

## **Tutorial**

### **Building Composite Applications for IBM Lotus Notes 8**

#### **For use with the IBM Lotus Notes 8 Beta 2 client**

Building composite applications is a process that involves multiple procedures. This tutorial has organized the procedures into multiple lessons, which you should perform in the order in which they are presented.

This tutorial contains the steps you'll need to create a NSF-based composite application that includes NSF components and Eclipse components. You can also reference these procedures when creating your own composite applications.

- Overview and set-up
- Lesson 1 - Create WSDL file to support inter-component communication
- Lesson 2 - Define Notes view columns that publish properties
- Lesson 3 - Define Notes actions for inter-component communication
- Lesson 4 - Assemble an NSF-based composite application
- Lesson 5 - Add NSF components to the composite application
- Lesson 6 - Wire components in the composite application
- Lesson 7 – Publishing properties using LotusScript APIs
- Lesson 8 - Add Eclipse component to the composite application
- Lesson 9 - Wire Eclipse and NSF components in a composite application
- Lesson 10 - Additional actions and wiring

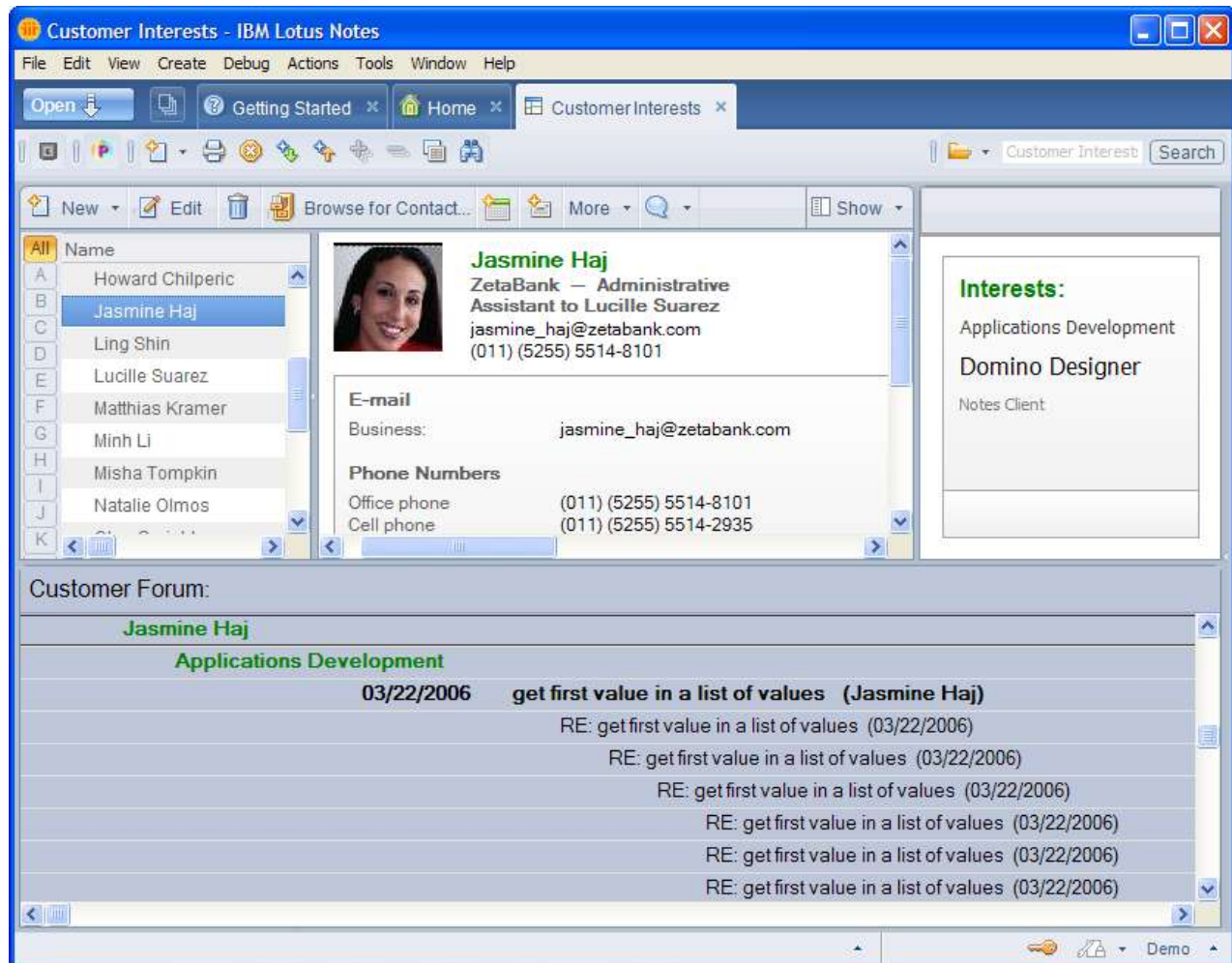
## Overview and set-up

This tutorial includes procedures to guide you in creating an NSF-based composite application. The final composite application will contain two NSF components from two Notes applications and one Eclipse component. These components will have loosely coupled causal relationships across application and system boundaries and will change their display based on user action in a single component.

In this tutorial, you will behave in the role of the developer, by using Domino Designer 8 Beta, to modify the Notes application design elements to support inter-component communication. You will also use the Composite Application Editor (CAE) feature of the Notes 8 Beta client to assemble the components and wire the components into causal relationships. The drag and drop capabilities of the CAE feature allows it to be used by developers or by line-of-business users that have proper access to a composite application.

Below is a screen capture of the end result of the enclosed procedures. The composite application includes inter-component communication. The user of the composite application can select a name in the Notes Contacts component displayed at the top left top left. The user name will be published by that component and the two other components will execute their predefined logic to display information relevant to the selected user.

The end user can also select a topic in the Tag Cloud component displayed in the upper right and the NDForum component, displayed at the bottom, will execute its logic to list documents related to the previously selected user.



## Applications for this tutorial:

### Profiles.NSF

This Notes application contains contact information for fictitious users. The documents in this Notes application are intended to be copied to the Notes 8 Contacts application, formerly known as the Personal Name and Address Book.

### NDForum.NSF

This Notes application is a forum of questions and answers about Lotus Notes and Domino related topics. Authors include the same fictitious users listed in Profiles.NSF. This content and design is a subset of the Notes and Domino forum accessible from IBM developerWorks.

You will use one Notes view as an NSF component and modify it to support inter-component communication.

### Tag Cloud Eclipse component

A tag cloud (or weighted list in visual design) can be used as a visual depiction of content tags used on a website. Often, more frequently used tags are depicted in a larger font or otherwise emphasized, while the displayed order is generally alphabetical. Thus both finding a tag by alphabet and by popularity is possible. Selecting a single tag within a tag cloud will generally lead to a collection of items that are associated with that tag. -- [http://en.wikipedia.org/wiki/Tag\\_cloud](http://en.wikipedia.org/wiki/Tag_cloud)

The Tag Cloud component was built by another developer using the Eclipse IDE. As the Notes 8 Beta client supports components of different technologies you will include this component in your composite application. The Tag Cloud component is available from a local Eclipse update site you must first extract to your local drive.

### Set-up procedure

1. Extract the contents of the ZIP file to a new folder of your naming.
2. Copy the two Notes applications to the data folder of your Notes 8 Beta client
  - NDForum.NSF
  - Profiles.NSF

Look at the content of these two Notes applications through the Lotus Notes 8 Beta client. You can open the applications by using the Lotus Notes 8 Beta client menu **File > Open > Lotus Notes Application**.
3. Open Profiles.NSF and use **Edit > Select All** to copy all documents.
4. Open your Notes Contacts (Names.nsf) and paste the documents into the view named **My Contacts**.
5. Extract the contents of the updatesite.zip file to a new folder on your root drive and preserve folder names. The recommended name for this new folder is c:\updatesite. This folder will contain the Eclipse update site you will use in a lesson of this tutorial.
6. The following two text files contain LotusScript code that you can paste into Notes actions through your IBM Lotus Domino Designer 8 Beta client as noted in Lessons 3 and 10.
  - SelectPerson\_Action\_for\_NDForum.txt
  - SelectSubCategory\_Action\_for\_NDForum.txt

**Note:** To complete all lessons in this tutorial, you must install and setup the following

- IBM Lotus Notes 8 Beta 2 client with the optional Composite Application Editor feature
- IBM Lotus Domino Designer Beta 2 client

## Lesson 1 - Create WSDL file to support inter-component communication

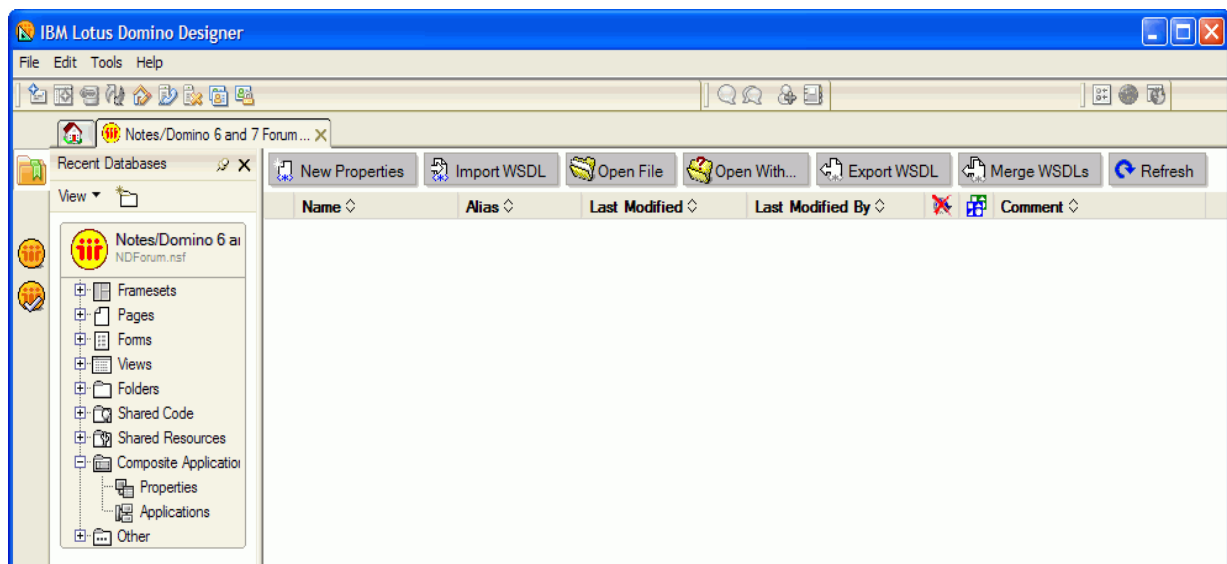
This lesson includes one procedure to create a WSDL file for use by the NSF components from the NDForum application. The Notes Contacts application and the supplied Tag Cloud Eclipse applications each have a WSDL file that was previously defined. A composite application component can participate in inter-component communication if it has properties and actions that can be wired to other components. The WSDL file lists the possible properties and actions that a component may expose, plus the namespace and other attributes. When a property and an action have the same namespace and datatype, they can be wired to form a causal relationship.

### Procedure 1.1 - Create a new WSDL to support properties and actions in NDForum.NSF

Complete the following steps to create a WSDL to support properties and actions:

1. If needed, start the Lotus Notes 8 Beta client.
2. Start the Lotus Domino Designer 8 Beta client.
3. In Domino Designer, click **File > Application > Open** and select NDForum.NSF from the list of Notes applications.
4. Expand **Composite Applications** in the design pane on the left and then click **Properties**.

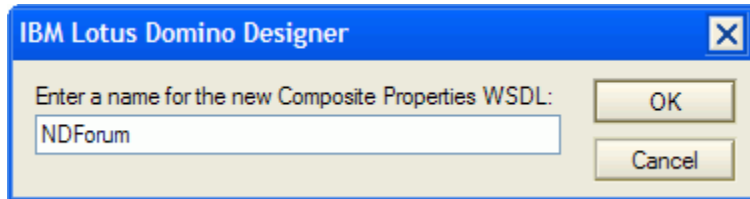
**Result:** The work pane appears as shown in the following screen capture



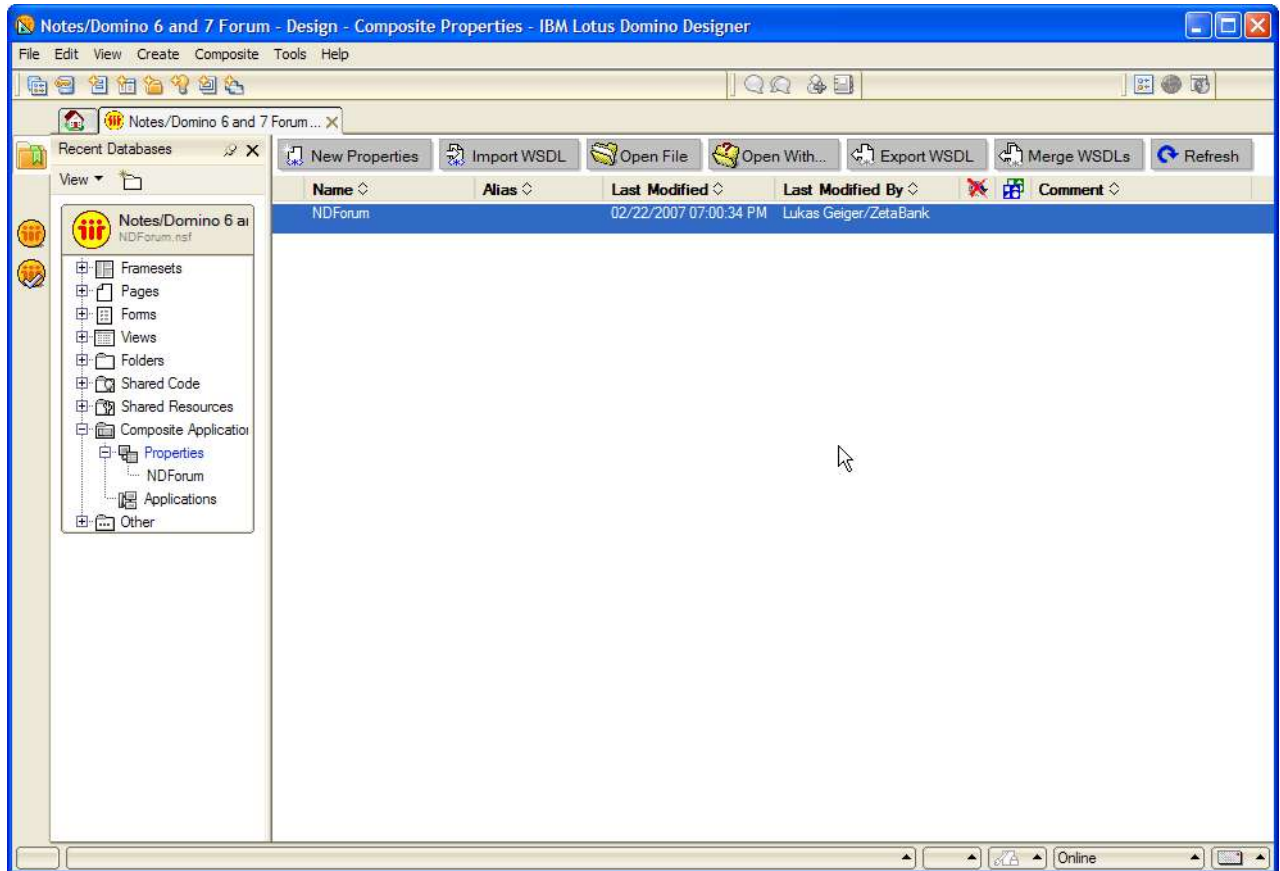
- Click the **New Properties** button at the top of the work pane.

**Result:** A dialog appears

- Type **NDForum** and click **OK**.

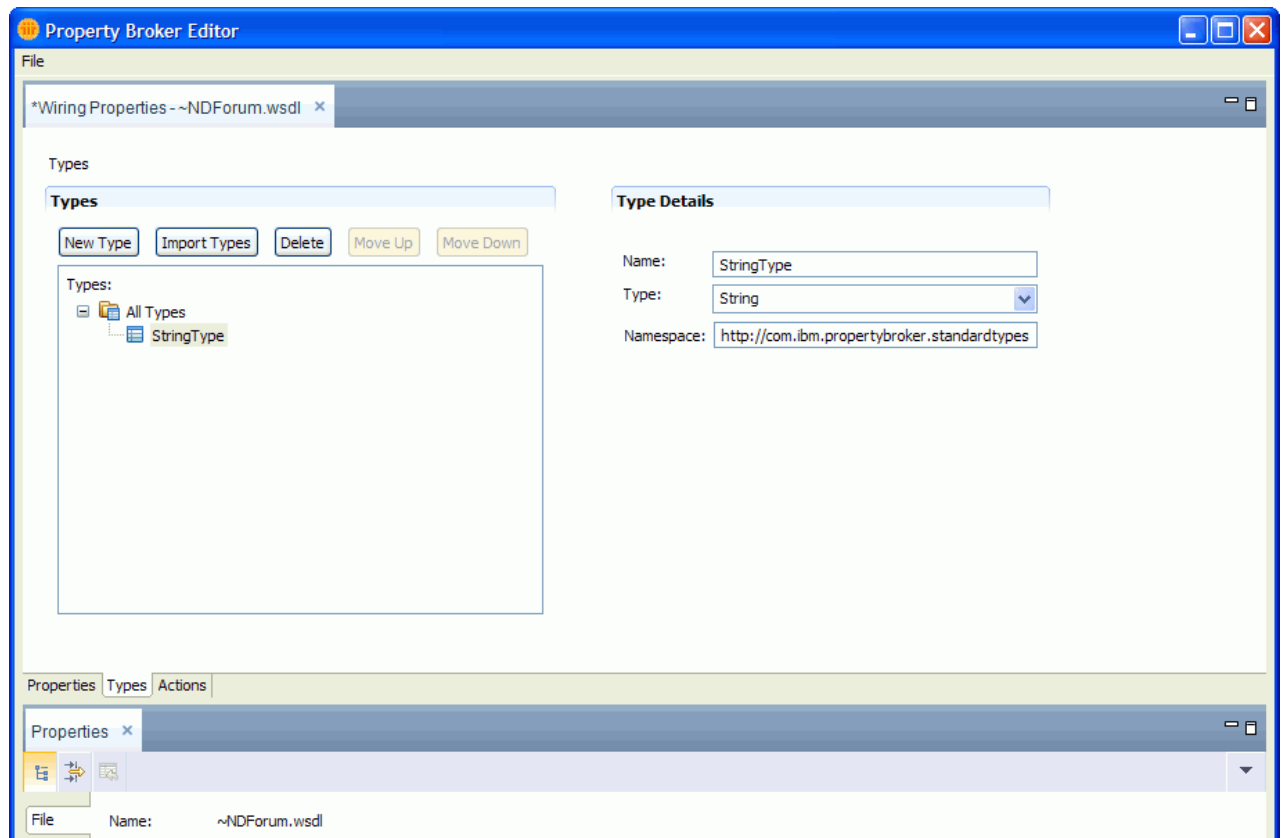


**Result:** The new WSDL file, NDForum.wsdl, is listed in the work pane as shown below.



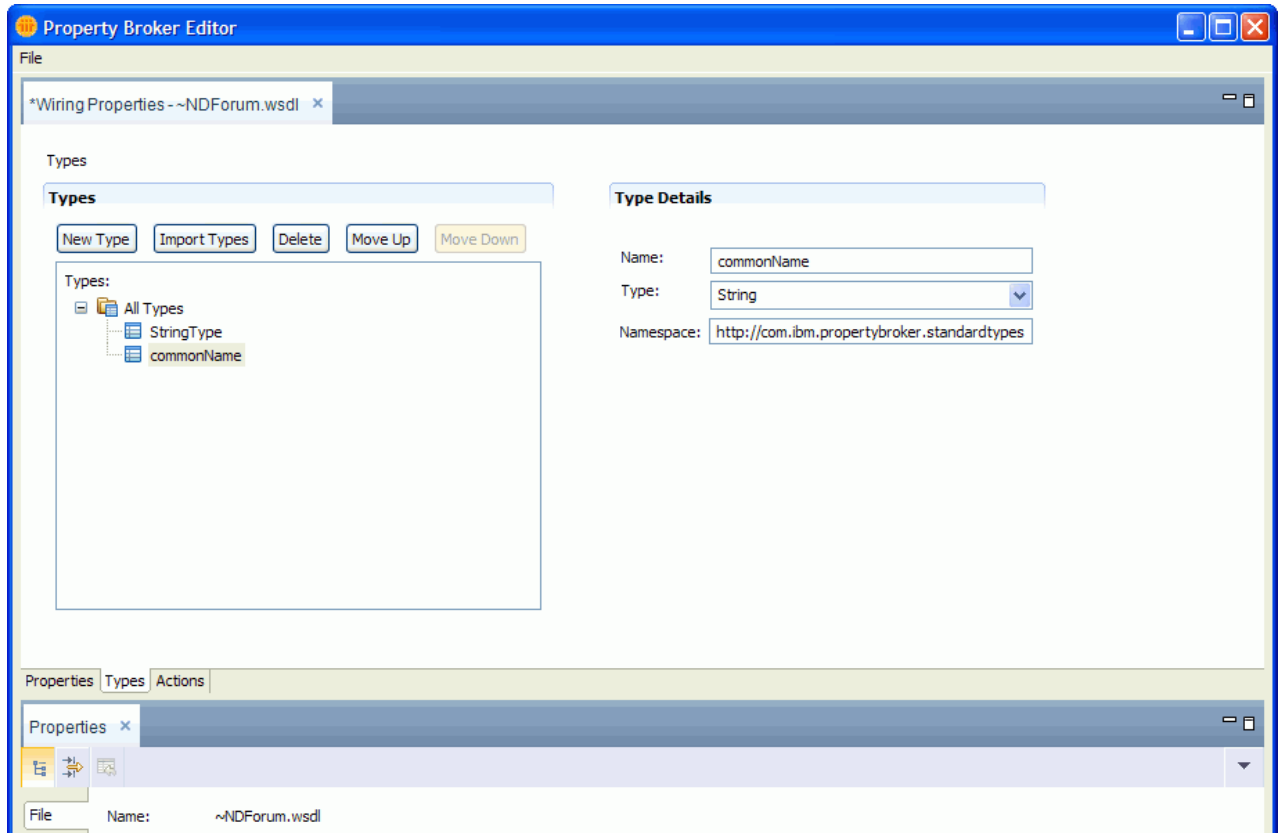
7. Click the **Open File** button at the top of the work pane.  
**Result:** The Property Broker Editor feature of Domino Designer 8 Beta appears.
8. Select the **Types** tab located in the middle of the Property Broker Editor feature.
9. Select the **New Type** button located on the upper left.
10. In the **Type Details** section in the upper right:  
Type "StringType" in the **Name** field  
Verify that "String" is selected in the **Type** field  
Type "http://com.ibm.propertybroker.standardtypes" in the **Namespace** field  
**Note:** This namespace is used in the Contacts and Tag Cloud components you will use later in this tutorial.

Your Types tab should be identical to the following screen capture.



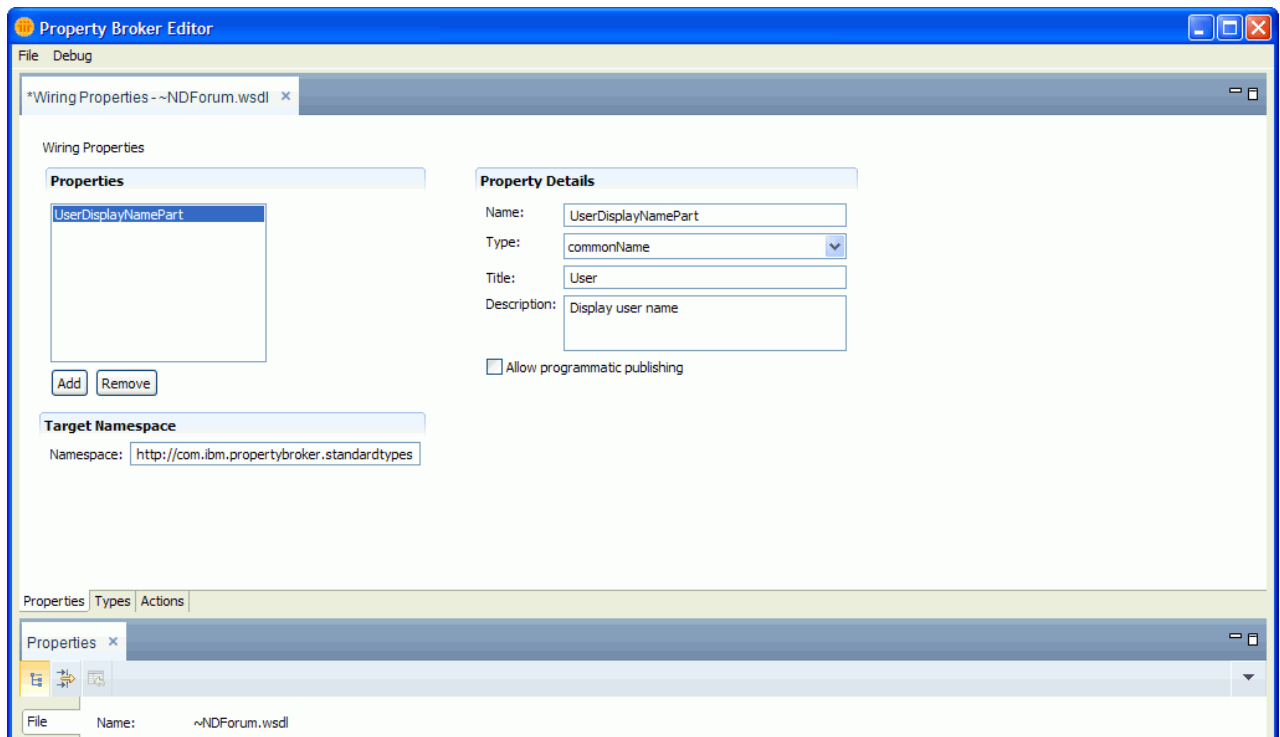
11. If needed, select the **Types** tab located in the middle of the Property Broker Editor feature.
12. Select the **New Type** button located on the upper left, to create a second datatype.
13. In the **Type Details** section in the upper right:  
Type "commonName" in the **Name** field  
Verify that "String" is selected in the **Type** field  
Type "http://com.ibm.propertybroker.standardtypes" in the **Namespace** field

Your Types tab should be identical to the following screen capture.



14. Select the **Properties** tab located in the middle of the Property Broker Editor feature.
15. Select the **Add** button that is located on the left.
16. In the **Property Details** section displayed in the upper right:  
Type "UserDisplayNamePart" in the **Name** field  
Select "commonName" for the **Type** field  
Type "User" in the **Title** field  
Type "Display user name" in the **Description** field  
Deselect the option **Allow programmatic publishing**
17. In the **Target Namespace** section  
Type "http://com.ibm.propertybroker.standardtypes" in the **Namespace** field

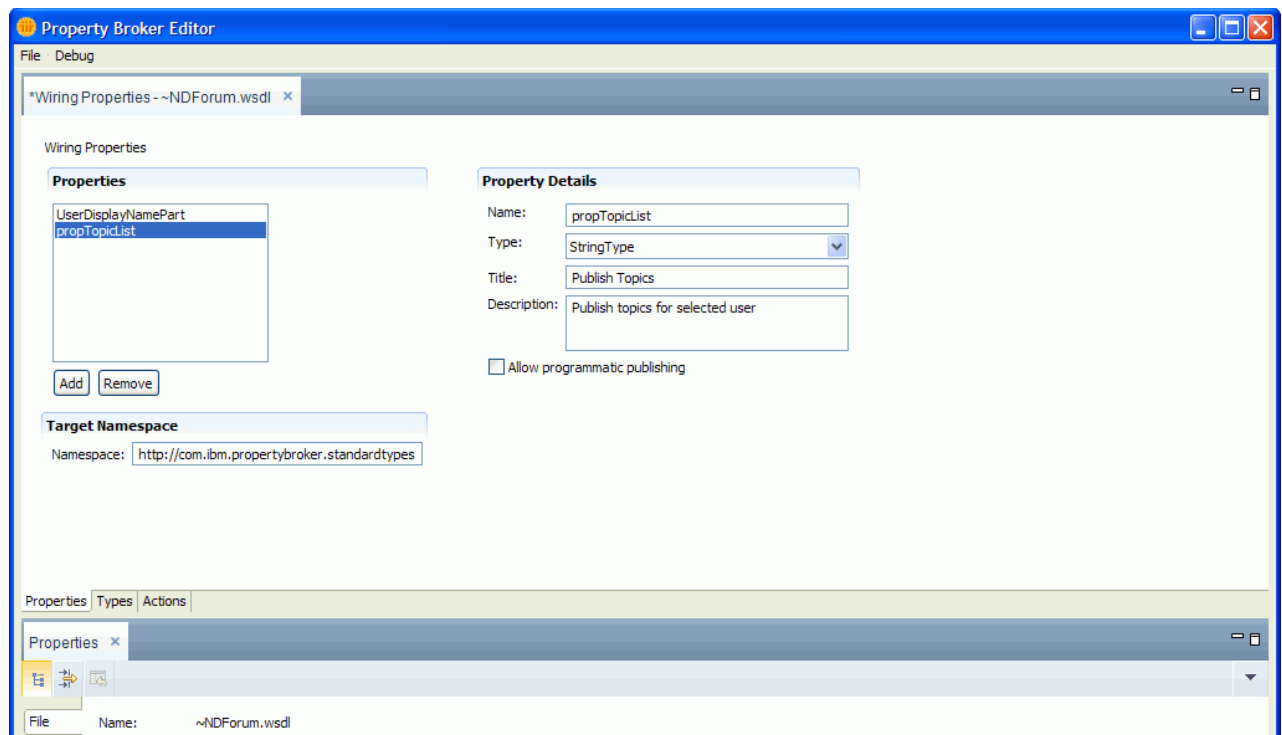
Your Properties tab should be identical to the following screen capture.





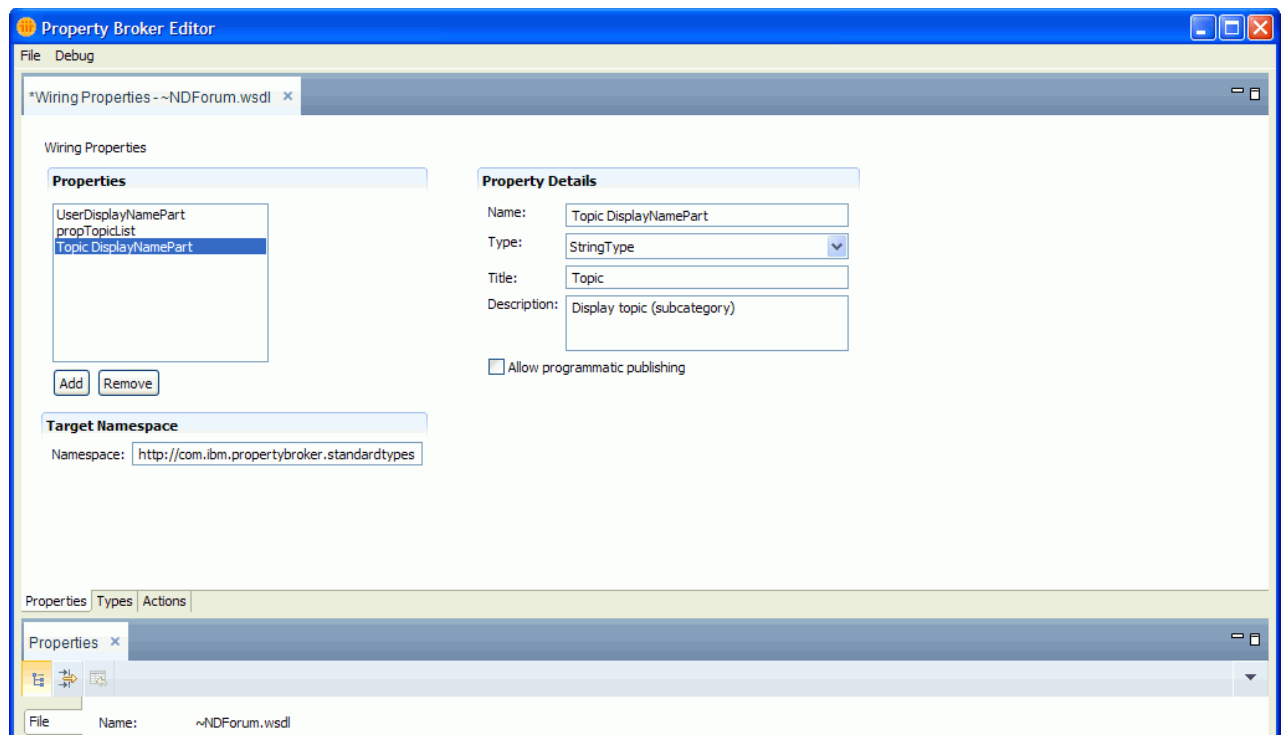
18. If needed, select the **Properties** tab located in the middle of the Property Broker Editor feature.
19. Select the **Add** button that is located on the left.
20. In the **Property Details** section displayed in the upper right:  
Type "propTopicList" in the **Name** field  
Select "StringType" for the **Type** field  
Type "Publish Topics" in the **Title** field  
Type "Publish topics for selected user" in the **Description** field  
Deselect the option **Allow programmatic publishing**
21. In the **Target Namespace** section  
If needed, type "http://com.ibm.propertybroker.standardtypes" in the **Namespace** field

Your Properties tab should be identical to the following screen capture.



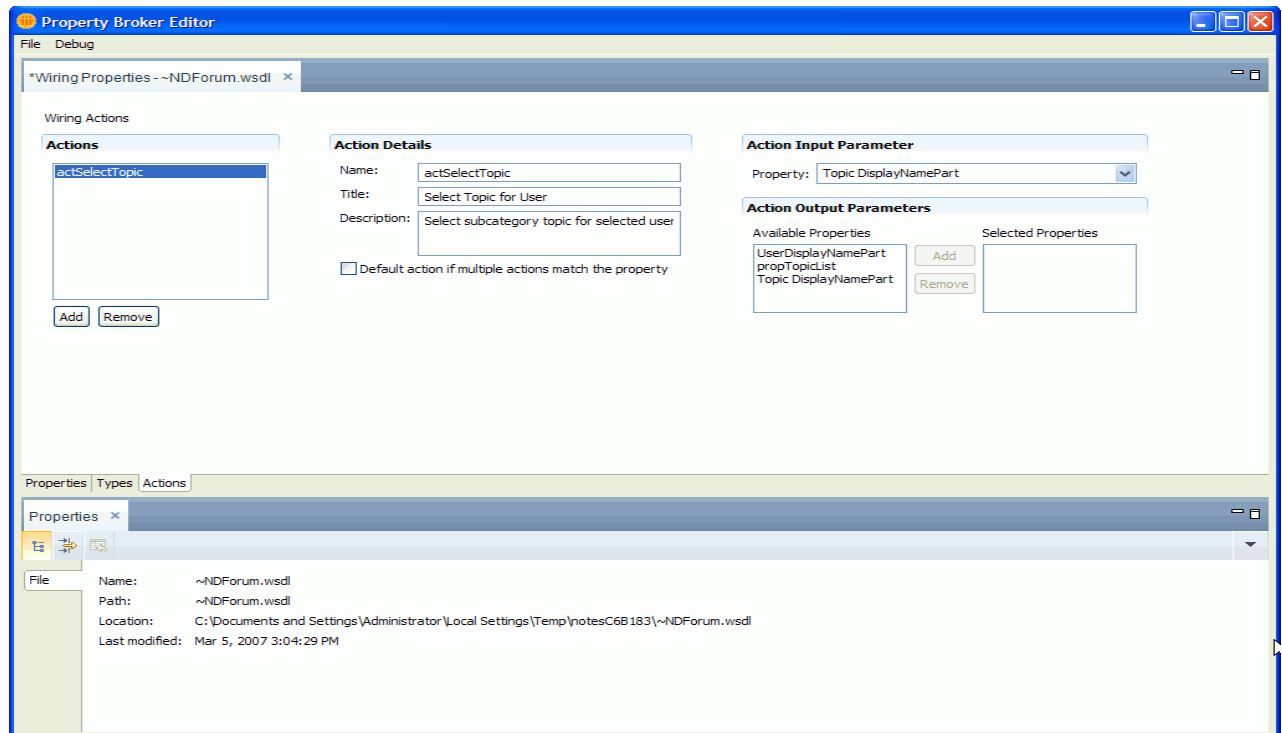
22. If needed, select the **Properties** tab located in the middle of the Property Broker Editor feature.
23. Select the **Add** button that is located on the left.
24. In the **Property Details** section displayed in the upper right:  
Type "TopicDisplayNamePart" in the **Name** field  
Select "StringType" for the **Type** field  
Type "Topic" in the **Title** field  
Type "Display topics (subcategory)" in the **Description** field  
Deselect the option **Allow programmatic publishing**
25. In the **Target Namespace** section  
If needed, type "http://com.ibm.propertybroker.standardtypes" in the **Namespace** field

Your Properties tab should be identical to the following screen capture.



26. Select the **Actions** tab located in the middle of the Property Broker Editor feature
27. Select the **Add** button located on the left. This action will be used to change the selection in the Notes view to the subcategory for the previously selected user.
28. In the **Action Details** section in the middle area:
  - Type "actSelectTopic" in the **Name** field
  - Type "Select Topic for User" in the **Title** field
  - Type "Select subcategory topic for selected user" in the **Description** field
  - Leave the **Default action if multiple actions match the property** option deselected
29. In the **Action Input Parameter** section on the upper right:
  - Select "TopicDisplayNamePart" from the **Property** list

Your **Actions** tab should be identical to the following screen capture.

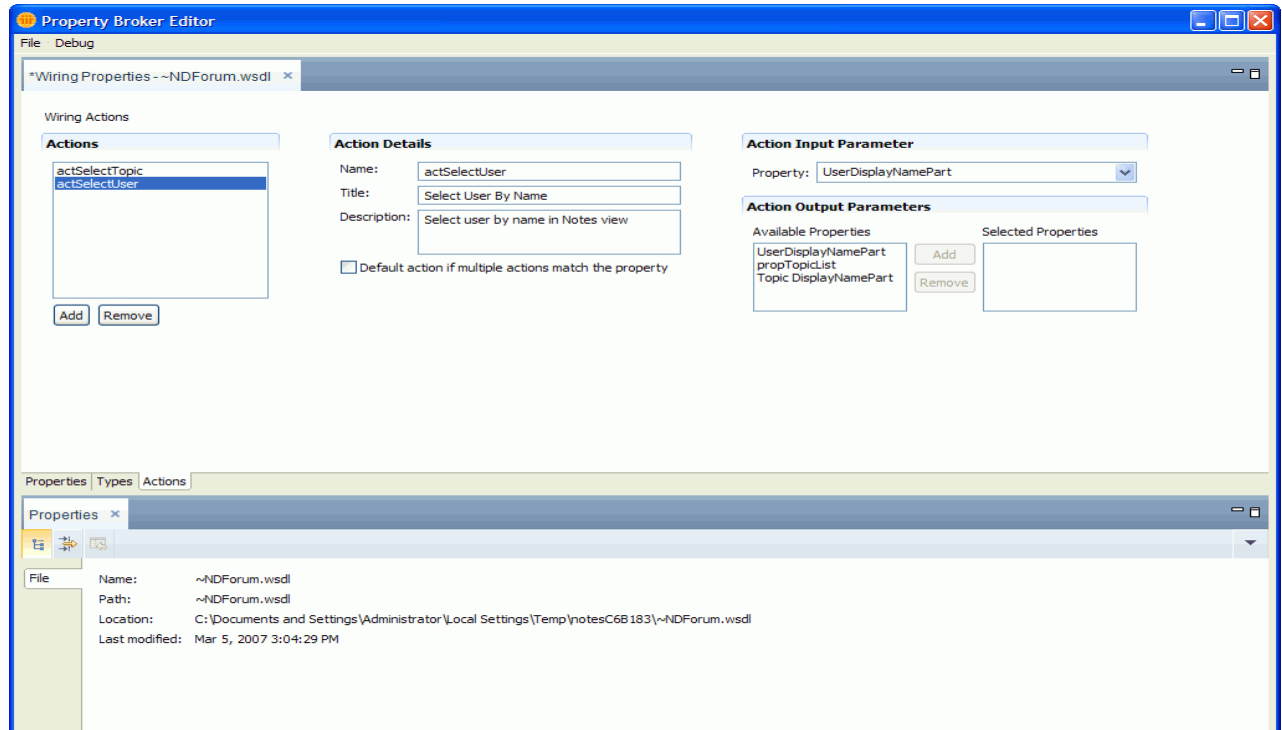


30. Select the **Actions** tab located in the middle of the Property Broker Editor feature

31. Select the **Add** button located on the left. This action will be used to change the selection in the Notes view to the category that matches the name of the selected user.
32. In the **Action Details** section in the middle area:  
Type "actSelectUser" in the **Name** field  
Type "Select User By Name" in the **Title** field  
Type "Select user by name in Notes view" in the **Description** field  
Leave the **Default action if multiple actions match the property** option deselected

In the **Action Input Parameter** section on the upper right:  
Select "UserDisplayNamePart" from the **Property** list

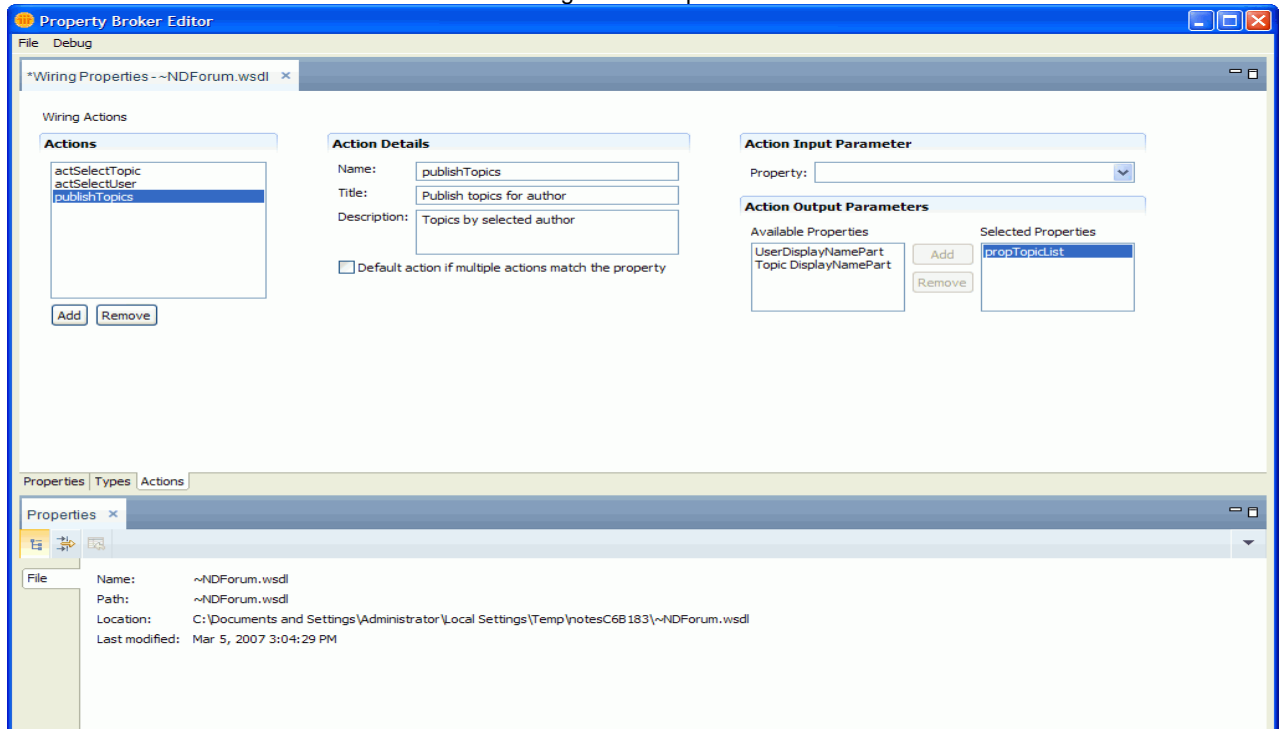
Your **Actions** tab should be identical to the following screen capture.



33. If needed, select the **Actions** tab located in the middle of the Property Broker Editor feature

34. Select the **Add** button located on the left to add a third action. This action will be used publish the list of found subcategories for the selected user and number of documents in the found subcategories.
35. In the **Action Details** section in the middle area:  
 Type "publishTopics" in the **Name** field  
 Type "Publish Topics For User" in the **Title** field  
 Type "Topics by selected user" in the **Description** field  
 Leave the **Default action if multiple actions match the property** option deselected
36. Leave the Property field empty in the **Action Input Parameter** section on the upper right:
37. In the **Action Output Parameter** section:  
 Select **propTopicList** in the **Available Properties** list and click **Add** to place it in the **Selected Properties** list.

Your **Actions** tab should be identical to the following screen capture.



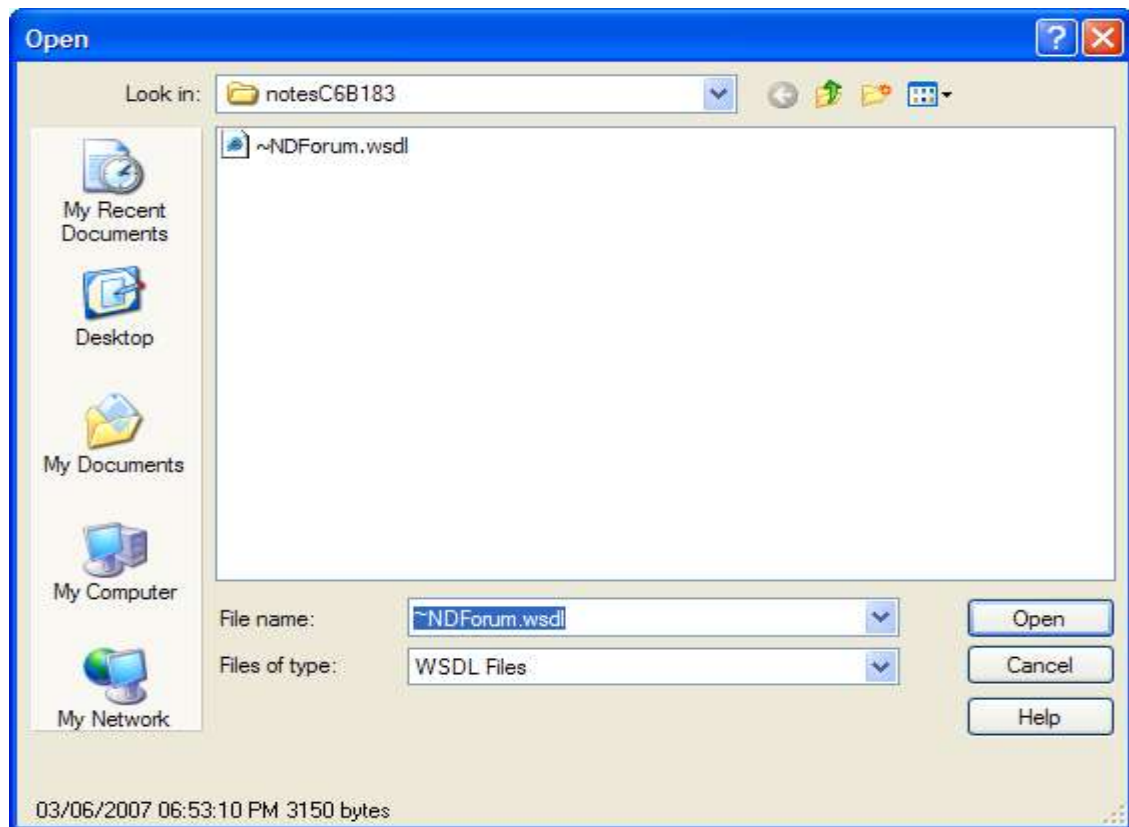
38. Select **File > Save** from the Property Broker Editor menu.

39. Select **File > Exit** from the Property Broker Editor menu.

**Result:** Property Broker Editor closes and you are returned to your Domino Designer 8 Beta client.

40. In the Composite Applications Properties work pane, click the Refresh button on the top right of the work pane.

**Result:** The Open Dialog appears as shown in the following screen capture.



41. Select the WSDL file you updated in the Property Broker Editor, ~NDForum.wsdl, and click **Open**.

**Result:** Your updated WSDL is stored in the Notes application design element.

By completing this procedure, you enabled NSF components from the NDForum Notes application to include properties to be published and actions to be executed. In later lessons, you will use Domino Designer 8 Beta to associate the properties and actions listed in this WSDL with Notes design elements and specify what information is published as a property and what logic will be executed as actions.

If you wish to examine the WSDL content in XML format, use the **Open With** button in the work pane then select a text editor such as Wordpad.

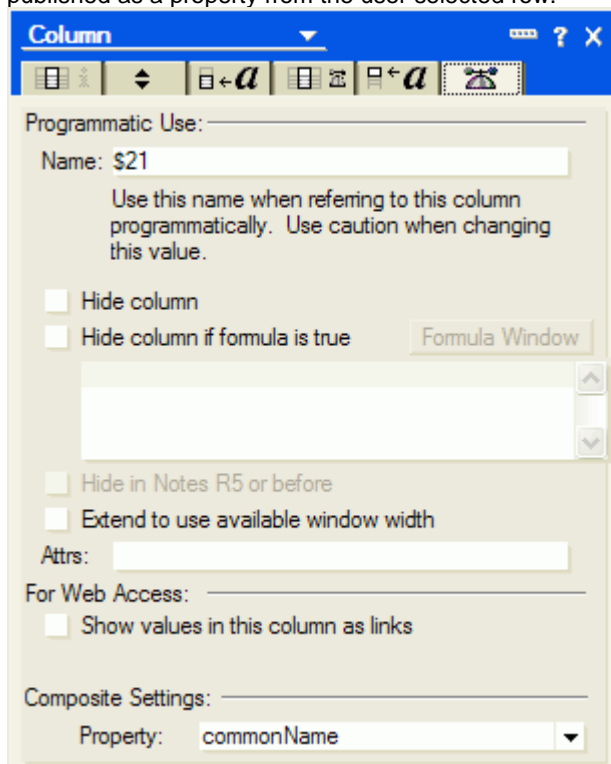
## Lesson 2 - Define Notes view columns that publish properties

In this lesson you examine the Contacts application to observe how the Domino Designer 8 Beta client can be used to publish a property from a Notes view column when the end user selects a new row in the view.

### Procedure 2.1 - Defining Notes view columns that publish properties

Complete the following steps to observe how a Notes view column can publish a component property:

1. In Domino Designer 8 Beta, click **File > Application > Open** and select **Names.nsf** from the dialog.
2. Click **Views** in the design pane on the left to display the view list in the work pane.
3. Select **My Contacts** from the work pane and press **Enter** to open the view in design mode and select **OK** in the warning dialog.
4. In the view, click on the second column, labeled **Name**, to select it.
5. Use **Design > Column Properties** to open the properties box for the selected column.
6. Select the last tab of the properties box and observe **commonName** is selected in the **Composite Settings Property** list as shown in the following screen capture. This option specifies that the column value will be published as a property from the user selected row.



7. Close the properties box.
8. Click **File > Close** to close the view. Do not save any changes you may have inadvertently made.

You have observed how a property can be published from a Notes view. Any one property defined in the WSDL for the Notes application can be selected from this view column property box option. The Notes Contacts application design template includes this property. In a later lesson you will use this view as an NSF component and wire it in a causal relationship to an NSF component in the NDForum application.

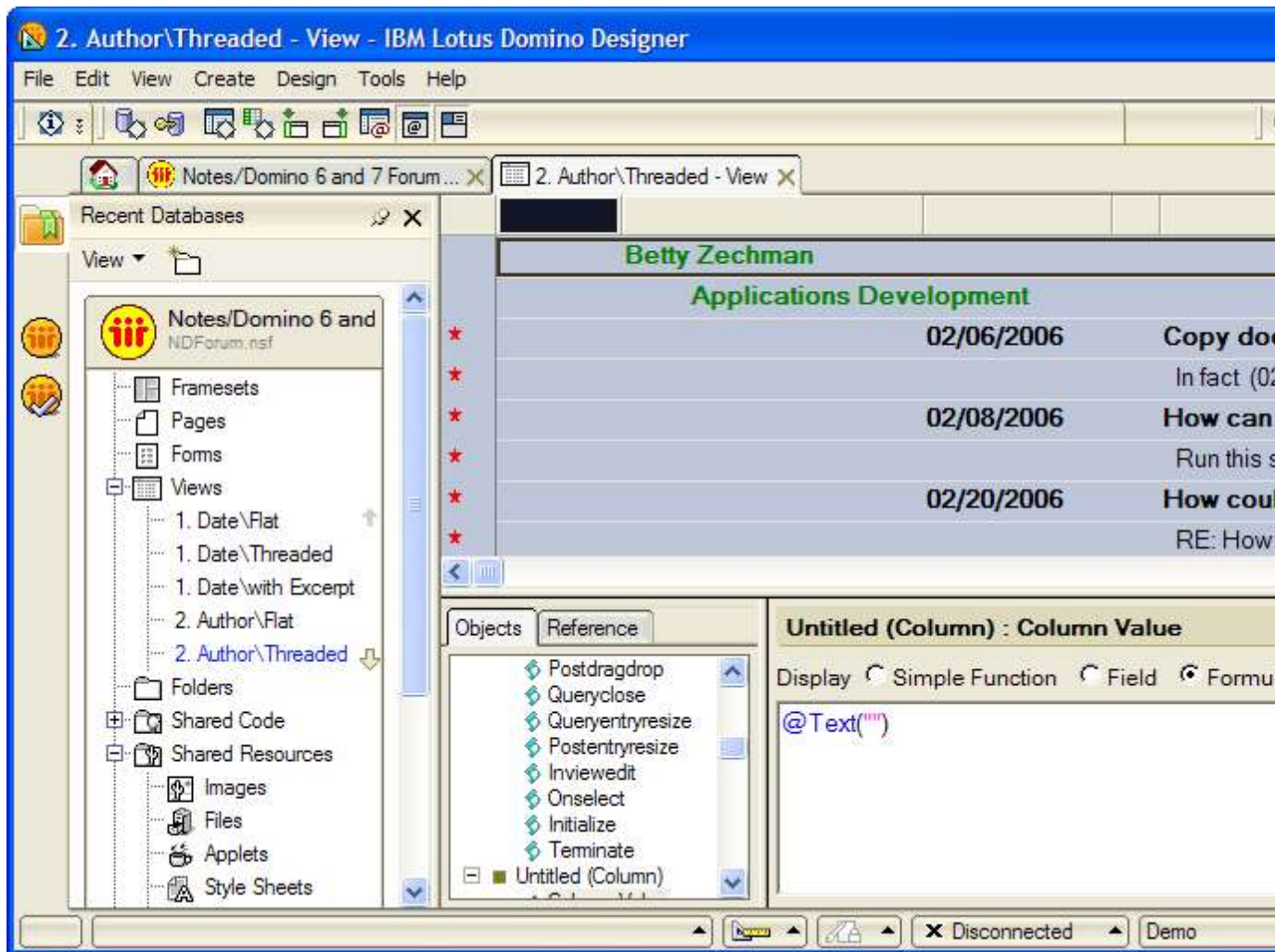
## Lesson 3 - Define Notes actions for inter-component communication

A component action is the pre-defined logic to be executed when the property to which the action is wired is published. You use Domino Designer to define actions on Notes design elements that will be used as NSF components.

### Procedure 3.1 - Defining actions on a Notes form

Complete the following steps to define the logic associated with a NSF component:

1. If Domino Designer 8 Beta is not already running, start it now and open NDForum.nsf.
2. If needed, switch to the Views list by clicking **Views** in the designer pane on the left.
3. Click the view **2. Author\Threaded** view and press **Enter** to open it in design mode. Your screen should look similar to the following screen capture.





4. Click **Create > Action > Action...** to add a new action to this view.  
**Result:** The action pane opens in the right top and the properties box appears for the new action.
5. On the first tab of the properties box, type "SelectPerson" in the **Name** field.

**Action**

Name: SelectPerson

Label:

Target Frame:

Type: Button

Value:

Display

- ☐ Include action in Action bar
  - ☐ Only show icon in Action bar
  - ☐ Right align action control
- ☐ Include action in Action menu
- ☐ Include in right mouse button menu

Icon

- ☒ None
- ☐ Notes
- ☐ Custom

6. Deselect the options **Include action in Action bar** and **Include action in Action menu**.
7. On the last tab of the properties box, click **actSelectUser** in the **Composite Settings Action Name** list at the bottom. This list of actions is from the WSDL you created earlier.

**Action**

☐ Publish Action with OLE object

When action is chosen

- ☐ Close OLE object and return to Notes
- ☒ OLE object remains open
- ☐ Bring document window to front

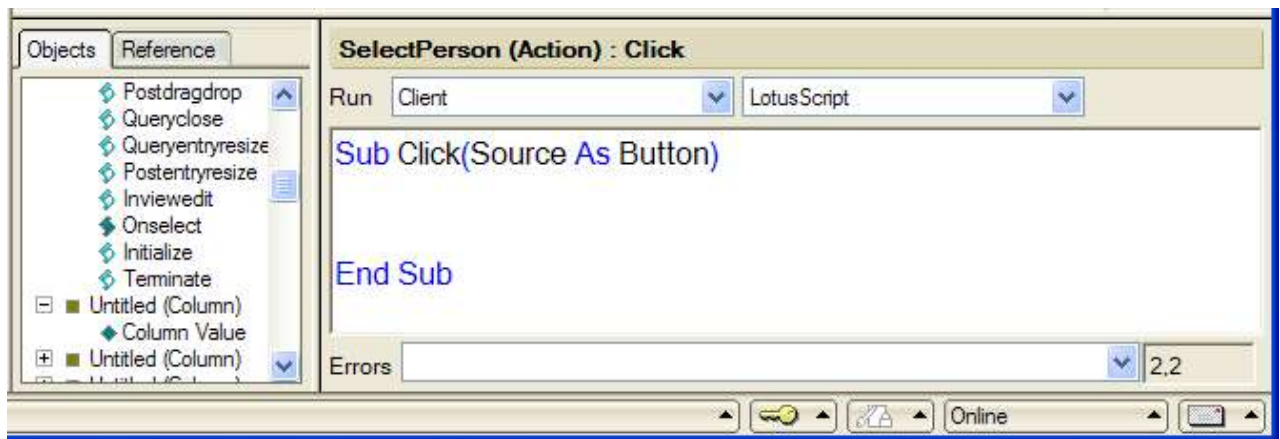
NotesFlow Publishing

Composite Settings

Action Name: actSelectUser

8. Close the properties box.

9. Click the newly created action **SelectPerson** in the action pane on the top right.
10. In the Domino Designer programmers pane at the bottom right, click **Client** and **LotusScript** for this action. The Client option may already be selected.



11. Input the following LotusScript code in the Click subroutine. The source code has been provided in the file: SelectPerson\_Action\_for\_NDForum.txt You can copy and paste this code.

**Note:** The action script will create an object of the new NotesPropertyBroker class and use the property published by another component to change the view selection to the category that matches the published value.

```
Sub Click(Source As Button)
Dim s As New NotesSession, ws As New NotesUIWorkspace
Dim uidb As NotesUIDatabase, db As NotesDatabase, doc As NotesDocument
Set uidb = ws.CurrentDatabase
Set db = s.CurrentDatabase
'Handle error 4719, when script is run without a property broker context
On Error 4719 Goto err4719
'NotesPropertyBroker code
Dim pb As NotesPropertyBroker 'create new NotesPropertyBroker object
Set pb = s.GetPropertyBroker
Dim pbInputProperty As NotesProperty, pbContext As Variant
pbContext = pb.InputPropertyContext 'array of NotesProperty objects
Set pbInputProperty = pbContext(0) 'handle to first NotesProperty object
Dim InputPropertyName As String, Namespace As String
InputPropertyName = pbInputProperty.Name 'name of NotesProperty object
Namespace = pbInputProperty.Namespace 'namespace of NotesProperty object
Dim pbValue As Variant
pbValue = pb.getPropertyValue(InputPropertyName) 'array of values of NotesProperty
cName$ = pbValue(0) 'first value from array NotesProperty object
SkipPropertyBroker:
If cName$ = "" Then Exit Sub
Set doc = db.GetProfileDocument("SelectedAuthor", s.UserName)
Call doc.ReplaceItemValue("SelectedAuthor", cName$)
Call uidb.OpenView("2. Author\Threaded", cName$ , False, False)
Exit Sub
err4719:
Print "Error " & Str(Err) & ": " & Error$
Resume skipPropertyBroker
End Sub
```

11. Click **File > Save** to save your view changes.
12. Click **File > Close** to exit design mode.

You have added an action to the Notes view and completed this lesson. When you later use this Notes view as a NSF component in a composite application, the component can consume a published property that it is wired to and perform the action you defined in your LotusScript code.

## Lesson 4 - Assemble an NSF-based composite application

The Lotus Notes 8 Beta client can access NSF-based composite applications, where the definition is stored in an NSF that runs locally on the Notes 8 Beta client and from the Domino server. The Lotus Notes 8 Beta client can also access composite applications hosted on WebSphere Portal.

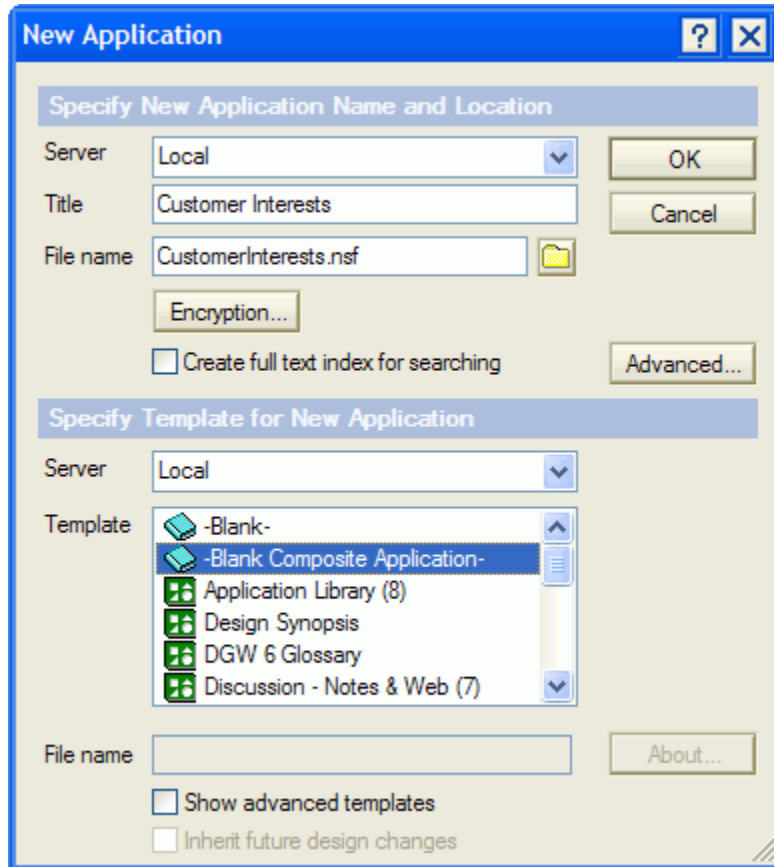
### Procedure 4.1 – Creating a NSF-based composite application

Complete the following steps to add the action to create a new NSF-based composite application:

1. If Lotus Notes 8 Beta is not already running, start it now.
2. Open the Workspace through the Open List in the upper left. Select **Open > Applications > Workspace**.
3. From the menu, click **File > Application > New**

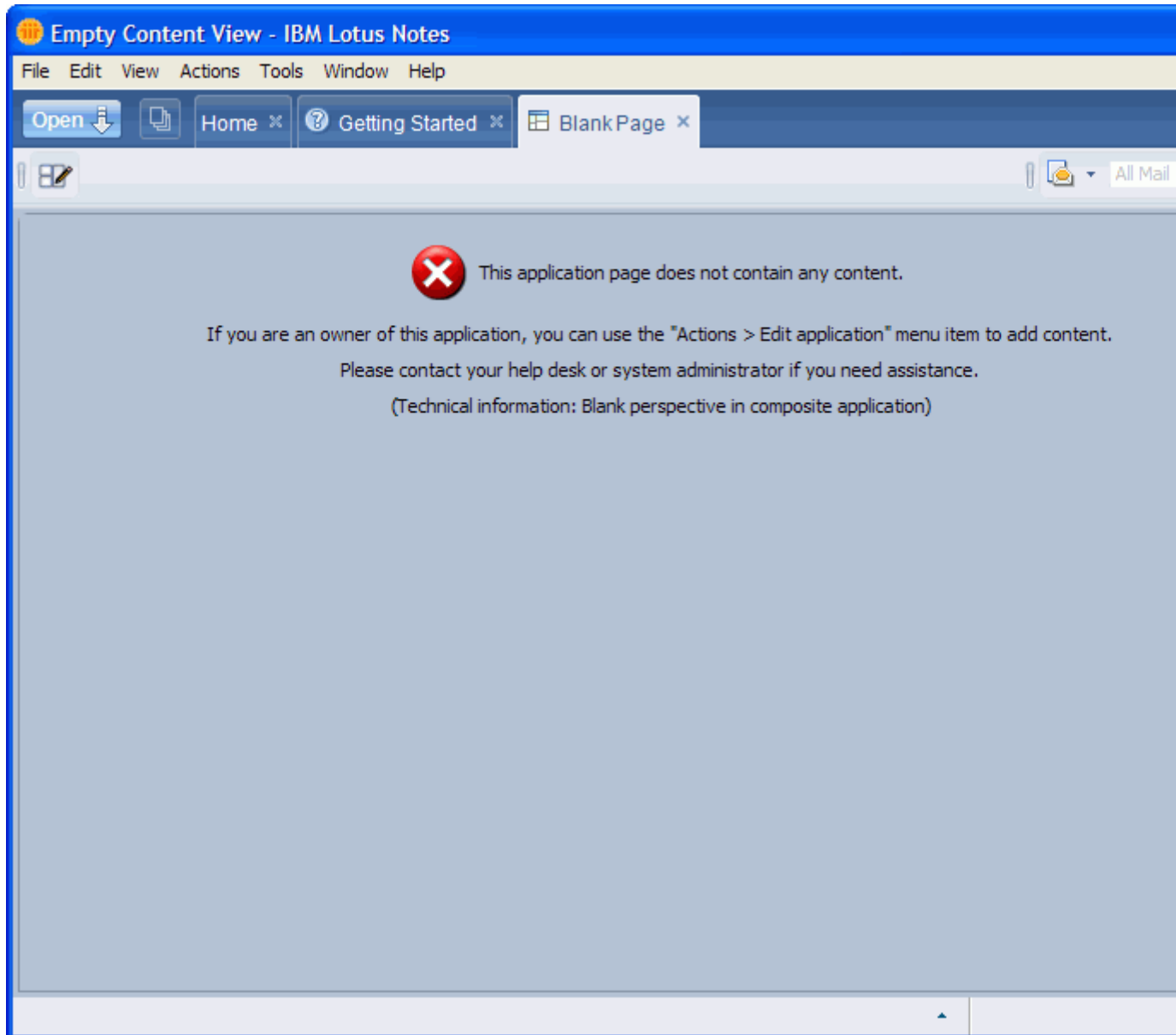
**Result:** The **New Database** dialog appears.

4. Leave the **Server** field value set to Local.
5. Type "Customer Interests" in the **Title** field.
6. Type "CustomerInterests.nsf" in the **File Name** field.
7. Select **-Blank Composite Application-** in the **Template** field.



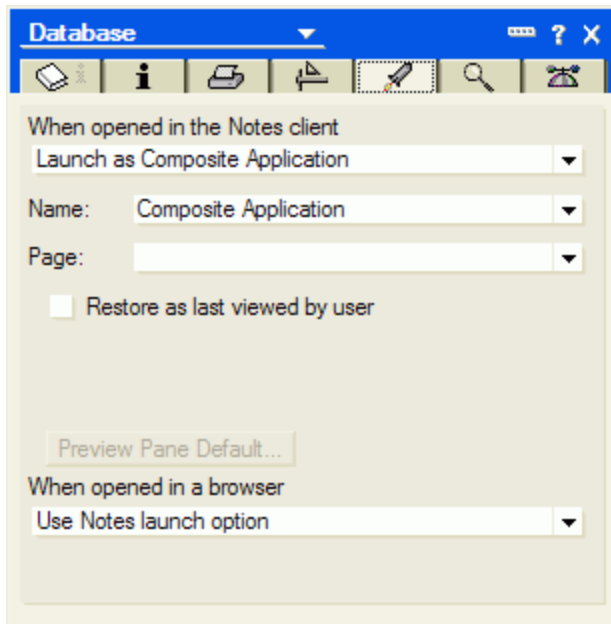
8. Click **OK** to close the dialog box and create the new database.

**Result:** The new database is created and a grey blank window appears as shown in the following screen capture.



**Note:** The newly created NSF-based composite application uses a new database launch property. **When opened in the Notes client** field is set to **Launch as Composite Application** for any NSF created with this template.

This option can be seen by opening the Notes workspace then accessing the application properties. To access the Notes workspace, use the Open List in the upper left of the Notes 8 Beta client, then choose Applications, then choose Workspace.



You have completed this lesson and created a NSF-based composite application that will store the XML in a Notes application design element that the Lotus Notes 8 client will use to present the composite application to the end user. The NSF-based application can be hosted on a Domino server and can be replicated locally.

In the following lessons, you will use the Composite Application Editor (CAE) feature to specify the XML and to aggregate NSF components and Eclipse components as a composite application. You will also use CAE to loosely couple the components in causal relationships by wiring of the properties and actions for inter-component communication.

## Lesson 5 - Add NSF components to the composite application

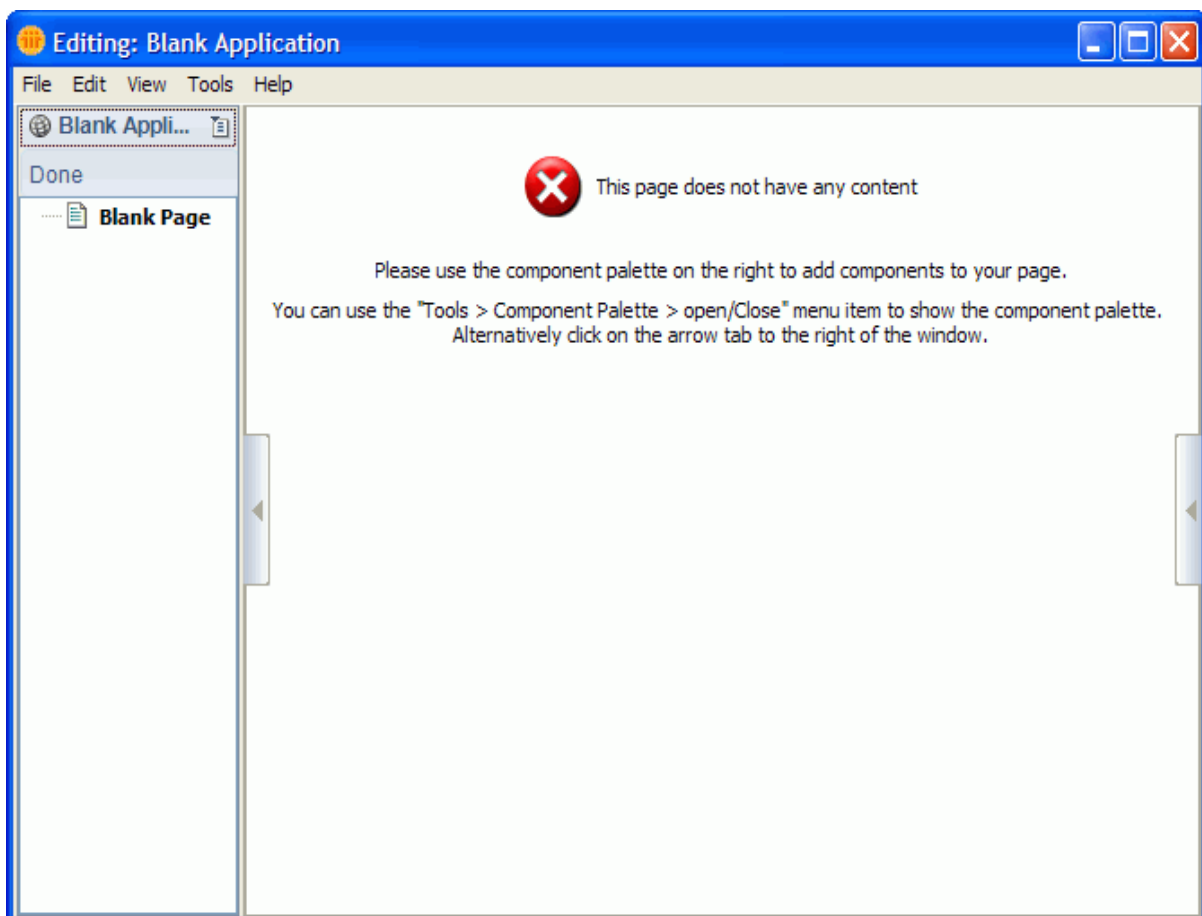
The Composite Application Editor (CAE) feature is a Notes 8 Beta client feature that allows developers and LOB users to define the components in a composite application and to wire those components together into causal relationships. CAE is an install option for the Notes 8 Beta client and can be used to modify composite applications that are NSF-based that run locally or from a Lotus Domino server. CAE can also be used to modify composite applications and composite application templates that are hosted on a Websphere Portal server.

### Procedure 5.1 – Using the Composite Application Editor to assemble a composite application

Complete the following steps to open the Composite Application Editor:

1. If Lotus Notes 8 Beta is not already running, start it now.
2. If needed, Use **File > Open > Lotus Notes Application** to open the **Customer Interests** composite application you created in the previous lesson.
3. From the menu, click **Actions > Edit Application**.

**Result:** The Composite Application Editor appears in a new window.



**Note:** A non-fatal message may appear if you are not connected to a server.

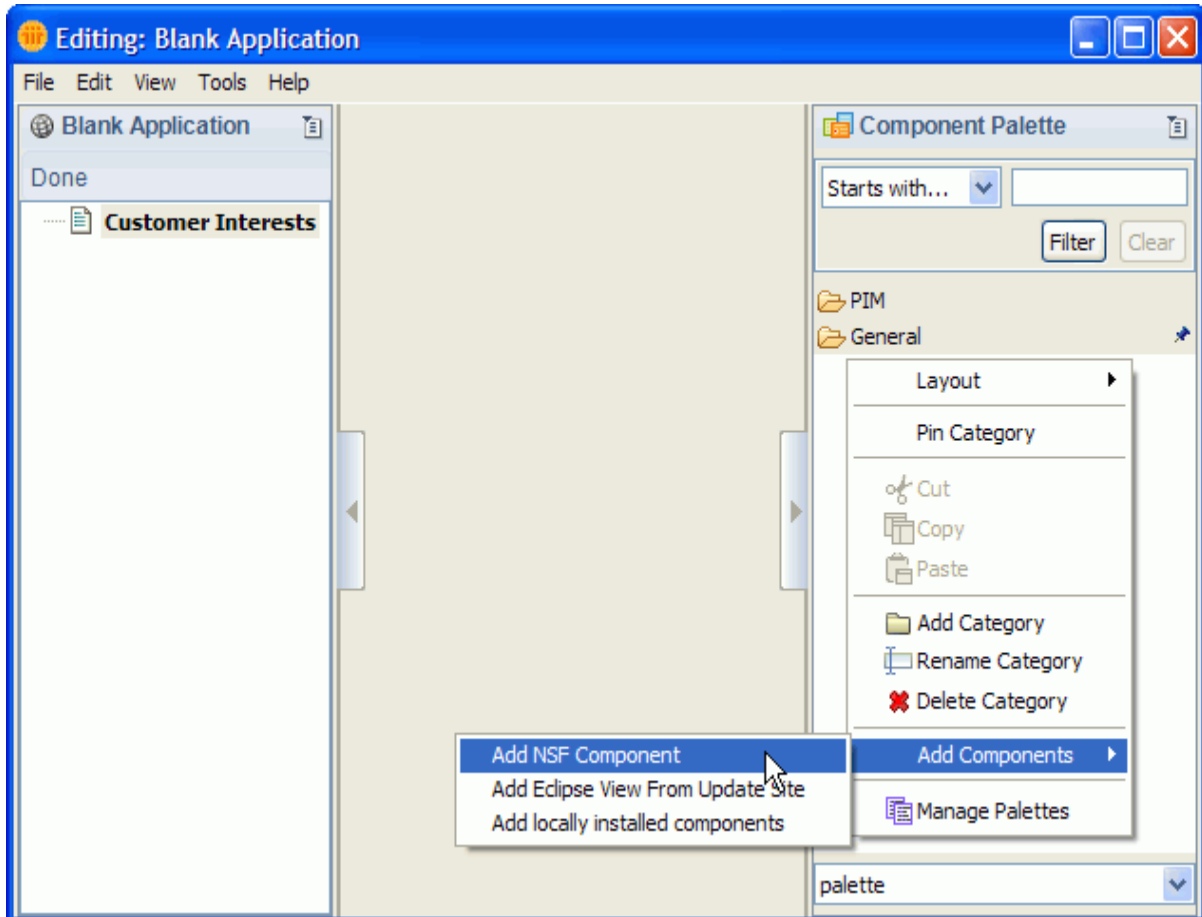
4. In the Navigator pane on the left, select and right-click on the page labeled **Blank Page**. Then select **Rename...** from the right-click menu.
5. Change the name to **Customer Interests** and press Enter to accept the new page name.

**Note:** This new string will appear as the page name in the Composite Application Editor. This string will also be the window tab title when the composite application is open in the Notes 8 Beta client.

### Procedure 5.2 – Add components to the Composite Application Editor Component Palette

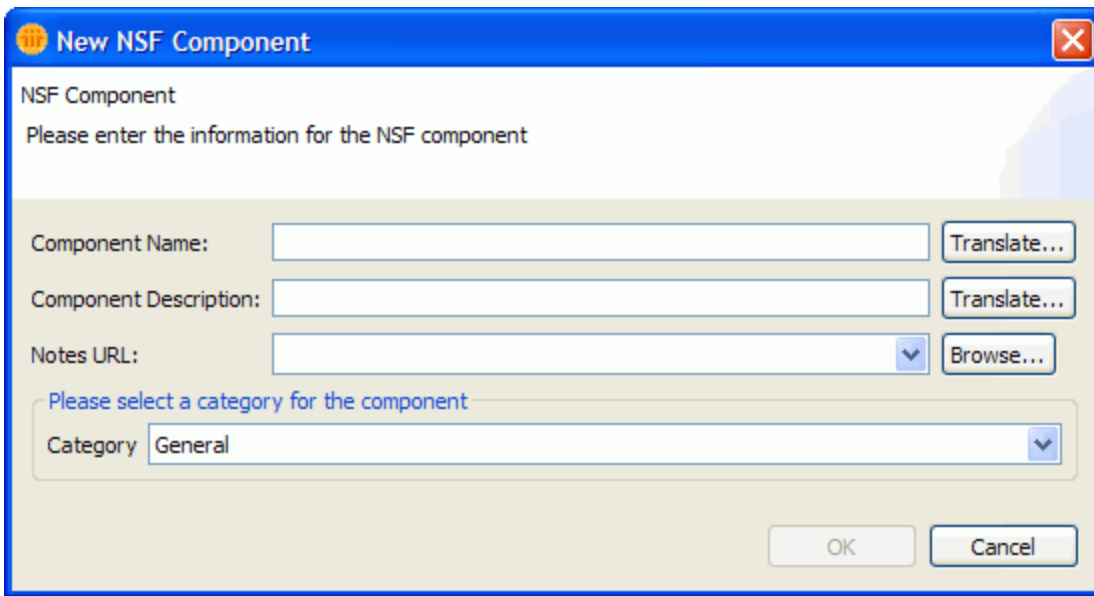
Complete the following steps to add components to the component palette of the Composite Application Editor feature:

1. Expand the **Component Palette** on the right and click the push pin icon on the upper right to pin open the display.
2. Place your mouse below the **General** folder, then right-click and select **Add Components > Add NSF Component** as shown in the graphic below.





**Result:** The **New NSF Component** dialog appears.



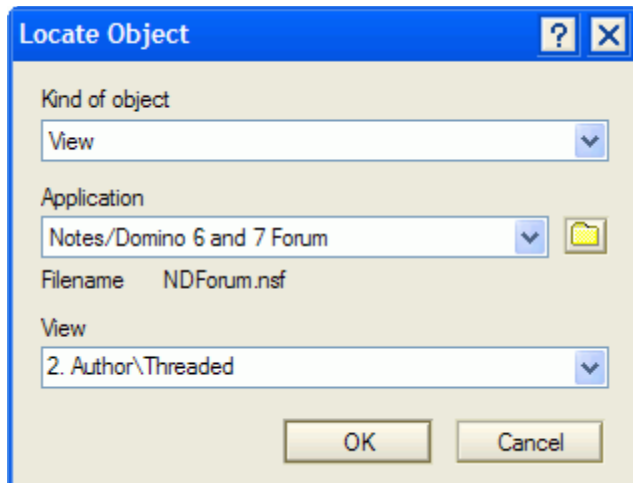
The **New NSF Component** dialog box is shown. It has a blue title bar with the text "New NSF Component" and a close button. The main area is light beige. At the top, it says "NSF Component" and "Please enter the information for the NSF component". Below this are three input fields: "Component Name:", "Component Description:", and "Notes URL:". Each field has a "Translate..." button to its right. Below the "Notes URL:" field is a "Browse..." button. At the bottom, there is a "Category" dropdown menu with "General" selected. At the very bottom are "OK" and "Cancel" buttons.

3. In the **New NSF Component** dialog, type the following values:

Type "NDForum - Author Threaded View" in the **Component Name** field.

Type "NDForum - Author Threaded View" in the **Component Description** field.

4. In the dialog, click the **Browse** button, then select the view "2. Author \Threaded" from NDForum.nsf



The **Locate Object** dialog box is shown. It has a blue title bar with the text "Locate Object" and help and close buttons. The main area is light beige. It contains four fields: "Kind of object" with a dropdown menu showing "View"; "Application" with a dropdown menu showing "Notes/Domino 6 and 7 Forum" and a folder icon button; "Filename" with the text "NDForum.nsf"; and "View" with a dropdown menu showing "2. Author \Threaded". At the bottom are "OK" and "Cancel" buttons.

5. Click **OK** to confirm these settings. These selections will populate the Notes URL field.

- Append "&HideNavigator" to the Notes URL. This option prevents the display of the Notes navigator pane for the component and allots the entire component display area to the Notes view pane.

Your result should match the following screen capture. Notice the Replica ID is used.

**New NSF Component**

NSF Component

Please enter the information for the NSF component

Component Name:

Component Description:

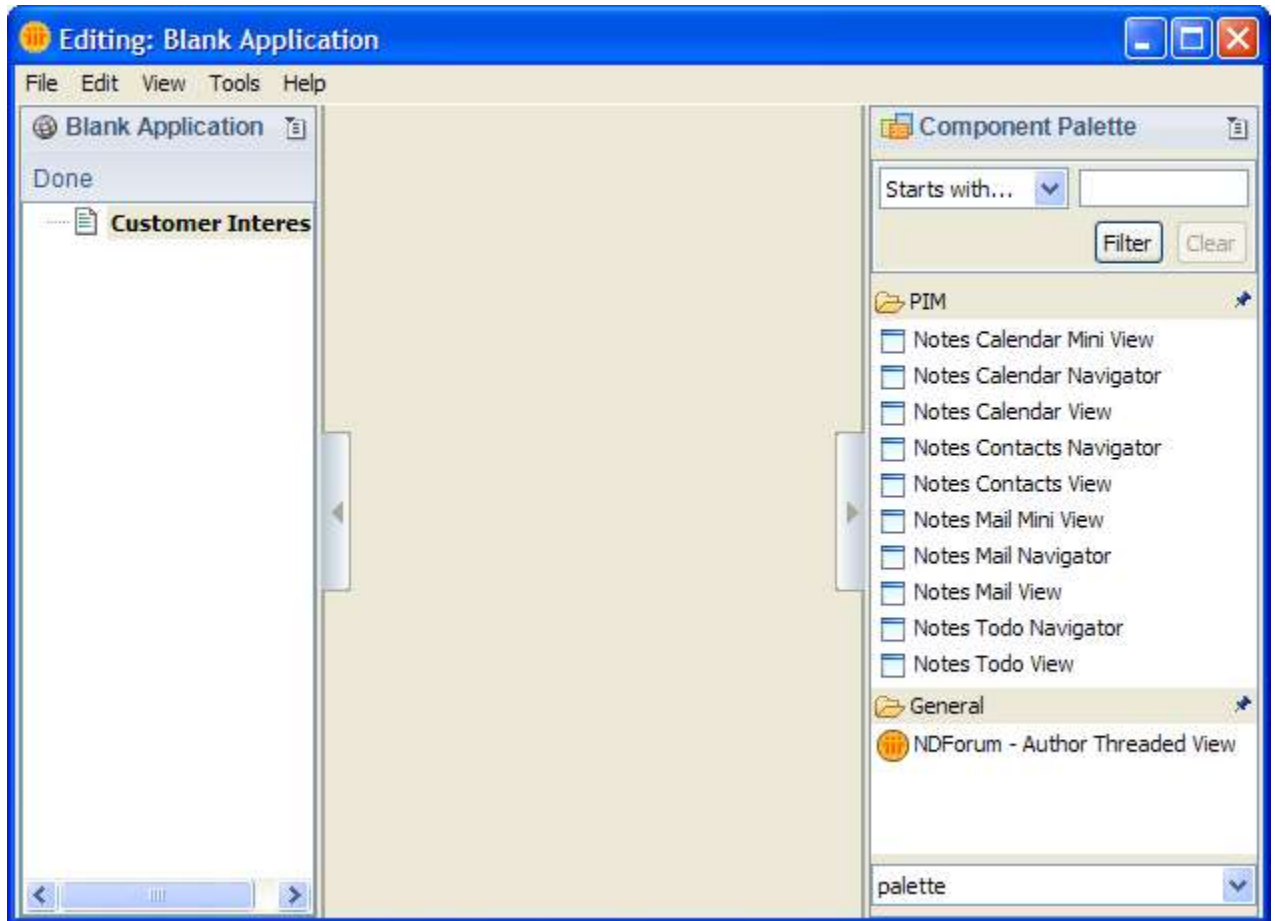
Notes URL:

Please select a category for the component

Category:

- Click **OK** to add this NSF component to the component palette.

**Result:** The component appears in the Component Palette on the right as shown in the following screen capture.



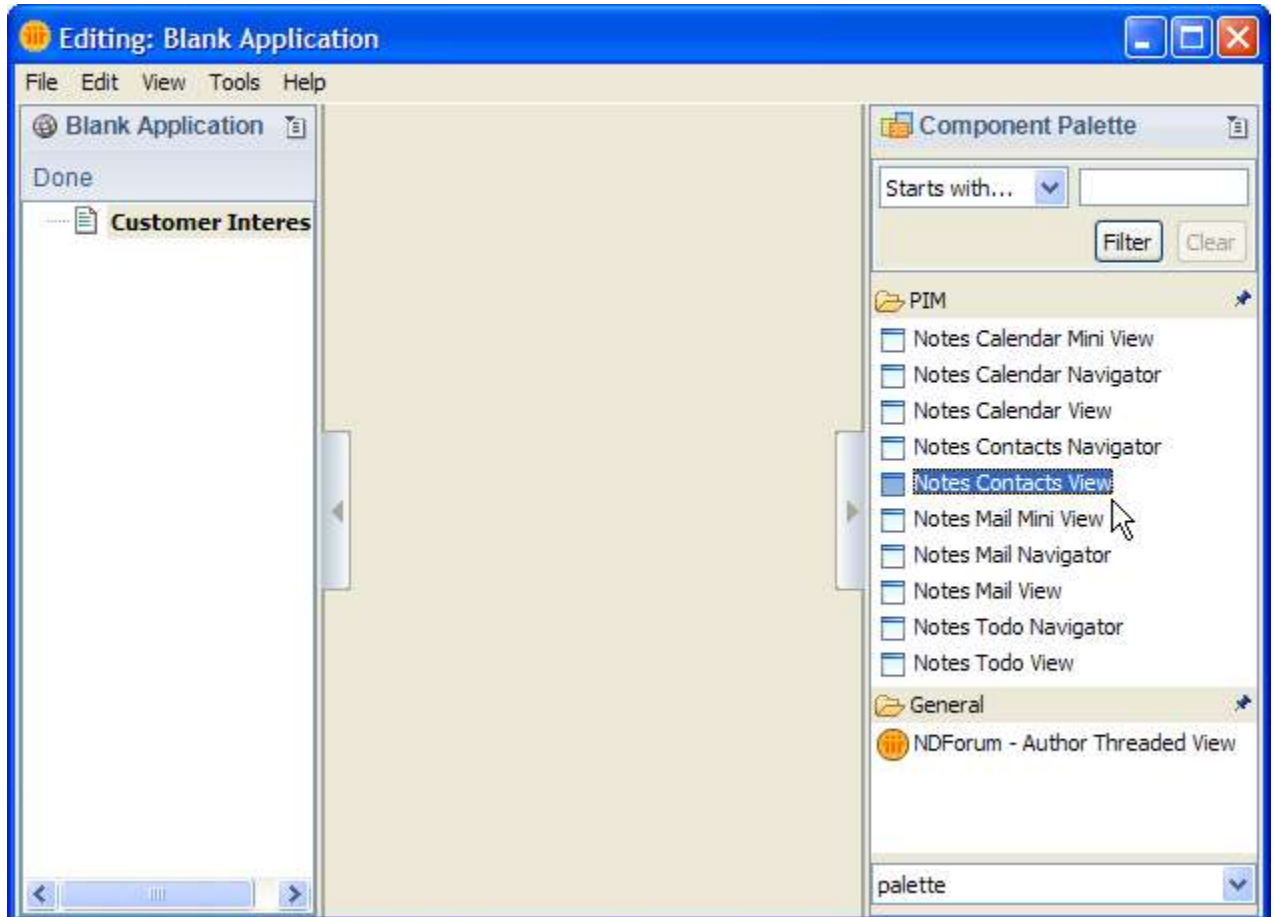
**Note:** Your newly added component is listed in the **General** category of the **Component Palette**. You may wish to create new categories in your Component Palette to organize your components.

### Procedure 5.3 – Define visual placement of components using drag and drop

Complete the following steps to add components to your composite application:

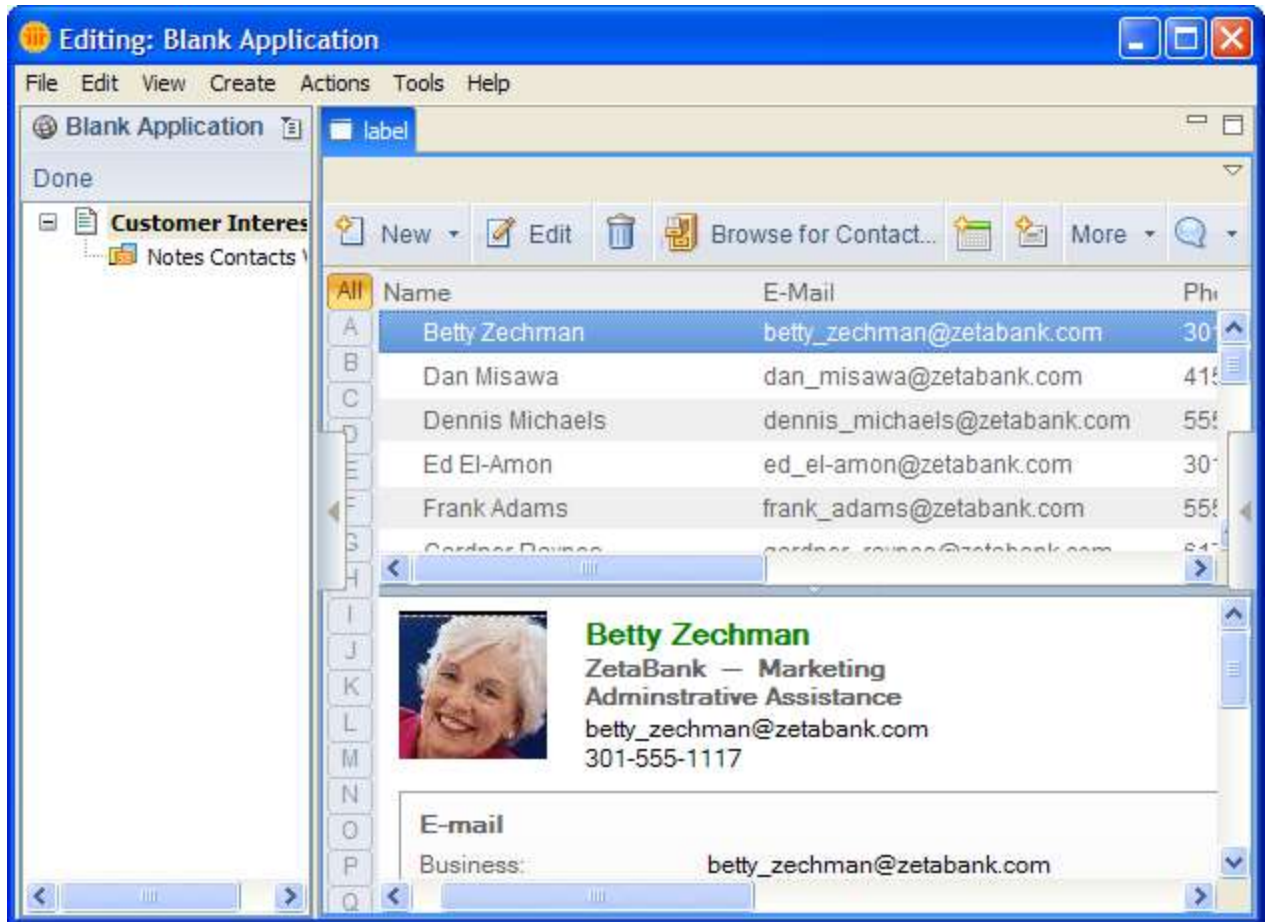
1. Select **Notes Contacts View** in the **PIM** category of the component palette on the right.

The PIM components allow component assemblers to include the Notes Mail, Calendar, Todo and Contacts applications for the current Notes client end user as components.



2. Drag the component from the component palette on the right to the **Blank Application** area in the center.

**Result:** The PIM component appears in the composite application as shown in the following screen capture.

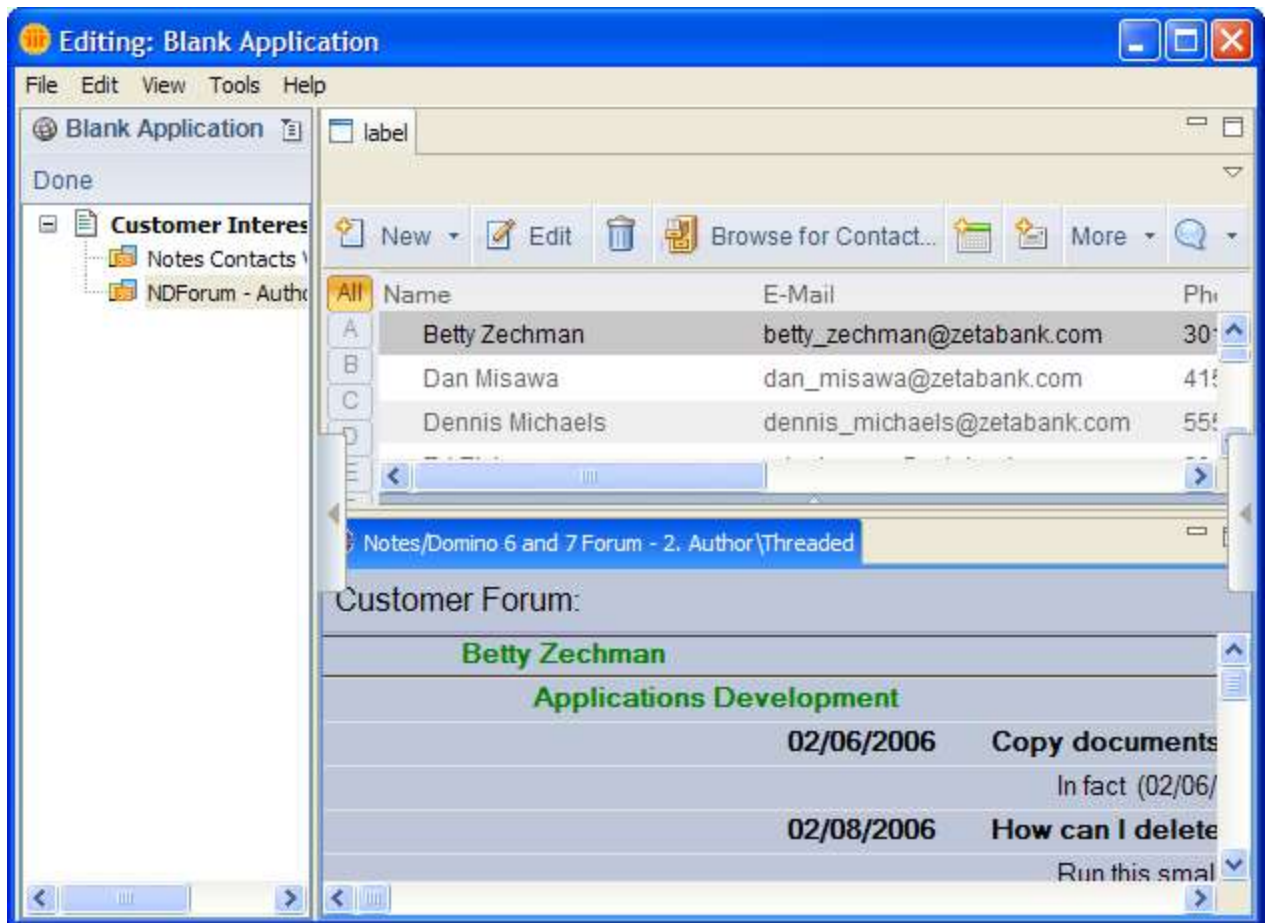


- Repeat the previous step to drag and place the **ND Forum - Author Threaded View** component. Drag the component from the Component Palette on the right to the lower half of the center area.

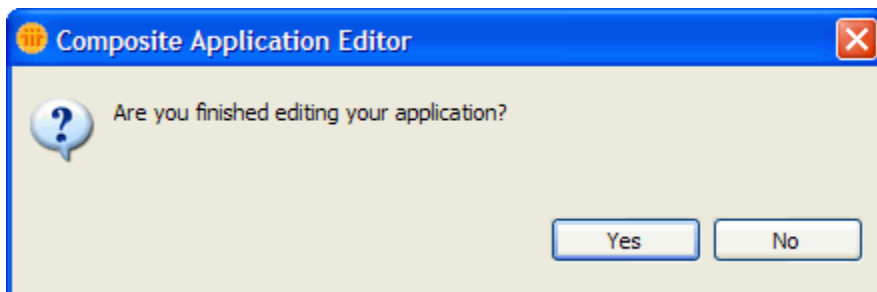
**Tip:** Drag your mouse to the bottom border until your mouse pointer changes to an arrow.



**Result:** The Composite Application Editor shows two components as in the following screen capture.

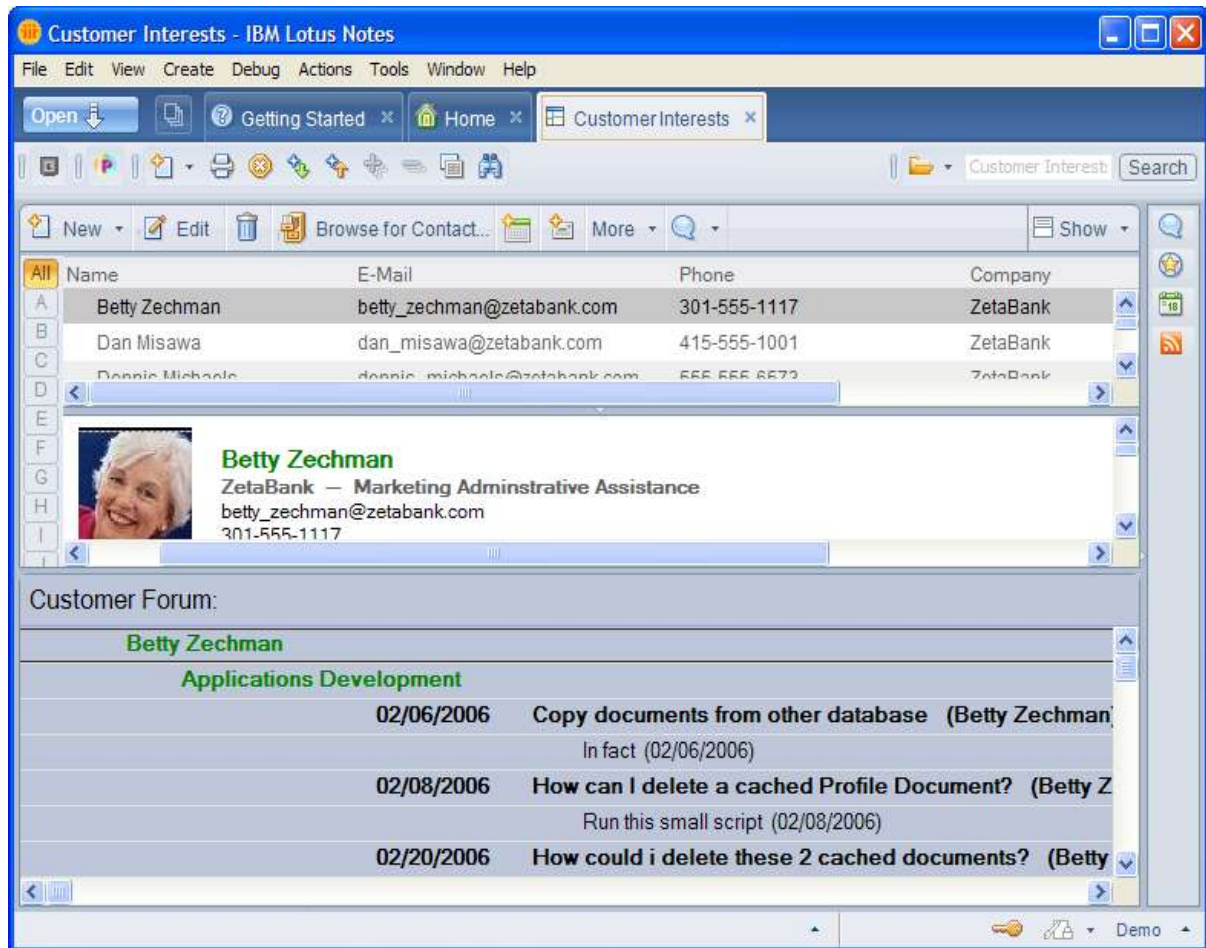


- Click **File > Finished Editing** to close the Composite Application Editor and save your changes.
- When prompted, click **Yes** to save your changes.



**Result:** The Composite Application Editor closes and the composite application appears in the Notes client.

At this time, there is no inter-component communication in your composite application, but you will add this functionality in the next lesson.



You have completed the lesson and assembled an NSF-based composite application using the Composite Application Editor feature of the Notes 8 Beta client. In this lesson you added NSF components. In later lessons you will add an Eclipse component and will observe how composite applications in the Notes 8 Beta client can include components built in different technologies. In the next lesson, you will wire the two NSF components together in a causal relationship.



## Lesson 6 - Wire components in the composite application

Components can be loosely coupled by wiring properties to actions in 1-to-1 or 1-to-many relationships. As the assembler, you define the wiring for a composite application through the CAE (Composite Application Editor) feature. The property broker functionality of the Notes 8 Beta client is the runtime environment for the wiring and inter-component communication.

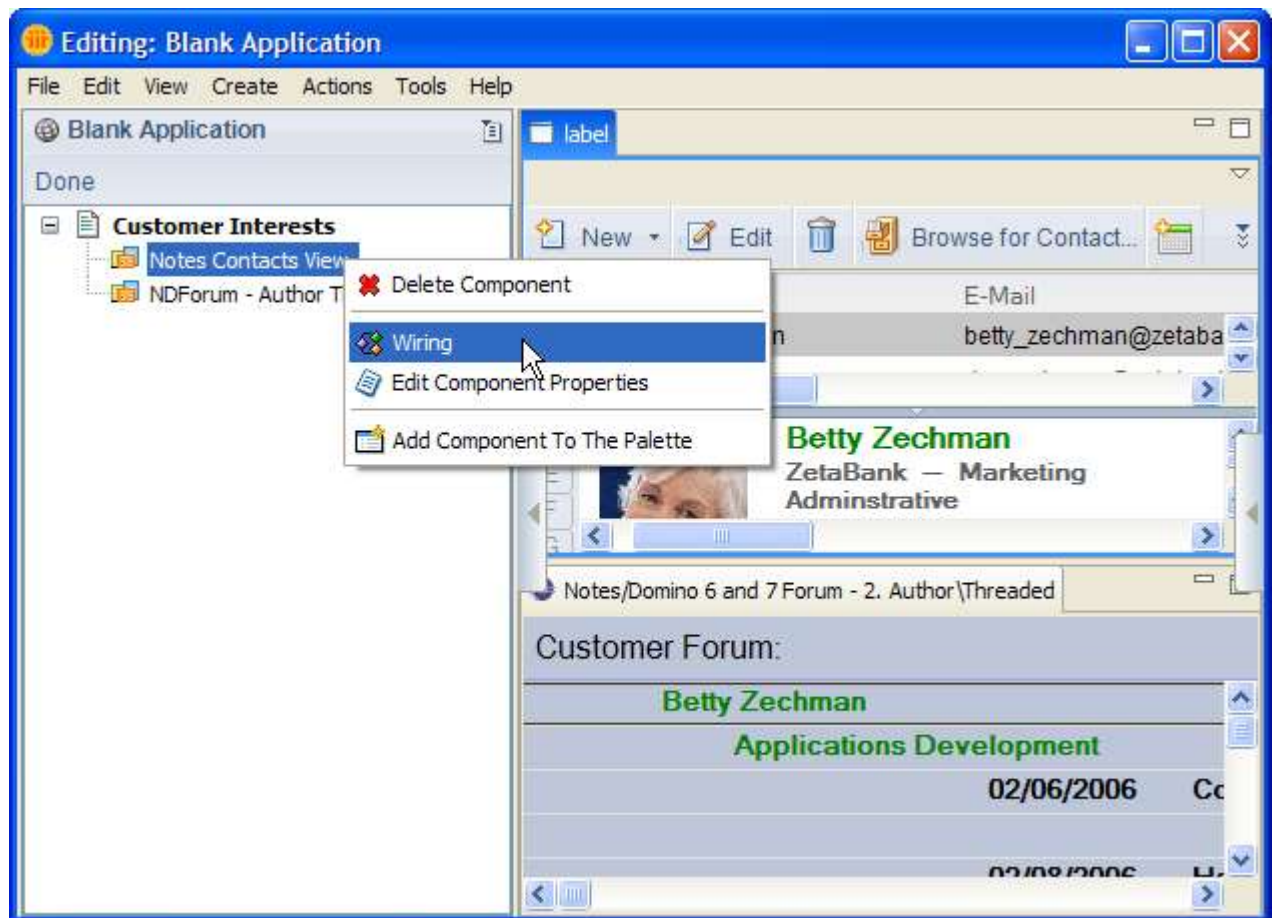
### Procedure 6.1 – Wiring components in a causal relationship

Complete the following steps to form causal relationships between your components:

1. If Lotus Notes 8 Beta is not already running, start it now.
2. Click **File > Open > Lotus Notes Application** to open the **Customer Interests** composite application you modified earlier.

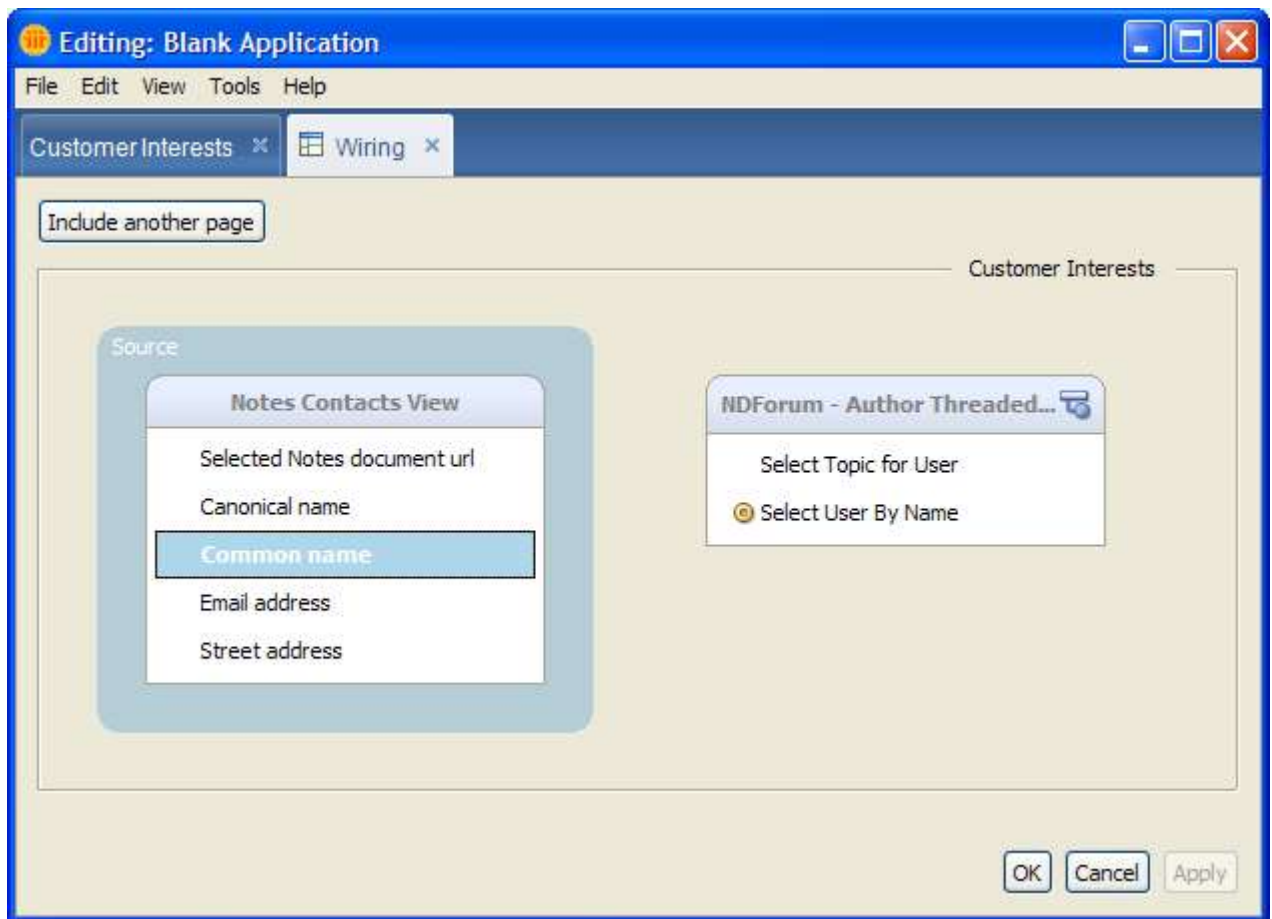
**Result:** The composite application appears and displays the two components you added earlier.

3. From the menu, click **Actions > Edit Application** to start the Composite Application Editor feature.
4. Select **Notes Contacts View** from the navigator on the left, then right-click and select **Wiring**.





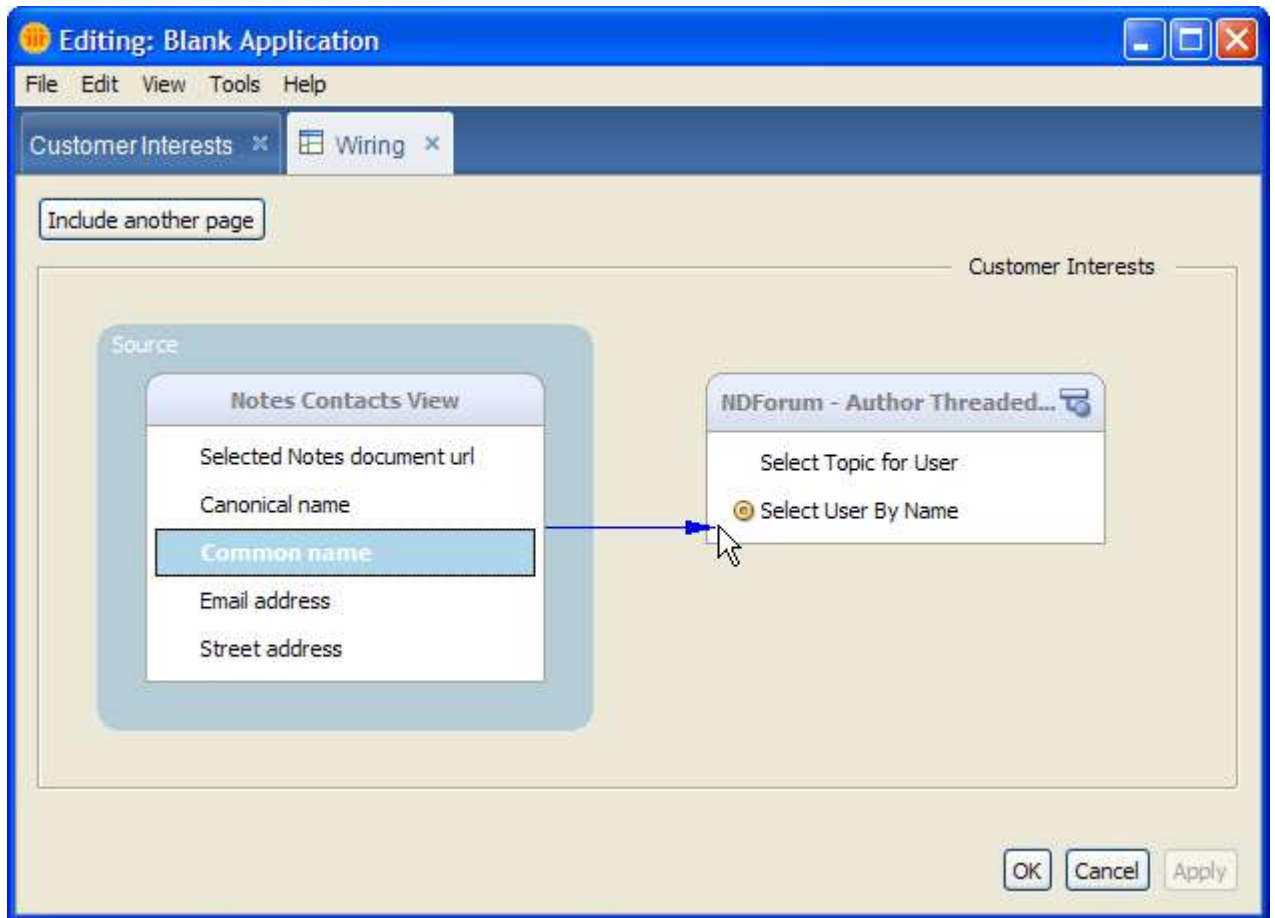
**Result:** The Composite Application Editor wiring interface appears, as shown below.



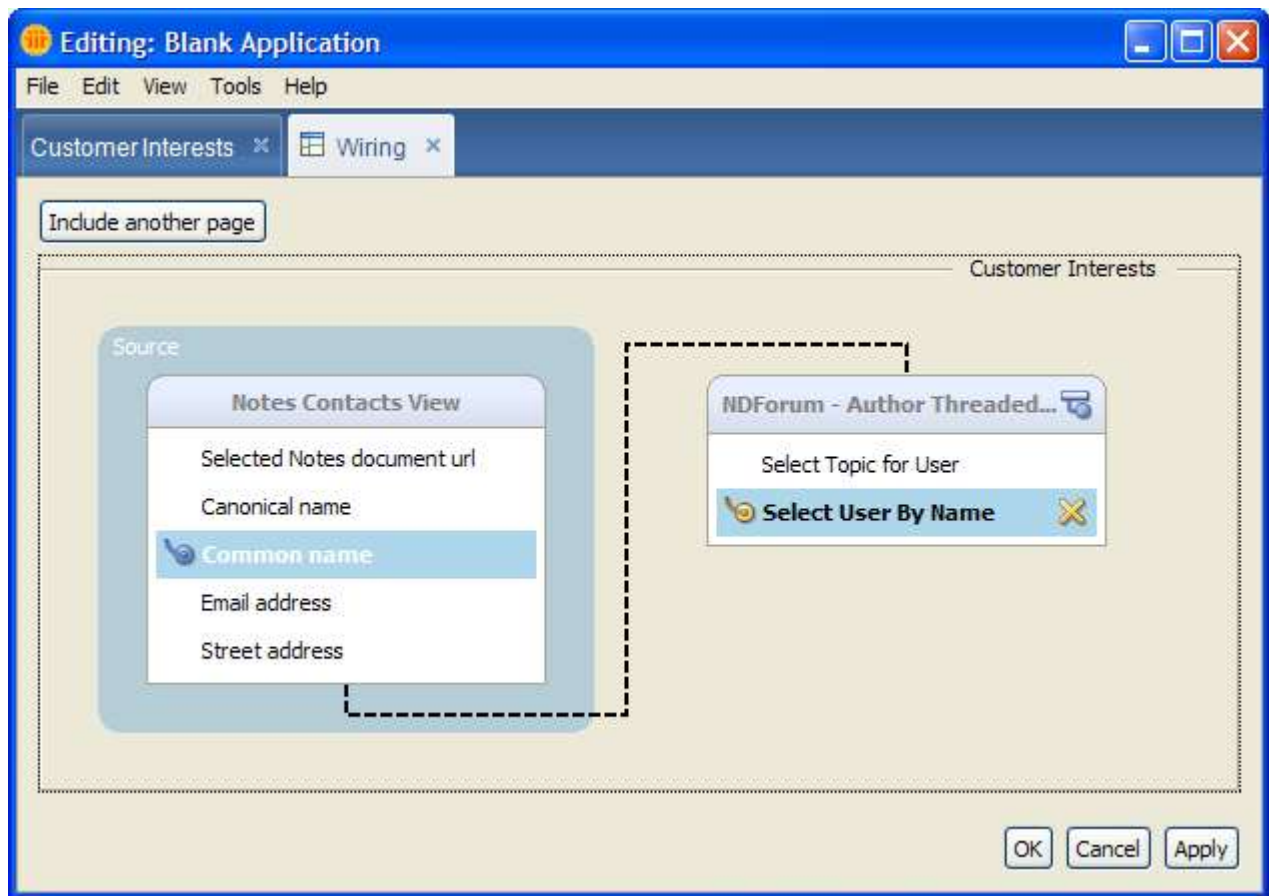
The wiring interface displays the properties for the component labeled source and the actions for any other components placed on the same page. When any one property in the source component is selected, actions that have matching namespace and datatype values, display a visual that indicates wiring is possible.

5. Select the **Common name** property in **Notes Contacts View** component, which is the component labeled 'Source'.
6. Hold your left mouse button and drag to the **Select User By Name** action listed in the **NDForum – Author Threaded View** component then release the mouse button.

**Note:** A blue arrow appears as you drag from the selected property.



**Result:** A dotted line and visuals indicate wiring between the property and action.

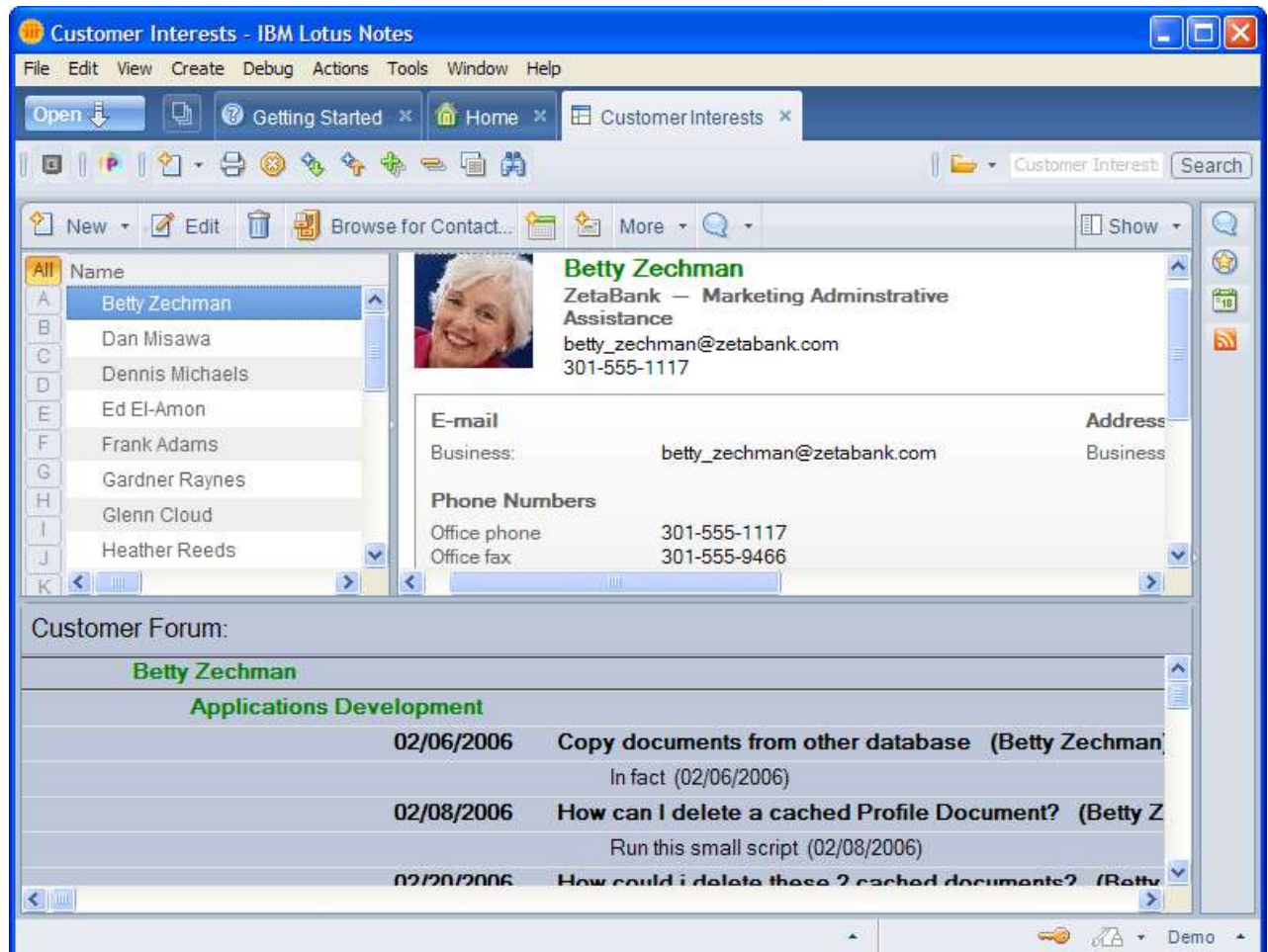


7. Click **OK** to save your wiring changes and close the wiring interface.

8. Click **File > Finished Editing** then select **Yes** to leave the Composite Application Editor feature.

**Result:** This closes the Composite Application Editor and displays the updated composite application in the Lotus Notes 8 Beta client.

**Tip:** Select **Show > Preview on Side** on the top right of the **Notes Contacts View** to change the placement of the preview pane in the Notes Contact component to resemble the following screen capture.



To observe the result of your wiring, select a row in the Notes Contacts View component located at the top left of the composite application. The NSF component from the NDForum application reacts and changes the row selection to the matching user name. The property is published from the Notes Contacts View component with each row selection you perform. The action you wired to this property executes and performs your defined logic, moving the row selection to the matching common name.

You have completed this lesson and used the Composite Application Editor feature to wire two NSF components into a causal relationship by connecting a property to one or more actions. Earlier, you used Domino Designer 8 Beta to specify the properties that would be published and actions that would be executed for the NSF components.

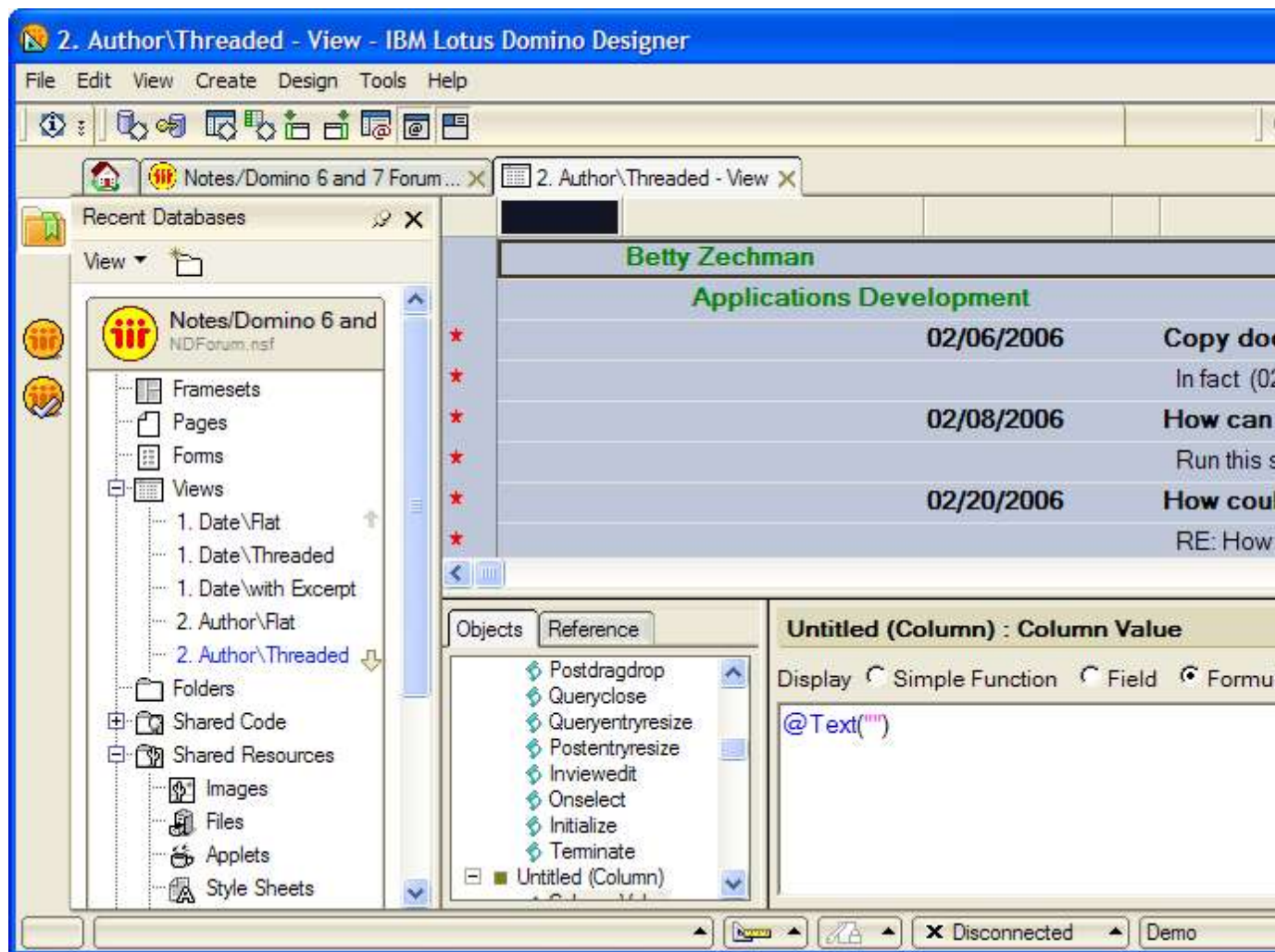
## Lesson 7 – Publishing properties using LotusScript APIs

Earlier, you observed how the view column properties box can be used to publish a property using the column value of the selected row. You can also make use of the new Onselect view event handler to publish more complex properties by using the LotusScript APIs. For this lesson, we will review this more complex code that is already in the NDForum Notes application.

### Procedure 7.1 – Defining addition actions on a NSF component

Complete the following steps to define the action code for a NSF component:

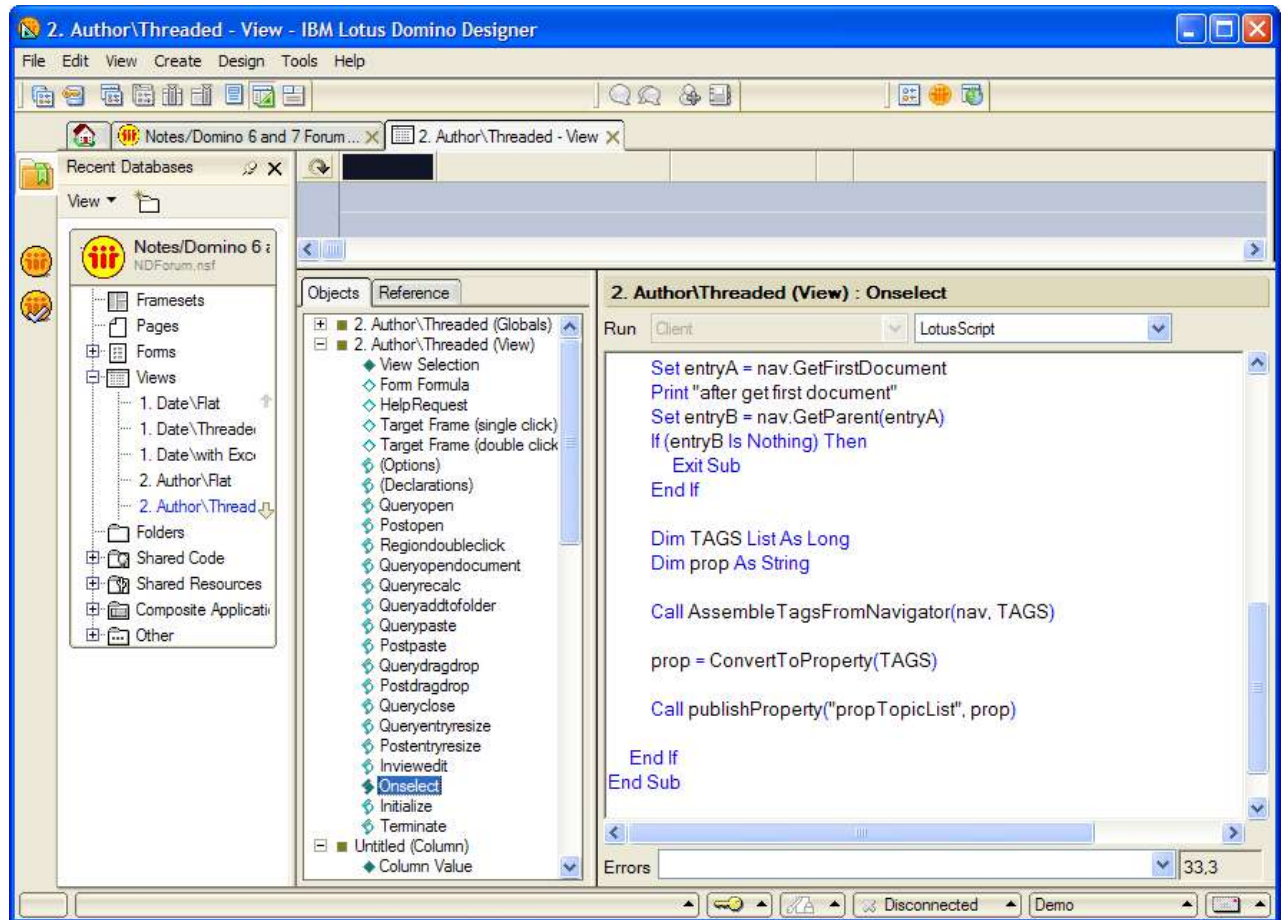
1. If Domino Designer 8 Beta is not already running, start it now and open NDForum.nsf.
2. If needed, switch to the Views list by clicking **Views** in the designer pane on the left.
3. Select the view **2. Author\Threaded** view and press **Enter** to open it in design mode. Your screen should look similar to the following screen capture.





4. Select the **Objects** tab.
5. Locate and expand **2. Author\Threaded (View)**
6. Select the **Onselect** event.
7. Remove the **%REM** statement at the start of the Onselect subroutine and the **%END REM** statement at the end of the OnSelect subroutine. Your screen should then resemble the following screen capture.

The LotusScript statements and subroutines in the Onselect and Globals event handlers determine the subcategories for the currently selected user and tally the number of documents per subcategory. This information is published as a property that can be wired to other components.



**Tip:** The last statement of the Onselect event handler uses the name of the property you defined earlier. If you used a different property name when you created the WSDL for NDForum.NSF, change the property name in the script to match.

8. Click **File > Save** to save your changes to the view.
9. Click **File > Close** to exit design mode.

You have completed this lesson and the property will be published when a top level category is selected in the view. Next, you will add an Eclipse Tag Cloud component to your NSF-based composite application. The Tag Cloud component has actions and properties already defined so you can easily specify causal relationships between it and other components.

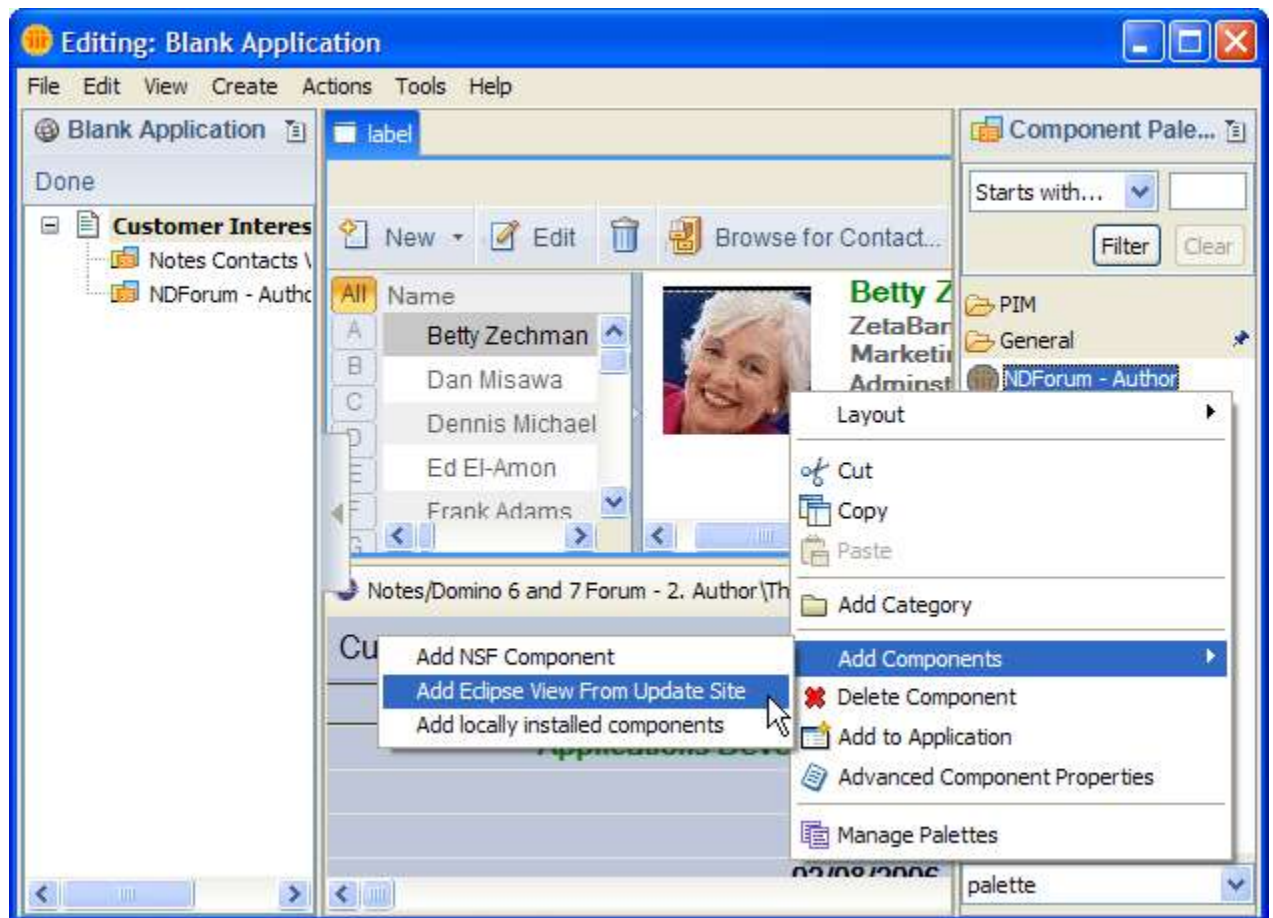
## Lesson 8 - Add Eclipse component to the composite application

Composite applications in the Notes 8 client can include any combination of NSF components and Eclipse components. The components can also be Lotus Component Designer produced components, embedded web browsers, JSR168 or WSRP portlets. In this lesson, you will add an Eclipse component to your composite application that already contains NSF components from Notes Contacts and NDForum.

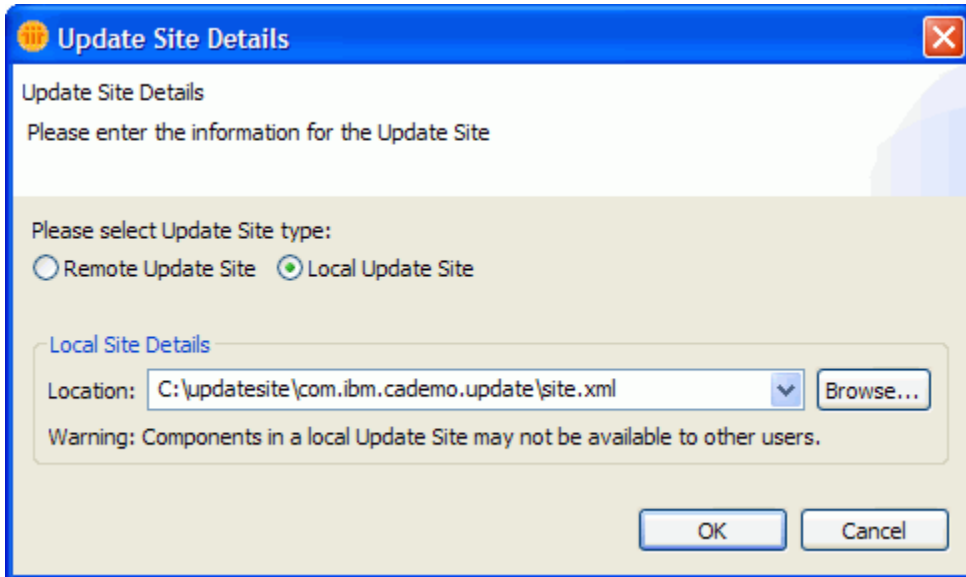
### Procedure 8.1 – Add an Eclipse component to a composite application

Complete the following steps to add an Eclipse component to the component palette:

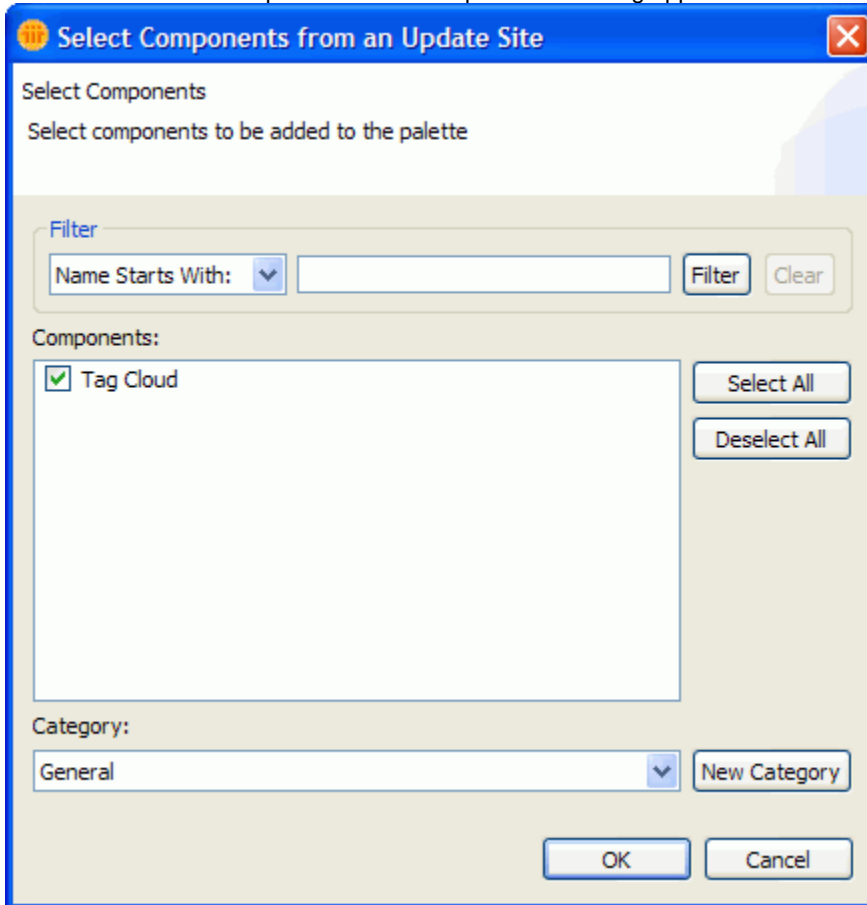
1. If needed, start your Lotus Notes 8 Beta client.
2. If needed, open the NSF-based composite application, **Customer Interests**, that you created earlier.
3. From the menu, select **Actions > Edit Application** to open the Composite Application Editor.
4. If needed, expand the Component Palette on the right side of the Composite Application Editor.
5. In the **Component Palette**, right click, then select **Add Components > Add Eclipse View From Update Site**.



6. In the resulting dialog, select **Local Update Site** and browse to c:\updatesite\com.ibm.cademo.update\site.xml
7. Click **OK**.



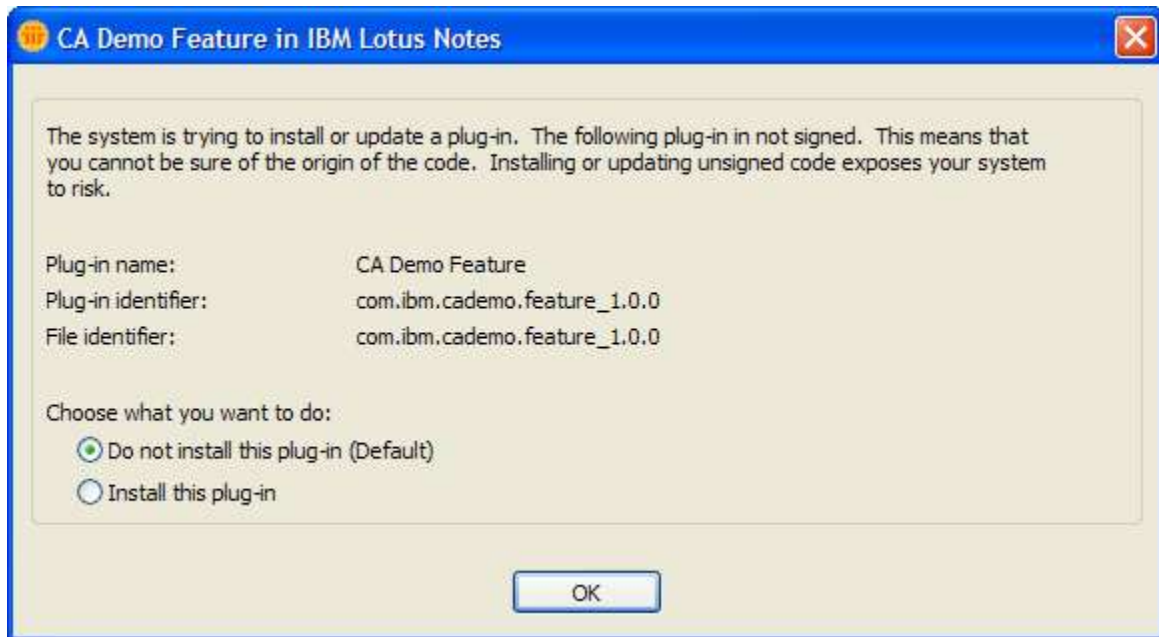
**Result:** The Select Components from an Update Site dialog appears



8. Select **Tag Cloud** and click **OK**.

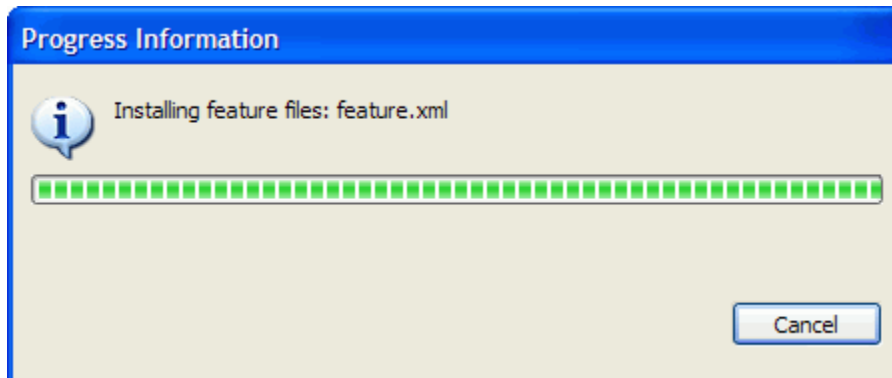


**Result:** The plug-in install dialog appears.

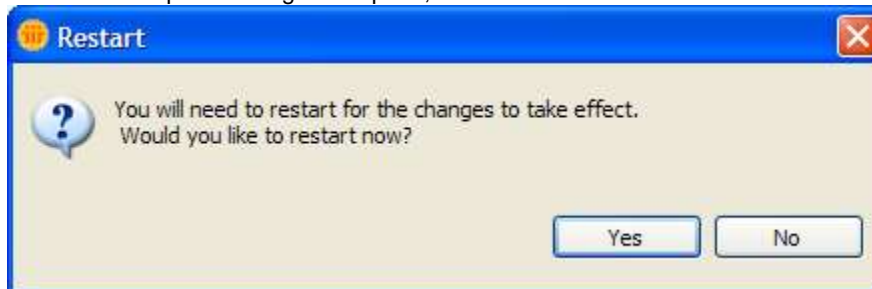


9. Choose the option **Install this plug-in** and select **OK**.

**Result:** A progress dialog displays indicating that the feature is being provisioned.



**Result:** When provisioning is complete,

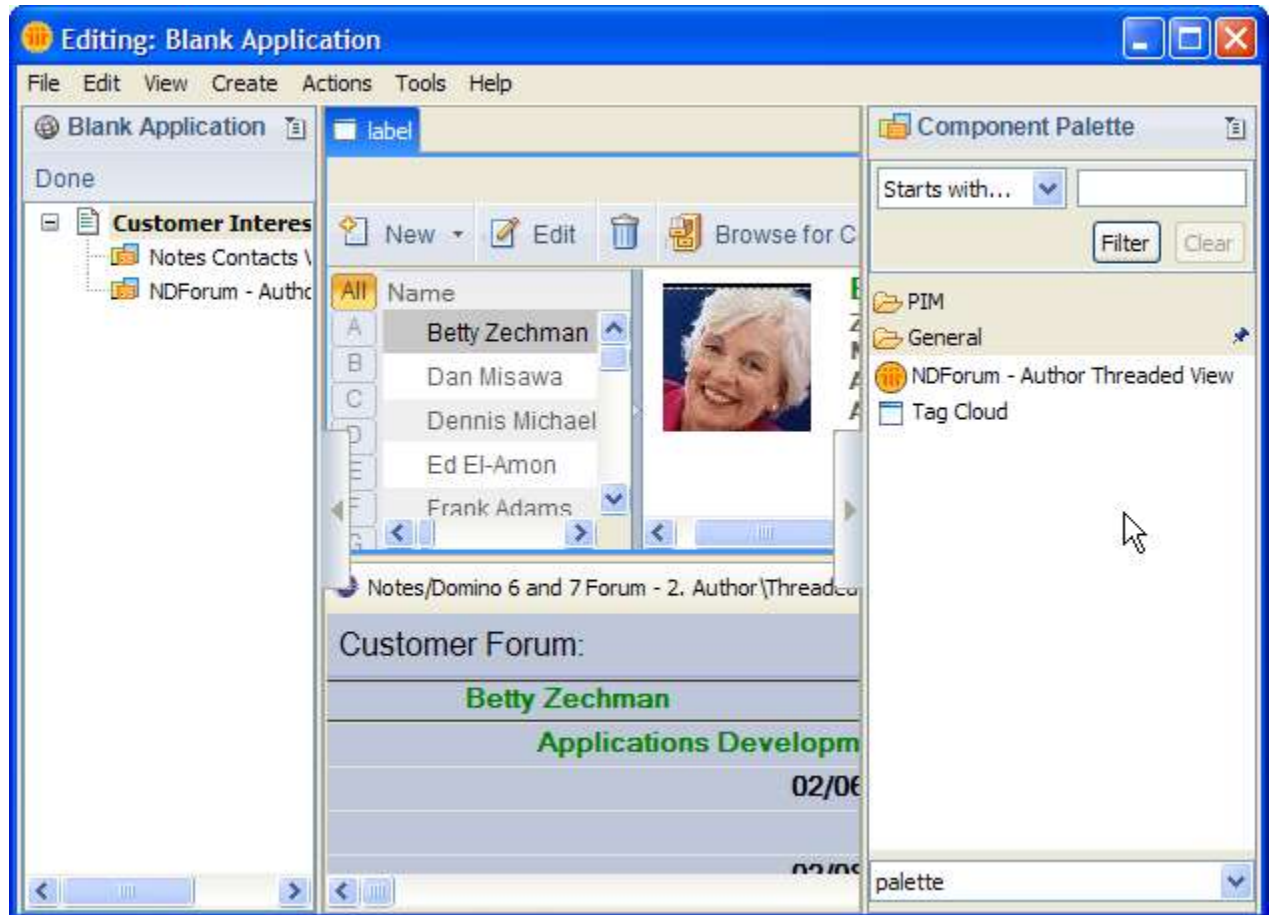


10. Select **No** when prompted to restart

**Note:** If you chose **Yes** to the preceding prompt, the Notes 8 Beta client will close. You can start the Notes 8

Beta client, open your **Customer Interests** composite application, use **Actions > Edit Application** and proceed to step 11.

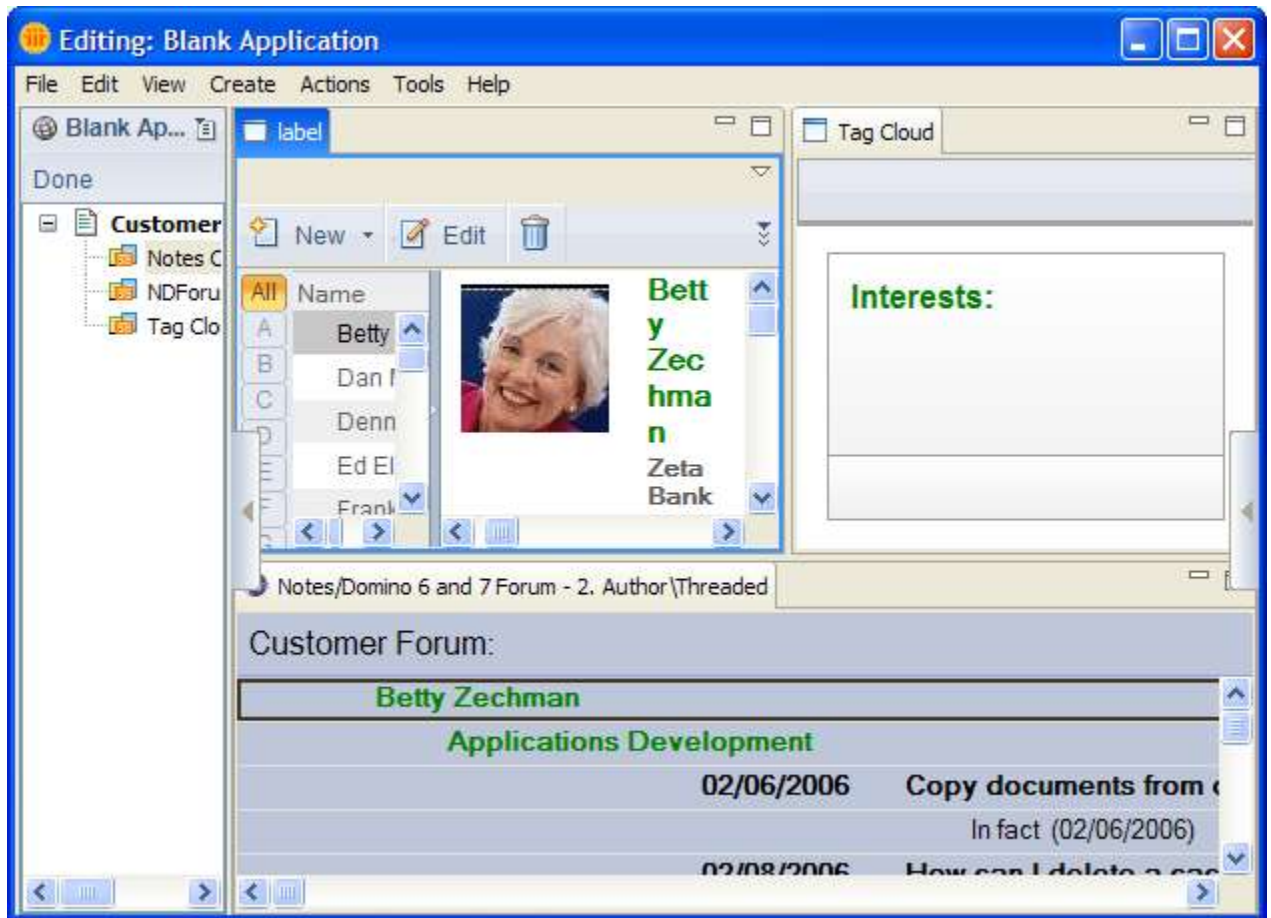
**Result:** The Tag Cloud component will be listed in the Component Palette as shown in the following screen capture.



11. Select the **Tag Cloud** component from the Component Palette and drag it into the upper right of the application.

**Tip:** Hold down the left mouse button as you drag your mouse against the right border of the center pane. When the mouse pointer changes to a right facing arrow, release the mouse button.

**Result:** Your composite application includes the Tag Cloud component that was developed in the Eclipse IDE.



You have completed this lesson and added an Eclipse component to your NSF-based composite application through the Composite Application Editor feature of the Notes 8 Beta client.

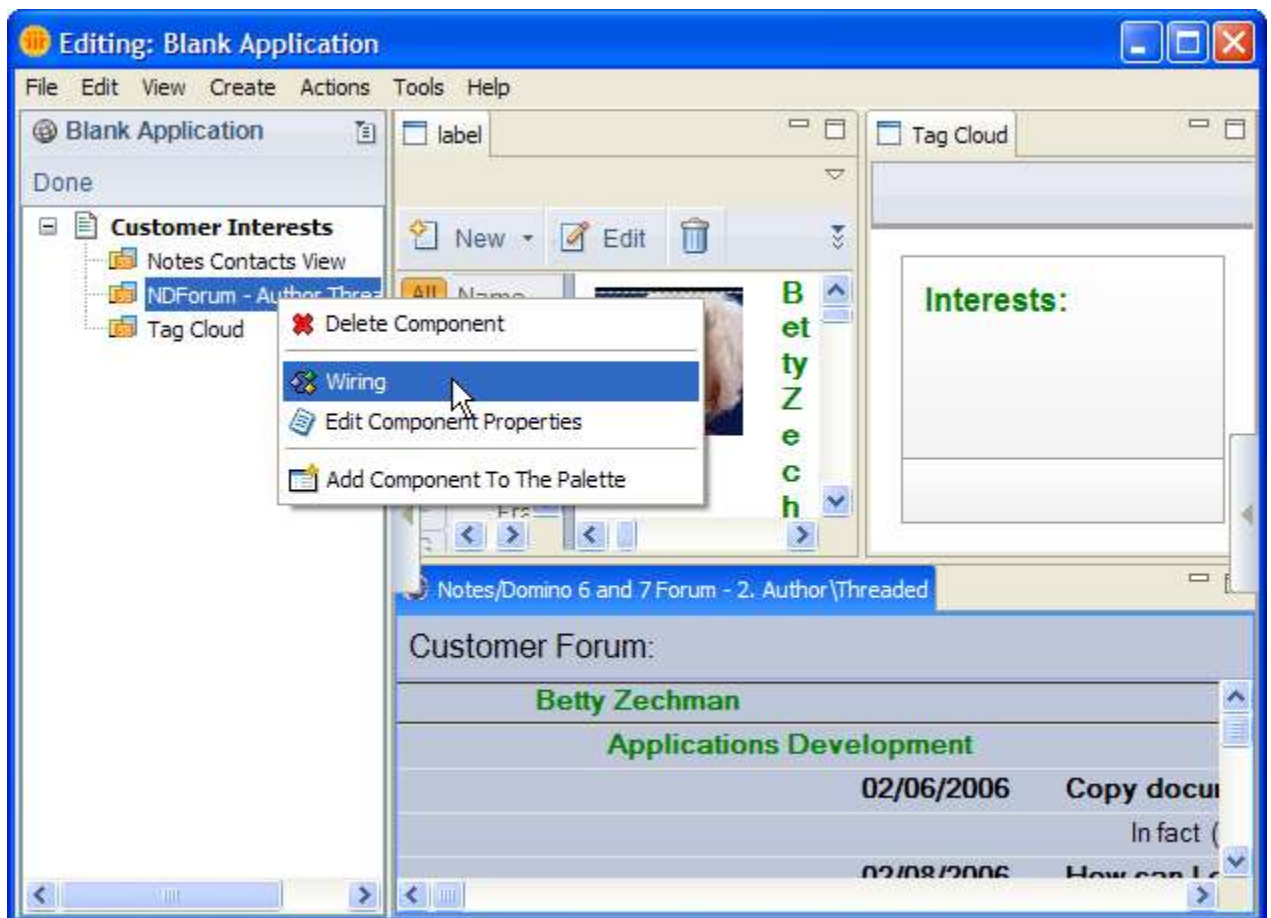
## Lesson 9 - Wire Eclipse and NSF components in a composite application

In earlier lessons, you saw how components can be loosely coupled in causal relationships specific to a composite application. The wiring capabilities of the Composite Application Editor feature can wire components of heterogeneous technologies. The property broker functionality of the Notes 8 Beta client is the runtime environment. In this lesson you will wire the Tag Cloud Eclipse component to a NSF component from the NDForum Notes application.

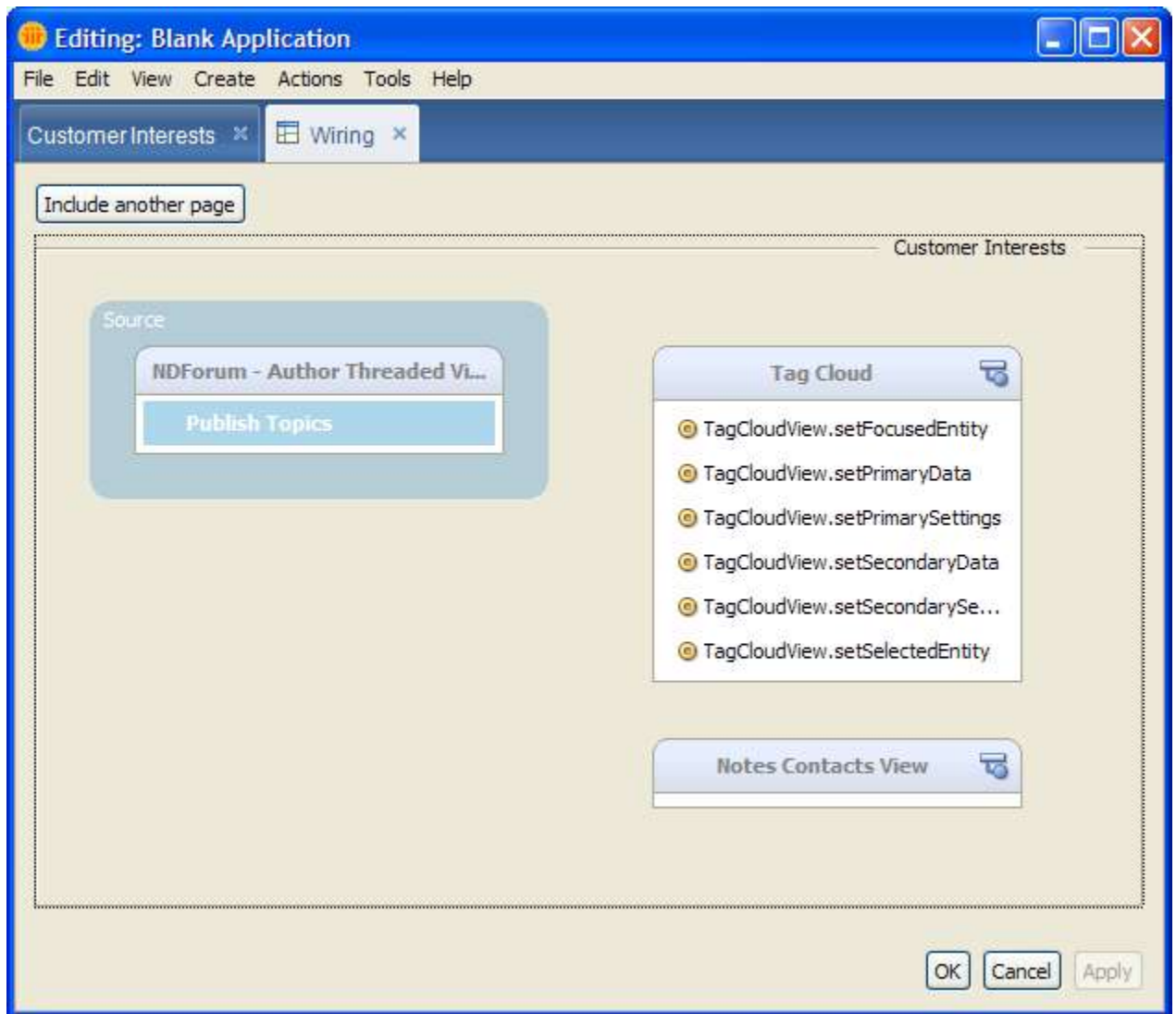
### Procedure 9.1 – Wiring components in the composite application

Complete the following steps to form a causal relationship between NSF and Eclipse components in your composite application:

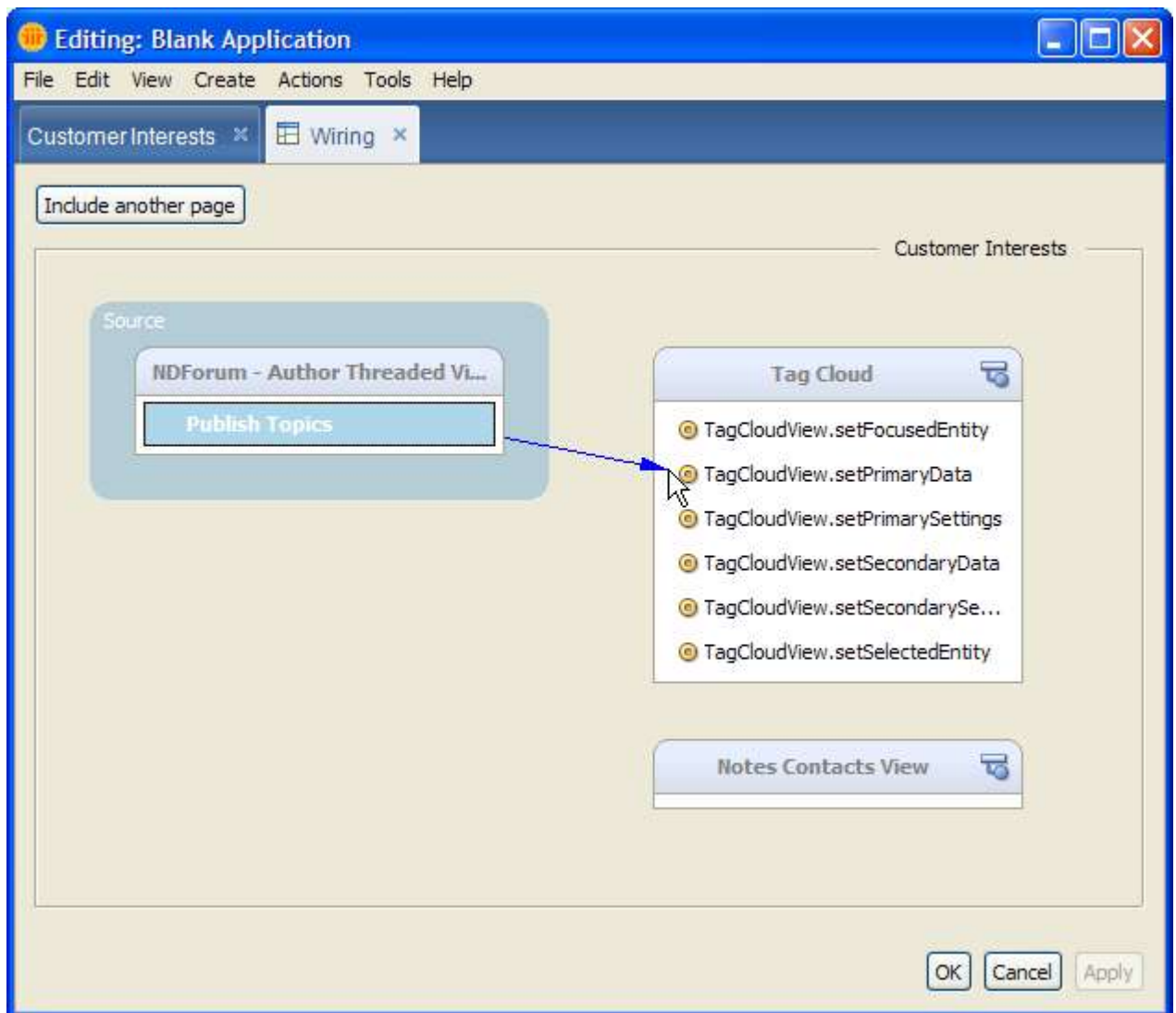
1. If needed, use **Actions > Edit Application** to open CAE for your **Customer Interests** composite application,
2. In the Navigator pane on the left, select the **NDForum - Author Threaded View** component then right click and select **Wiring**.



**Result:** The Wiring Pane appears with NDForum as the source component that lists its single property. The Notes Contacts View component has no actions and the Tag Cloud has six actions with the same namespace and datatype as the single property of the NDForum component.

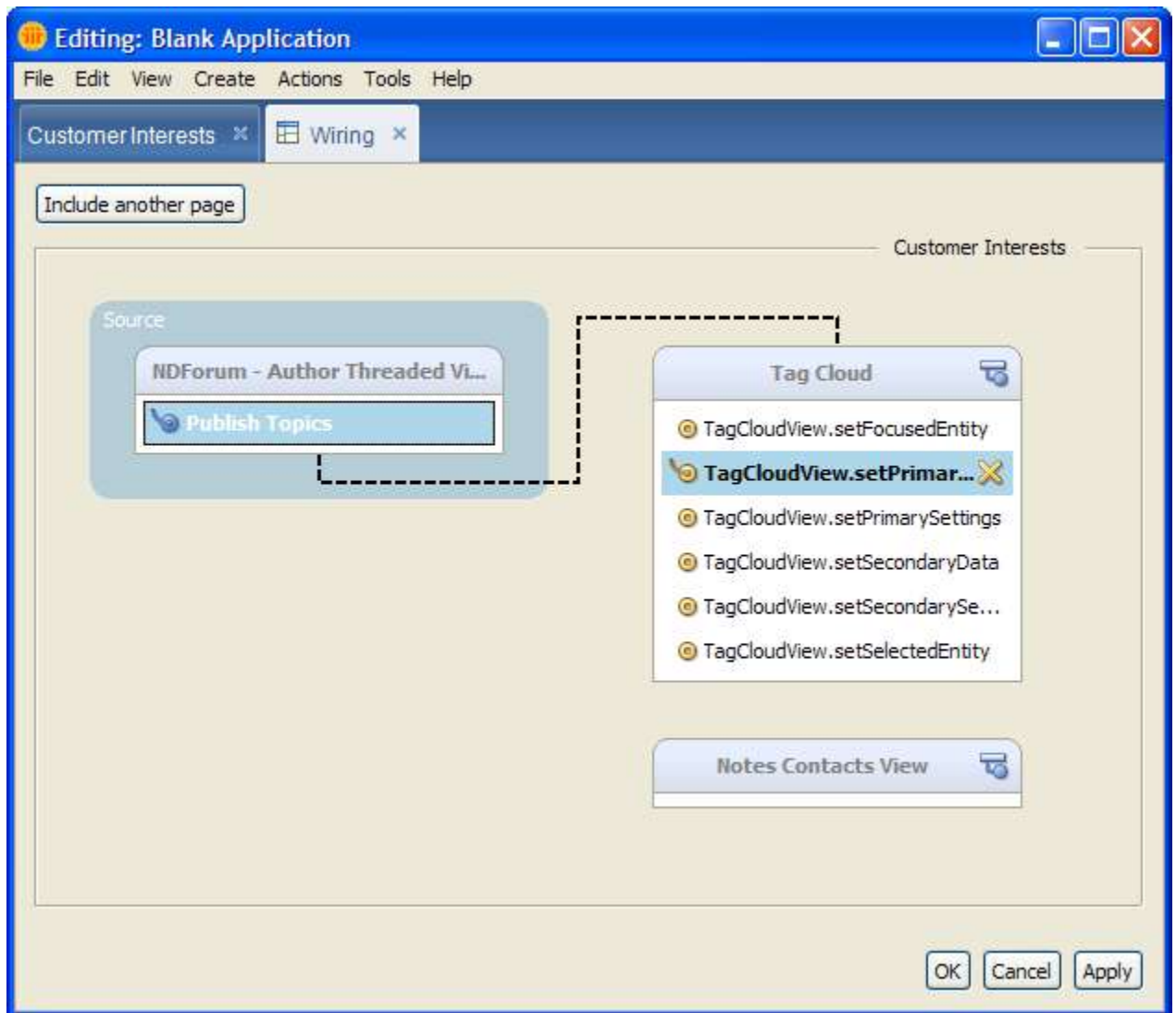


3. Wire the **Publish Topics** property of the ND Forum – Author Threaded View component to the **setPrimaryData** action of the Tag Cloud component.

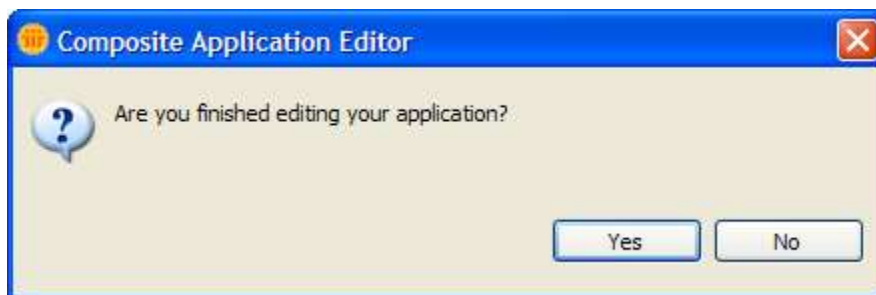


**Result:** The wire between the property and action is displayed.



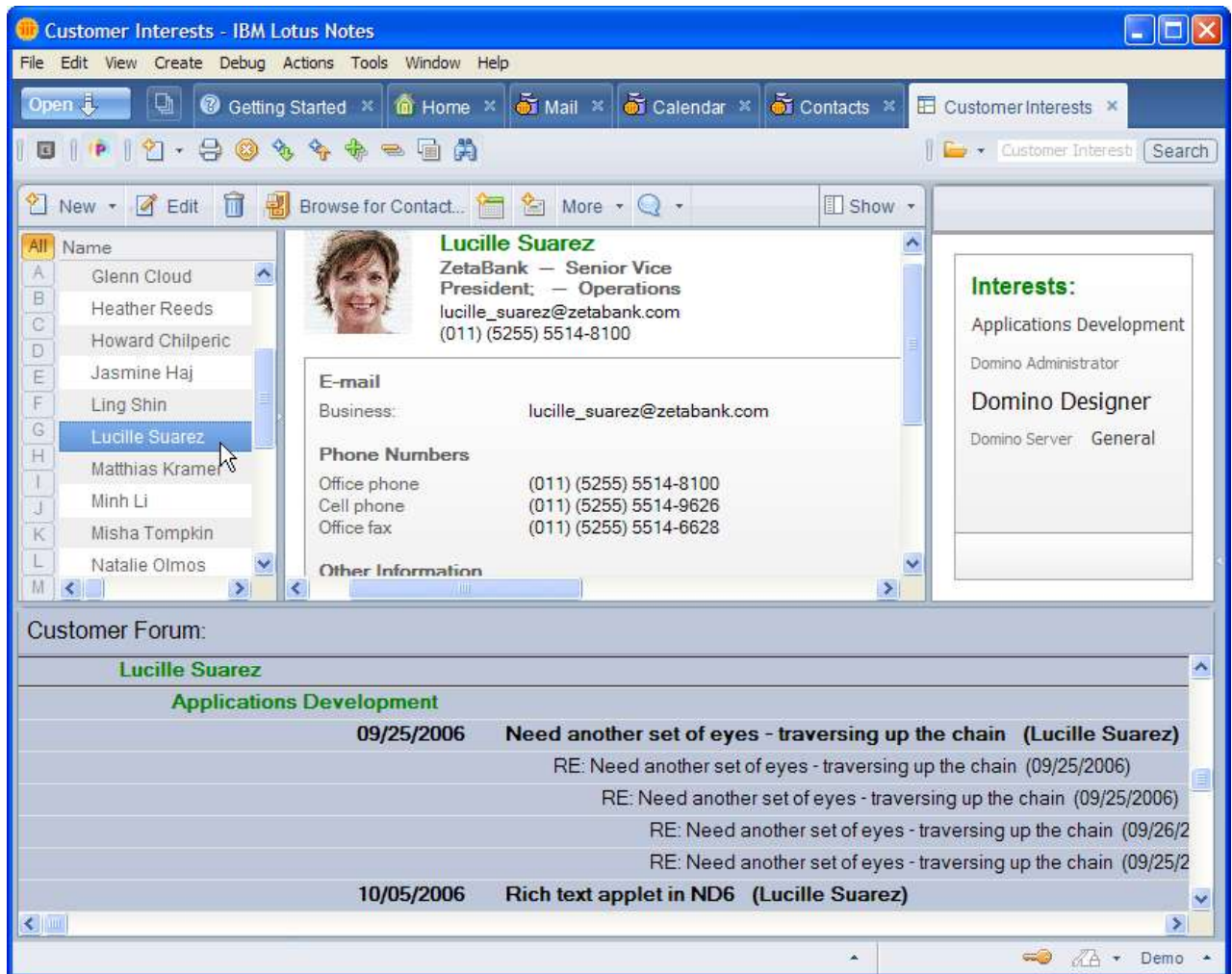


4. Click **OK** to confirm your wiring changes. The wiring pane will close.
5. Select **File > Finish Editing** to confirm the changes to your composite application
6. Answer **Yes** to the confirmation dialog.



**Result:** After closing the Composite Application Editor, your updated composite application will display.

**Tip:** You can resize the components within the composite application to your liking.



To see the new causal relationships created through your wiring changes, select a document in the Notes Contacts View component in the top left. The property is published from this component and the selection in the NDForum component changes. The NDForum component also publishes a property triggered by the row selection change where the newly selected row is a top level category in the Notes view. You wired this property to the Tag Cloud and its action displays a list of subcategories for the selected user based on the content of the NDForum component.

You have completed this lesson and added a wire between the NSF component in the NDForum Notes application and the Eclipse component in the Tag Cloud application through the Composite Application Editor feature of the Notes 8 Beta client. This is in addition to the wire you defined earlier between the NSF component in the Profiles Notes application and the NSF component in the NDForum Notes application.



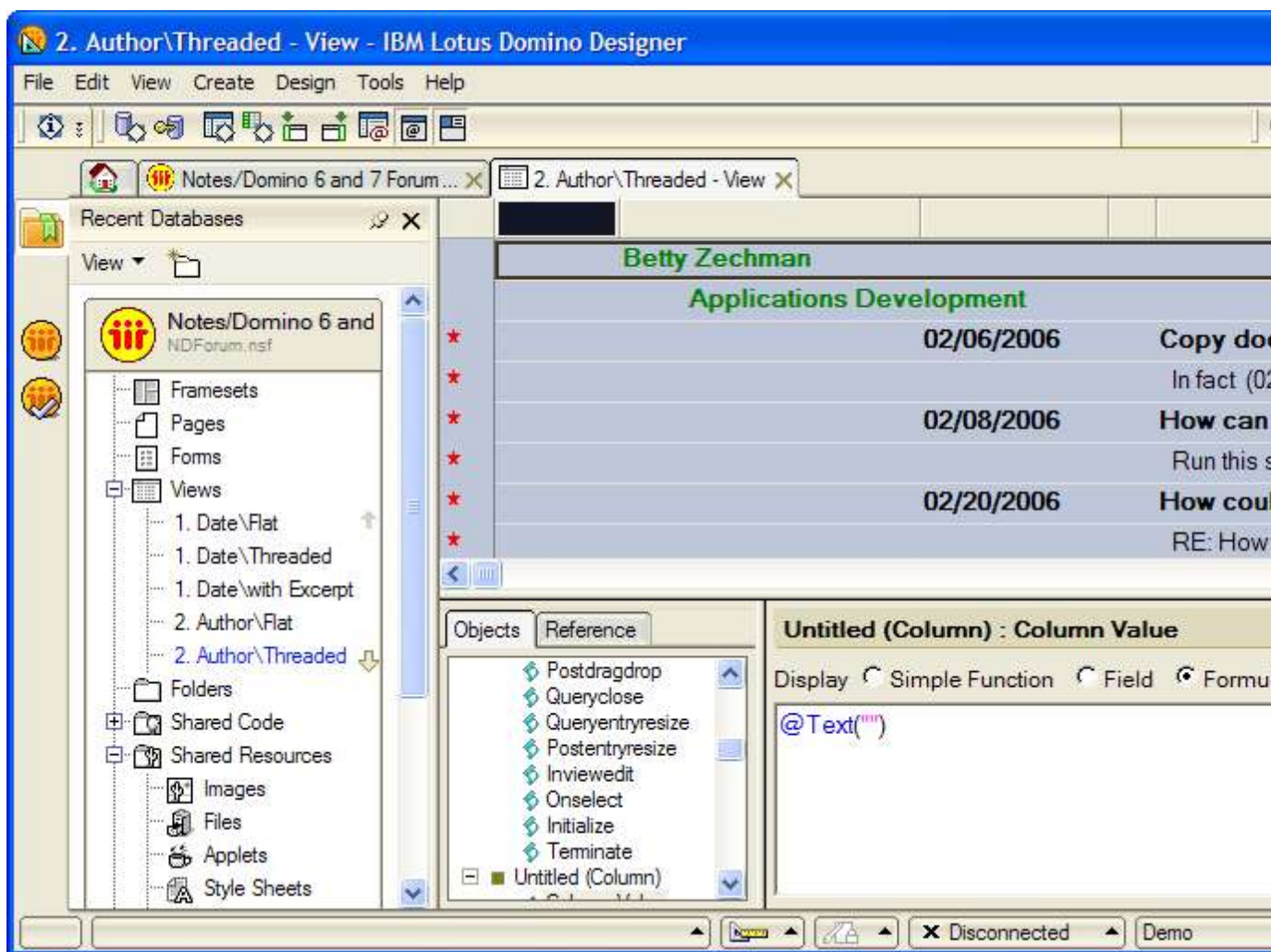
## Lesson 10 - Additional actions and wiring

The Tag Cloud component can also publish information as it has properties defined. The application user can select a displayed topic from the tag cloud and this can be published as a property. The NSF component from the NDForum application can change the view selection to show the subcategory that matches the selected topic for the previously selected user. In this lesson, you will add an additional action to the NSF component to change the view selection based on the tag selected in the Tag Cloud component.

### Procedure 10.1 – Defining actions on a Notes view

Complete the following steps to define the logic for a NSF component action:

1. If Domino Designer 8 Beta is not already running, start it now and open NDForum.nsf.
2. If needed, switch to the Views list by clicking **Views** in the designer pane on the left.
3. Select the view **2. Author\Threaded** view and press **Enter** to open it in design mode. Your screen should look similar to the following screen capture.



4. Click **Create > Action > Action...** to add a new action to this view.  
**Result:** The action pane opens in the right top and the properties box appears for the new action.
5. On the first tab of the properties box, type "SelectSubCategory" in the **Name** field.

**Action**

Name: SelectSubCategory

Label:

Target Frame:

Type: Button

Value:

Display

- ☐ Include action in Action bar
  - ☐ Only show icon in Action bar
  - ☐ Right align action control
- ☐ Include action in Action menu
- ☐ Include in right mouse button menu

Icon

- ☒ None
- ☐ Notes
- ☐ Custom

6. Deselect the options **Include action in Action bar** and **Include action in Action menu**.
7. On the last tab of the properties box, click **actSelectTopic** in the **Composite Settings Action Name** field list at the bottom.

**Action**

☐ Publish Action with OLE object

When action is chosen

- ☐ Close OLE object and return to Notes
- ☒ OLE object remains open
- ☐ Bring document window to front

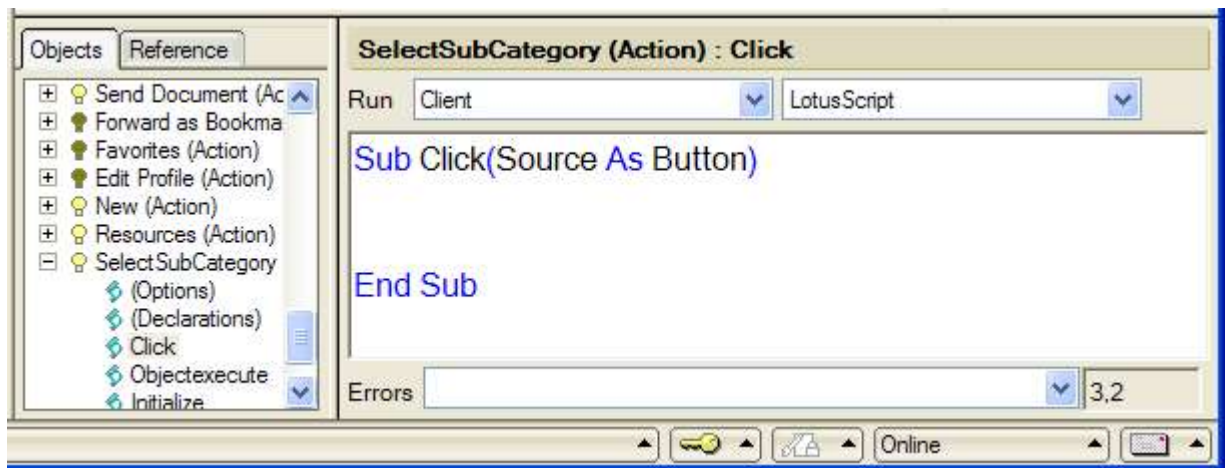
NotesFlow Publishing

Composite Settings

Action Name: actSelectTopic

8. Close the properties box.

9. Click the newly created action **SelectSubCategory** in the action pane on the top right.
10. In the Domino Designer programmers pane at the bottom right, click **Client** and **LotusScript** for this action. The Client option may already be selected.



11. Input the following LotusScript code in the Click subroutine. The source code has been provided in the file `SelectSubCategory_Action_for_NDForum.txt` You can paste this code into the Click event.

**Note:** The action script will create an object of the new NotesPropertyBroker class and use the property published by the other component to change the view selection to the subcategory for the selected user.

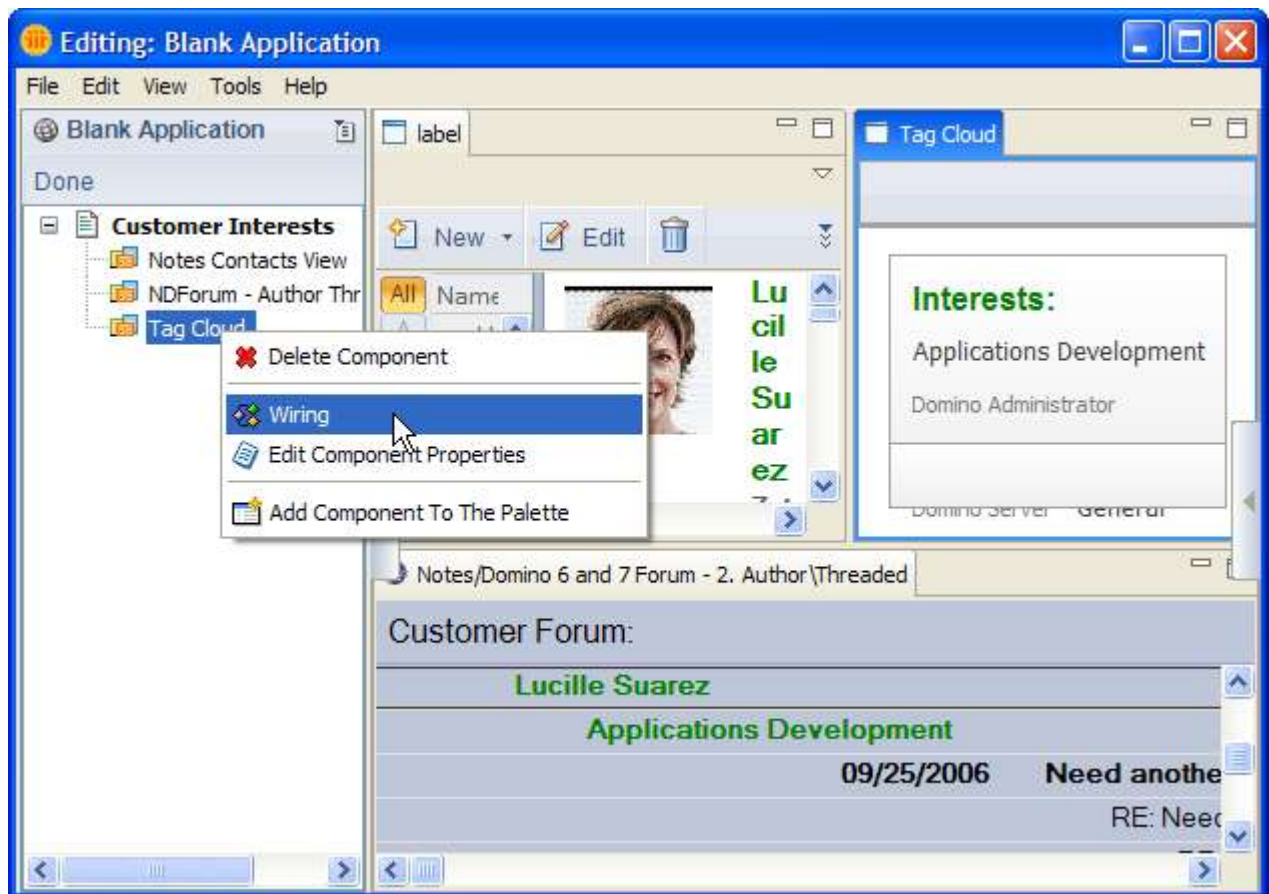
```
Sub Click(Source As Button)
Dim ws As New notesuiworkspace, s As New notessession
Dim db As notesdatabase, uidb As NotesUIDatabase
Dim author As String, category As String
Set db = s.currentdatabase
Set uidb = ws.CurrentDatabase
Set doc = db.GetProfileDocument("SelectedAuthor", s.UserName)
author = doc.GetItemValue("SelectedAuthor")(0)
'Handle error 4719, when the script is run without a property broker context
On Error 4719 Goto err4719
Dim pb As NotesPropertyBroker 'New PropertyBroker code
Set pb = s.GetPropertyBroker
Dim pbInputProperty As NotesProperty, pbContext As Variant
pbContext = pb.InputPropertyContext 'array of NotesProperty objects
Set pbInputProperty = pbContext(0) 'handle to first NotesProperty object
Dim InputPropertyName As String, Namespace As String
InputPropertyName = pbInputProperty.Name 'name of NotesProperty object
Namespace = pbInputProperty.Namespace 'namespace of NotesProperty object
Dim pbValue As Variant
pbValue = pb.getPropertyValue(InputPropertyName) 'array values of NotesProperty
category = pbValue(0) 'first value from array NotesProperty object
SkipPropertyBroker:
If category = "" Then Exit Sub
Call uidb.OpenView("2. Author\Threaded", author + "\" + category, False, False)
Exit Sub
err4719:
Print "Error " & Str(Err) & ": " & Error$
Resume skipPropertyBroker
End Sub
```

12. Click **File > Save** to save your view changes.
13. Click **File > Close** to exit design mode.

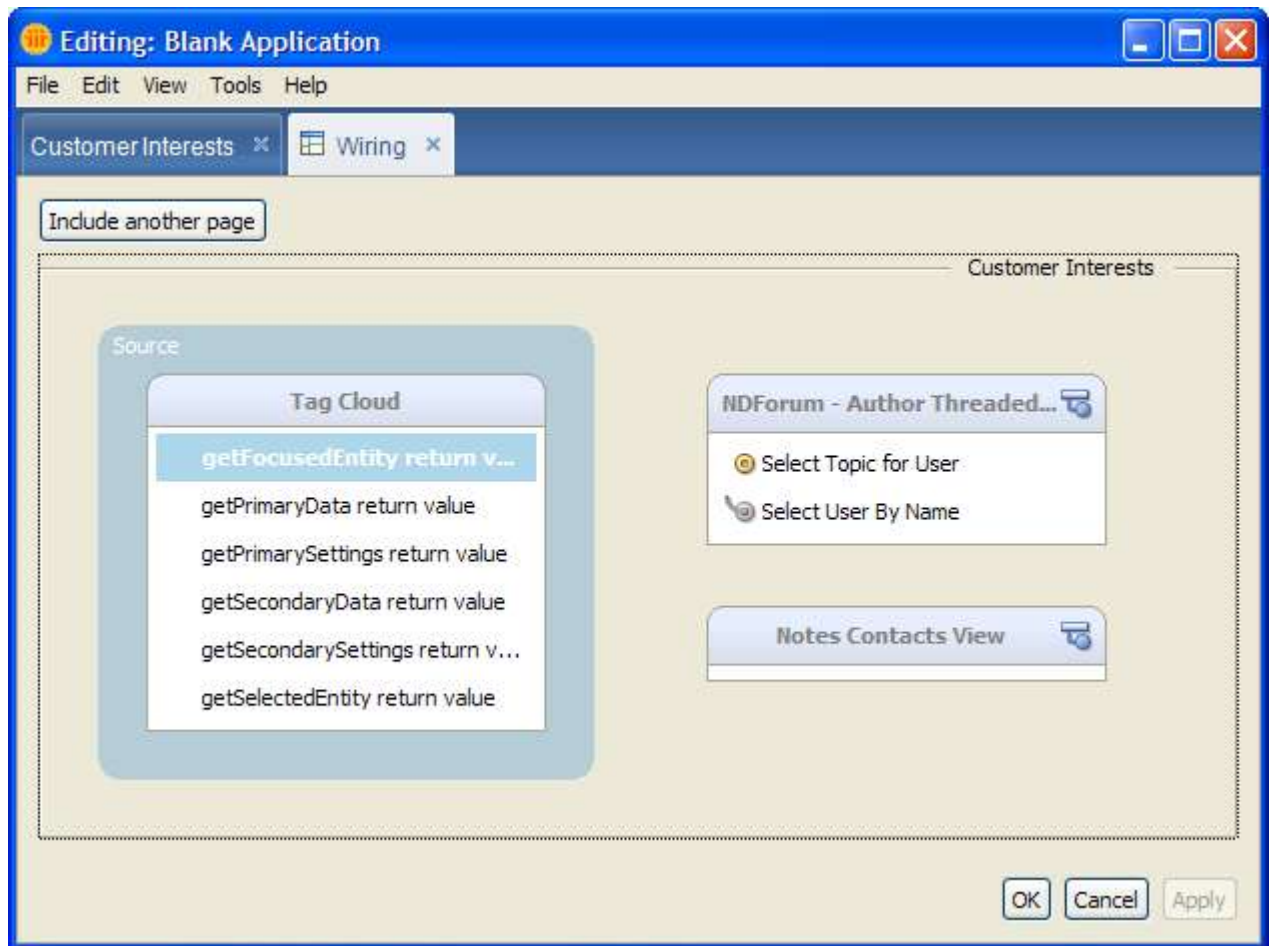
### Procedure 10.2 – Wiring the properties and actions

Complete the following steps to create a causal relationship between the Tag Cloud and NDForum components:

1. If needed, open your Lotus Notes 8 Beta client.
2. Use **File > Open > Lotus Notes Application** to open your Customer Interests composite application.
3. Use **Actions > Edit Application** to open CAE.
4. Select the Tag Cloud component from the navigator pane on the left, then right-click then select **Wiring**.

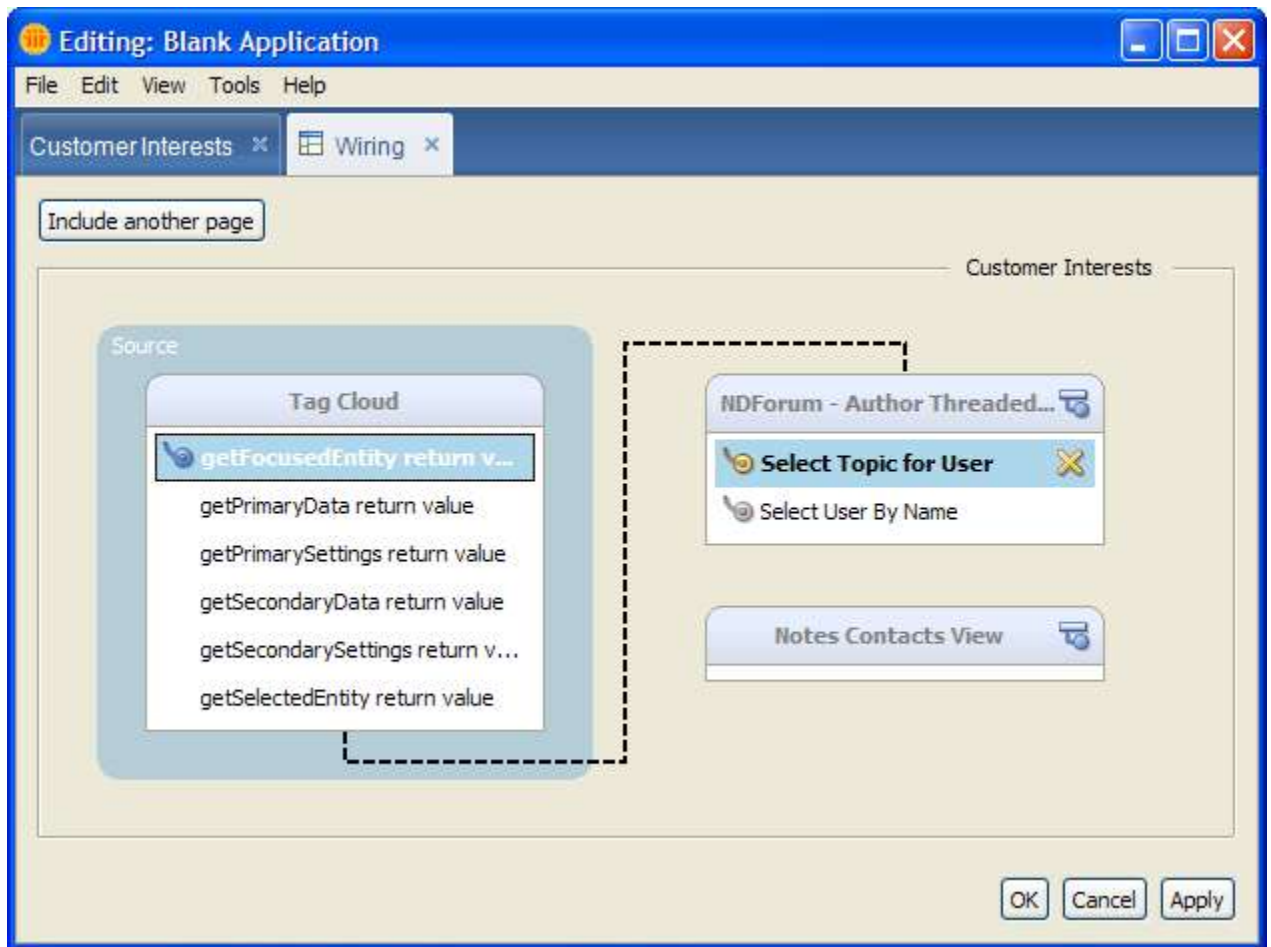


**Result:** The wiring pane appears.

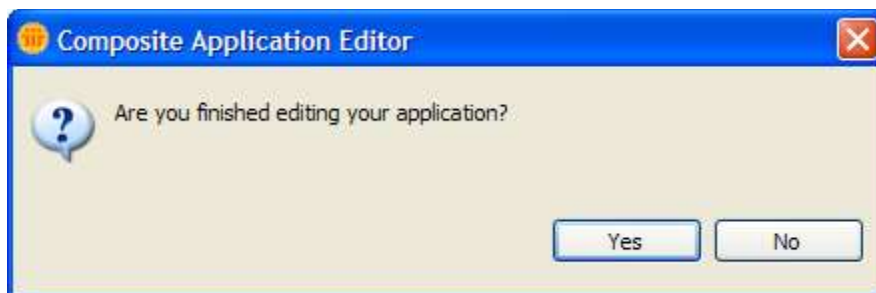




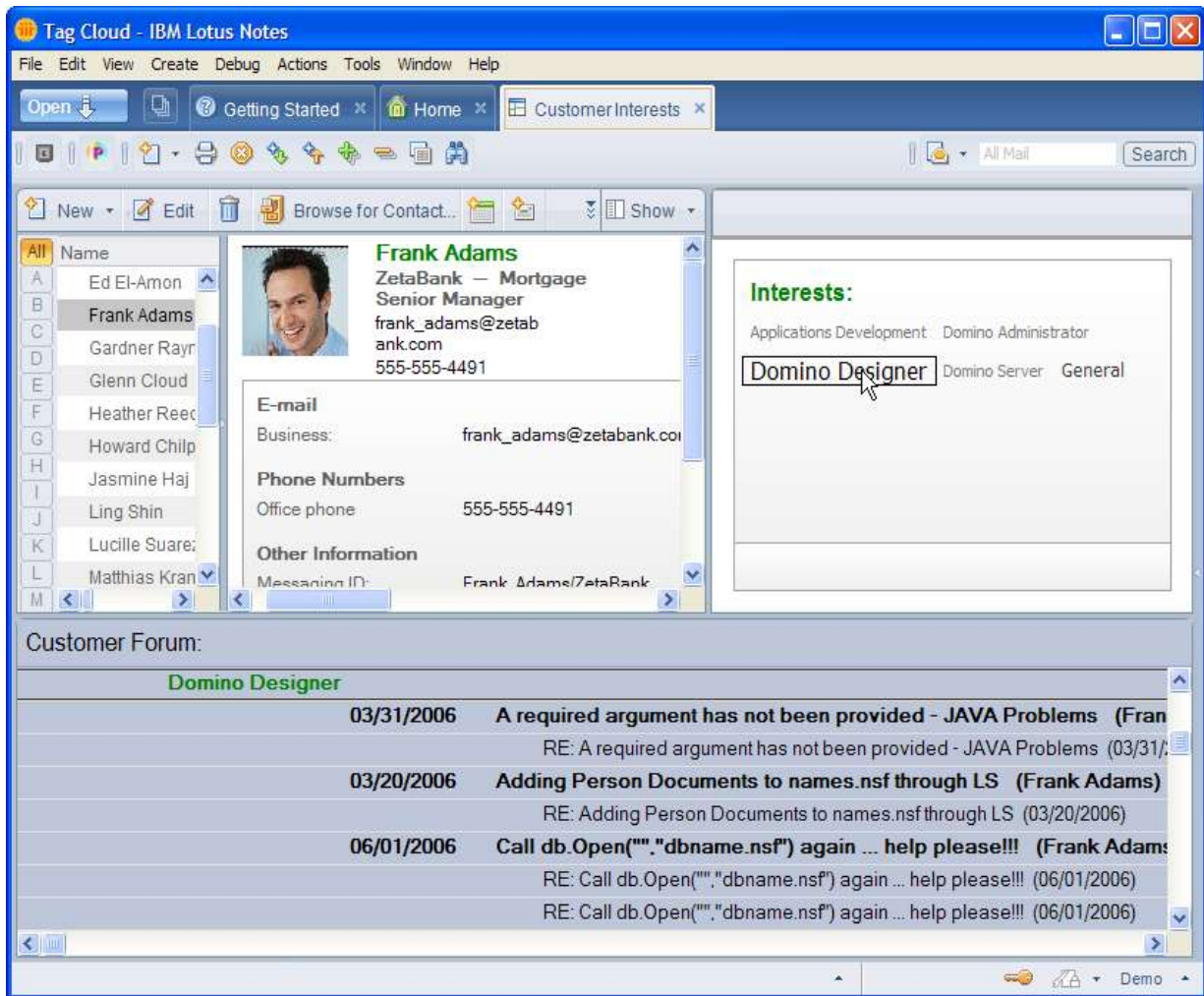
7. Create a wire from the **getFocusedEntity** property of the Tag Cloud Eclipse component to the **Select Topic for User** action of the NDForum NSF component as shown in the screen capture below.



8. Click **OK** to confirm your wiring changes. The wiring pane will close.
9. Select **File > Finish Editing** to confirm the changes to your composite application.
10. Answer **Yes** to the confirmation dialog.



**Result:** After closing the Composite Application Editor, your updated composite application will display.



To see the result of your updated wiring, first select a row from the Notes Contacts component, then select a topic from the Tag Cloud component. The NSF component from the NDForum application will perform its defined logic and the row selection will move to the subcategory that matches the topic selected for the previously selected user.

You have completed this lesson and added an additional action to the Notes view that is used as an NSF component. You also wired this action to a property published by the Eclipse component for inter-component communication.

You have completed all lessons in this tutorial and seen how composite applications and inter-component communication works across application and system boundaries to provide a single context of collaboration to improve user and organizational effectiveness.

You have used the Composite Application Editor feature of the Notes 8 Beta client with an NSF-based composite application that contains reusable NSF components and Eclipse components that are loosely coupled in a causal relationship through wiring. You have also used Domino Designer 8 Beta to extend existing NSF design elements to provide properties and actions that support inter-component communication.