

---

# Reffing Pro Assistant

---

Final Project Write-up  
MBA 614  
Section 1

---

Authors:

David Long  
Adam Richards

Date:

11 April 2012

---

## CONTENTS

|  |    |
|--|----|
| Executive Summary .....  | 1  |
| Implementation/User's Guide .....                                    | 1  |
| Starting the application .....                                       | 1  |
| Principal Dashboard.....   | 1  |
| Settings .....   | 2  |
| Scheduler .....  | 4  |
| Edit Events.....   | 5  |
| Calculate Mileage .....  | 7  |
| Record Results.....  | 9  |
| Print Reports .....  | 9  |
| Admin Mode .....   | 12 |
| Future Enhancements.....   | 12 |
| Discussion of Learning and Conceptual Difficulties Encountered ..... | 12 |
| Assistance .....   | 14 |

## EXECUTIVE SUMMARY

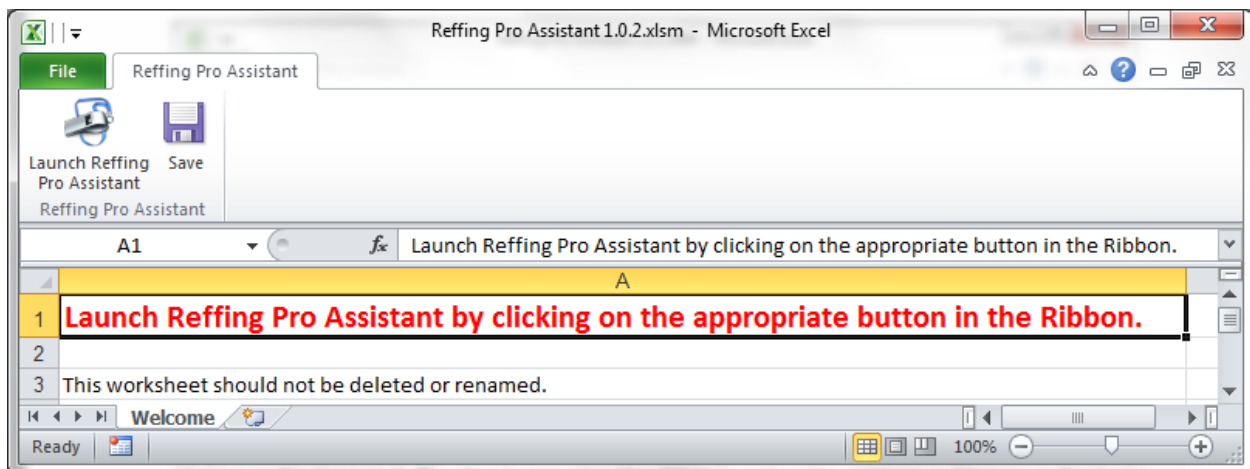
We have created a program to be used by soccer referees. The program will aid referees by tracking match details such as: date, location, mileage, position (center, AR, etc.), league/level (e.g., Boys Under 18, Adult Division III, etc.), money received, and comments. Additionally, the program allows referees to print off match report information cards before games and facilitates online reporting systems by automatically logging the referee onto the web based reporting system. Future versions will further automate the reporting process. The heart of the program is the calendar/scheduler functionality that allows a referee to manage all of his assignments in one place as well as provide him the ability to block out time when he will be unavailable.

The following sections provide an implementation/user's guide and then the end of the document will discuss learning points as well as future enhancements.

## IMPLEMENTATION/USER'S GUIDE

### STARTING THE APPLICATION

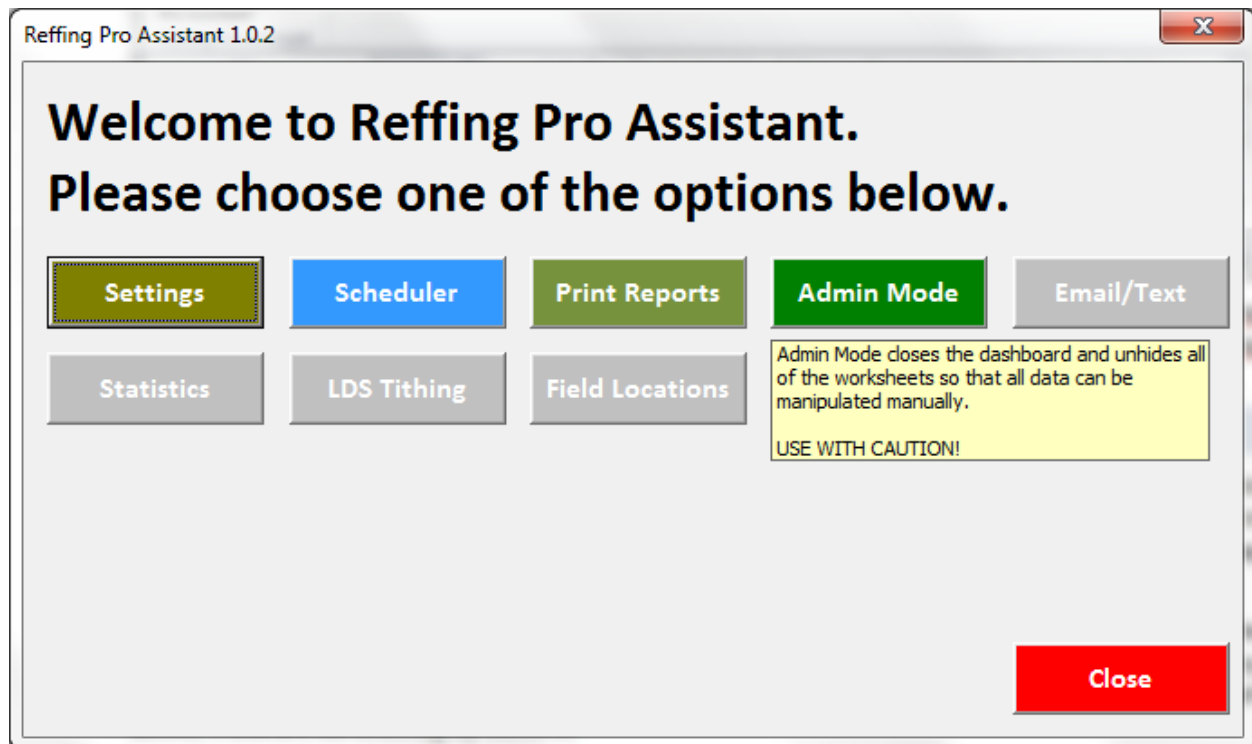
When the user opens the Reffing Pro Assistant.xlsm file he is created by the following Welcome worksheet as shown below.



The file has a custom UI xml that hides most of the ribbon elements except a new tab titled **Reffing Pro Assistant**. Additionally, all of the worksheets are hidden except the **Welcome** worksheet. The user can launch the principal dashboard userform by clicking on **Launch Reffing Pro Assistant** in the ribbon which is denoted by the whistle icon.

### PRINCIPAL DASHBOARD

The dashboard userform is the control hub for the Reffing Pro Assistant program. It is from this userform that the user can select various options and functions of the program that will be explained in detail below. One item of note is that features that are not currently supported are denoted by gray buttons. See below for a screenshot of the userform. NOTE: The yellow label below the Admin button is only visible when a user places the mouse over the **Admin Mode** button.



The next sections explain each of the buttons on the principal dashboard userform in more detail.

## SETTINGS

The **Settings** button calls a userform where the referee can input his personal information. This information is used by other parts of the program which will be further explained in those sections as appropriate. The next page contains a screen shot of the **Settings** userform.

The screenshot shows a 'Settings' window with three main sections: Personal Information, League Information, and Miscellaneous.

**Personal Information:**

- Referee Name: [Text Field]
- Email Address: [Text Field]
- Cell Phone: [Text Field]
- Cell Phone Carrier: [Dropdown Menu] (04 - Verizon)
- Address (include APT # if applicable): [Text Field]
- City: [Text Field] (Provo)
- State: [Dropdown Menu] (UT)
- ZIP Code: [Text Field] (84604)
- USSF Referee ID: [Text Field]
- USSF Website Login ID: [Text Field]
- USSF Website Password: [Text Field] (\*\*\*\*)
- Login to USSF Site** (Green Button)

**League Information:**

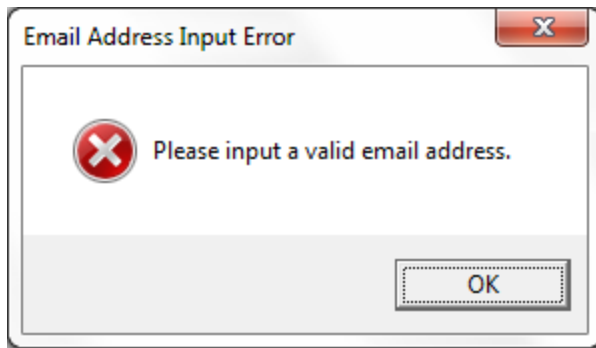
- Youth League Website: [Text Field] (http://uysa.affinitysoccer.com/reg/index.asp?sessionguid=)
- Youth League Username: [Text Field]
- Youth League Password: [Text Field] (\*\*\*\*\*)
- Login to Youth League Site** (Blue Button)
- Adult League Website: [Text Field]
- Adult League Username: [Text Field]
- Adult League Password: [Text Field]
- Login to Adult League Site** (Brown Button)

**Miscellaneous:**

- ☐ Track LDS Tithing?

The user can access his United States Soccer Federation (USSF) official account by inputting the appropriate information and clicking the **Login to USSF Site**. Similarly, the user can access the Utah Youth League website by clicking on **Login to Youth League Site**. The **Login to Adult League Site** button is not yet functional, because the Utah Adult League website is not yet up and running. The **Track LDS Tithing?** checkbox indicates whether the system will hide or show the **Tithing Paid?** checkbox on the **Edit Events** userform. The Miscellaneous section would ultimately include more information needed to send emails and text messages.

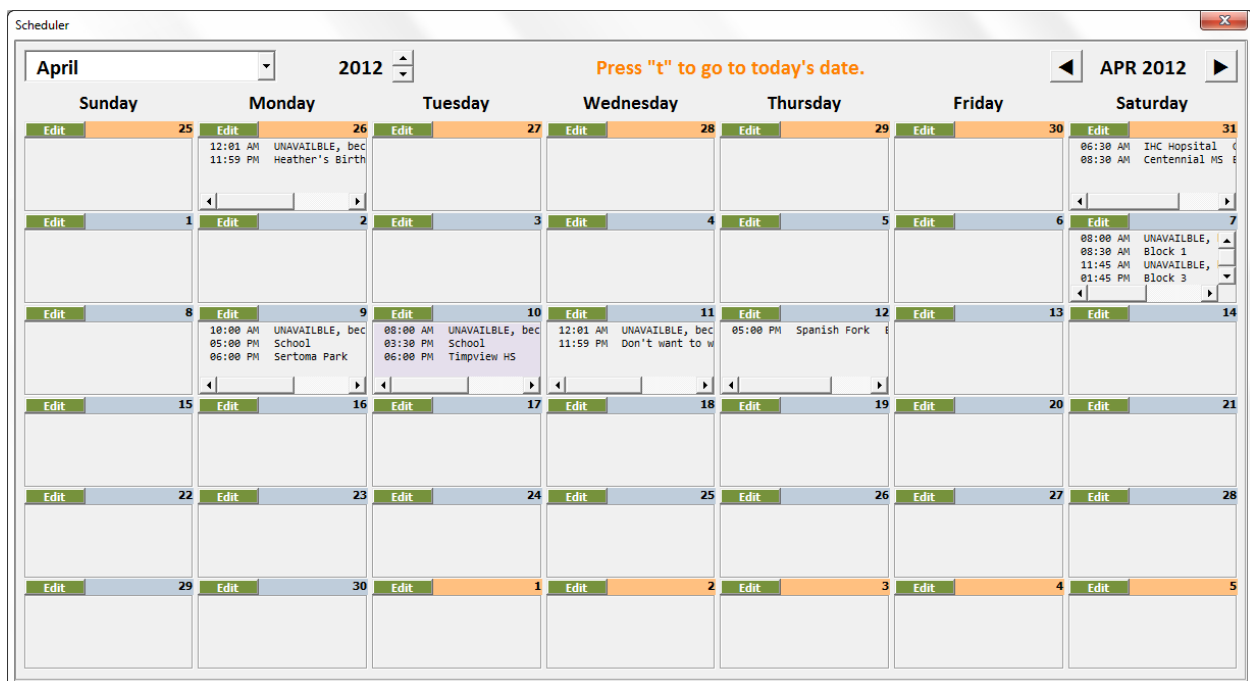
All of the information on this userform is read from and written to the **Settings** worksheet. Information is saved once a user exits a specific field as long as the data passes specific validation relevant to the field. Furthermore, some fields are formatted upon exit to ensure standardization (e.g., Phone Number: (###) ###-####). See below for a sample validation error message.



Closing the **Settings** userform returns the user to the principal dashboard.

## SCHEDULER

Clicking **Scheduler** launches the **Scheduler** userform which is comprised of a calendar. When the userform is launched it will default to the current date's month. See below for a screenshot.



The userform's information is populated via 42 labels that display the day of the month and 42 listboxes that display applicable information for each of the dates as appropriate. This information comes from information worksheets (one per calendar year) that the system creates as needed.

The current date is denoted by a light purple background. Additionally, the background color for the label that shows the day of the month is blue for dates that reside within the month shown in the combobox in the top left form of the userform while it is orange for dates that fall before or after the specified month.

As stated on the userform the user can return to the current date's month and year at any time by simply pressing **t** on the keyboard. The user can also manipulate the date using the combobox in the top left to change months and the spin buttons to increase or decrease the year. Alternatively, the user

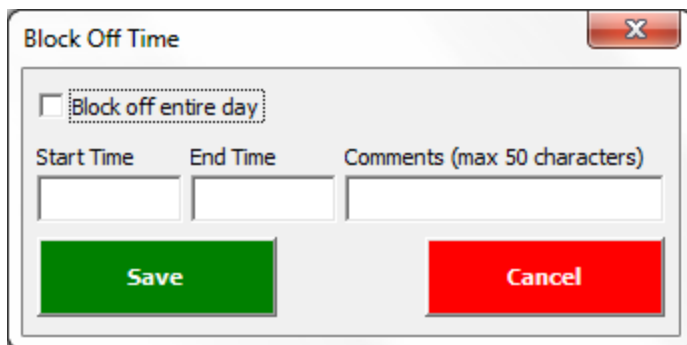
can move backwards or forwards one month at a time by using the command buttons in the top right of the userform. NOTE: January 2012 is the earliest Month/Year allowed by the userform.

At the top of each day in the calendar is an **Edit** button (technically it is a label) that can be used to manage the events and information for any specific date.

## EDIT EVENTS

Clicking on **Edit** launches the **Edit Events** userform. This userform is the backbone of the program. A screenshot is shown below.

The left side of the userform shows the currently scheduled events (e.g., games) for that day along with buttons to remove an event, create an event, and block off time as unavailable. If time is blocked out it is entered as 2 events. The first entry shows the start time with the text **UNAVAILABLE, because:** while the second entry shows the ending time along with the reason the referee is unavailable. When blocking off time all other fields are grayed out and the user is unable to add further event details. Blocking off Time is accomplished by clicking on **Blocking Off Time** which launches the **Block Off Time** userform as shown below.



The 'Block Off Time' dialog box features a title bar with a close button (X). Inside, there is a checkbox labeled 'Block off entire day'. Below this, there are three input fields: 'Start Time', 'End Time', and 'Comments (max 50 characters)'. At the bottom, there are two buttons: a green 'Save' button and a red 'Cancel' button.

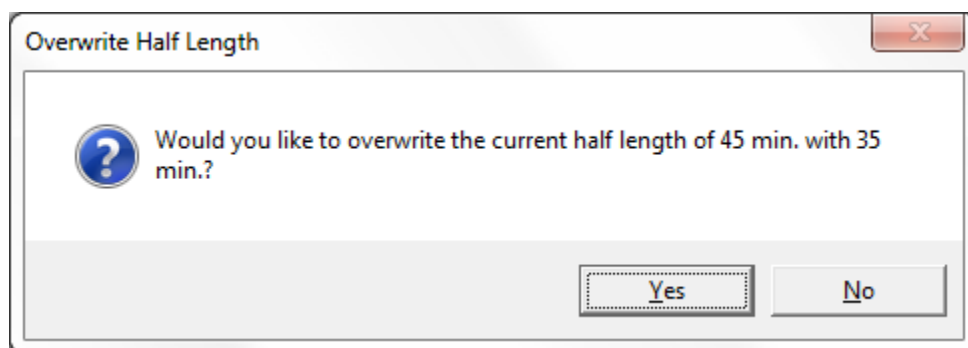
Checking the **Block off entire day** checkbox will populate the **Start Time** with **12:01 AM** and the **End Time** with **11:59 PM**. If this option is not checked the user can input a start and end time; however as the user exits the time fields data validation ensures that the values are indeed times. When the user clicks **Save** the form is validated to make sure that:

- **Start Time**, **End Time**, and **Comments** are not blank.
- The **Start Time** is after **12:00 AM** and before the **End Time**.

The right side of the **Edit Events** populates with the event details and allows the user to modify any of the fields. The **Add/Update** button updates the values in the appropriate worksheet or adds new information as applicable. If this button is not pressed data will not be saved to the appropriate worksheet.

To remove an event the user highlights the appropriate event in the **Currently Scheduled Events** listbox and presses the **Remove Event** button. If the selected item is part of a block off time event the system will remove both items from the listbox.

Data validation occurs throughout the userform. Like the **Block Off Time** userform the **Time** field's value is validated to ensure it is a time. The **Level** combobox contains default values that can bring in other information as noted below. However, a user can type in whatever is required in case none of the default options work. The **Position** combobox is restricted to the values in the list. Once both the **Level** and **Position** comboboxes have been populated **Halves (min)**, **Ball Size**, and **Fee** will default in from the **Settings** worksheet if there are values associated with the **Level** and **Position**. Currently, the user must use **Admin Mode** to edit the default values on this worksheet. If data exist in the previously



The 'Overwrite Half Length' message box has a title bar with a close button (X). It contains a question mark icon and the text: 'Would you like to overwrite the current half length of 45 min. with 35 min.?' At the bottom, there are two buttons: 'Yes' and 'No'.

mentioned fields then a series of Yes/No message boxes will prompt the user to indicate whether to overwrite the existing values or not. See above for a sample message box.

The location fields require the combobox items to dynamically change based upon the values in the **State** and **City** fields. When the userform is first opened or a new event is created by clicking on **Create New Event** the Referee's home state and city as input on the **Settings** userform default into the **State** and **City** fields respectively. The **City** combobox values are dependent upon the **State** combobox and



the **Location** field is dependent on the **State** and **City** fields. If the **State** field is blank then all possible cities on the **FieldLocations** worksheet are displayed in the **City** field. The **Location** combobox has similar functionality with respect to the **State** and **City** comboboxes. When a user exits the **Location** combobox data validation ensures that the user has selected a value from the list. See screenshot to the right for a sample of what the **City** combobox looks like when it has a series of cities populated in it.

If a new location needs to be added the user simply clicks on the **Add New Location** button which brings up the userform shown below.

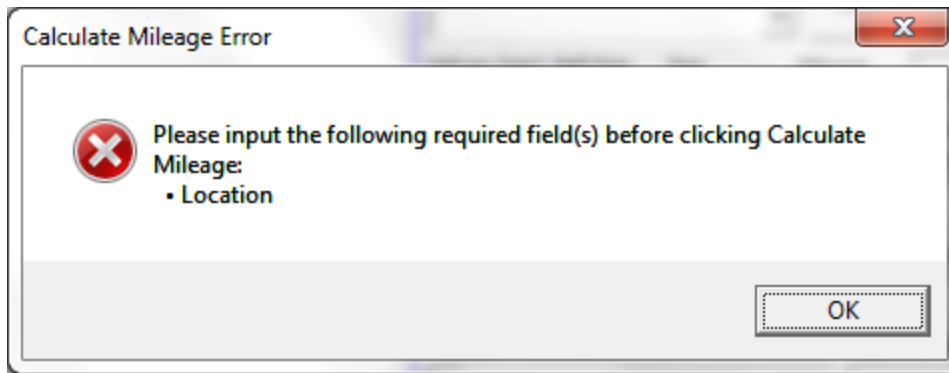
It is important to note the the user only needs to do this once for each location. When the user clicks **Save** the values are validated to ensure:

- **Field Location Name, Address, City, and State** are not blank
- The **Field Location Name** must be unique from all existing names in the database. If the name already exists the user is asked if he wants to overwrite the existing address information with the information in the **Add New Field Location** userform as long as the address are different. If the name and address are the same an error message is displayed to the user.
- Similiary if the new address matches an existing entry then the user if prompted to overwrite the old **Field Location Name** with the new one.

Reffing Pro Assistant can also calculate the mileage between two addresses. Mileage is tracked for tax purposes.

## CALCULATE MILEAGE

To calculate the milegae the user presses the **Calculate Mileage** button on the **Edit Events** userform. If the user tries to calculate mileage without first providing the game location the program displays the following message box.



If the necessary information has been provided the **Calculate Mileage** userform opens. See screenshot below.

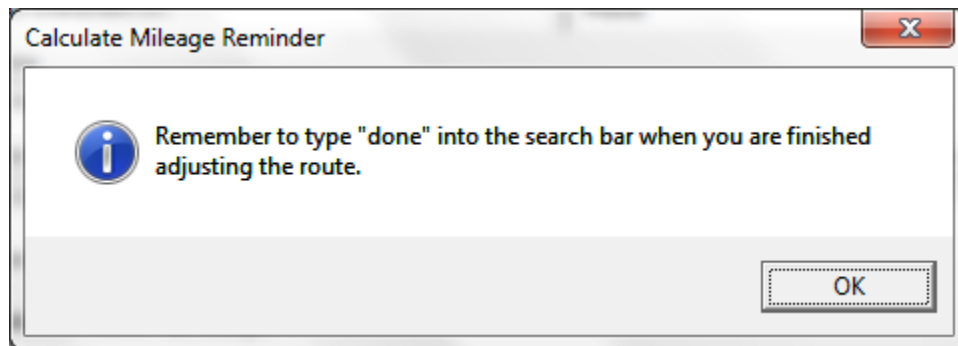
When the userform is initialized it loads the address of the prior event into the **From** fields or if the prior event is a blocked off event /does not exist then the referee's home address information from the **Settings** userform is loaded. The user can select any of the address already into the database via the **From Name** combobox. Additionally the user can manually input a from address into the appropriate fields.

The user can force a recalculation using Google Maps by selecting the **Force Internet Map Calculation** checkbox. The user may also calculate the round trip mileage by selecting the **Calculate Round Trip Mileage?** Checkbox.

When the user presses the **Calculate Mileage** button the following validation occurs:

- **From Address, City, and State** must all be populated.
- The **From Address, City, and State** and the **To Address, City, and State** fields can not be identical.

The first time that mileage is calculated between two points the program uses Google Maps to calculate the distance. After clicking **Calculate Mileage** the following message box opens.



After Google Maps calculates the route and the user has adjusted the route to his preference he types “done” into the search bar. At this point he will be returned to the **Edit** Event userform. The distance calculated is stored in the **Mileage** worksheet within the program for use in the future. The system stores the from/to addresses and mileage in both directions.

## RECORD RESULTS

Reffing Pro Assistant will eventually include the ability to upload the games results to the league webpages. Currently only the youth league webpage is up and running. The adult league’s website is due to be completed later this year. For that reason, the system only connects with and logs into the youth league site when the **Record Results** button is pressed if the **Level** field corresponds with a youth league match as indicated on the **Settings** worksheet. Currently the relationships between levels and youth/adult games can only be updated in **Admin Mode**.

NOTE: For some reason this connection works on some computers but fails on others. We were not able to figure out what caused the problem.

## PRINT REPORTS

The user can print off match reports for upcoming games by clicking on **Print Reports** on the principal dashboard userform. Clicking the button launches the userform shown on the next page.

**Print Match Reports**

**List of Games to Referee**

Select All Deselect All

|                          | DATE | TIME | LOCATION | LEVEL/LEAGUE | POSITION |
|--------------------------|------|------|----------|--------------|----------|
| <input type="checkbox"/> |      |      |          |              |          |

**List Details & Print Info**

Start Date: 04/10/2012 End Date: 04/17/2012 **Generate List**

Minimum Number of Blank Reports to Print: 0

**Print Match Reports**

**Close**

When the userform initializes the **Start Date** is populated with today's date and the **End Date** is populated one week from the **Start Date**. Like other forms, the **Start Date** and **End Date** fields are validated to ensure that only date values are input into the fields. Additionally, when the user enters the **Start Date** or **End Date** fields a label will become visible indicating that the letter **t** can be used to input today's date. The system converts the letter **t** into today's date as appropriate for those two fields.

When the user clicks on the **Generate List** button the following validation occurs:

- **Start Date** and **End Date** must both be populated.
- The **End Date** must be greater than the **Start Date**.
- The **End Date's** year cannot be more than one year greater than the **Start Date's** year. This is because, currently the system will only populate data from two **MatchDetails** worksheets and each year has its own **MatchDetails** worksheet.

If the above validation is successful then the **List of Games to Referee** listbox is populated with the appropriate matches and is sorted in chronological order. The information for this listbox is first written to and then read from the **PrintMatchReports** worksheet using the listbox's RowSource property. Additionally, a label at the bottom of the listbox will become visible that indicates how many games in total were found.

The user can then manually indicate the games for which he wants to print match reports by Ctrl/Shift clicking items in the listbox or alternatively using the **Select All** and **Deselect All** buttons above the listbox.

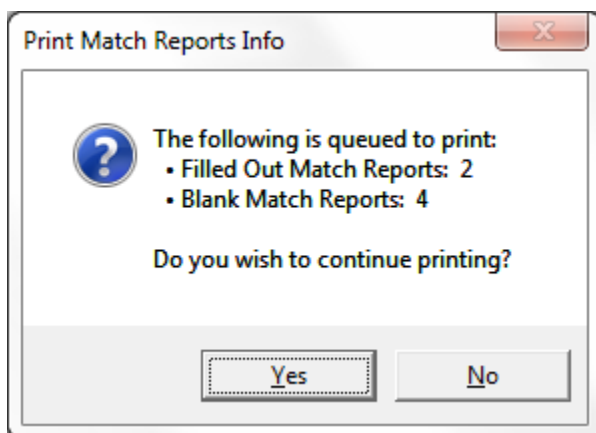
Next, the user indicates the minimum number of blank match reports he would like printed via the appropriate combobox. This combobox validates that an item was selected and if a valid item was not selected resets the combobox's value to 0 upon exiting from the combobox field. The user can only

specify the minimum number of blank reports to print, because two match reports print per page, and when this information is combined with the number of matches to print it may increase the number of blank match report forms by one.

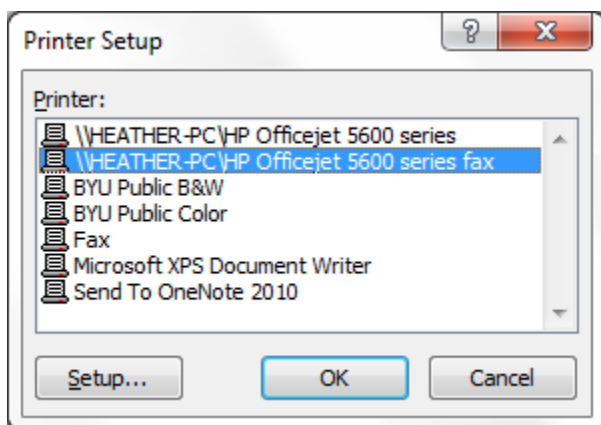
Once the user has indicated the matches that he would like to print as well as how many blank reports he would the system to print he clicks on **Print Match Reports**. The system then first performs the following validation:

- At least one match must be selected in the **List of Games to Referee** listbox or the **Minimum Number of Blank Reports to Print:** combobox must be greater than 0.

If the above validation is successful the system determines the number of blank match reports that will be printed. If the sum of the number of match reports to be printed from the **List of Games to Referee** listbox and the **Minimum Number of Blank Reports to Print:** combobox is odd then the system will automatically increase the number of blank reports by one. Once the system has determined the number of match reports to print a message box is displayed with the results and asks the user if he would like to continue. See below for a sample message box.



If the user answers yes to the message box question then the **Printer Setup** dialog is displayed. See below for an example.



If the user selects a valid printer and presses the **OK** button the system disables ScreenUpdating, unhides the **RefereeMatchReport** worksheet and copies information from the **PrintMatchReports** worksheet for all of the selected matches into the appropriate fields on the **RefereeMatchReport** and prints off the expected number of pages. Each time a page is printed the system clears the fields on the

**RefereeMatchReport** worksheet so that it is ready to accept new values or print off blank match reports.

Clicking the **Close** button closes the **Print Match Reports** userform and returns the user to the principal dashboard userform.

## ADMIN MODE

From the principal dashboard if the user clicks on **Admin Mode** the userform closes and all of the worksheets are unhidden. Remember to use this feature with caution! This mode allows the user to manually manipulate any of the information in the database.

## FUTURE ENHANCEMENTS

As noted earlier there are a few buttons on the principal dashboard userform that are not yet supported. Below is a brief discussion of these items and other potential future enhancements.

- **Email/Text:** This will provide a way to email and/or text game information. This will require a userform similar to **Print Match Reports** as well as for additional **Settings** information to be captured.
- **Statistics:** This will provide a way to view information such as year to date income, mileage, etc. as well as have the ability to compare months and year data.
- **LDS Tithing:** This userform would provide a way to get a list of games where the user has not paid tithing and the outstanding total tithing amount as well as provide the ability to indicate when tithing has been paid on those matches.
- **Field Locations:** This will provide an interface to better manage the **FieldLocations** worksheet in order to add, remove, or modify existing information in the database.
- **Record Results:** This will hopefully launch another userform where the match details can be input in order to then open internet explorer, log onto a secure website and submit a match report for the appropriate game as indicated by the game number.
- **Settings:** This screen could be enhanced to better handle generic league information (especially logging onto websites, etc.) and would also need to hold more information in order to make the **Email/Text** functionality work correctly.
- **Phasing Out the Principal Dashboard with Ribbon buttons:** We believe we could modify the program to use more buttons in the ribbon instead of relying on the principal dashboard for navigation.

## DISCUSSION OF LEARNING AND CONCEPTUAL DIFFICULTIES ENCOUNTERED

We learned a lot while working on this assignment. However, we did not get everything done that we had originally planned. With respect to the original scope we were not able to get the information connecting to a SQL database and we also could not completely submit the match reports as intended. The online match reports was due to the fact that David Long could not be assigned matches and spoke with Professor Allen about it. However, we did additional functionality of being able to print off match reports and made the Google Maps functionality more interactive and better than originally planned. We were not able to connect with Professor Allen due to his limited availability the last two weeks of the semester. Likewise, we were able to get the Internet Explorer automation working on one machine all of the time, but on others we would get a run-time error of 70 or 91 occasionally which we were not able to fully track down. Below you will find a discussion of learning points by topic.

## VBA Events

We learned a great deal about events and when they fire especially with respect to userforms. This enabled us to create validation and formatting routines as well as to implement some keyboard shortcuts with the letter **t**.

Additionally, we were able to solve a strange issue with the userform **Activate** event. One of our userforms' **Initialize** event fired before we knew exactly what data need to be populated on the userform, so we used the **Activate** event to populate the fields. However, after implementing the calculate mileage Internet Explorer automation we noticed that occasionally we were getting unexpected behavior on the form. When we stepped through the code the **Activate** event never fired, but by putting message boxes in the code we were able to find when the **Activate** event was firing and correct the problem.

## ListBoxes

We learned a lot about how listboxes behave such as the IntegralHeight property which was causing havoc with the **Scheduler** userform for awhile. Additionally, we learned about multiple columns in listboxes and how to read/write that data even from columns that have a width of 0. Additionally, we learned how to interact with listboxes that allow multiple items to be selected.

## Data Validation and Data Formatting

We learned a lot about how to validate data and format it so it looks uniform. For example, we were able to accept phone numbers of any type and then format the output so that no matter how it was input it would come out as (###) ###-####. The most difficult one we dealt with was the validation for time fields, because it involved a complex Regular Expression. We came across Regular Expressions when searching for an email validation, but were able to take what we found without too much modification. However, the time validation required some small, but significant modifications to the Regular Expression we found.

## Arrays

We learned a lot more about how to use arrays and especially sort the values in them. We were able to find a QuickSort (QSort) algorithm that would sort our multi-dimensional array for us by recursively calling itself. The QSort was used in multiple places throughout our project and allowed us to do sorting without having to write values to a worksheet and then use Excel's native sort functionality.

## Internet Explorer Automation

We learned a lot more about how to manipulate Internet Explorer and how to use the .Show method to expose the source code as well as how to use Google Chrome's Inspect Element feature to see changes in realtime. For example, we were having an issue figuring out which one of the up to three selected routes the user had clicked on, but using the Inspect Element feature tipped us off to what the code could look for in the html to know what had been selected. We became more familiar with the agent class and how to suspend Internet Explorer code until a specific html value was encountered (more about this below in the Assistance section).

## Print Automation

We learned that you can read and write from hidden worksheets, but that in order to print from a worksheet it needed to be visible. We also learned about the built-in **Print Setup** dialog that we were able to implement with our project.

## Class Properties

We also learned a lot about how to create, set and read class properties. This is something that the textbook mentioned in lieu of using public/global variables. We were able to do the project without needing any public/global variables but instead passed arguments to procedures and we created userform class module properties in order to make sure that the userforms behaved as expected.

## ASSISTANCE

We spoke with Professor Allen on a few occasions about the project and used the guidance he gave. We probably spent about 20 minutes with Professor Allen. Specifically, Professor Allen showed us how to create a loop in our code that would check for a specific value in the html before exiting the loop. This proved invaluable in getting the Google Maps functionality to work by allowing the user to type the word “done” into the search bar in order to indicate to the code that it could proceed. All of the other significant research was done with the textbook and our good friend Google.