Technical Note

ClientAce® WPF Project Example

1. Introduction

Traditional Windows forms are being replaced by Windows Presentation Foundation[1] (WPF) forms. WPF forms are fundamentally different and designed using XAML code. Conventional controls designed for a Windows forms cannot be used in a WPF form without being hosted through an interop. The controls of the ClientAce group in the Visual Studio Toolbox are grayed out, but this document provides the steps to create a project with the ClientAce API to get data to a form.

1.1 Required Software

The following are required to duplicate the process outlined in this document.

- Visual Studio 2008 SP1, 2010, 2012, and 2013
- ClientAce V3.5 or 4.0
- .Net 3.5 SP1 or 4.x

2. Creating the WPF Project

This is one simple example of how to create a WPF OPC with ClientAce; it is not intended to teach anyone how to write code.

2.1 Create the WPF Application

1. In VS 2008, create a new project of type: VB Windows WPF application.
2. By default, the project is created with a window and a grid panel. From the Common Controls, select a text box and place it in the grid panel.

[1] The Windows Foundation Classes were introduced in VS 2008.

2.2 Add the ClientAce References

3. Open the properties for the project by right-clicking on the project in the Solution Explorer.
4. Click on the **References** tab.
5. Click **Add...** to create a new .Net reference.
6. In the Add Reference dialog box, scroll through the .Net components to find the Kepware.ClientAce 3.5.0.3 component.
7. Select it and click **OK**.

2.3 Adding Code to the Project

8. To declare global variables at the form level, click on the **xaml.vb** tab.
9. Enter the declarations from EX1 below.

```
' Create an OPC DA Server Management object for each server to connect.
' This only connects to one server at a time, so create one and make it global
' Use the "WithEvents" modifier to receive data changed notifications for
' subscribed items.
Dim WithEvents daServerMgt As New Kepware.ClientAce.OpcDaClient.DaServerMgt

' When creating a subscription, the server provides a handle that
' must be used when referencing that subscription in the server.
Dim activeClientSubscriptionHandle As Integer

' Where the server subscription handle refers to a subscription within the
' server, specify a client subscription handle to refer to that subscription.
Dim activeServerSubscriptionHandle As Integer
```

EX 1: Project Declarations

10. Add code to initialize parameters for connecting to the server as shown in EX2 below. This code is executed when the form's loaded event is called (i.e. Private Sub Window1_Loaded()).

```
' The URL describes the type of server connection, the server program ID,
' and optional ClsID.
Dim url As String = "opcda:///Kepware.KEPServerEX.V5"
' The client handle is a unique identifier for each client connection.
Dim clientHandle As Integer = 1

' The connectInfo structure defines a number of connection parameters.
Dim connectInfo As New Kepware.ClientAce.OpcDaClient.ConnectInfo

' The LocalID member specifies supported language options. Use "en" for English.
connectInfo.LocalId = "en"
```



```
' The KeepAliveTime member is the time interval or rate, in ms, at which
' the connection to the server is checked by the API.
connectInfo.KeepAliveTime = 1000

' The RetryAfterConnectionError instructs the API to automatically
' attempt to reconnect after a connection loss. Set to True:
connectInfo.RetryAfterConnectionError = True

' The RetryInitialConnection instructs the API to continue to try to
' establish an initial connection. This is best if it is known that
' the server is really present and will likely allow a connection.
' If not, it creates a huge delay. Set to False:
connectInfo.RetryInitialConnection = False

' The connectFailed is set by the API. Check this value after the Connect
' method. Initialize to False in case Connect throws an exception before
' it can be set.
Dim connectFailed As Boolean = False
```

EX 2: Initialization of Connection Parameters

11. Once the connection parameters are initialized, the connection to the server can be attempted.
12. Enter the code in EX3 below into the form's loaded event function.

Note: After the connection attempt, call the Subscribe_Data() function to request the item(s) to poll. Name this function as appropriate for the application or item.

```
' Call the Connect API method:
Try
   daServerMgt.Connect(url, clientHandle, connectInfo, connectFailed)
   Call Subscribe_Data()
Catch ex As Exception
   MsgBox("Handled Connect exception. Reason: " & ex.Message)

   ' Verify the following code detects connection failed:
   connectFailed = True
End Try

' Handle result:
If connectFailed Then
   ' Tell user connection attempt failed:
   MsgBox("Connect failed")
End If
```

EX 3: Server Management Object Connection Method

13. Create the Subscribe_Data function by which to subscribe or request the items to be polled. To do this, create an array of object identifiers. This array is passed to the server in the subscription method of the Server Management object created for connecting to the server.
14. Enter the code in EX4 to initialize the parameters of the item identifier and the subscription parameters.

```
Dim clientSubscriptionHandle As Integer = 1

' The active parameter is used to indicate to the server data is required
' for the subscribed items now or not. The active state can be changed
' later with a call to SubscriptionModify.
Dim active As Boolean = True

' The updateRate parameter is used to indicate to the server the speed of
' data updates. This is a REQUESTED rate in milliseconds. The server may
' not be able to honor this request.
Dim updateRate As Integer = 1000

' The deadband parameter specifies the minimum deviation to be considered
' a change of value. It is expressed as a percentage (0 - 100).
Dim deadBand As Single = 0

' The itemIdentifiers array describes the items to enroll in this subscription.
' Each member of the ItemIdentifier structure is described below.
' Create only 1 for this example.
Dim itemIdentifiers(0) As Kepware.ClientAce.OpcDaClient.ItemIdentifier

itemIdentifiers(0) = New Kepware.ClientAce.OpcDaClient.ItemIdentifier

' The itemName parameter is the identity of the data object in the server.
itemIdentifiers(0).ItemName = "Channel_1.Device_1.R0"

' The ClientHandle is used by the server to reference items in data
' changed events. This handle should uniquely identify each item.
itemIdentifiers(0).ClientHandle = 0

' Request a specific data type if desired.
itemIdentifiers(0).DataType = Nothing
' The revisedUpdateRate parameter is the actual update rate of the server.
Dim revisedUpdateRate As Integer
```

EX 4: Server Data Subscription Initialization

15. Add the code for the Subscription as shown in EX5 below.

```
    ' Call the Subscribe API method:
    Try
       daServerMgt.Subscribe(clientSubscriptionHandle, active, updateRate,
revisedUpdateRate, deadBand, itemIdentifiers, activeServerSubscriptionHandle)

       ' Save the active client subscription handle for use in
       ' DataChanged events:
       activeClientSubscriptionHandle = clientSubscriptionHandle

       ' Check item result ID:
         If itemIdentifiers(0).ResultID.Succeeded = False Then

         ' Show a message box if an item could not be added to subscription.
         MsgBox("Failed to add item" & itemIdentifiers(0).ItemName & " to
subscription")
      End If

    Catch ex As Exception
      MsgBox("Handled Subscribe exception. Reason: " & ex.Message)
    End Try
```

EX 5: Subscription Method

16. To handle the Data Change event sent to this application every time the value of the subscribed item changes in the sever, enter the code in EX6.

```
    Private Sub daServerMgt_DataChanged(ByVal clientSubscription As Integer, ByVal
allQualitiesGood As Boolean, ByVal noErrors As Boolean, ByVal itemValues() As
Kepware.ClientAce.OpcDaClient.ItemValueCallback) Handles daServerMgt.DataChanged
        ' Loop over values returned. Do not assume that data for all items enrolled
        ' in a subscription are included in every data changed event.
        ' The number of values likely varies each time.
        Dim itemValue As Kepware.ClientAce.OpcDaClient.ItemValueCallback

        For Each itemValue In itemValues
          ' Get the item handle. Use the Items Client handle to update the specific
          ' control associated. For larger projects, create a control array and
          ' associate the client handle with the control array index.

          Select Case itemValue.ClientHandle
            ' Update value control (could be NULL if quality goes bad):
            Case 0
              If IsNothing(itemValue.Value) Then
                TextBox1.Text = "Unknown"
              Else
```



```
                TextBox1.Text = itemValue.Value.ToString()
            End If
          End Select
        Next
    End Sub
```

EX 6: Data Change Event

17. It is necessary to handle closing the form and disconnecting from the server, which are accomplished with two separate functions, as seen in EX 7 below. The first function is a Window or Form Closed event triggered when the form closes normally. This uses a function called DisconnectOPCServer(), which tests the connection to the server and, in connected, executes the Disconnect method of the Server Management Object.

```
    Private Sub Window1_Closed(ByVal sender As Object, ByVal e As
System.Windows.RoutedEventArgs) Handles Me.Unloaded
        'On unload of the form, call the disconnection process for the server.
        DisconnectOPCServer()

    End Sub
    Private Sub DisconnectOPCServer()
        ' Call Disconnect API method:
        Try
            If daServerMgt.IsConnected Then
                daServerMgt.Disconnect()
            End If

        Catch ex As Exception
            MsgBox("Handled Disconnect exception. Reason: " & ex.Message)
        End Try
    End Sub
```

EX 7: Form Unload and Server Disconnection

18. Run the project and verify data from the server is displayed in the WPF form.

3. Summary

Completing the steps described above should provide a foundation for creating a WPF OPC project of greater complexity.

