Working with OpenSpan Automations

Pega® ACADEMY

Trademarks

For information about the third-party software that is delivered with the product, refer to the third-party license file on your installation media that is specific to your release.

Notices

This publication describes and/or represents products and services of Pegasystems Inc. It may contain trade secrets and proprietary information that are protected by various federal, state, and international laws, and distributed under licenses restricting their use, copying, modification, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

This publication is current as of the date of publication only. Changes to the publication may be made from time to time at the discretion of Pegasystems Inc. This publication remains the property of Pegasystems Inc. and must be returned to it upon request. This publication does not imply any commitment to offer or deliver the products or services described herein.

This publication may include references to Pegasystems Inc. product features that have not been licensed by you or your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems Inc. services consultant.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors, as well as technical inaccuracies. Pegasystems Inc. may make improvements and/or changes to the publication at any time.

Any references in this publication to non-Pegasystems websites are provided for convenience only and do not serve as an endorsement of these websites. The materials at these websites are not part of the material for Pegasystems products, and use of those websites is at your own risk.

Information concerning non-Pegasystems products was obtained from the suppliers of those products, their publications, or other publicly available sources. Address questions about non-Pegasystems products to the suppliers of those products.

This publication may contain examples used in daily business operations that include the names of people, companies, products, and other third-party publications. Such examples are fictitious and any similarity to the names or other data used by an actual business enterprise or individual is coincidental.

This information is the property of:

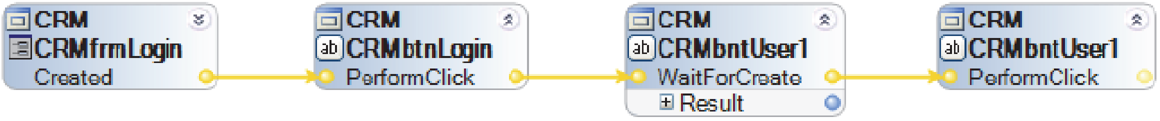
Pegasystems Inc.
One Rogers Street
Cambridge, MA 02142-1209
USA
Phone: (617) 374-9600
Fax: (617) 374-9620
www.pega.com

Chapter 7: WORKING WITH OPENSPAN AUTOMATIONS

The controls for the project are now defined, and we have set the multiple projects to start when we launch the TrainingCertification in OpenSpan Studio. The next step is to set up the logic (automations) which ties the applications together.

The logic (automations) is developed by establishing connections between the Windows Training CRM application and Main-UI Customer Information window properties, methods, and events. You establish connections using design blocks to create OpenSpan Studio automations. Design blocks are visual representations of the properties, events, and methods for the adapters, controls, and components used to create automations.

In this chapter, you build the CRM_E_CRMfrmLogin_Created.os automation. The first automation sets the logic to ensure that once the CRM Login form has been created, it automatically performs the login using the PerformClick method. Then, it uses the PerformClick method on the User 1 button on the Training CRM application.



Building Blocks

When you finish this chapter, you should know how to:

- Add an automation to a solution.
- Use recommended naming practices for an automation.
- Display the component names in an automation.
- Work with the properties, events, and methods for a control.
- Connect the automation logic using the Execution and Data paths.

This chapter contains the following topics:

Project 1: Creating a New Automation

Group Exercise 1: Creating the CRM_E_CRMfrmLogin_Created Automation

This automation will perform the Login and User 1 action when it has been matched on the CRM Login window, causing immediate login to the Training CRM application. Before we create the automation, let’s look at key recommended practice for naming our automations.

When you finish this project, you will know how to:

- Add an automation to a solution
- Use recommended naming practices for an automation
- Display the component names in an automation
- Work with properties, events, methods for a control

Recommended Naming Conventions for Automations

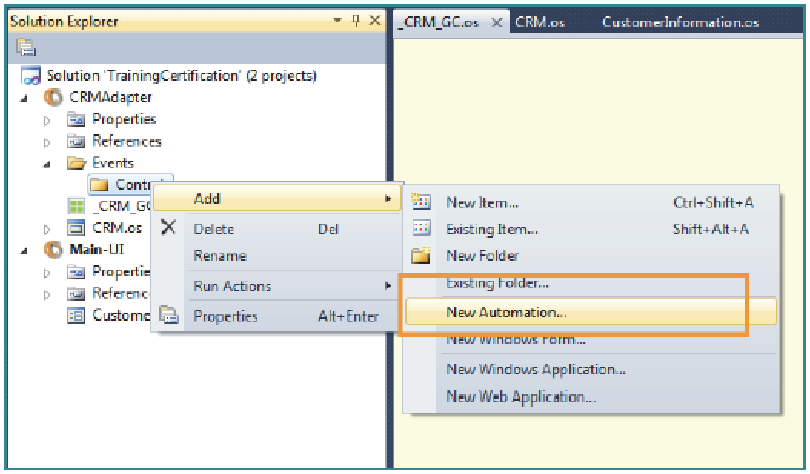
Automations should be named so you can easily identify the action they perform. We suggest that automations follow the same naming conventions as methods in their code counterparts.

The first automation in our exercise is named *CRM_E_CRMfrmLogin_Created.os,* where:

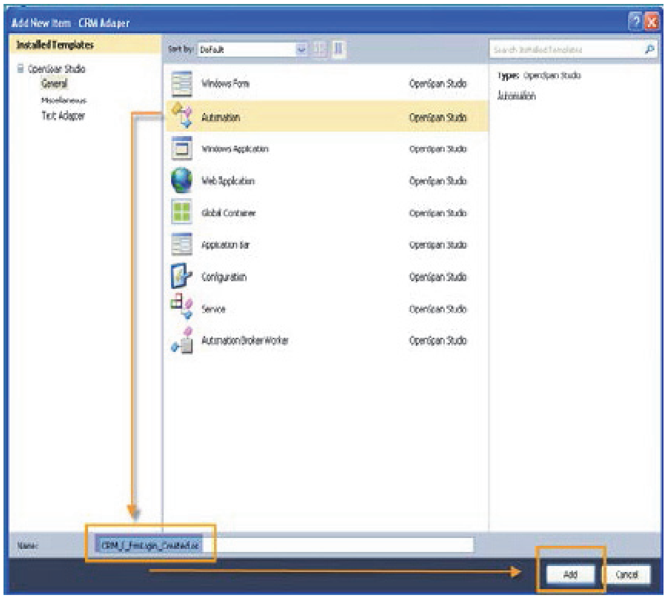
This part of the name	Is the:
CRM_E_CRMfrmLogin_Created	Adapter project within the solution.
CRM_***E_***CRMfrmLogin_Created	Event within the adapter.
CRM_E_***CRMfrmLogin_***Created	Control name within the adapter project.
CRM_E_CRMfrmLogin_***Created***	Event being responded to in the automation.

Follow these steps to create the **CRM_E_CRMfrmLogin_Created** automation:

1. In Solution Explorer and within the **CRMAdapter** project, expand the **Events** folder. Right-click the **Controls** folder and select **Add | New Automation**.

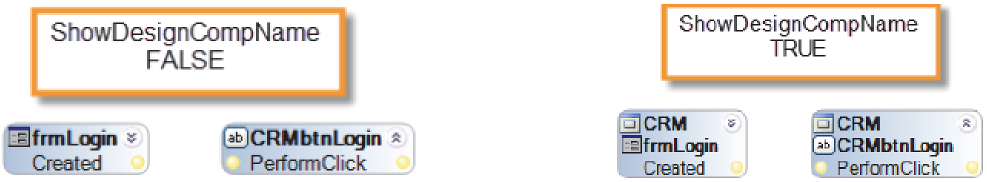


2. The Automation template should be highlighted. Type CRM_E_CRMfrmLogin_Created in the **Name** field and click **Add**.

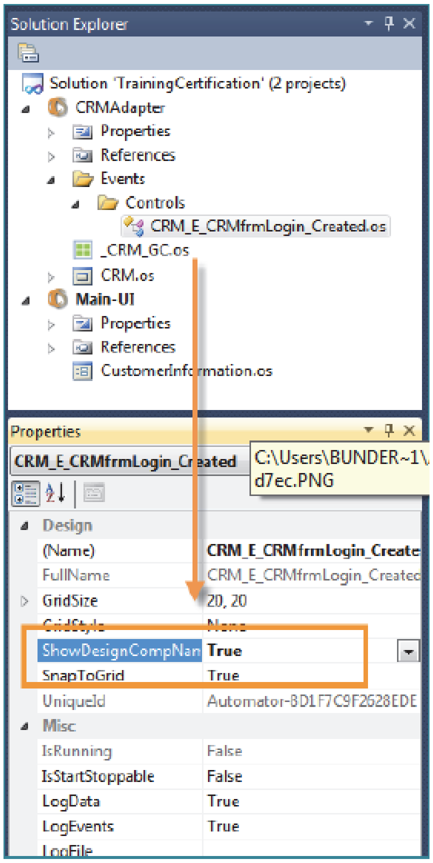

Add New Item - CRM Adapter
Installed Templates
OpenSpan Studio
General
Miscellaneous
Text Adapter
Sort by: Default
Search Installed Templates
Windows Form OpenSpan Studio
Automation OpenSpan Studio
Windows Application OpenSpan Studio
Web Application OpenSpan Studio
Global Container OpenSpan Studio
Application Bar OpenSpan Studio
Configuration OpenSpan Studio
Service OpenSpan Studio
AutomationBrokerWorker OpenSpan Studio
Type: OpenSpan Studio
Automation
Name:
CRM_E_FrmLogin_Created.os
Add
Cancel

The automation Design properties appear in the Properties windows (**View | Properties**). These properties define how controls and components of the automation are displayed.

For this exercise, the recommended practice when working with automation is to set the automation’s **ShowDesignCompName** property to True. OpenSpan then displays the full design name of the components in the automation design block.



3. Highlight and double-click the **CRM_E_CRMfrmLogin_Created** automation in Solution Explorer. In the Design section of the Properties windows, set the **ShowDesignCompNames** property to True so the adapter name (CRM) appears as the top row in the automation design blocks.



Solution Explorer

- Solution 'TrainingCertification' (2 projects)
  - CRMAdapter
    - Properties
    - References
    - Events
      - Controls
        - CRM_E_CRMfrmLogin_Created.os
    - _CRM_GC.os
    - CRM.os
  - Main-UI
    - Properties
    - References
    - CustomerInformation.os

Properties

CRM_E_CRMfrmLogin_Created

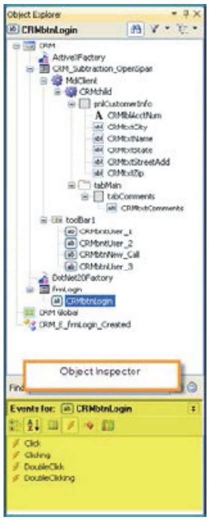
C:\Users\BUNDER~1\
d7ec.PNG

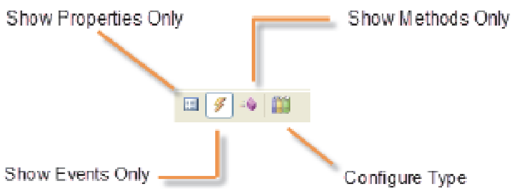
Design	
(Name)	CRM_E_CRMfrmLogin_Create
FullName	CRM_E_CRMfrmLogin_Createc
GridSize	20, 20
GridStyle	None
ShowDesignCompNan	True
SnapToGrid	True
UniqueId	Automator-8D1F7C9F2628EDE
Misc	
IsRunning	False
IsStartStoppable	False
LogData	True
LogEvents	True
LogFile	

Group Exercise 2: Adding Automation Logic

Now, begin building the automation. First, highlight objects on the Object Explorer window. Then, select properties, events, or methods, you need for the object.

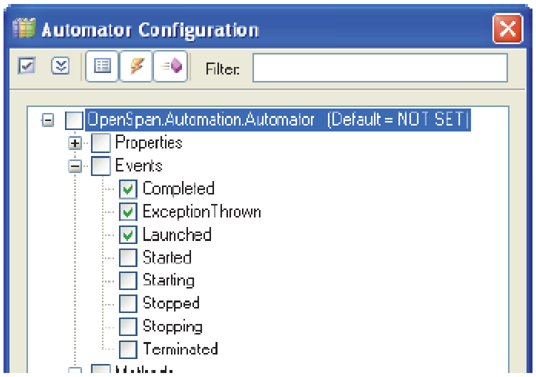
After you have selected an object, click the appropriate icon in the Object Inspector window at the bottom of the Object Explorer window to select from the available default properties, events, and methods for your selection.





To create a design block, simply choose an event and drag your selection onto the Designer window area.

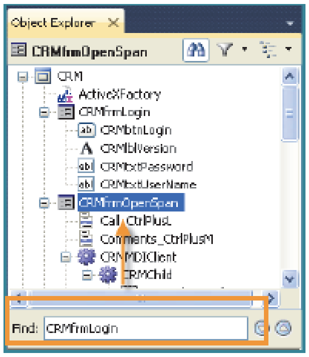
If any property, event, or method you need is not displayed in the Objector Inspector, click the **Configure Type** icon to open a dialog showing all available properties, events, and methods for the control on the Configuration dialog.



- □ OpenSpan.Automation.Automator (Default = NOT SET)
  - □ Properties
  - □ Events
    - ☑ Completed
    - ☑ ExceptionThrown
    - ☑ Launched
    - □ Started
    - □ Starting
    - □ Stopped
    - □ Stopping
    - □ Terminated
  - □ Methods

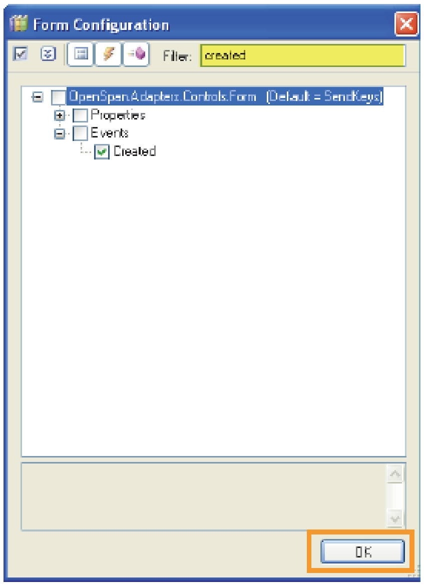
1. In Object Explorer, select the **CRMfrmLogin** control.

You can also search for the control by typing in the Find field in the Object Hierarchy.



Find: CRMfrmLogin

2. In Object Inspector, click the **Show Events Only** icon. The available events for the **CRMfrmLogin** control appear. You need to use the **Created** event for this automation, but it is not listed in the in Object Inspector.

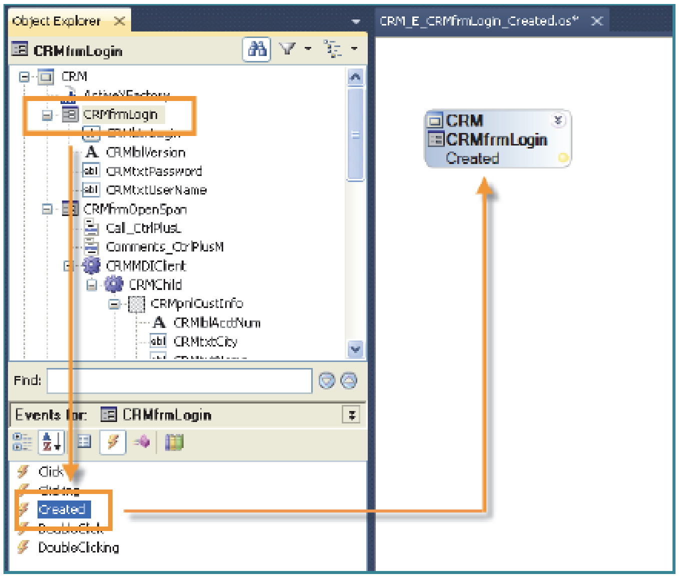


Filter: created

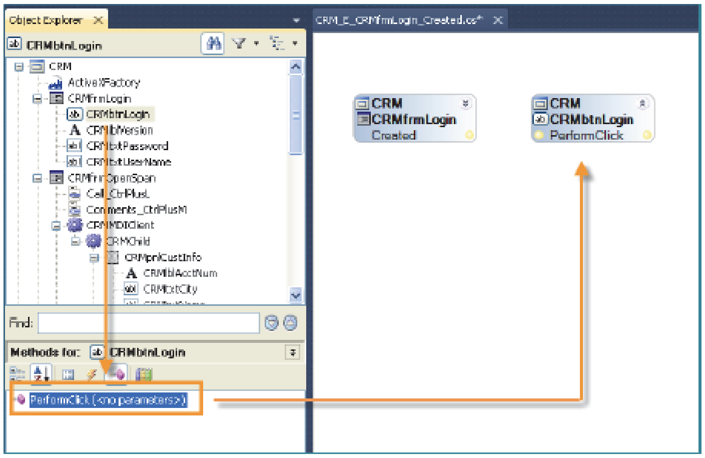
☑ Created

3. Click the **Configure Type** icon to display the **Form Configuration** dialog. To quickly locate the event, you can use the **Filter** field and enter Created. Expand **Events**, check **Created**, and click **OK** to add the event to the Object Inspector.

4. In the Events section, select the **Created** event and drag and drop to the **CRM_E_CRMfrmLogin_Created** automation.



5. Highlight the **CRMbtnLogin** control and select the **Method** icon in the Object Inspector window. Select the **PerformClick** method, then drag and drop it to the automation.

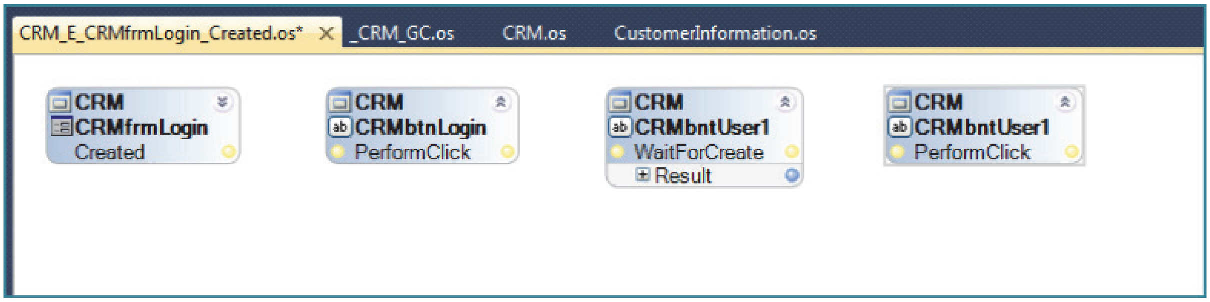


6. Next, add the **WaitForCreate** design block for the CRMbtnUser1 control to the automation. Highlight the **CRMbtnUser1** control and select the **Method** icon in the Object Inspector window. Select the **PerformClick** method, then drag and drop it to the automation.

If the WaitForCreate method is not listed in the object inspector, click the Configure Type icon to locate.

Using the WaitForCreate method allows enough time for the CRMbtnUser1 button to paint onto the form window before the PerformClick method is executed. There is an implicit wait time of 30 seconds for the control to paint before an exception.

7. To finish this part of the automation, add the final design block that performs the click action on the User 1 button. Highlight the **CRMbtnUser_1** from the **CRMToolBar** control in the Object Hierarchy. Select the **Method** icon, then drag and drop the **PerformClick** method to the automation.



8. Select **File | Save All**.

Project 2: Connecting the Execution and Data Paths

When you finish this project, you should know how to:

- Connect the automation logic using the Execution and Data paths

Group Exercise 1: Connecting the Execution and Data Paths

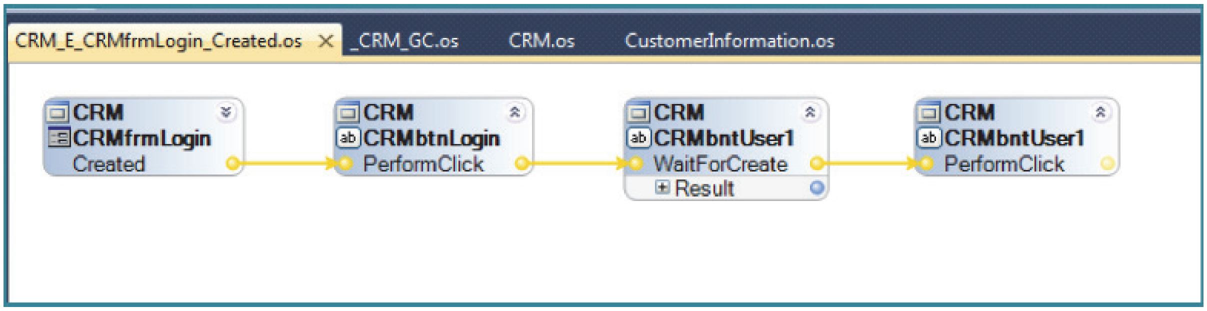
At this point, all properties, events, and methods for the controls in the solution are exposed for use. In the next steps, you connect the execution paths to complete the logic flow for the automation.

Note Yellow lines represent the execution path. Blue lines represent the data path. To draw lines between design blocks, click a colored port on one object and drag to the target object port and release.

1. Connect the design blocks as shown here:

Source Component	Connect To	Target Component
CRM.CRMfrmLogin.Created Event output	⇨	**CRM.CRMbtnLogin.PerformClick** Method input
CRM.CRMbtnLogin.PerformClick Method input	⇨	**CRM.CRMbtnUser1.WaitForCreate** Method input
CRM.CRMbtnUser1.WaitForCreate Method output	⇨	**CRM.CRMbtnUser_1.PerformClick** method input

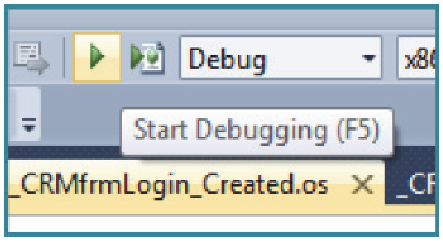
The **CRM_E_CRMfrmLogin_Created.os** automation should look as follows:



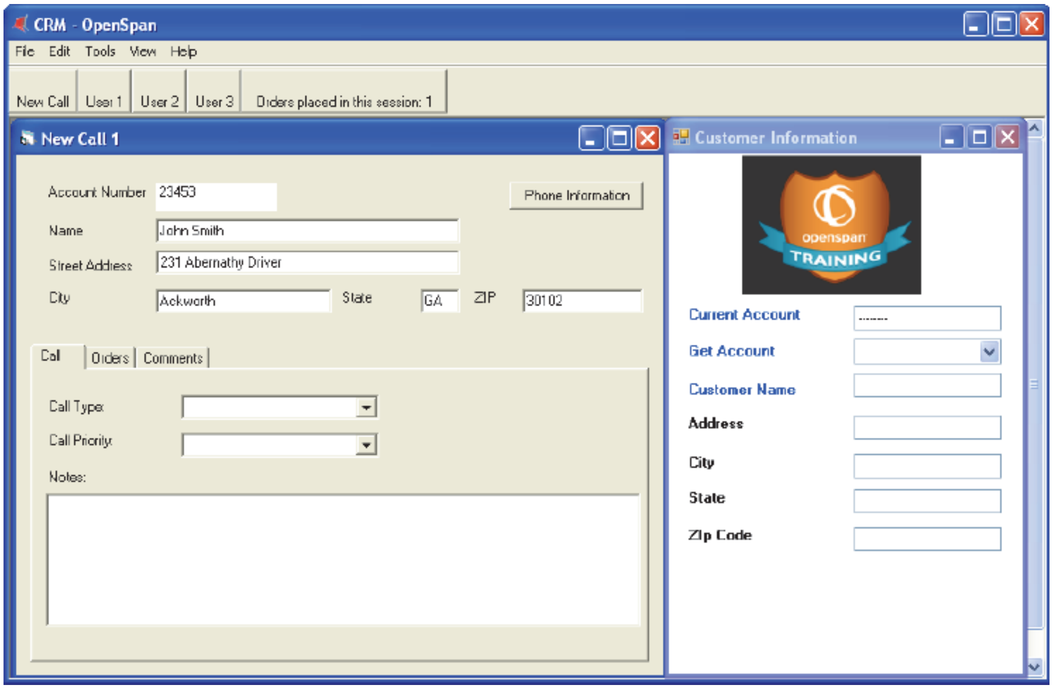
Testing the CRM_E_CRMfrmLogin_Created Automation

Test your automation logic with the following scenarios:

1. Click **Start Debugging** (F5) to start the project.

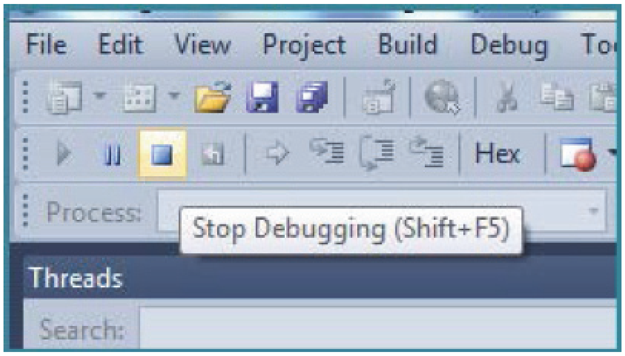


The automation should automatically log in the user and set the **New Call 1** on the **CRM – OpenSpan** application window for **John Smith,** and launch the **CustomerInformation** dialog**.**



Account Number 23453
Name John Smith
Street Address 231 Abernathy Driver
City Ackworth State GA ZIP 30102
Current Account -------

2. Click **Stop Debugging** (Shift-F5) to stop the running of the solution.



Note In the following group exercises, new concepts are introduced when working with the CRM_E_CRMChild_Created automation:

- MDI windows
- Interactions
- Interaction Manager

In this chapter, you learned how to:

- Add the CRM_E_CRMChild_Created automation to automate the process for the user to be logged into the CRM application and land on the New Call window.
- Use OpenSpan Recommended Practices when naming an automation.
- Set the ShowDesignName properties to display the name of the application in the Designer window of the automation.
- Use recommended naming practices for an automation.
- Display the component names in an automation.
- Work with properties, events, methods for a control using the Inspector Object and Configure Type feature.