

Text File IO

User Guide - Read/Write Text File

Limnor ships with a TextFileWriter performer. Your applications may use it to write text to files, for example, write log files.

Limnor ships with a TextFileReader performer. Your applications may use it to read text from files, for example, show log file contents.

If you want to use a text file as a configuration file for your application, you may use another performer, Config, which is more specialized on that purpose. This sample application shows read/write text files with TextFileReader and TextFileWriter performers.

[Write Text File](#)

[Handling Events](#)

[Read Text File](#)

[Handling Events](#)

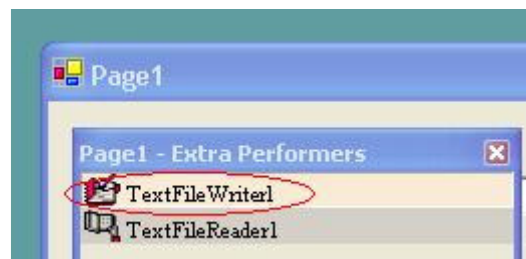
[Handling AppendCRLF Property](#)


Write Text File

Drop a TextFileWriter performer to a page:

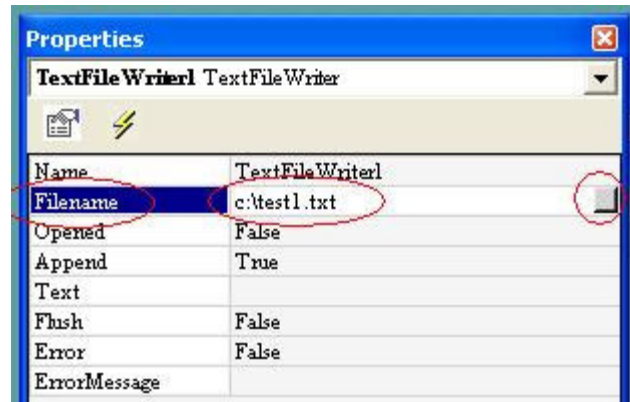


Because the TextFileWriter does not have a graphic user interface on the page, it appears in the "Extra Performers" window for that page:



Right-click it and open its properties window. Choose "Filename" property and click button  to set the file to be written. You may also simply type in the full file path and name;

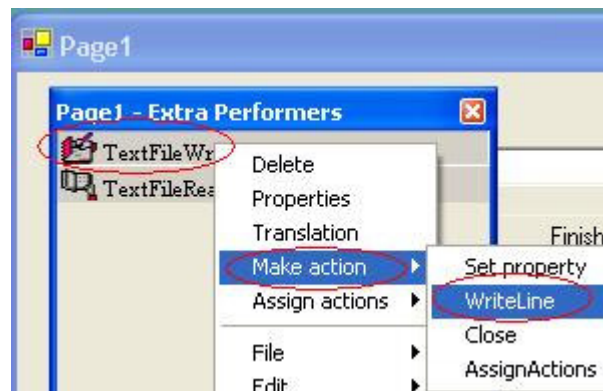
Text File IO



If you set "Append" property to True then text will be append to the end of the file if the file exists. If Append property is False then every time your application re-start writing (call WriteLine method after calling Close method), the existing file will be overwritten.

The "Text" property is the text to be written by WriteLine method. The typical usage is to set the Text property, and call WriteLine; set Text property again and call WriteLine again; and so on.

Now let's make an action using the WriteLine method. Right-click on the TextFileWriter performer, choose "Make Action", choose "WriteLine":

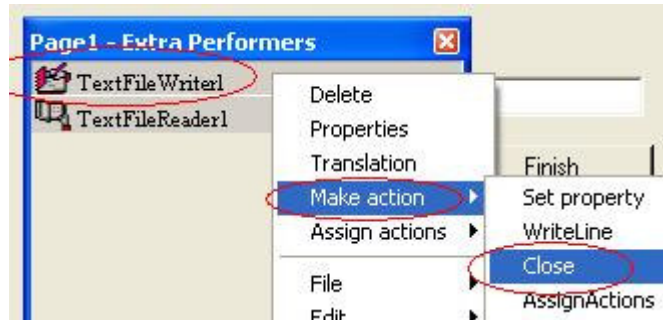


Accept the default action name TextFileWriter1.WriteLine:

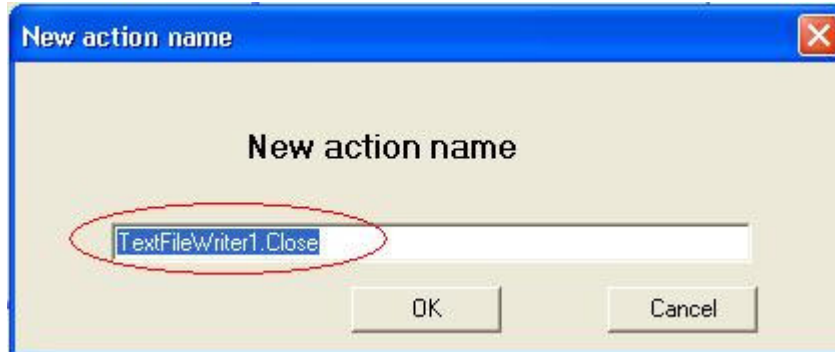


Let's make another action to finish the writing. Right-click on the TextFileWriter performer, choose "Make Action", choose "Close":

Text File IO



Accept the default action name, TextFileWriter1.Close:

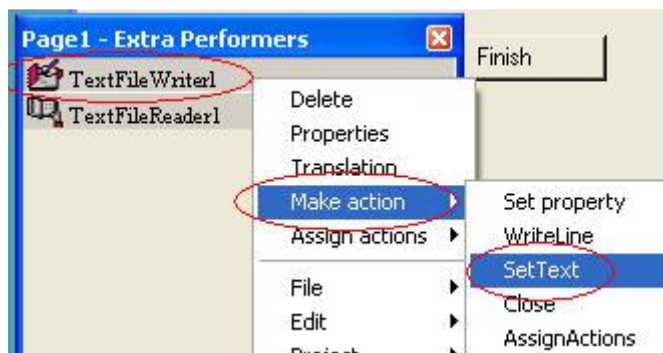


Calling the "Close" method will release the computer resources taken by the TextFileWriter performer. So at the time when your application no longer needs to write to a file, call "Close" method to free the resources taken by it. Once "Close" is called, the next time calling of WriteLine method will overwrite the existing file if "Append" property if False.

Now the question: what do we want to write? For this sample application, we use a TextBox performer to let the user type in what the user wants to write to the file. The user may type in some text, click a button to write the text to the file:



We need to make an action to copy the Text property of the TextBox performer to the TextFileWriter performer. Right-click on the TextFileWriter performer, "Make Action", choose "SetText":

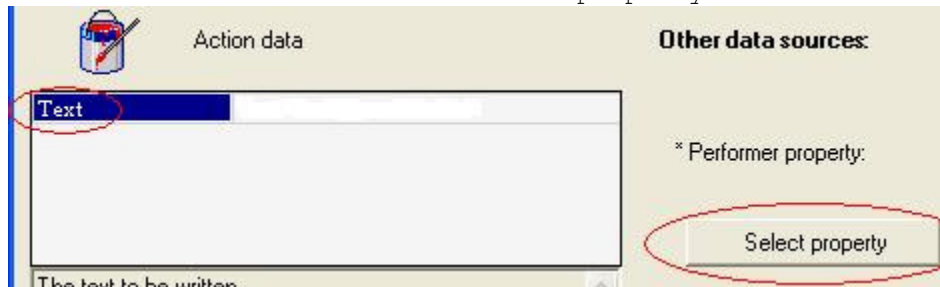


Accept the default action name, TextFileWriter1.SetText:

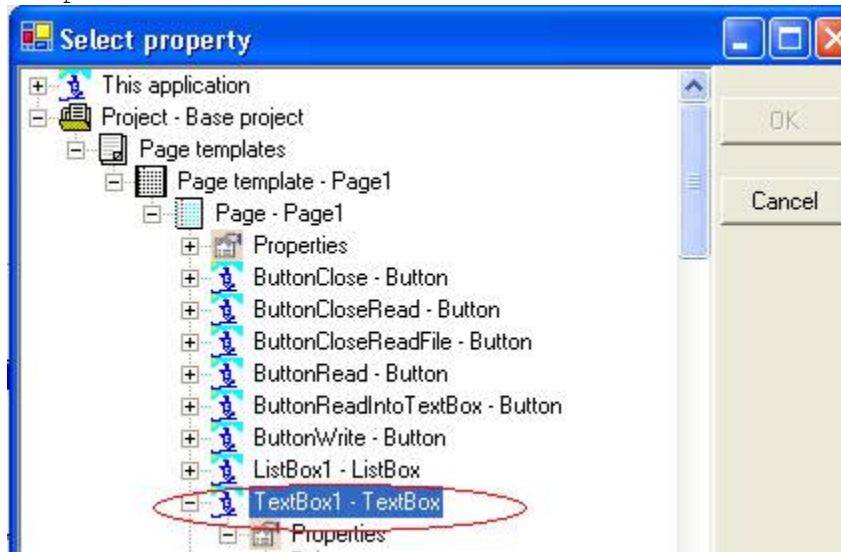
Text File IO



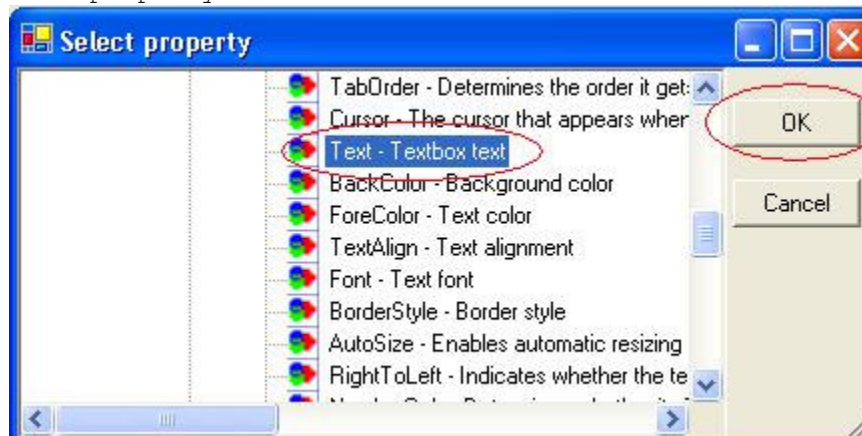
It asks for the text to be set. Click "Select property" button:



Find the TextBox performer:

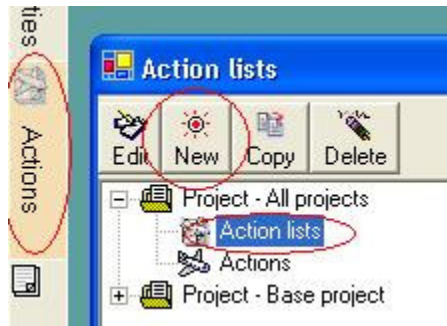


Select its "Text" property and click OK:



Text File IO

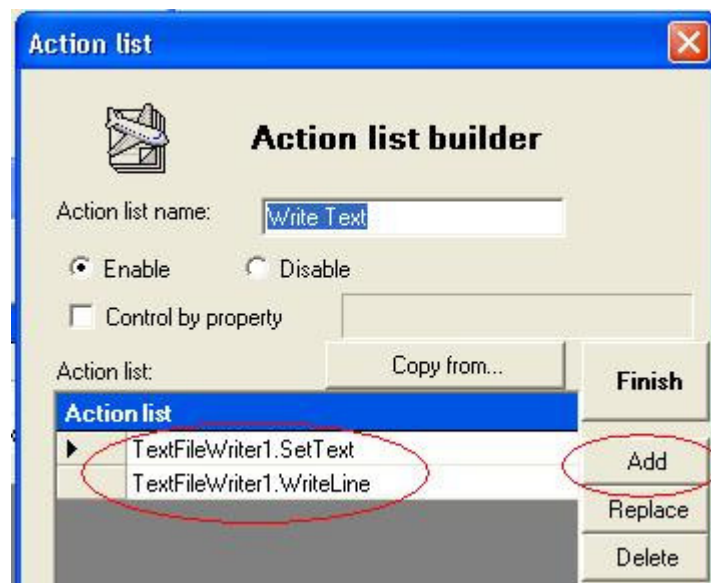
Now we have an action to prepare the text to be written and an action to actually write the text. We want to create an action list to include these two actions, and assign this action list to a button. Open "Action lists" window, select "Action lists", click "New":



Give an action list name, say, "Write Text":

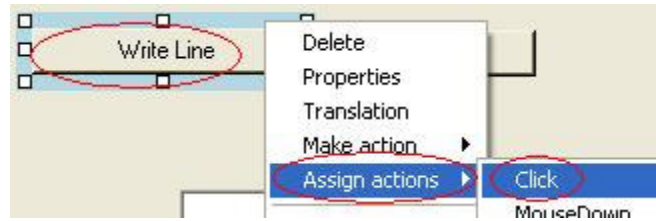


Use the Add button to add the two actions:

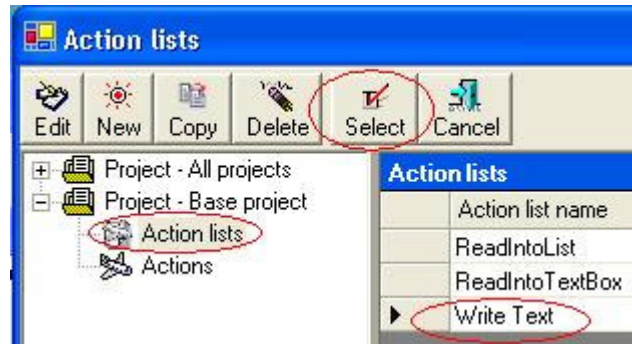


Now, assign this new action list to the "Write Line" button:

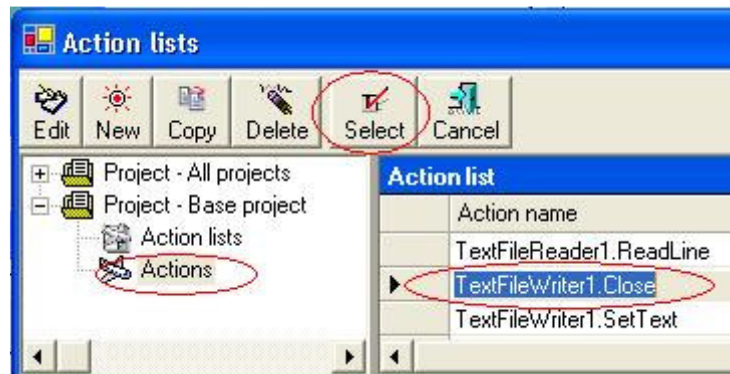
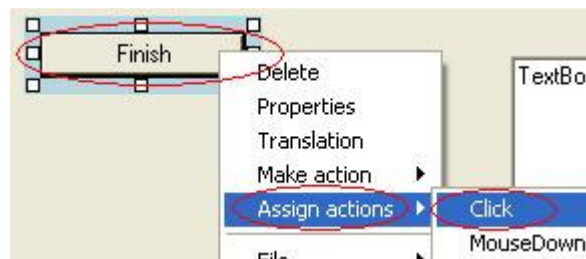
Text File IO



Select "Action lists", then select "Write text", click "Select" button:



We'll use a button to let the user to finish the file writing:



We are basically done. Run the program, type in something in the text box, click "Write line" button; type something in the text box again and click the button again; and so on. Click "Finish" button to close the file. Then you may use notepad to open the file to see what you typed has been saved in the file.

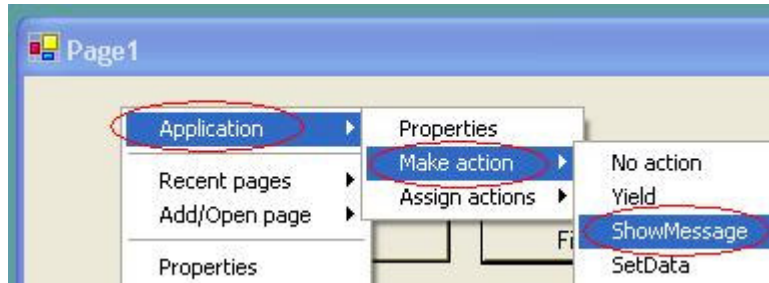
Handling Events

The `TextFileWriter` performer will fire two types of events. "Open" event is fired when the first time "WriteLine" method is called to write to a file, that is, the file is opened. "Error" event is fired when something wrong in accessing the file.

Text File IO

We'll demonstrate these two events by simply show a message box at these two events.

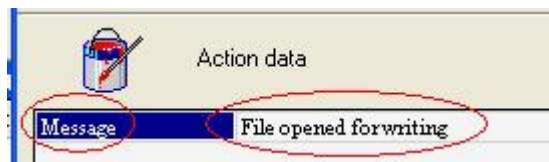
First, let's make an action to show a message when file is opened. Right-click on the page, choose "Application", choose "Make Action", choose "ShowMessage":



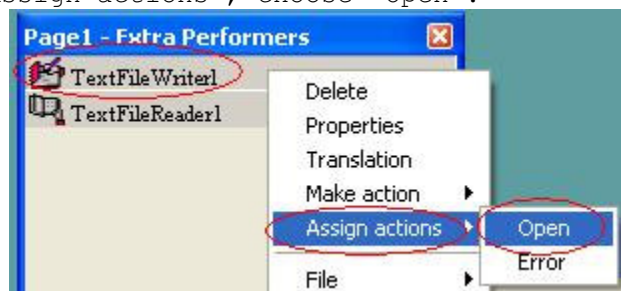
Give an action name:



For the Message, type in "File opened for writing" (you may type in anything you want):

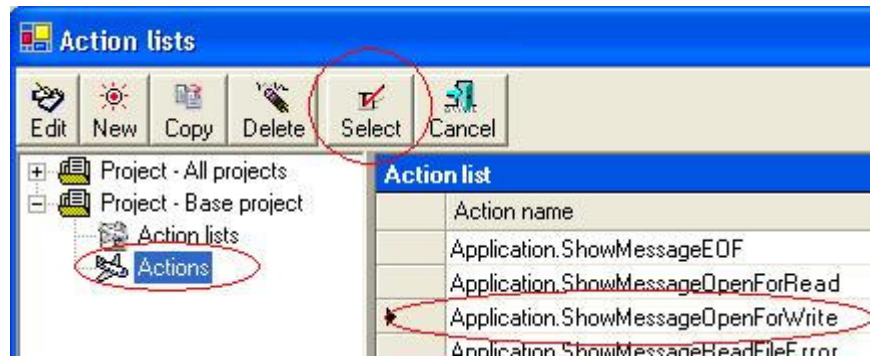


Now we assign this action to the Open event. Right-click the TextFileWriter performer, choose "Assign actions", choose "Open":



Select "Actions", select the action name, click "Select" button:

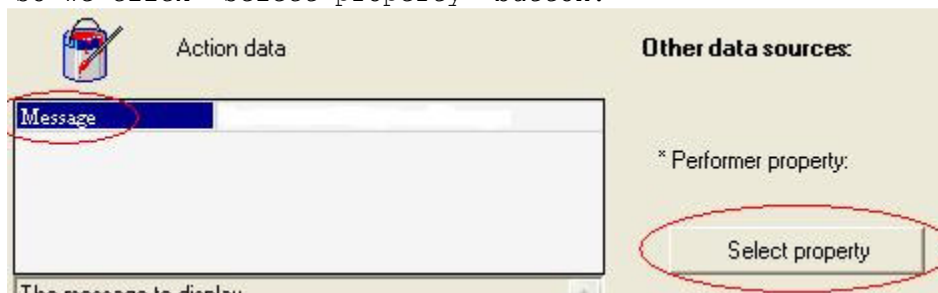
Text File IO



In the same way, we make an action to show error message:

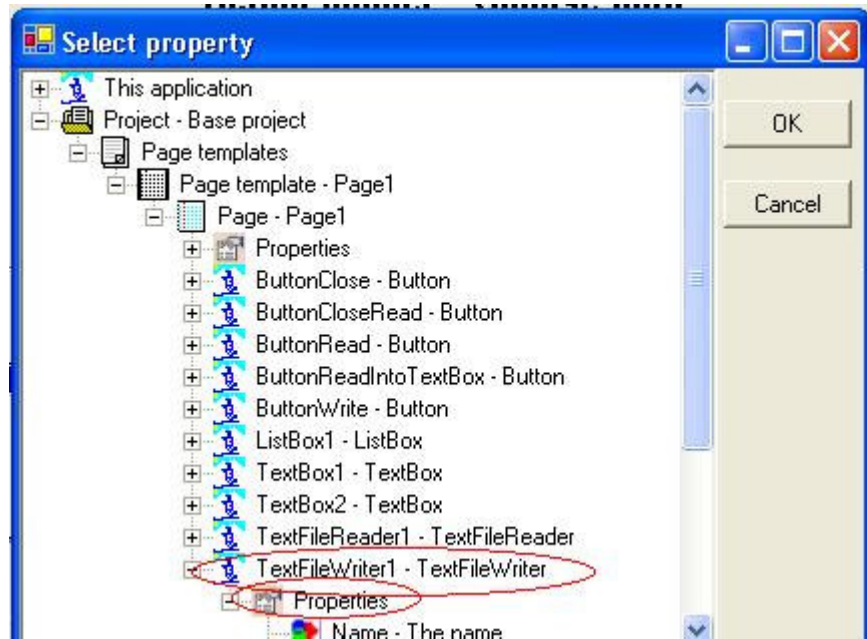


For Message, we want to use the ErrorMessage Property from the TextFileWriter performer. So we click "Select property" button:

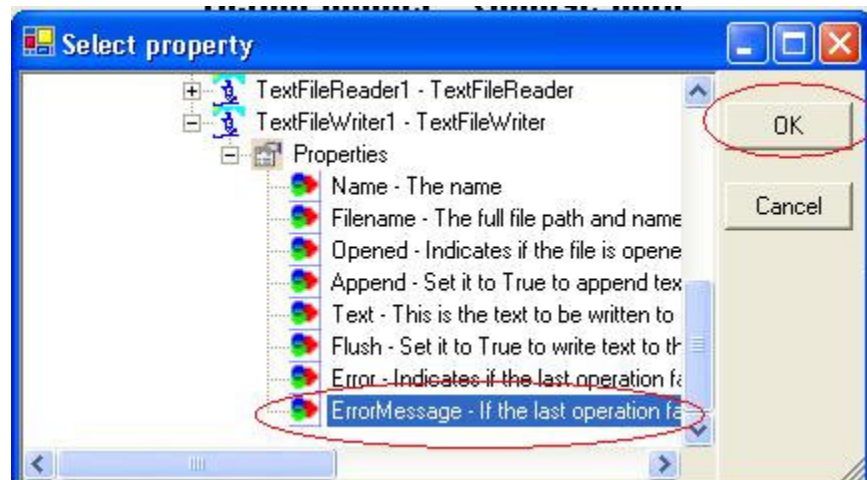


Find the TextFileWriter performer:

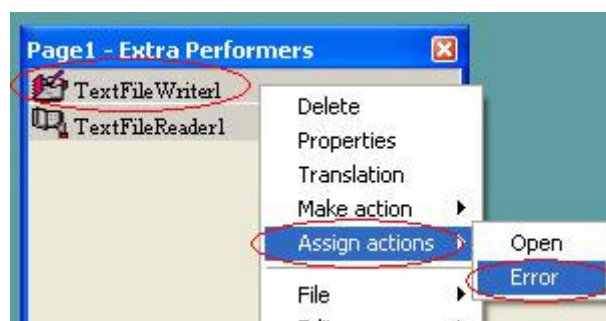
Text File IO



Choose its ErrorMessage Property:

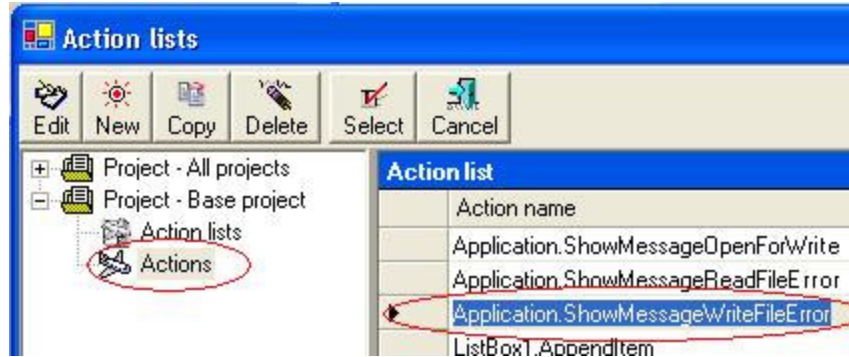


Now we assign this action to the "Error" event. Right-click the TextFileWriter performer, choose "Assign actions", choose "Error":



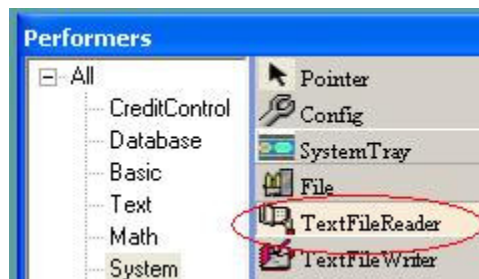
Select "Actions", select the action name, click "Select" button:

Text File IO

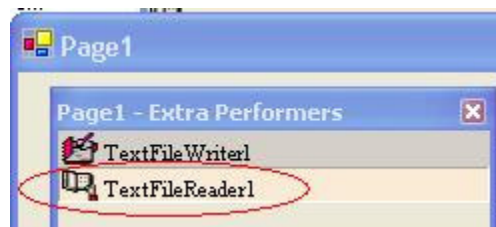



Read Text File

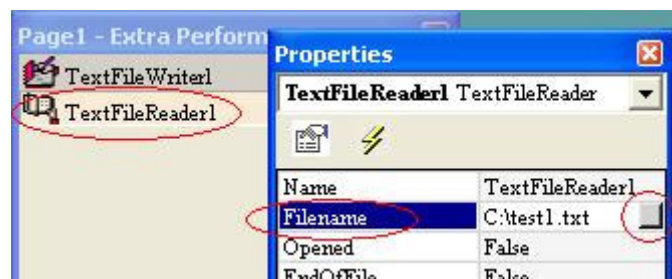
Drop a TextFileReader performer to a page:



Because the TextFileReader does not have a graphic user interface on the page, it appears in the "Extra Performers" window for that page:



Right-click it and open its properties window. Choose "Filename" property and click button  to set the file to be written. You may also simply type in the full file path and name;

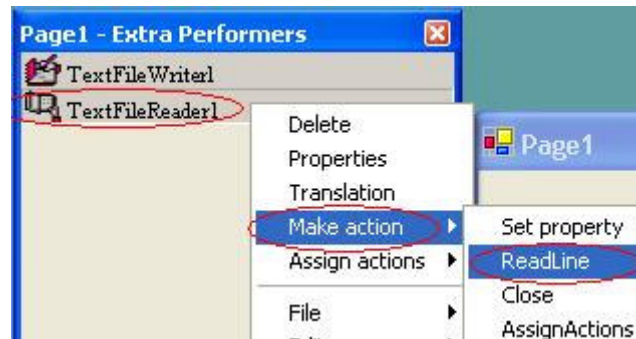


We need to decide where to put the text read from the file. Here we put each line of text to a list box. Drop a list box performer to the page:

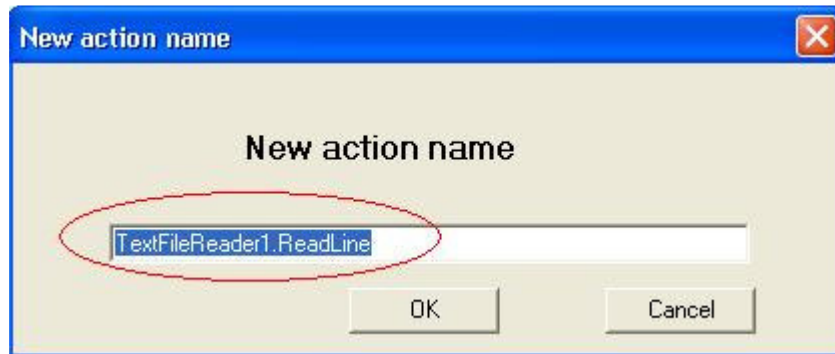
Text File IO



We use "ReadLine" method of the TextFileReader performer to make an action to read file. Right-click on TextFileReader, choose "Make action", choose "ReadLine":

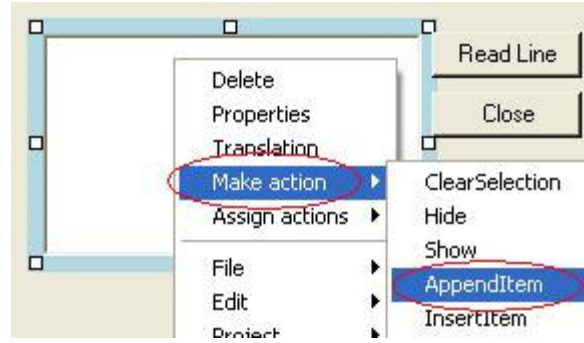


Accept the default action name:



The "Text" property of the TextFileReader is the text read from the file. We append it to the list box. Right-click on the list box, choose "Make action", choose "AppendItem":

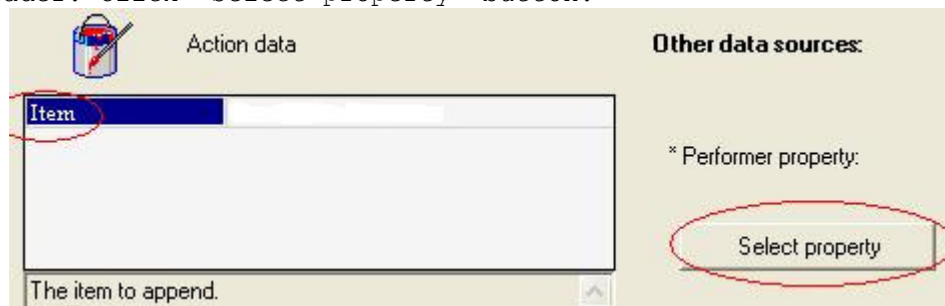
Text File IO



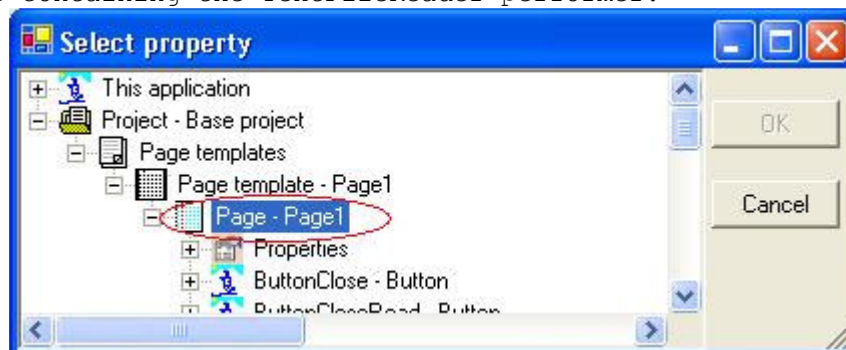
Accept the default action name:



It asks you what to append. We want to use the Text property of the TextFileReader. Click "Select property" button:



Find the page containing the TextFileReader performer:

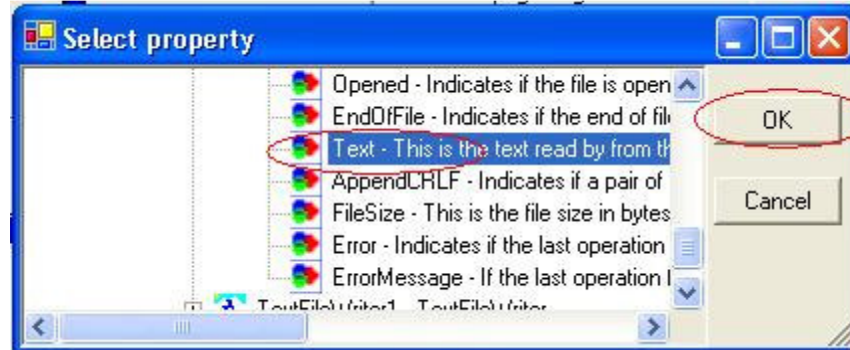


Find the TextFileReader performer:

Text File IO



Find and select its Text property and click OK:

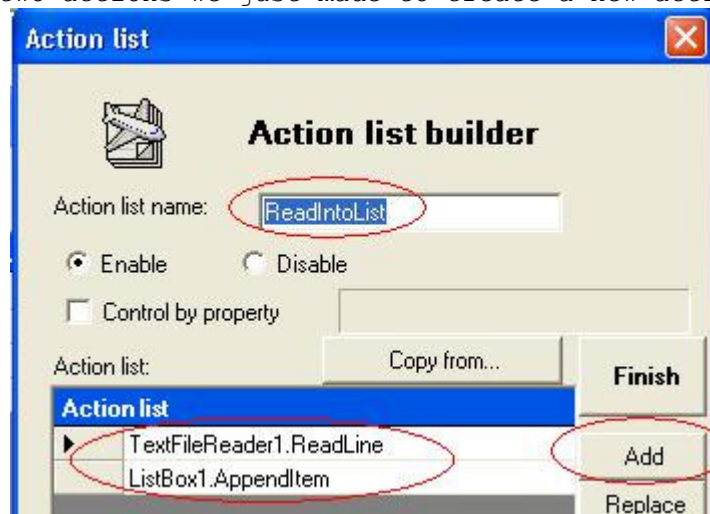


One more thing to consider is that when ReadLine method hits the end of file, we should not let this action to be carried out. We choose "disable" and check "Control by property", and select the "EndOfFile" property of the TextFileReader:



When the ReadLine method hits the end of file, the EndOfFile property becomes True, and thus disables this action so that there is not an item appended to the list box.

We use the above two actions we just made to create a new action list:



And assign this action list to a button's Click event:

Text File IO



As the `TextFileWriter`, `TextFileReader` also has a "Close" method. We can use it to make an action and assign it to a button's Click event:

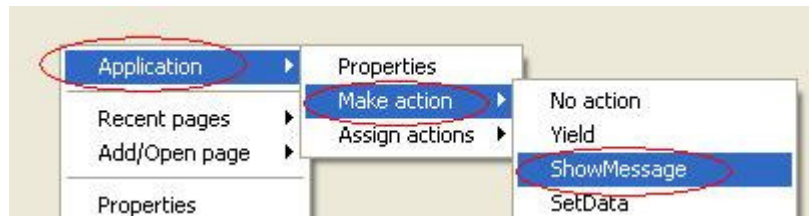


We are basically done. Keep clicking the "Read Line" button, each line of the text file will appear in the list box.

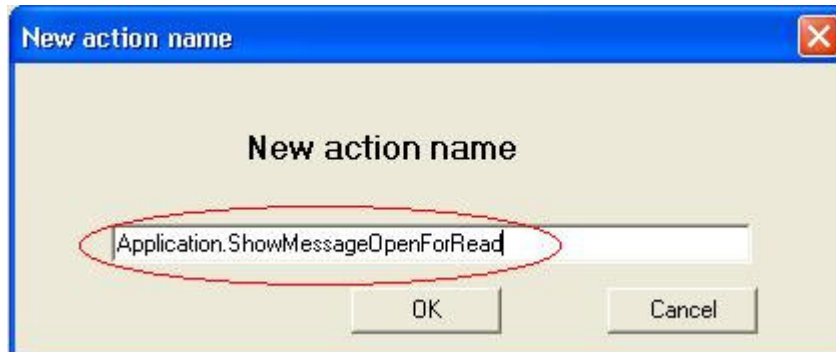
Handling Events

The `TextFileReader` performer will fire 3 types of events. "Open" event is fired when the first time "ReadLine" method is called to read from a file, that is, the file is opened. "Error" event is fired when something wrong in accessing the file. "EndOfFile" event is fired when "ReadLine" method hits the end of file. We'll demonstrate these two events by simply show a message at these 3 events.

First, let's make an action to show a message when file is opened. Right-click on the page, choose "Application", choose "Make Action", choose "ShowMessage":

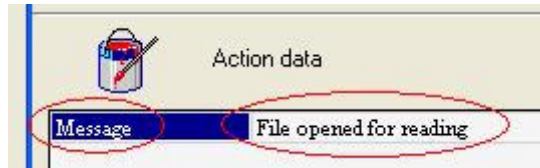


Give an action name:

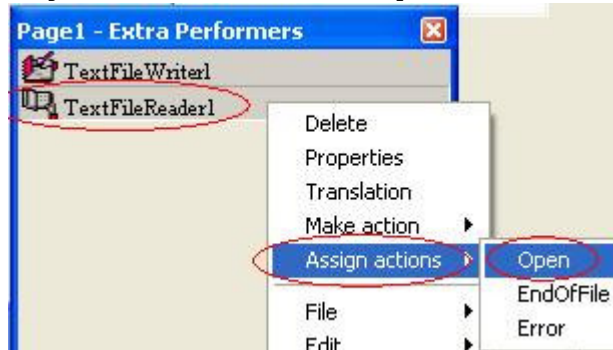


Type in the message we want to show:

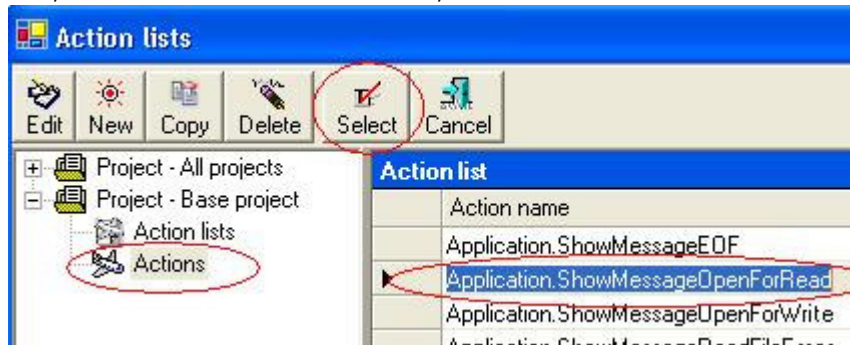
Text File IO



Now we assign this action to the Open event. Right-click the TextFileReader performer, choose "Assign actions", choose "Open":



Select "Actions", select the action name, click "Select" button:

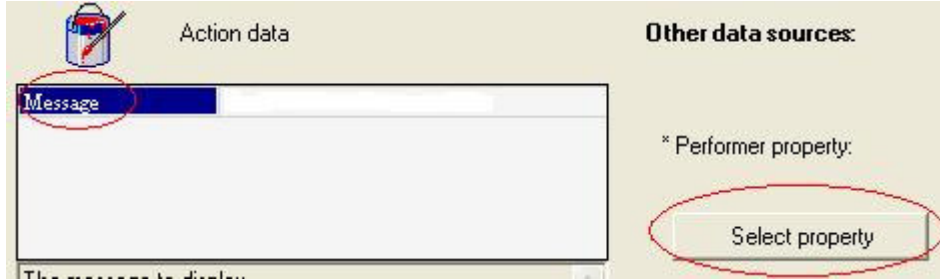


In the same way, we make an action to show error message:

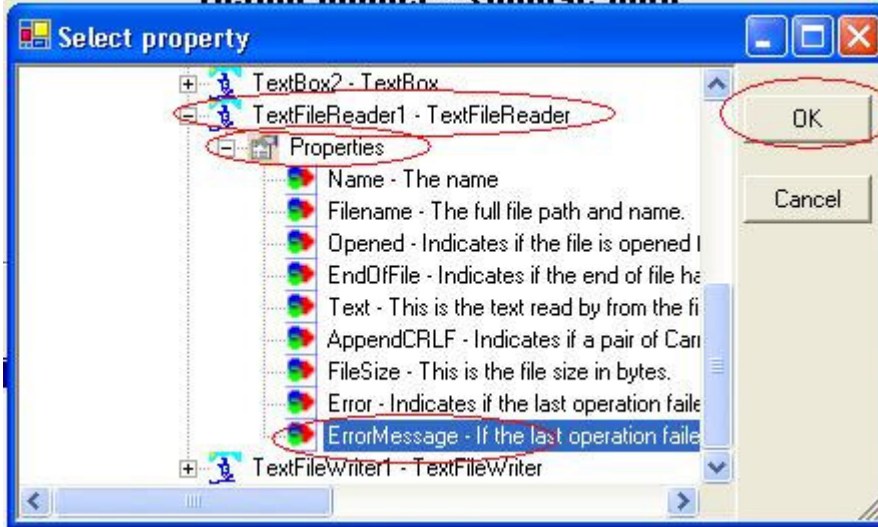


For Message, we want to use the ErrorMessage Property from the TextFileReader performer. So we click "Select property" button:

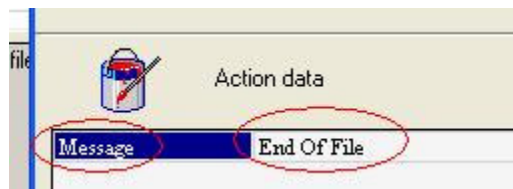
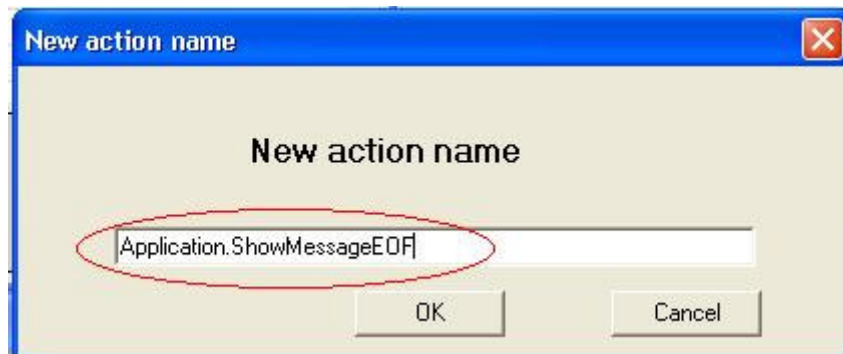
Text File IO



Find the TextFileReader performer, choose its ErrorMessage Property:

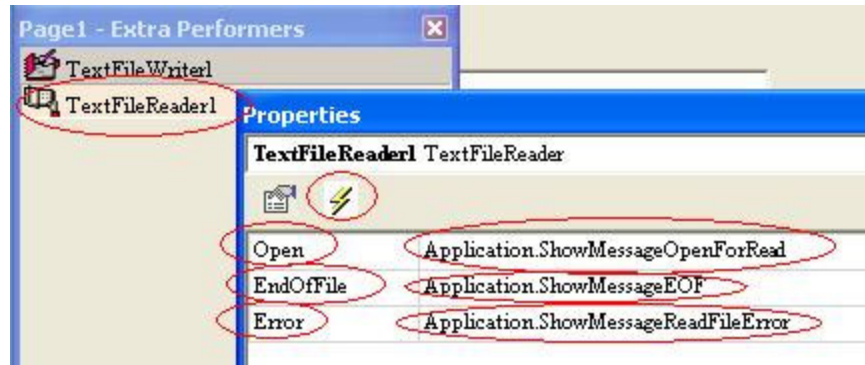


In the same way, we make an action to show "End of file" message:



We assign these 3 files to "Open", "EndOfFile" ,and "Error" events respectively:

Text File IO

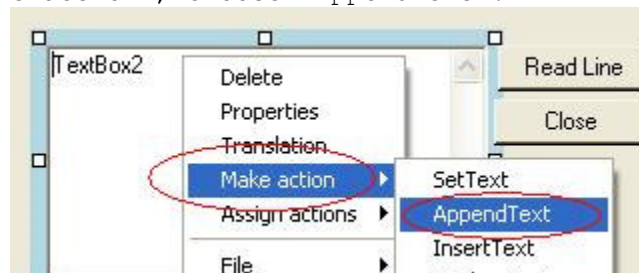


Handling AppendCRLF Property

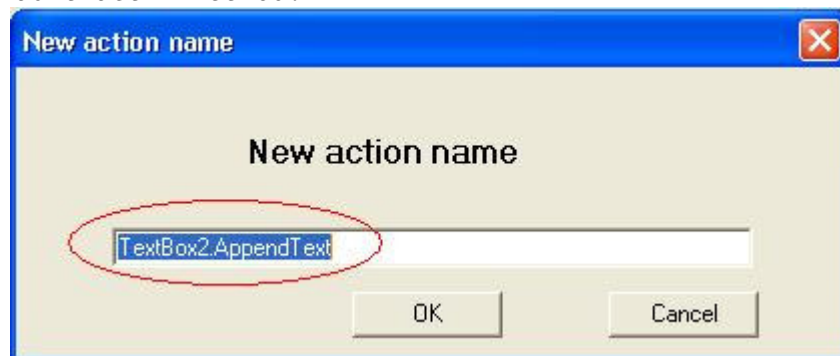
Suppose we use a TextBox performer to show text read from the file:



A TextBox performer has an "AppendText" method. We use it to append the Text property of the TextFileReader performer to the text box. Right-click on the TextBox, choose "Make action", choose "AppendText":

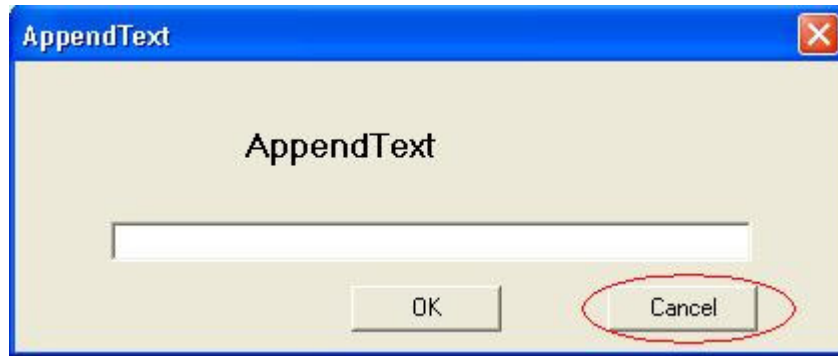


Accept the default action method:

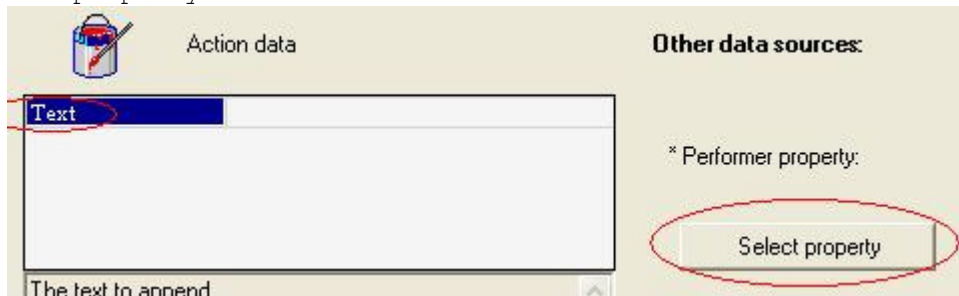


It asks for the text to be appended. Click "Cancel" because we are not using a constant text. We are going to use a property:

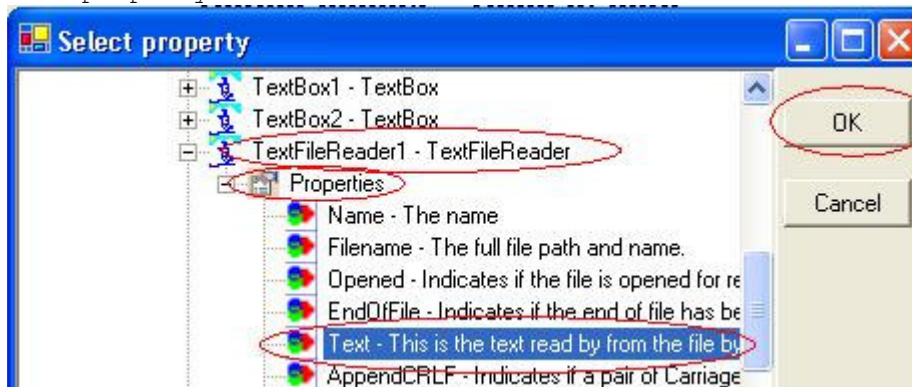
Text File IO



Click "Select property" button:



Select the Text property of the TextFileReader and click OK:

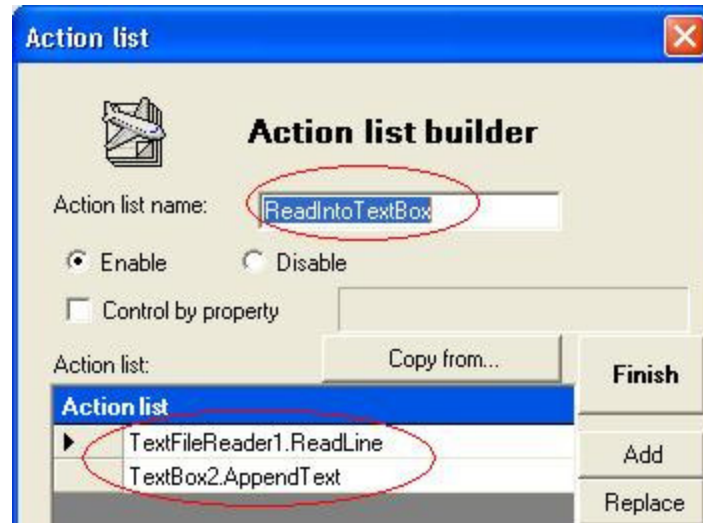


Like for appending to the list box, we also disable this action if the end of file is hit:

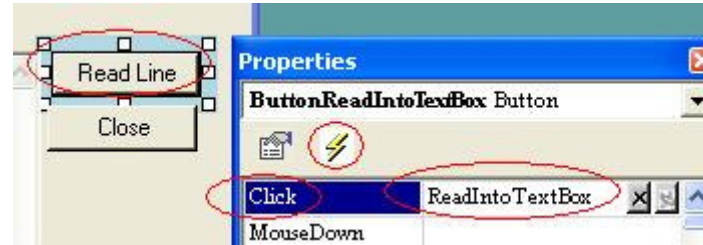


We make an action list to read a line of text from file and append it to the text box:

Text File IO



And assign this action list to a button's Click event:



Run the application and click the button several times, each time a new piece of text appended to the end of text box. You may notice that all pieces of text are not in separate lines, but in one paragraph. You may set "AppendCRLF" property to True to append line terminators to the text read from the file:



Now run the application again, click the button several times, you will see each piece of text is in a separate line.

=== EOF ===