DevOps Kata

# Introduction to Modern Debugging with Visual Studio Enterprise 2015

Last updated: 10/07/2016



1. **TABLE OF CONTENT**

[Introduction to IntelliTest with Visual Studio Enterprise 2015 1](#_Toc463630694)

[Overview 3](#_Toc463630695)

[Prerequisites 3](#_Toc463630696)

[Exercises 3](#_Toc463630697)

[Exercise 1: Introduction to modern debugging 4](#_Toc463630698)

[Task 1: Run the Application 4](#_Toc463630699)

[Task 2: Using Intellitrace 8](#_Toc463630700)

## Overview

Traditional or live debugging shows only your application's current state, with limited data about past events. You either have to infer these events based on the application's current state, or you have to recreate these events by rerunning your application.

IntelliTrace expands this traditional debugging experience by recording specific events and data at these points in time. This lets you see what happened in your application without restarting it, especially if you step past where the bug is. IntelliTrace is turned on by default during traditional debugging and collects data automatically and invisibly. This lets you switch easily between traditional debugging and IntelliTrace debugging to see the recorded information.

IntelliTrace can also help you debug errors that are hard to reproduce or that happen in deployment. You can collect IntelliTrace data and save it to an IntelliTrace log file (.iTrace file). An .iTrace file contains details about exceptions, performance events, Web requests, test data, threads, modules, and other system information. You can open this file in Visual Studio Enterprise, select an item, and start debugging with IntelliTrace. This lets you go to any event in the file and see specific details about your application at that point in time.

### Prerequisites

* 1. In order to complete this exercise, you will need Visual Studio 2015 Enterprise Edition and the source code that you can find at the following link: <https://msdnshared.blob.core.windows.net/media/MSDNBlogsFS/prod.evol.blogs.msdn.com/CommunityServer.Components.PostAttachments/00/10/61/02/33/SocialClub.zip>

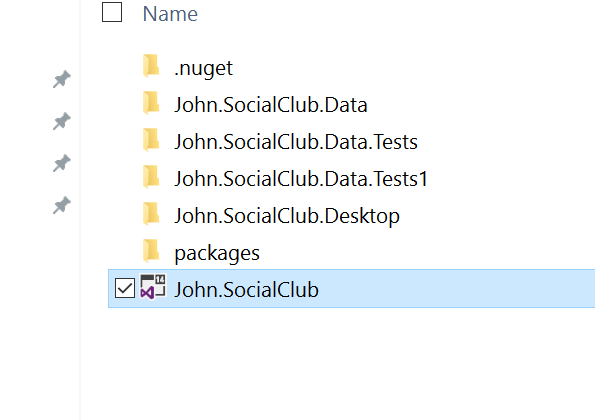
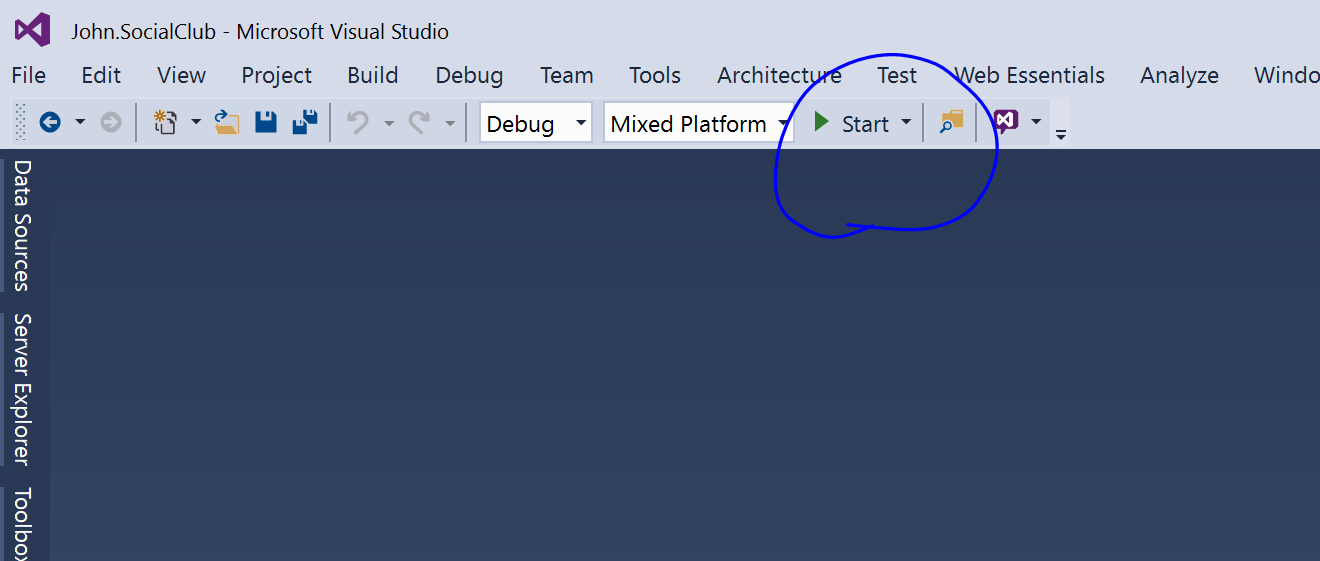
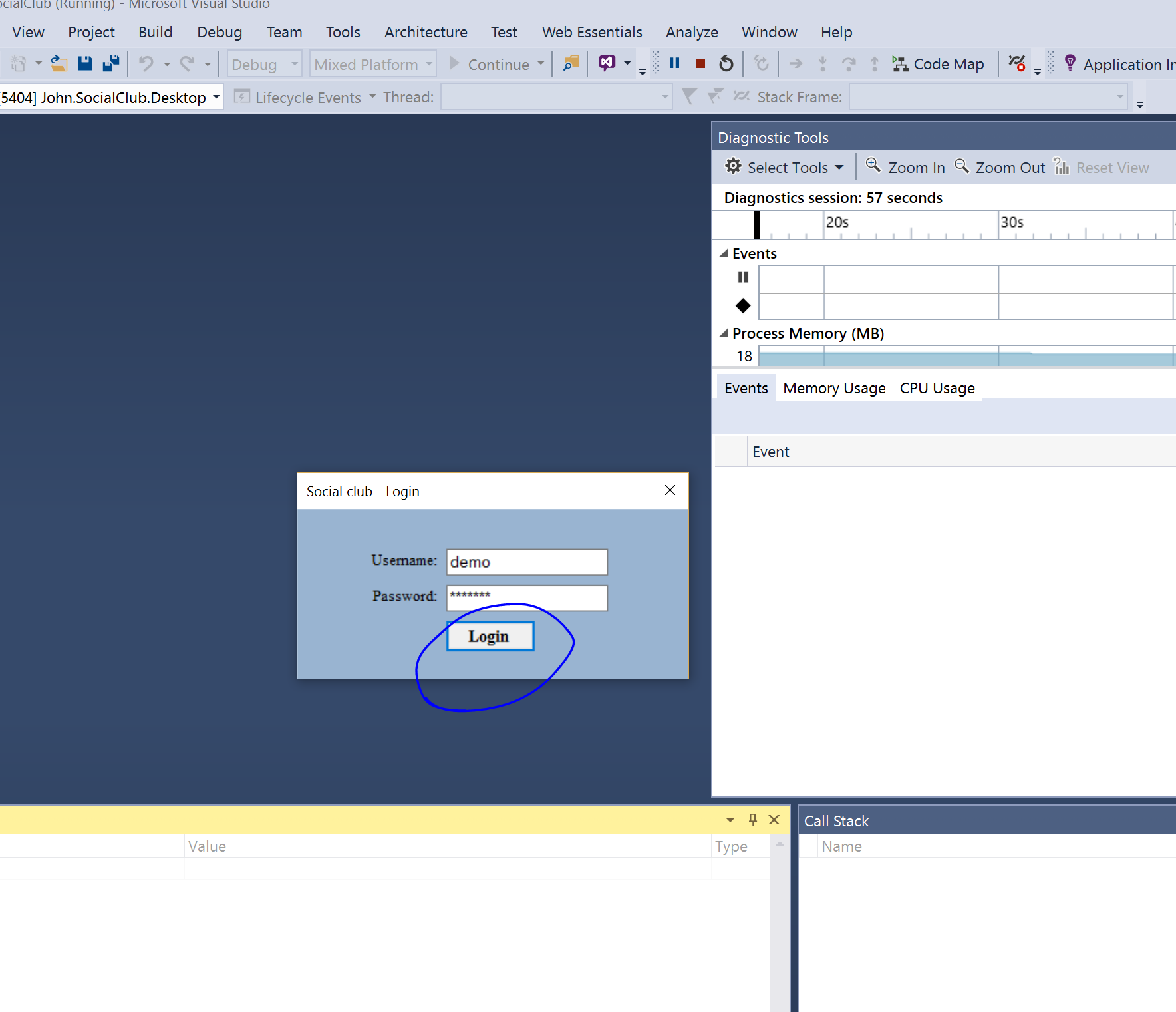
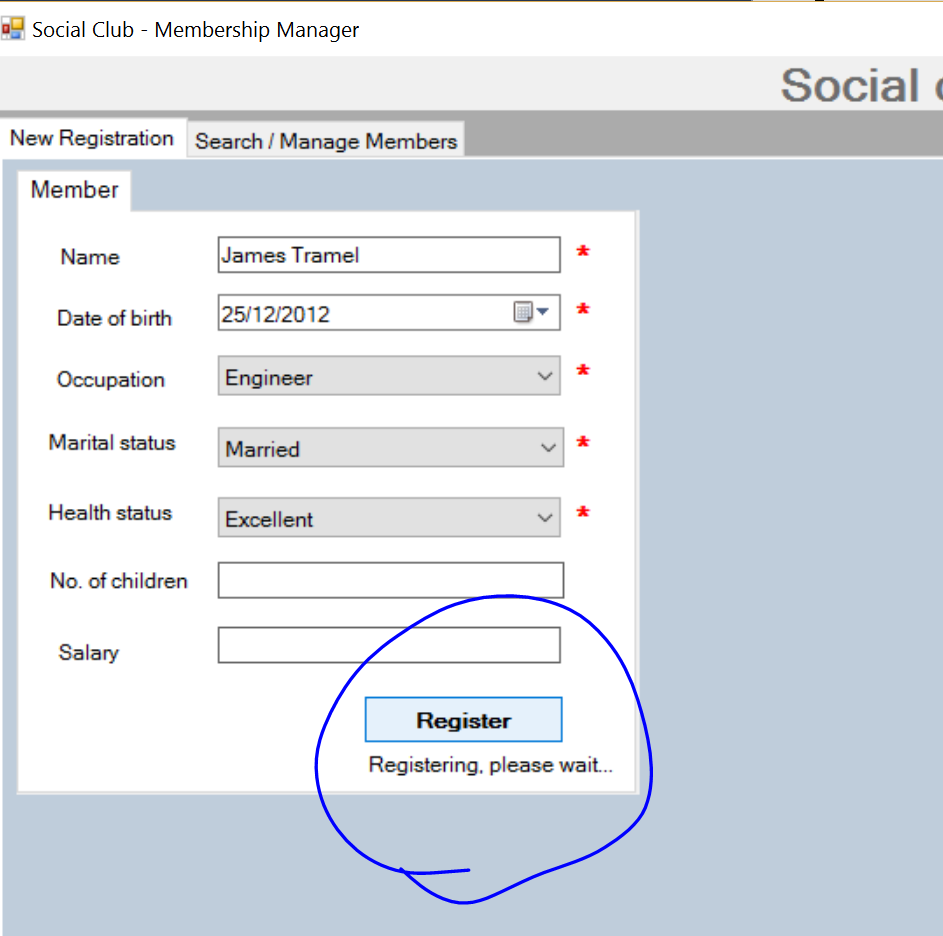
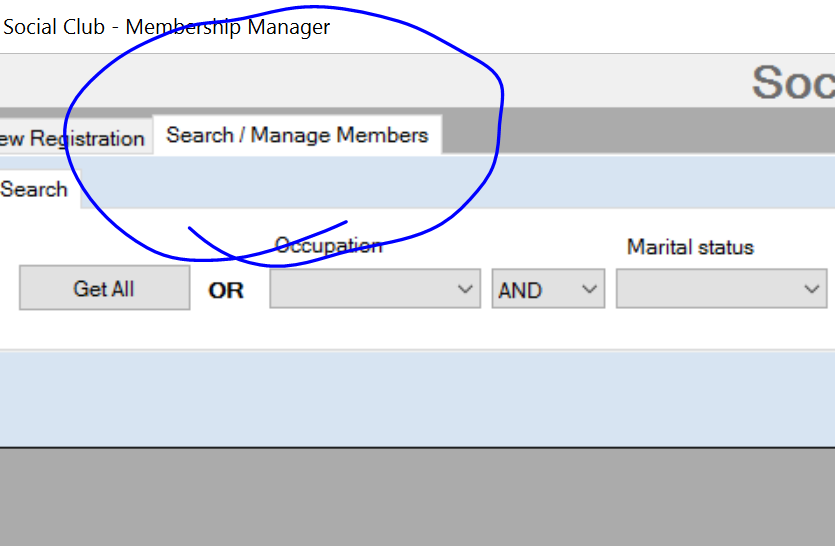
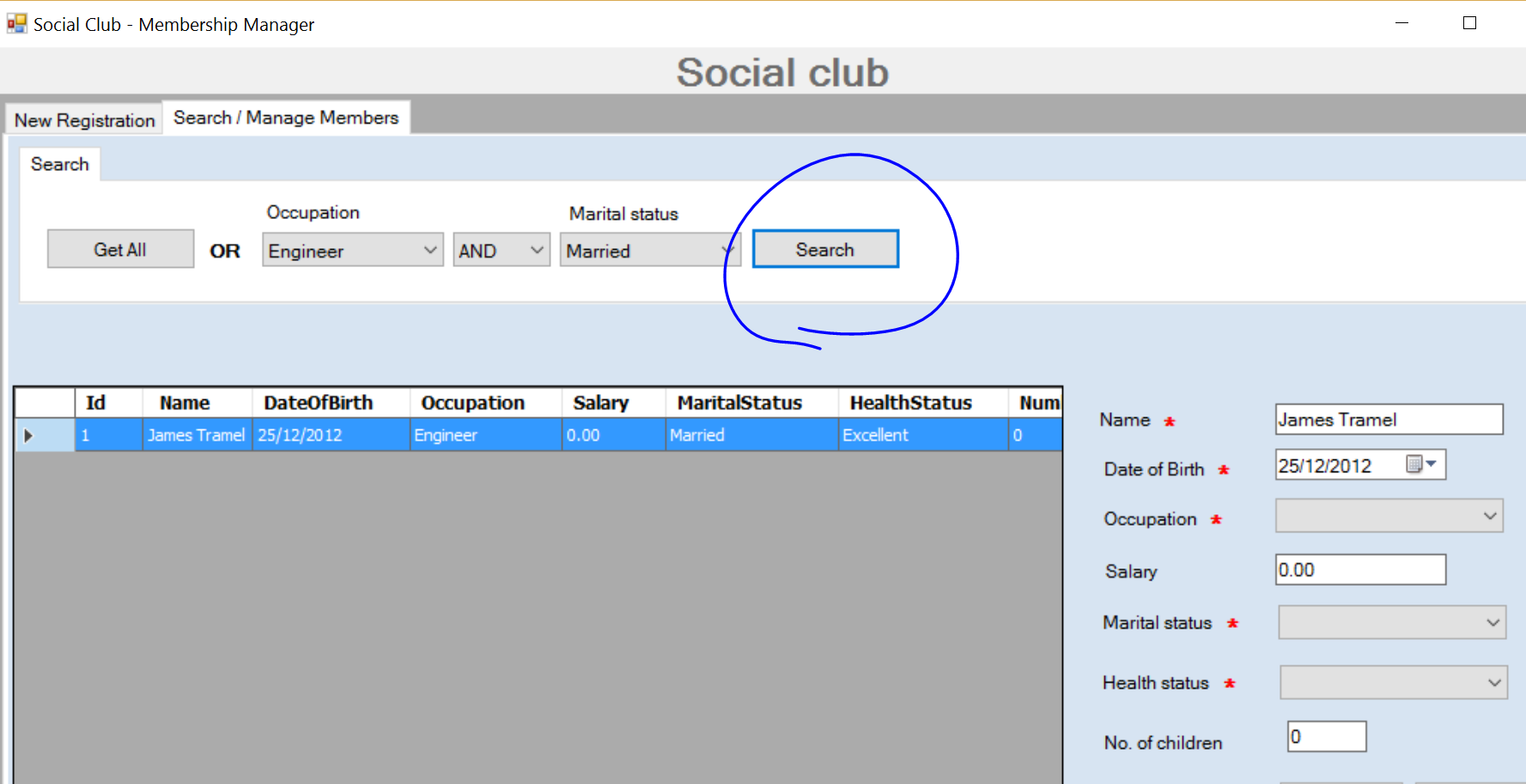
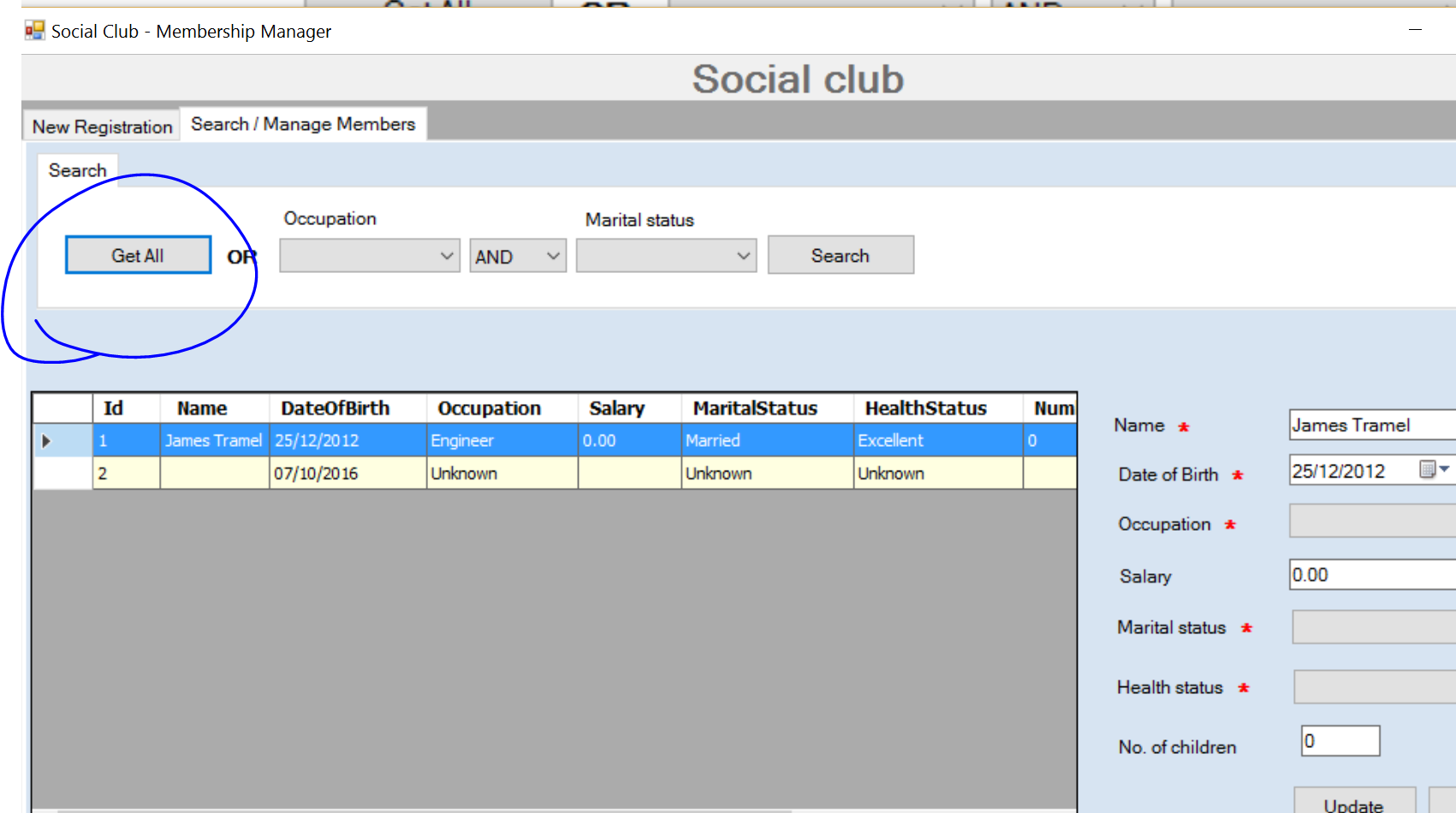
### Exercises

* 1. This hands-on lab includes the following exercises:
  2. Recreate the error while using the application
  3. Use IntelliTrace to track down the error
  4. Detail how you would resolve this issue, and send your fix to the instructor
  5. Estimated time to complete this exersize: **15 minutes**.

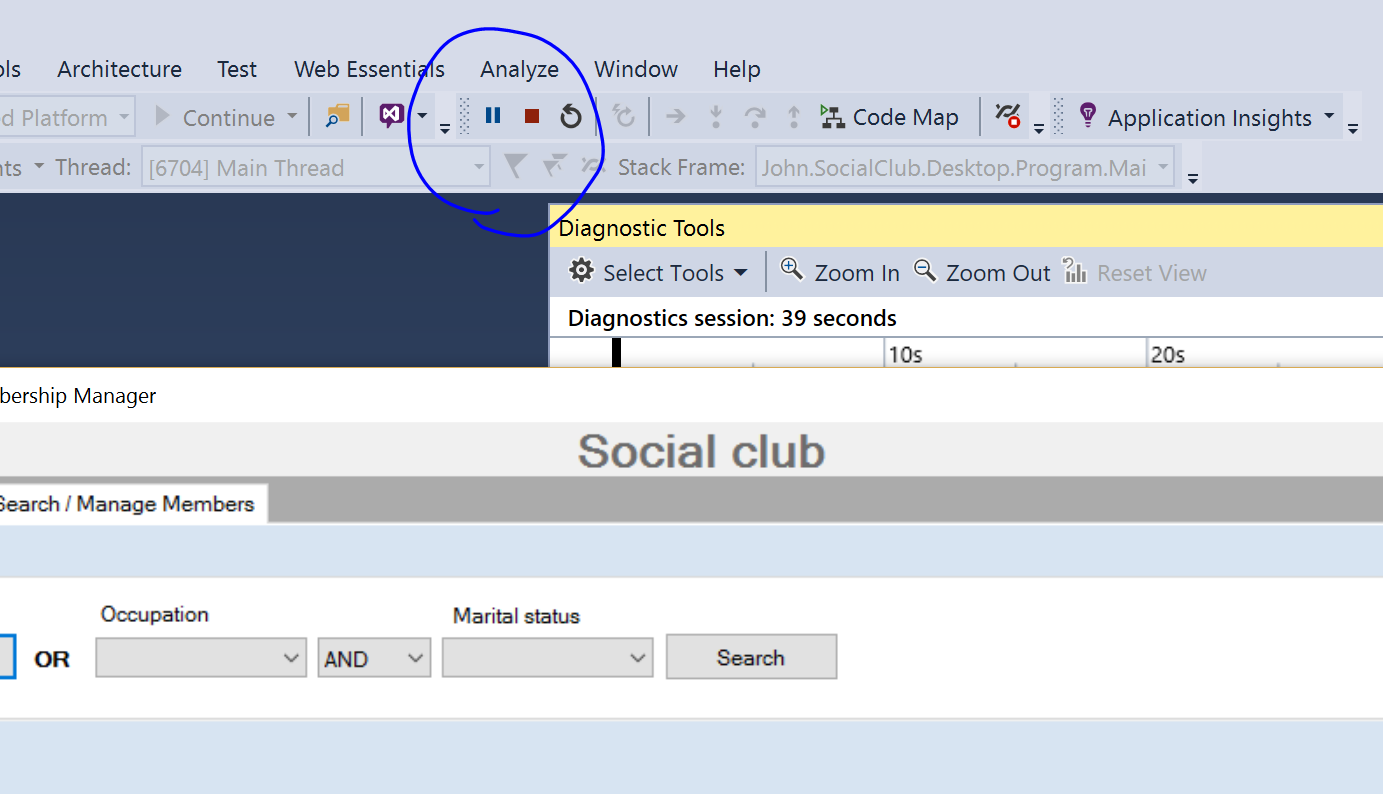
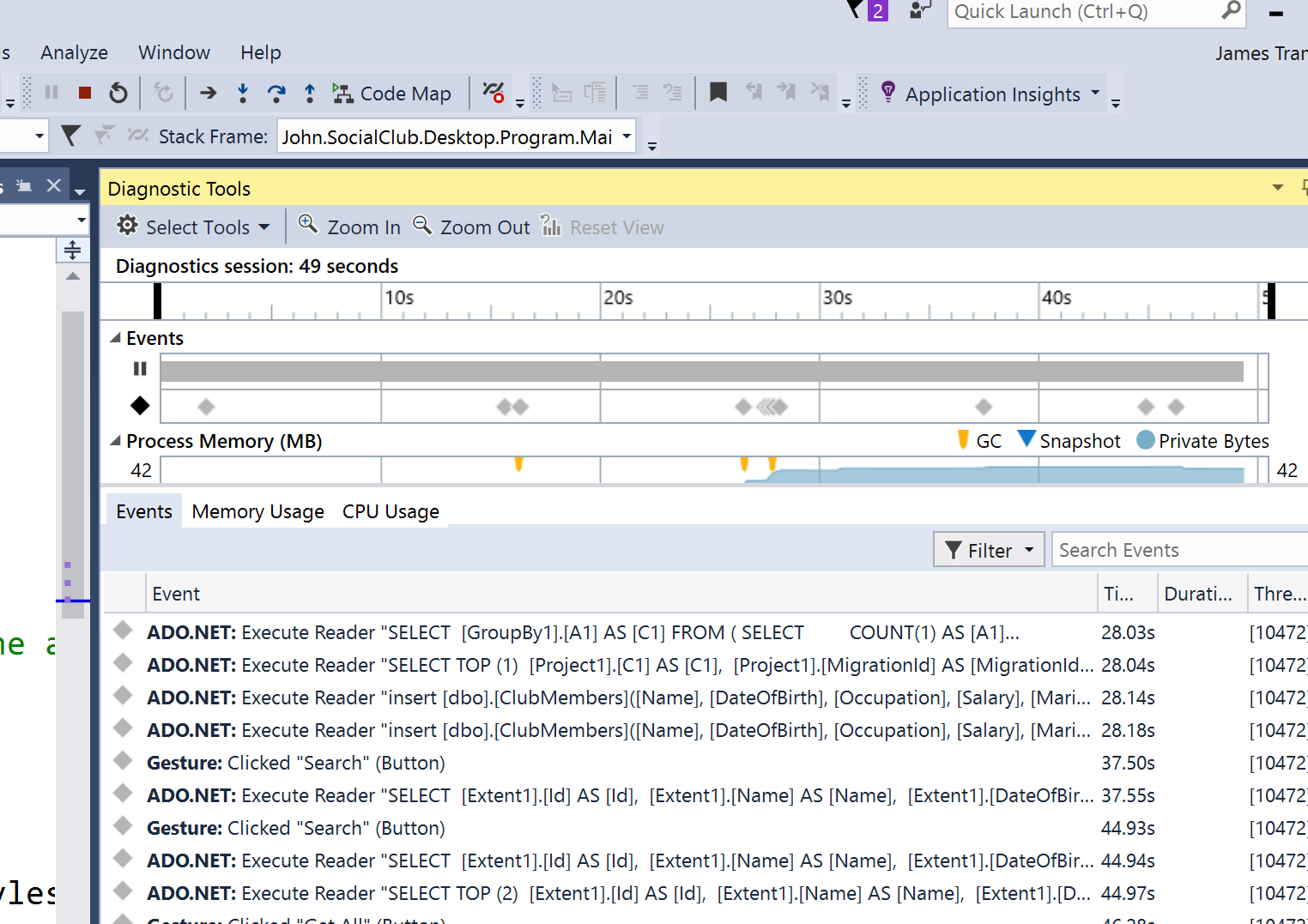
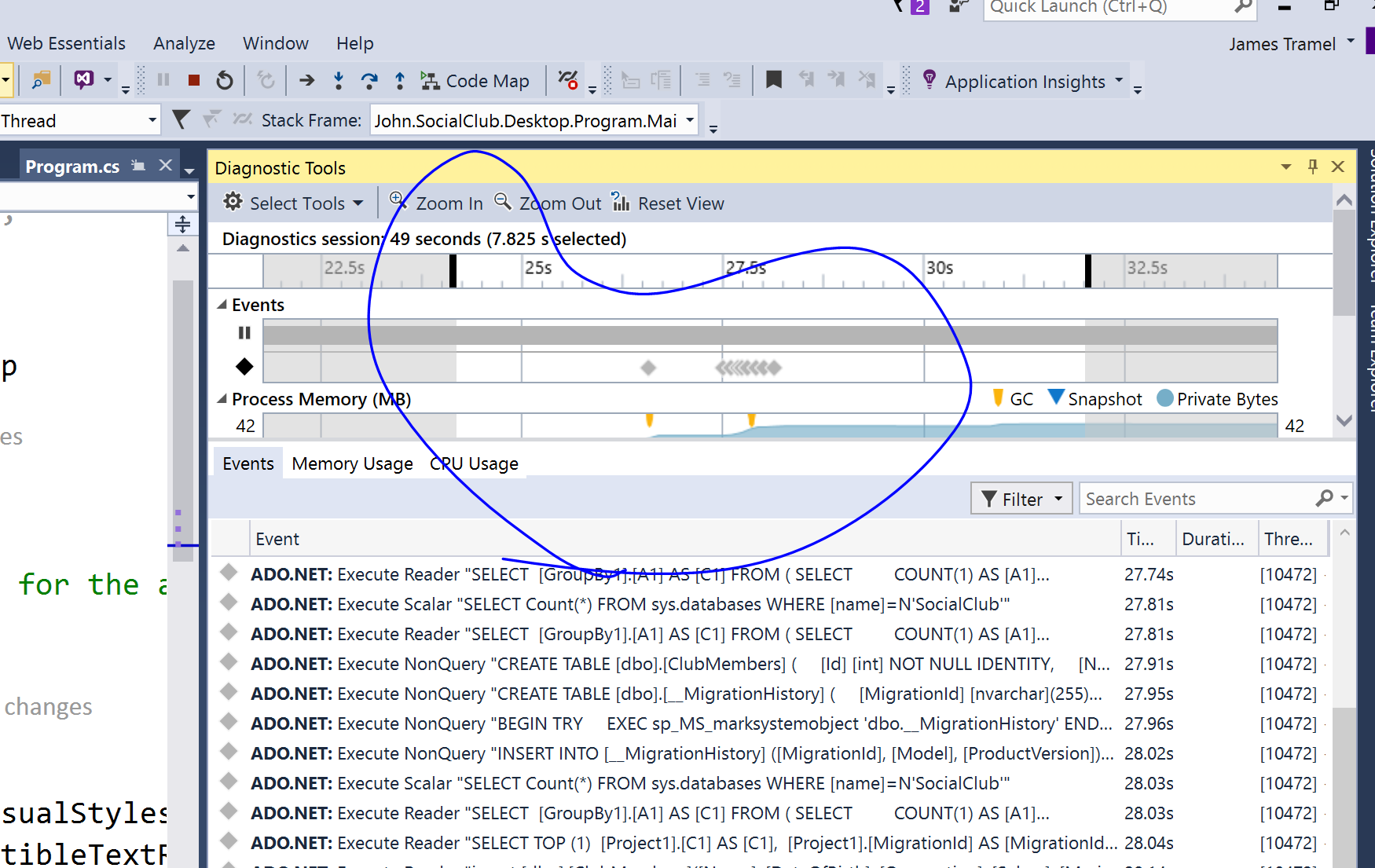
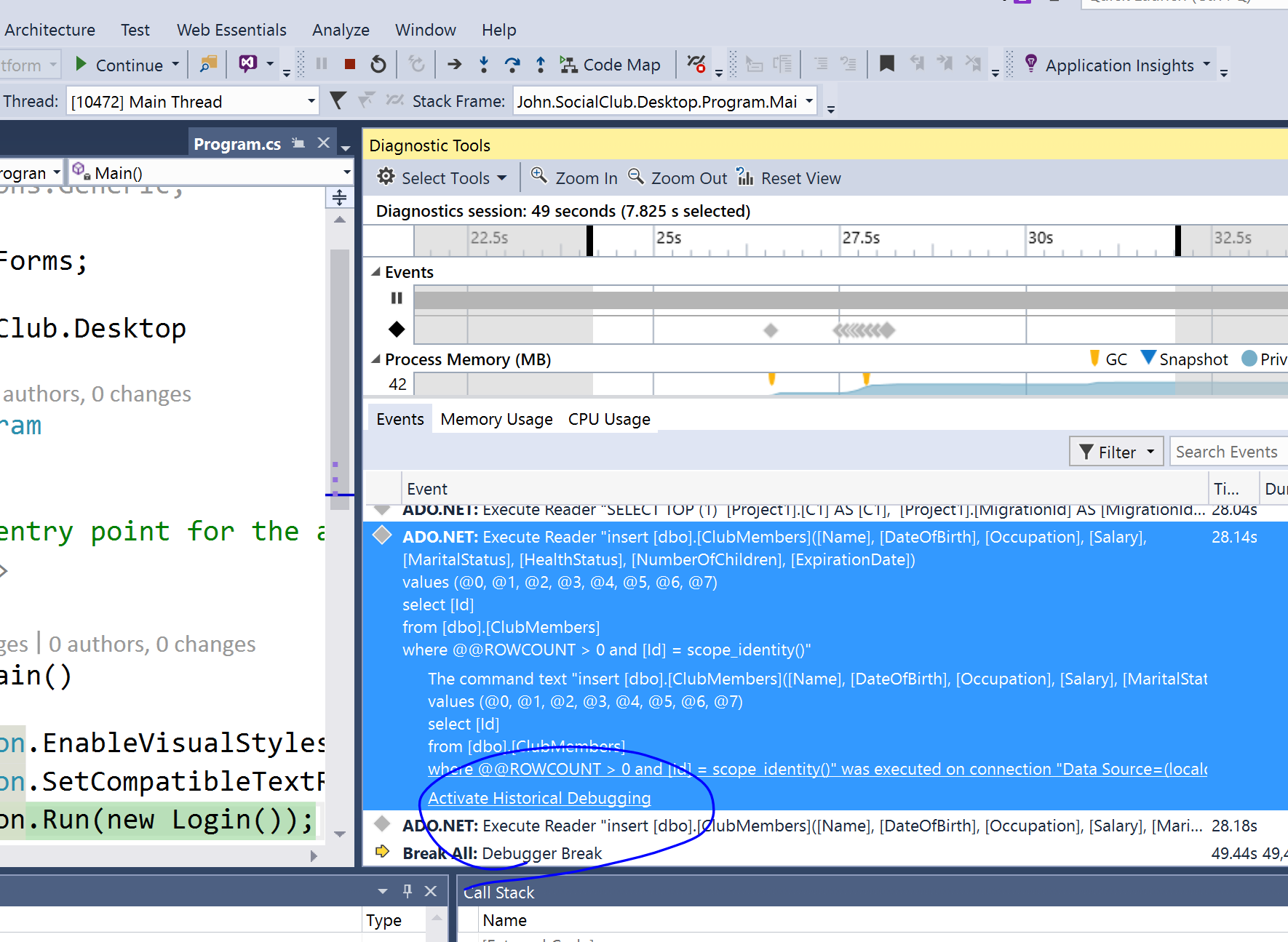
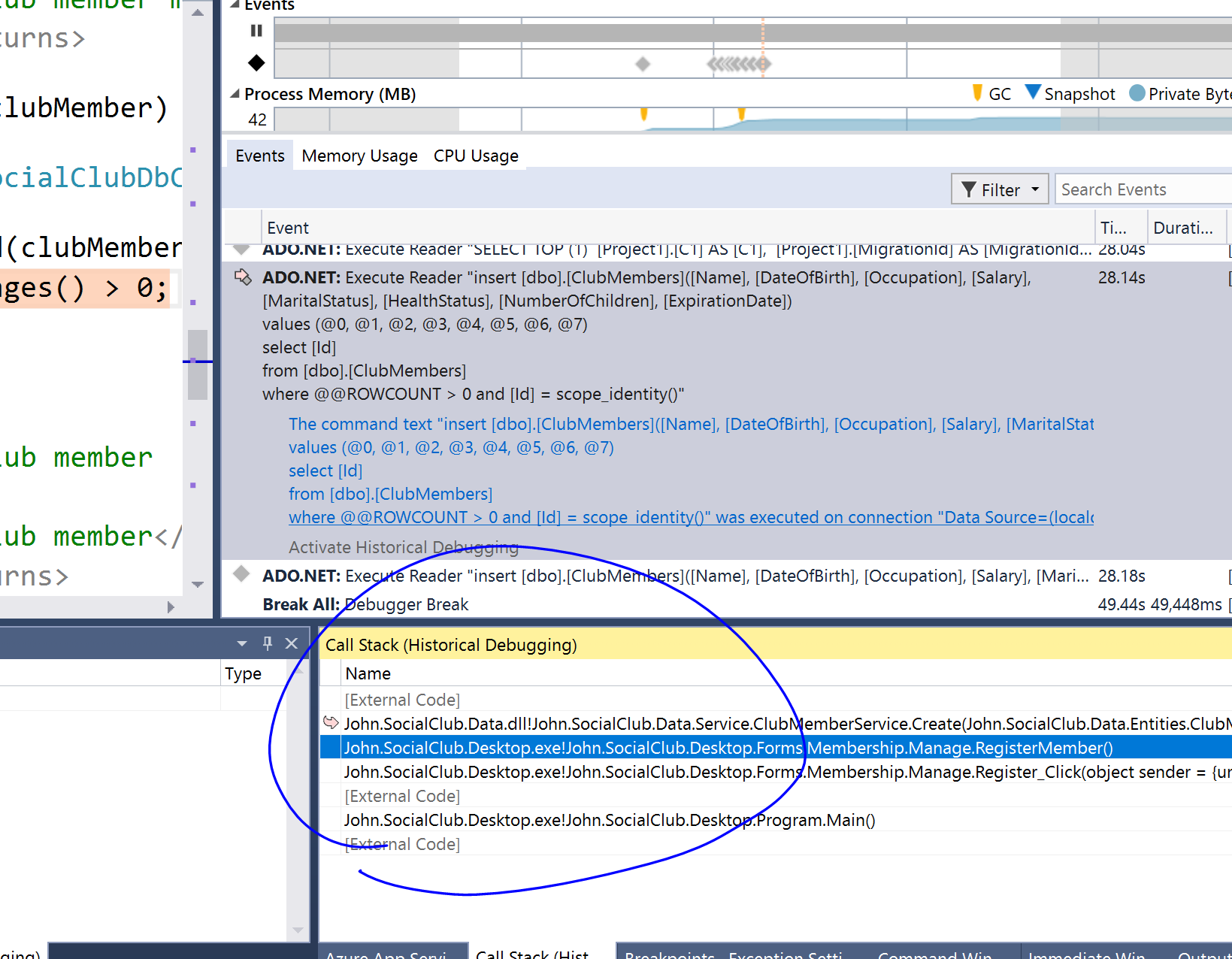
## Exercise 1: Introduction to Modern Debugging

* 1. The application we are going to be working with is a WinForms application from CodeProject called SocialClub. The application lets you maintain a database of members for a social club. The bug I want to fix is that search behaves erratically after a member is registered. To reproduce the bug, I start the application and register a new member

#### Task 1: Run the Application

* 1. Open an **Explorer** window and then open **John.SocialClub.sln** in Visual Studio 2015 Enterprise from the source code provided.
     1. 
  2. Click Start to debug the application  
     
  3. Click Login  
     
  4. Enter your first name, last name. Occupation of Engineer, marital status as married and health as excellent, then click register. 
  5. Click the Search / Manage Members tab
  6. Search for Engineeers and Married, and click Search
  7. Click Get all to get the unexpected result

#### Task 2: Using Intellitrace

* 1. Press the Pause, or Break all Button
  2. Explore the Diagnostic Tooling. 
  3. Zoom in on the likely events:  
     
  4. Scroll down in the event pane and find the first insert event. Click it once to select it, and then click Activate Historical Debugging
  5. Go through the call stack to find how members are registered /insterted  
       
     
  6. Repeat this process for the next event. To do this, select the second insert and select activate historical debugging again. Go through the call stack to find how the other item in registered / inserted.