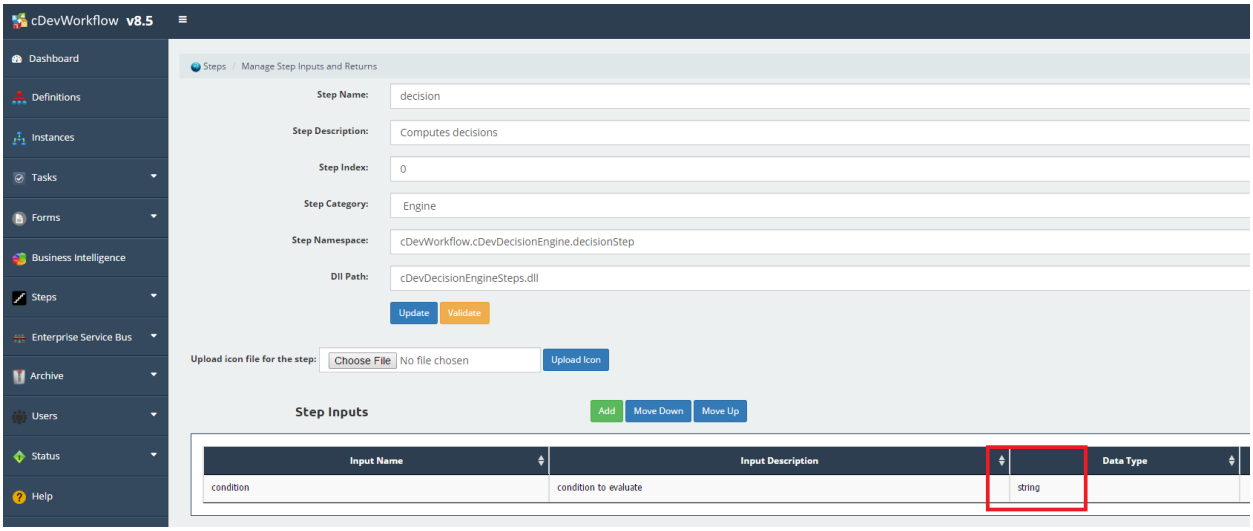
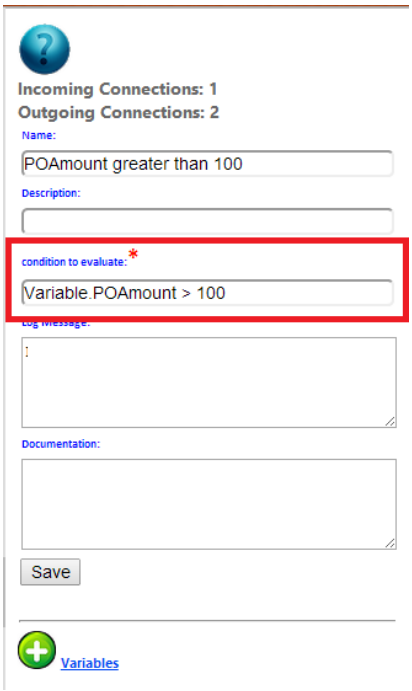
What is a custom data type?

Custom data type is a data type used on a step property.

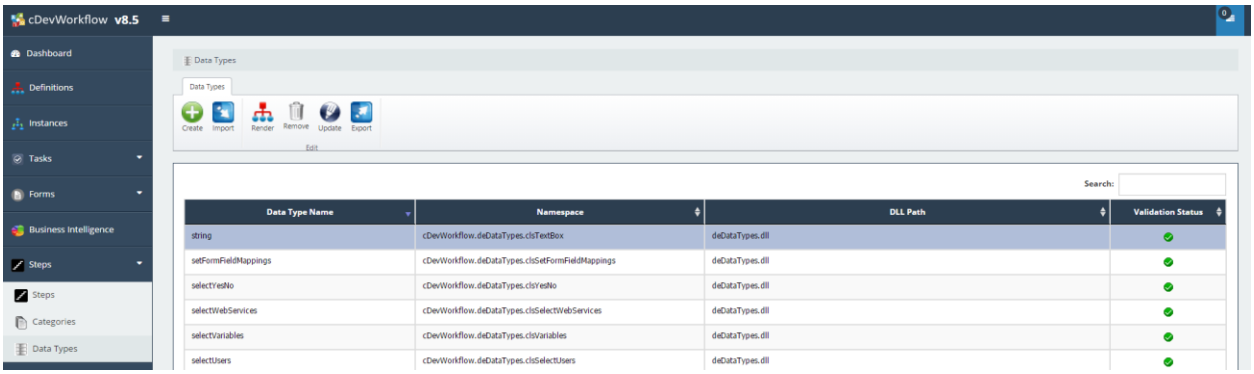


In reference to the above diagram, the decision step has 1 property called “condition” that is “string” data type. When the step is rendered within the workflow designer, the “condition” property is rendered as a textbox.

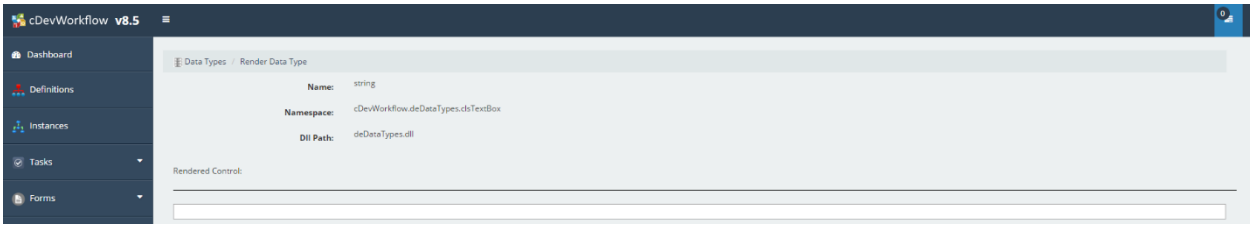


Custom data types

Custom data types can be configured using the data type screen of the cDevWorkfow Configuration Manager.



When the “string” data type is rendered, it looks as follows:



FLOWRIGHT
powered by cDevWorkflow

Writing a custom data type

Custom data type can be written easily by implementing the “IDataType” interface. The following code shows the actual code for the “string” data type.

```
public class clsTextBox : IDataType
{
   public string render(string controlName, string[] defaultValues, string se
lectedValue, HttpRequest oRequest)
   {
      TextBox oBox = new TextBox();
      oBox.ID = controlName;
      oBox.Width = Unit.Percentage(99);

      if (selectedValue != null)
      {
         oBox.Text = selectedValue;
      }

      StringWriter stringWriter = new StringWriter();
      HtmlTextWriter writer = new HtmlTextWriter(stringWriter);

      oBox.RenderControl(writer);

      return (stringWriter.ToString());
   }
}
```

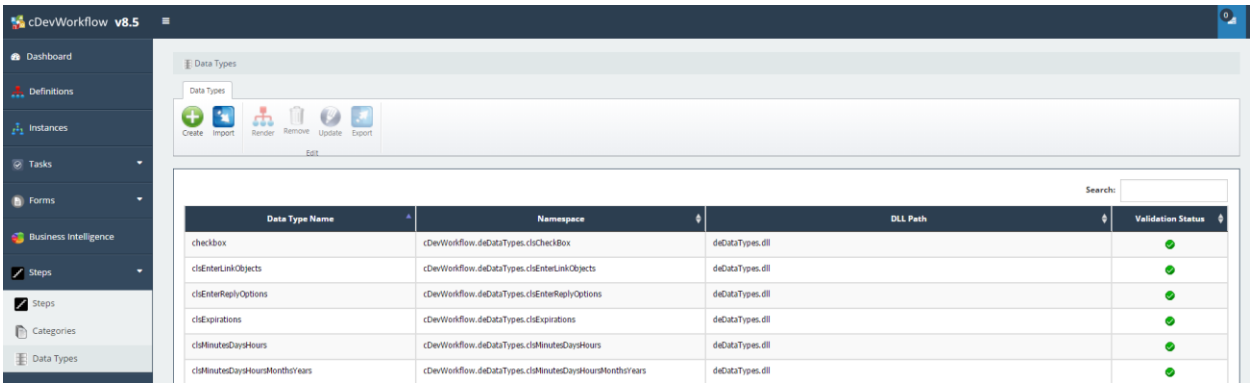
Data type interface “IDataType” has one single method that needs to be implemented:

```
public string render(string controlName, string[] defaultValues, string selec
tedValue, HttpRequest oRequest)
```

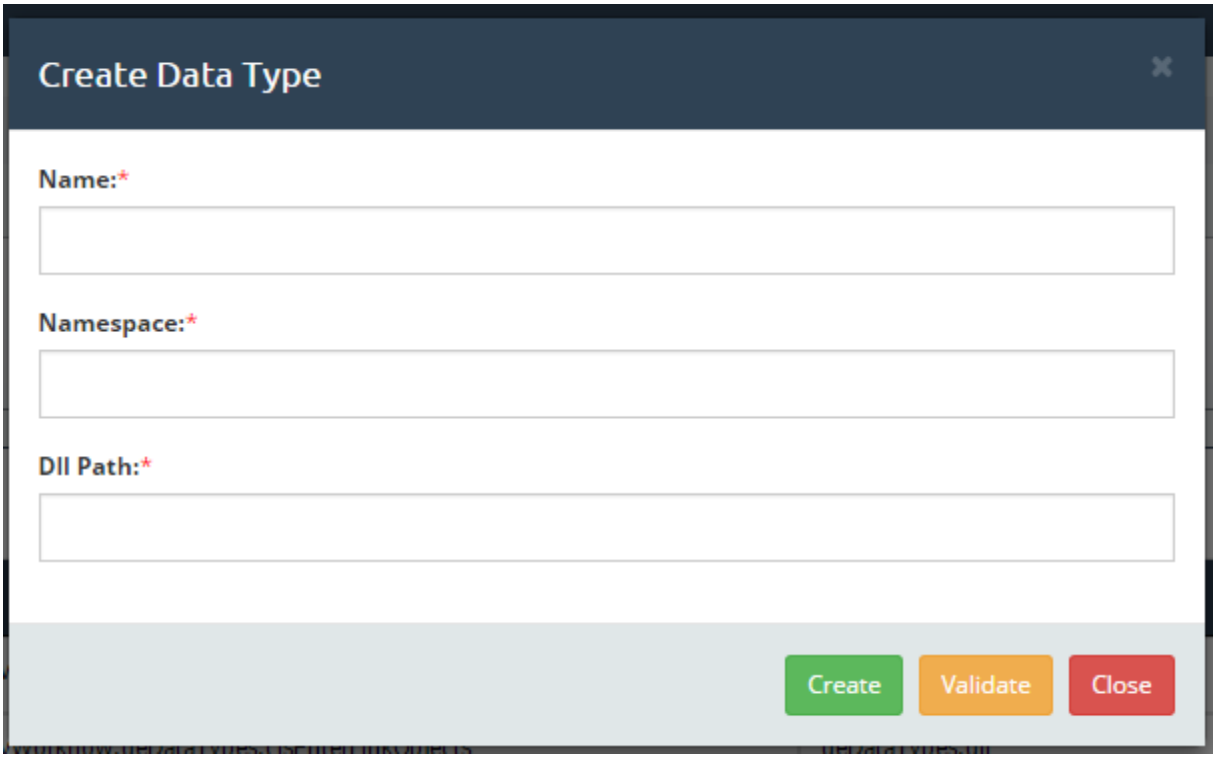
The return from the “render” function is to return a HTML string to render the HTML controls for the property.

Configuring the custom data type

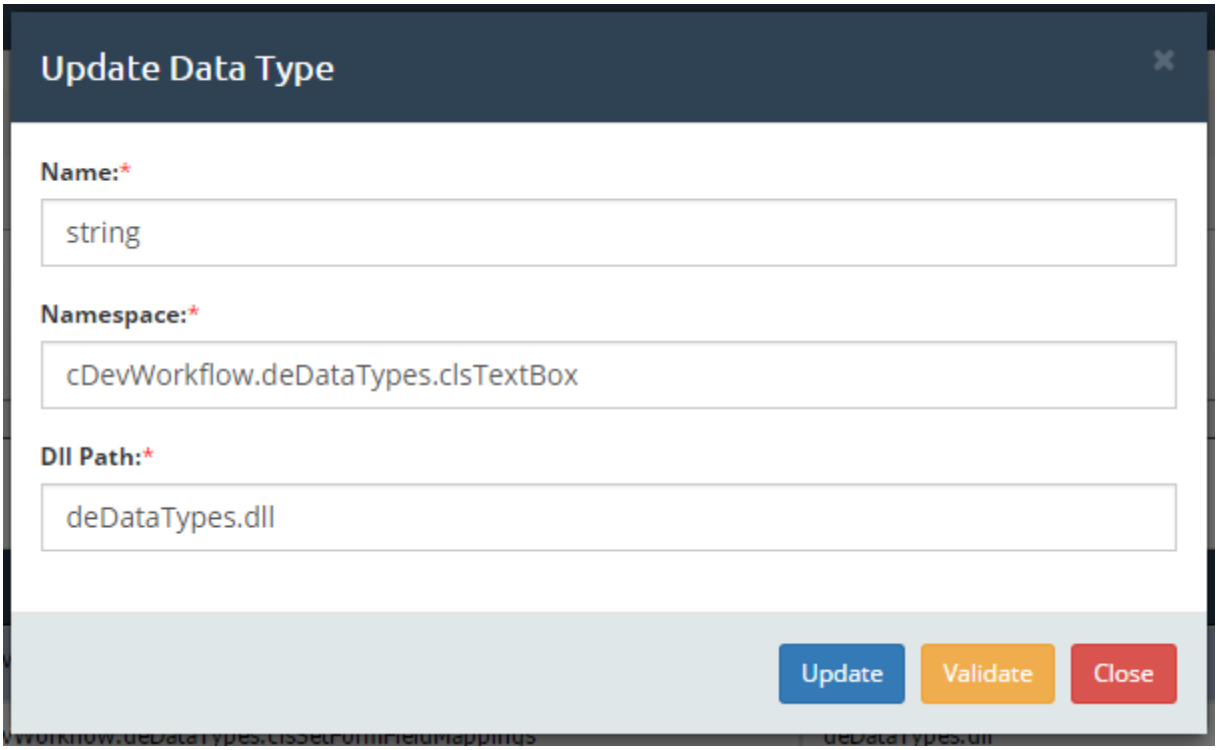
Once the custom data type is compiled into a DLL, it can be configured through the cDevWorkflow Configuration Manager. Navigate to the “Data Types” page within the Configuration Manager.



Click the “Create Data Type” button to configure the definition for the new step.



Here's what the configuration looks for the "string" data type step:



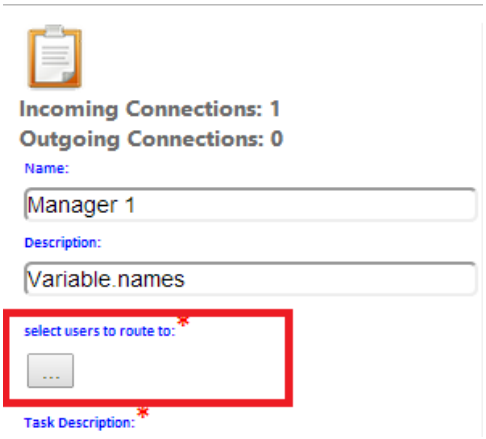
Once the data type is configured, data type can be used for defining properties for a given step.

Building and configuring an advanced data type

Advanced data types are data types that perform complex operations and store the value in a property. Data type “SelectUsers” is one of them. The following is the actual code for the step:

```
public class clsSelectUsers : IDataType
{
   public string render(string controlName, string[] defaultValues, string se
lectedValue, HttpRequest oRequest)
   {
      StringWriter stringWriter = new StringWriter();
      HtmlTextWriter writer = new HtmlTextWriter(stringWriter);

      Button oBtn = new Button();
      oBtn.Text = "...";
      oBtn.OnClientClick = string.Format("window.open('deSelectUsers.aspx?fie
ldName={0}','','resizable=1,status=1,scrollbars=1,width=500,height=350');", c
ontrolName);

      oBtn.UseSubmitBehavior = false;

      oBtn.RenderControl(writer);

      string sHTML = HttpUtility.HtmlDecode(stringWriter.ToString());

      return (sHTML);
   }
}
```

The above code renders a button for the property and the button launches the window for user selection.



Clicking the button will display the following User selection dialog: