Read Xml Document

Contents

Introduction

Class XmlDocument in Microsoft .Net Framework can be used to search and manipulate XML contents. We use a simple sample to show how to search Xml contents.

A user (enannos) posted following XML contents in the user forum:

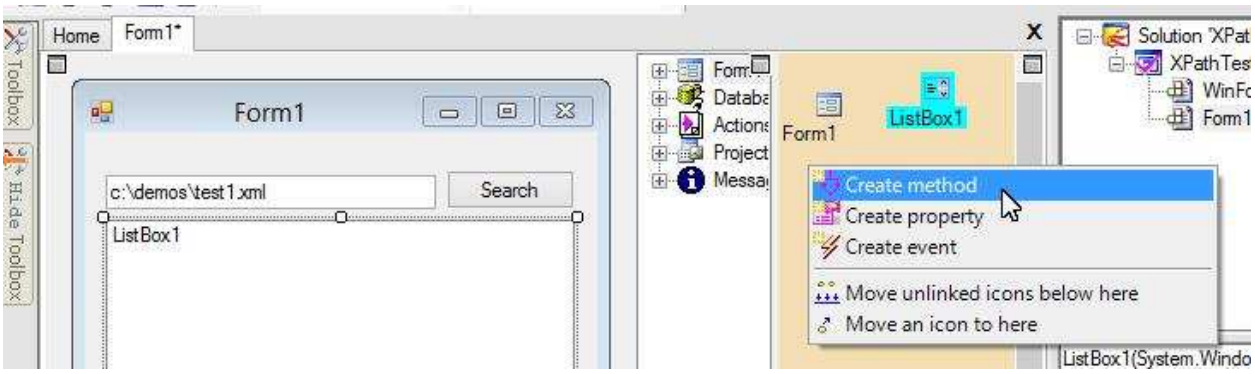
```
<Root>
<Contact id="5">
   <name>pc1</name>
    <note>
    </note>
    <vpn>
    </vpn>
    <type>0</type>
    <gwaddr>
    </gwaddr>
    <defgw>False</defgw>
    <meth>0</meth>
    <caddr>192.168.1.150</caddr>
```

```
    <url>
    </url>
    <email />
</Contact>
<Contact id="6">
    <name>Tablet</name>
    <note>
    </note>
    <vpn>
    </vpn>
    <type>0</type>
    <gwaddr>
    </gwaddr>
    <defgw>False</defgw>
    <meth>0</meth>
    <caddr>192.168.123.27</caddr>
    <url>
    </url>
</Contact>
</Root>
```

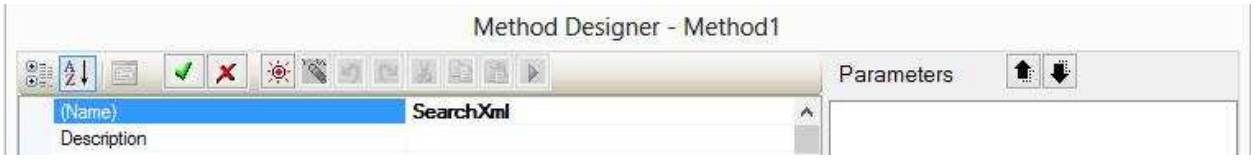
It was requested to search for the IP addresses in the contents. Let's use a list box to show the IP addresses found in the contents.

Create method

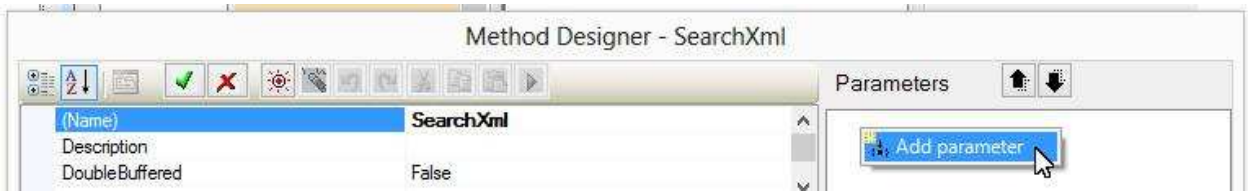
We create a method to search an XML file:



Rename the method to "SearchXml":



Add a method parameter to represent an Xml file path:

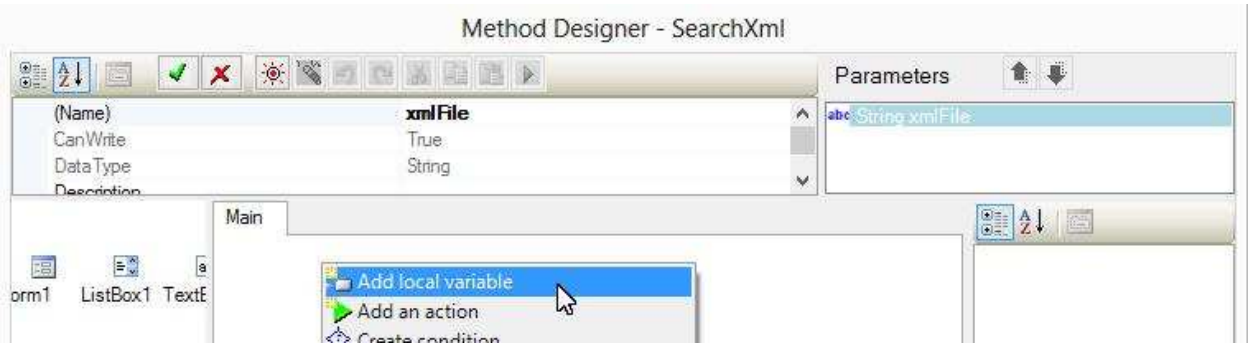


Choose parameter type:

Rename the parameter to “xmlfile”:

Add XmlDocument variable

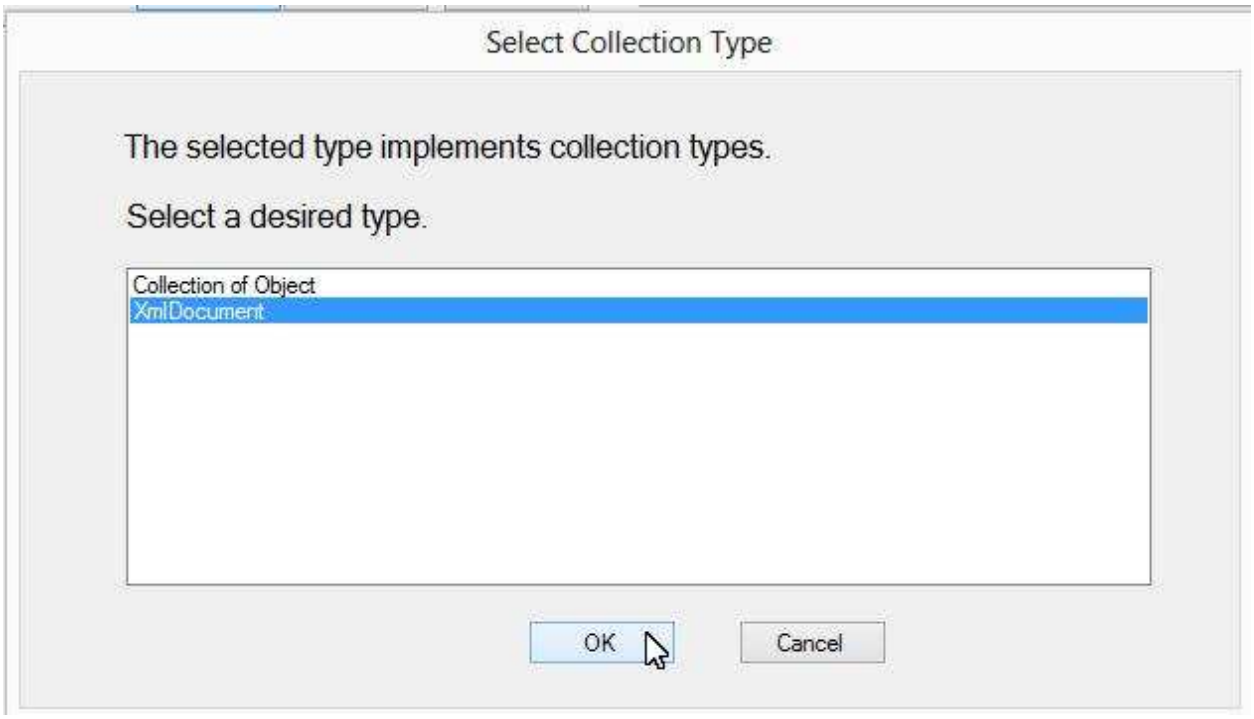
Add an XmlDocument variable to represent an instance of XML document:



Select XmlDocument class:

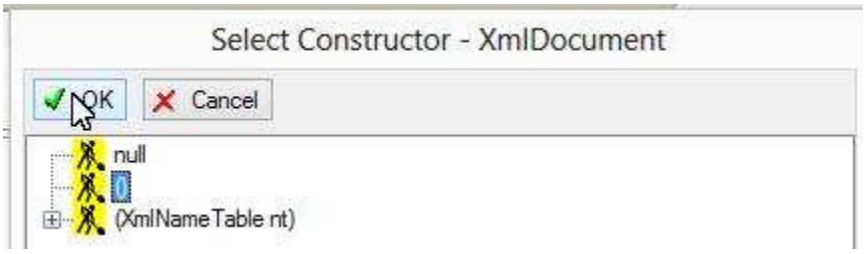

Object picker
Back
Next
Exit
Add
System.Windows.Input
System.Windows.Markup
System.Xml
ConformanceLevel
DtdProcessing


Object picker
Back
Next
Exit
Add
XmlDateTimeSerializationMode
XmlDeclaration
XmlDocument
XmlDocumentFragment

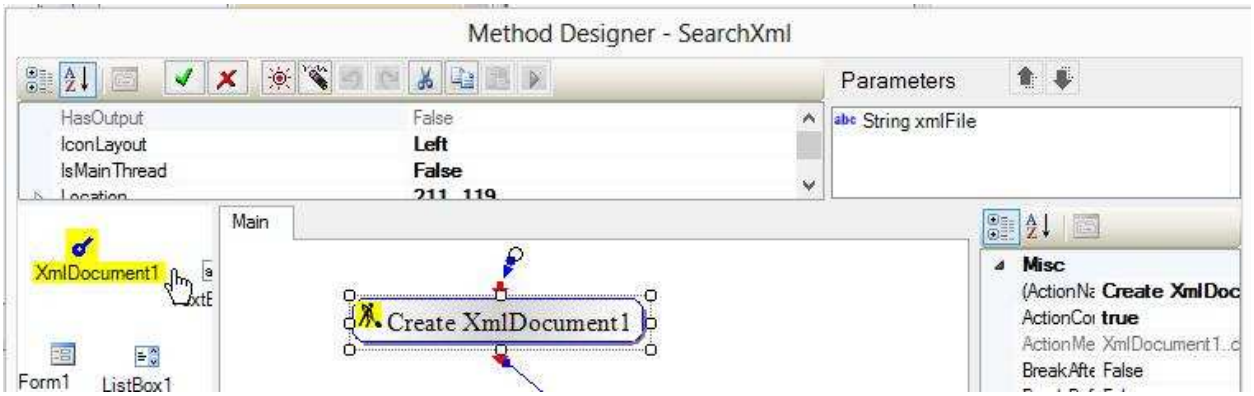

Select Collection Type
The selected type implements collection types.
Select a desired type.
Collection of Object
XmlDocument
OK
Cancel


Variable Name
Name:
XmlDocument1
OK

Choose a constructor:

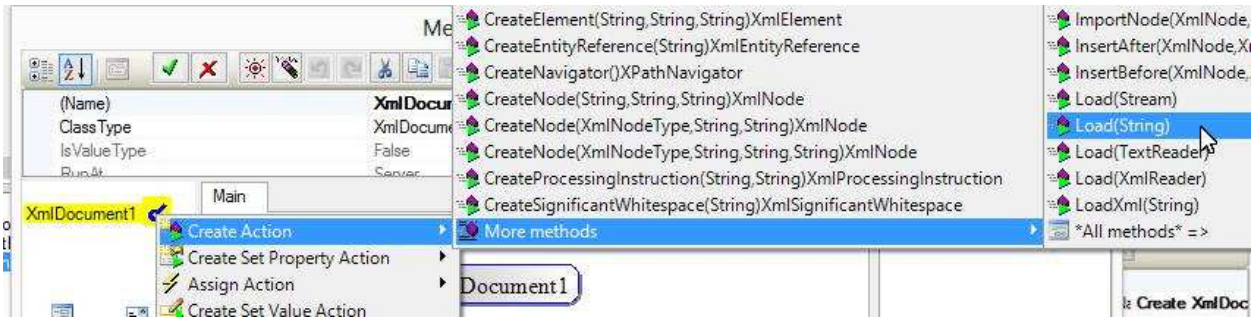


An instance of XmlDocument appears in the variable pane, and an action for constructing the instance appears in the Action pane:

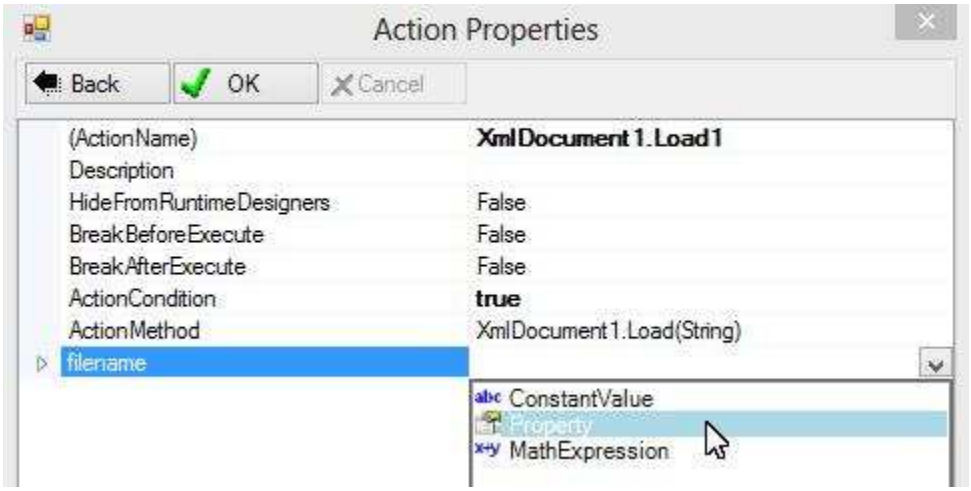


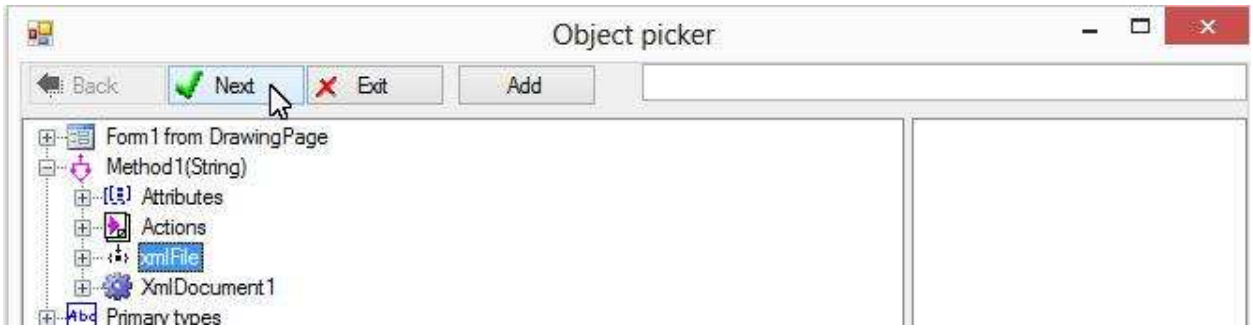
Load Xml file

Create an action to load an XML file into the XmlDocument variable:

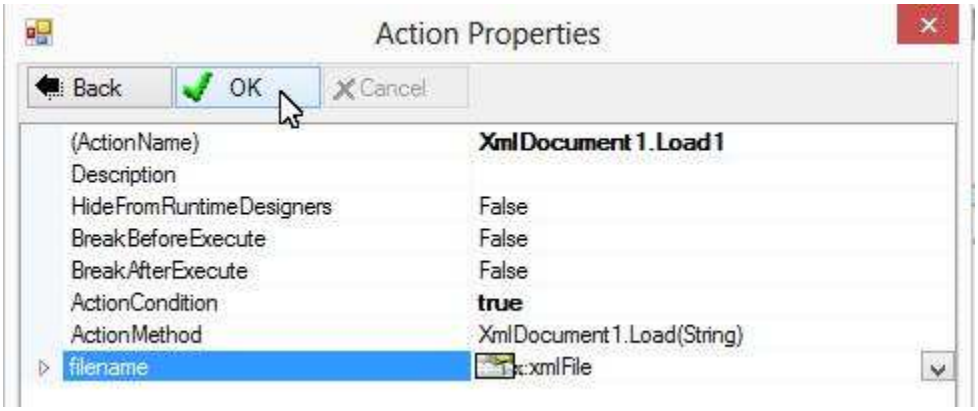


Use the method parameter as the action parameter:

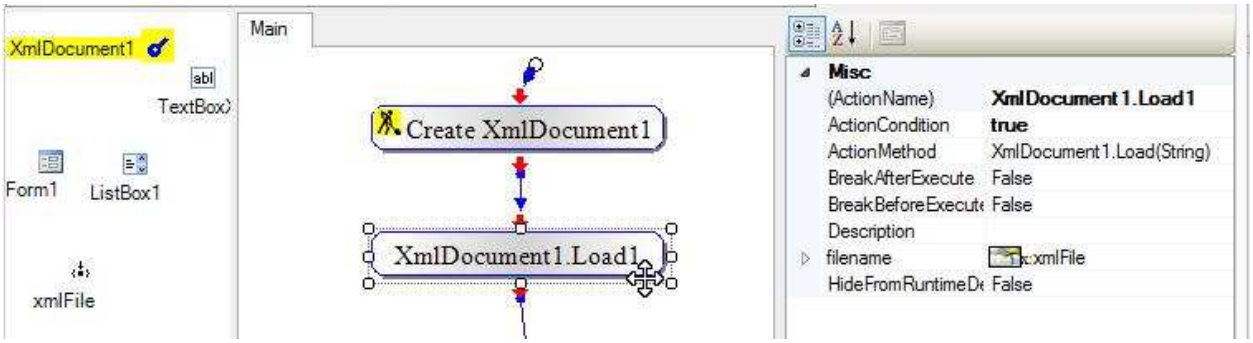




Click OK to finish creating the action:

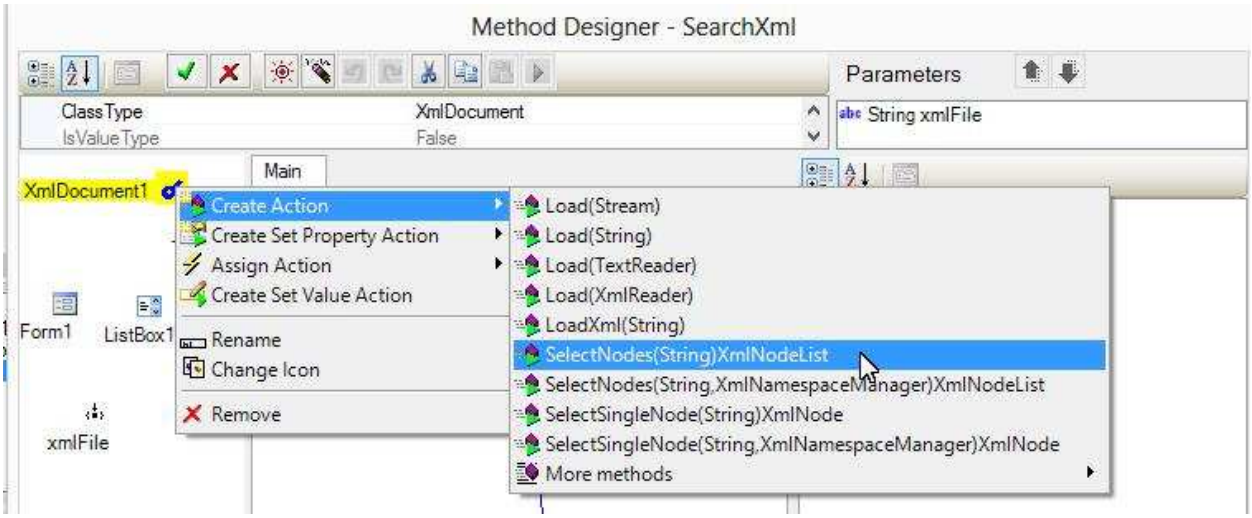


The action is created and appears in the action pane. Link it to the last action:



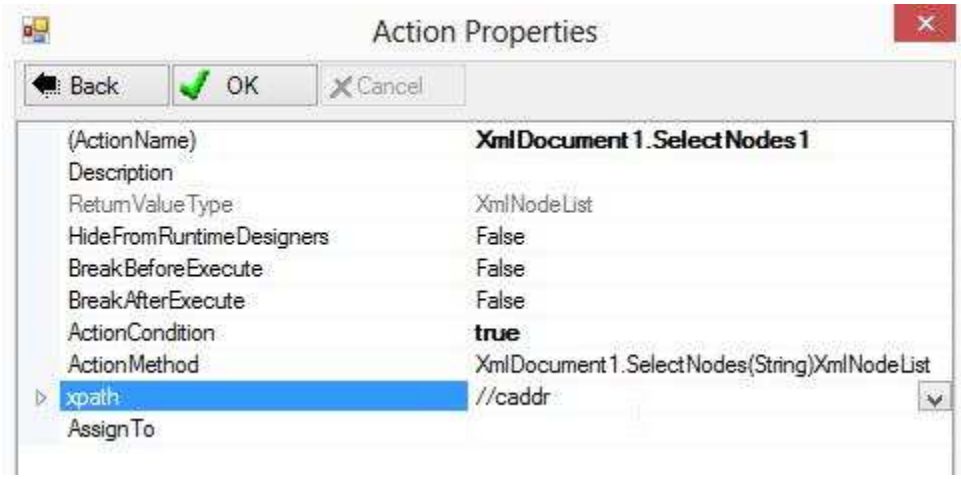
Search XML contents

In this sample XML file, the IP addresses are contained in nodes <caddr>. We may use a query "//caddr" to get all <caddr> nodes by an action of SelectNodes:

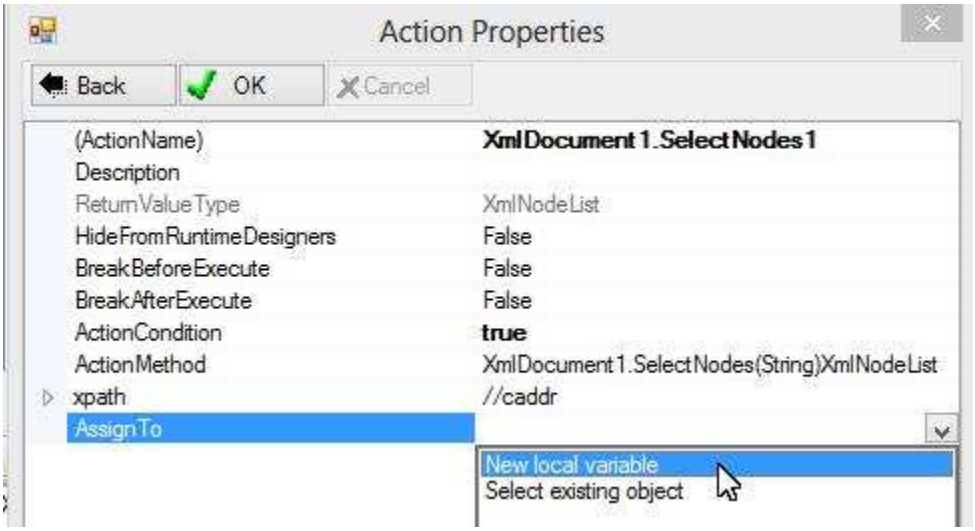


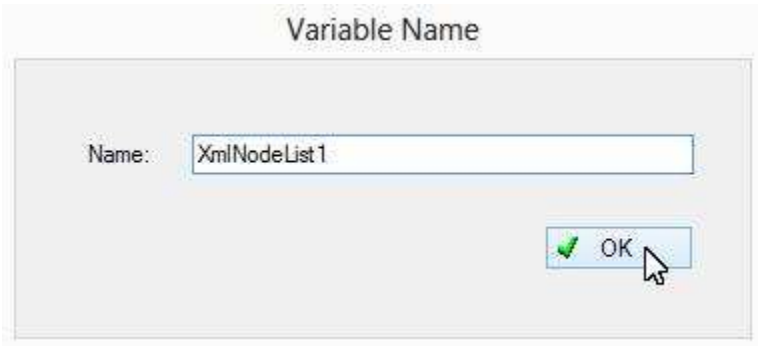
Note that the context menu for the XmlDocument is different than what we used in the last section when we were creating Load action. This is because we changed the context menu to show commonly used methods. You can do it by selecting "More methods" and then selecting "All methods".

Use //caddr for "xpath" parameter:

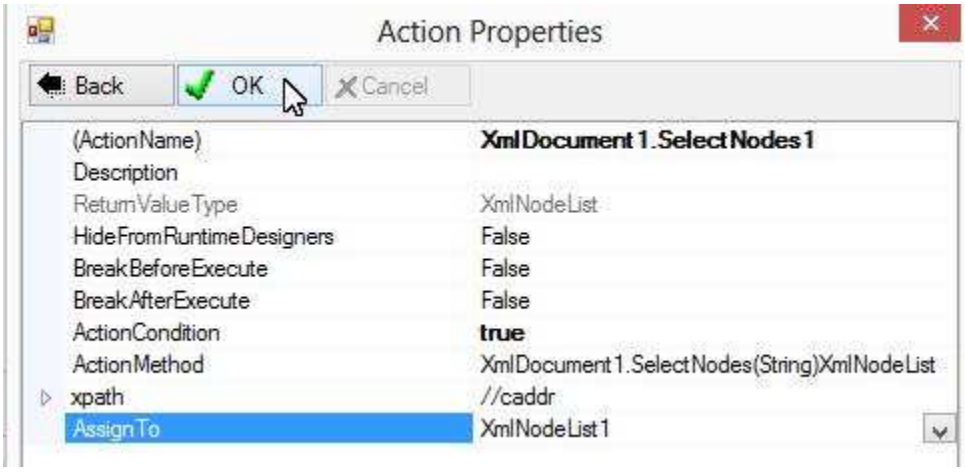


For “AssignTo” of the action, choose “New local variable”:

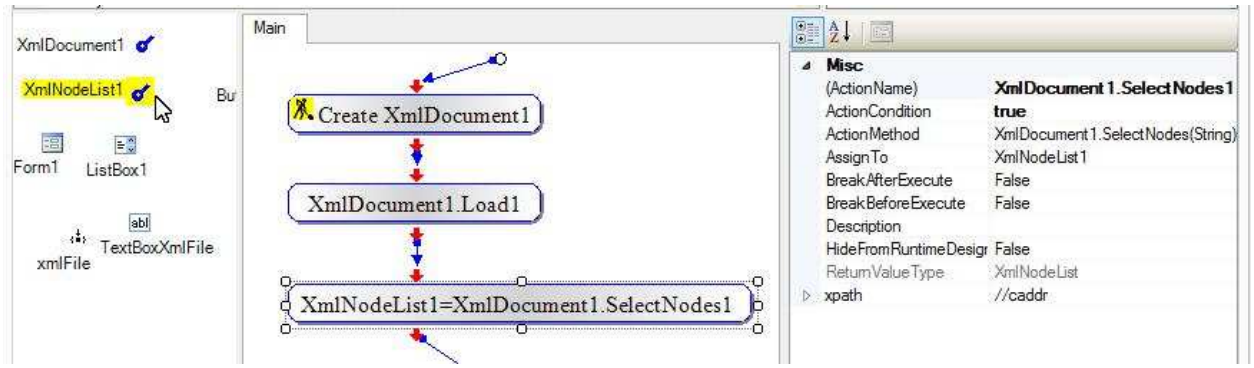




Click OK to finish creating the action:

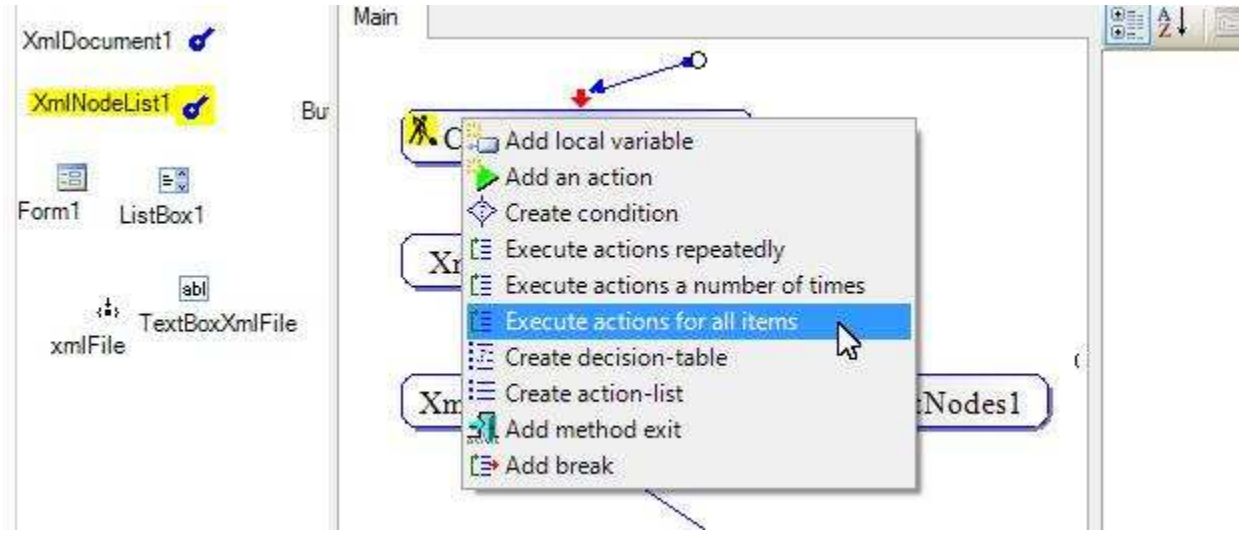


The action is created and appears in the Action Pane; a new variable appears in the variable pane. Link the action to the last action.



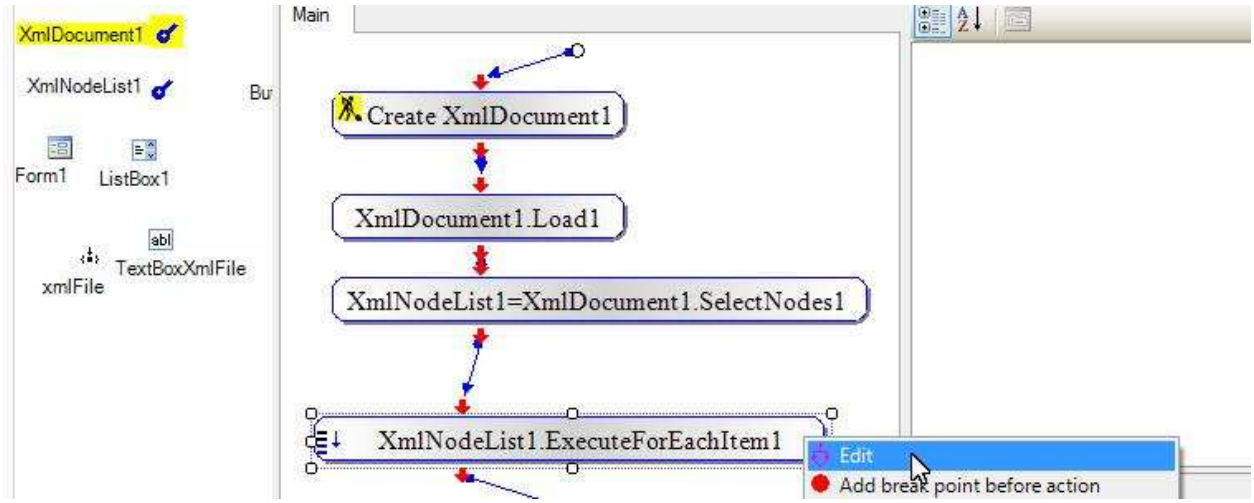
Go through XML nodes

Variable XmlNodeList1 contains all <caddr> nodes. We may create an action to go through these nodes. Right-click the Action Pane; choose “Execute actions for all items”:

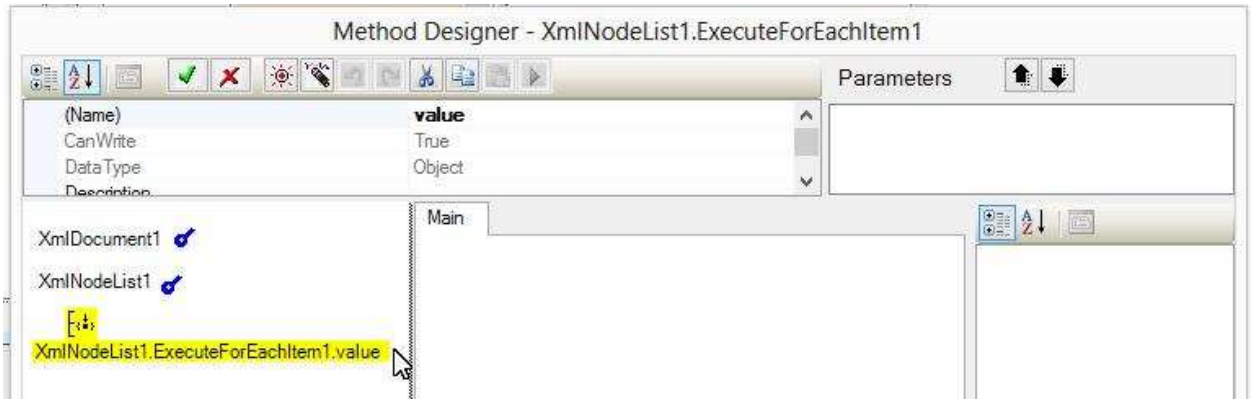


Choose variable XmlNodeList1:

A loop action appears in the action pane. Link it to the last action. To specify what we want to do about each <caddr> node, right-click the loop action and choose "Edit":



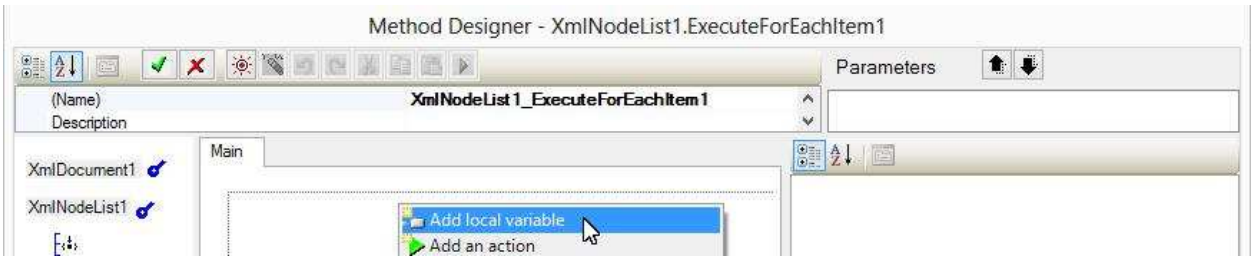
A new method editor appears to let us specify what to do:

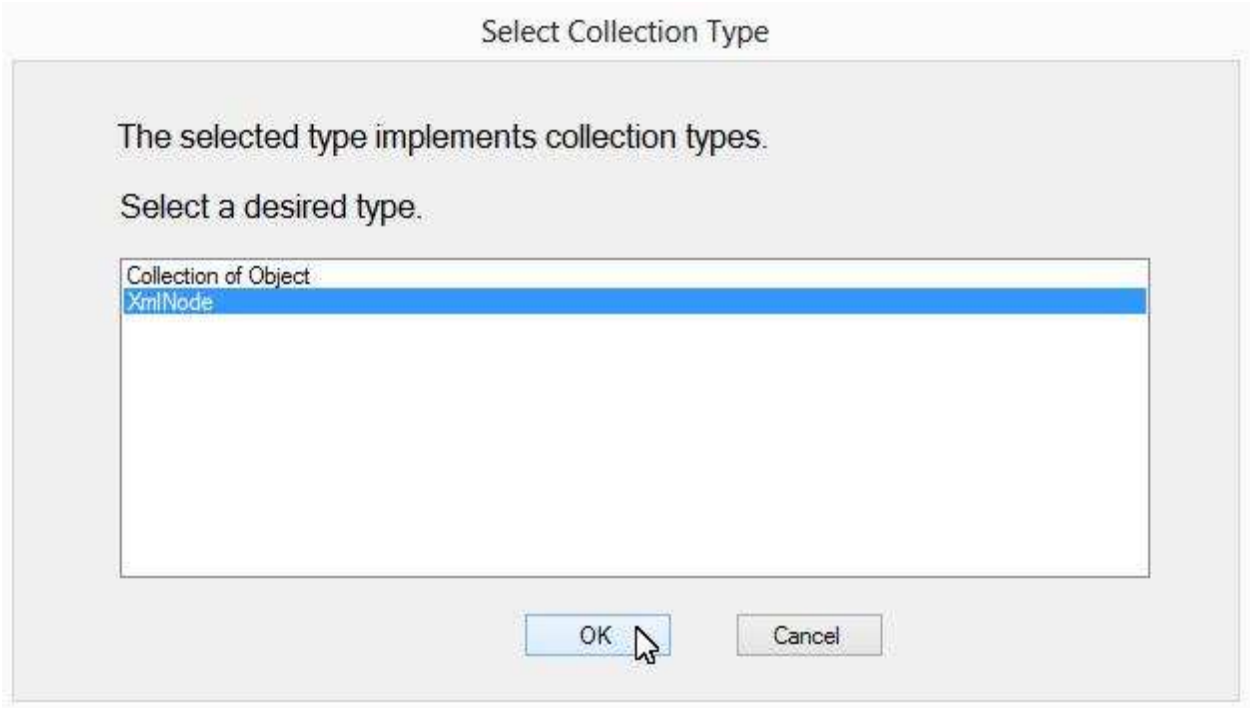


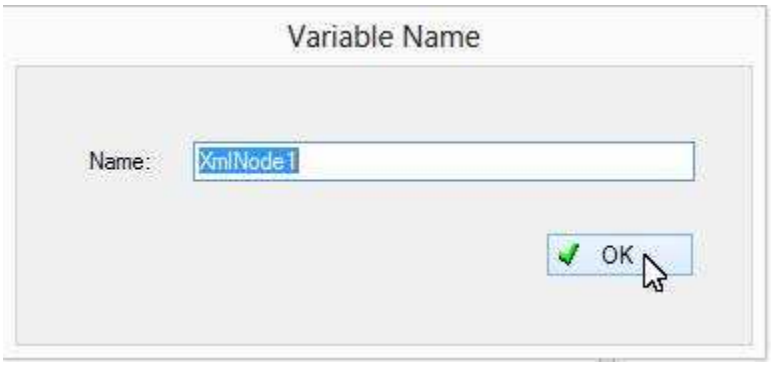
Note the variable "XmlNodeList1.ExecuteForEachItem1.value". It represents the <caddr> node to be worked on.

Cast item to XmlNode

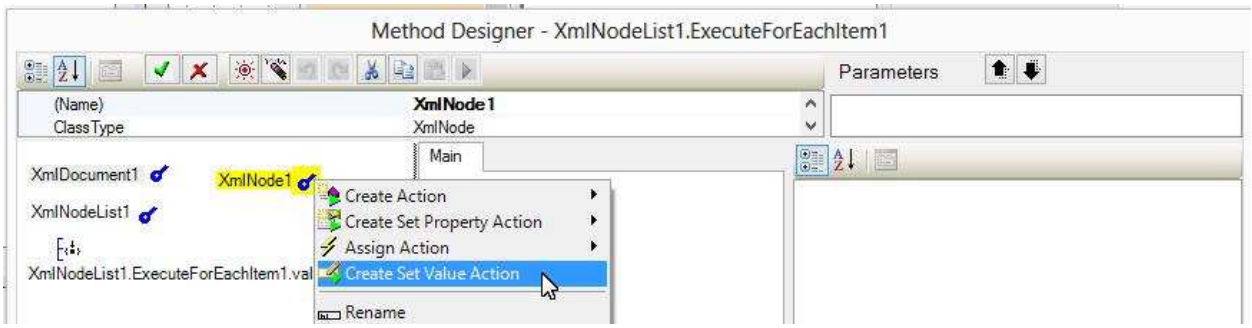
The variable “XmlNodeList1.ExecuteForEachItem1.value” is an XmlNode object. But currently Limnor Studio cannot detect it. So, let’s add an XmlNode variable to cast “XmlNodeList1.ExecuteForEachItem1.value” to it:



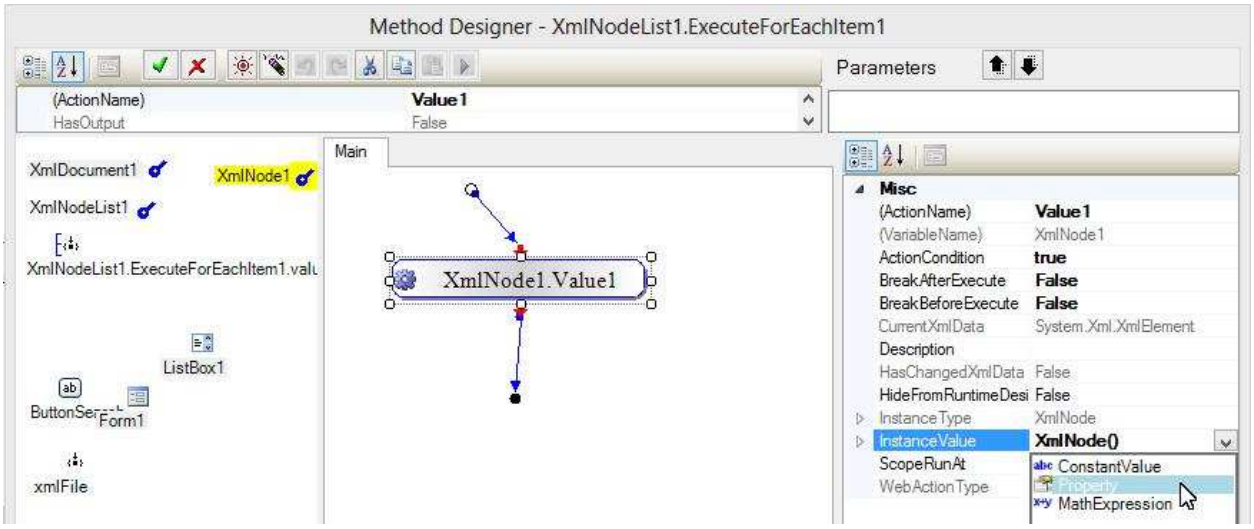


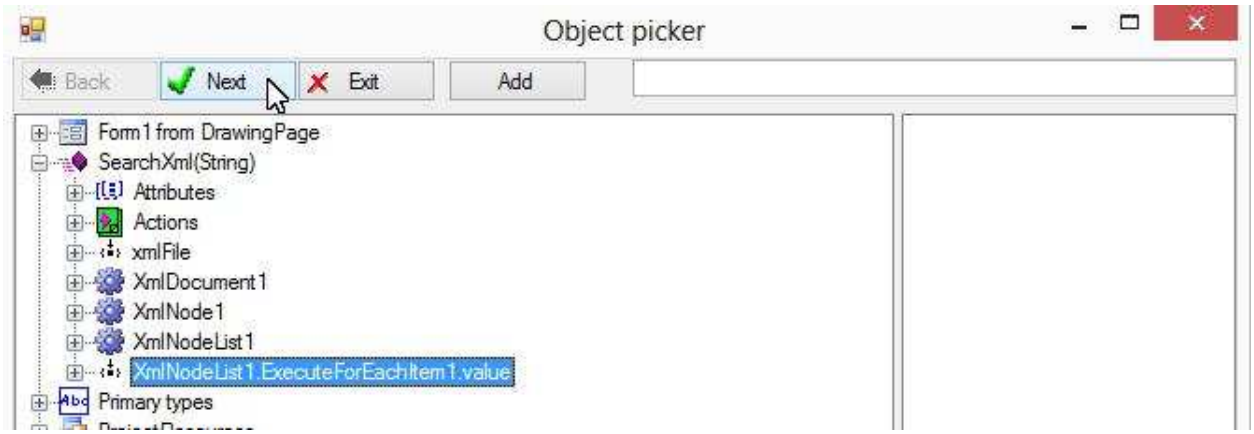


A new variable appears in the variable pane. Create an action to cast "XmlNodeList1.ExecuteForEachItem1.value" to it:

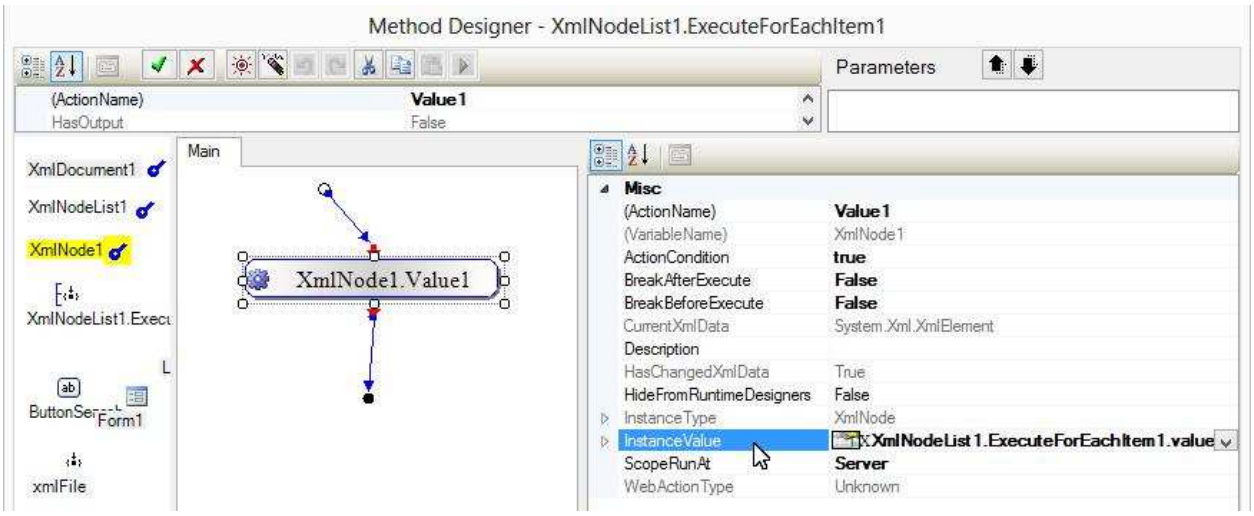


An action appears in the Action Pane. Set its instanceValue to "XmlNodeList1.ExecuteForEachItem1.value"





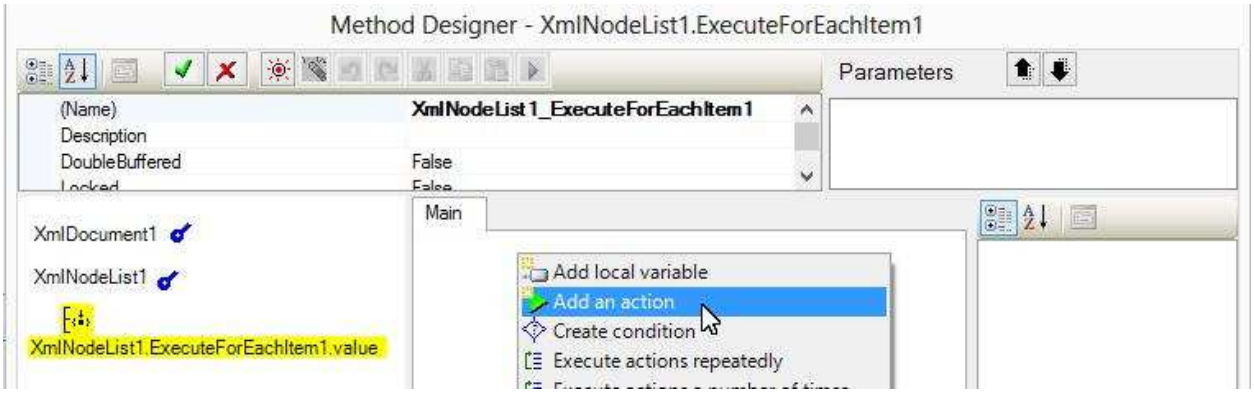
Now variable XmlNode1 is an <caddr> node to be used:



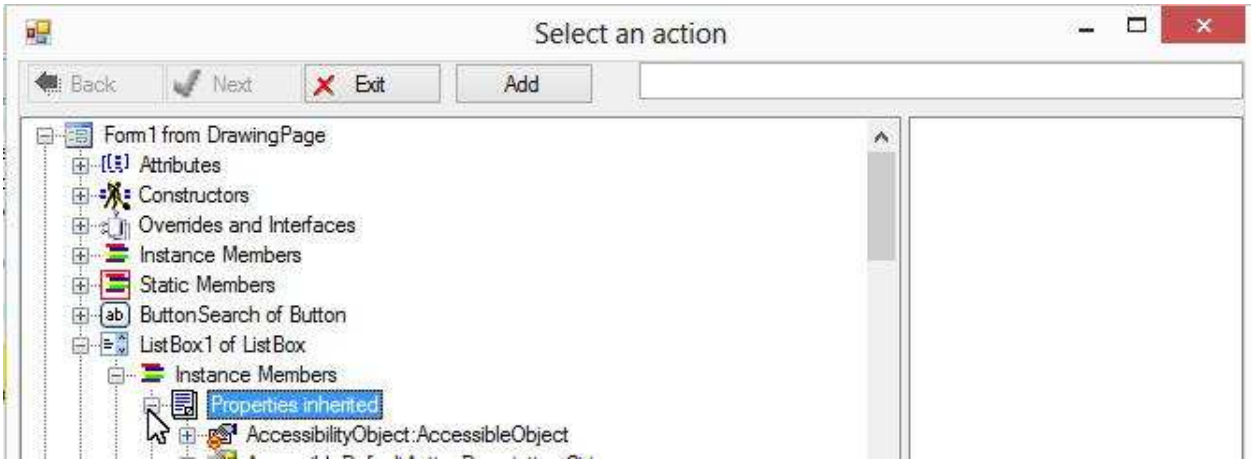
Show node contents

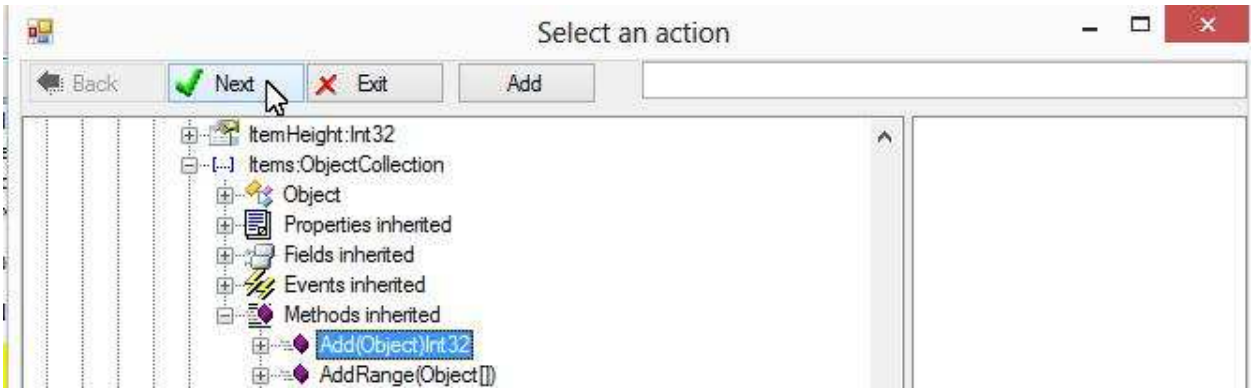
An XML node may contain attribute strings, text and child nodes. In this sample, we simply show the text of the <caddr> node in the list box. We may create an Add method of the Items property of the list box.

Right-click the Action Pane; choose “Add an action”:

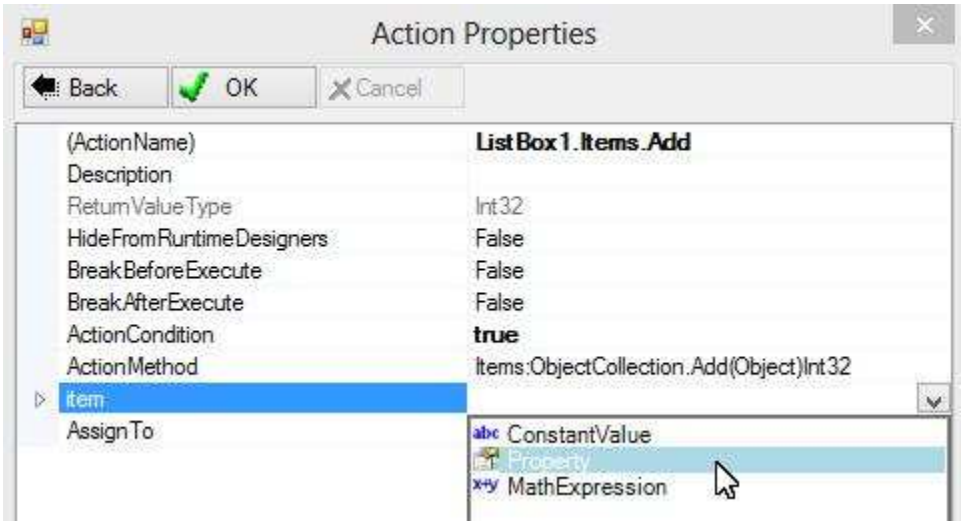


Select the Add method of the Items property of the list box:

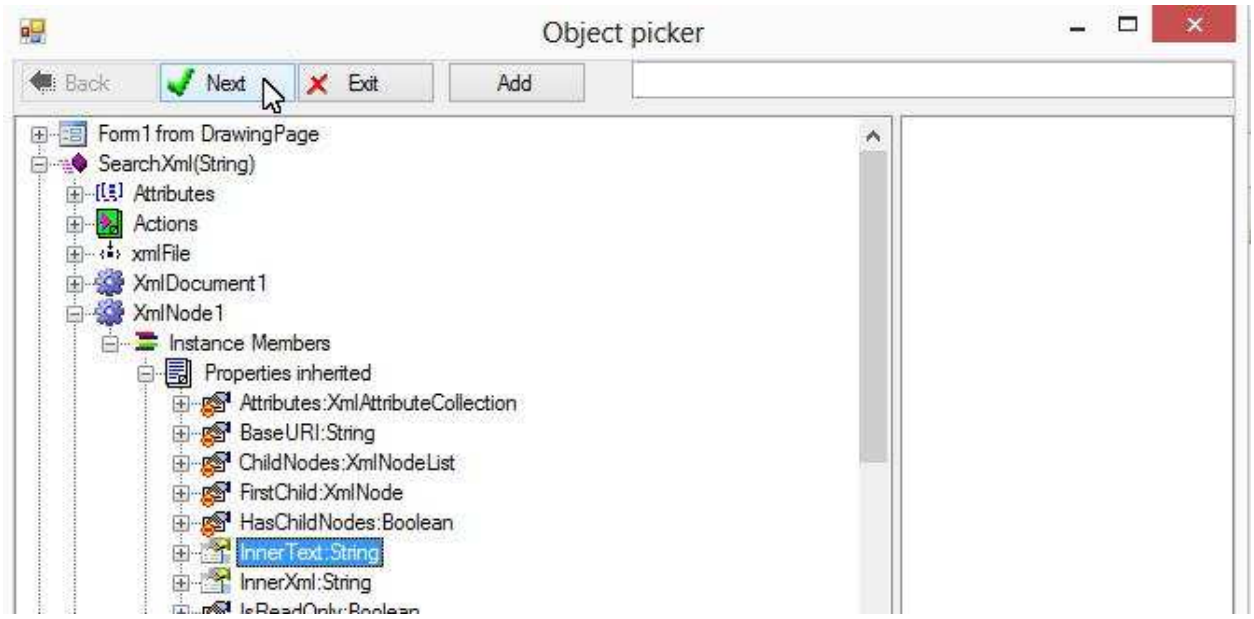




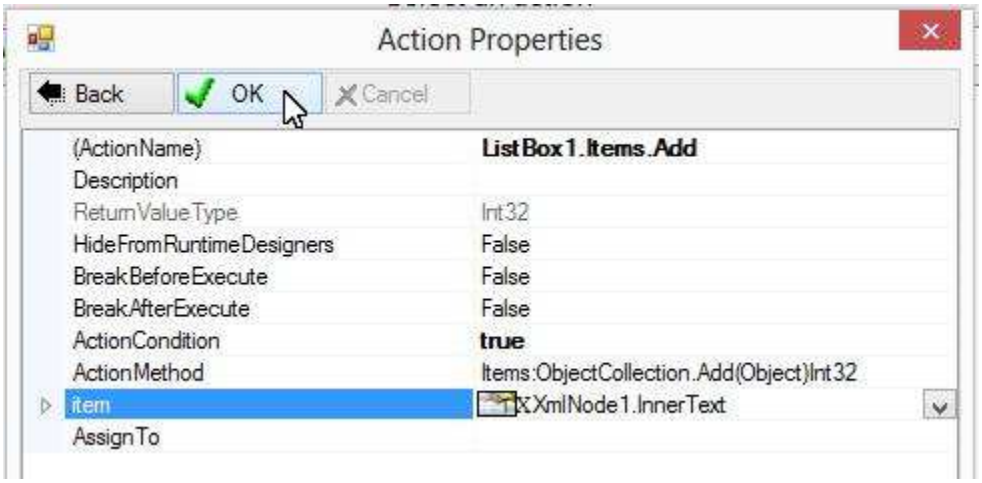
The “item” of the Add action represents the object to be added to the list box. In this sample, we may add the InnerText property of the <caddr> node to the list box:



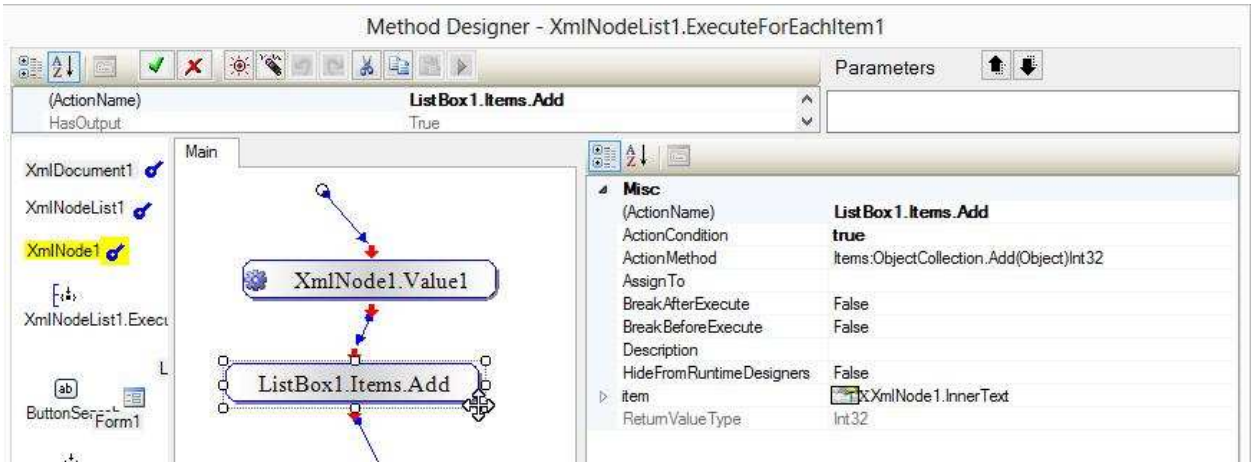
Select the InnerText property of the variable XmlNode1:



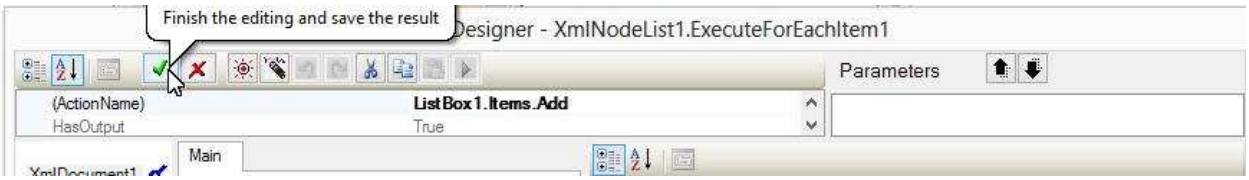
Click OK to finish creating the action:



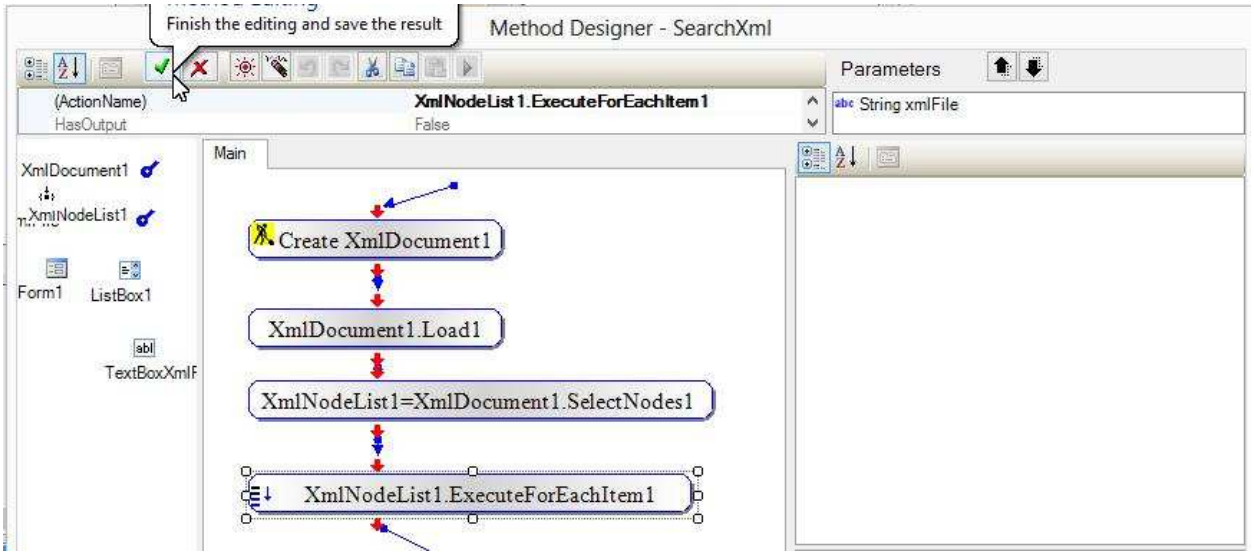
The action appears in the Action Pane. Link it to the last action:



For this sample, that is all we need to do for this <caddr> node:

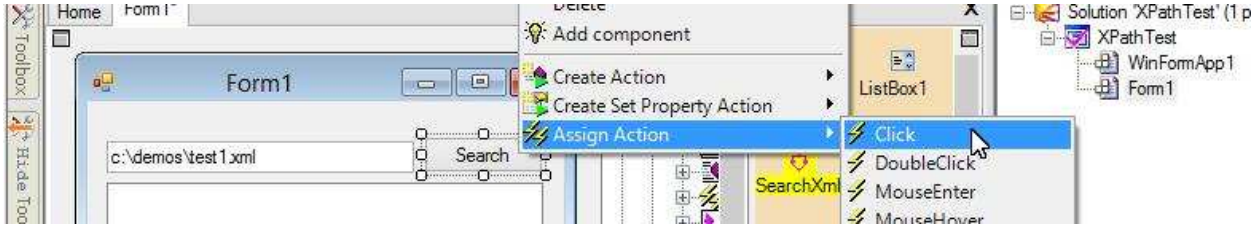


That is all for this method:

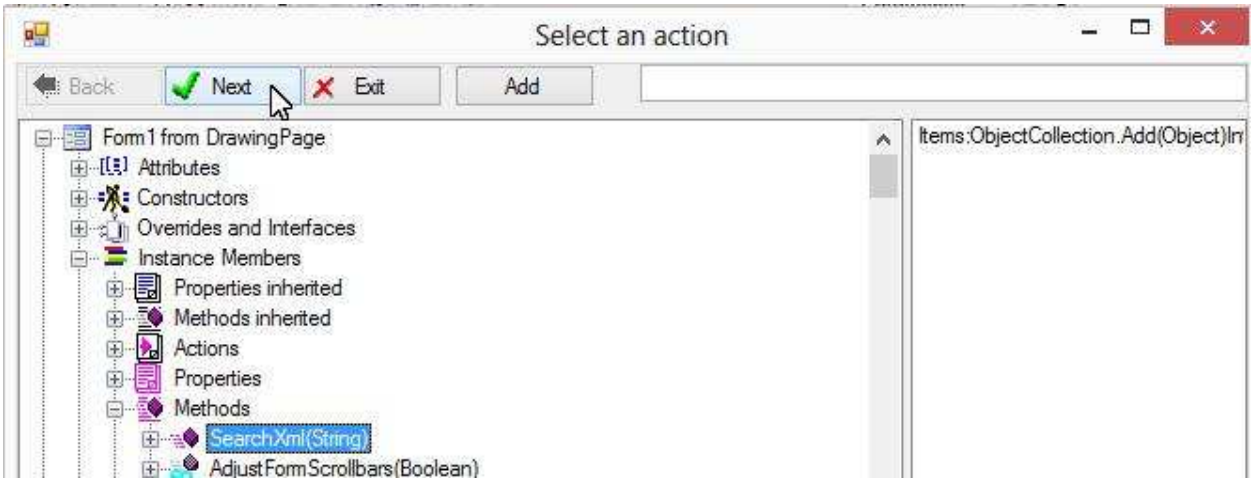


Execute method

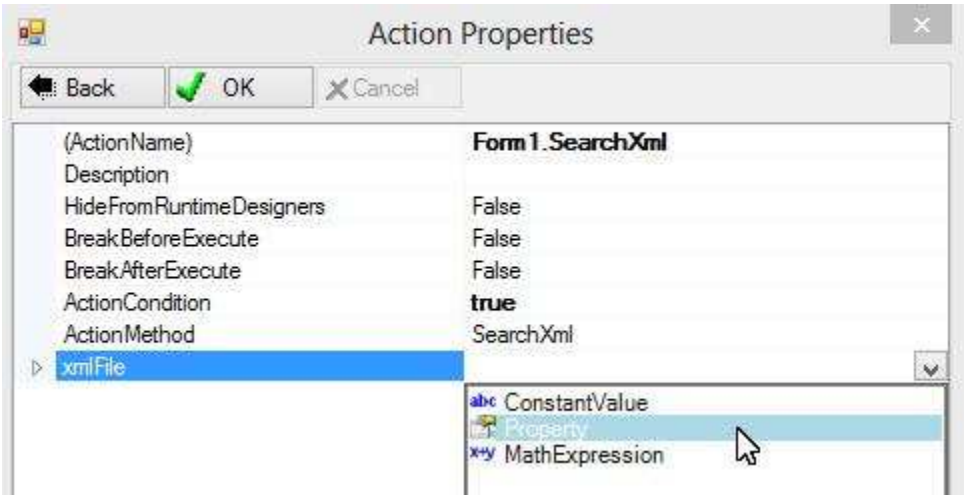
We use a button to trigger the execution of the method:



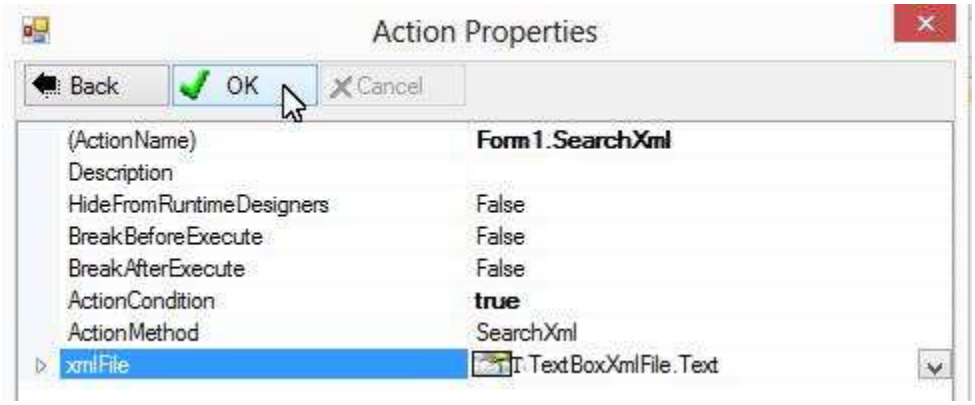
Select the method we just created:



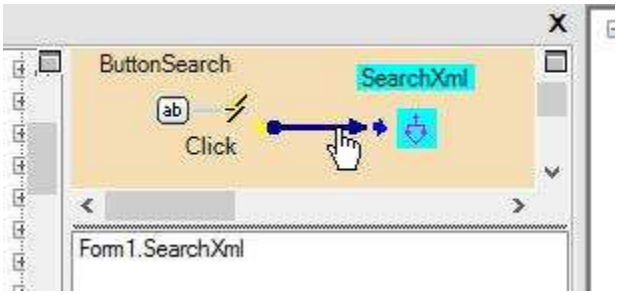
Use the text box for the action parameter:



Click OK to finish creating the action:

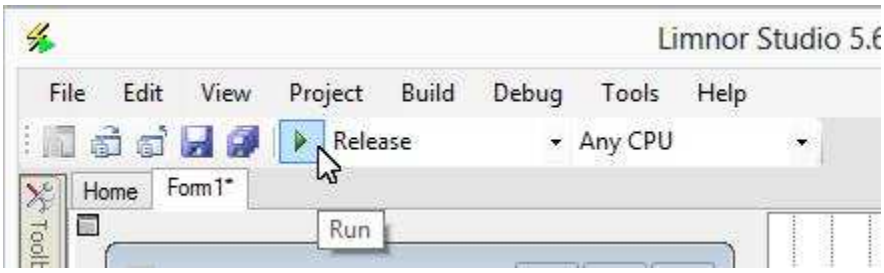


The action is created and assigned to the button:

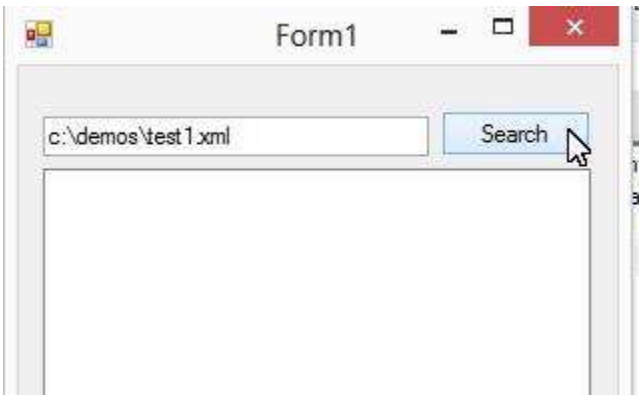


Test

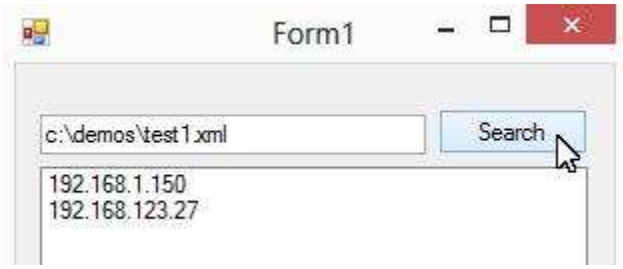
Click the Run button to test the project:



The form appears. Click the button:



We can see two IP addresses are found:



Feedback

Please send your feedback to support@limnor.com.