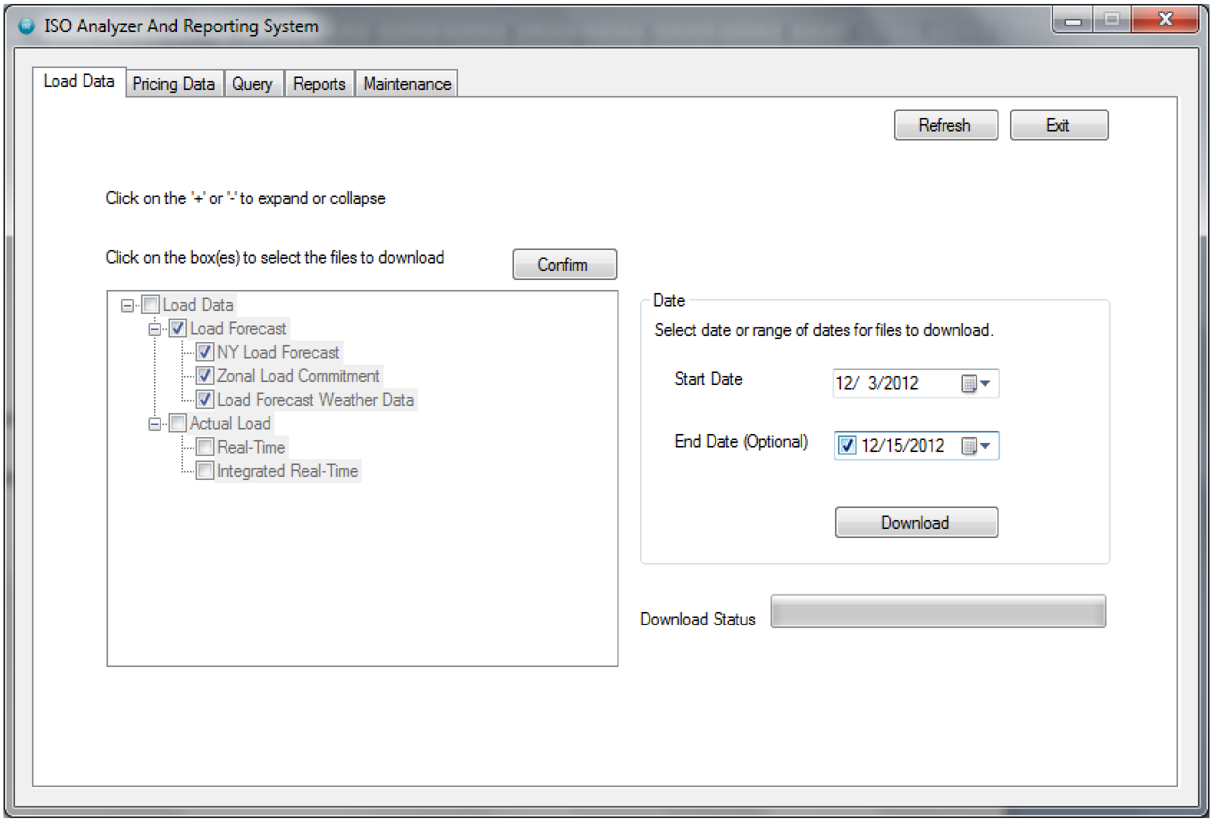
Click on the box(es) to select the files to download

- □ Load Data
  - ☑ Load Forecast
    - ☑ NY Load Forecast
    - ☑ Zonal Load Commitment
    - ☑ Load Forecast Weather Data
  - □ Actual Load
    - □ Real-Time
    - □ Integrated Real-Time

Figure 4. ISO Analyzer and Reporting System Screenshot 1

3.2. File Selection

The ISO Analyzer and Reporting Sytem's scope is limited to NYISO's files for load data and pricing data. Because NYISO maintains the files by date, the ISO Analyzer and Reporting Sytem allow users to select files for a particular date or a range of dates. The date area is visible only after the user chooses at least one file by selecting the checkbox next to it and pressing the Confirm button to confirm the files to download. When user select dates system verifies the date range and display an error message if the selected date is out of range. User can also use refresh button at the bottom to change the date selection. Figure 5 shows the files selected for certain dates.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- □ Load Data
  - ☑ Load Forecast
    - ☑ NY Load Forecast
    - ☑ Zonal Load Commitment
    - ☑ Load Forecast Weather Data
  - □ Actual Load
    - □ Real-Time
    - □ Integrated Real-Time

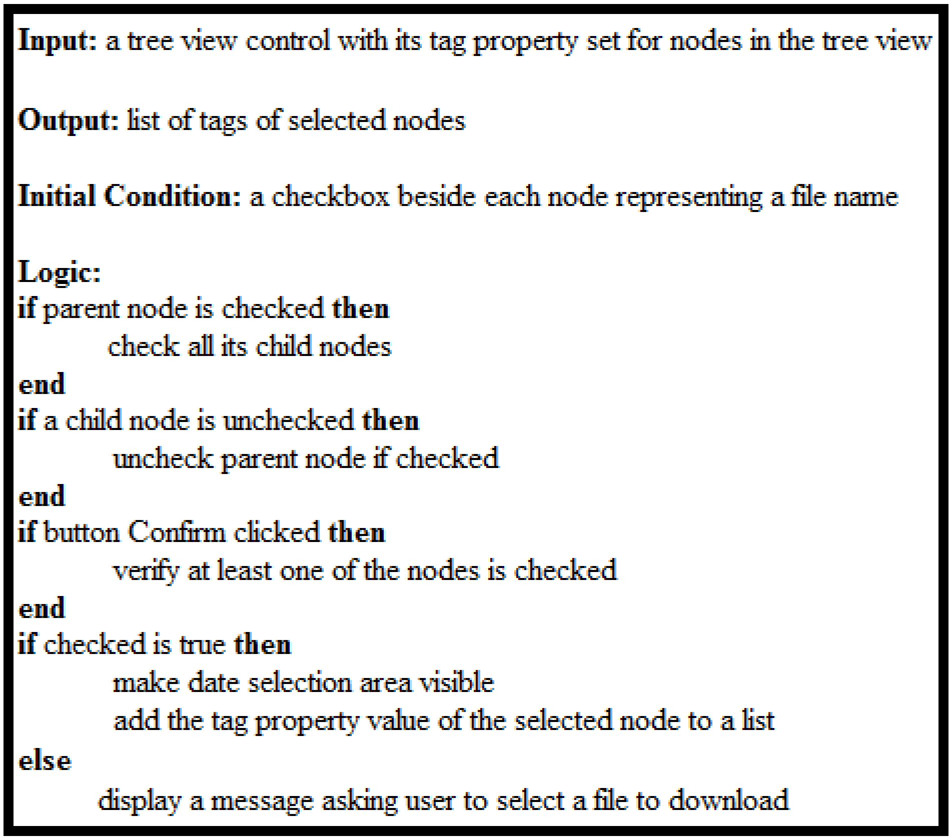
Select date or range of dates for files to download.

Start Date: 12/ 3/2012

End Date (Optional): ☑ 12/15/2012

Figure 5. ISO Analyzer and Reporting System Screenshot 2

Each file denoted as a node in the tree view control has a tag property which is used to identify the type of file and is later used to create the name of the file to download. For example, Zonal Load Commitment has the tag property value of “zonalBidLoad.” This tag property decides which template table to use when creating a table to hold data from the file. Also, using the tree view control to display different files makes it easy for users to make file selections. One can just click on the parent node, and all its child nodes would be automatically selected. When the user confirms the selected files, the tree view control becomes read only, and the date-selection area becomes visible. Clicking the Refresh button allows a new file selection. The Exit button can be used to close the application. Figure 6 is an algorithm of what happens during file selection.



Input: a tree view control with its tag property set for nodes in the tree view

Output: list of tags of selected nodes

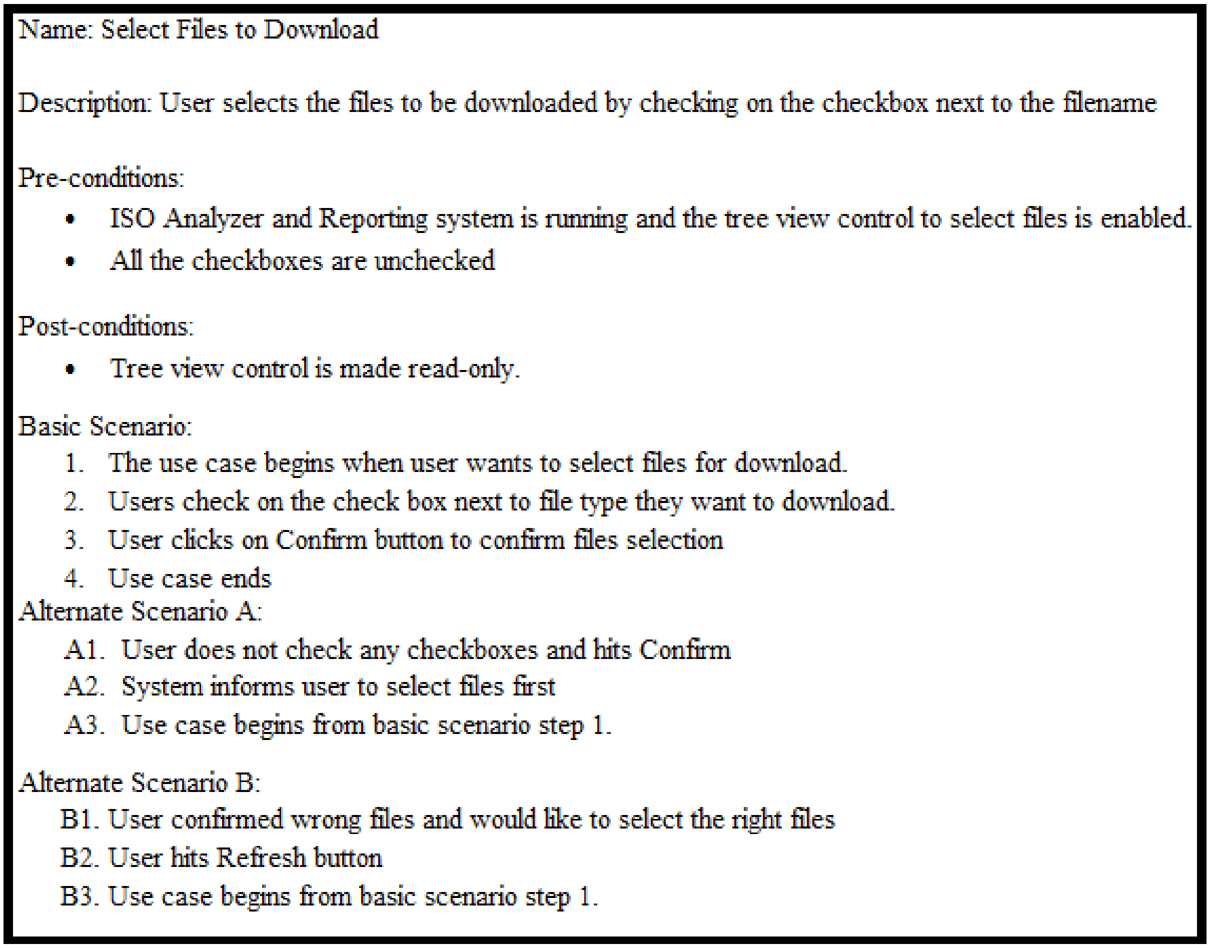
Initial Condition: a checkbox beside each node representing a file name

Logic:

```
if parent node is checked then
        check all its child nodes
end
if a child node is unchecked then
        uncheck parent node if checked
end
if button Confirm clicked then
        verify at least one of the nodes is checked
end
if checked is true then
        make date selection area visible
        add the tag property value of the selected node to a list
else
        display a message asking user to select a file to download
```

Figure 6. File Selection

A user has to select files to download for generating reports before the actual download process can be executed. Users can select the files to download by selecting the checkboxes next to the file names in the tree view control provided in the ISO Analyzer and Reporting System’s GUI. Figure 7 is a use case for selecting files to download.



Name: Select Files to Download

Description: User selects the files to be downloaded by checking on the checkbox next to the filename

Pre-conditions:

- ISO Analyzer and Reporting system is running and the tree view control to select files is enabled.
- All the checkboxes are unchecked

Post-conditions:

- Tree view control is made read-only.

Basic Scenario:

1. The use case begins when user wants to select files for download.
2. Users check on the check box next to file type they want to download.
3. User clicks on Confirm button to confirm files selection
4. Use case ends

Alternate Scenario A:

A1. User does not check any checkboxes and hits Confirm
A2. System informs user to select files first
A3. Use case begins from basic scenario step 1.

Alternate Scenario B:

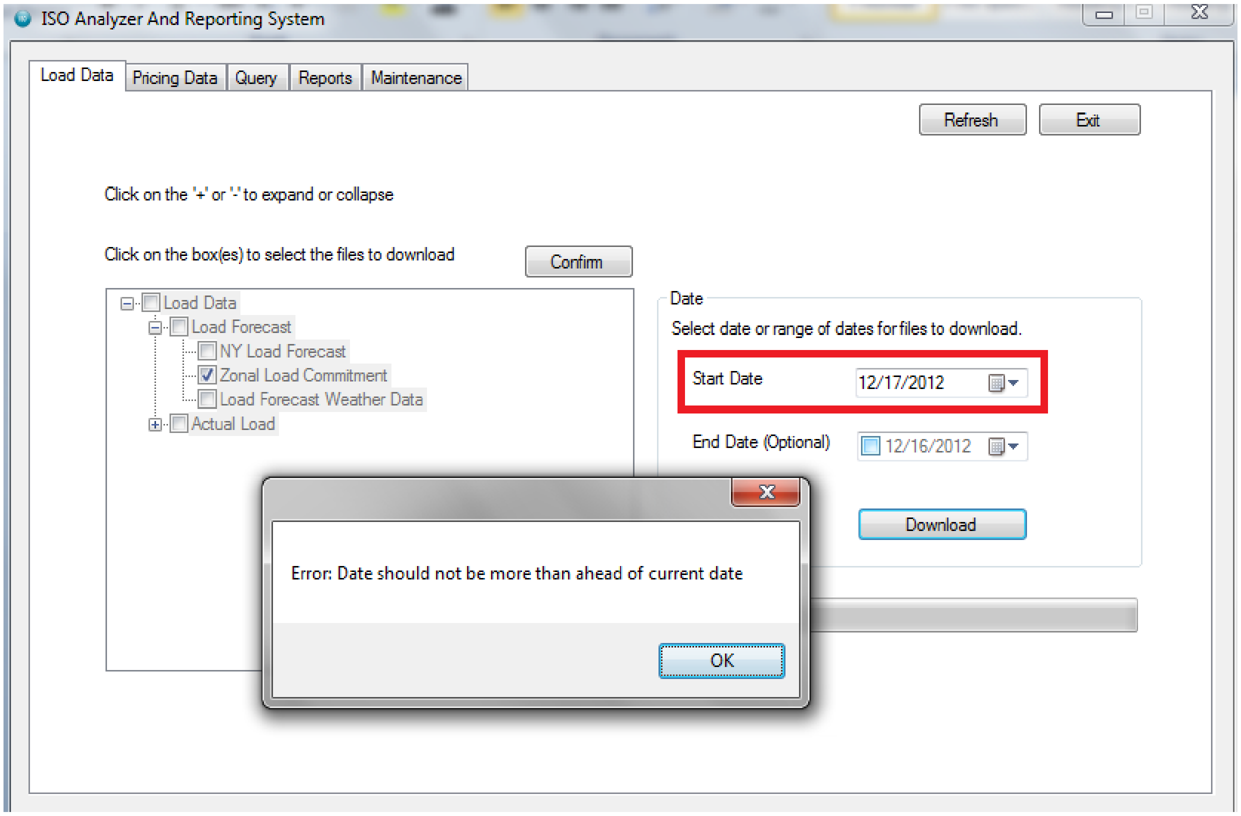
B1. User confirmed wrong files and would like to select the right files
B2. User hits Refresh button
B3. Use case begins from basic scenario step 1.

Figure 7. File Selection Use Case

3.3. Download

After a user selects the files to be downloaded for certain date or range of dates, pressing the Download button first validates the dates. The system checks to see if the files selected are available for download from NYISO for those dates. If a file is not available, the user is notified.

In Figure 8, the user has selected to download the zonal load for the Load Forecast for December 17, 2012, i.e., one day ahead of the current date. The system displays the error message shown in the figure 8.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- □ Load Data
  - □ Load Forecast
    - □ NY Load Forecast
    - ☑ Zonal Load Commitment
    - □ Load Forecast Weather Data
  - □ Actual Load

Select date or range of dates for files to download.

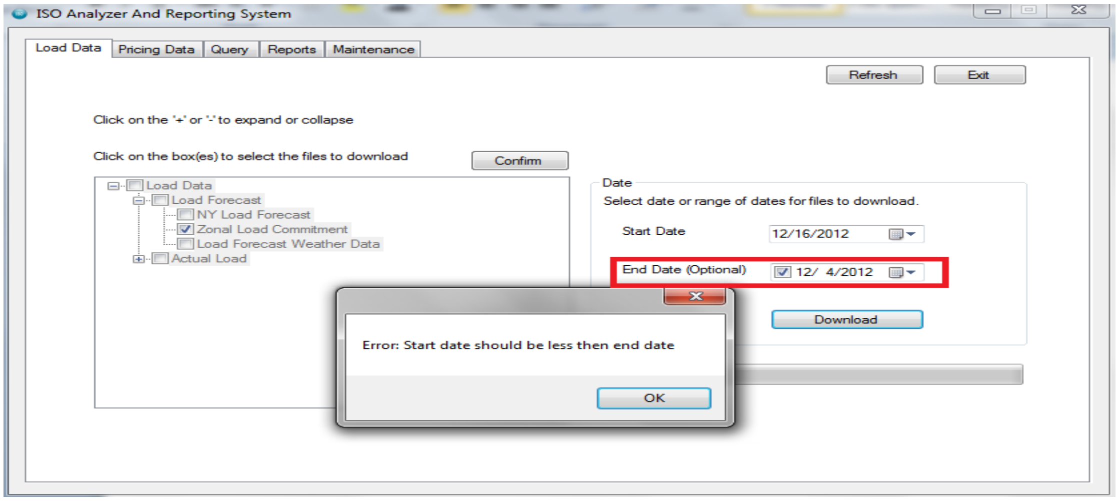
Start Date 12/17/2012

End Date (Optional) □ 12/16/2012

Error: Date should not be more than ahead of current date

Figure 8. ISO Analyzer and Reporting System Screenshot 3

In Figure 9, the user selects an end date that is before the current date to download the files. The system displays the error message shown in the figure.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- □ Load Data
  - □ Load Forecast
    - □ NY Load Forecast
    - ☑ Zonal Load Commitment
    - □ Load Forecast Weather Data
  - □ Actual Load

Select date or range of dates for files to download.

Start Date 12/16/2012

End Date (Optional) ☑ 12/ 4/2012

Error: Start date should be less then end date

Figure 9. ISO Analyzer and Reporting System Screenshot 4

When a user selects the files to download for certain dates, the ISO Analyzer and Reporting System downloads the files to a local system with the names by concatenating the date and the tag property value of the selected node in the tree view. These names are later used to enter the data into SQL server database tables. For each individual file, the data are appended into the respected table for the same file type. For example, if a user selects the NY Load Forecast file for November 22, 2012, the tag property value of the node representing the NY Load Forecast is "isolf." Therefore, the downloaded file name would be 20121122isolf. The resource file for this particular file on the NYISO's website would have the name, 20121122isolf.csv.

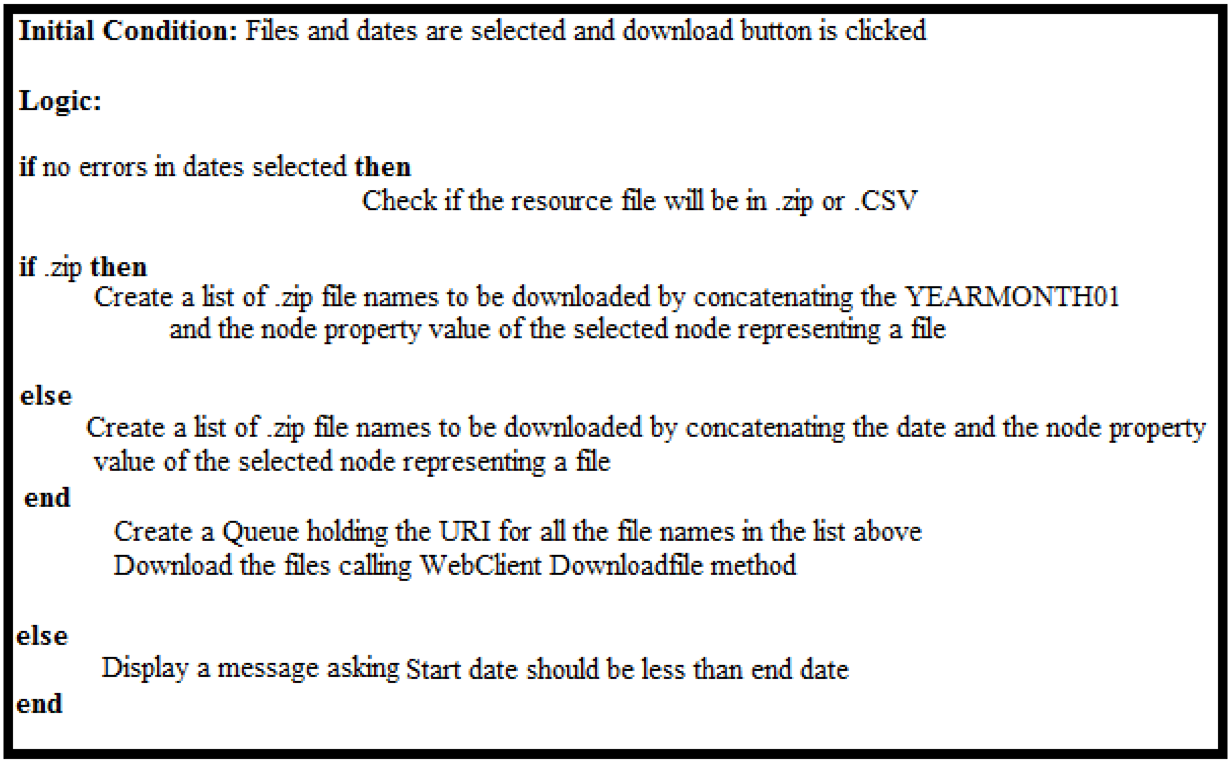
The data for this file are stored in the dbo.isolf table. If you downloaded the same file type for November 15, 2012, the data are appended into the dbo.isolf table in the database.

To download files from web, the Uniform Resource Indicator (URI) should include the exact file name to be downloaded. The NYISO keeps recent files dating 8 or 9 days back from the current date in CSV format and older files in a zipped format. For example, if today is October 4, 2012, then the Zonal Load Commitment file for October 12, 2012, is named 20121001zonalBidLoad_csv. A similar file for March 3, 2012, would be in a zip file named 20120301zonalBidLoad.zip. This zip file would also contain all files for that particular month.

Files are downloaded using the WebClient class found in the System.Net namespace [16]. The ISO Analyzer and Reporting System call its DownloadFile method which downloads the files without halting the calling thread. This feature enables users to interact with the GUI while files are being downloaded in the backend. While creating any GUI application, if there is a thread that requires more time and resources to execute, usually, the task is accomplished by using the Background Worker class which makes the GUI responsive at all times through multi-threading [17]. Here, is a signature of the method.

WebClient.DownloadFile(URI address, string filename)

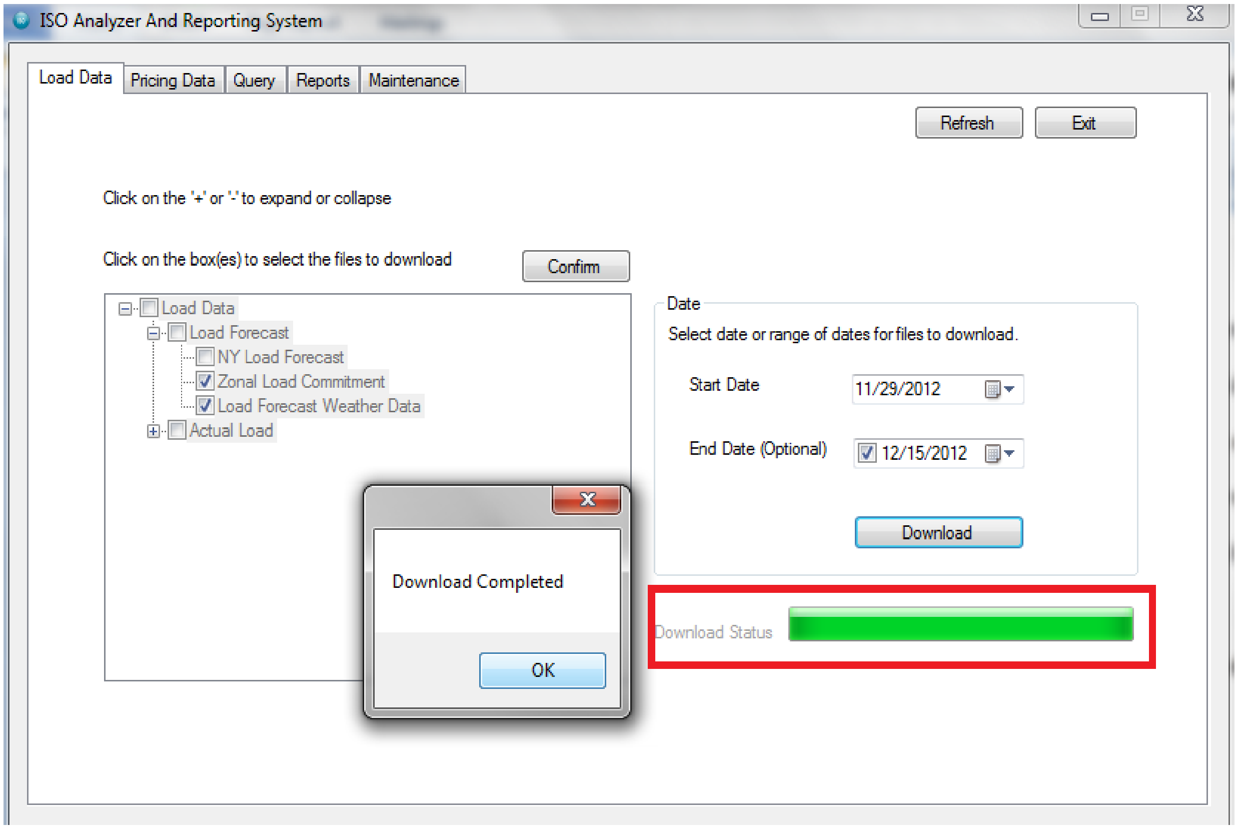
The ISO Analyzer and Reporting System needs the files to be downloaded to perform its query operations, so instead of Background Worker, an alternative was to use the WebClient class DownloadFile method which, pretty much, accomplishes the same thing. The logic behind file-download process is presented in Figure 10.



```
Initial Condition: Files and dates are selected and download button is clicked

Logic:

if no errors in dates selected then
                              Check if the resource file will be in .zip or .CSV

if .zip then
      Create a list of .zip file names to be downloaded by concatenating the YEARMONTH01
            and the node property value of the selected node representing a file

else
     Create a list of .zip file names to be downloaded by concatenating the date and the node property
      value of the selected node representing a file
end
        Create a Queue holding the URI for all the file names in the list above
        Download the files calling WebClient Downloadfile method

else
       Display a message asking Start date should be less than end date
end
```

Figure 10. File Download

In Figure 11, the user has selected the Zonal Load Commitment and Load Forecast Weather Data files to download for dates ranging from November 29, 2012, to December 15, 2012. After the Download button is clicked and there are no errors, the actual downloading of the files starts as indicated by a progress bar that shows up near the lower-right corner. Once the download is successfully completed, the user is notified with a label that says "Download Complete," and the progress bar is fully filled in as shown in Figure 11.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- □ Load Data
  - □ Load Forecast
    - □ NY Load Forecast
    - ☑ Zonal Load Commitment
    - ☑ Load Forecast Weather Data
  - □ Actual Load

Select date or range of dates for files to download.

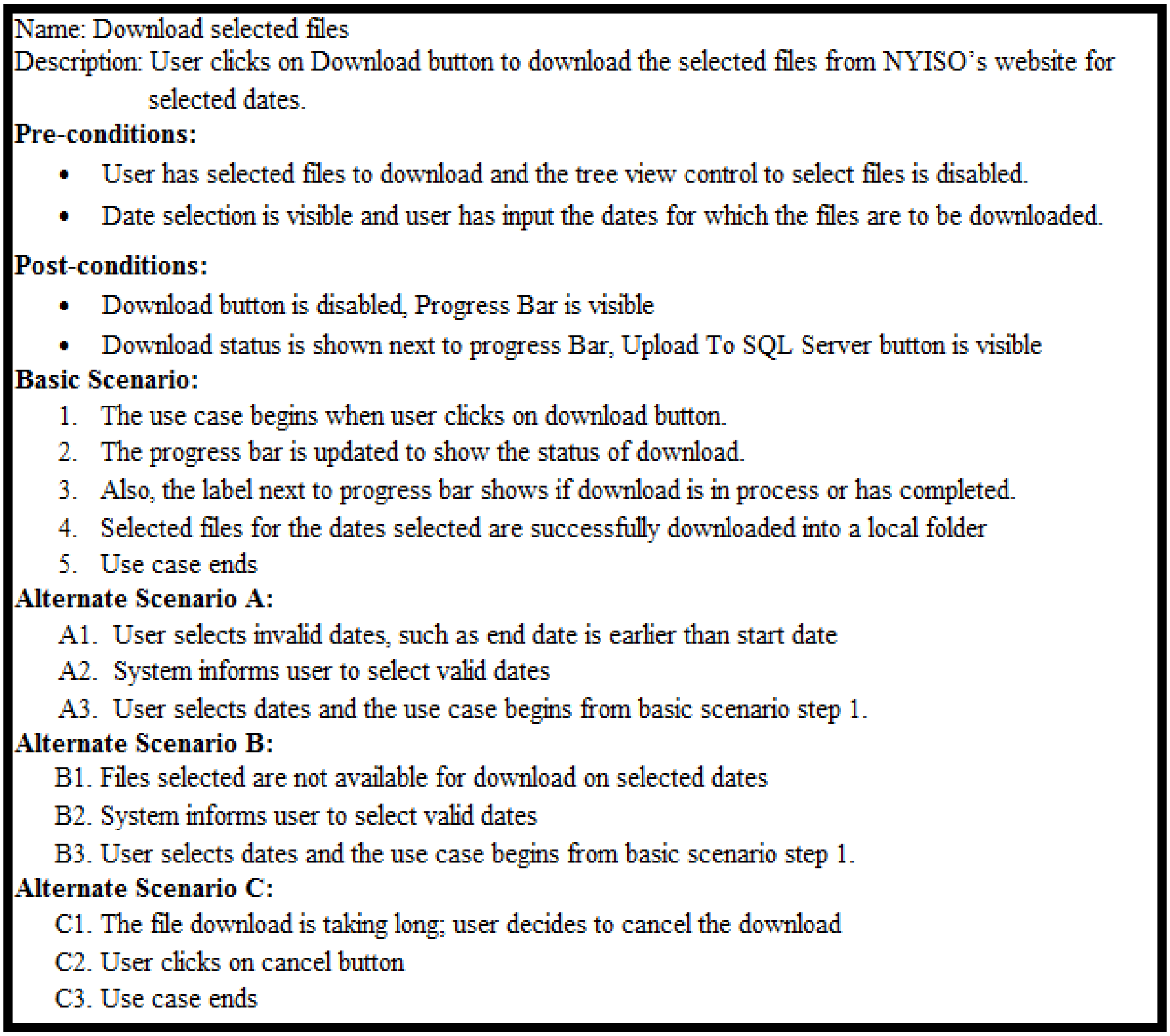
Start Date: 11/29/2012

End Date (Optional): ☑ 12/15/2012

Download Completed

Figure 11. ISO Analyzer and Reporting System Screenshot 5

While downloading, the following scenarios may occur. If there are no errors in the date selection and if files are available for the chosen dates, then the download process begins. If there are any errors, then the download process does not begin until all the errors have been corrected. In Figure 12, different use case scenarios for downloading are described.



Name: Download selected files

Description: User clicks on Download button to download the selected files from NYISO's website for selected dates.

Pre-conditions:

- User has selected files to download and the tree view control to select files is disabled.
- Date selection is visible and user has input the dates for which the files are to be downloaded.

Post-conditions:

- Download button is disabled, Progress Bar is visible
- Download status is shown next to progress Bar, Upload To SQL Server button is visible

Basic Scenario:

1. The use case begins when user clicks on download button.
2. The progress bar is updated to show the status of download.
3. Also, the label next to progress bar shows if download is in process or has completed.
4. Selected files for the dates selected are successfully downloaded into a local folder
5. Use case ends

Alternate Scenario A:

A1. User selects invalid dates, such as end date is earlier than start date
A2. System informs user to select valid dates
A3. User selects dates and the use case begins from basic scenario step 1.

Alternate Scenario B:

B1. Files selected are not available for download on selected dates
B2. System informs user to select valid dates
B3. User selects dates and the use case begins from basic scenario step 1.

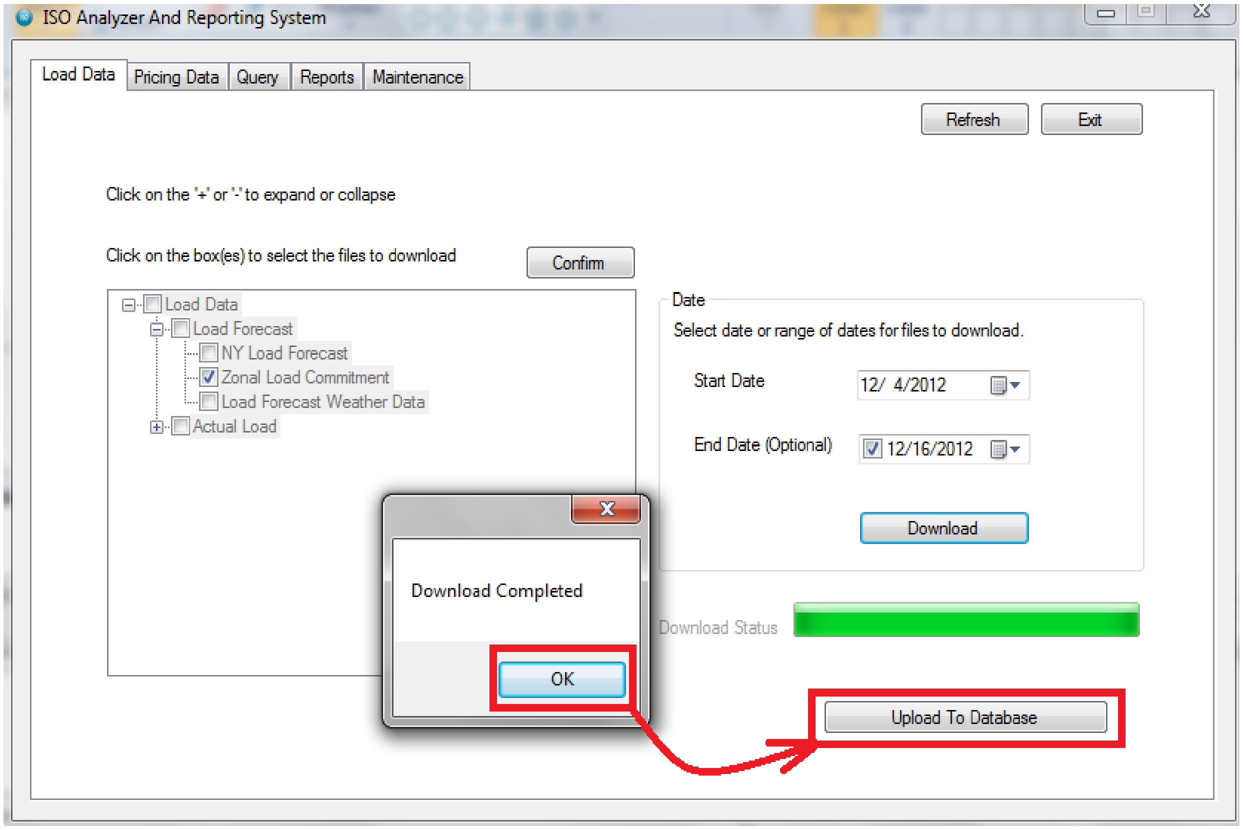
Alternate Scenario C:

C1. The file download is taking long; user decides to cancel the download
C2. User clicks on cancel button
C3. Use case ends

Figure 12. File Download Use Case

3.4. SQL Server Upload

After the files are successfully downloaded to the local folder named "DownloadedFiles," the ISO Analyzer and Reporting System lets the user know the download is complete using a label that says "Download Complete." Once the user clicks the OK button on "Download Complete" screen, the "Upload to Database" button should be enabled as shown in the Figure 13.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- ☐ Load Data
  - ☐ Load Forecast
    - ☐ NY Load Forecast
    - ☑ Zonal Load Commitment
    - ☐ Load Forecast Weather Data
  - ☐ Actual Load

Date

Select date or range of dates for files to download.

Start Date 12/ 4/2012

End Date (Optional) ☑ 12/16/2012

Download Completed

Figure 13. Upload Database: ISO Screenshot 6

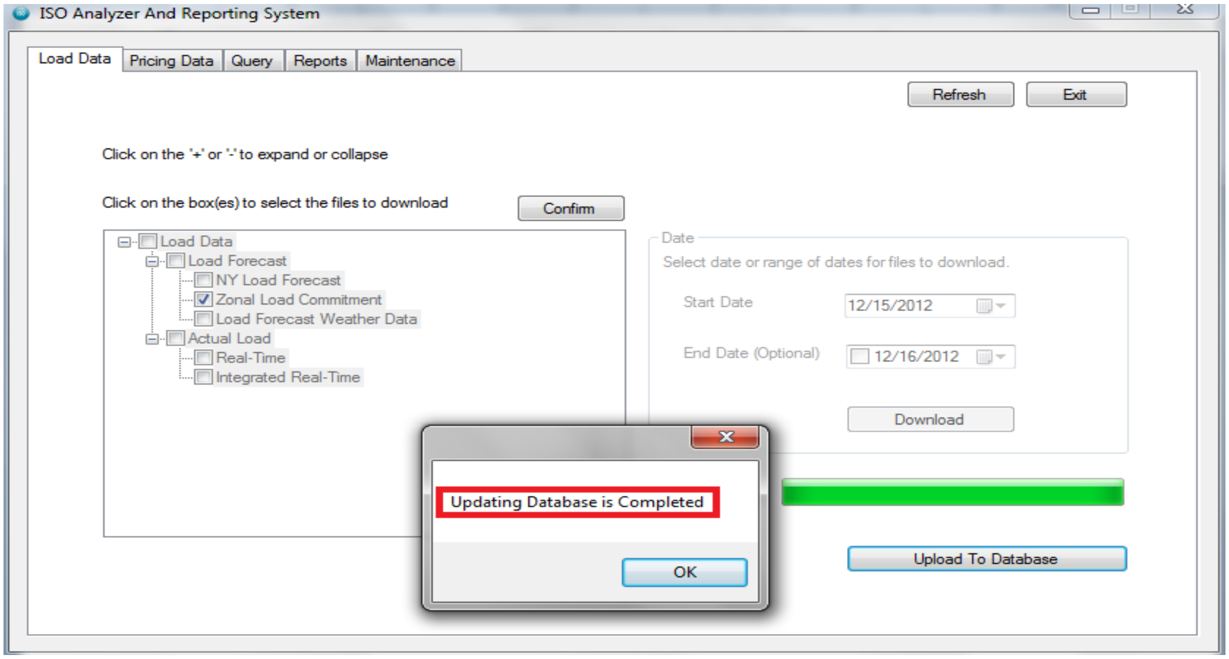
The next step is to upload data from the files to the SQL server database tables by clicking the "Upload To Database" button. As mentioned before, the data are stored in the respective table for the file type. For the downloaded zipped folders, all the files will be extracted into a folder, and the selected file(s) has(have) to be read.

The ISO Analyzer and Reporting System uses the Extractedzippedfile() method to extract files from the zipped folder, and this method reads only the files selected by a user. Data from these individual files are passed to the SQL server database tables.

For example, if a user downloads a Zonal Load Commitment data file for May 14, 2012, there will be a file named 20120501zonalBidLoad_csv.zip, the file name of the resource file on the NYISO's website. Because we only want data for May 14th, the ISO

Analyzer and Reporting System uses the Extractedzippedfile to extract the particular file named 20120514zonalBidLoad.csv. The tag property value of the node representing Zonal Load Commitment is “zonalBidLoad,” so the downloaded file name created for the .csv file is 20120514zonalBidLoad. The data are stored in the dbo.ZonalBidLoad table.

In Figure 14, the Zonal Load Commitment file for December 15, 2012, has already been downloaded to the user’s local machine, and the user has clicked the “Upload To Database” button. After clicking the button, the ISO Analyzer and Reporting System uploads the data into the dbo.ZonalBidLoad table. After the data are loaded into the database table, user is notified that the “Updating Database is completed” as shown in Figure 14.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- □ Load Data
  - □ Load Forecast
    - □ NY Load Forecast
    - ☑ Zonal Load Commitment
    - □ Load Forecast Weather Data
  - □ Actual Load
    - □ Real-Time
    - □ Integrated Real-Time

Date

Select date or range of dates for files to download.

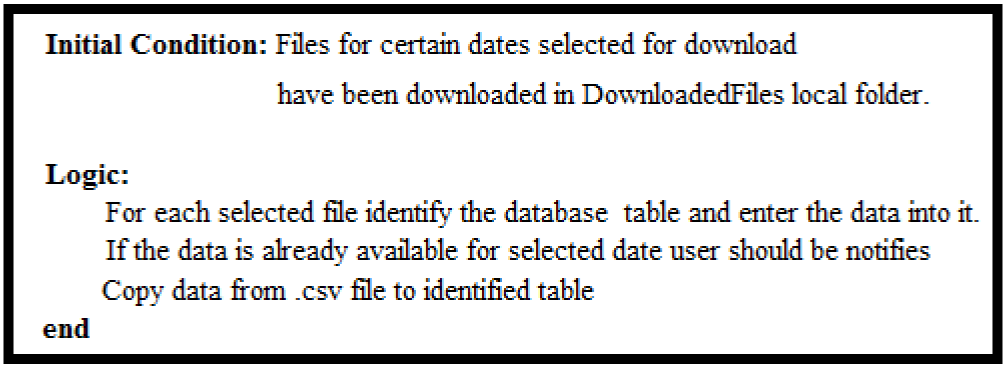
Start Date 12/15/2012

End Date (Optional) □ 12/16/2012

Updating Database is Completed

Figure 14. Uploading Files to the Database: ISO Screenshot 7

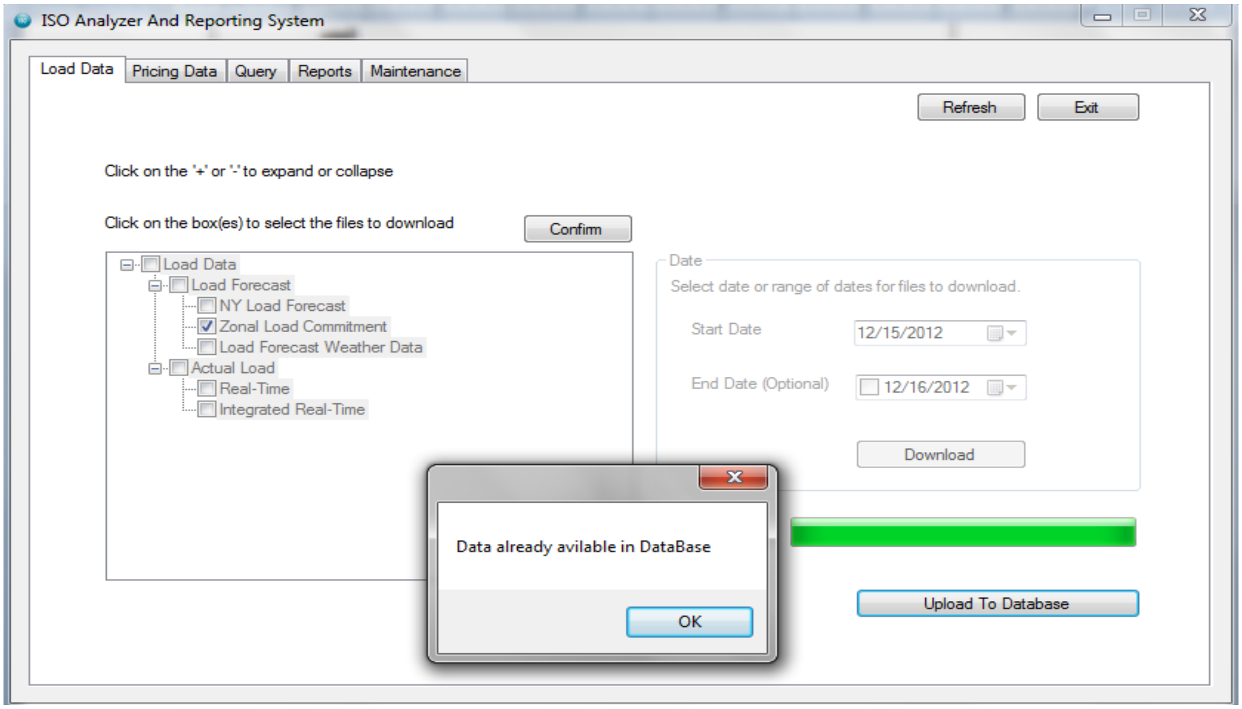
Data can be uploaded into SQL server tables following an immediate download provided the files download and upload into tables by using the same running instance of the ISO Analyzer and Reporting System. Figure 15 describes the logic for creating and uploading tables into an SQL server database.



```
Initial Condition: Files for certain dates selected for download
                   have been downloaded in DownloadedFiles local folder.

Logic:
      For each selected file identify the database  table and enter the data into it.
      If the data is already available for selected date user should be notifies
      Copy data from .csv file to identified table
end
```

Figure 15. SQL Server Upload

Because we are uploading data into the same table, the uploading process will notify the user if the data are already available for the selected file for the selected date. In Figure 16, the user is trying to upload the Zonal Load Commitment data for December 15, 2012. Because the data are already available for this file, the system displays an error message as shown in Figure 16.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- □ Load Data
  - □ Load Forecast
    - □ NY Load Forecast
    - ☑ Zonal Load Commitment
    - □ Load Forecast Weather Data
  - □ Actual Load
    - □ Real-Time
    - □ Integrated Real-Time

Date

Select date or range of dates for files to download.

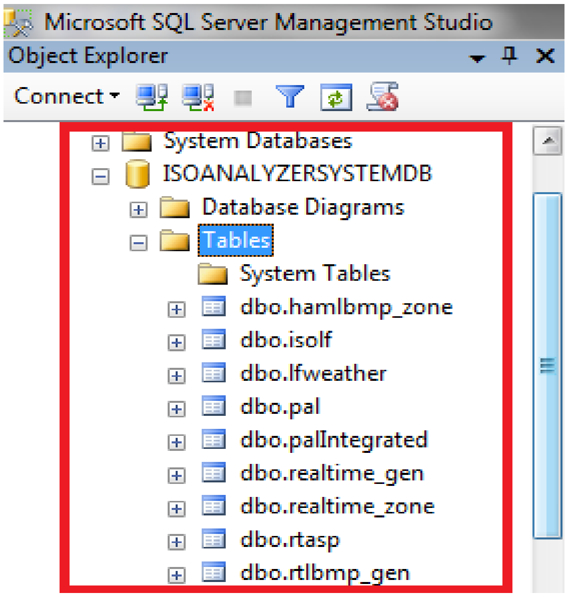
Start Date: 12/15/2012

End Date (Optional): □ 12/16/2012

Data already avilable in DataBase

Figure 16. SQL Server Upload error ISO Screenshot 8

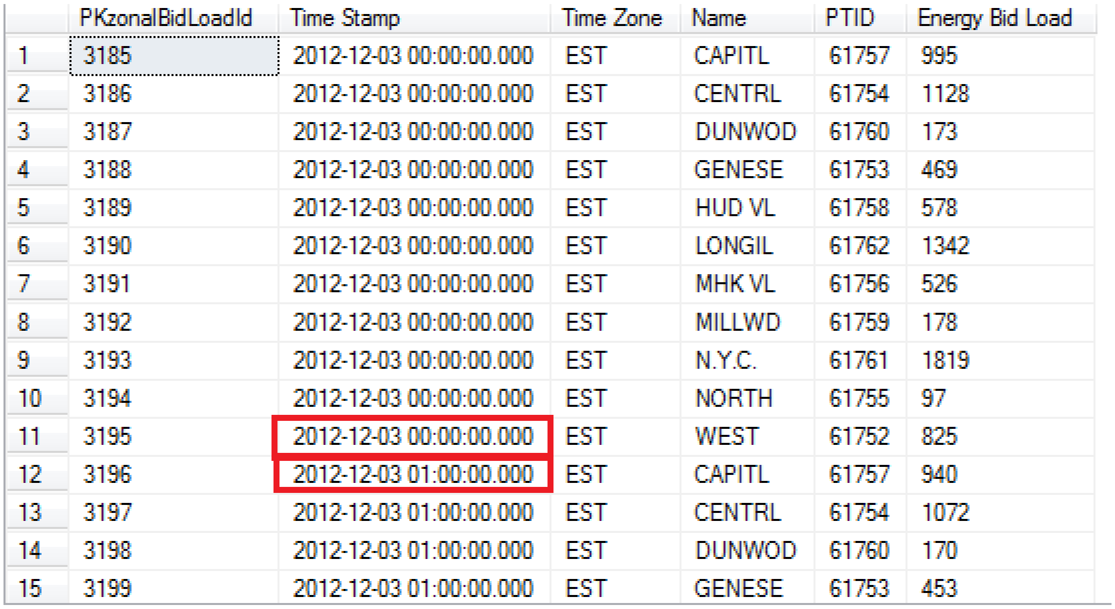
Figure 17 is a screenshot of tables uploaded to the SQL server database. For example, to view the tables uploaded into the SQL server database, SQL Server Management Studio 2008 IDE's Server Explorer is used to establish a data connection to the ISO Analyzer SQL Server database.



- System Databases
- ISOANALYZERSYSTEMDB
  - Database Diagrams
  - Tables
    - System Tables
    - dbo.hamlbmp_zone
    - dbo.isolf
    - dbo.lfweather
    - dbo.pal
    - dbo.palIntegrated
    - dbo.realtime_gen
    - dbo.realtime_zone
    - dbo.rtasp
    - dbo.rtlbmp_gen

Figure 17. Tables in the ISO ANALYZER Database

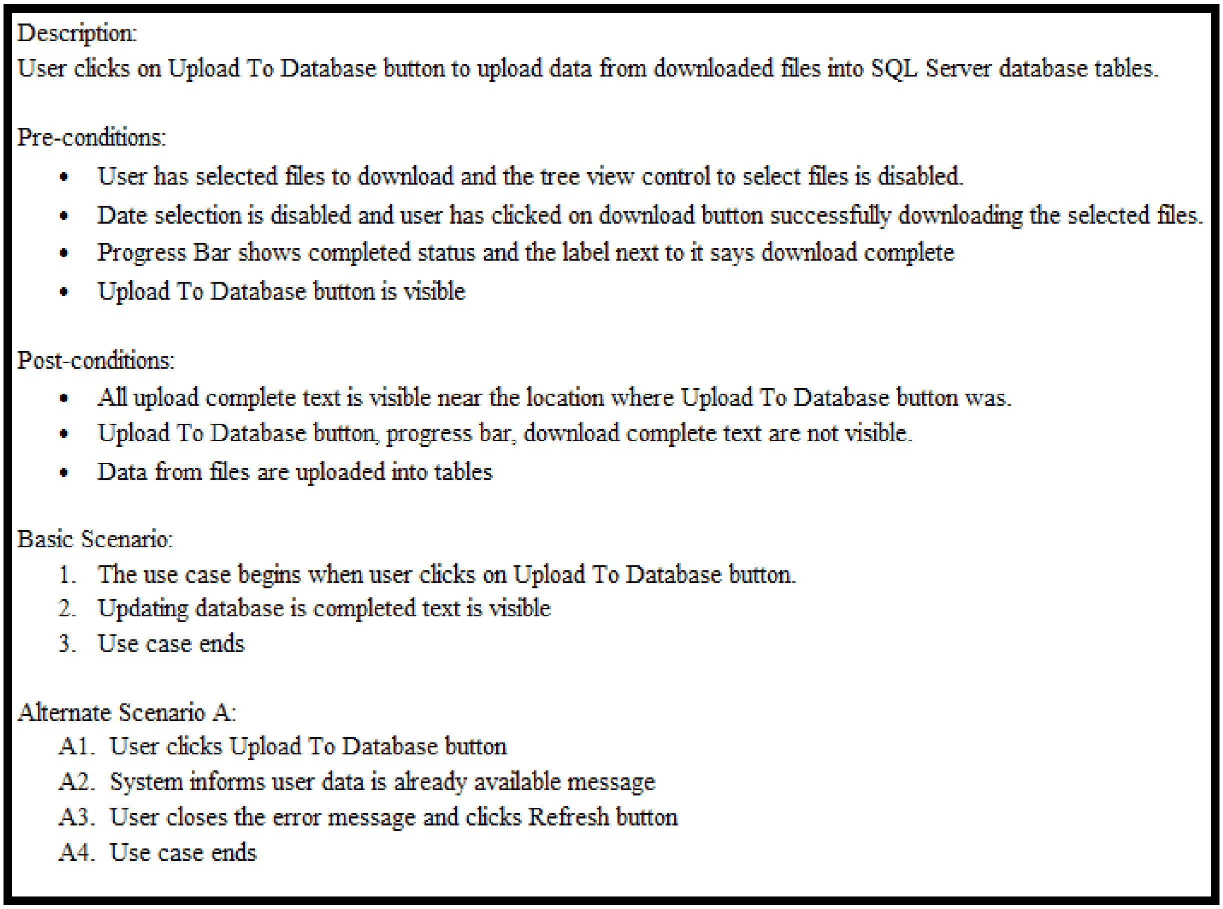
In Figure 18, partial content from the dbo.zonalbidload table is shown. This table is created from Zonal Load Commitment files. This table has six different columns. The TimeStamp column contains the timestamp for when the energy-load recording was done. In the figure, the timestamps increase every hour for all zones.



	PKzonalBidLoadId	Time Stamp	Time Zone	Name	PTID	Energy Bid Load
1	3185	2012-12-03 00:00:00.000	EST	CAPITL	61757	995
2	3186	2012-12-03 00:00:00.000	EST	CENTRL	61754	1128
3	3187	2012-12-03 00:00:00.000	EST	DUNWOD	61760	173
4	3188	2012-12-03 00:00:00.000	EST	GENESE	61753	469
5	3189	2012-12-03 00:00:00.000	EST	HUD VL	61758	578
6	3190	2012-12-03 00:00:00.000	EST	LONGIL	61762	1342
7	3191	2012-12-03 00:00:00.000	EST	MHK VL	61756	526
8	3192	2012-12-03 00:00:00.000	EST	MILLWD	61759	178
9	3193	2012-12-03 00:00:00.000	EST	N.Y.C.	61761	1819
10	3194	2012-12-03 00:00:00.000	EST	NORTH	61755	97
11	3195	2012-12-03 00:00:00.000	EST	WEST	61752	825
12	3196	2012-12-03 01:00:00.000	EST	CAPITL	61757	940
13	3197	2012-12-03 01:00:00.000	EST	CENTRL	61754	1072
14	3198	2012-12-03 01:00:00.000	EST	DUNWOD	61760	170
15	3199	2012-12-03 01:00:00.000	EST	GENESE	61753	453

Figure 18. Sample Data in a Table

After selecting the files to download and successfully downloading them onto the user’s local machine, the next step is to upload the data in the files into SQL server database tables. The use case in Figure 19 begins when a user clicks the “Upload To SQL Server” button. While uploading to database tables, data append to tables for each downloaded file. The ISO Analyzer and Reporting System figures out which database tables to be used to enter data. Finally, the data are uploaded to the related tables.



Description:
User clicks on Upload To Database button to upload data from downloaded files into SQL Server database tables.

Pre-conditions:

- User has selected files to download and the tree view control to select files is disabled.
- Date selection is disabled and user has clicked on download button successfully downloading the selected files.
- Progress Bar shows completed status and the label next to it says download complete
- Upload To Database button is visible

Post-conditions:

- All upload complete text is visible near the location where Upload To Database button was.
- Upload To Database button, progress bar, download complete text are not visible.
- Data from files are uploaded into tables

Basic Scenario:

1. The use case begins when user clicks on Upload To Database button.
2. Updating database is completed text is visible
3. Use case ends

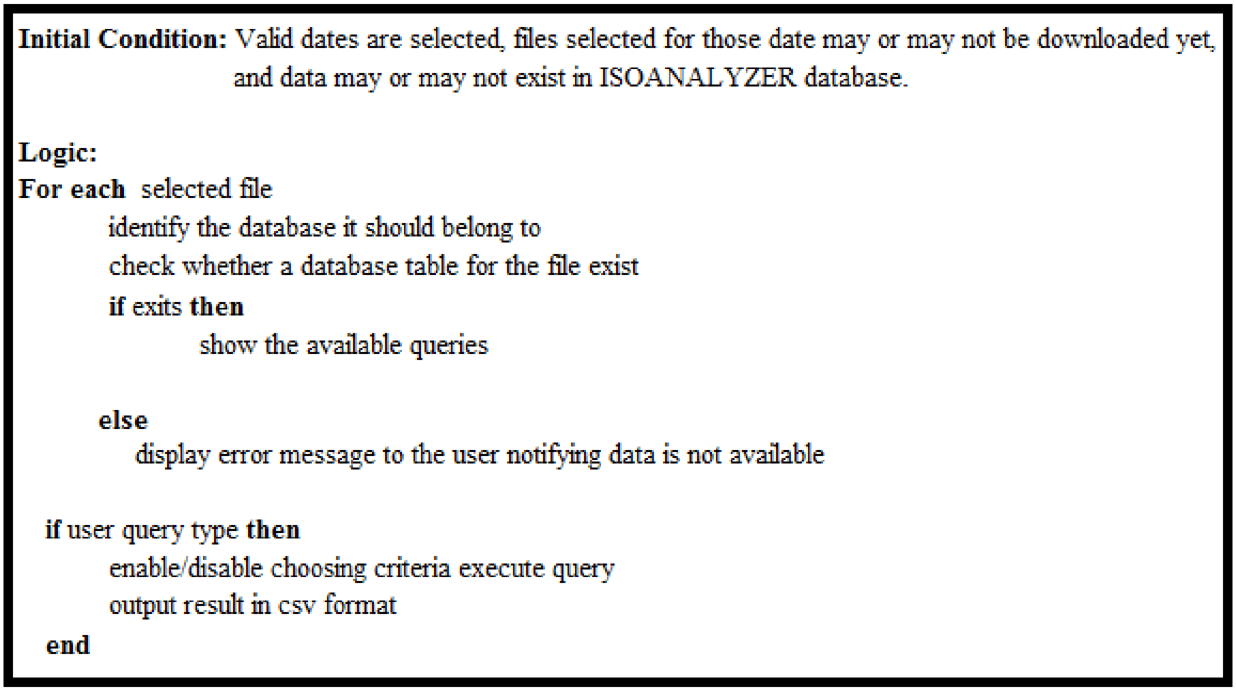
Alternate Scenario A:

A1. User clicks Upload To Database button
A2. System informs user data is already available message
A3. User closes the error message and clicks Refresh button
A4. Use case ends

Figure 19. Upload to SQL Server Use Case

3.5. Query

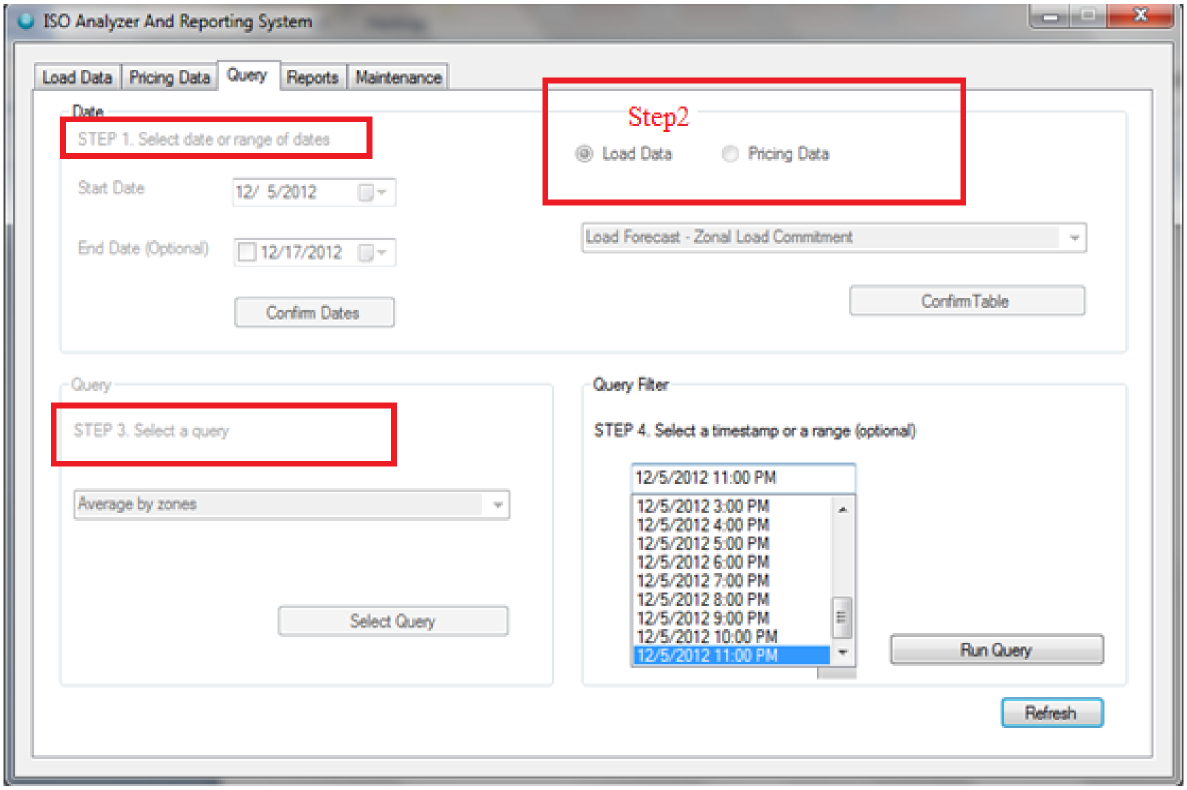
The Query tab in the ISO Analyzer and Reporting System queries the database tables in the ISOANALYZER database. The user can select the dates and files to query; where the ISO Analyzer and Reporting System identify the tables in the database. If the files are not in the database, then the user is notified. The ISO Analyzer and Reporting System provide customized query capability for these file types. Under Load Data, built-in queries are available for Zonal Load Commitment and Actual Load Integrated Real-time. Under Pricing Data, built-in queries are available for Day-Ahead Market LBMP and Integrated Real-Time market LBMP based on Zones. Figure 20 shows the logic behind Query tab implementation.



```
Initial Condition: Valid dates are selected, files selected for those date may or may not be downloaded yet,
                   and data may or may not exist in ISOANALYZER database.

Logic:
For each  selected file
     identify the database it should belong to
     check whether a database table for the file exist
     if exits then
          show the available queries

     else
       display error message to the user notifying data is not available

 if user query type then
     enable/disable choosing criteria execute query
     output result in csv format
 end
```

Figure 20. Query

The Query tab allows users to query the loaddata and pricingdata tables in the ISOANALYZER database. Users can select the date and the type of data, and based on whether they are loaddata or pricingdata, a dropdown box with a list of table types available for querying is populated. Then, STEP 3 lists the queries that are available for that table type. Figure 21 is a screenshot of the Query tab, where the user has selected a query to find the average by zones from a file for Zonal Load Commitment for December 5, 2012. STEP 4 lists the filters than can be applied to the user's query.



STEP 1. Select date or range of dates

Start Date: 12/ 5/2012

□ End Date (Optional): 12/17/2012

● Load Data ○ Pricing Data

Load Forecast - Zonal Load Commitment

STEP 3. Select a query

Average by zones

STEP 4. Select a timestamp or a range (optional)

12/5/2012 11:00 PM

12/5/2012 3:00 PM
12/5/2012 4:00 PM
12/5/2012 5:00 PM
12/5/2012 6:00 PM
12/5/2012 7:00 PM
12/5/2012 8:00 PM
12/5/2012 9:00 PM
12/5/2012 10:00 PM
12/5/2012 11:00 PM

Figure 21. Uploading Files into Database ISO Screenshot 9

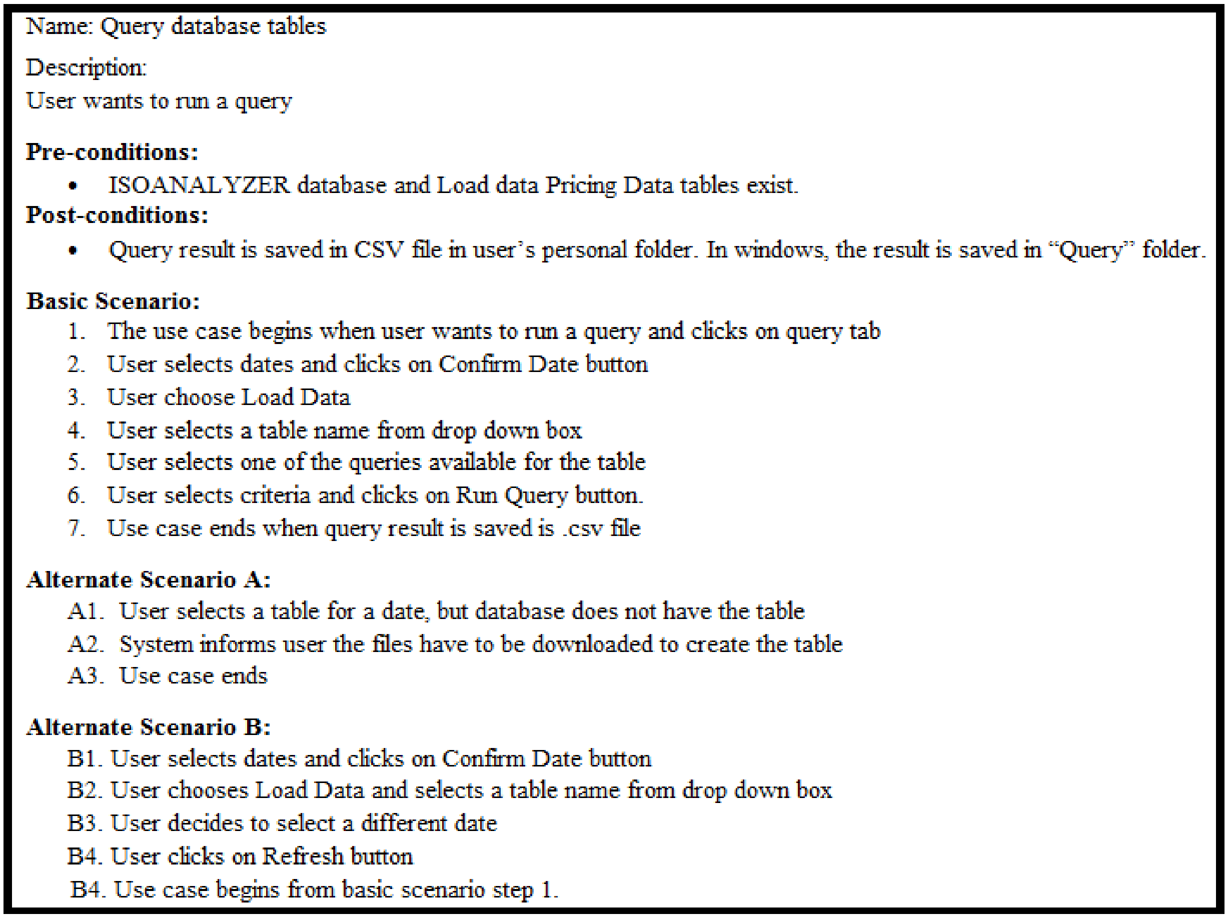
3.6. Custom Queries

Queries available for Zonal Load Commitment under load data and Day-Ahead Market LBMP under Pricing Data are averaged by zones. Queries available for Actual, Integrated Load, Real-time under Load Data and Integrated Real-time market LBMP under Pricing Data include these two queries in addition to the above queries, zonal average by timestamp interval and timestamp interval average by zones.

The average by zones query gives the average energy load, or average LBMP, based on the file queried. Zonal average by timestamp interval allows the users to select time intervals and to view the data according to different intervals. The intervals are multiples of 1. Also, the timestamp interval average by zones query allows the user to select time intervals and returns the results grouped by zones.

The result of a query is saved in the user's personal folder. If the user is utilizing Windows, then result is saved in the "Query" folders. If you run the same query over a period of time, it replaces the old file with the new result. It saves some space when the system overwrites the files.

Figure 22 shows the basic and alternative scenarios for Query tab. The use case for Query begins when a user wants to run a query and selects the Query tab. The user selects the date or a date range in the pricing or load data tab and then selects the table to be queried. This process creates the names of the tables to be queried. The ISO Analyzer and Reporting System then checks whether the data is available in the database tables that match the table names to be queried. If a table to be queried is missing from the database, the user is notified. The next course of action could be downloading that particular file and uploading the contents into an SQL server database for querying.



Name: Query database tables

Description:
User wants to run a query

Pre-conditions:

- ISOANALYZER database and Load data Pricing Data tables exist.

Post-conditions:

- Query result is saved in CSV file in user's personal folder. In windows, the result is saved in "Query" folder.

Basic Scenario:

1. The use case begins when user wants to run a query and clicks on query tab
2. User selects dates and clicks on Confirm Date button
3. User choose Load Data
4. User selects a table name from drop down box
5. User selects one of the queries available for the table
6. User selects criteria and clicks on Run Query button.
7. Use case ends when query result is saved is .csv file

Alternate Scenario A:

A1. User selects a table for a date, but database does not have the table
A2. System informs user the files have to be downloaded to create the table
A3. Use case ends

Alternate Scenario B:

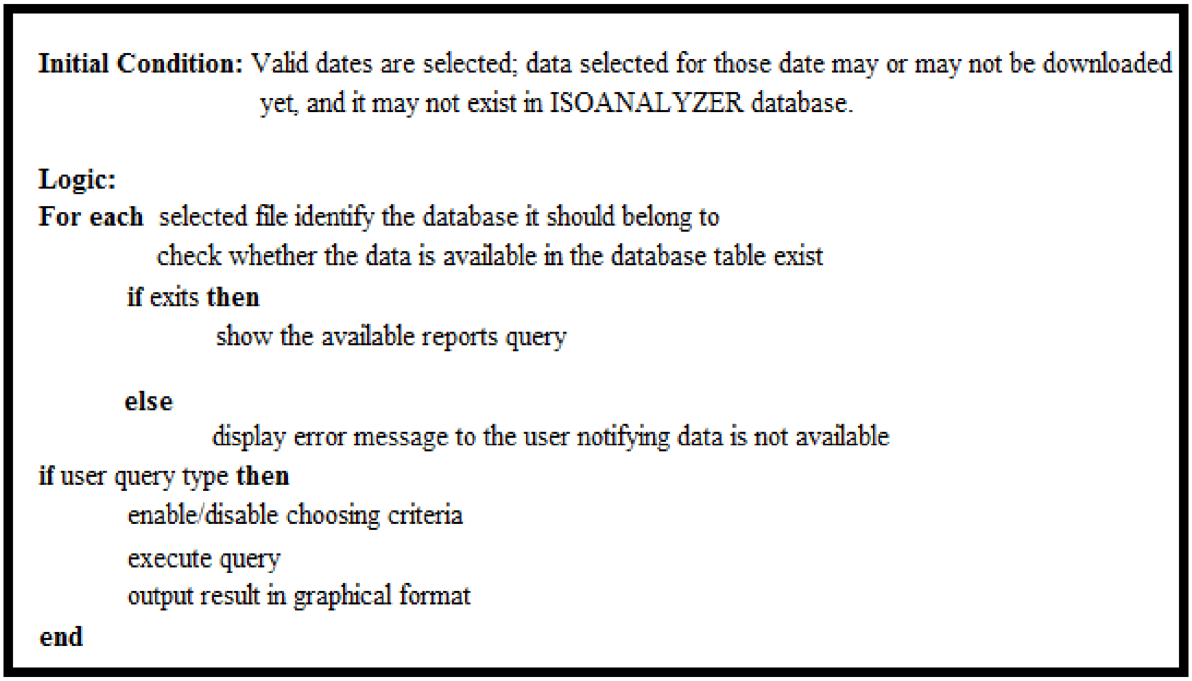
B1. User selects dates and clicks on Confirm Date button
B2. User chooses Load Data and selects a table name from drop down box
B3. User decides to select a different date
B4. User clicks on Refresh button
B4. Use case begins from basic scenario step 1.

Figure 22. Query Use Case

3.7. Reports

The Reports tab in the ISO Analyzer and Reporting Sytem queries the database tables to generate reports using the data available in the ISOANALYZER database tables. Users can select the dates and files they want to query; where the ISO Analyzer and Reporting System identify the tables in the database. If the data are not available in the tables for selected dates, then the user is notified. Under Load Data, built-in queries are available to generate reports for Zonal Load Commitment and Actual Load Integrated Real-Time; also, the queries for Zonal Load Commitment and Actual Integrated Real-Time are compared.

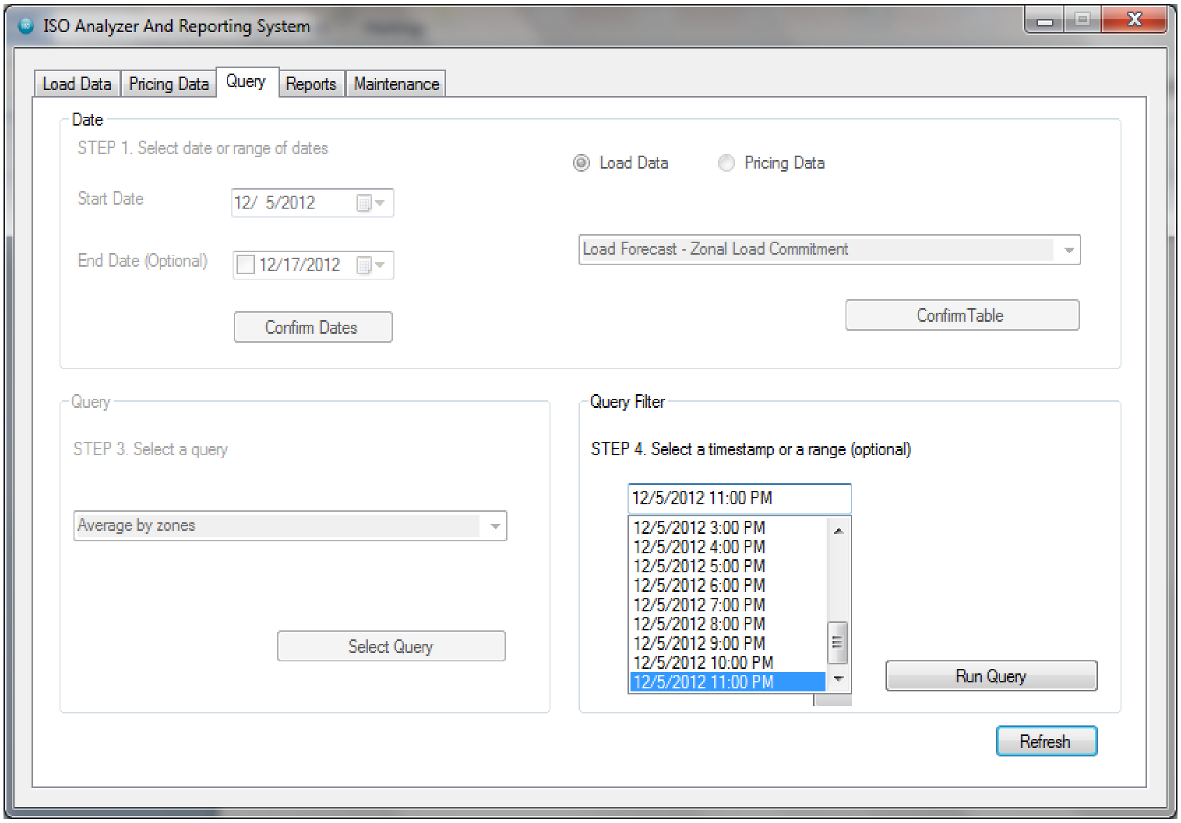
Under Pricing Data, built-in queries are available to generate reports for Day-Ahead Market LBMP and Integrated Real-Time Market LBMP, and to compare queries for Day-Ahead Market LBMP and Integrated Real-Time Market LBMP based on Zones. All these reports generate graphs in a new window. The logic behind Query tab implementation is shown in Figure 23.



```
Initial Condition: Valid dates are selected; data selected for those date may or may not be downloaded
                   yet, and it may not exist in ISOANALYZER database.

Logic:
For each  selected file identify the database it should belong to
          check whether the data is available in the database table exist
        if exits then
                show the available reports query

        else
                display error message to the user notifying data is not available
if user query type then
        enable/disable choosing criteria
        execute query
        output result in graphical format
end
```

Figure 23. Reports

The Reports tab allows the users to generate reports for the loaddata and pricingdata tables in an ISOANALYZER database. Users can select the date and the type of data, and based on whether they are loaddata or pricingdata, a dropdown box with a list of table types available for querying is populated. Then, STEP 3 lists the queries that are available for that table type. Figure 24 is a screenshot of the Query tab where the user has selected a query to find the average by zones from a file for the Zonal Load Commitment on December 5, 2012. STEP 4 lists the filters than can be applied to the user’s query.



Date

STEP 1. Select date or range of dates

● Load Data ○ Pricing Data

Start Date 12/ 5/2012

End Date (Optional) □ 12/17/2012

Load Forecast - Zonal Load Commitment

Query

STEP 3. Select a query

Average by zones

Query Filter

STEP 4. Select a timestamp or a range (optional)

12/5/2012 11:00 PM

12/5/2012 3:00 PM
12/5/2012 4:00 PM
12/5/2012 5:00 PM
12/5/2012 6:00 PM
12/5/2012 7:00 PM
12/5/2012 8:00 PM
12/5/2012 9:00 PM
12/5/2012 10:00 PM
12/5/2012 11:00 PM

Figure 24. Uploading Files into Database ISO Screenshot 10

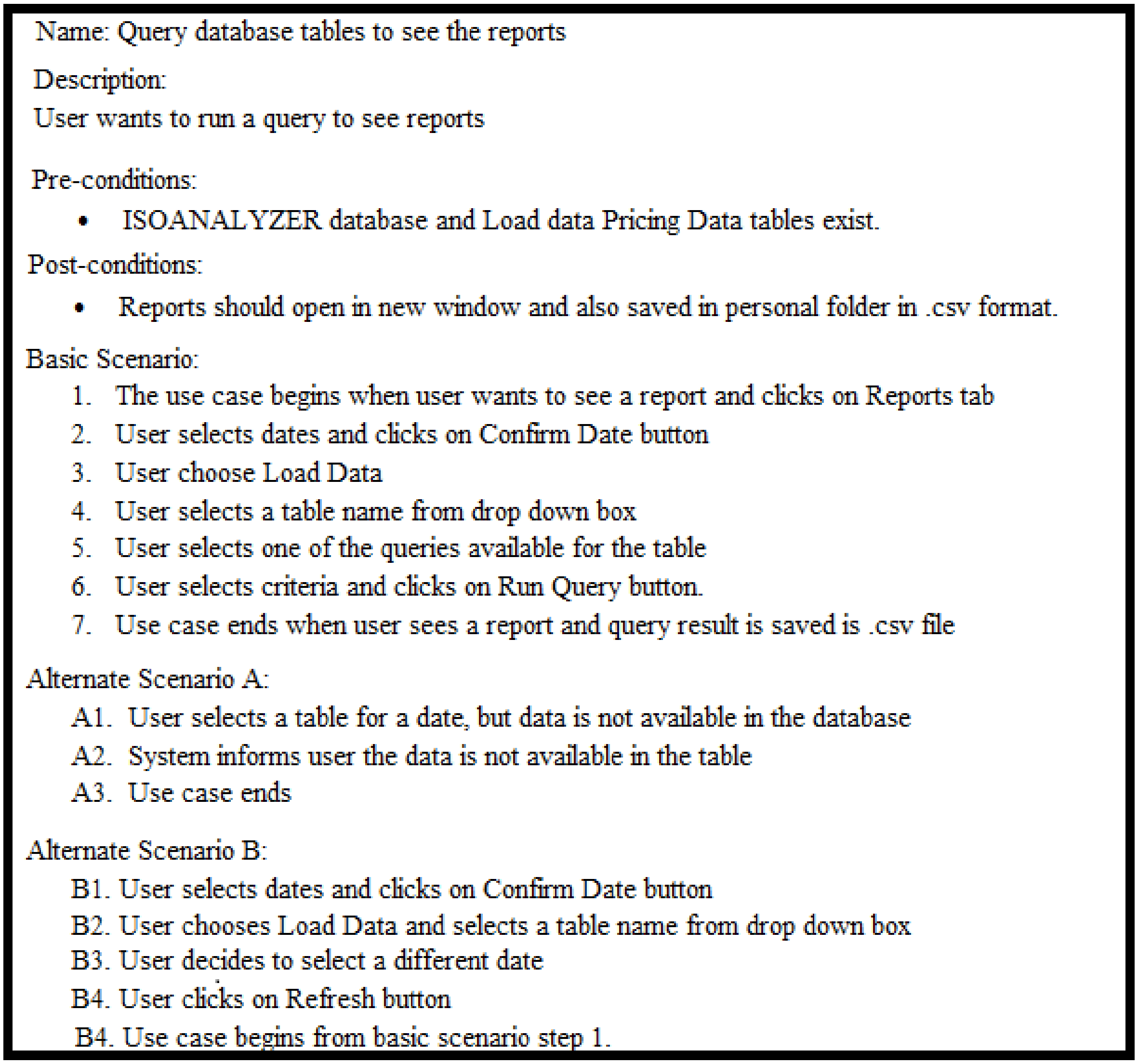
3.8. Custom Reports

Reports are available for Zonal Load Commitment, Integrated Real time under Load Data and Day-Ahead Market LBMP, Integrated RT-Zonal under Pricing Data are averaged by zones. Reports available for comparing the zonal load commitment and the Actual Integrated Real time under Load Data are “Compare loads.” Another report is available under pricing data to compare the data between Day-Ahead Market LBMP and Integrated RT-Zonal.

The average by zones report gives the average energy load, or average LBMP, based on the file queried. Comparing loads allows the users to contrast the average loads between LBMP and Integrated RT. The above two queries allow users to select time intervals and to view the data based on different intervals, such as listing the data in 1-hour intervals, etc. The intervals are multiples of 1. Also, the average by zones query allows the user to select time intervals and returns the result grouped by zones.

The result of a query generates a report in a graphical representation in a new Internet Explorer window. The result is also saved as a .csv fileformat as discussed in the Query section.

Figure 25 shows the basic and alternative scenarios for Reports. The use case for Reports begins when a user wants to generate a report and selects the Reports tab. The user selects a date or a date range as well as if it is pricing or load data, and then selects the table to be used. This process creates names for the tables to be queried to see the reports. The ISO Analyzer and Reporting System checks whether the data is available in the database tables matching with the table names selected by the user. If the data are not available in the database table, the user is notified. The next course of action could be downloading that particular file and uploading the contents into an SQL server database to see the reports.



Name: Query database tables to see the reports

Description:
User wants to run a query to see reports

Pre-conditions:

- ISOANALYZER database and Load data Pricing Data tables exist.

Post-conditions:

- Reports should open in new window and also saved in personal folder in .csv format.

Basic Scenario:

1. The use case begins when user wants to see a report and clicks on Reports tab
2. User selects dates and clicks on Confirm Date button
3. User choose Load Data
4. User selects a table name from drop down box
5. User selects one of the queries available for the table
6. User selects criteria and clicks on Run Query button.
7. Use case ends when user sees a report and query result is saved is .csv file

Alternate Scenario A:

A1. User selects a table for a date, but data is not available in the database
A2. System informs user the data is not available in the table
A3. Use case ends

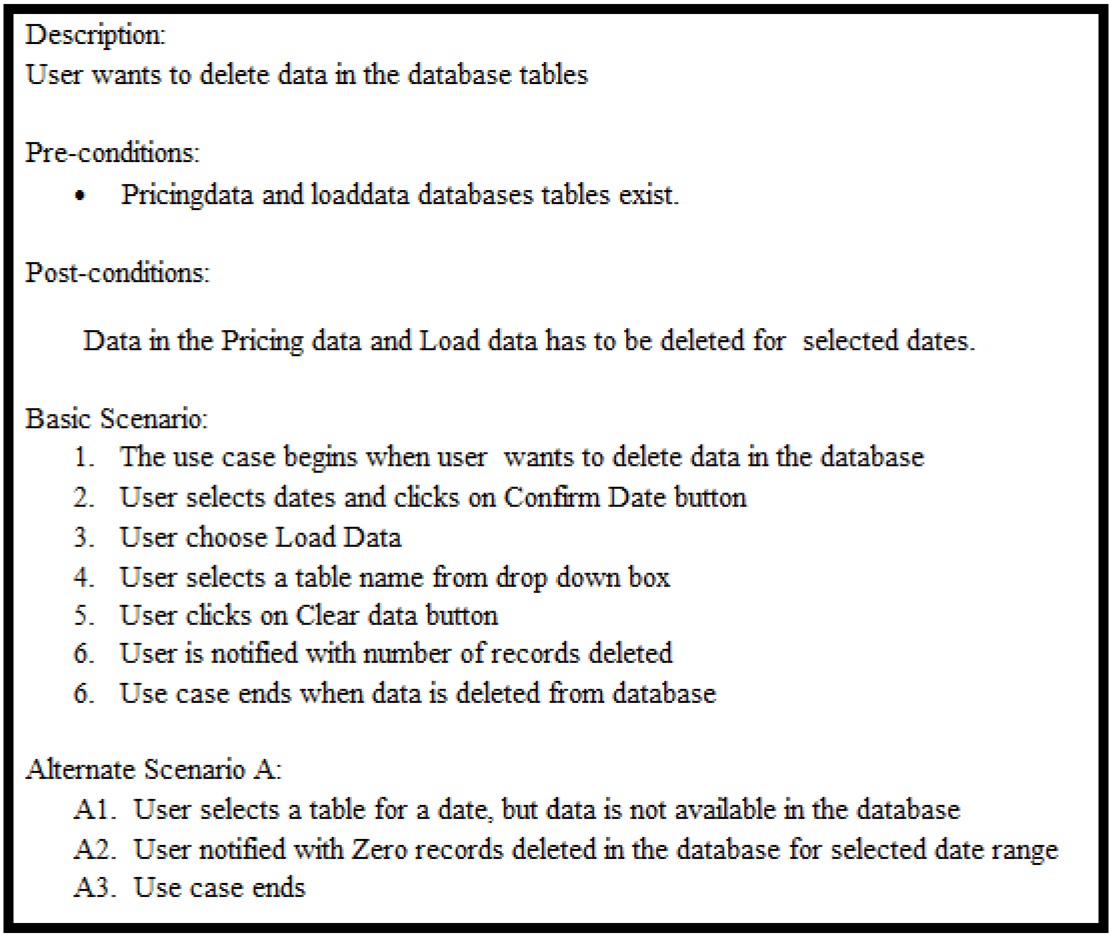
Alternate Scenario B:

B1. User selects dates and clicks on Confirm Date button
B2. User chooses Load Data and selects a table name from drop down box
B3. User decides to select a different date
B4. User clicks on Refresh button
B4. Use case begins from basic scenario step 1.

Figure 25. Reports Use Case

3.9. Maintenance

Uploading data into a database over a period of time increases the database size. For better performance, we need to clean the data in the ISOANALYZER database. The maintenance tab in the ISO Analyzer and Reporting System performs the database cleaning task. The user selects the date range and confirms the dates. The user selects the Load Data/Pricing Data tables to delete the data and clicks the Delete tab to delete the data. Figure 26 shows the logic behind the Maintenance tab implementation.



Description:
User wants to delete data in the database tables

Pre-conditions:

- Pricingdata and loaddata databases tables exist.

Post-conditions:

Data in the Pricing data and Load data has to be deleted for selected dates.

Basic Scenario:

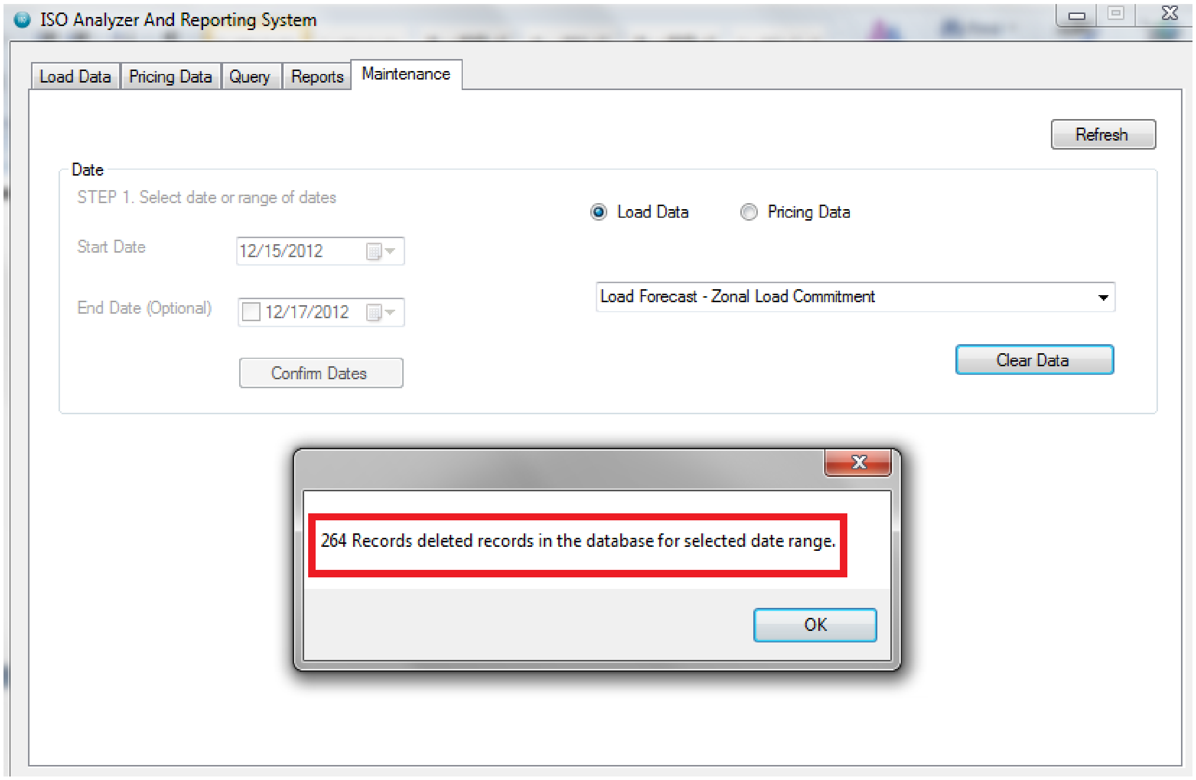
1. The use case begins when user wants to delete data in the database
2. User selects dates and clicks on Confirm Date button
3. User choose Load Data
4. User selects a table name from drop down box
5. User clicks on Clear data button
6. User is notified with number of records deleted
6. Use case ends when data is deleted from database

Alternate Scenario A:

A1. User selects a table for a date, but data is not available in the database
A2. User notified with Zero records deleted in the database for selected date range
A3. Use case ends

Figure 26. Maintenance Use Case

The screenshot in Figure 27 describes the Maintenance tab's functionality.



Start Date: 12/15/2012

End Date (Optional): □ 12/17/2012

● Load Data ○ Pricing Data

Load Forecast - Zonal Load Commitment

264 Records deleted records in the database for selected date range.

Figure 27. ISO Analyzer and Reporting System Screenshot 11

3.10. ISO Analyzer and Reporting System Class Diagram

Below is the static structure of the ISO Analyzer and Reporting System. We will briefly go over the class diagram after presenting Figure 27.

Figure 28 shows the classes that make up the ISO Analyzer and Reporting System. Each class has its own set of fields and methods. The Getters and Setters methods have been not been included in the diagram for readability and space. In the figure, the form Main class contains event handlers which are called when a user interacts with the system, such as clicking the Download button, which would call a button-click event handler for the Download button. Tasks assigned to the event handler are conducted by calling various methods in the classes provided.

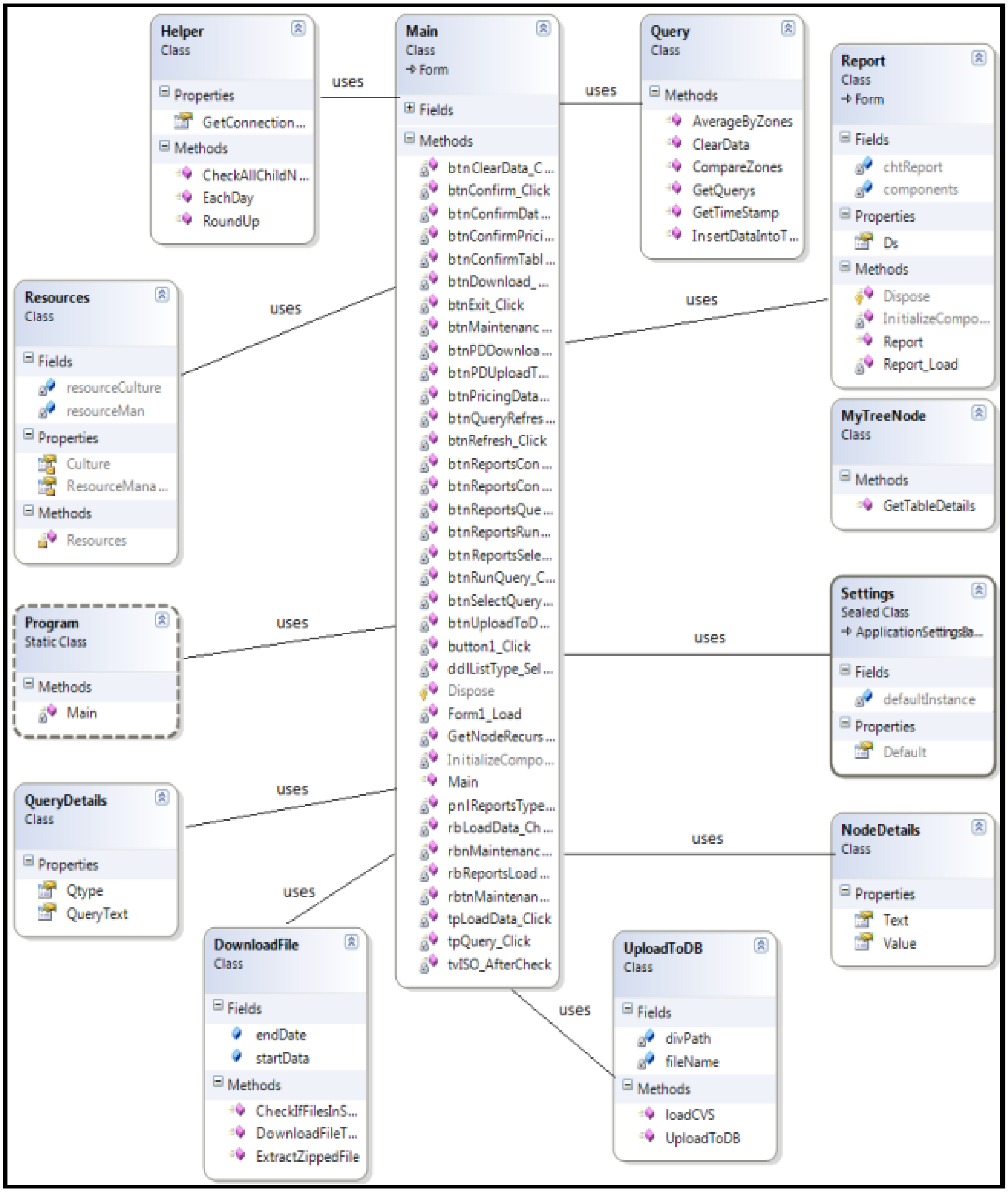
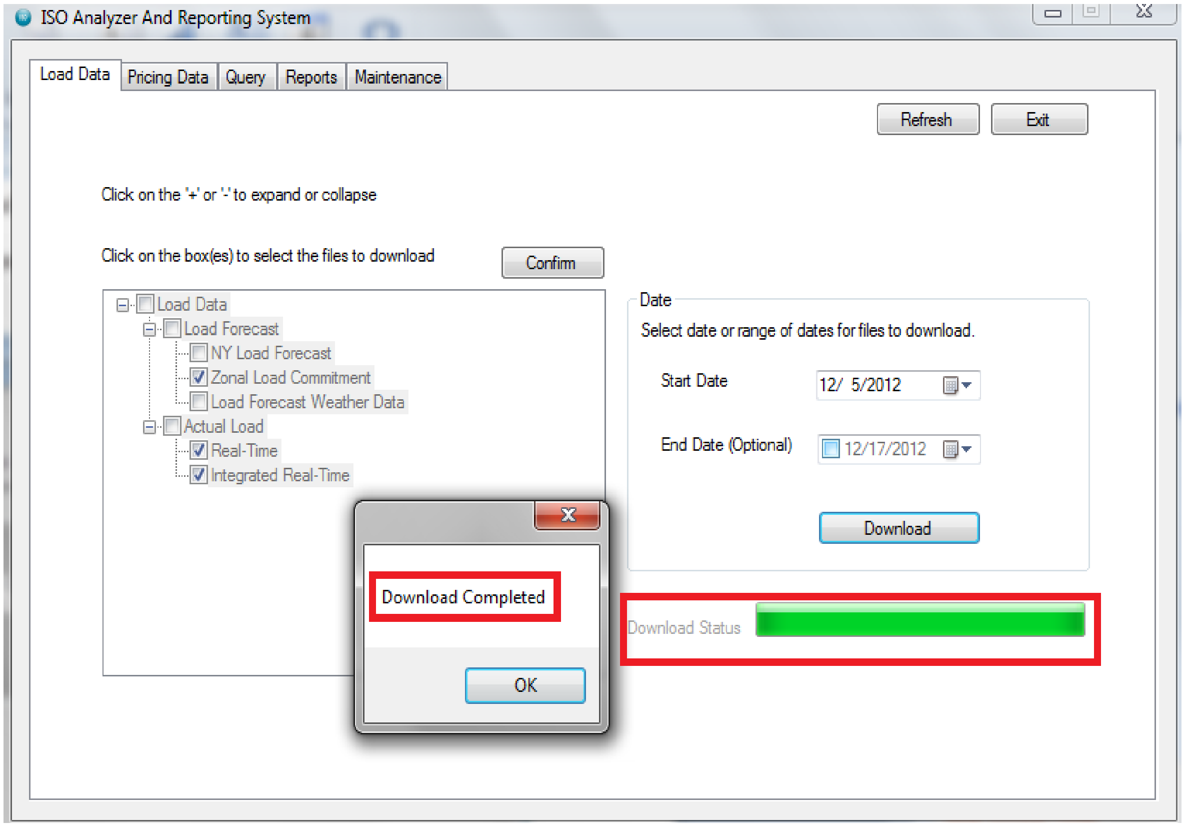


Figure 28. ISO Analyzer and Reporting System Class Diagram

For example, the MyTreeNode class methods are called to automatically check on the checkboxes next to all the child nodes if their parent node is checked in the tree view.

4. SYSTEM OPERATION

This chapter illustrates how the ISO Analyzer and Reporting System allows the selection of a file for a certain date to be downloaded from NYISO's website and how data from the file are uploaded into an SQL server database table for querying and reporting. Before running the ISO Analyzer and Reporting System program, loaddata and pricingdata databases, and their tables should be created in a SQL Server database. For example the user wants to download the real-time actual load under load data for December 5, 2012. After making the date and file selections, the user clicks the Download button to download the file. Figure 29 shows the file and date selections as well as the download in completed status.



Click on the '+' or '-' to expand or collapse

Click on the box(es) to select the files to download

- □ Load Data
  - □ Load Forecast
    - □ NY Load Forecast
    - ☑ Zonal Load Commitment
    - □ Load Forecast Weather Data
  - □ Actual Load
    - ☑ Real-Time
    - ☑ Integrated Real-Time

Date

Select date or range of dates for files to download.

Start Date 12/ 5/2012

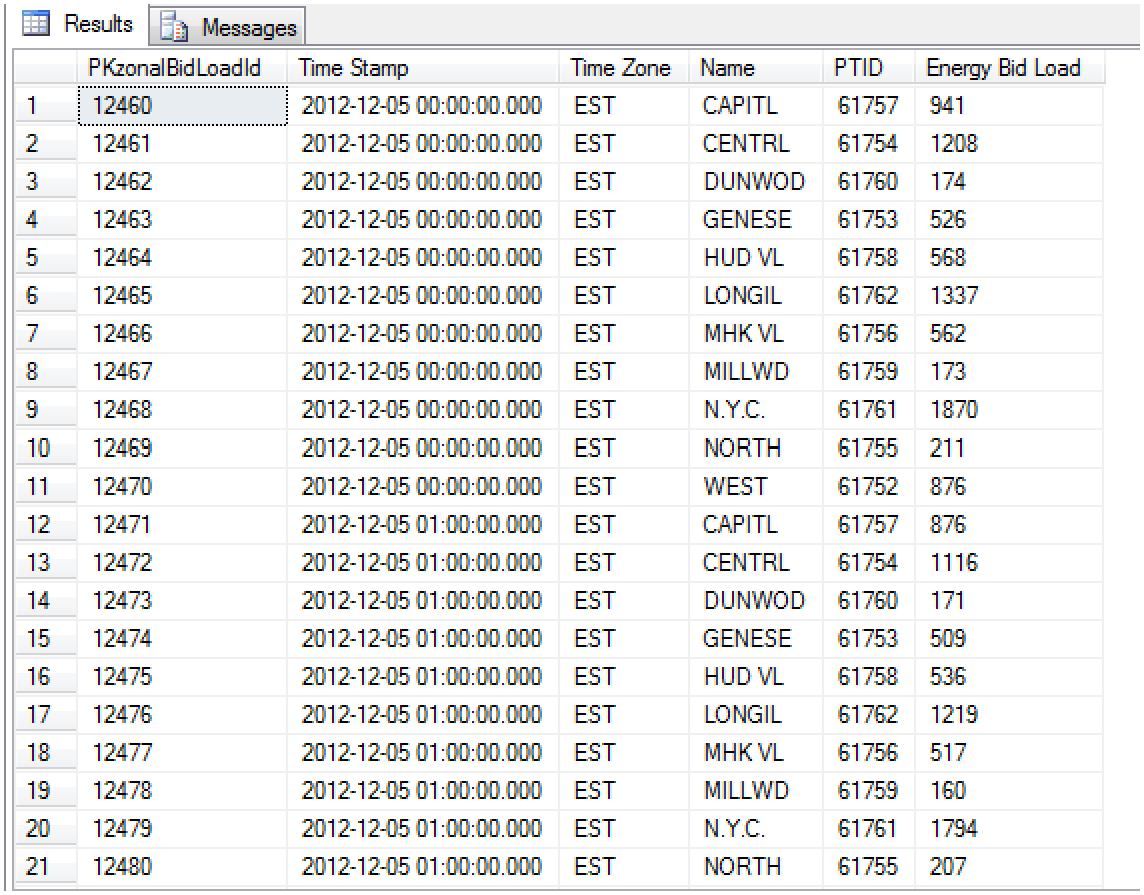
End Date (Optional) □ 12/17/2012

Download Completed

Download Status

Figure 29. ISO Analyzer and Reporting System Screenshot 12

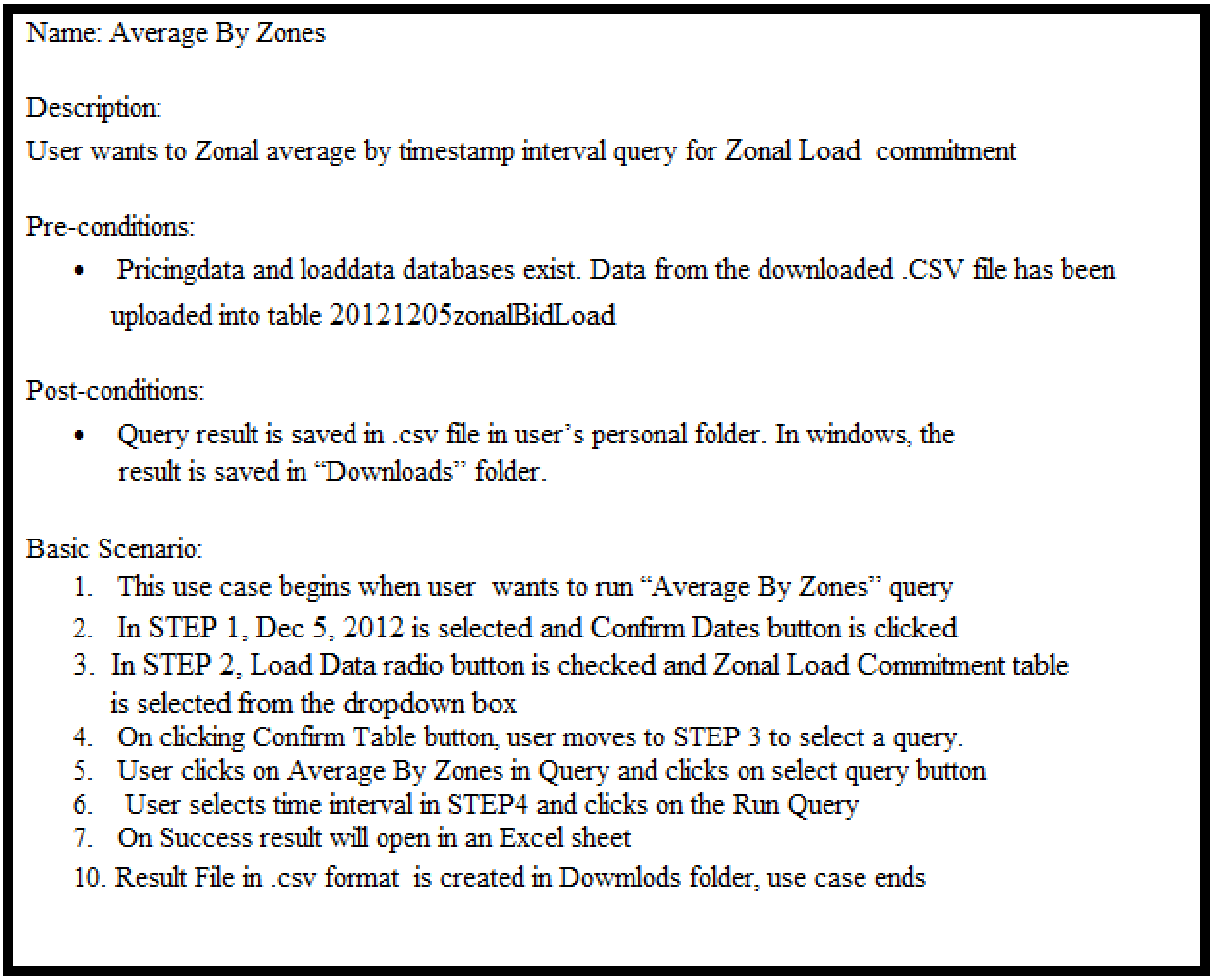
Once you click the OK button, the "Upload To Database" button is displayed. After clicking the "Upload To Database" button, the data from the downloaded file are uploaded to a new SQL server table. Figure 30 shows a partial screenshot of the new table as seen using SQL Server Management Studio 10.



	PKzonalBidLoadId	Time Stamp	Time Zone	Name	PTID	Energy Bid Load
1	12460	2012-12-05 00:00:00.000	EST	CAPITL	61757	941
2	12461	2012-12-05 00:00:00.000	EST	CENTRL	61754	1208
3	12462	2012-12-05 00:00:00.000	EST	DUNWOD	61760	174
4	12463	2012-12-05 00:00:00.000	EST	GENESE	61753	526
5	12464	2012-12-05 00:00:00.000	EST	HUD VL	61758	568
6	12465	2012-12-05 00:00:00.000	EST	LONGIL	61762	1337
7	12466	2012-12-05 00:00:00.000	EST	MHK VL	61756	562
8	12467	2012-12-05 00:00:00.000	EST	MILLWD	61759	173
9	12468	2012-12-05 00:00:00.000	EST	N.Y.C.	61761	1870
10	12469	2012-12-05 00:00:00.000	EST	NORTH	61755	211
11	12470	2012-12-05 00:00:00.000	EST	WEST	61752	876
12	12471	2012-12-05 01:00:00.000	EST	CAPITL	61757	876
13	12472	2012-12-05 01:00:00.000	EST	CENTRL	61754	1116
14	12473	2012-12-05 01:00:00.000	EST	DUNWOD	61760	171
15	12474	2012-12-05 01:00:00.000	EST	GENESE	61753	509
16	12475	2012-12-05 01:00:00.000	EST	HUD VL	61758	536
17	12476	2012-12-05 01:00:00.000	EST	LONGIL	61762	1219
18	12477	2012-12-05 01:00:00.000	EST	MHK VL	61756	517
19	12478	2012-12-05 01:00:00.000	EST	MILLWD	61759	160
20	12479	2012-12-05 01:00:00.000	EST	N.Y.C.	61761	1794
21	12480	2012-12-05 01:00:00.000	EST	NORTH	61755	207

Figure 30. Sample Table Data After SQL Server Upload

The Query tab in the ISO Analyzer and Reporting System helps the user to query the database tables. We want to query the file just downloaded; the data from the file are already uploaded in a table called 20121205zonalBidLoad. In the Query tab, we make selections as demonstrated in Figure 31. Below is the use case we will follow for demonstration purposes.



Name: Average By Zones

Description:
User wants to Zonal average by timestamp interval query for Zonal Load commitment

Pre-conditions:

- Pricingdata and loaddata databases exist. Data from the downloaded .CSV file has been uploaded into table 20121205zonalBidLoad

Post-conditions:

- Query result is saved in .csv file in user's personal folder. In windows, the result is saved in "Downloads" folder.

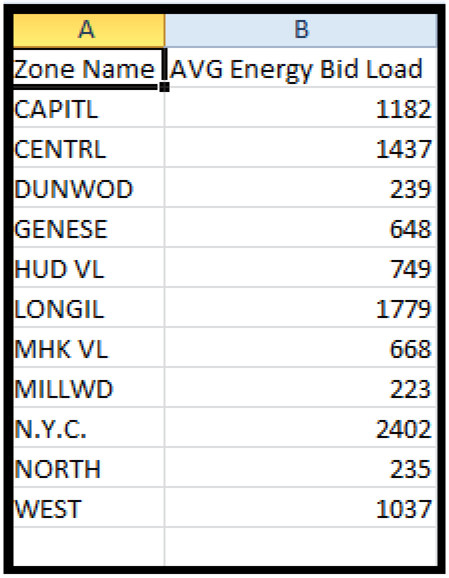
Basic Scenario:

1. This use case begins when user wants to run "Average By Zones" query
2. In STEP 1, Dec 5, 2012 is selected and Confirm Dates button is clicked
3. In STEP 2, Load Data radio button is checked and Zonal Load Commitment table is selected from the dropdown box
4. On clicking Confirm Table button, user moves to STEP 3 to select a query.
5. User clicks on Average By Zones in Query and clicks on select query button
6. User selects time interval in STEP4 and clicks on the Run Query
7. On Success result will open in an Excel sheet

10. Result File in .csv format is created in Dowmlods folder, use case ends

Figure 31. Zonal Average by Timestamp Interval Query Basic Use Case

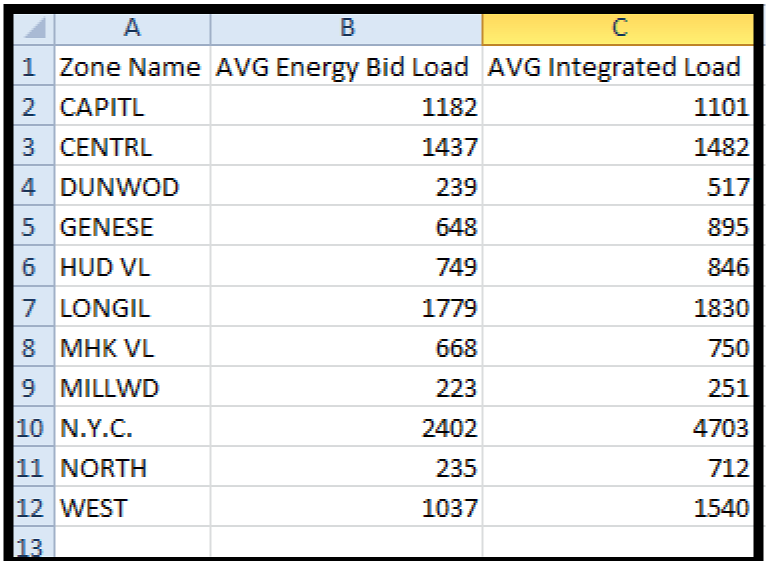
The result of the query is in .csv format, which can be easily opened in Excel. Figure 32 shows the result in the form of an Excel file with data in a tabular format. It shows the average of all zones by "Zone Name".



Zone Name	AVG Energy Bid Load
CAPITL	1182
CENTRL	1437
DUNWOD	239
GENESE	648
HUD VL	749
LONGIL	1779
MHK VL	668
MILLWD	223
N.Y.C.	2402
NORTH	235
WEST	1037

Figure 32. Average by Zones of Load Commitment Data

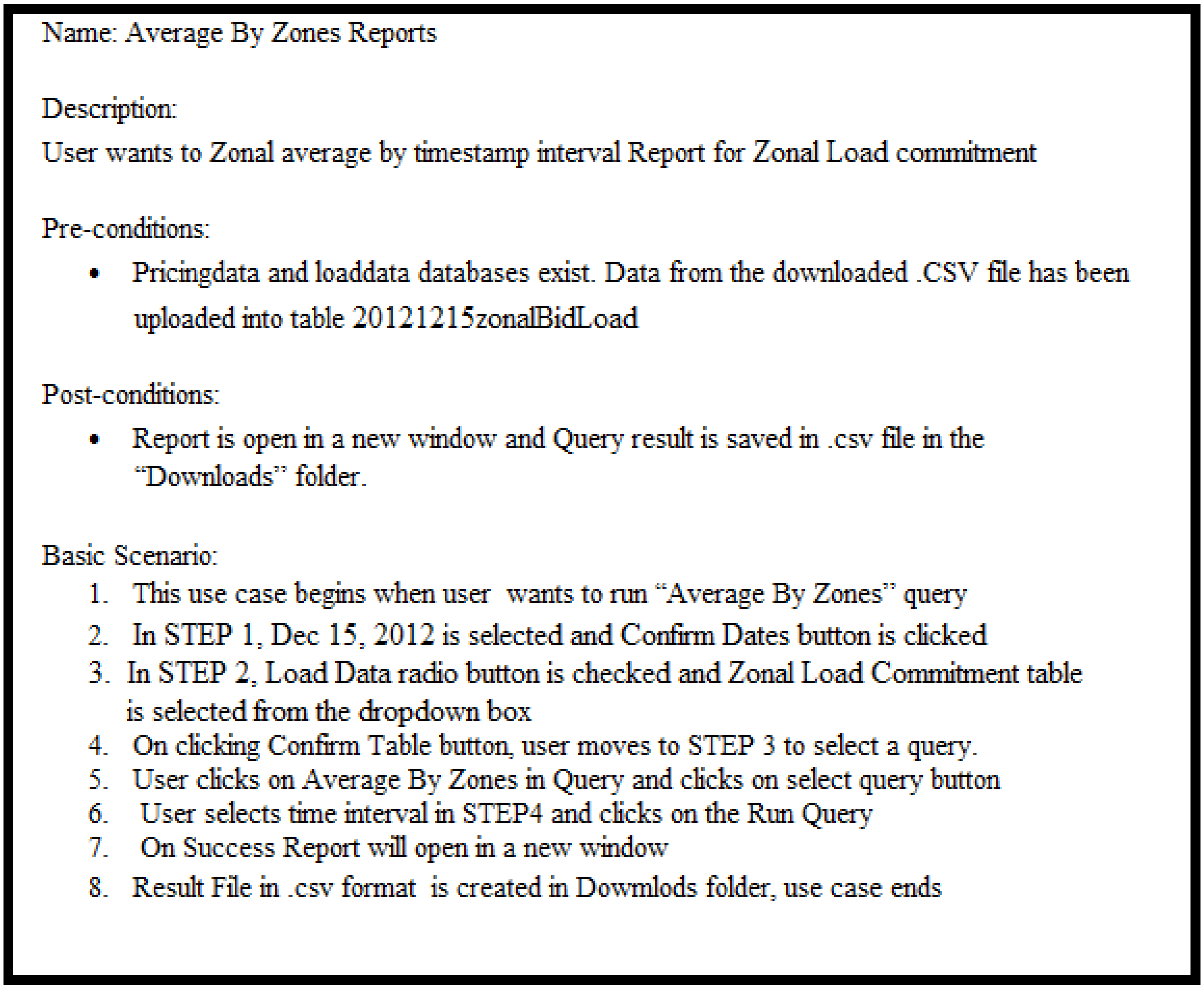
Similarly, if another query, Compare Loads, was selected, the resulting file would compare data between the Average Energy Bid Load and Average Integrated Load. Figure 33 shows the results in Excel.



1	Zone Name	AVG Energy Bid Load	AVG Integrated Load
2	CAPITL	1182	1101
3	CENTRL	1437	1482
4	DUNWOD	239	517
5	GENESE	648	895
6	HUD VL	749	846
7	LONGIL	1779	1830
8	MHK VL	668	750
9	MILLWD	223	251
10	N.Y.C.	2402	4703
11	NORTH	235	712
12	WEST	1037	1540
13			

Figure 33. Compare Loads by Zone Average

The Reports tab in the ISO Analyzer and Reporting System helps the user to generate reports on the data available in the database according to user queries. We want to see the reports for a file just downloaded; the data from the file are already uploaded in a table called 20121215zonalBidLoad. In the Query tab, we make the selections demonstrated in Figure 34. Below is the use case we will follow for demonstration purposes.



Name: Average By Zones Reports

Description:
User wants to Zonal average by timestamp interval Report for Zonal Load commitment

Pre-conditions:

- Pricingdata and loaddata databases exist. Data from the downloaded .CSV file has been uploaded into table 20121215zonalBidLoad

Post-conditions:

- Report is open in a new window and Query result is saved in .csv file in the "Downloads" folder.

Basic Scenario:

1. This use case begins when user wants to run "Average By Zones" query
2. In STEP 1, Dec 15, 2012 is selected and Confirm Dates button is clicked
3. In STEP 2, Load Data radio button is checked and Zonal Load Commitment table is selected from the dropdown box
4. On clicking Confirm Table button, user moves to STEP 3 to select a query.
5. User clicks on Average By Zones in Query and clicks on select query button
6. User selects time interval in STEP4 and clicks on the Run Query
7. On Success Report will open in a new window
8. Result File in .csv format is created in Dowmlods folder, use case ends

Figure 34. Average by Zones Report Use Case

The result of the query opens as a report in a new window. Figure 35 shows the result in the form of a graphical representation. It shows the average of all zones by "Zone Name".

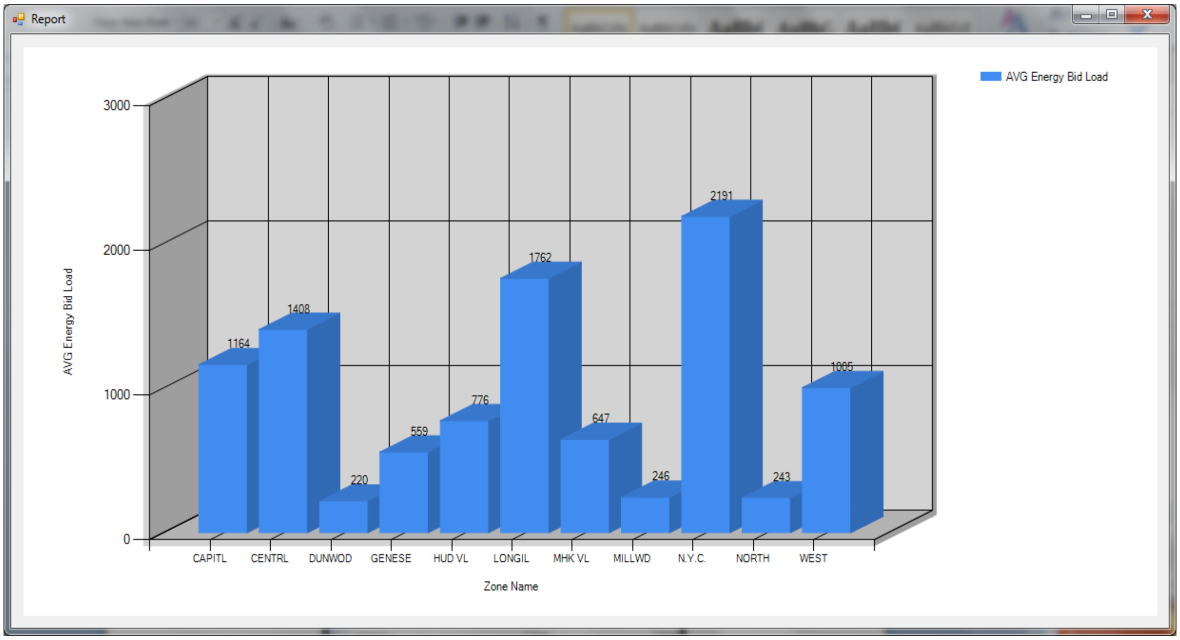


Figure 35. Average by Zones of Load Commitment Data Report

Similarly, Figure 36 shows the result in the form of a graphical representation. It shows the average of all zones by Zone name for Integrated Real-time data.

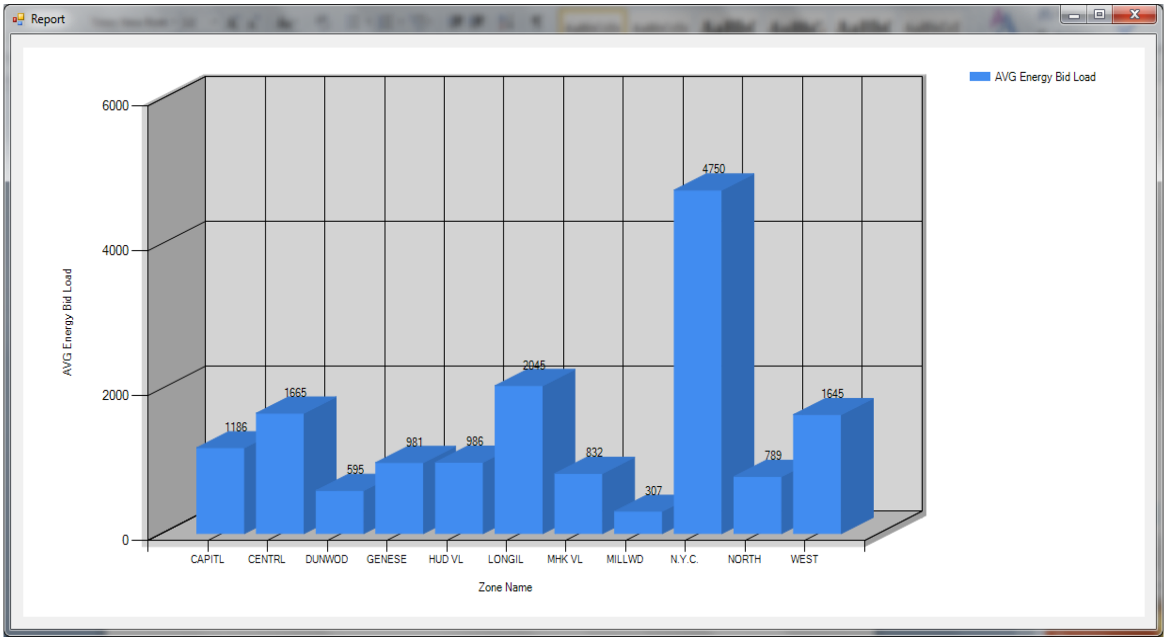


Figure 36. Average by Zones of Actual, Integrated, Real-Time Data

If another query, Compare Loads, is selected, the results are shown in Figure 37.

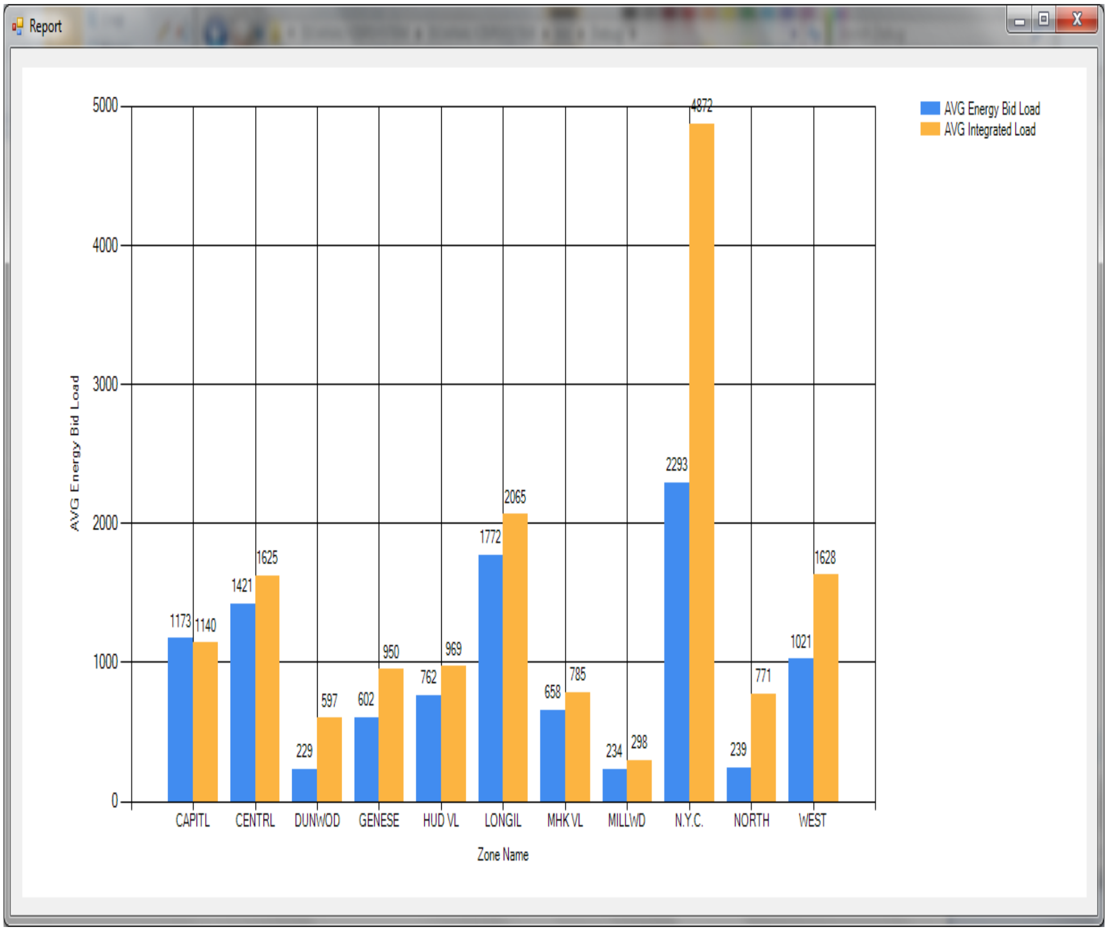
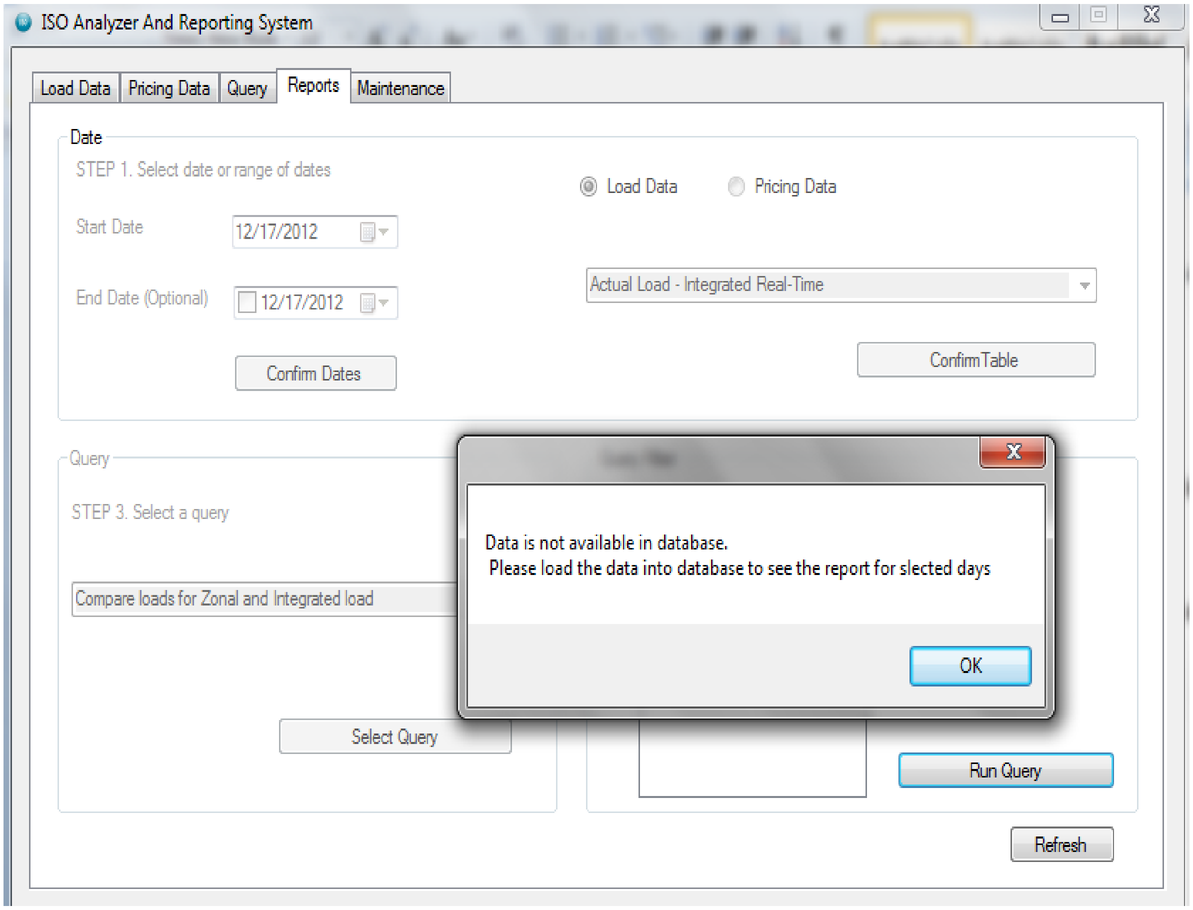


Figure 37. Compare Loads Average by Zones

Here is an alternative scenario for Figure 37: Compare Loads by Zones. When a user decides to query a database, the data in the tables may not exist in the database, so a user has to download the related files from NYISO’s website and upload the data into the tables. An appropriate error message or instruction needs to be given to the user in cases such as the data not being available in the database, and then please load the data into the database to see the report for the selected dates.



Start Date 12/17/2012

End Date (Optional) □ 12/17/2012

● Load Data ○ Pricing Data

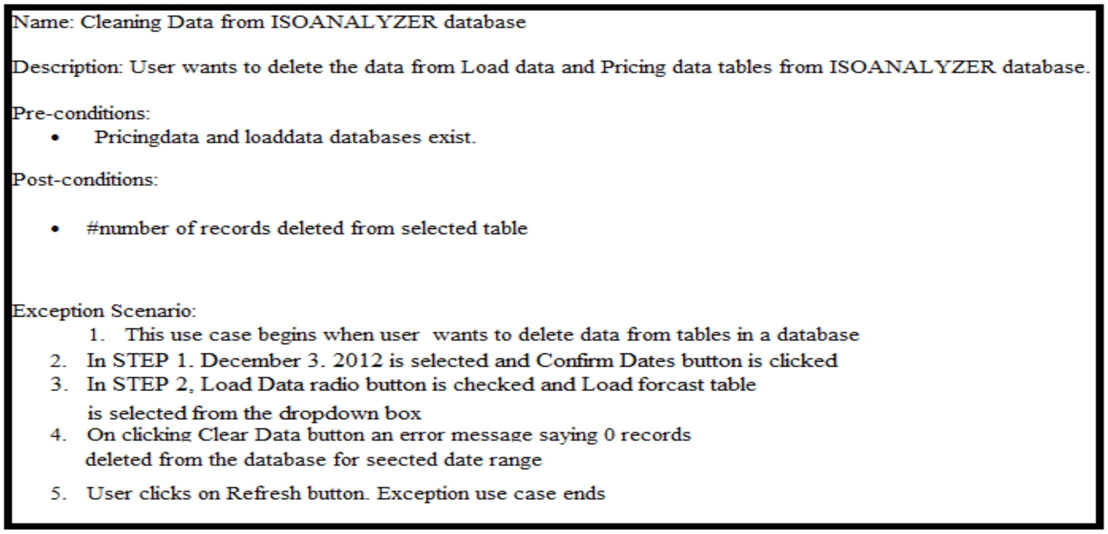
Actual Load - Integrated Real-Time

Compare loads for Zonal and Integrated load

Data is not available in database.
Please load the data into database to see the report for slected days

Figure 38. ISO Analyzer and Reporting System Screenshot 13

The Maintenance tab in the ISO Analyzer and Reporting System helps the user to clean the database as a regular database maintenance activity. The data from the Load Data files are already uploaded into the relative tables in the ISOANALYZER database. In the Maintenance tab, we make the selections as demonstrated in Figure 39 to delete the data from a table. Below is the use case we will follow for demonstration purposes. The most crucial steps in the use case are to look for the pre-conditions which are required for the use case to execute successfully. The user can verify if the data has been deleted from the database or not after post-deletion.



Name: Cleaning Data from ISOANALYZER database

Description: User wants to delete the data from Load data and Pricing data tables from ISOANALYZER database.

Pre-conditions:

- Pricingdata and loaddata databases exist.

Post-conditions:

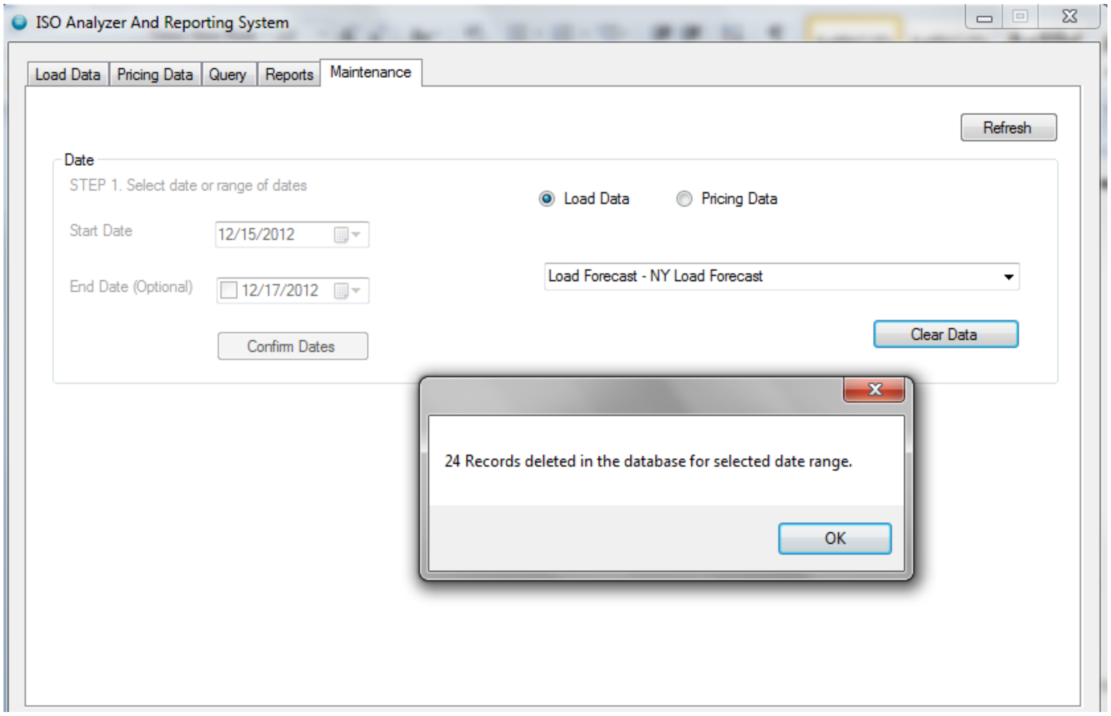
- #number of records deleted from selected table

Exception Scenario:

1. This use case begins when user wants to delete data from tables in a database
2. In STEP 1. December 3. 2012 is selected and Confirm Dates button is clicked
3. In STEP 2, Load Data radio button is checked and Load forcast table is selected from the dropdown box
4. On clicking Clear Data button an error message saying 0 records deleted from the database for seected date range
5. User clicks on Refresh button. Exception use case ends

Figure 39. Deleting the Data from Tables in the Database

Figure 40 is a screenshot for deleting data for the load forecast table in the database. In the example below there are 24 records which were deleted for the selected range of dates.



Start Date 12/15/2012

End Date (Optional) □ 12/17/2012

● Load Data ○ Pricing Data

Load Forecast - NY Load Forecast

24 Records deleted in the database for selected date range.

Figure 40. ISO Analyzer and Reporting System Screenshot 14

5. CONCLUSION AND FUTURE WORK

Using the ISO Analyzer and Reporting System, we can run queries and generate reports on data to obtain desired information in the .csv file format and to see the reports in a graphical representation. The ISO Analyzer and Reporting System takes in data from a regular CSV file and uploads them to SQL server database tables which can be queried and used when generating reports using a GUI. Data from a period of time can be analyzed quickly using query results and generating graphs instead of having to download data every time a user has to see them. Downloaded files can be easily opened in Excel format, and reports can be generated anytime using a GUI.

The ISO Analyzer and Reporting System automates the process of downloading different file formats (.csv or .zip) from the NYISO site and uploading them to a database to help the users visualize the electricity-market trends. Though the ISO Analyzer and Reporting System download the zipped folder, only selected files are extracted. For future work, this tool can be enhanced and implemented with a MIDWEST ISO system.

6. REFERENCES

1. "NYISO (Markets Operations)." NYISO. http://www.nyiso.com/public/markets_operations/index.jsp (accessed December 10, 2012).
2. "Load Data and Pricing Data files." LBMP Results Index. http://mis.nyiso.com/public/ (accessed December 5, 2012).
3. Nagel, Christian, and Bill Evjen. *Professional C# 2008*. Indianapolis, Ind.: Wiley Publishing, Inc., 2008.
4. "Introducing Visual Studio .NET." MSDN. http://msdn.microsoft.com/en-us/library/fx6bk1f4%28v=vs.71%29.aspx (accessed December 15, 2012).
5. Doyle, Barbara. *Microsoft Visual C# .NET programming: from problem analysis to program design*. Boston, Mass.: Thomson/Course Technology, 2004.
6. "Base Class Library." MSDN. http://msdn.microsoft.com/en-us/netframework/aa569603.aspx (accessed November 30, 2012).
7. ".NET Framework." MSDN. http://msdn.microsoft.com/en-us/netframework (accessed December 15, 2012).
8. "Microsoft Visual Studio Premium." Microsoft Corporation. http://www.microsoft.com/visualstudio/en-us/products/2010-editions/premium/overview (accessed December 15, 2012).
9. "Microsoft SQL Server." Microsoft Corporation. http://www.microsoft.com/sqlserver/en/us/default.aspx (accessed December 16, 2012).
10. "NYISO LBMP ZONES." NYISO. www.eei.org/meetings/Meeting%20Documents/NYISOnelson.ppt (accessed December 15, 2012).
11. "Locational Based Margin Price (LBMP)." LBMP. www.nyiso.com/public/services/market_training/online_resources/lbmp_online.pdf (accessed December 16, 2012).
12. Bai, Ying. *Practical database programming with Visual Basic.NET*. New York: Cambridge University Press, 2009.
13. Fratarcangeli, Claudio. "Technique for universal quantification in SQL." *ACM* 20, no. 3 (1991): 16-24. http://dl.acm.org/citation.cfm?id=126482.126484 (accessed December 15, 2012).
14. Joachim Klein, Hans. "Null Values in Relational Databases and Sure Information Answers." In *Semantics in databases: second international workshop, Dagstuhl Castle, Germany, January 7-12, 2001 : revised papers*. Berlin: Springer, 2003. 119-138
15. Kawash, Jalal. "Complex quantification in Structured Query Language (SQL): A tutorial using relational calculus." *Journal of Computers in Mathematics and Science Teaching* 23, no. 2 (2003): 169-190.

16. First Look at the New C++ IDE Productivity Features in the Next Version of Visual Studio." MSDN Blogs. http://blogs.msdn.com/b/vcblog/archive/2011/08/24/10200097.aspx (accessed December 15, 2012).
17. "Windows Form Chart Control." MSDN . http://msdn.microsoft.com/en-us/library/system.windows.forms.datavisualization.charting.aspx (accessed December 15, 2012).