Working with MDI Child Windows

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 8: WORKING WITH MDI CHILD WINDOWS

When building solutions for Multiple Document Interface (MDI) Windows applications, you have to manage how data and events are directed for the child windows because you may have multiple instances of these windows opened at run time. In this chapter, you learn how to do this.

Building Blocks

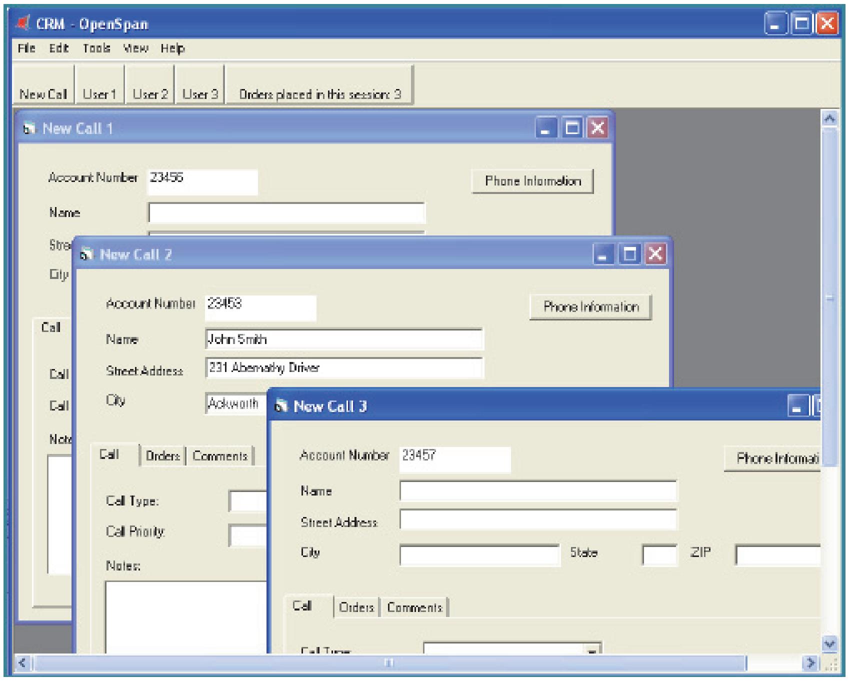
When you complete this chapter, you should be able to:

- Work with MDI windows.
- Set the MDI Child window UseKeys property.
- Use MDI Child windows in a project.

This chapter includes these topics:

Project 1: Working with MDI Child Windows

When building solutions for MDI Windows applications, you must manage how data and events are directed for child windows since multiple instances of these windows may be opened at run time.



New Call 1: Account Number 23456

New Call 2: Account Number 23453; Name John Smith; Street Address 231 Abernathy Driver; City Ackworth

New Call 3: Account Number 23457

This example shows how applications that contain MDI child windows require special interrogation and design consideration.

Suppose the target application is the Training CRM application in the OpenSpan Developer Certification solution. A scenario could occur where a user has a New Call window open and is entering notes. A new call comes in and a second New Call window opens.

If your solution interacts with a New Call window, how does it know which New Call window to use?

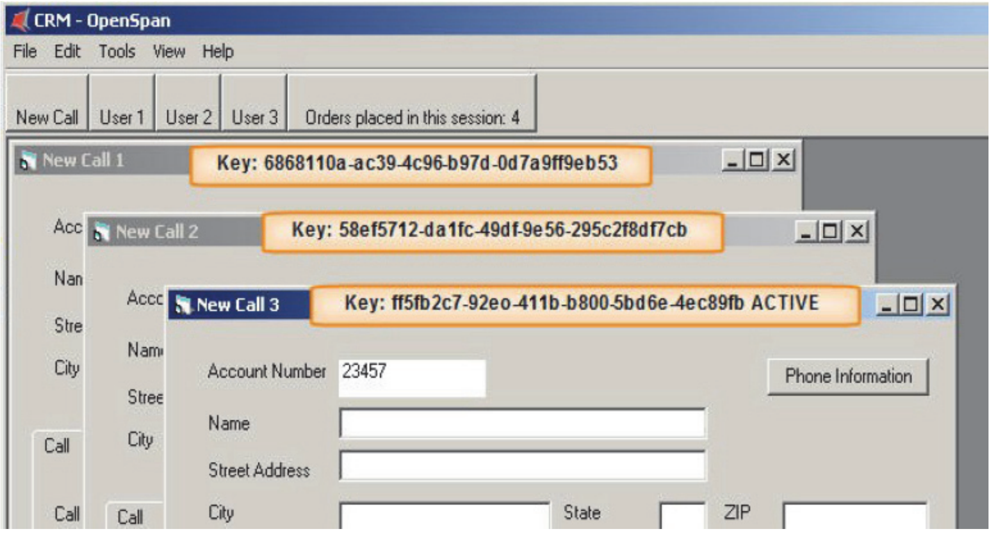
In this group exercise, you use the Training CRM application to develop a solution that works with MDI child windows. Before you begin the exercise, review the following information on how to work with multiple instances of controls, such as MDI windows.

MDI Applications and OpenSpan Studio

OpenSpan Studio creates a Key property for all interrogated controls. The Key property is used to uniquely identify the object at run time. It has the following values:

Relative	Active/Null
Absolute	The GUID (HGlobally Unique Identifier) of the form

For example, when running an OpenSpan project which contains the CRM application, OpenSpan Studio automatically assigns the key as each MDI child windows opens. The window and all of the objects on the window are uniquely identified by the key. OpenSpan also designates the key as Active for the active window, as shown in the following example:



Key: 6868110a-ac39-4c96-b97d-0d7a9ff9eb53

Key: 58ef5712-da1fc-49df-9e56-295c2f8df7cb

Key: ff5fb2c7-92eo-411b-b800-5bd6e-4ec89fb ACTIVE

Account Number 23457

Note You can use the assigned key value or set the key yourself.

Application of Keys

OpenSpan creates and uses keys whenever a context exists for cloneable objects. Here, the term **context** defines any instance of an object for which multiple instances (clones) can be created. For example, the MDI child window is a context.

The creation of keys and the setting of keys for windows objects is governed by the existence of a context, as seen in the following list:

- Events set the context.
- Event from No Context to a Context requires a key assignment.
- Event from Context to Child Context requires a key assignment.
- Event from Context to Parent Context does not require a key assignment.
- Operations within the same context do not require a key assignment.

Project 2: Developing a Solution that Uses MDI Child Key Values

When you finish this project, you should be able to:

- Set the MDI Child window UseKeys property.
- Use the MDI Child windows in a project.

This project uses the OpenSpan CRM application and the Customer Information form to work with MDI child windows and keys. Each time the User 1, 2, or 3 buttons are clicked in the CRM application, a New Call window opens. These windows are MDI child type windows. The solution assigns a key to each window based on the account number displayed on the window.

The solution also stores the keys of the open windows in the combo box on the Customer Information window. As the windows are closed, the keys are removed from the combo box.

Data is also pushed to the New Call window from the Customer Information window to illustrate how to set the key value at the beginning of execution logic.

The topics and concepts represented by this sample solution include:

- Interrogating MDI child windows.
- Setting and Using the MDI child window key value.

  Concepts: Using the Key property, WaitForCreate method, and combo box methods to create a list of key values.
- Activating a Window based on key value.

  Concepts: Using Activate method to activate an MDI child window.
- Identifying the controls for the execution path.

  Concepts: Setting the upstream key value for identifying controls on MDI child windows.

Group Exercise 1: Setting the UseKeys Value on the MDI Child Window

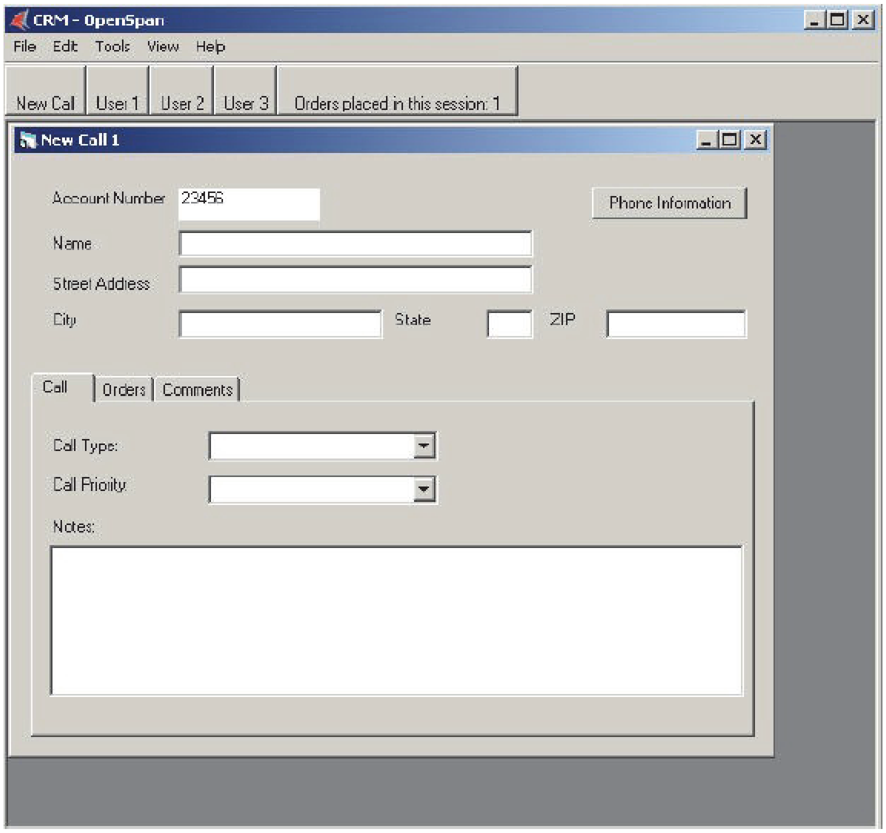
In this part of the exercise, you create **CRM_E_CRMChild_Created** automations to work with the Key property of the New Call window (MDI child window). The solution uses the Customer Information form that you created earlier.

1. Open the **TrainingCertification** solution.

2. Open the **CRM.os** project item.

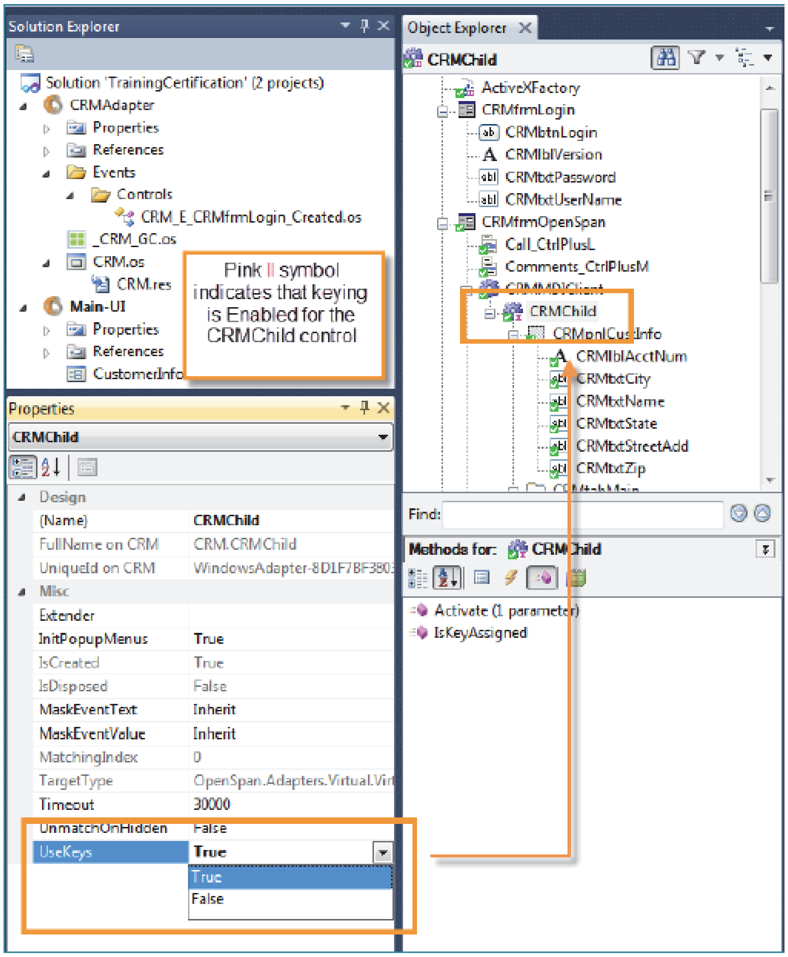
3. Click **Start Interrogation** on the CRM.os Designer window. The CRM application launches and the Login window appears.

4. Click **Login** on the **CRM application**, and click the **New Call** button. A New Call MDI window appears and all the controls for the New Call 1 window are matched, as indicated by the green check mark (√).



Account Number 23456

5. Highlight the **CRMChild** window control in Object Explorer. In the **Properties** window, set the **UseKeys** property for the control to **True**. Notice that the CRMChild item in Object Explorer has a pink pipe (||) symbol, indicating that keying is enabled for that item.



Pink symbol indicates that keying is Enabled for the CRMChild control

Design	
(Name)	CRMChild
FullName on CRM	CRM.CRMChild
UniqueId on CRM	WindowsAdapter-8D1F7BF380
Misc	
Extender	
InitPopupMenus	True
IsCreated	True
IsDisposed	False
MaskEventText	Inherit
MaskEventValue	Inherit
MatchingIndex	0
TargetType	OpenSpan.Adapters.Virtual.Vir
Timeout	30000
UnmatchOnHidden	False
UseKeys	True
	True False

6. Stop the **Interrogator**.

7. Select **File | Save All.**

Note Before you can develop solutions using MDI child windows, you need to understand and work with the Interaction.xml file, interaction component, and activities. The next chapters discuss these components in further detail.

In this chapter, you learned how to:

- Manage how data and events are directed for child windows since multiple instances of these windows may be opened at run time.
- Set the MDI Child Window Key Value to True on the CRMChild control item to use multiple document interface windows in the project.