



# Kepware Technologies

---

## ClientAce: Creating a Project with the DA\_Junction

June, 2013  
Ref. 1.03

©Kepware Technologies

# Table of Contents

- 1. Overview.....1
  - 1.1 Requirements .....1
- 2. Creating a Windows Form Application .....1
  - 2.1 Adding Objects to the Form .....1
  - 2.2 Adding the DA\_Junction to the Form.....2
  - 2.3 Configuring the DA\_Junction.....2
- 3. Testing the Project .....7
- 4. Summary .....8

# 1. Overview

This document intends to demonstrate how to create a Windows Form application and connect to an OPC DA or OPC UA Server using the ClientAce DA\_Junction control.

## 1.1 Requirements

- Visual Studio 2008, 2010, or 2012. The Express versions are not supported.
- The latest version of ClientAce.
- A fundamental understanding of the Visual Studio VB.Net or C Sharp (C#) development environments.

**Note:** The project used in the following example was created in Visual Studio 2010 with ClientAce version 4.0.1.0. It is set to target the .Net 4.0 framework and the x86 CPU at compilation.

## 2. Creating a Windows Form Application

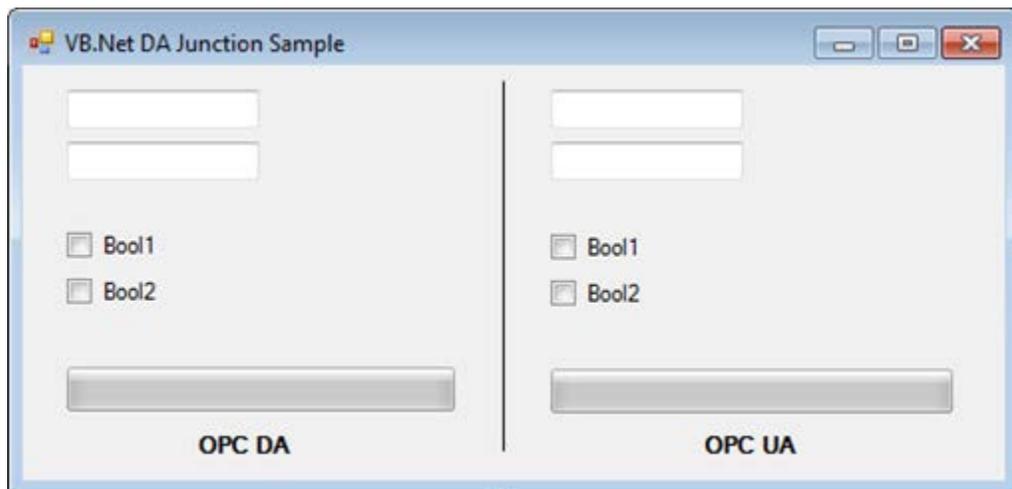
1. To start, launch Visual Studio and select **New Project**.
2. Next, select a VB.Net or C# **Windows Application Project** template. Then, specify the project name.
3. Once finished, click **OK** to create the solution and project.

### 2.1 Adding Objects to the Form

Controls must be added to the form before the application can display process data.

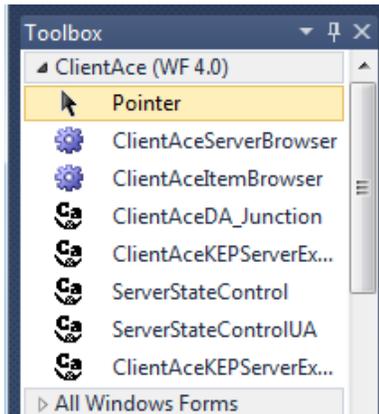
1. Use the **Toolbox** to drag several controls to the form.

**Note:** In this example, the controls for four checkboxes, four textboxes, and two progress bars will be added. The controls on the left side of the form will be updated via an OPC DA connection, and the controls on the right side of the form will be updated via an OPC UA connection.

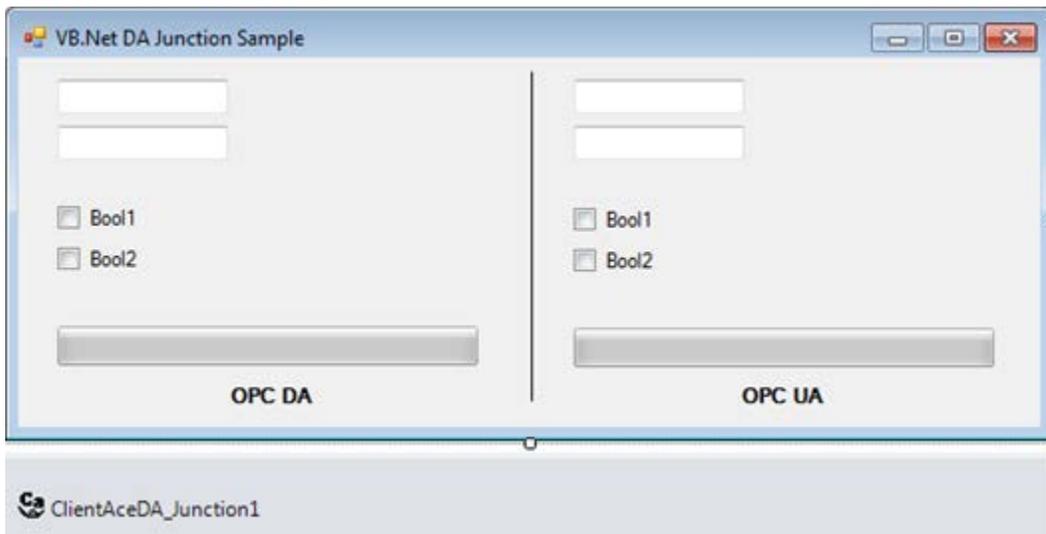


## 2.2 Adding the DA\_Junction to the Form

1. In the **Toolbox**, expand the **ClientAce Controls** group.
2. Select **ClientAceDA\_Junction**, and then drag and drop it onto the form.

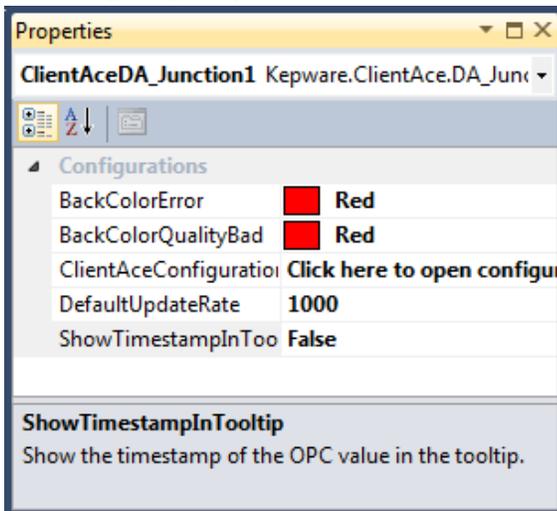


**Note:** Because the DA\_Junction is a background control (like a Timer), an instance will be created below the form's display area.



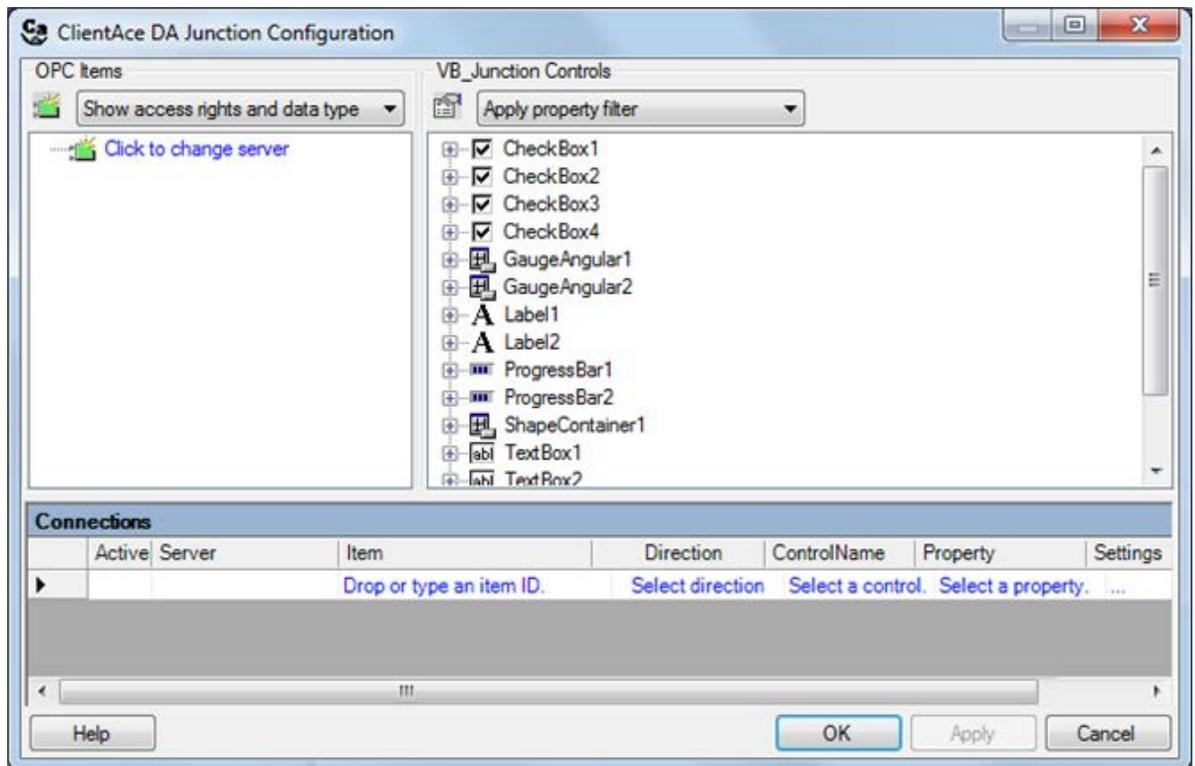
## 2.3 Configuring the DA\_Junction

1. Select the new DA\_Junction control on the form. Then, open its properties in the **Property Explorer**.



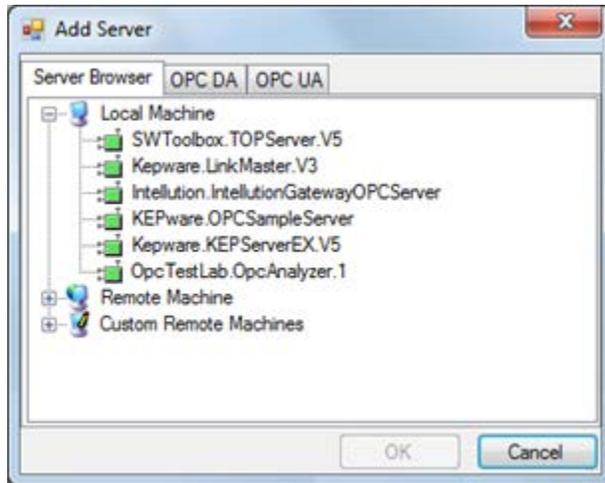
2. The following properties are configurable at design time:
  - **BackColorError:** This property specifies the background color that will be used on a control to indicate that there is an error when communicating with the server (such as, if the items used in DA\_Junction could not be added to the server).
  - **BackColorQualityBad:** This property specifies the background color that will be used on a control to indicate that the Input OPC Item's quality is Bad.
  - **DefaultUpdateRate:** This property specifies the default rate at which the Input OPC Items will be polled. Each item can be configured to use a different update rate.
  - **ShowTimestampInTooltip:** This property displays the timestamp for each item in a tooltip whenever the cursor hovers over a control.
3. Next, select **ClientAceConfiguration** to open the **ClientAce DA Junction Configuration**.

**Note:** The controls that have been added to the form will be discovered and displayed automatically.

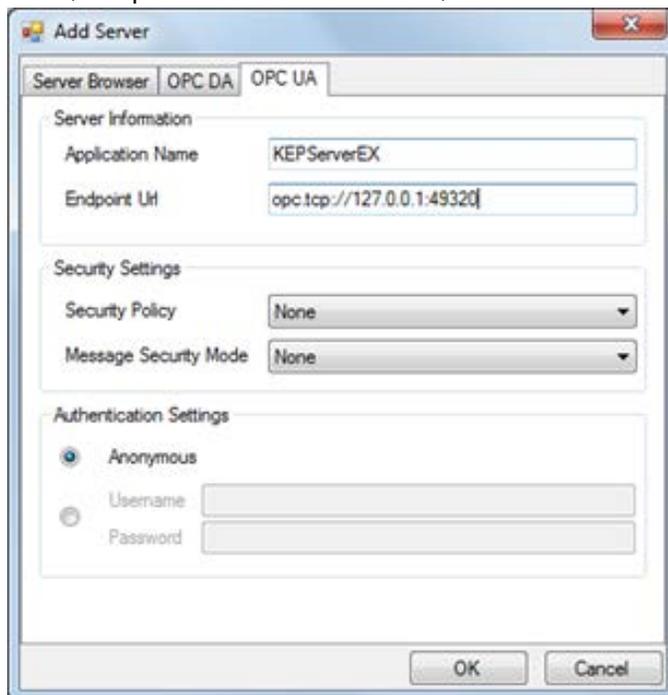


4. Beneath **OPC Items**, select the **Click to change server** prompt located in the left pane.
5. In **Add Server**, open the **Server Browser** tab. Then, browse for and select an OPC DA server.

**Note:** If an OPC UA Local Discovery Server is installed on the machine, the browser will show all registered OPC UA Endpoints.

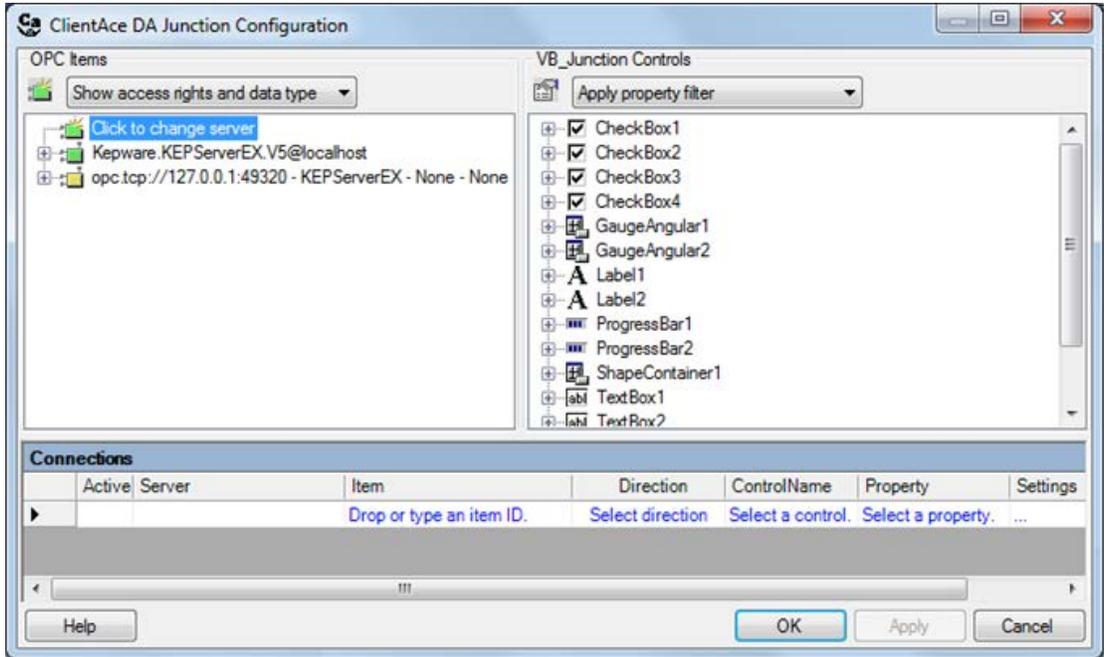


6. Once finished, click **OK**.
7. Next, reopen **Add Server**. Then, select the **OPC UA** tab.

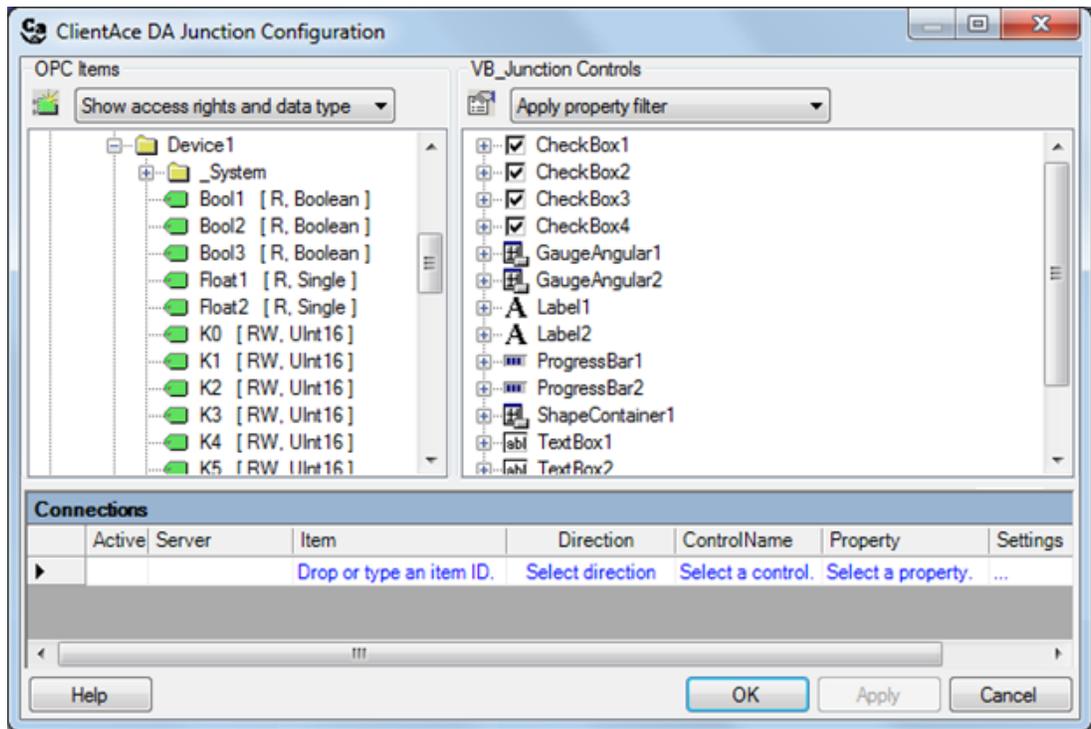


8. OPC UA connections that do not use security require both an **Application Name** and an **Endpoint Uri**. Descriptions of the parameters are as follows:
  - **Application Name:** This parameter specifies the name that will be used to identify the connection to the server. This can be the server name or a name that identifies the process.
  - **Endpoint Uri:** This parameter specifies the type of OPC UA connection, the location of the server, and the port on which the server is listening for connections.

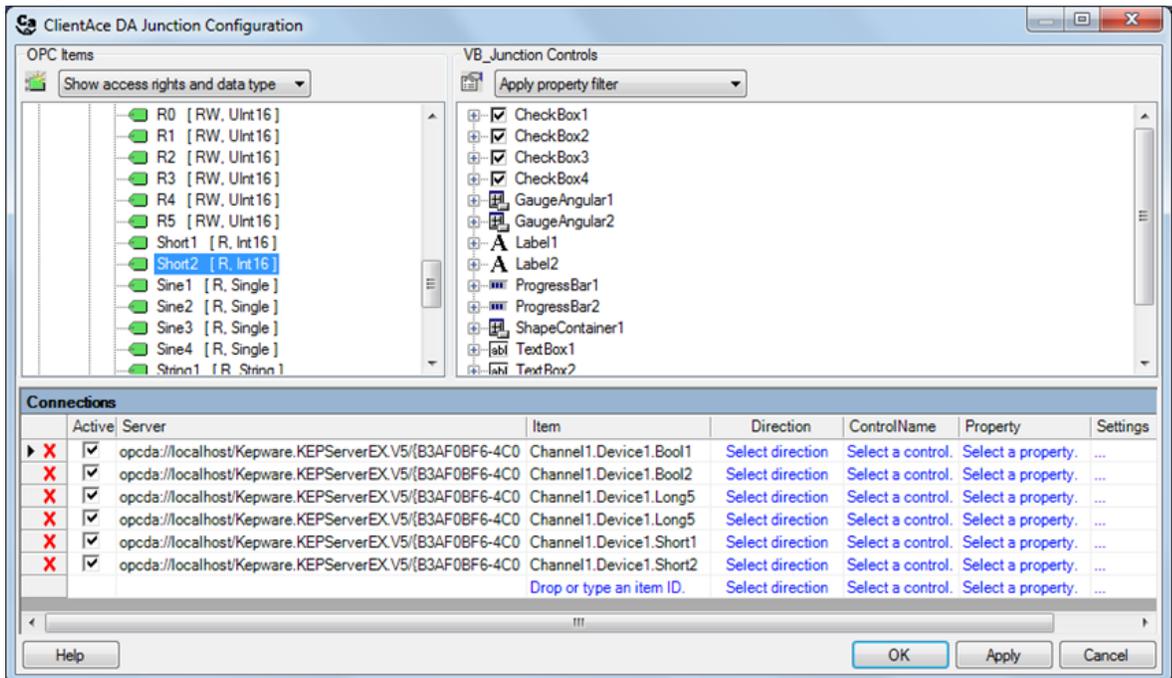
**Note:** At this point, there should be two connected servers in the OPC Items pane.



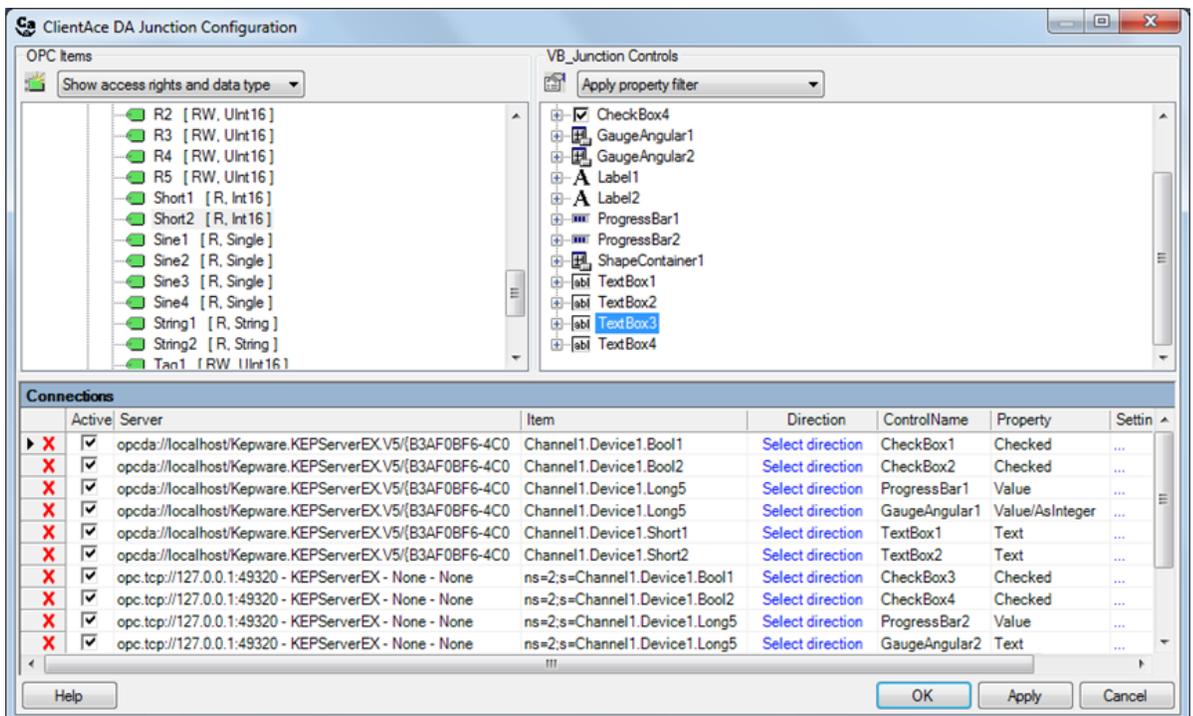
9. Expand the **OPC DA Server** folder to display the tags that will be used.



- Next, add the OPC Items that will be used to display data in the application to the **Connections** pane. To do so, either double-click on the OPC Item or drag and drop it onto the pane.



- Expand the **OPC UA Server** to display the items that will be used.
- To link the OPC Items to the form controls, select a control beneath **Form1 Controls** and then drag it to the item in the **Connections** pane that will be updating it.



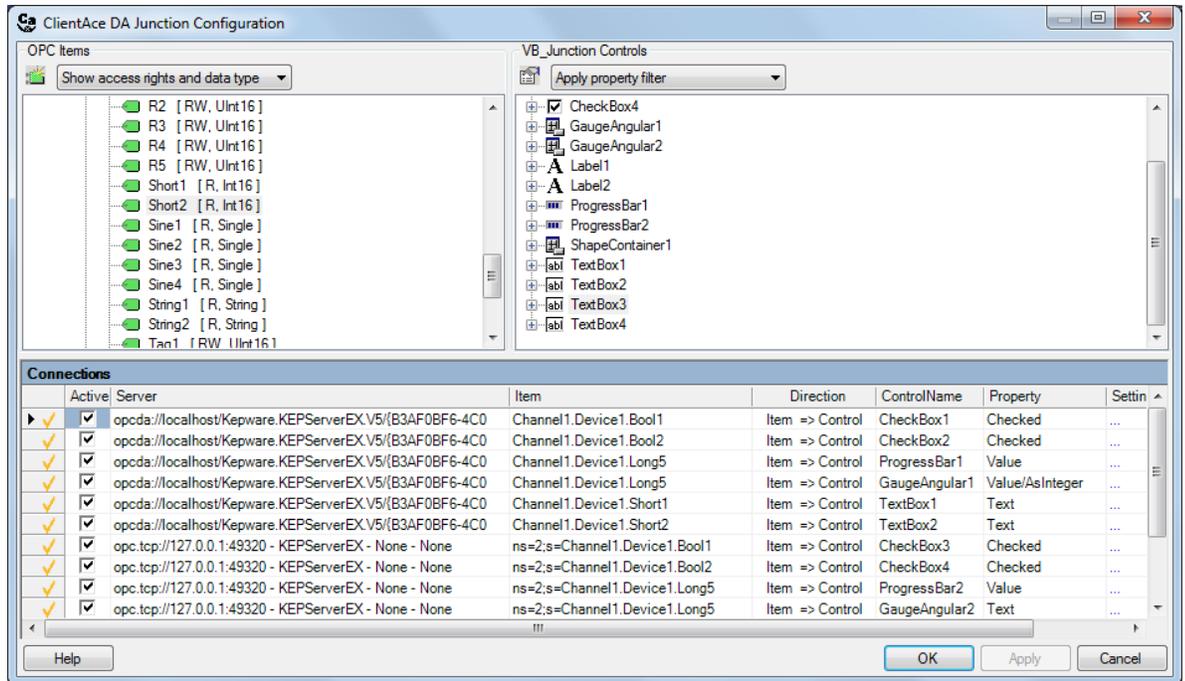
**Important:** The DA\_Junction automatically associates the most common controls with the property that they update. For example, the Text property will be associated with a Textbox, and the Value property will be associated

with the Progress Bar. In some cases, users will need to change the property chosen by the DA\_Junction (such as changing a property from Text to Value/AsInteger).

13. Next, specify the direction in which data will flow during Runtime. The following directions are supported:

- Item to Control
- Control to Item
- Bi-directional (between the Item and the Control)

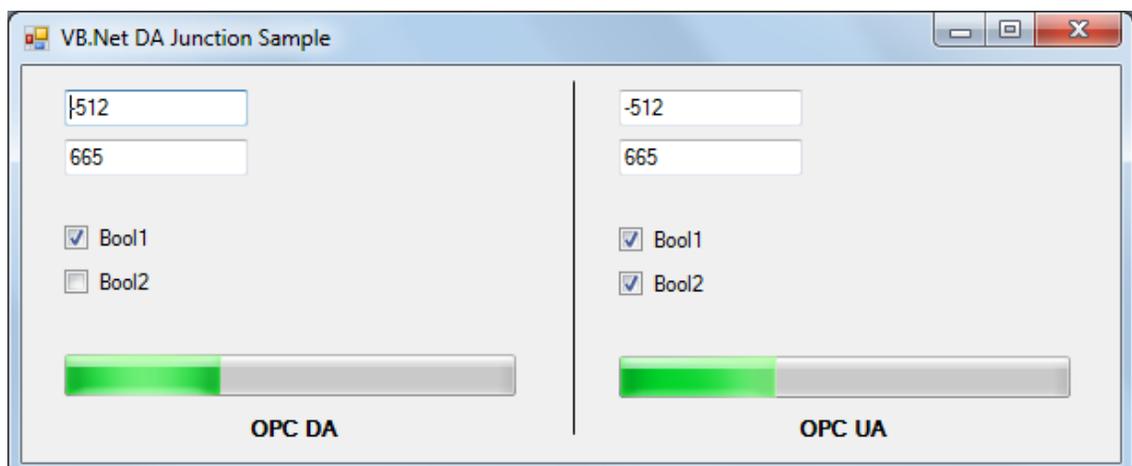
**Note:** Because only data is going to be displayed in this example, Item to Control is selected.



14. Click **Apply** to save the configuration. Then, click **OK**.

### 3. Testing the Project

1. To test the project, run it in Debug Mode. To do so, press the **F5** key while in the **Visual Studio Development Environment**.



2. If the server is running, data will be displayed in the form.

## 4. Summary

At this point, users should have a basic understanding of how to create a Windows Form application using ClientAce. As demonstrated, this type of application does not require coding. For more information on the DA\_Junction's other features, refer to the DA\_Junction example that installs with ClientAce.