NDDE Samples

Contents

Introduction

DDE, Dynamic Data Exchange, is an old technology for data exchange. NDDE is a library to make DDE available for .Net Framework programming. The NDDE open source library can be downloaded from http://ndde.codeplex.com/.

This document shows the re-creation of the NDDE Server and Client samples in Limnor Studio. The samples can be found at the folder Samples\cs under the NDDE installation folder.

The samples use callback functions and use constructors with parameters.

Work with callback function

The NDDE uses callback functions to support asynchronous operations. When creating an asynchronous action, we provide a method as a parameter value of the action. When an asynchronous action is executed, it appears that the action finishes immediately. But the real operation the action starts may still go on at background. When the real operation finishes the callback function is executed.

The DDE Client sample shows the creating and using of the callback functions.

Work with constructors

An instance of an object must be created from a “class”. A class must have one or more constructors in order for it to create instances.

A constructor is an initialization event. It executes actions at the time an object is constructed. The event may or may not need parameters.

Some classes, for example, MessageBox, do not have constructors and thus cannot create object instances.

When we use a constructor to create an object instance, if the constructor requires parameters then we must provide values for those parameters. If a class provides more than one constructor then we must choose one of the constructors to be used.

Most classes are built with a parameter-less constructor. Many classes are built with only a parameter-less constructor. Therefore we do not have to choose constructors and we do not need to provide constructor parameter values. Thus we do not need to deal with the concept of "constructor" when creating object instances.

For example, the Toolbox in Microsoft Visual Studio only accepts classes built with parameter-less constructors. It further requires that the classes must implement IComponent interface. Thus, the majority of the classes cannot be put in the Microsoft Visual Studio Toolbox.

The Toolbox in Limnor Studio does not require the classes to implement IComponent interface. It accepts any classes with any constructors. Therefore in some cases we have to deal with the concept of "constructors".

The NDDE samples show how to work with constructors:

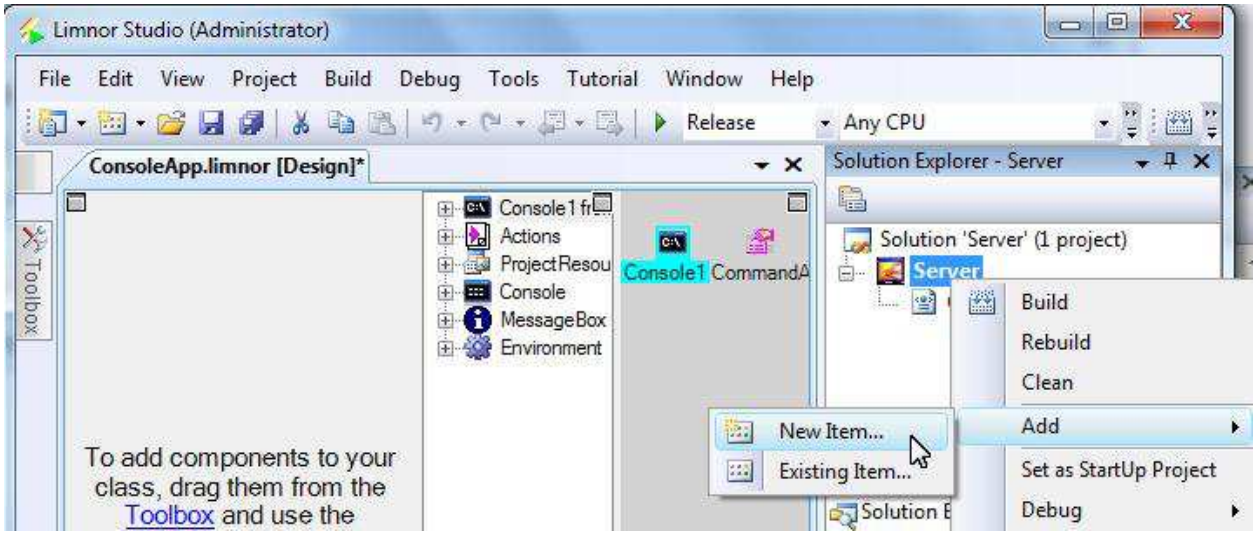
- Create constructors
- Select constructors
- Use constructors

Create DDE Server

The NDDE provides a class named DdeServer acting as a data server for DDE. This is an abstract class. An abstract class cannot be used to create object instances. The designed way of using DdeServer, according to the NDDE sample, is to derive a new class from it. We will re-create the Server sample in Limnor Studio.

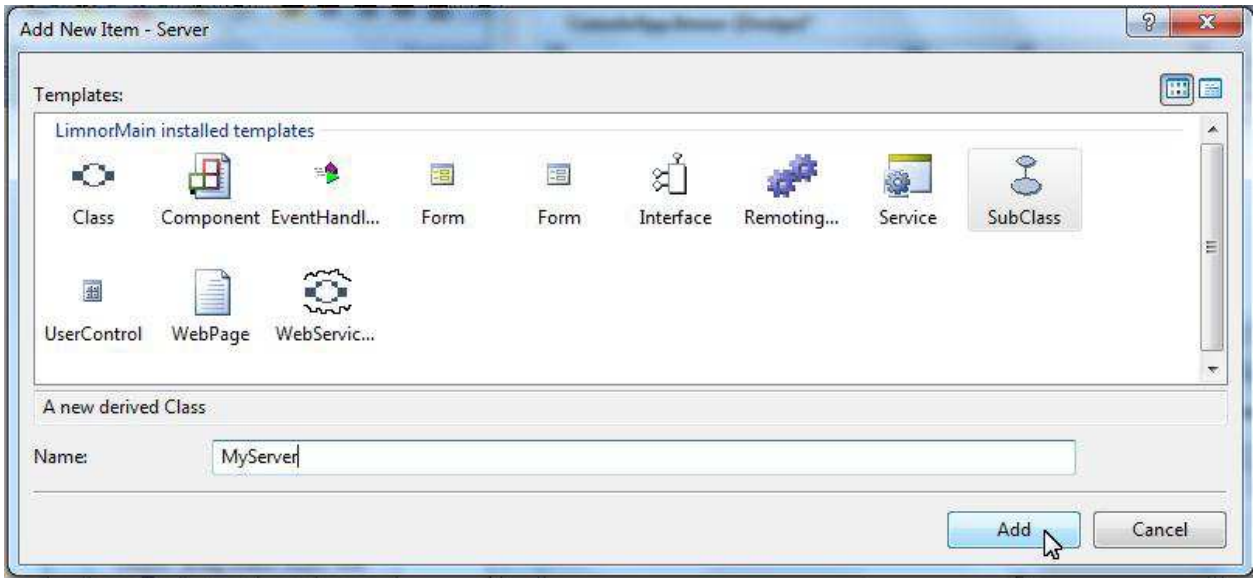
Create a new class

Create a console project. Add a new class to the project.

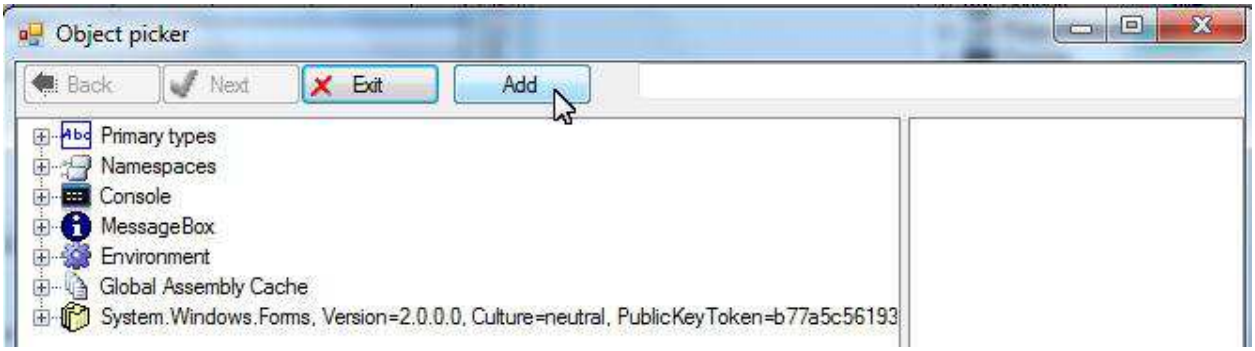


Name the new class MyServer.

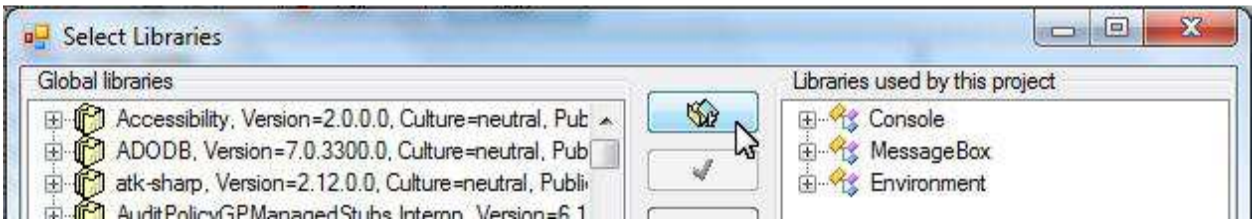
A new class must be derived from an existing class. Limnor Studio provides a list of common base classes for us to select from. If the base class we want to use is not among the list then select “SubClass”. It will let us choose a class from libraries.

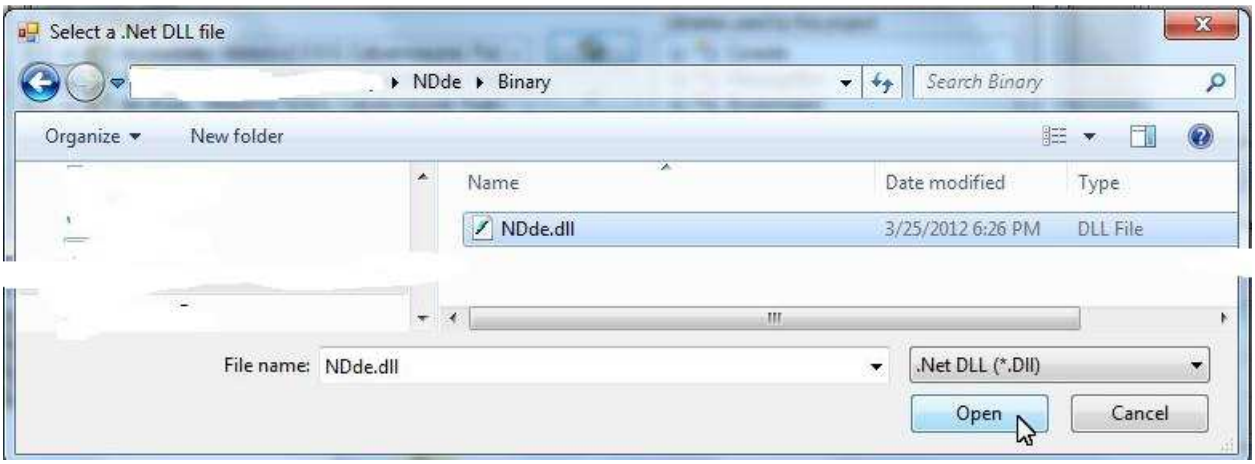


Limnor Studio lists some common places for us to choose a base class. Click Add button because the NDDE library is not among the common places:

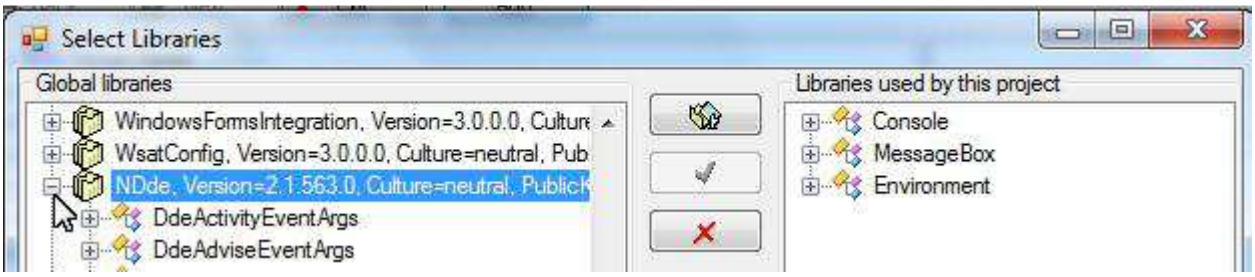


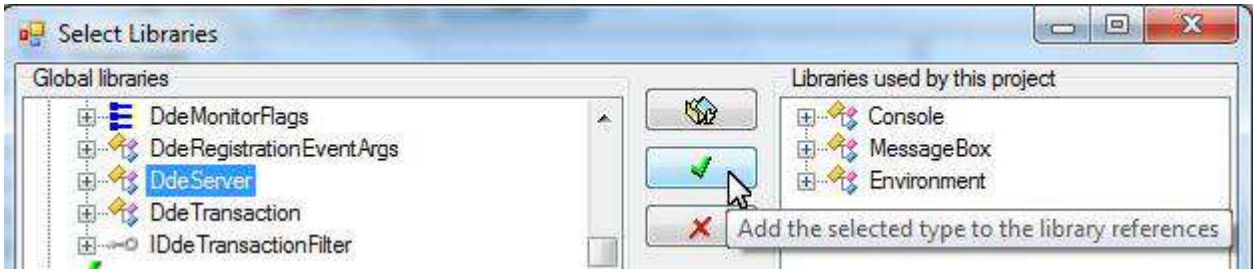
Click  to load NDDE library.

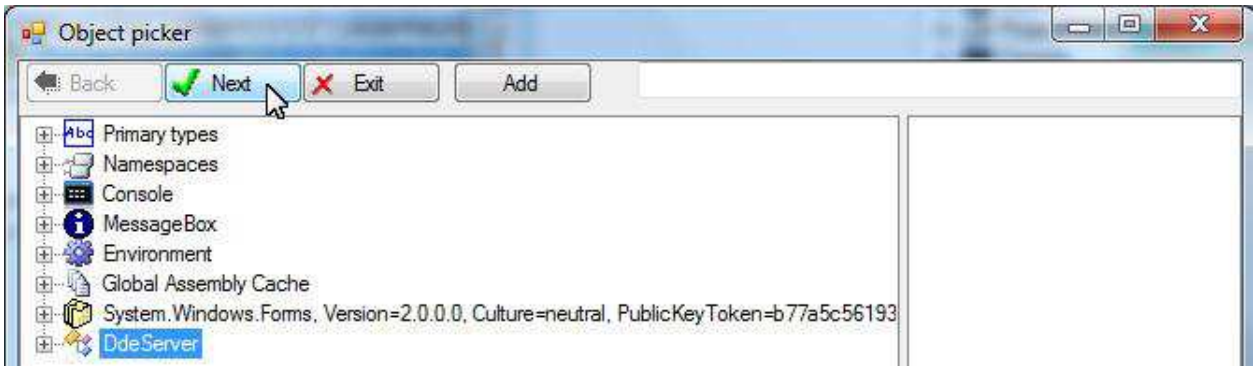




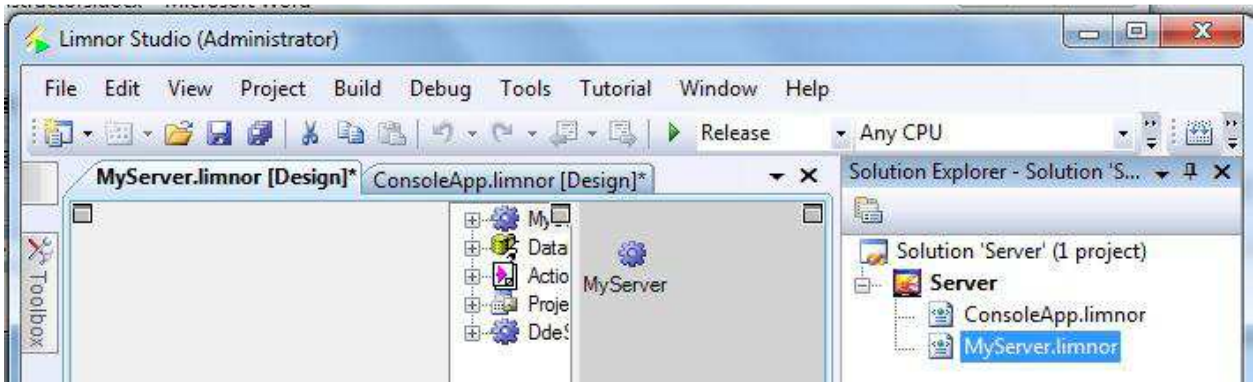
Select class DdeServer:







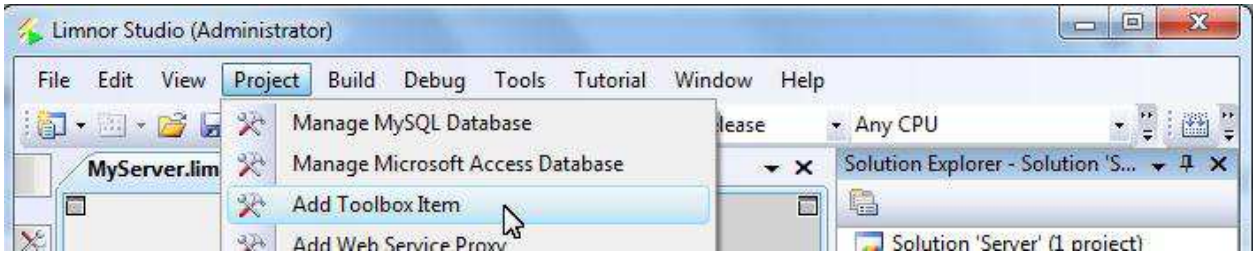
A new class named MyServer is created in the project:



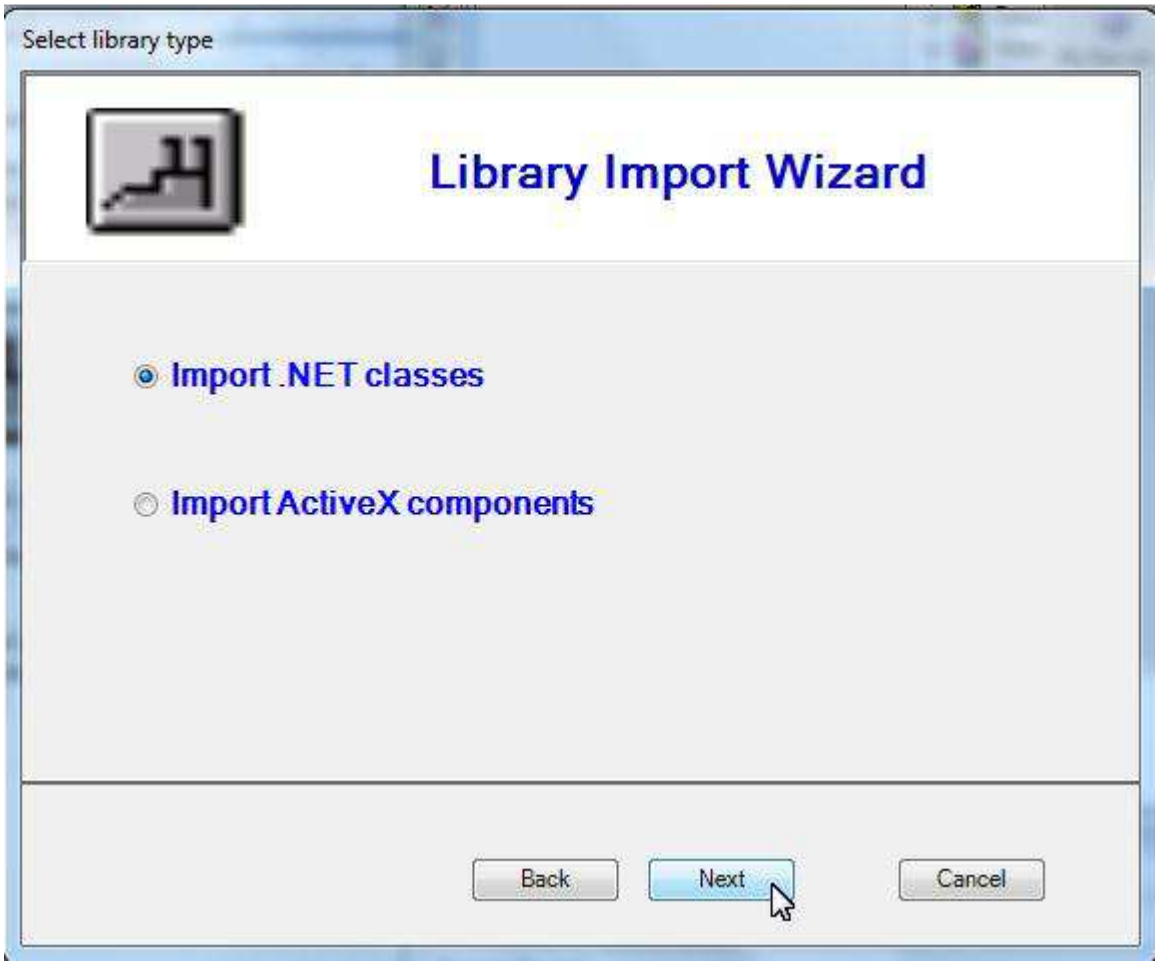
Add components to class

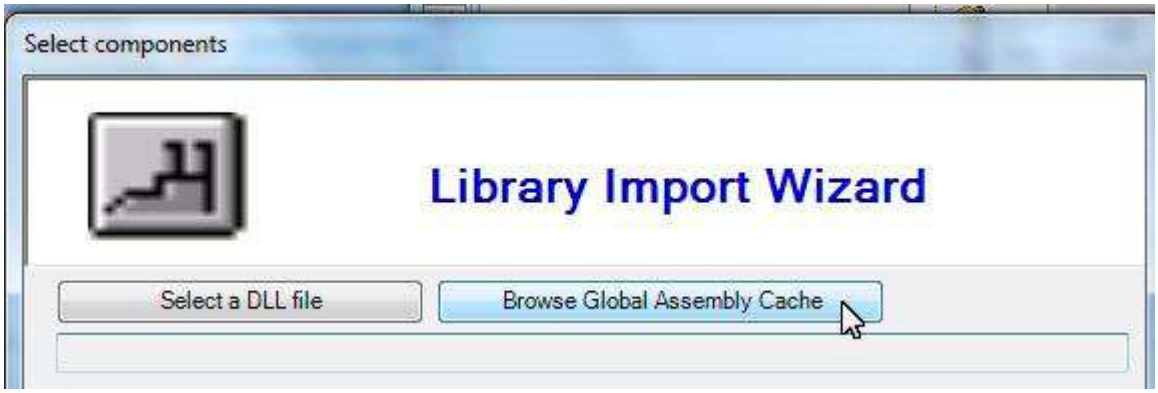
The sample DDE server uses a timer to issue DDE advice periodically. Add a timer component to the class.

Note that the Timer component in the Toolbox is only for Forms. Do not use it here. Let's add the timer from the System namespace to our Toolbox.

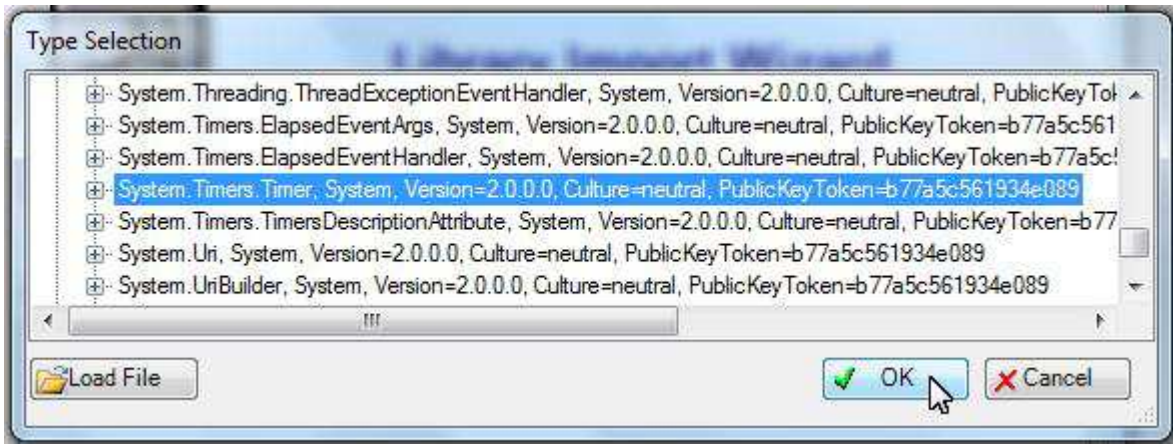


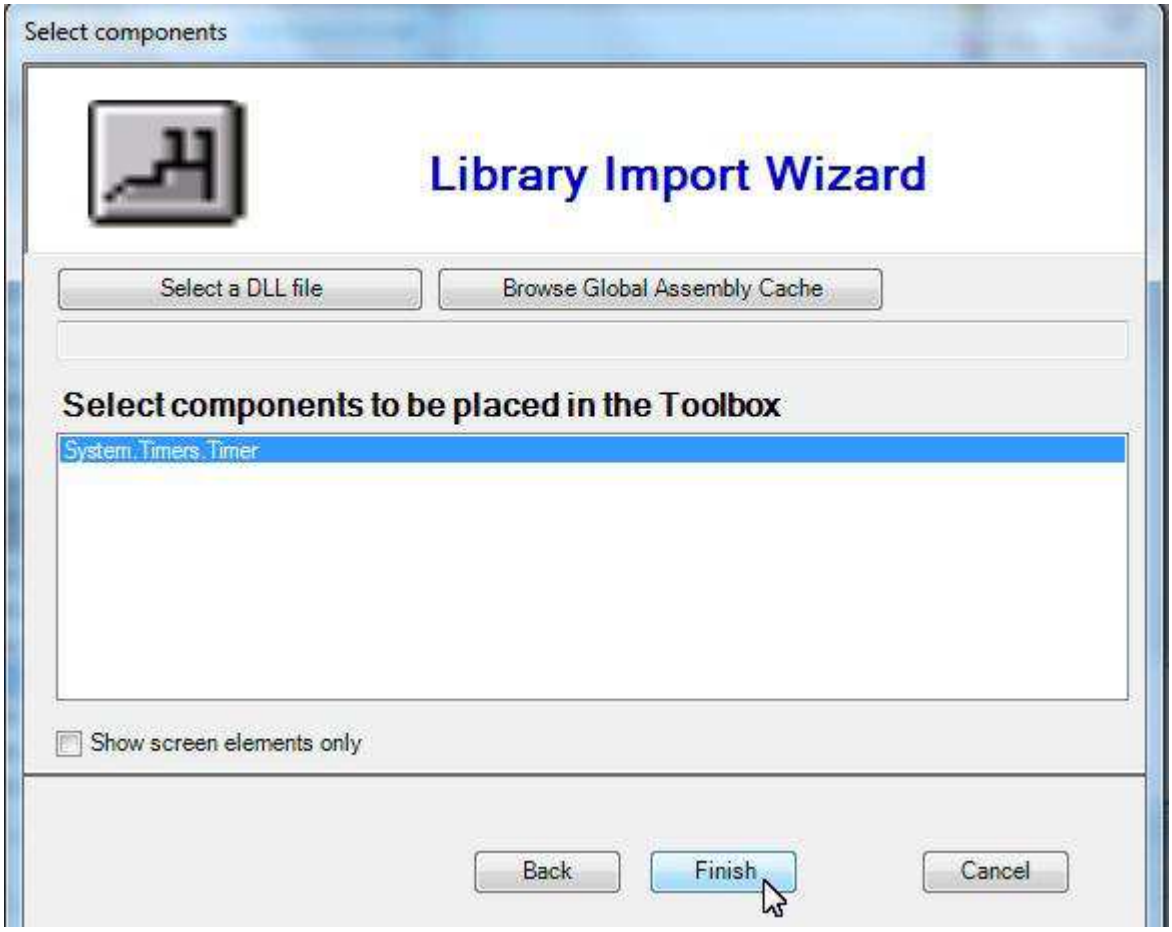

Library Import
Library Importer
This tool imports ActiveX components or Microsoft .NET classes into your projects. The imported components will appear in the Toolbox under the tabs named after your projects.
To import an ActiveX component some new DLL files will be created in your project folder to wrap the ActiveX component to be imported in .Net classes.
Start
Finish


Select library type
Library Import Wizard
Import .NET classes
Import ActiveX components
Back
Next
Cancel

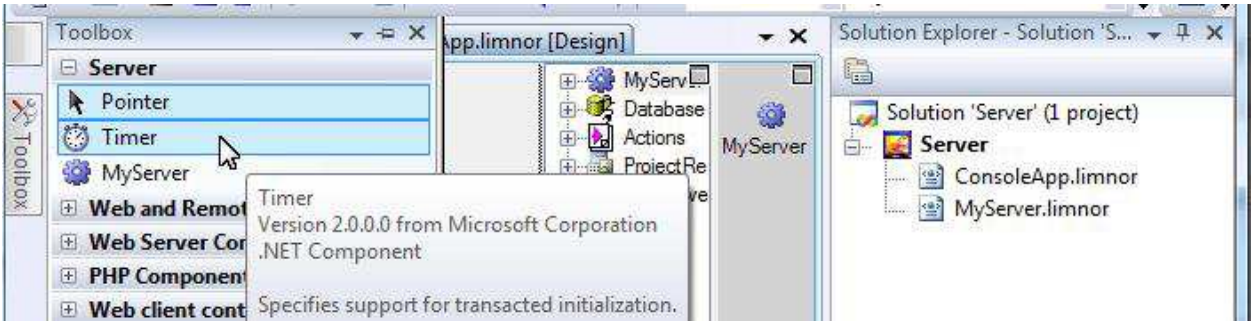

Select components
Library Import Wizard
Select a DLL file
Browse Global Assembly Cache


Type Selection
sysglobl, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitectu
System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089, processorArchitect
Microsoft.CSharp.CSharpCodeProvider, System, Version=2.0.0.0, Culture=neutral, PublicKeyTok
Microsoft.VisualBasic.VBCodeProvider, System, Version=2.0.0.0, Culture=neutral, PublicKeyToke

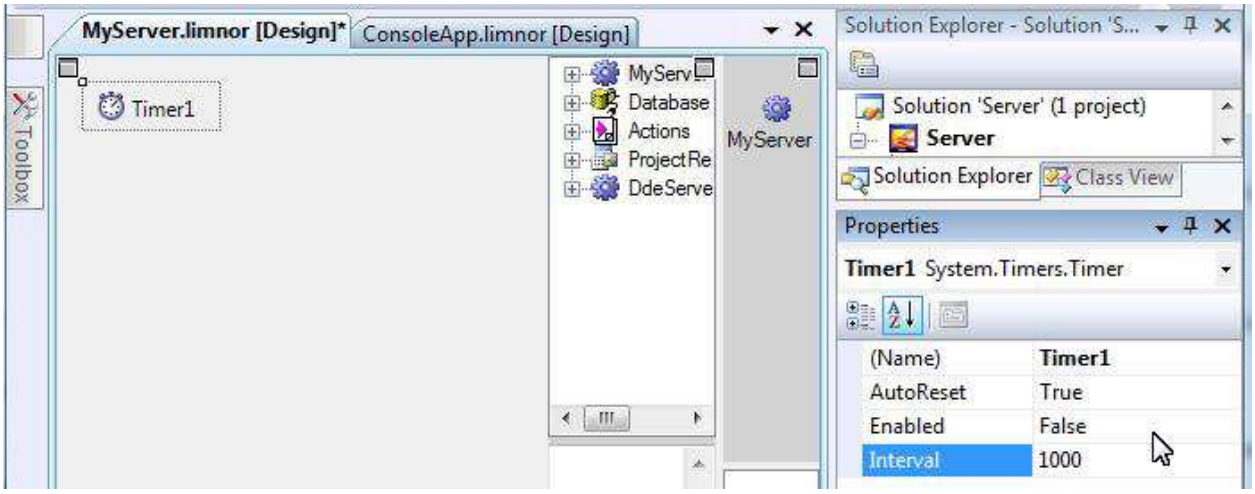

Type Selection
System.Threading.ThreadExceptionEventHandler, System, Version=2.0.0.0, Culture=neutral, PublicKeyTok
System.Timers.ElapsedEventArgs, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561
System.Timers.ElapsedEventHandler, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c
System.Timers.Timer, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
System.Timers.TimersDescriptionAttribute, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77
System.Uri, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
System.UriBuilder, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
Load File
OK
Cancel



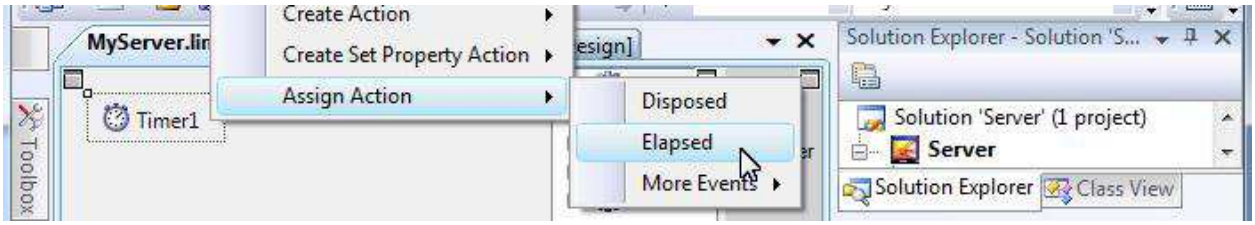
Drop a timer to the class:

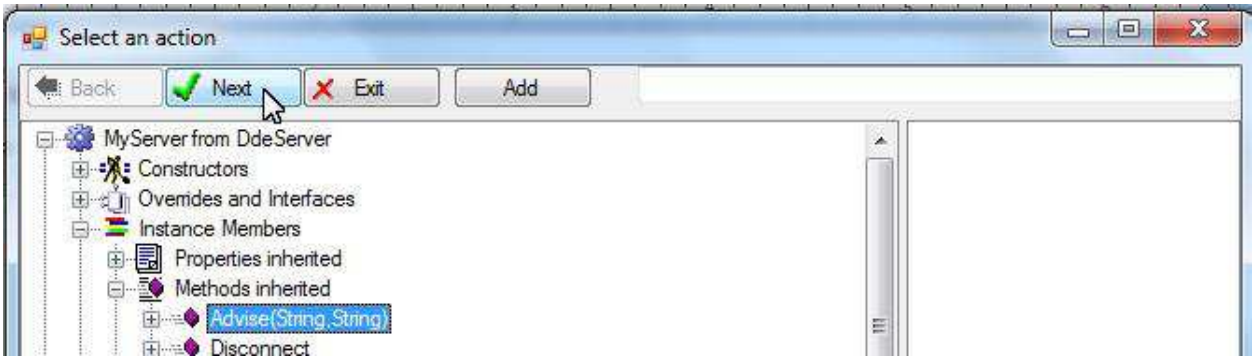


Set its properties:

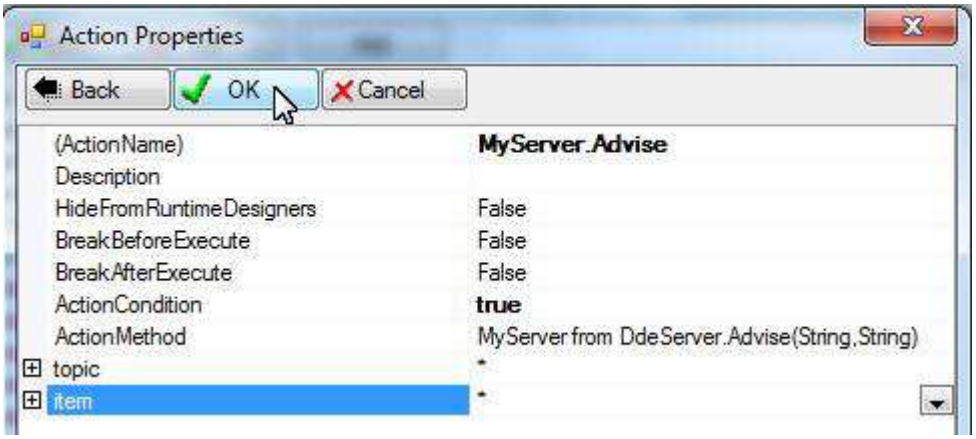


When the timer elapsed event occurs we want to execute an Advice action.

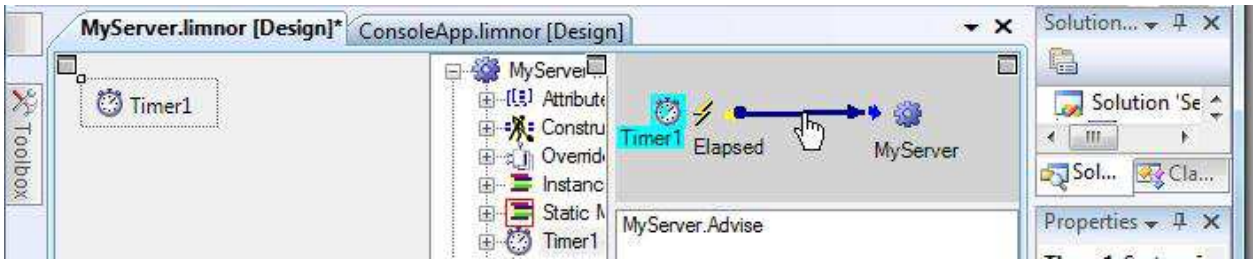




Use * for both parameters “topic” and “item”:



The action is created and assigned to timer elapsed:

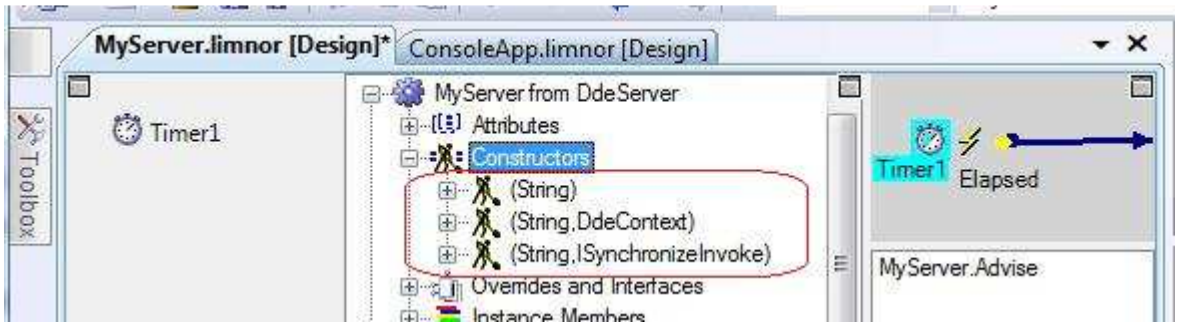


Derive constructors

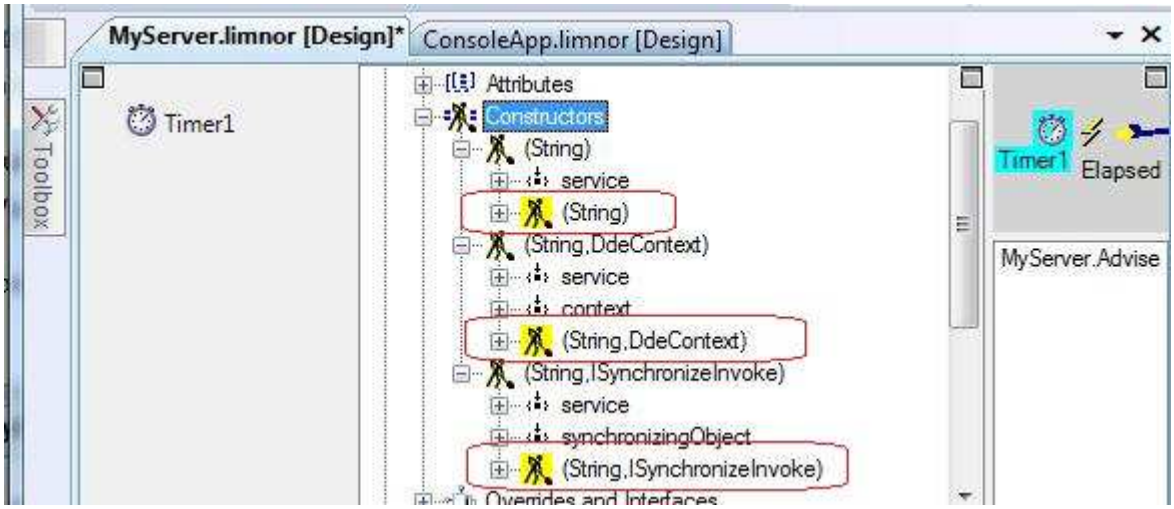
Creating a constructor is like handling an initialization event. We must follow some rules.

- Our constrictor must be derived from an existing constructor of the base class.
- We may add new parameters to our constructors but we cannot remove existing parameters.

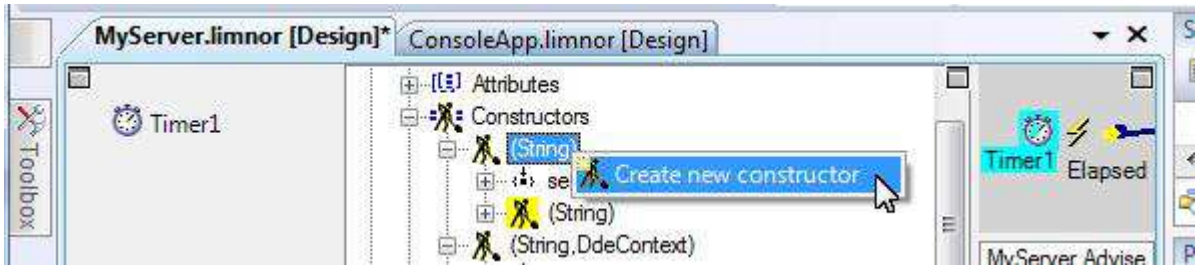
Existing constructors can be found under “Constructors”. For DdeServer class, it provides 3 existing constructors.



Our own constructors are represented by icon  under each existing constructor:

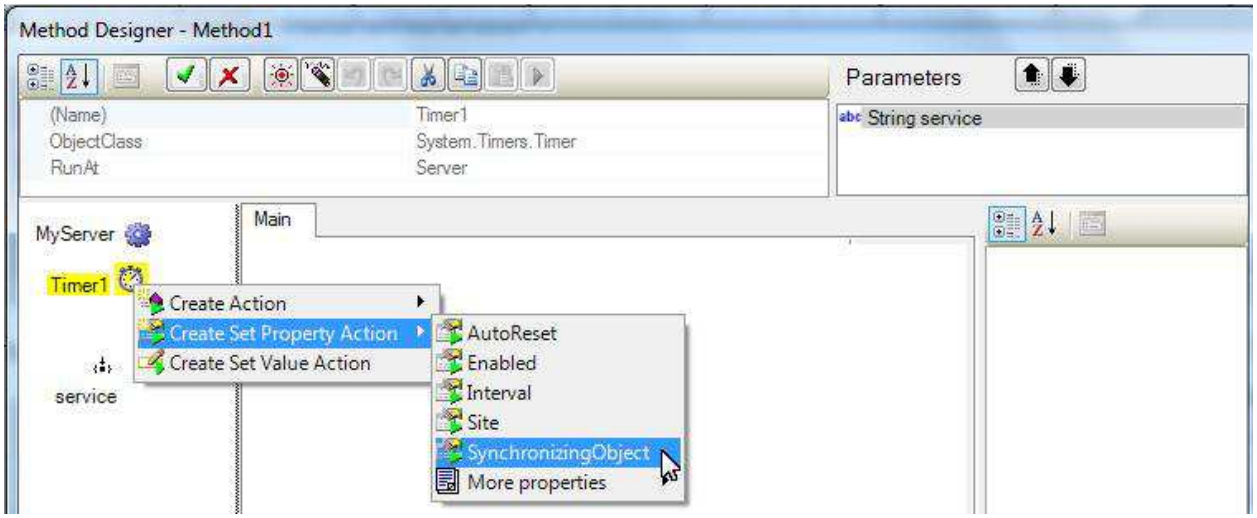


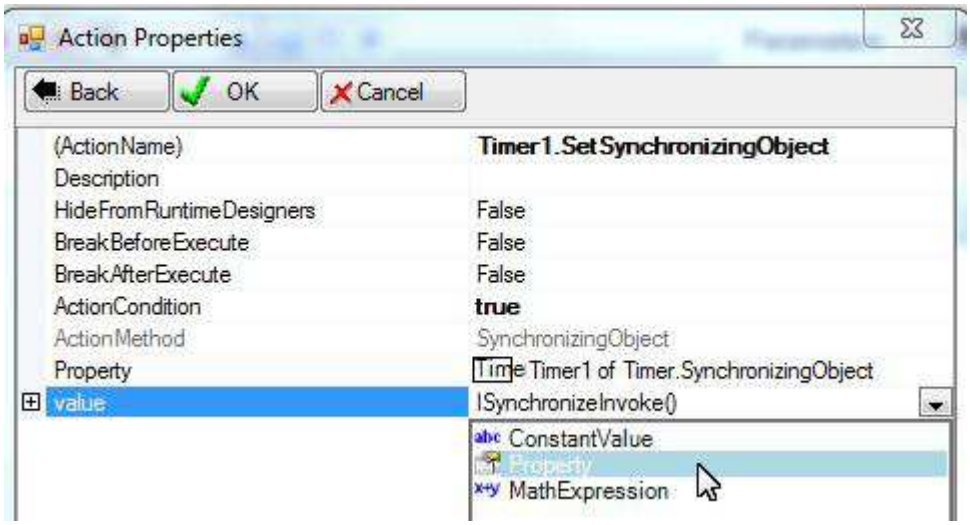
If we want to add new parameters then we have to create new constructors. Right-click an existing constructor and choose “Create new constructor”.

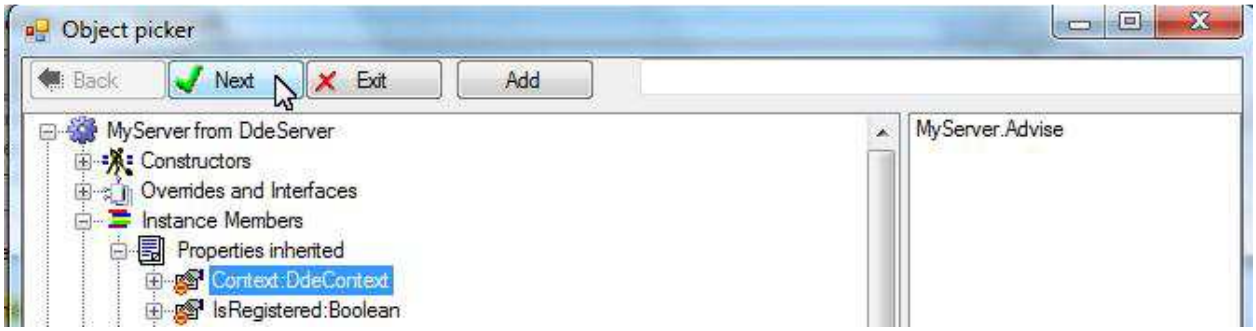


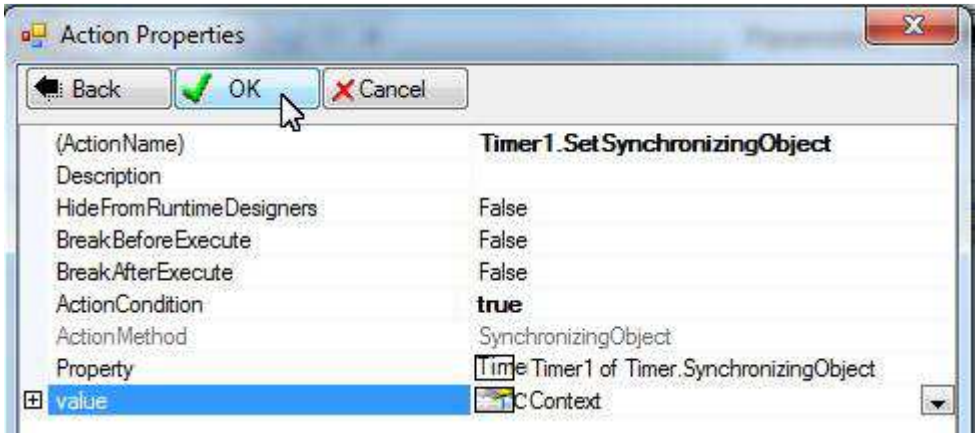
To add actions to a constructor, right-click it and choose “Edit”:

A Method Editor appears. For this sample, we set the synchronization object of the timer to the DDE server context.

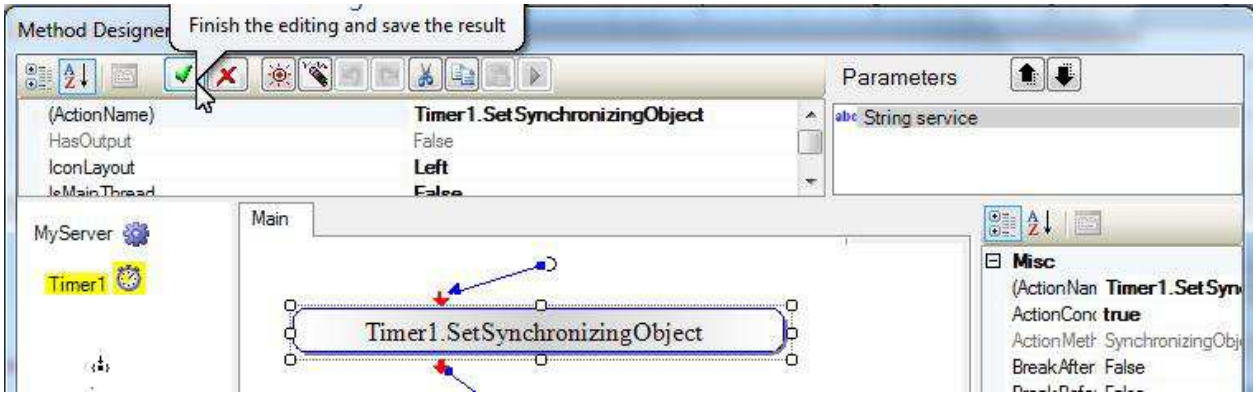






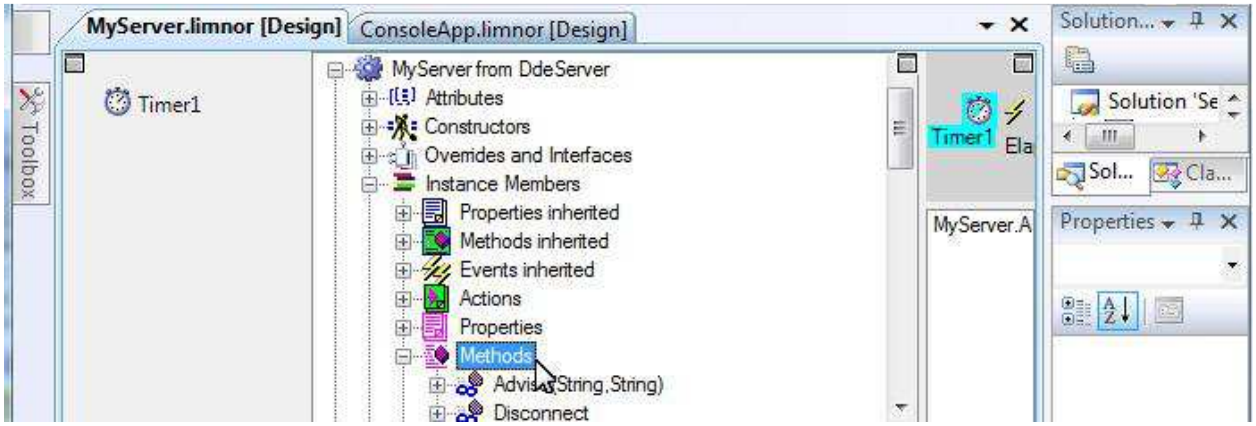


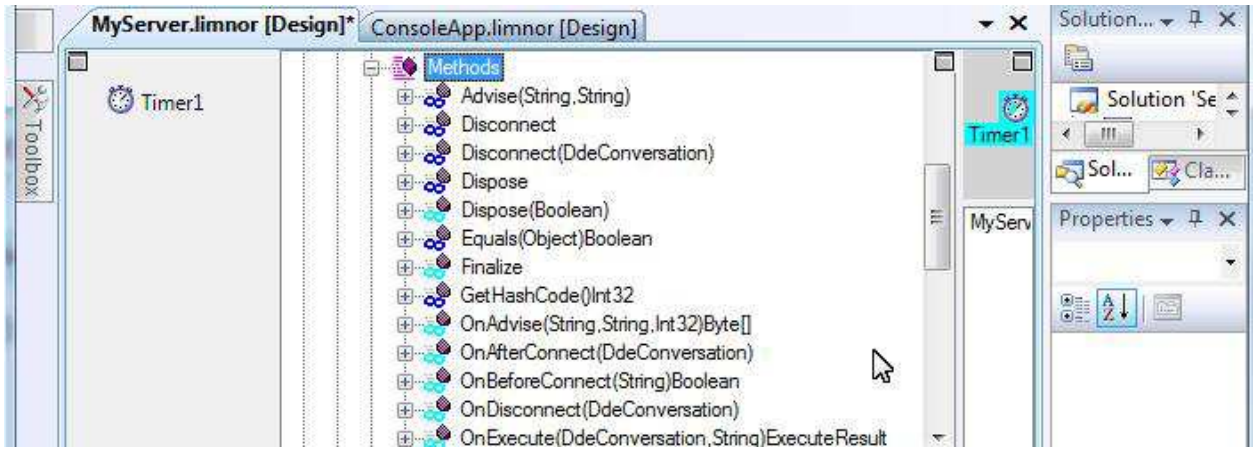
For this sample, that is all what we need for this constructor:



Override base functions

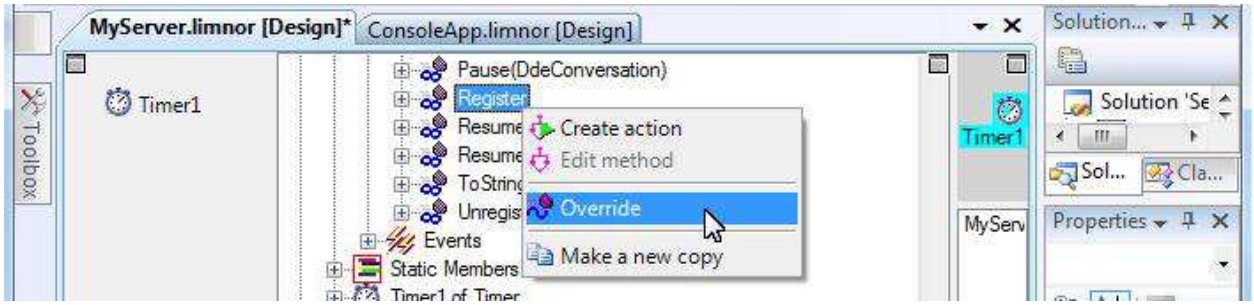
The DdeServer provides many methods which can be overridden. Such methods are listed under "Methods".



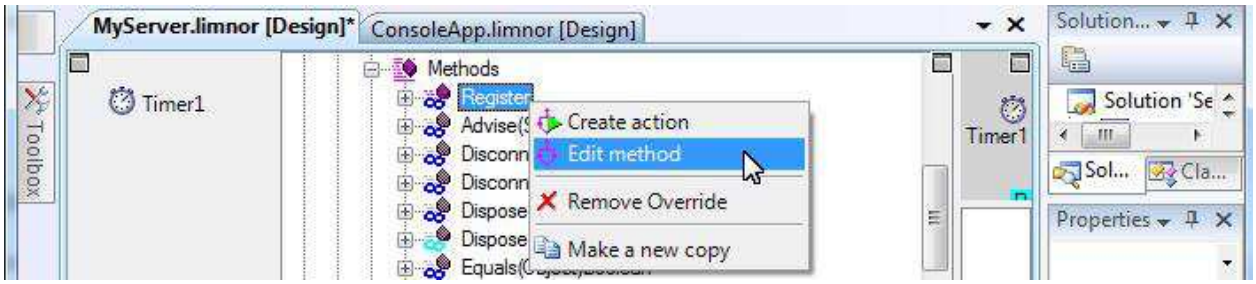


Start timer

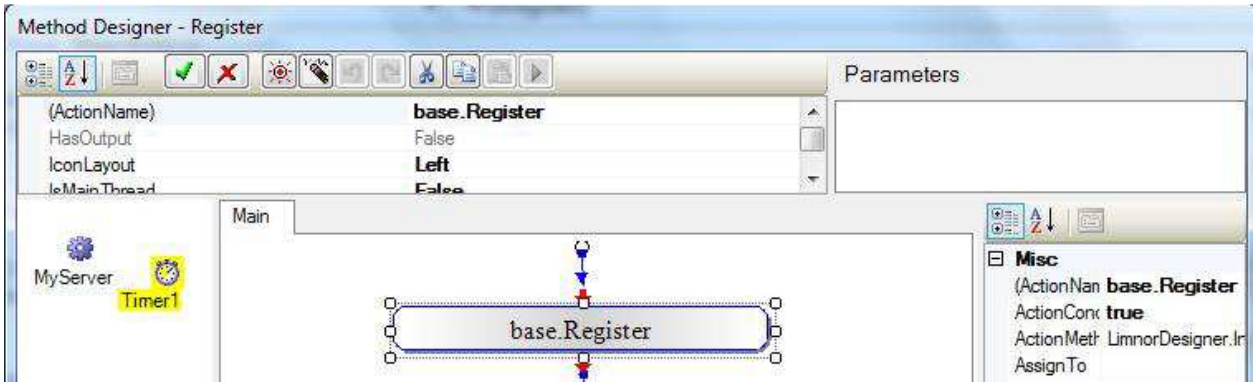
We override Register method to start the timer.



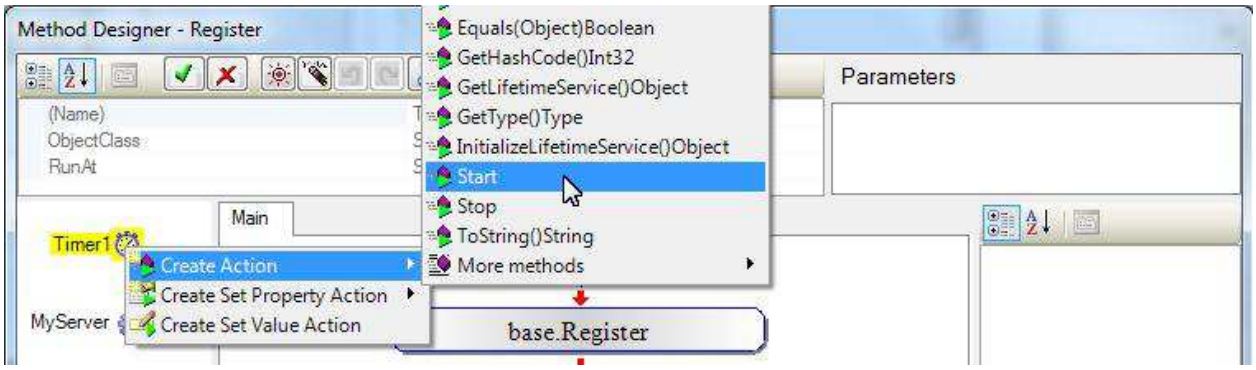
Edit the overridden method:

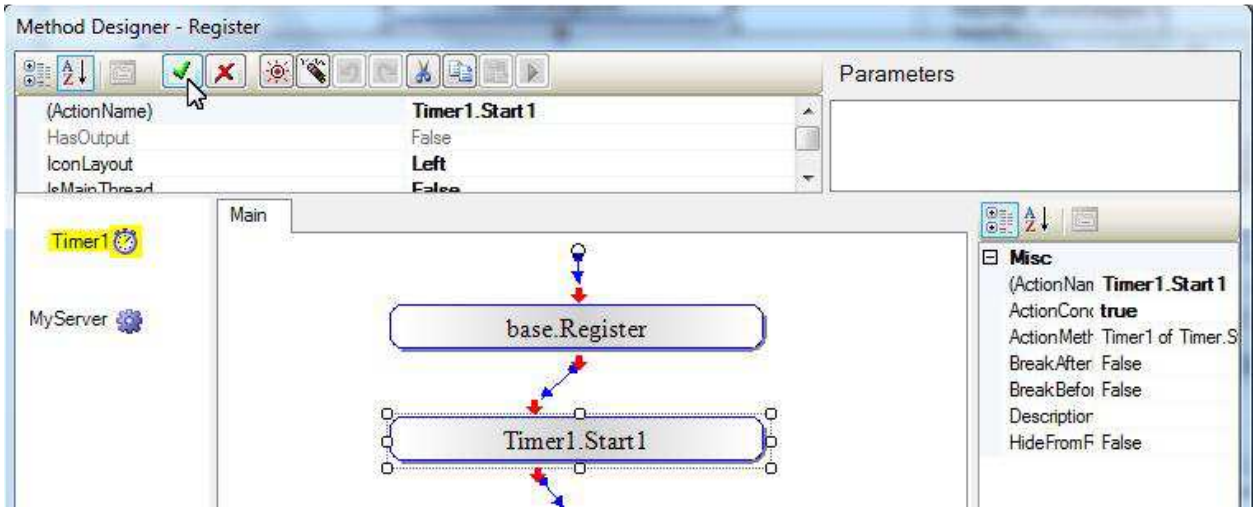


By default the overridden method has an action to execute the base version of the method.



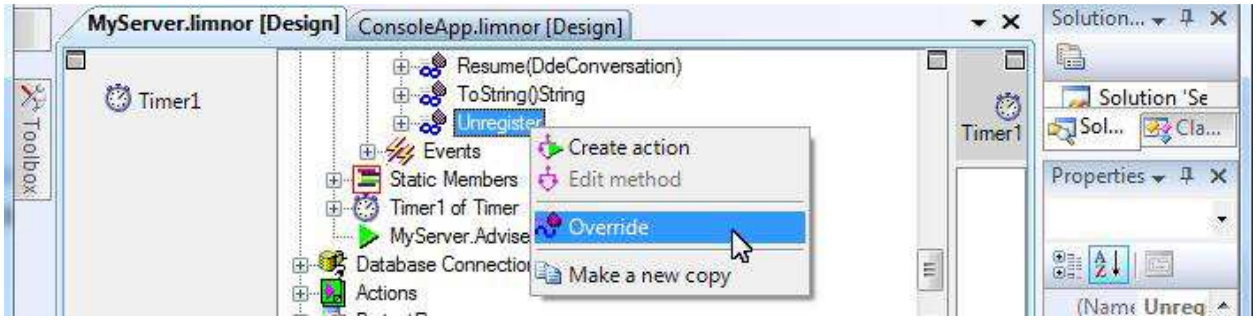
Add an action to start the timer:

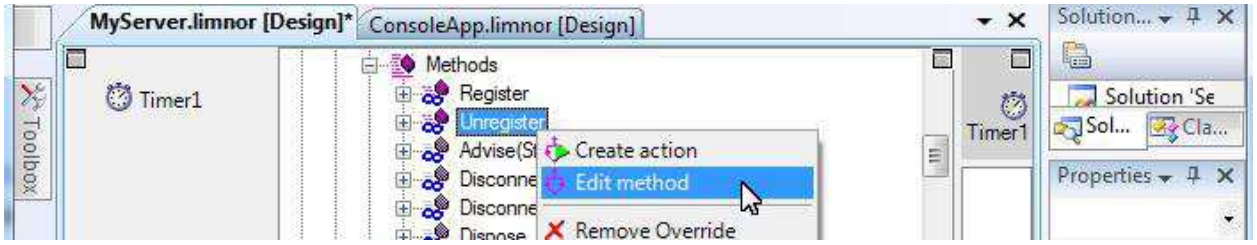




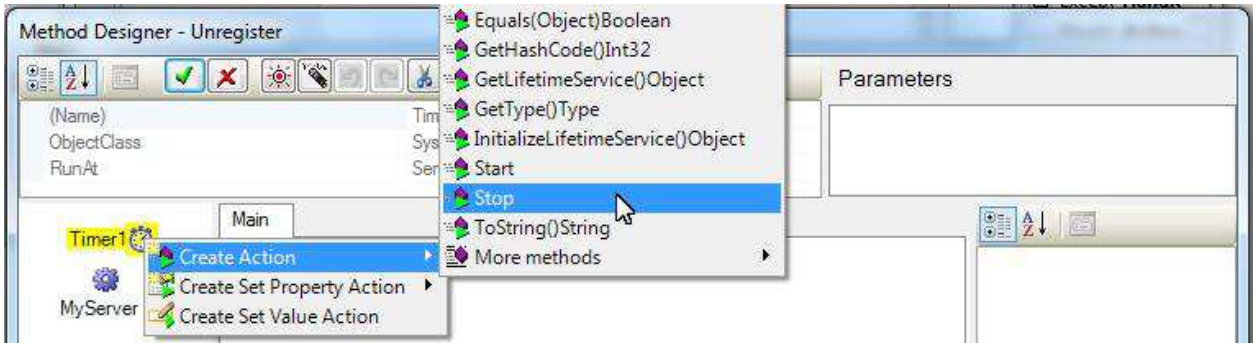
Stop the timer

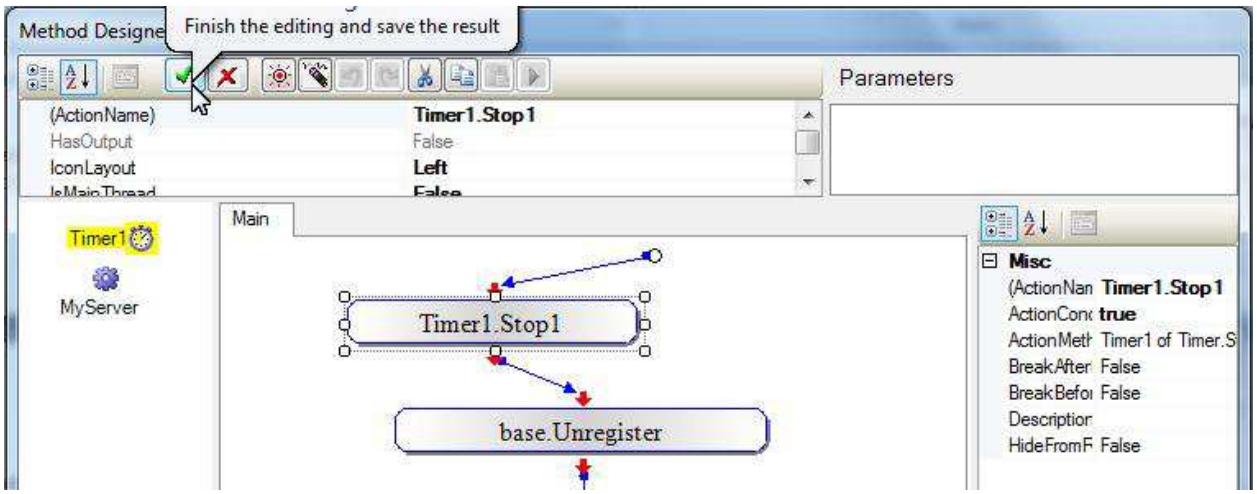
We override Unregister method to stop the timer.





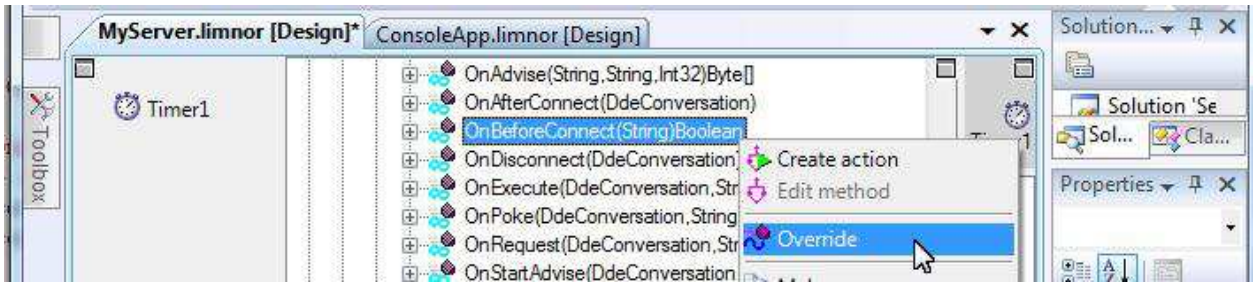
Add a Stop action:

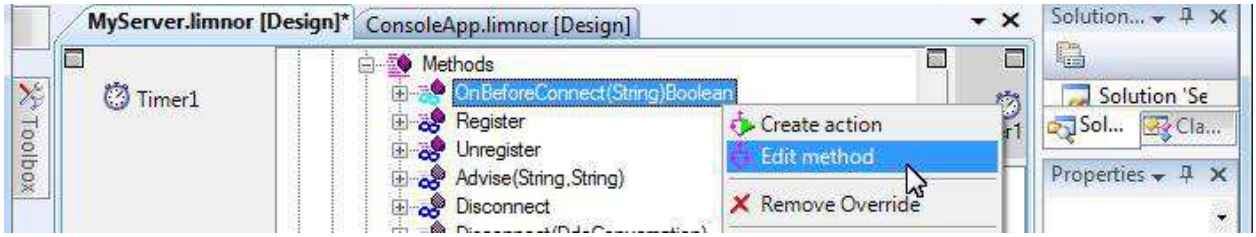




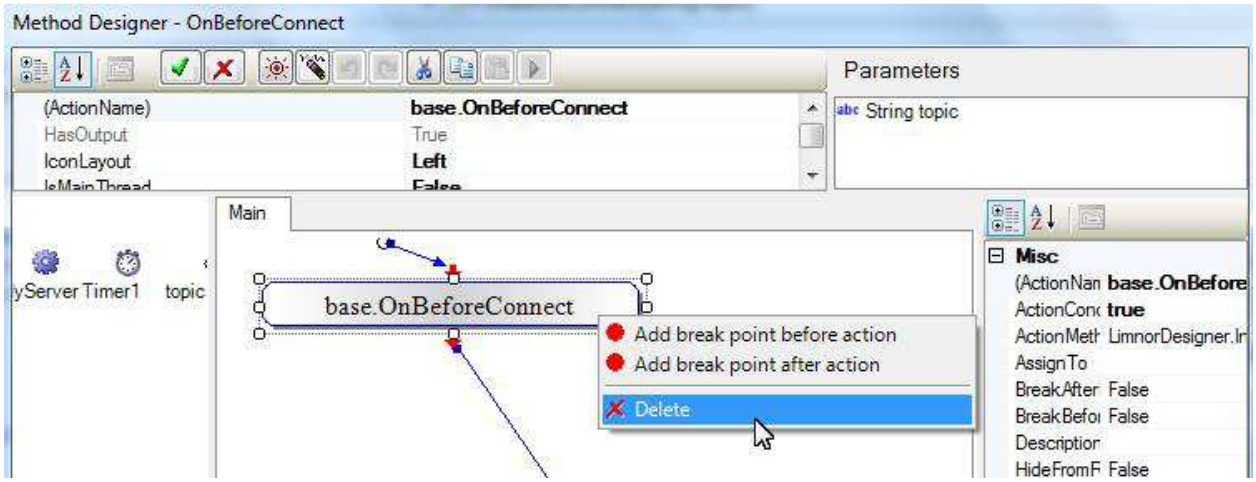
Override other methods

This NDDE sample overrides other DdeServer methods by simply writing the method parameters to the console for informational purpose. We show the process of overriding one of the methods.

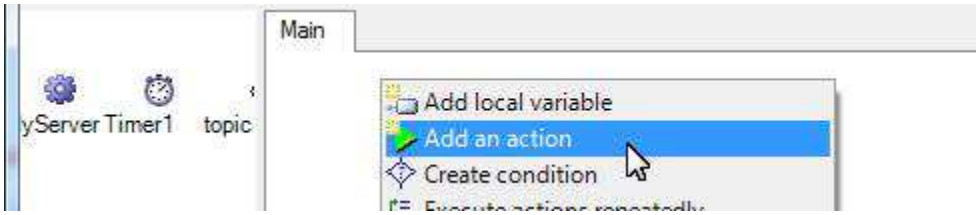


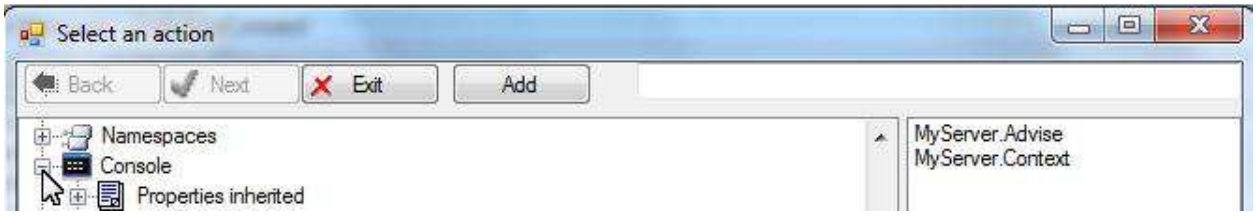


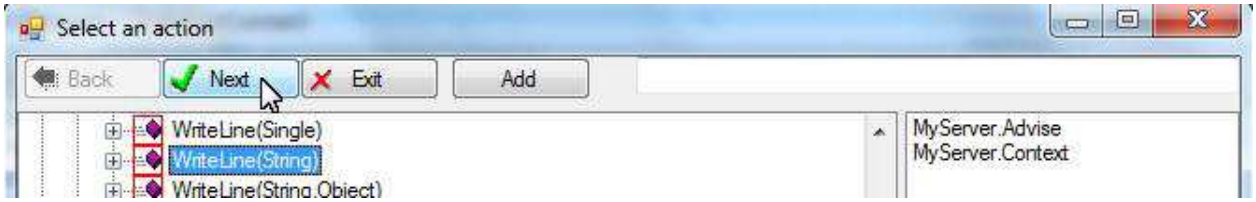
No need to execute the base version of the method. Delete it:



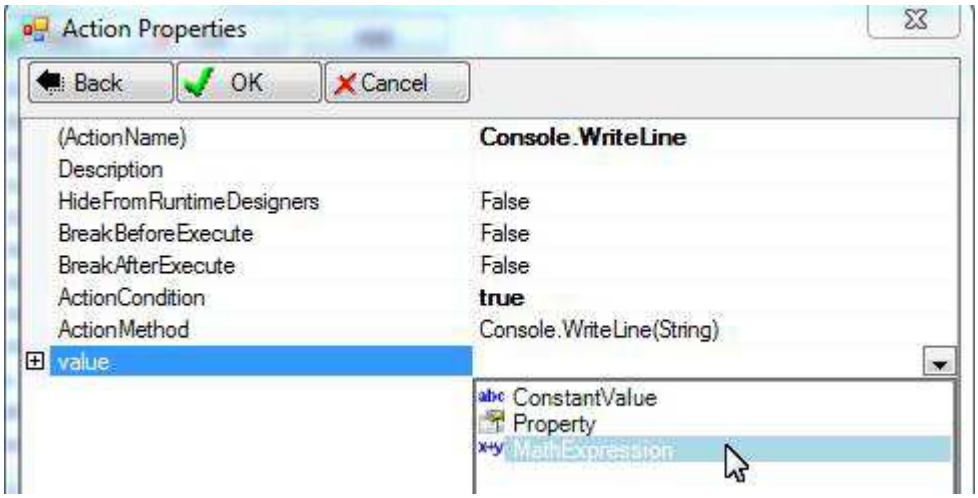
Add an action to write information to console:



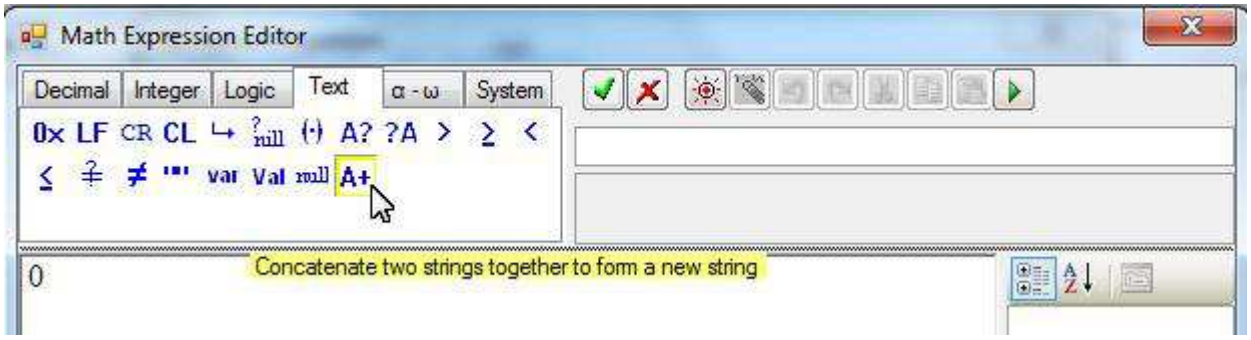


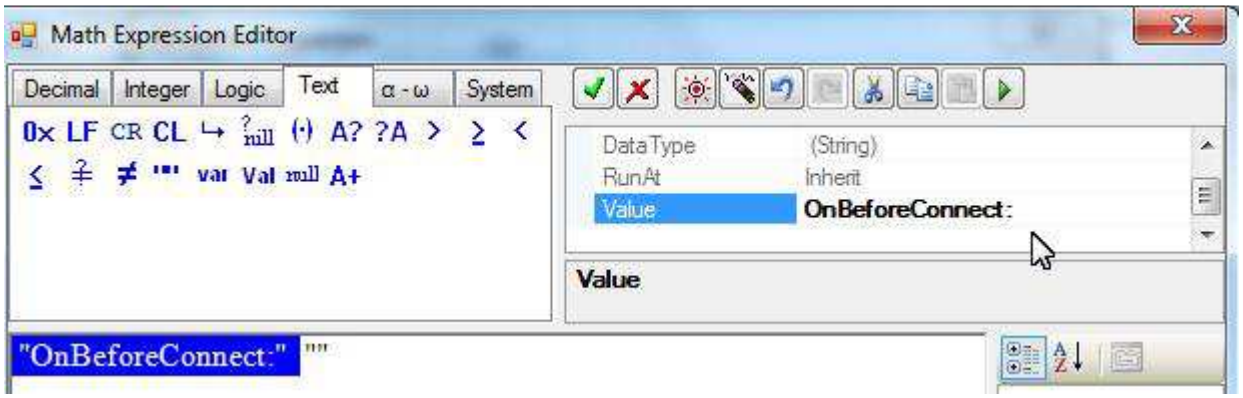


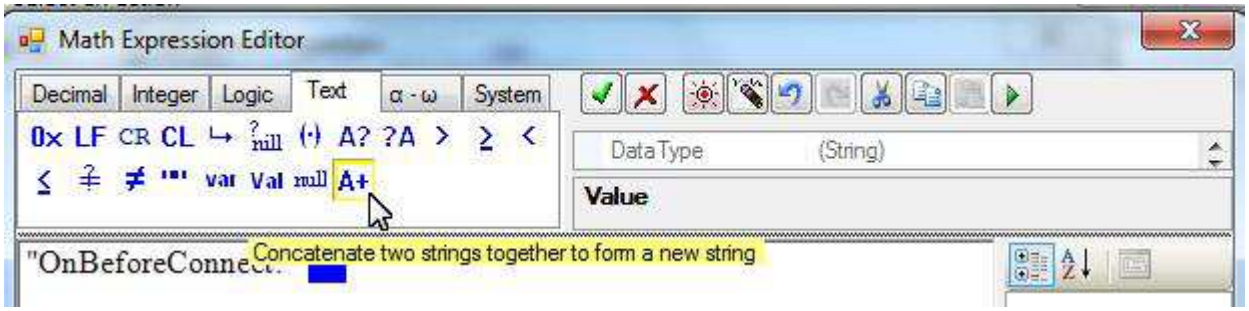
Use an expression to form the information

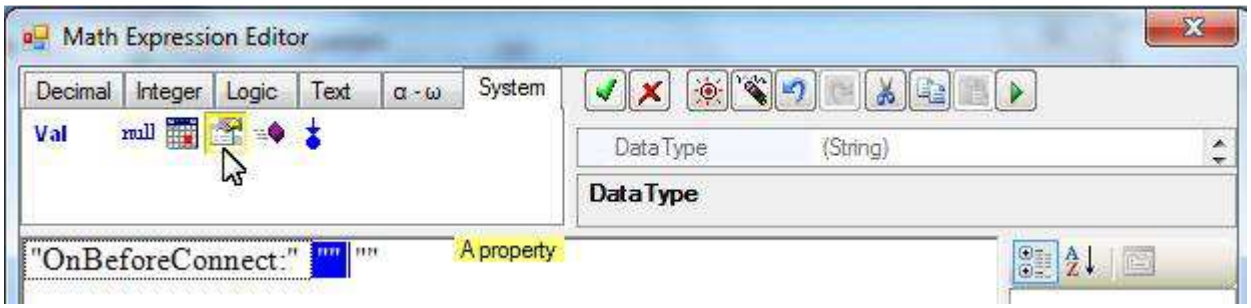


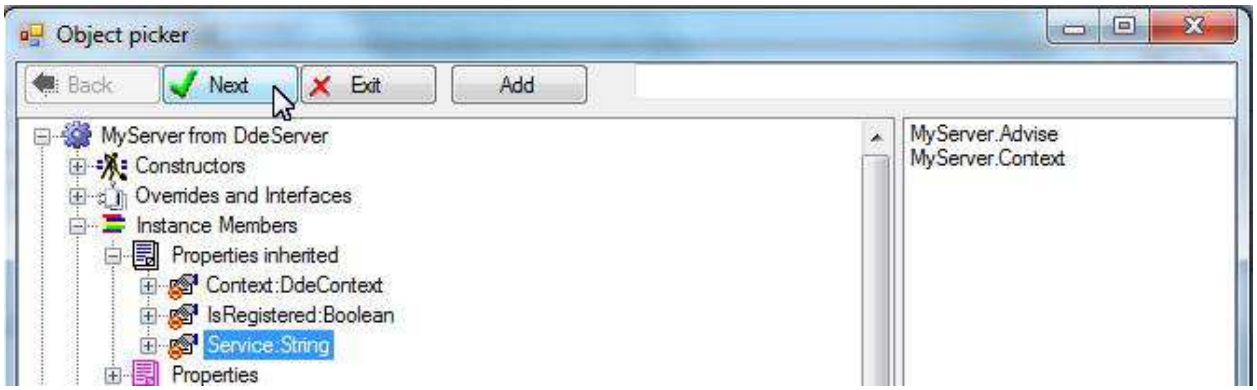
Click A+, to add string concatenation:

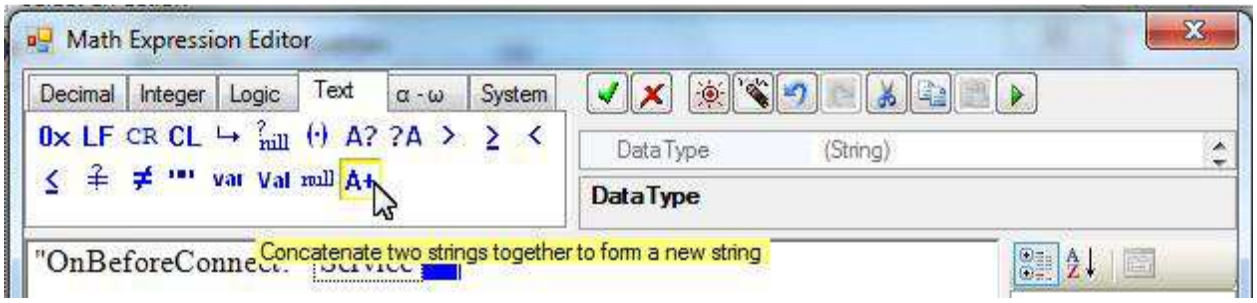


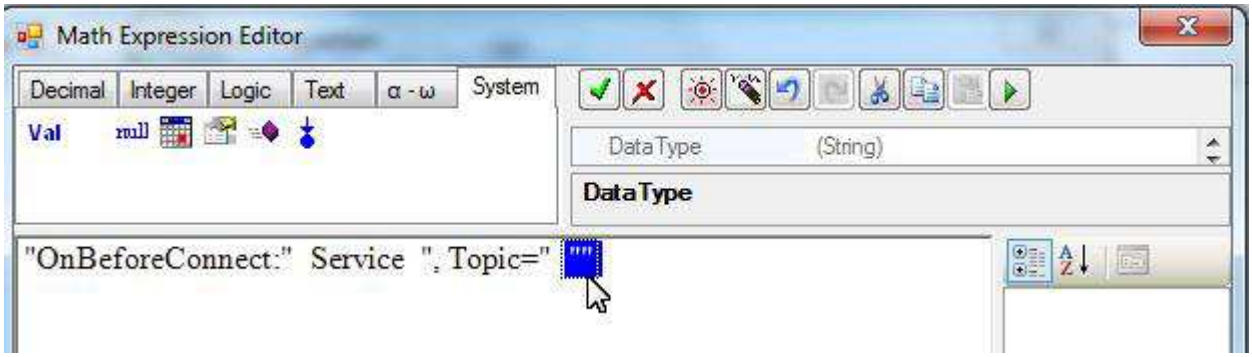


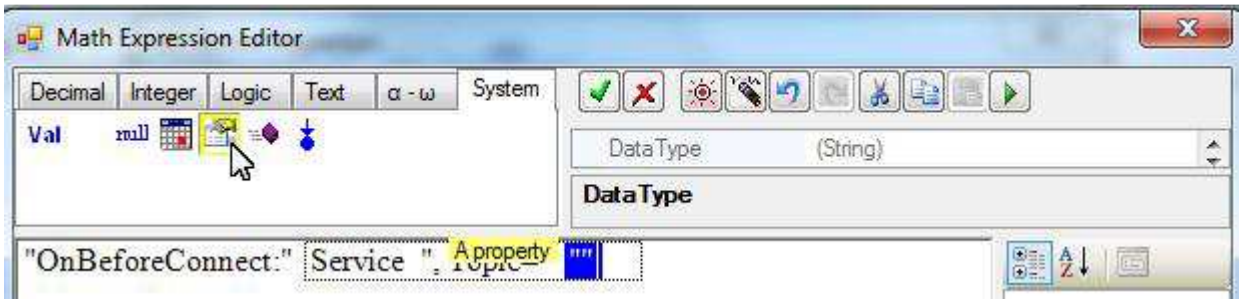

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
0x LF CR CL
A? ?A
DataType
(String)
Value
"OnBeforeConnect:"
Concatenate two strings together to form a new string

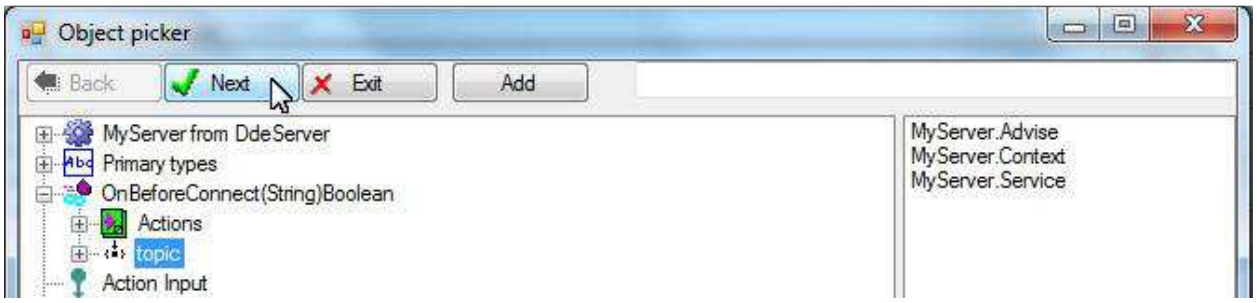

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(String)
DataType
"OnBeforeConnect:"
""
A property

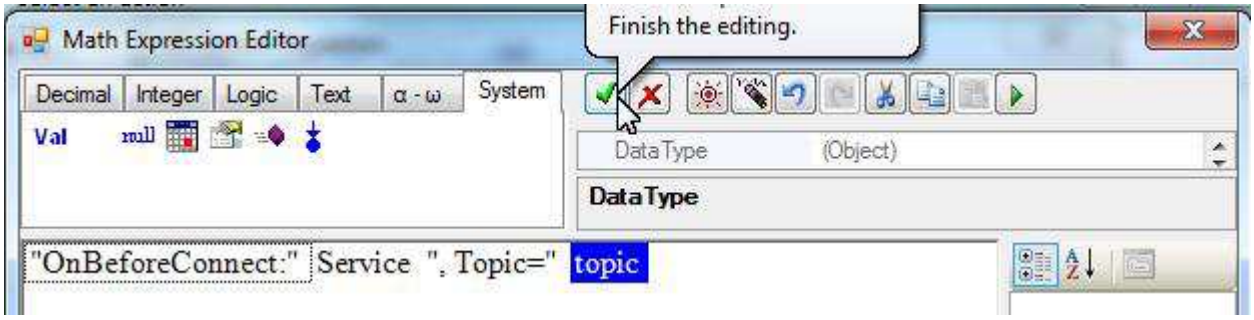

Object picker
Back
Next
Exit
Add
MyServer from DdeServer
Constructors
Overrides and Interfaces
Instance Members
Properties inherited
Context:DdeContext
IsRegistered:Boolean
Service:String
Properties
MyServer.Advise
MyServer.Context

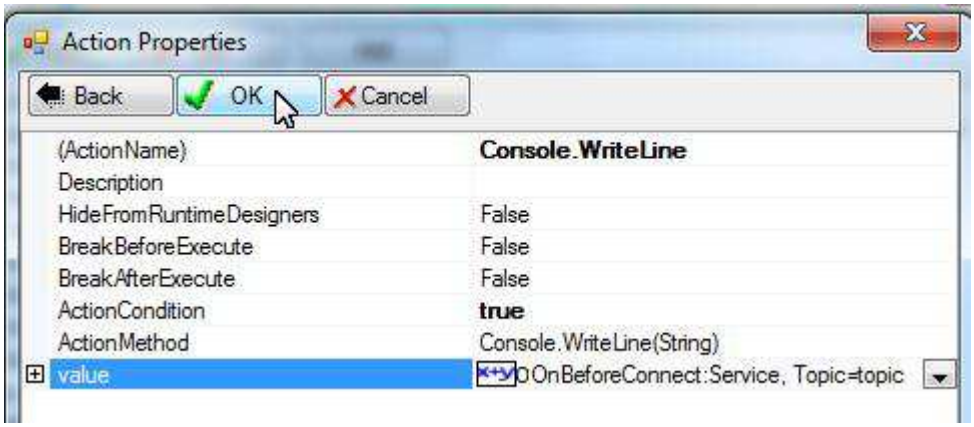

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
0x LF CR CL
A? ?A
DataType
(String)
DataType
"OnBeforeConnect:"
Concatenate two strings together to form a new string


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(String)
DataType
"OnBeforeConnect:" Service ", Topic=" ""

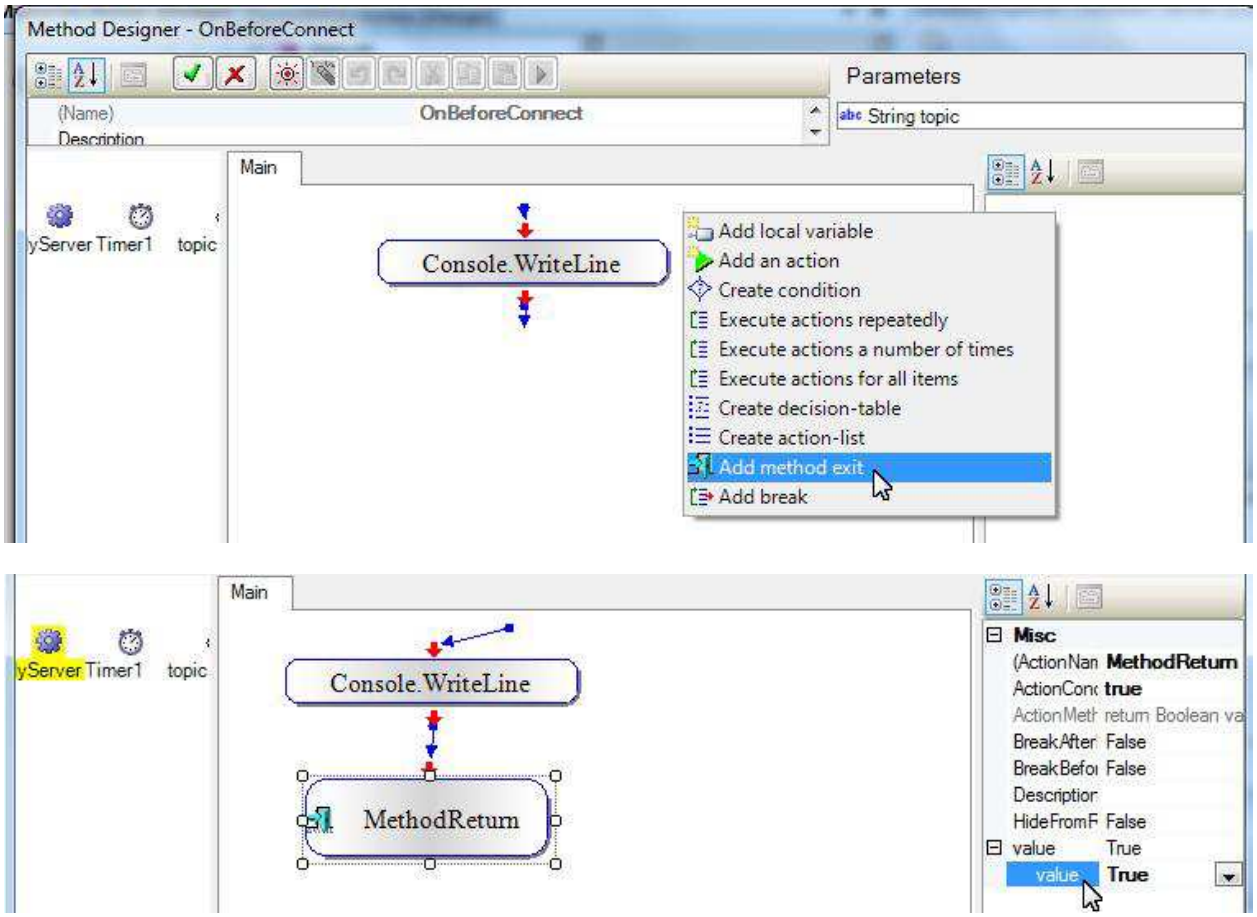

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(String)
DataType
"OnBeforeConnect:" Service ", Topic=" ""
A property


Object picker
Back
Next
Exit
Add
MyServer from DdeServer
Primary types
OnBeforeConnect(String)Boolean
Actions
topic
Action Input
MyServer.Advise
MyServer.Context
MyServer.Service


Finish the editing.
Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Object)
DataType
"OnBeforeConnect:" Service ", Topic=" topic



Add a method return action to return True to indicate that this server accepts the connection:



This method is thus overridden. We may override other methods in the similar way according to the NDDE sample file \Samples\cs\Server\Program.cs in the NDDE installation folder.

Below we show a string formatting technique. We override OnStartAdvice method to write following message to the console:

```
OnStartAdvise: Service= {conversation.Service}, Topic= {conversation.Topic}, Handle= {conversation.Handle}, Item={item}, Format= {format}
```

Where the information enclosed in { and } are variables. Forming such an expression is too long. One way to make an expression shorter is to use variables. For example,

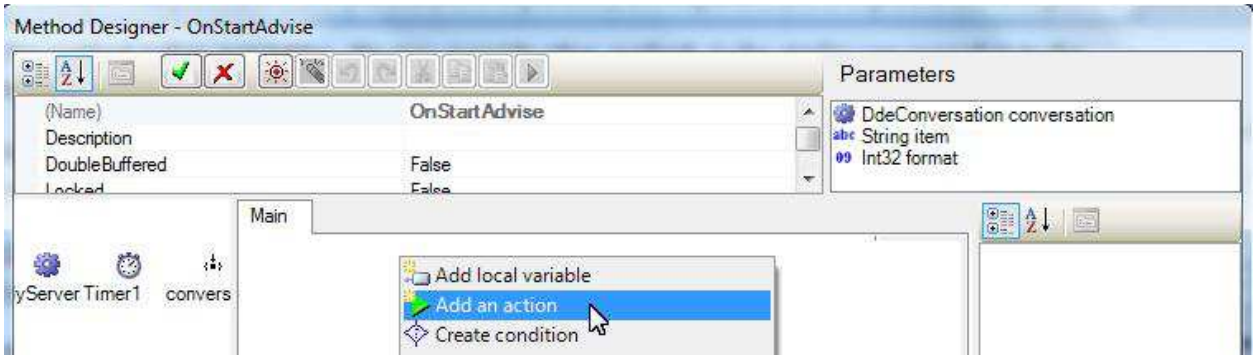
```
OnStartAdvise: Service= {s}, Topic= {t}, Handle= {h}, Item={i}, Format= {f}
```

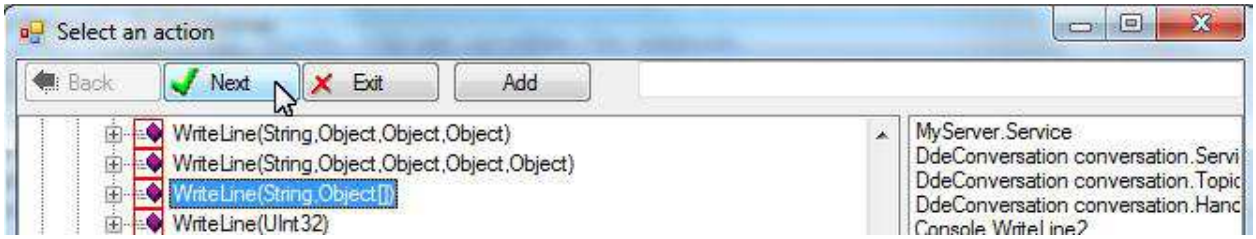
And then map the variables s = conversation.Service; t = conversation.Topic; h = conversation.Handle; i = item; and f = format.

Another way of formatting text is to use StringTool component. See
http://www.limnor.com/support/FormatText.pdf

Below we show another way of formatting text.

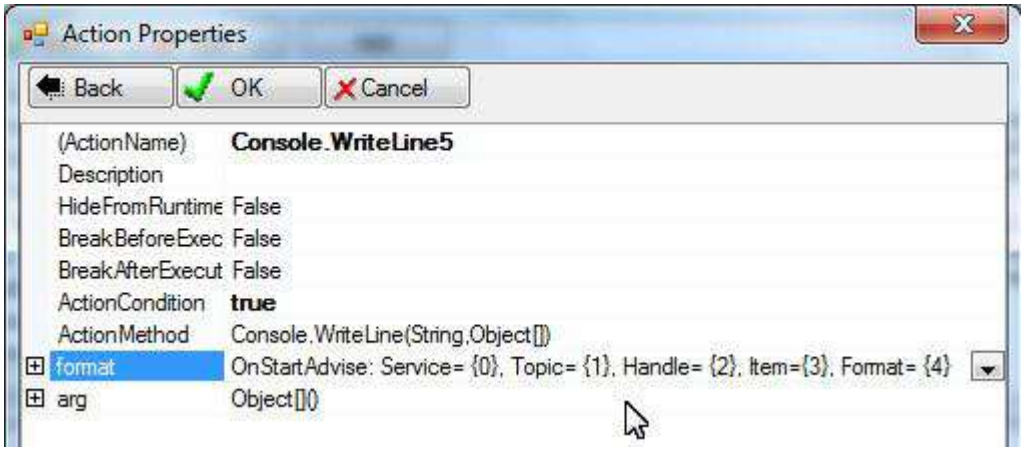
Instead of choosing "WriteLine(string)", we choose "WriteLine(string, object[])":





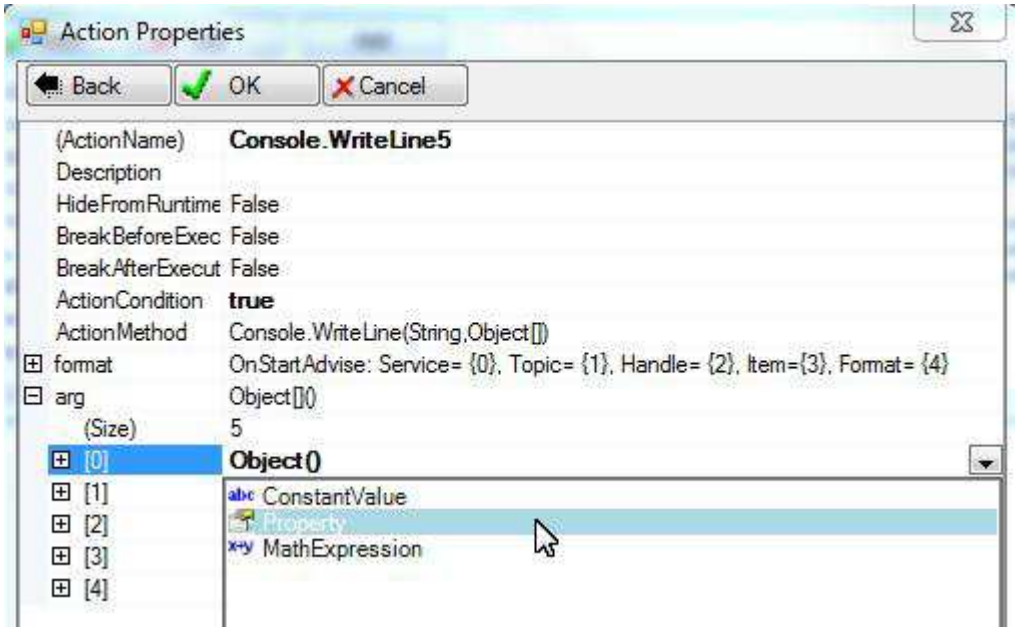
For the "format" parameter, we use

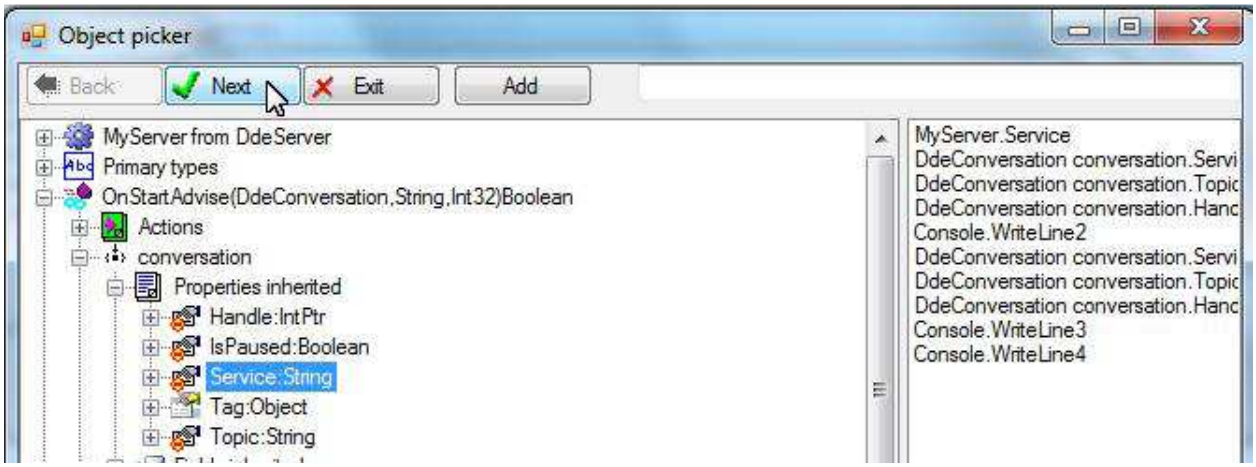
```
OnStartAdvise: Service= {0}, Topic= {1}, Handle= {2}, Item={3}, Format= {4}
```



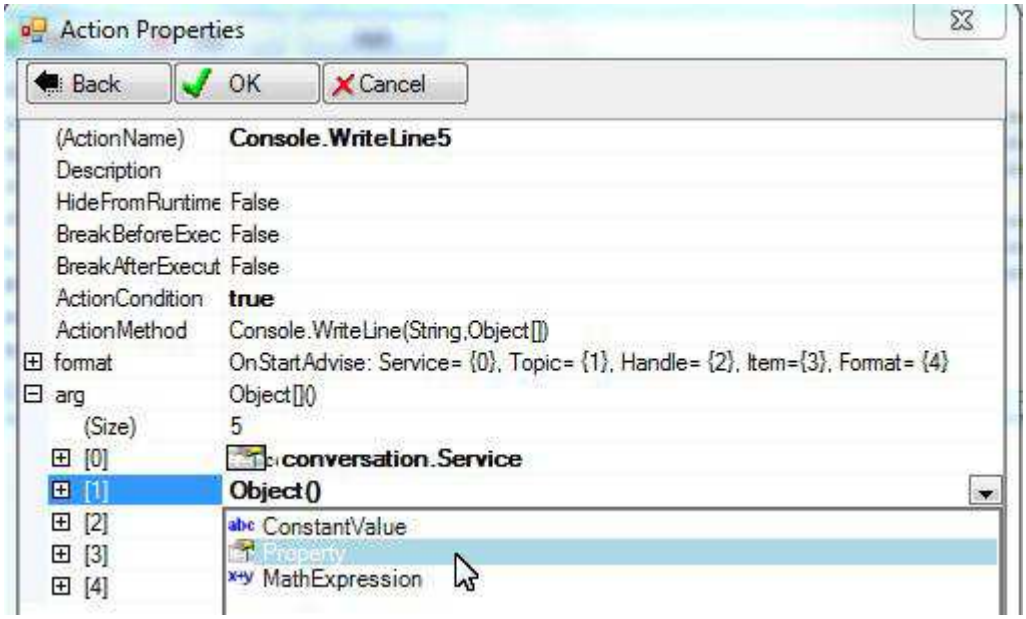
Where each {n}, n=0,1,2,3,4, is a place holder to be supplied by “arg”. So, we set “arg” to be of 5 elements:

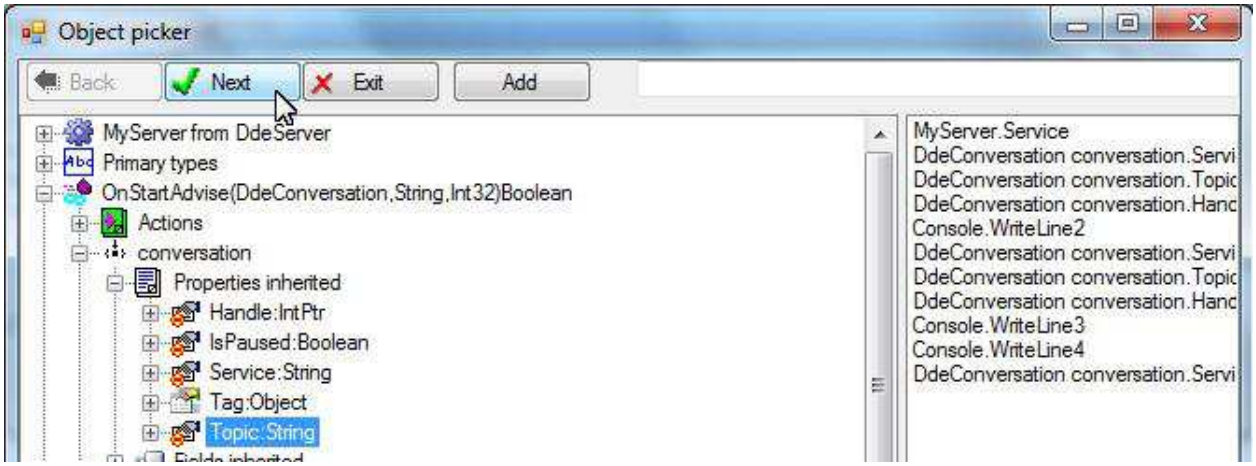
Set the first element to the service of the conversation:



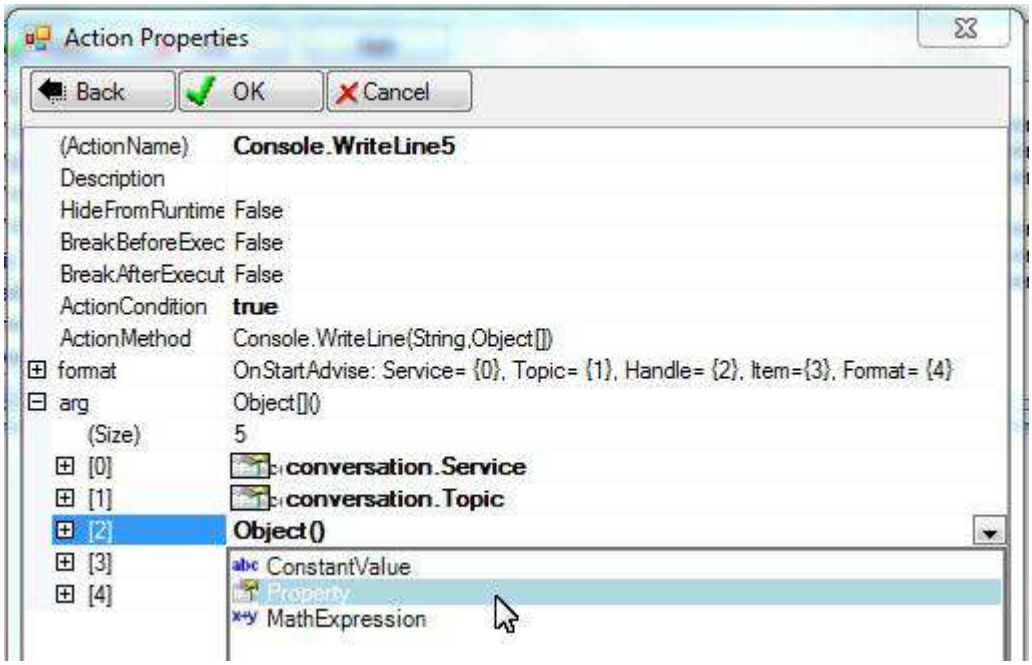


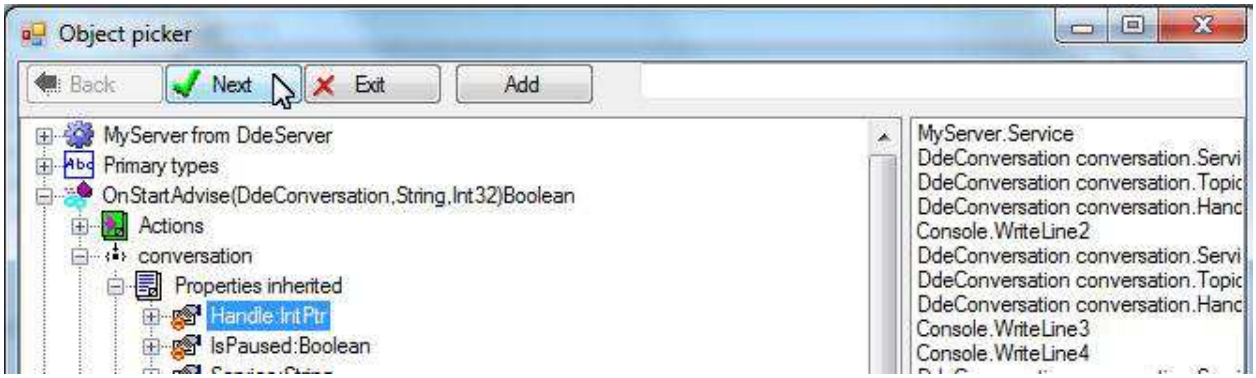
Set the second element to the topic of the conversation:



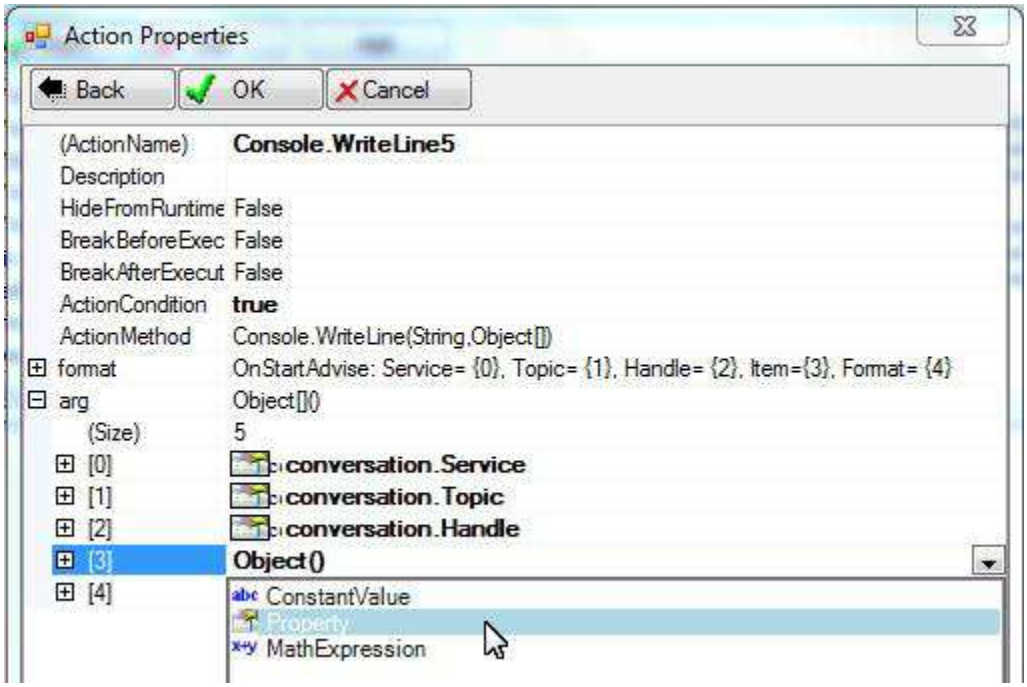


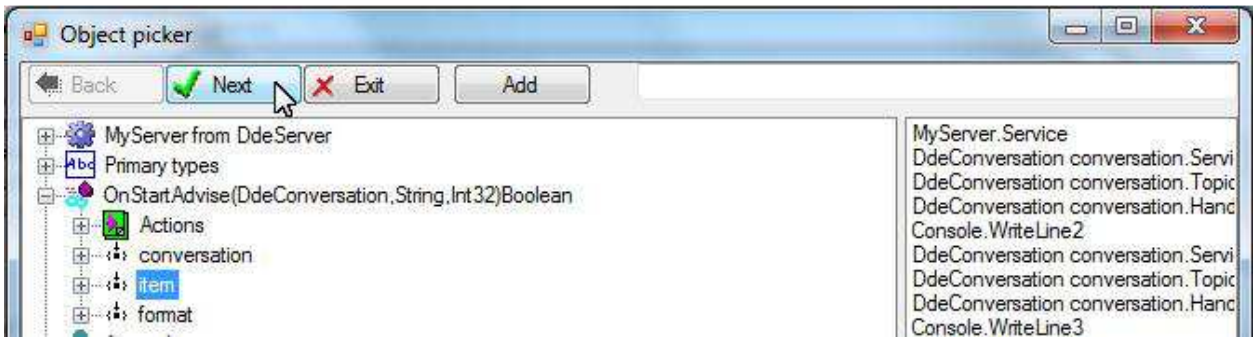
Set the third element to conversation handle:



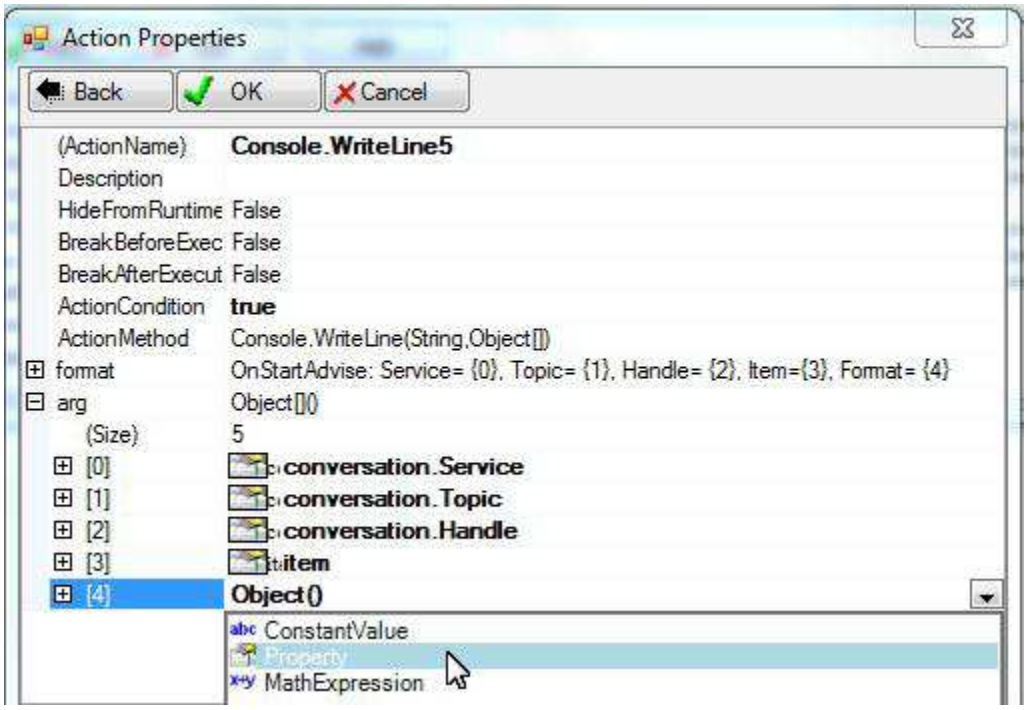


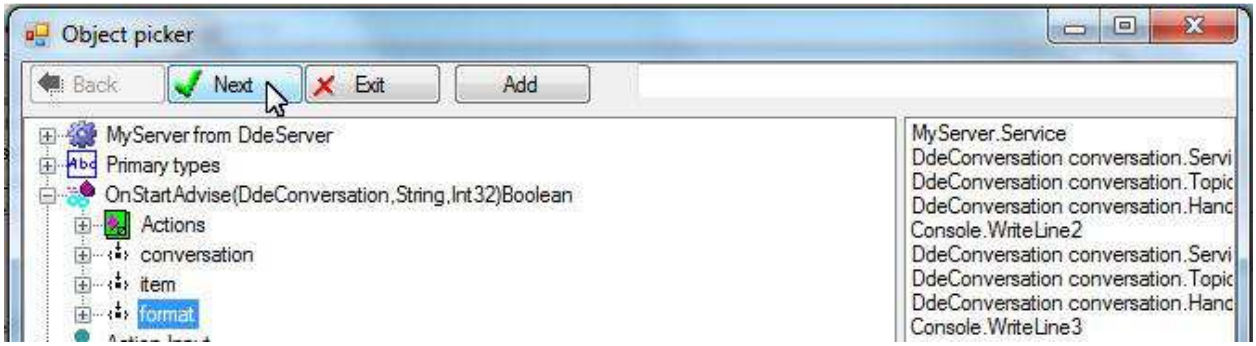
Set the fourth element to item:



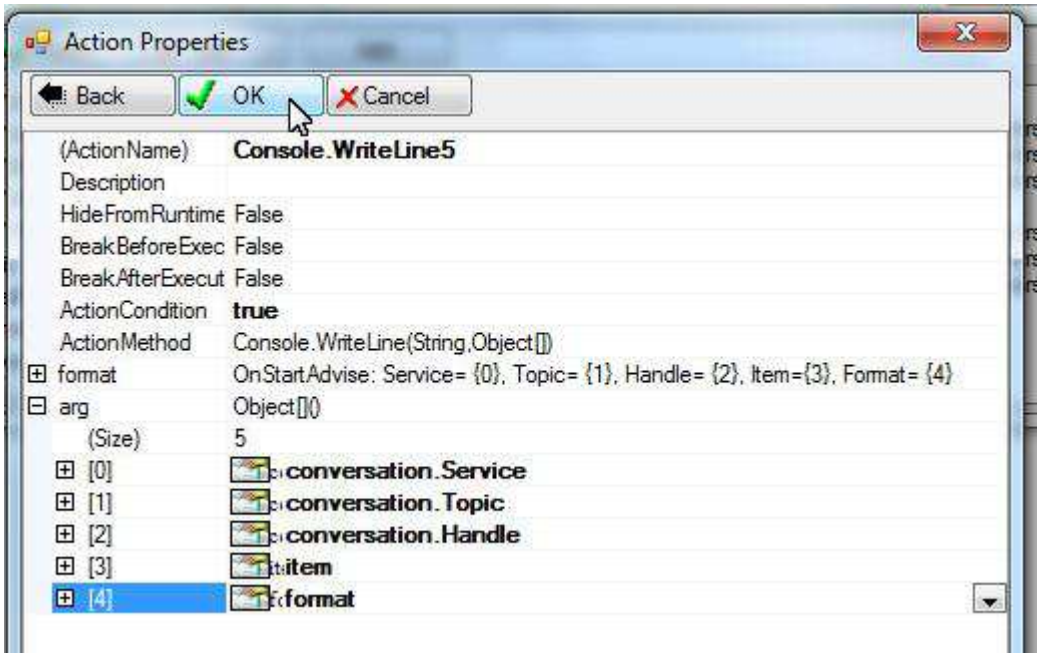


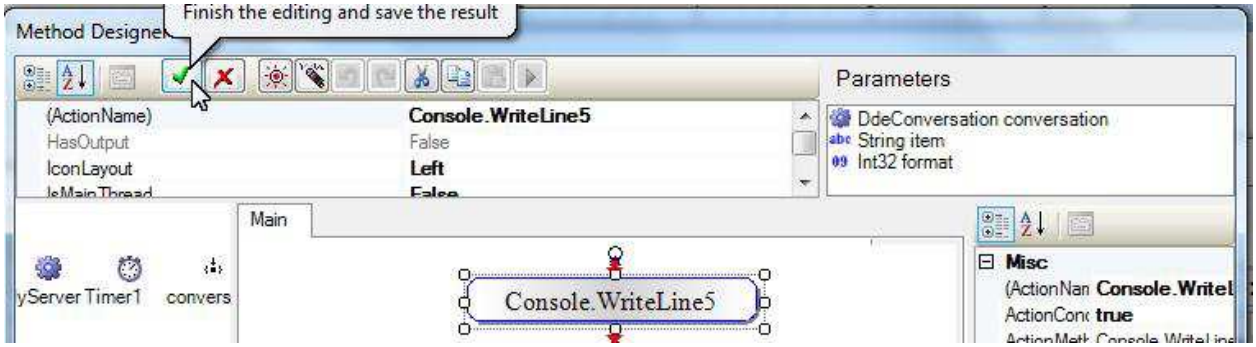
Set the fifth element to the format:


Action Properties
Back
OK
Cancel
(ActionName) Console.WriteLine5
Description
HideFromRuntime False
BreakBeforeExec False
BreakAfterExecut False
ActionCondition true
ActionMethod Console.WriteLine(String,Object[])
format OnStartAdvise: Service= {0}, Topic= {1}, Handle= {2}, Item={3}, Format= {4}
arg Object[]()
(Size) 5
[0] conversation.Service
[1] conversation.Topic
[2] conversation.Handle
[3] item
[4] Object()
ConstantValue
Property
MathExpression

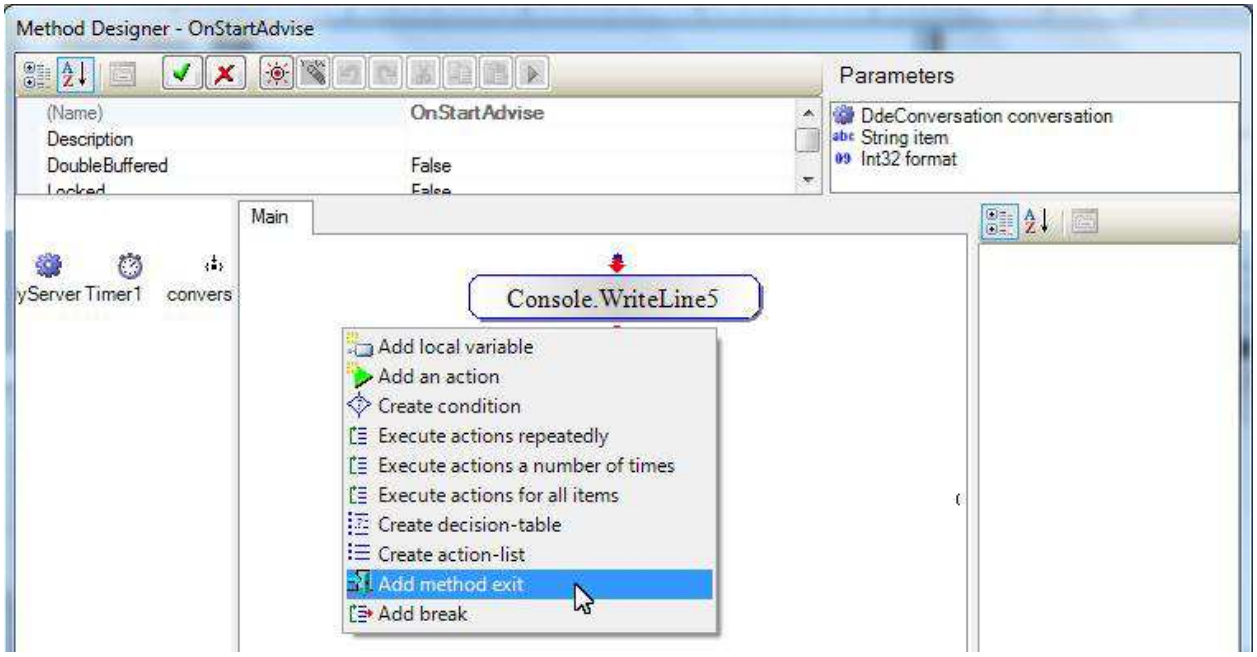

Object picker
Back
Next
Exit
Add
MyServer from DdeServer
Primary types
OnStartAdvise(DdeConversation,String,Int32)Boolean
Actions
conversation
item
format
MyServer.Service
DdeConversation conversation.Servi
DdeConversation conversation.Topic
DdeConversation conversation.Hanc
Console.WriteLine2
DdeConversation conversation.Servi
DdeConversation conversation.Topic
DdeConversation conversation.Hanc
Console.WriteLine3

Click OK:

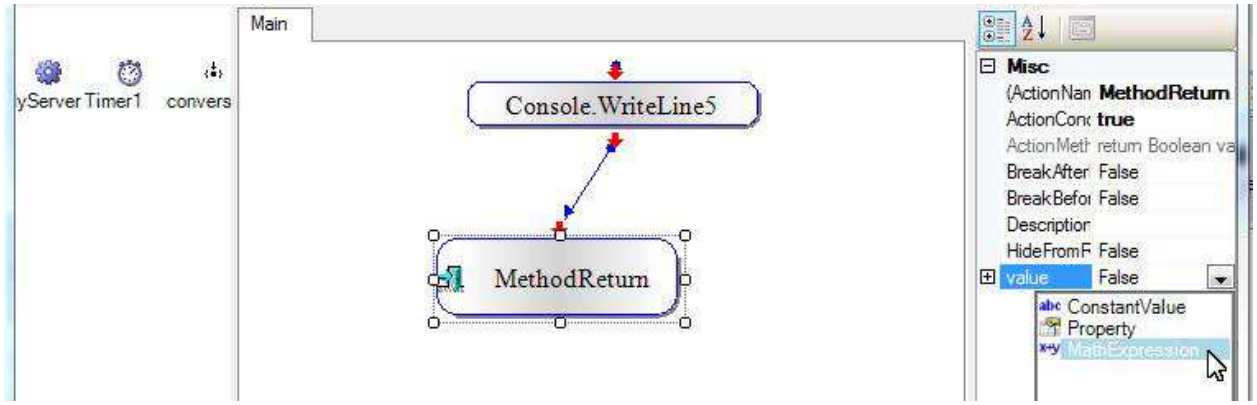


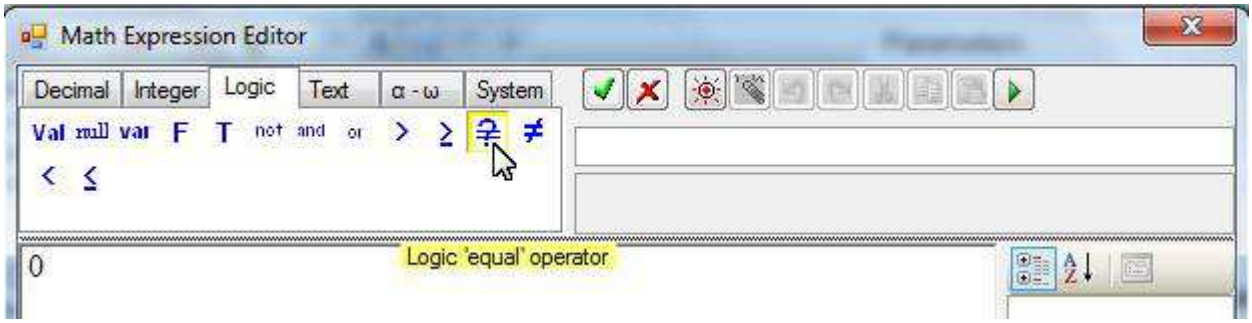


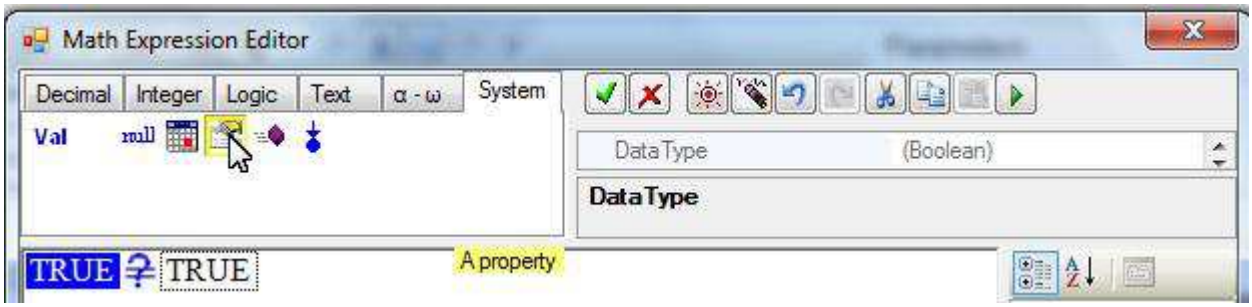
According to NDDE sample, for OnStartAdvice, the method returns true if the format parameter is 1. We add a method return action to do it:

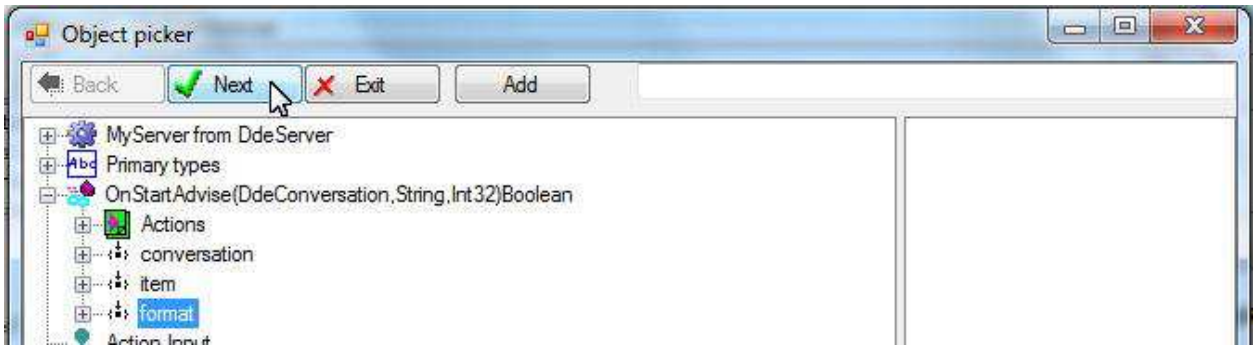


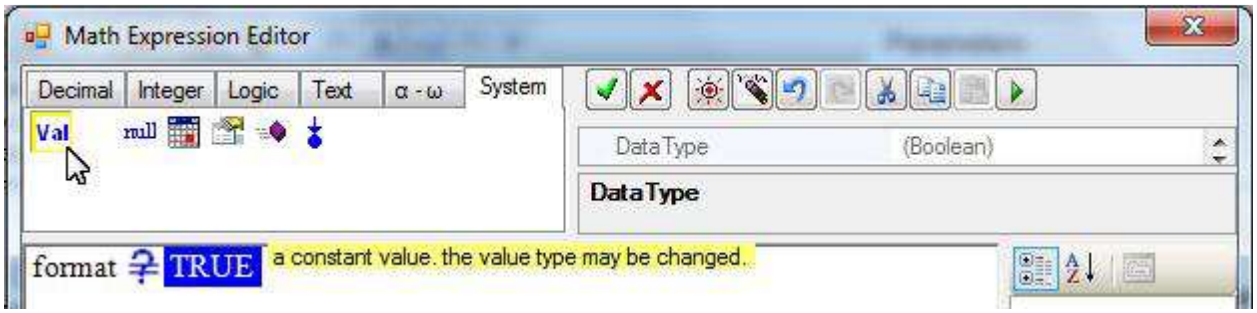
Set the return value to an expression:

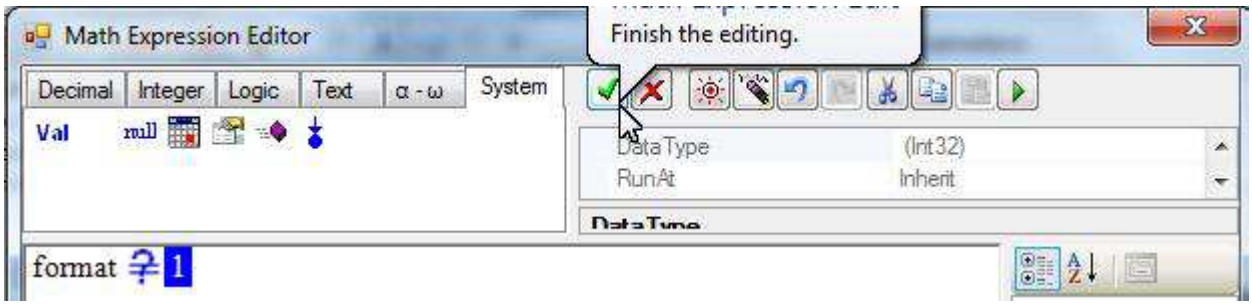


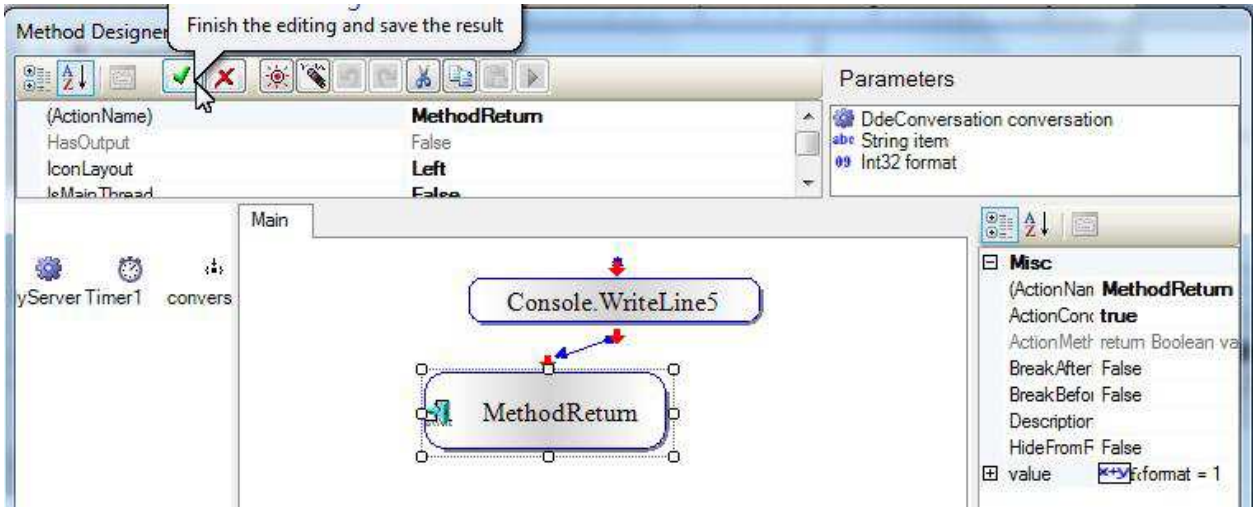



Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Boolean)
DataType
TRUE
TRUE
A property

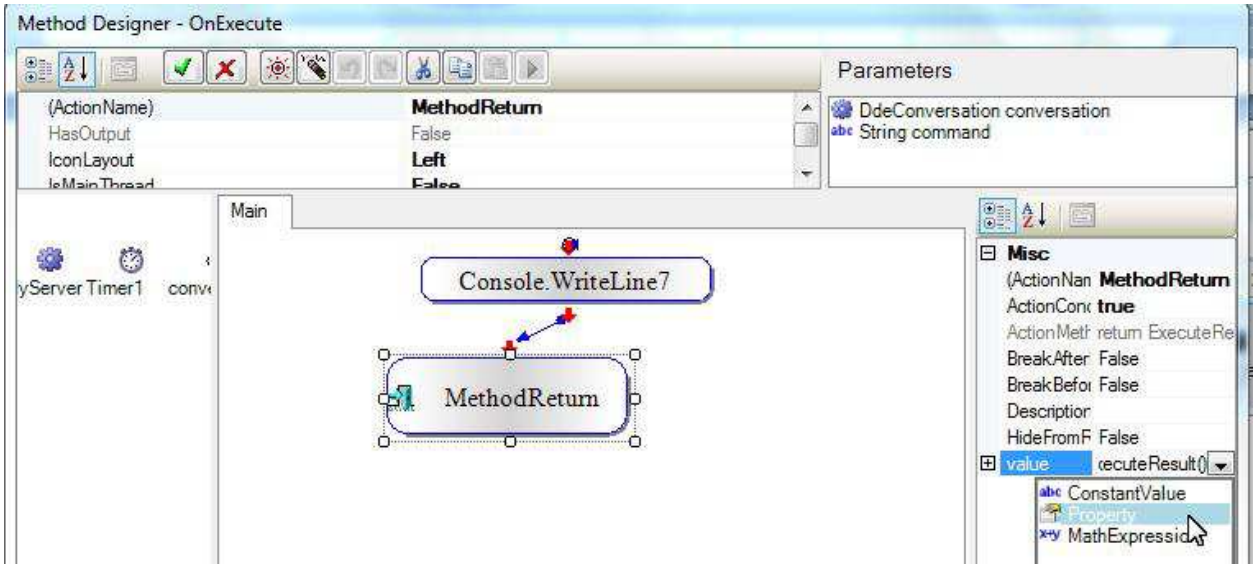

Object picker
Back
Next
Exit
Add
MyServer from DdeServer
Primary types
OnStartAdvise(DdeConversation,String,Int32)Boolean
Actions
conversation
item
format

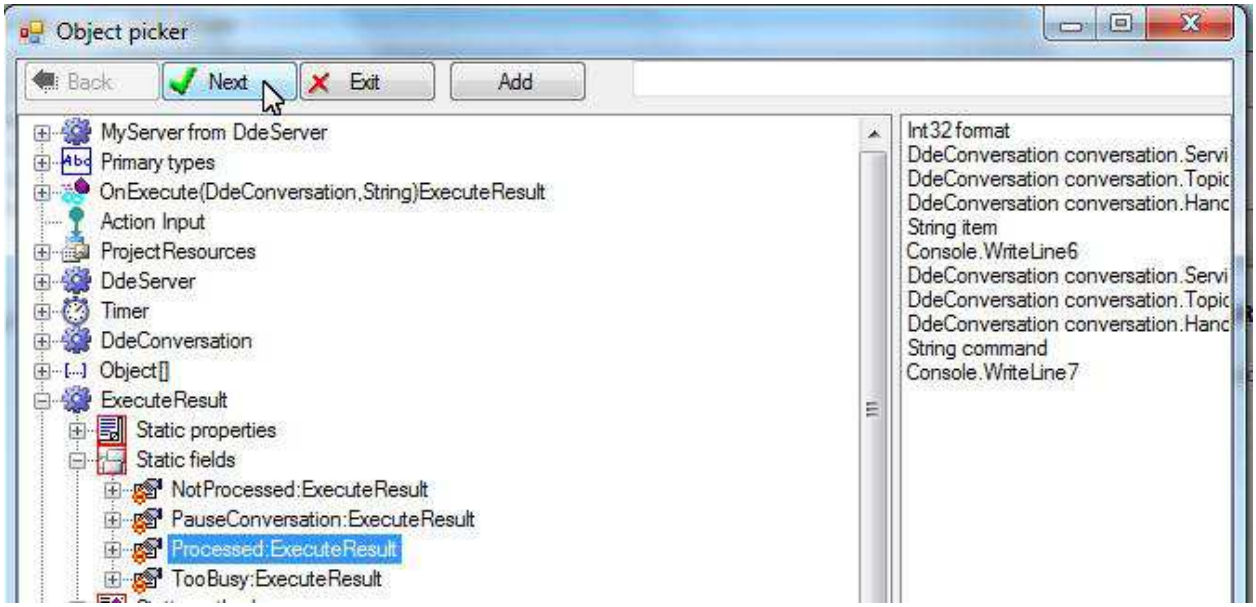

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Boolean)
DataType
format
TRUE
a constant value. the value type may be changed.

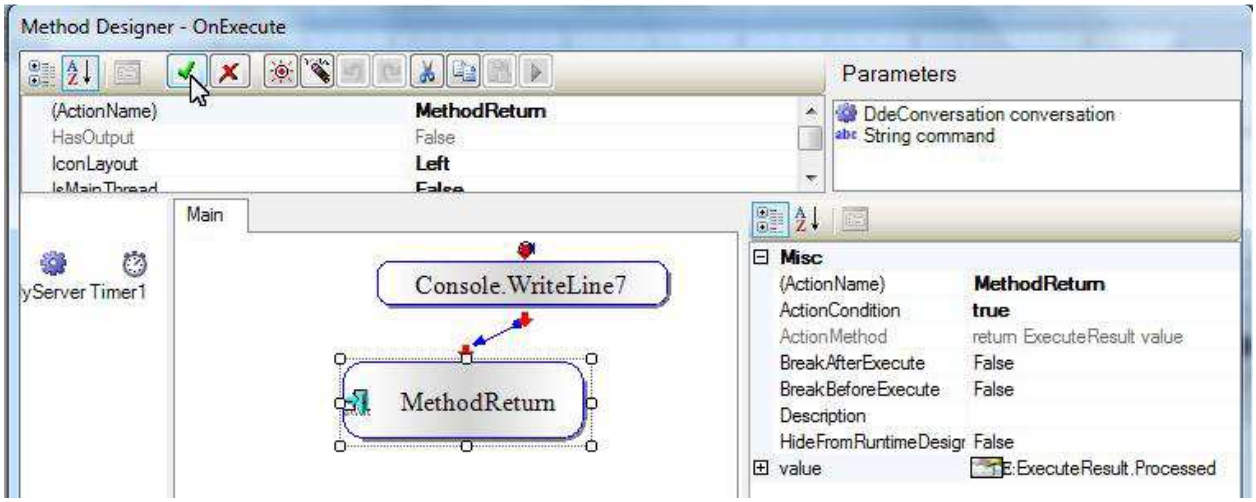

Math Expression Editor
Finish the editing.
Decimal
Integer
Logic
Text
α - ω
System
Val
null
DataType
(Int32)
RunAt
Inherit
format
1



When overriding OnExecute, it returns ExecuteResult.Processed. It is done as shown below:



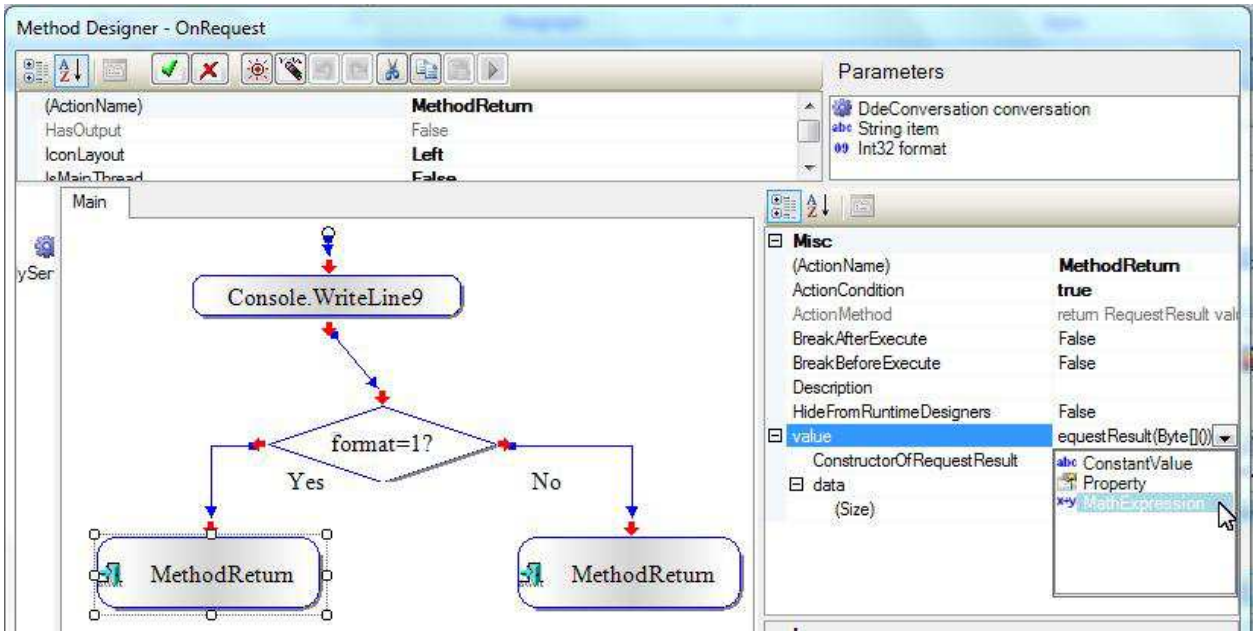


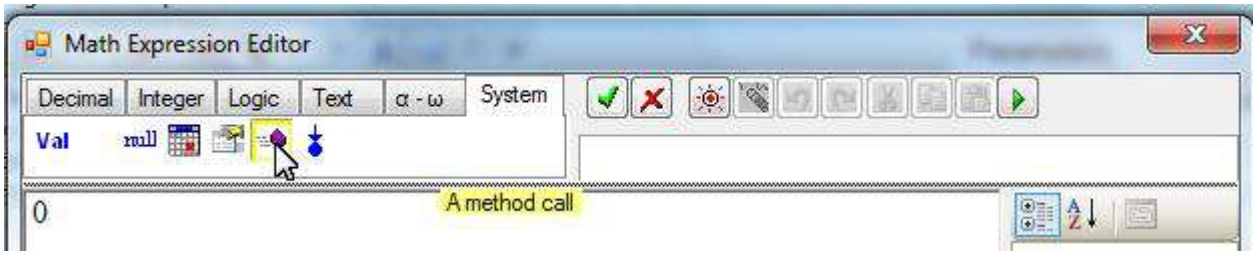


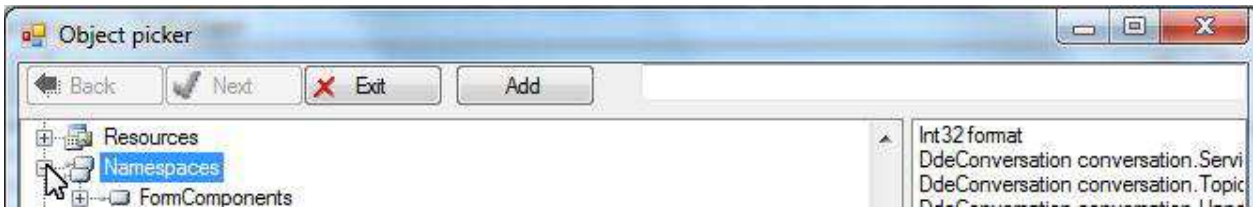
When overriding OnRequest, the NDDE sample uses the following C# code for the method return:

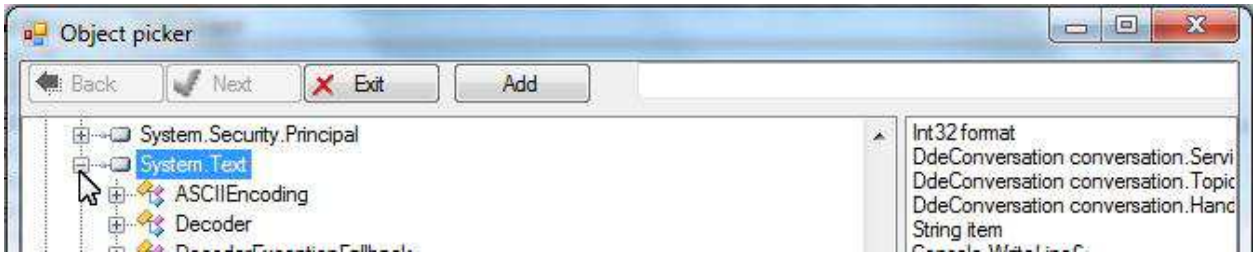
```
return new RequestResult(System.Text.Encoding.ASCII.GetBytes("Time=" +
DateTime.Now.ToString() + "\0"));
```

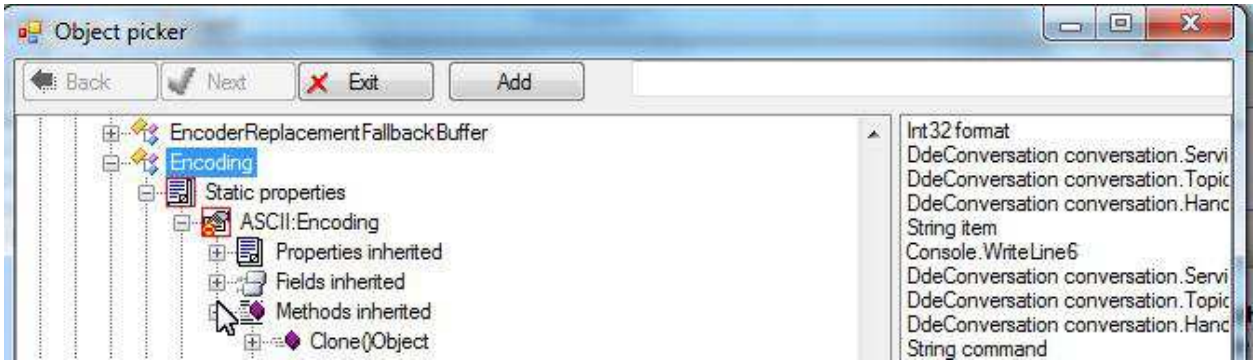
We re-create the above code in Limnor Studio as shown below.

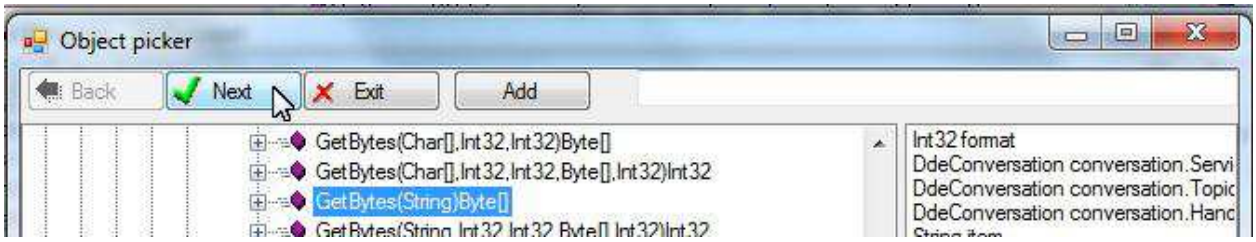

Method Designer - OnRequest
Parameters
(ActionName)
MethodReturn
HasOutput
False
IconLayout
Left
DdeConversation conversation
String item
Int32 format
Main
Console.WriteLine9
format=1?
Yes
No
MethodReturn
MethodReturn
Misc
(ActionName)
MethodReturn
ActionCondition
true
BreakAfterExecute
False
BreakBeforeExecute
False
Description
HideFromRuntimeDesigners
False
value
ConstructorOfRequestResult
data
(Size)
ConstantValue
Property


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
A method call
0

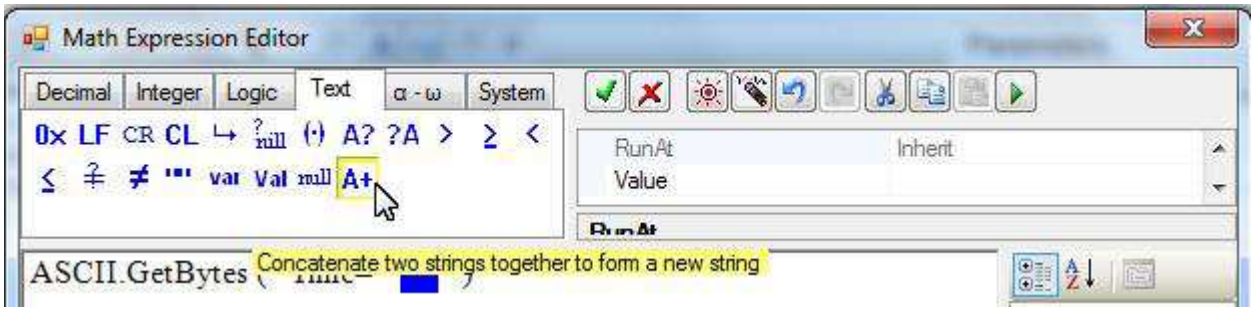

Object picker
Back
Next
Exit
Add
Resources
Namespaces
FormComponents
Int32 format
DdeConversation conversation.Servi
DdeConversation conversation.Topic

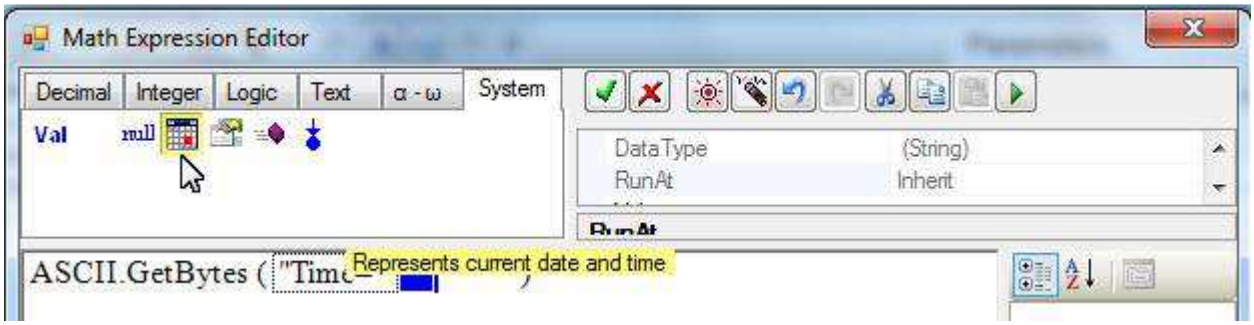

Object picker
Back
Next
Exit
Add
System.Security.Principal
System.Text
ASCIIEncoding
Decoder
Int32 format
DdeConversation conversation.Servi
DdeConversation conversation.Topic
DdeConversation conversation.Hanc
String item

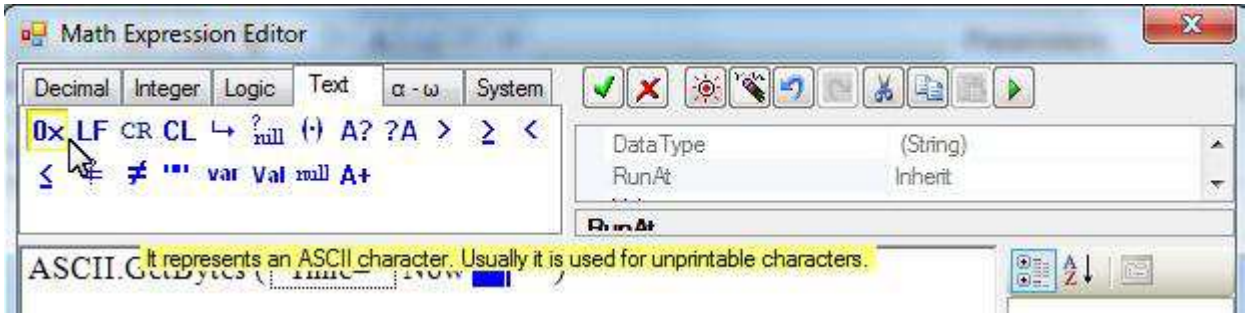

Object picker
Back
Next
Exit
Add
EncoderReplacementFallbackBuffer
Encoding
Static properties
ASCII:Encoding
Properties inherited
Fields inherited
Methods inherited
Clone()Object
Int32 format
DdeConversation conversation.Servi
DdeConversation conversation.Topic
DdeConversation conversation.Hanc
String item
Console.WriteLine6
DdeConversation conversation.Servi
DdeConversation conversation.Topic
DdeConversation conversation.Hanc
String command

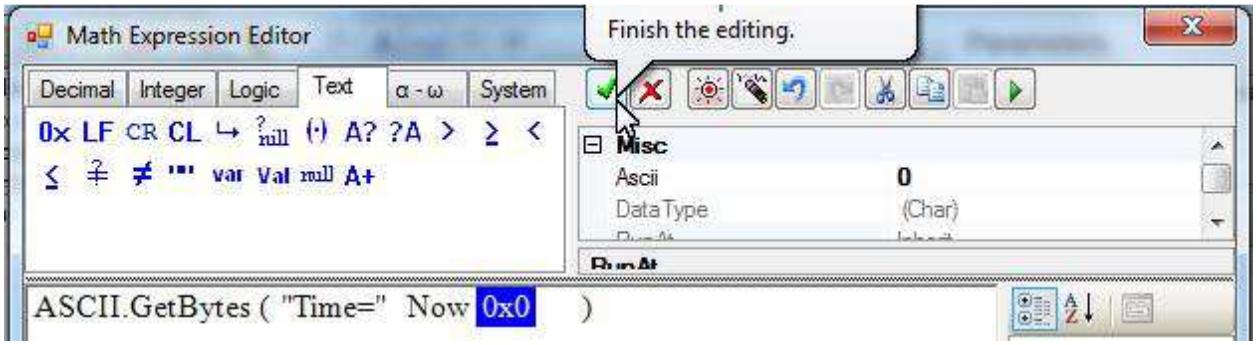

Object picker
Back
Next
Exit
Add
GetBytes(Char[],Int32,Int32)Byte[]
GetBytes(Char[],Int32,Int32,Byte[],Int32)Int32
GetBytes(String)Byte[]
Int32 format
DdeConversation conversation.Servi
DdeConversation conversation.Topic
DdeConversation conversation.Hanc

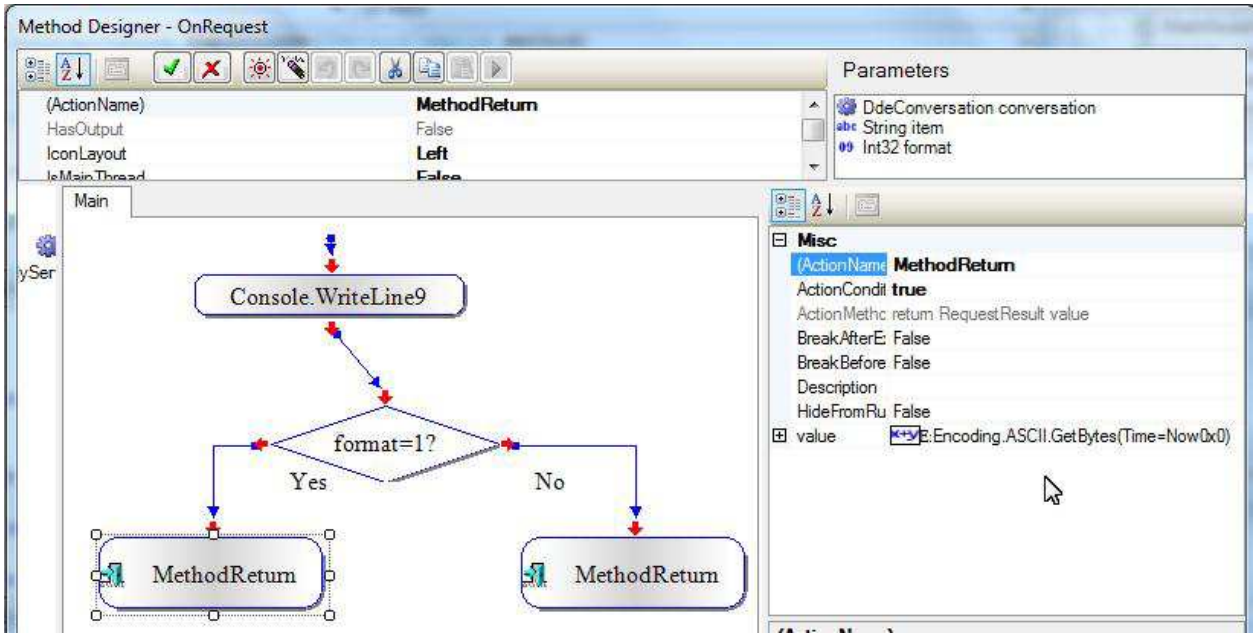

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Name
Name
ASCII.GetBytes (
Concatenate two strings together to form a new string


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
RunAt
Inherit
Value
ASCII.GetBytes (
Concatenate two strings together to form a new string


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
DataType
(String)
RunAt
Inherit
ASCII.GetBytes (
Represents current date and time

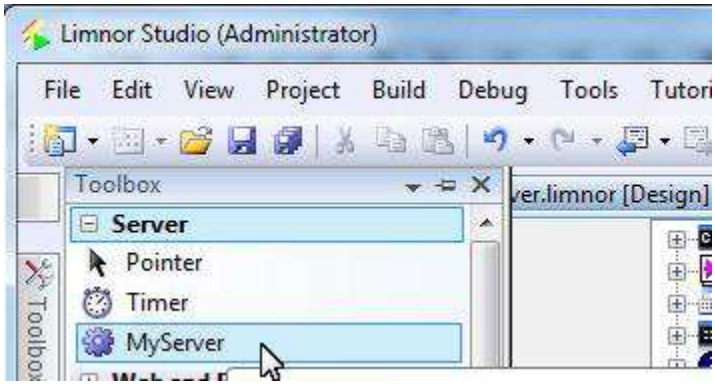




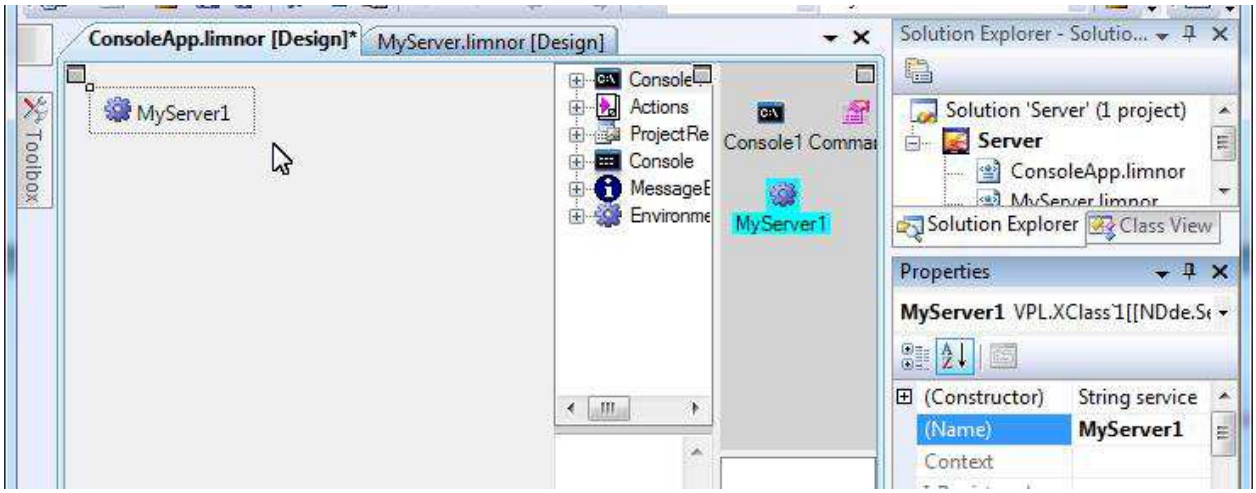


User DDE Server

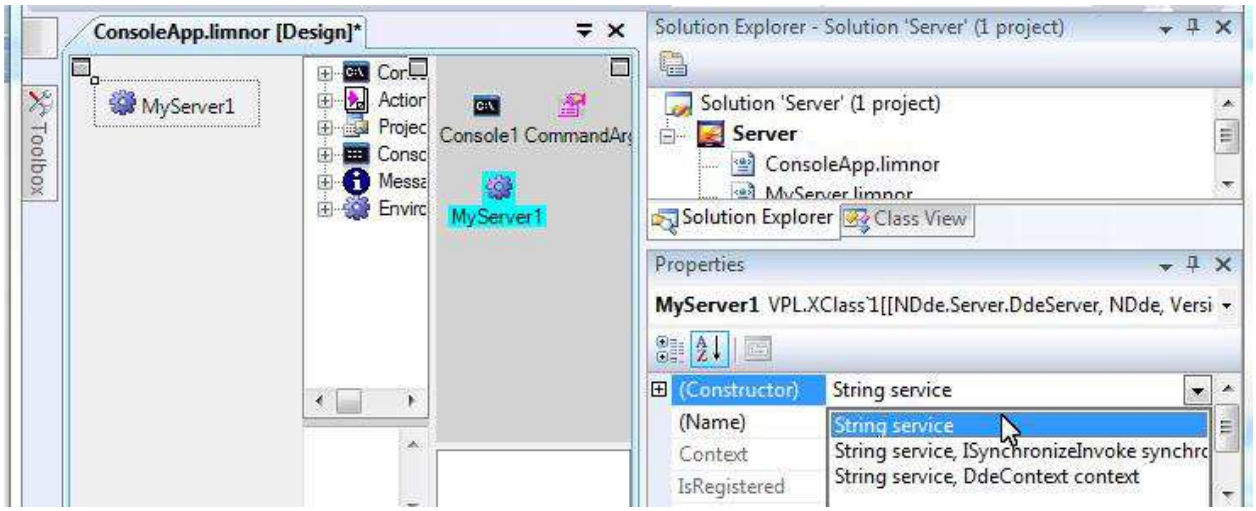
We can find the new classes in the Toolbox:



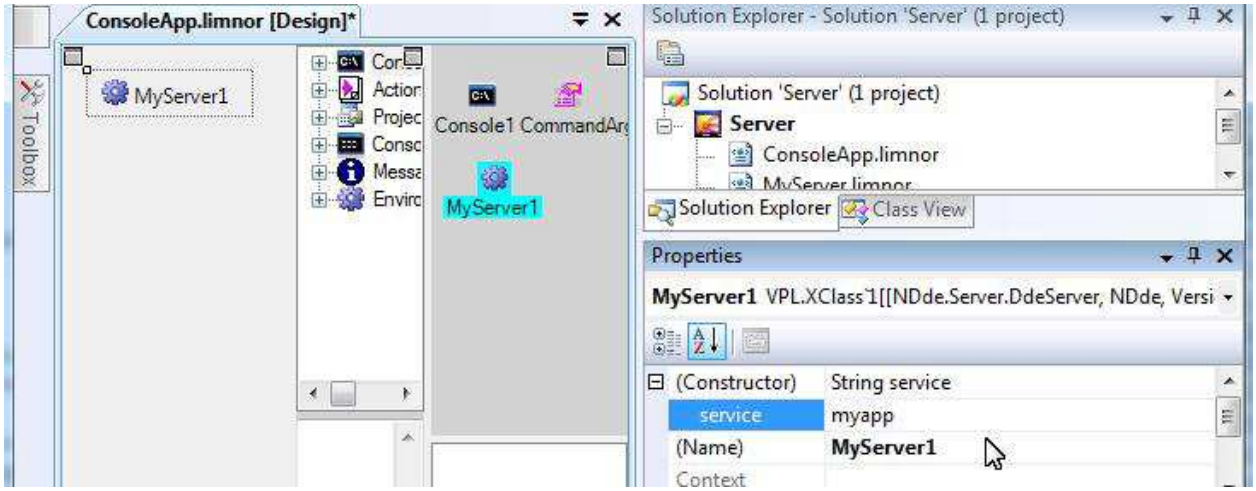
Drop MyServer from the toolbox to the console application class to create an instance of MyServer:



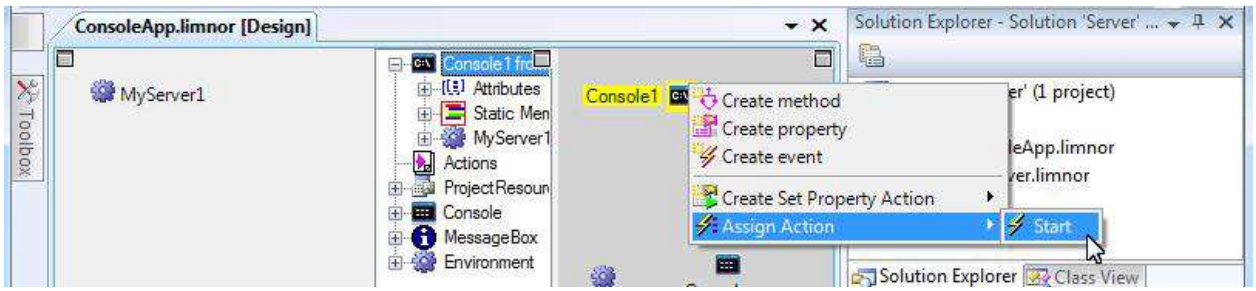
We need to select a constructor for the object MyServer1:



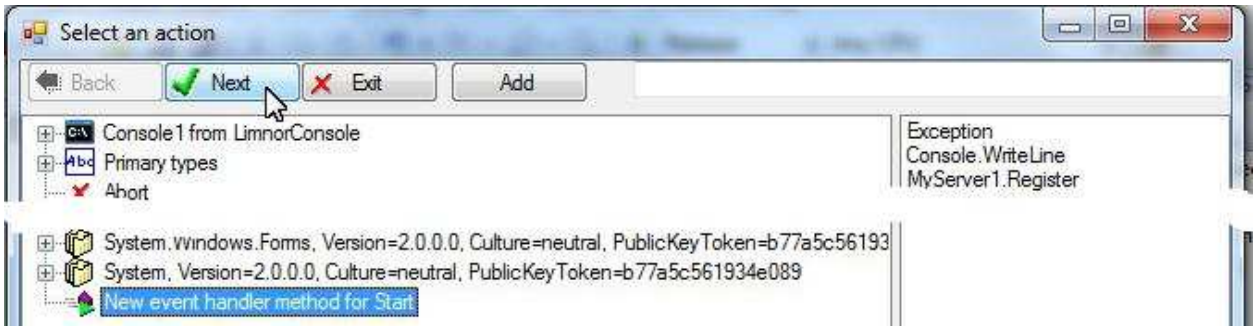
Provide parameter value for selected constructor:



We create an event handler for Start event of the console application:

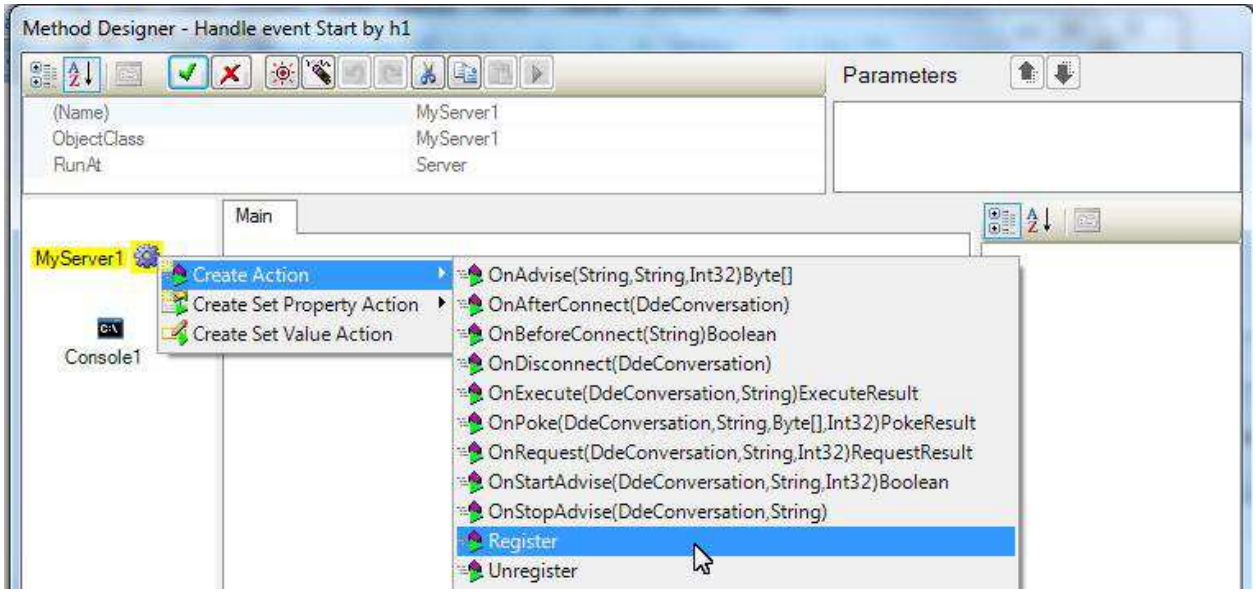


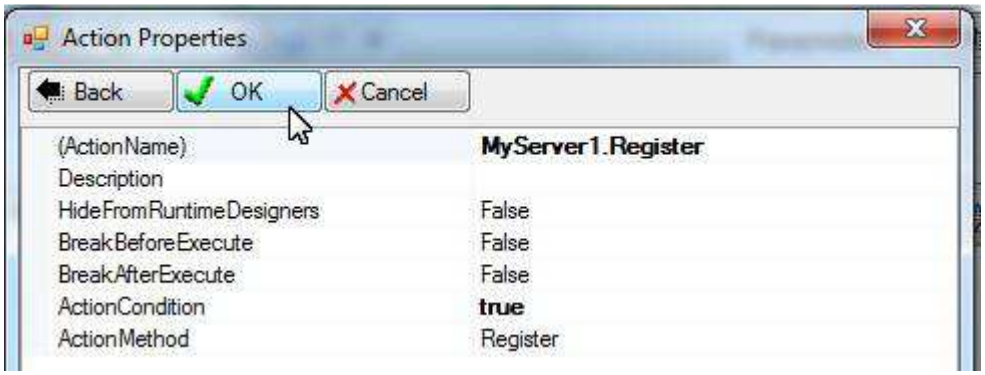
Select “New event handler for Start”:

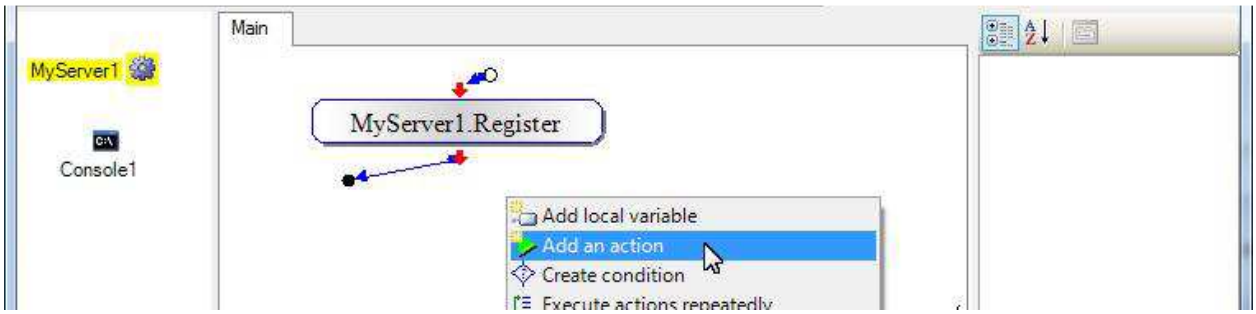


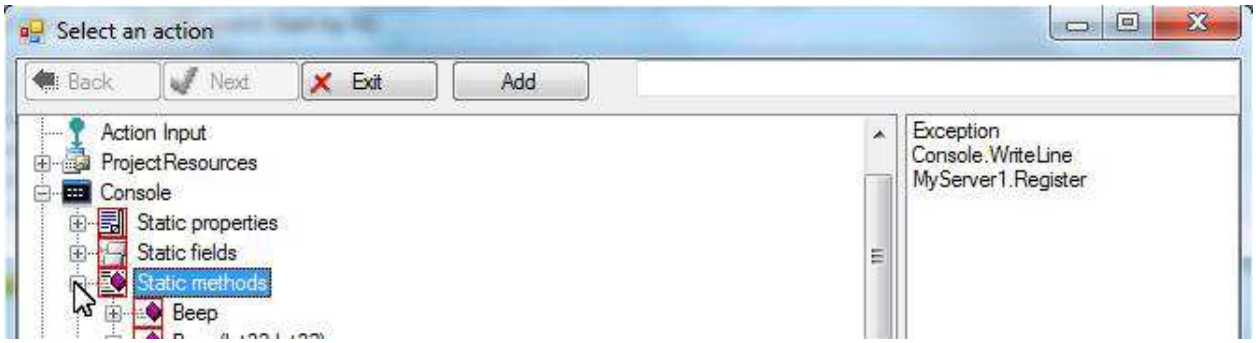
We execute following actions to handle this event:

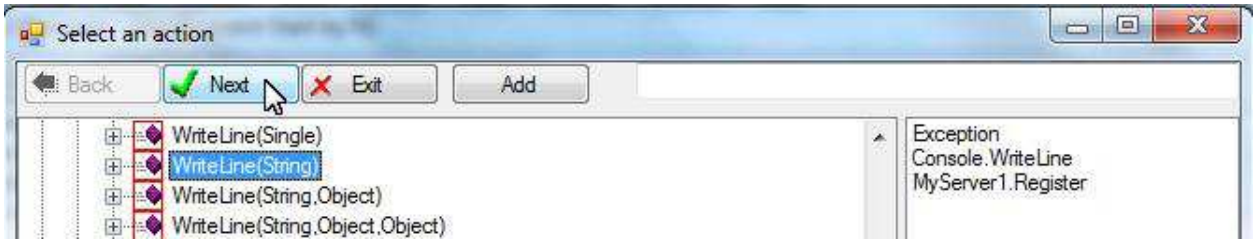
- Execute a Register action.
- Show a message on the console.
- Wait for the user to press “Enter” key to quit.

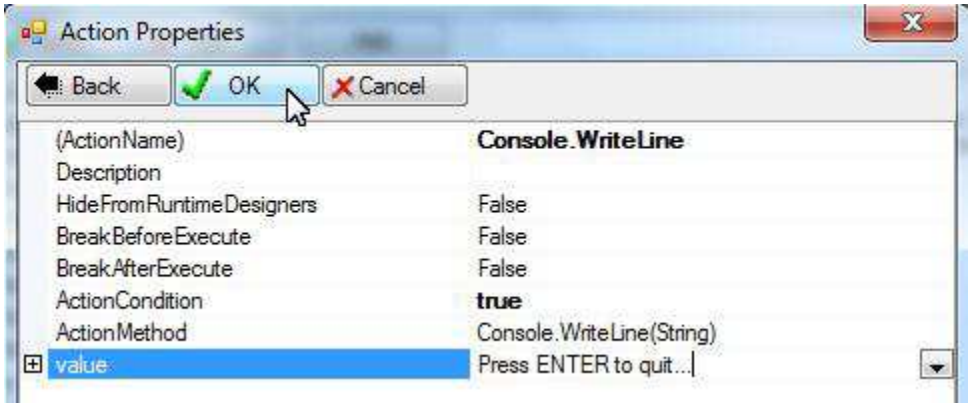

Method Designer - Handle event Start by h1
Parameters
(Name) MyServer1
ObjectClass MyServer1
RunAt Server
Main
MyServer1
Console1
Create Action
Create Set Property Action
Create Set Value Action
OnAdvise(String,String,Int32)Byte[]
OnAfterConnect(DdeConversation)
OnBeforeConnect(String)Boolean
OnDisconnect(DdeConversation)
OnExecute(DdeConversation,String)ExecuteResult
OnPoke(DdeConversation,String,Byte[],Int32)PokeResult
OnRequest(DdeConversation,String,Int32)RequestResult
OnStartAdvise(DdeConversation,String,Int32)Boolean
OnStopAdvise(DdeConversation,String)
Register
Unregister

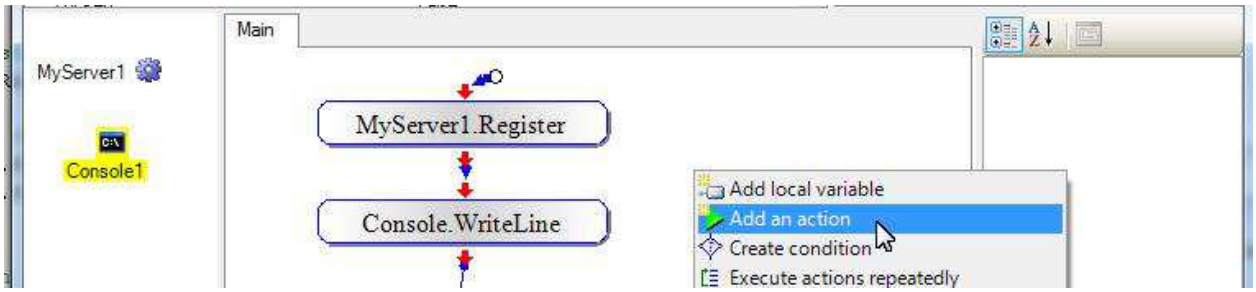

Action Properties
Back
OK
Cancel
(ActionName) MyServer1.Register
Description
HideFromRuntimeDesigners False
BreakBeforeExecute False
BreakAfterExecute False
ActionCondition true
ActionMethod Register

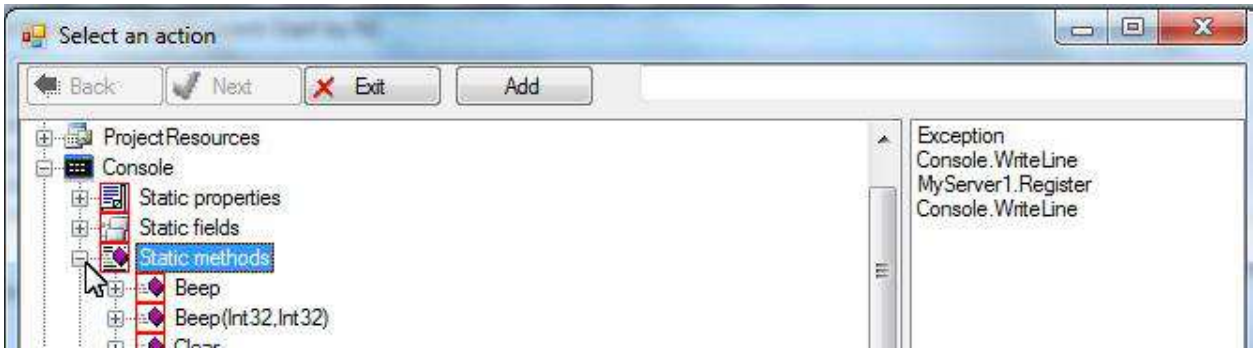

Main
MyServer1
Console1
MyServer1.Register
Add local variable
Add an action
Create condition

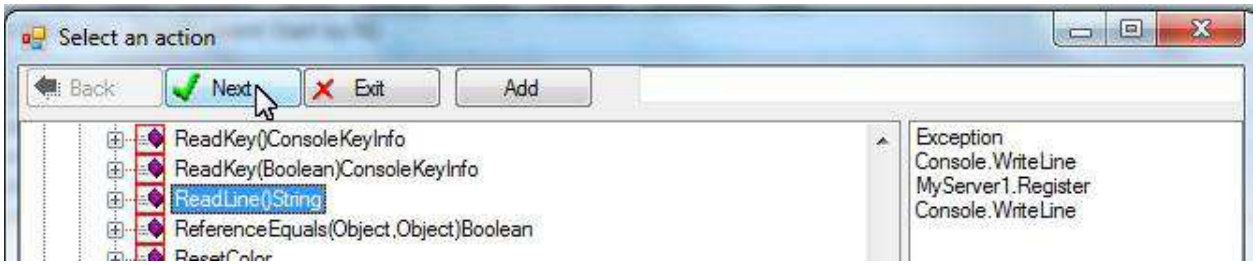

Select an action
Back
Next
Exit
Add
Action Input
ProjectResources
Console
Static properties
Static fields
Static methods
Beep
Exception
Console.WriteLine
MyServer1.Register

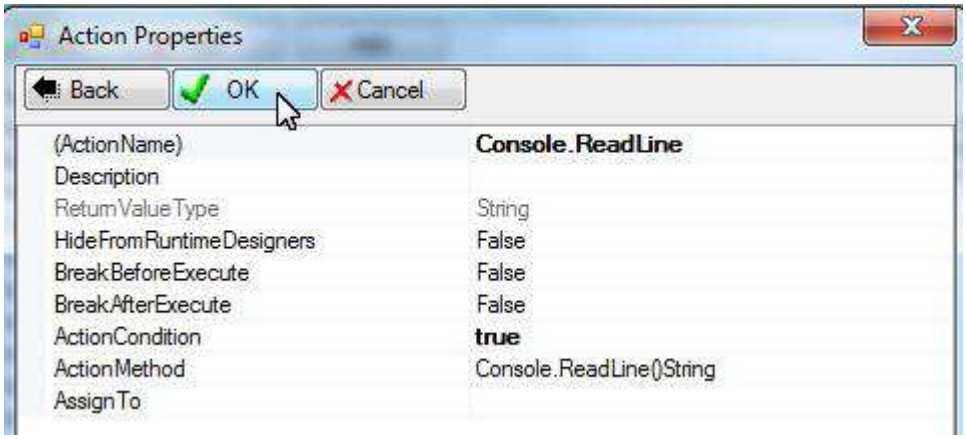

Select an action
Back
Next
Exit
Add
WriteLine(Single)
WriteLine(String)
WriteLine(String,Object)
WriteLine(String,Object,Object)
Exception
Console.WriteLine
MyServer1.Register

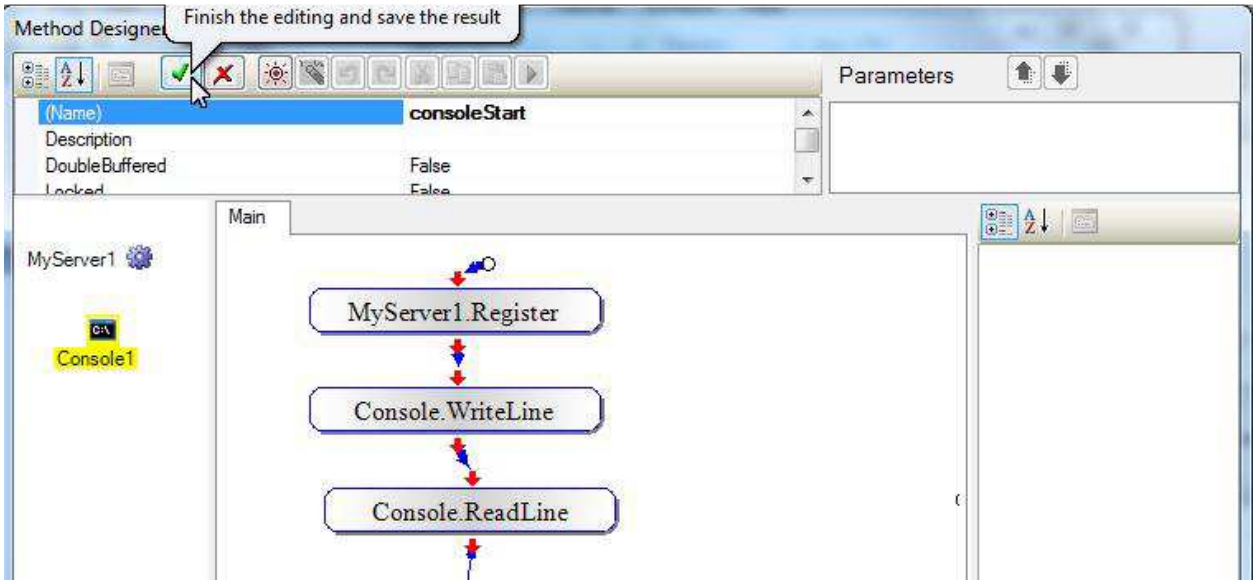

Action Properties
Back
OK
Cancel
(ActionName) Console.WriteLine
Description
HideFromRuntimeDesigners False
BreakBeforeExecute False
BreakAfterExecute False
ActionCondition true
ActionMethod Console.WriteLine(String)
value Press ENTER to quit...


Main
MyServer1
Console1
MyServer1.Register
Console.WriteLine
Add local variable
Add an action
Create condition
Execute actions repeatedly


Select an action
Back
Next
Exit
Add
ProjectResources
Console
Static properties
Static fields
Static methods
Beep
Beep(Int32,Int32)
Exception
Console.WriteLine
MyServer1.Register
Console.WriteLine

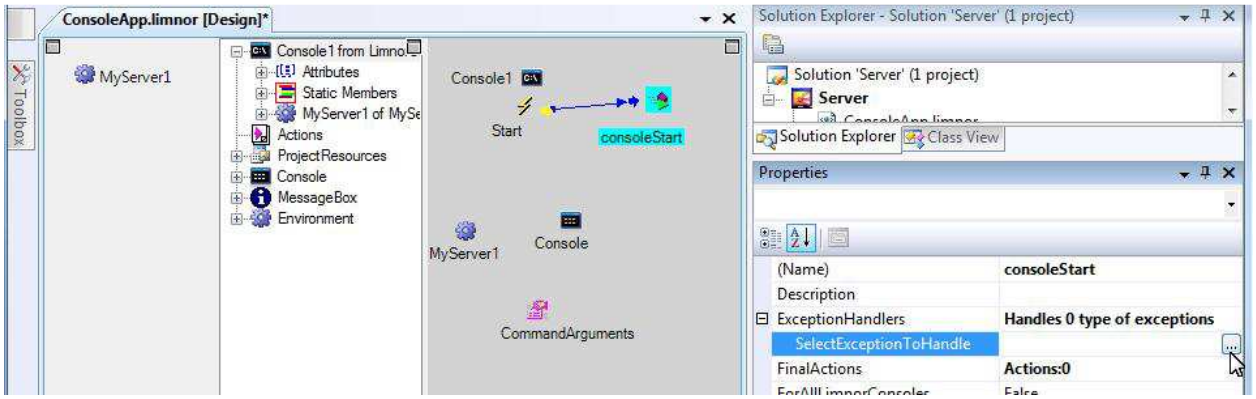

Select an action
Back
Next
Exit
Add
ReadKey()ConsoleKeyInfo
ReadKey(Boolean)ConsoleKeyInfo
ReadLine()String
ReferenceEquals(Object,Object)Boolean
Exception
Console.WriteLine
MyServer1.Register
Console.WriteLine

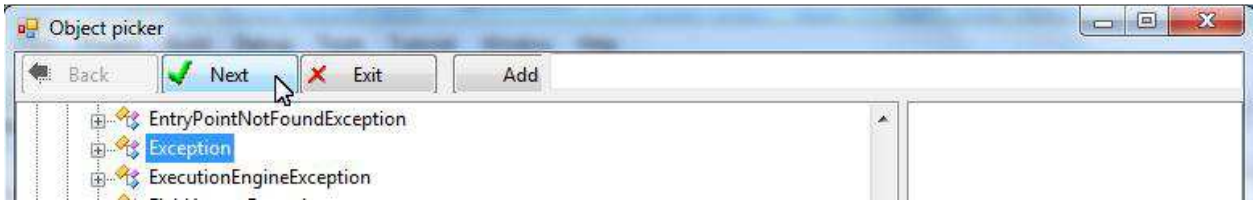




Handle Exceptions

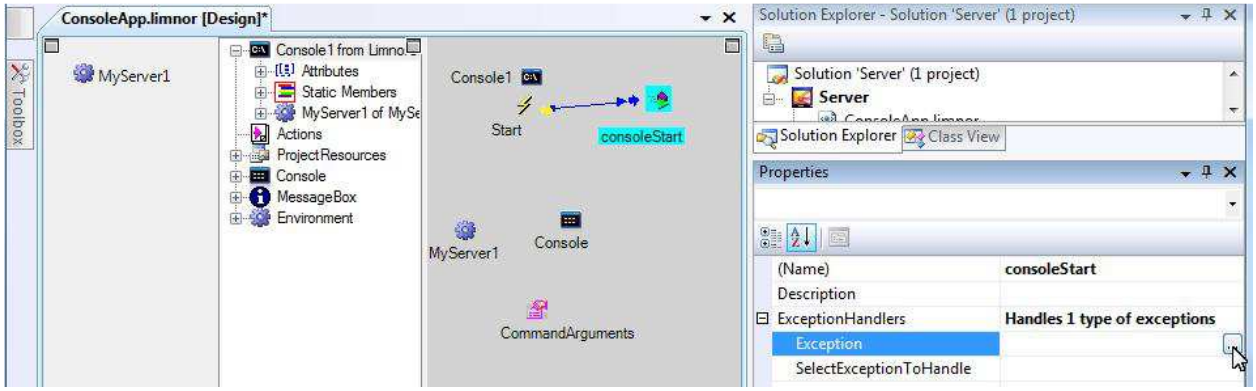
The NDDE sample handles exceptions by showing the exception on the console. We'll do the same. Select the generic exception to catch:

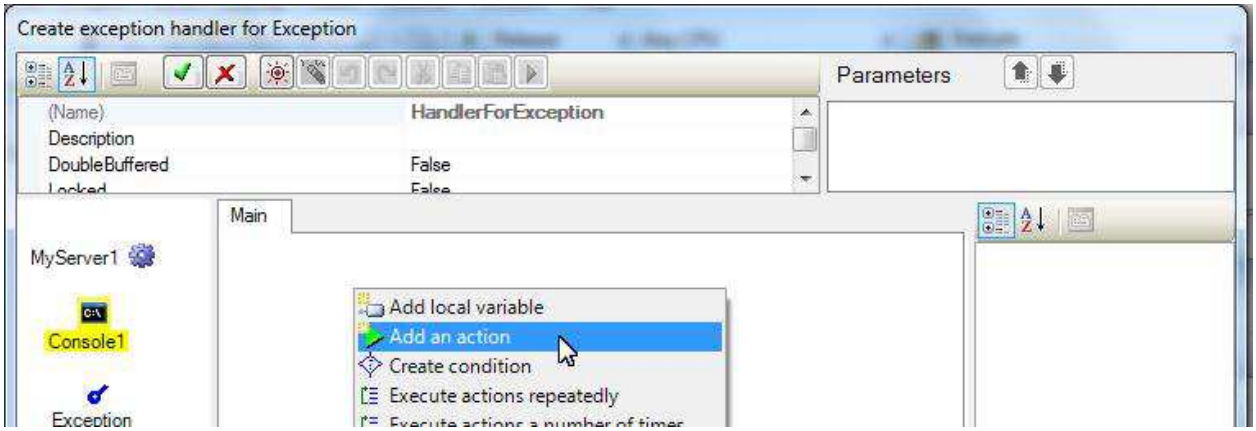


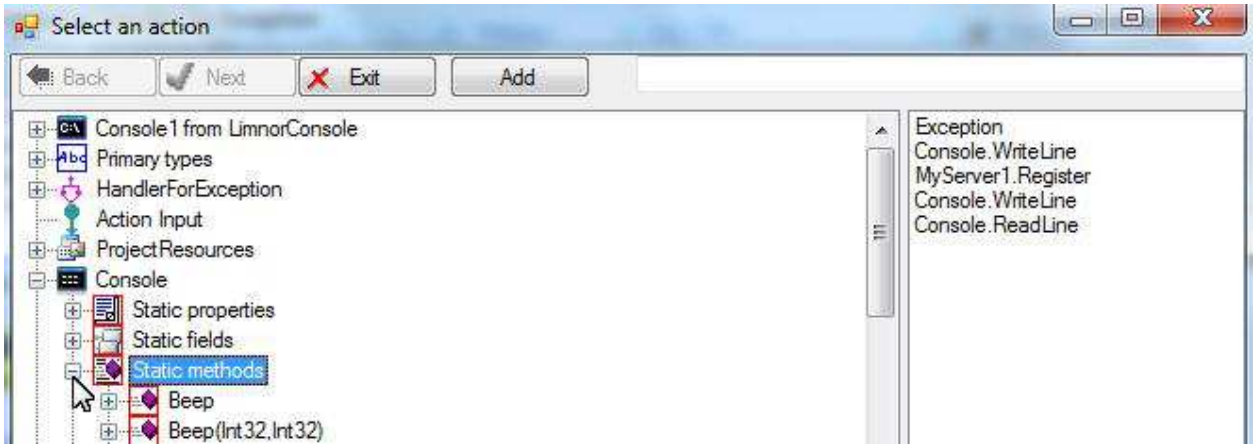


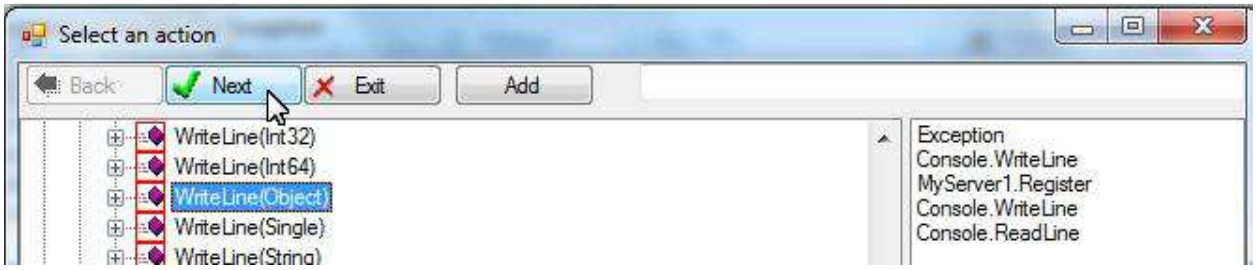
"Exception" class represents all exceptions.

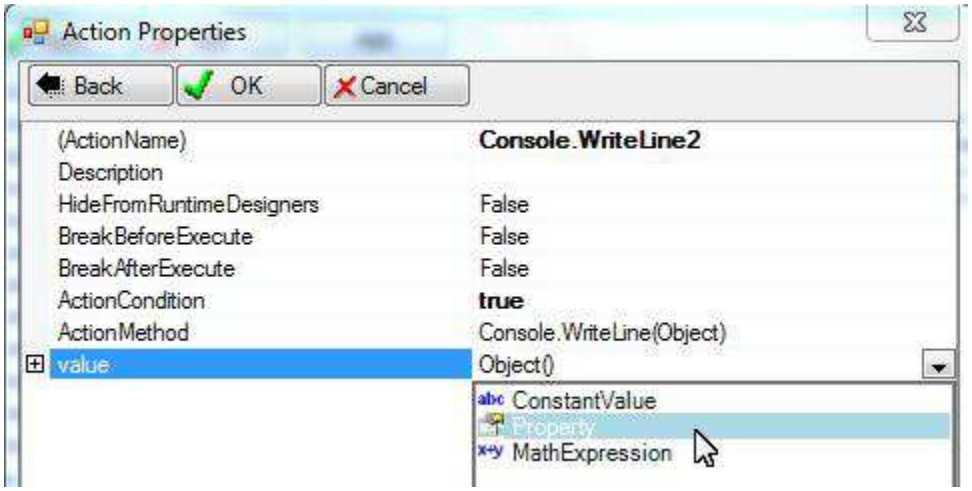
Let's add actions to handle the exception:

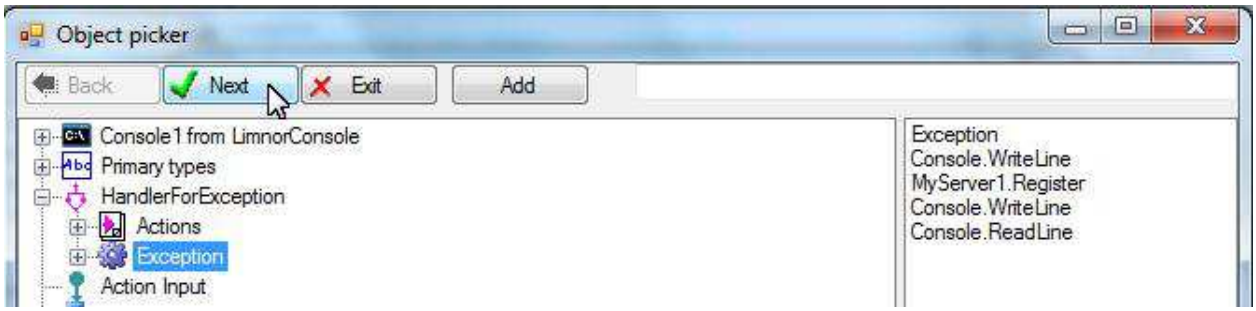


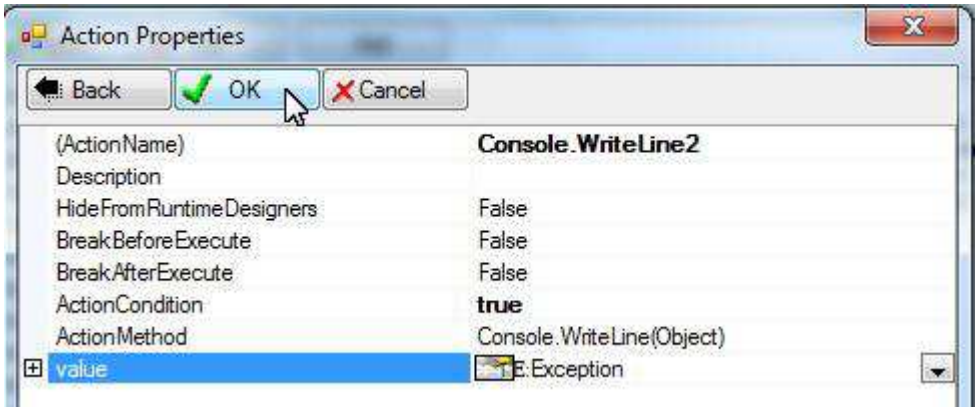


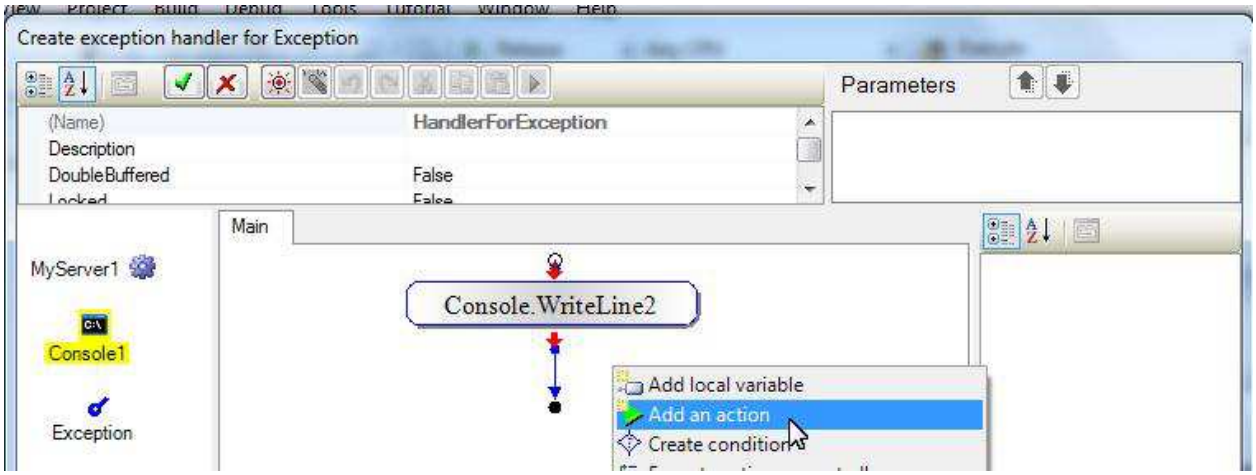

Select an action
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
HandlerForException
Action Input
Project Resources
Console
Static properties
Static fields
Static methods
Beep
Beep(Int32,Int32)
Exception
Console.WriteLine
MyServer1.Register
Console.WriteLine
Console.ReadLine

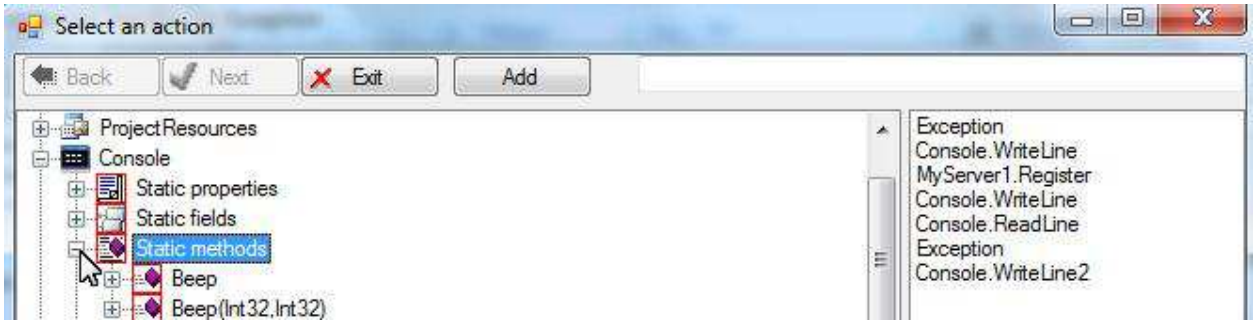

Select an action
Back
Next
Exit
Add
WriteLine(Int32)
WriteLine(Int64)
WriteLine(Object)
WriteLine(Single)
WriteLine(String)
Exception
Console.WriteLine
MyServer1.Register
Console.WriteLine
Console.ReadLine

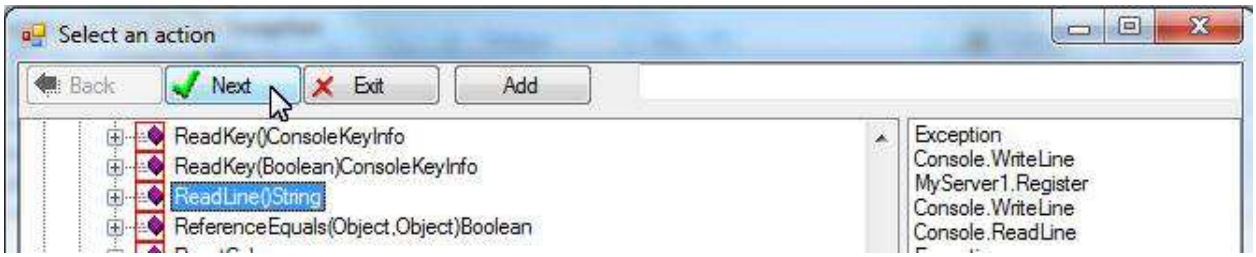

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine2
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(Object)
value
Object()
ConstantValue
Property
MathExpression

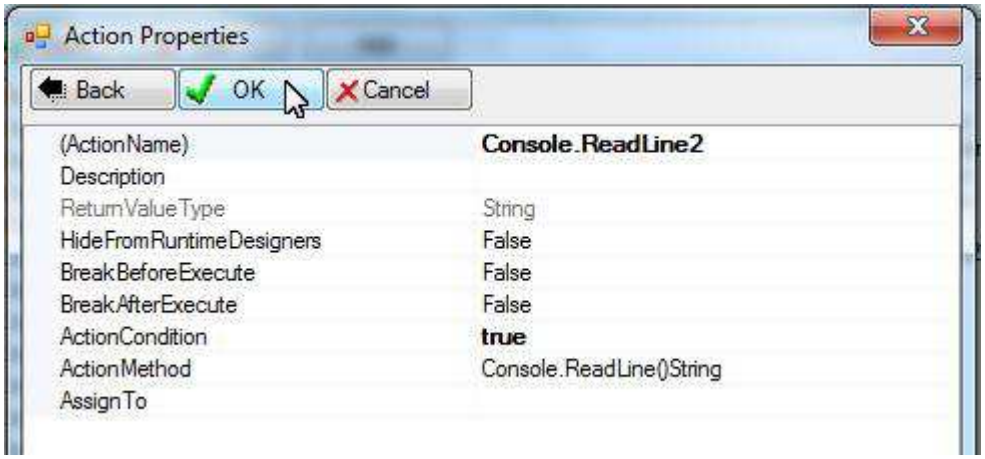

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
HandlerForException
Actions
Exception
Action Input
Exception
Console.WriteLine
MyServer1.Register
Console.WriteLine
Console.ReadLine

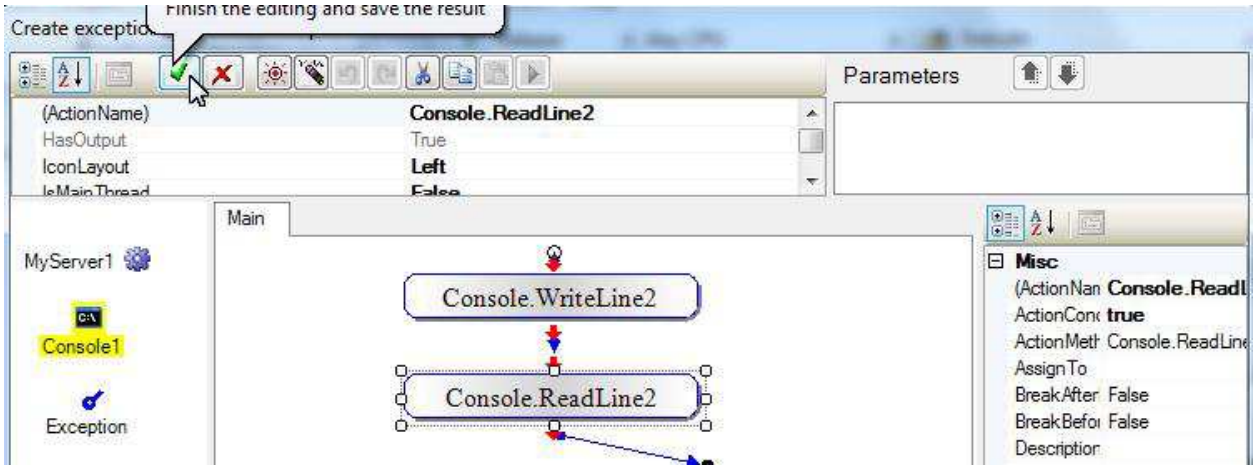

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine2
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(Object)
value
Exception


Create exception handler for Exception
Parameters
(Name)
HandlerForException
Description
DoubleBuffered
False
Main
MyServer1
Console1
Exception
Console.WriteLine2
Add local variable
Add an action
Create condition


Select an action
Back
Next
Exit
Add
ProjectResources
Console
Static properties
Static fields
Static methods
Beep
Beep(Int32,Int32)
Exception
Console.WriteLine
MyServer1.Register
Console.WriteLine
Console.ReadLine
Exception
Console.WriteLine2

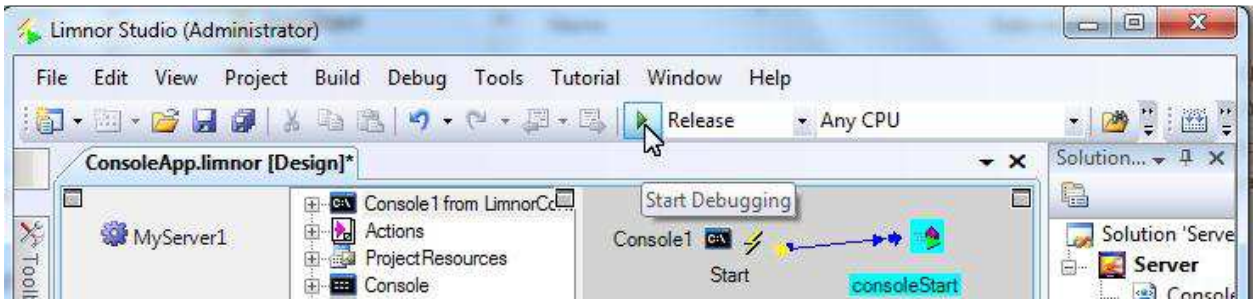

Select an action
Back
Next
Exit
Add
ReadKey()ConsoleKeyInfo
ReadKey(Boolean)ConsoleKeyInfo
ReadLine()String
ReferenceEquals(Object,Object)Boolean
Exception
Console.WriteLine
MyServer1.Register
Console.WriteLine
Console.ReadLine





Start DDE Server

We may run the project now:



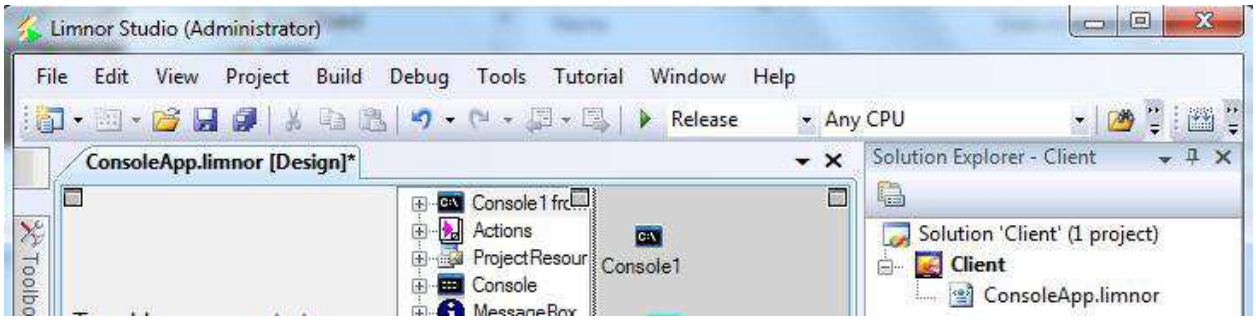
The console starts and waiting for the DDE client requests.

Create DDE Client

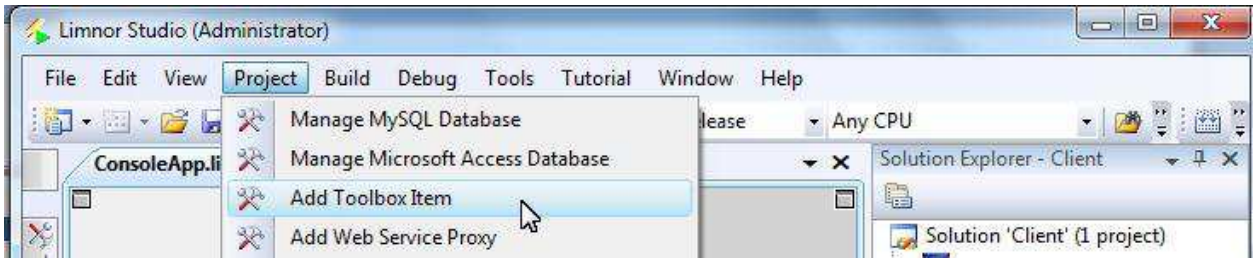
The NDDE provides a class named DdeClient acting as a data client for DDE.

Create DdeClient object

Create a console project to use DdeClient.

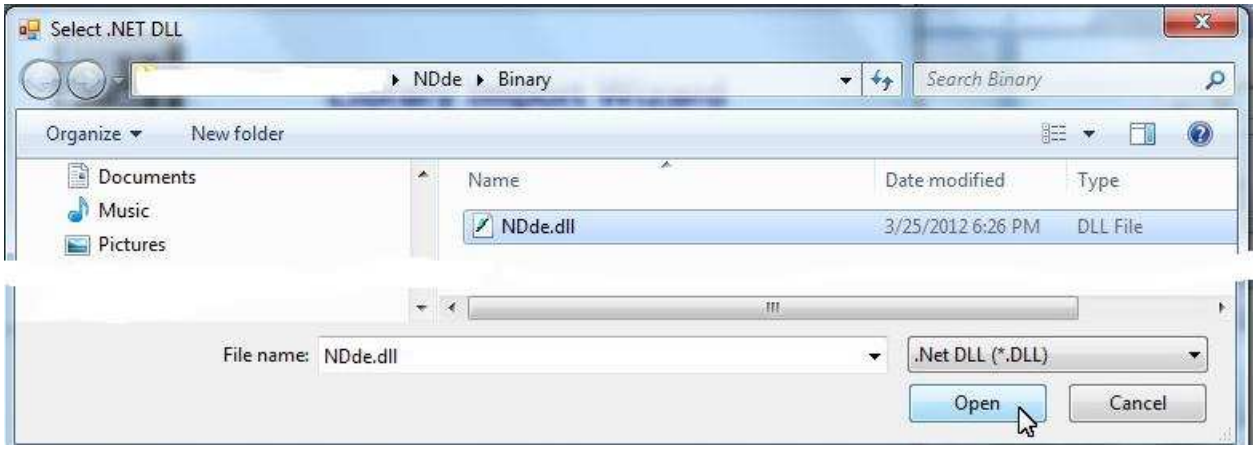


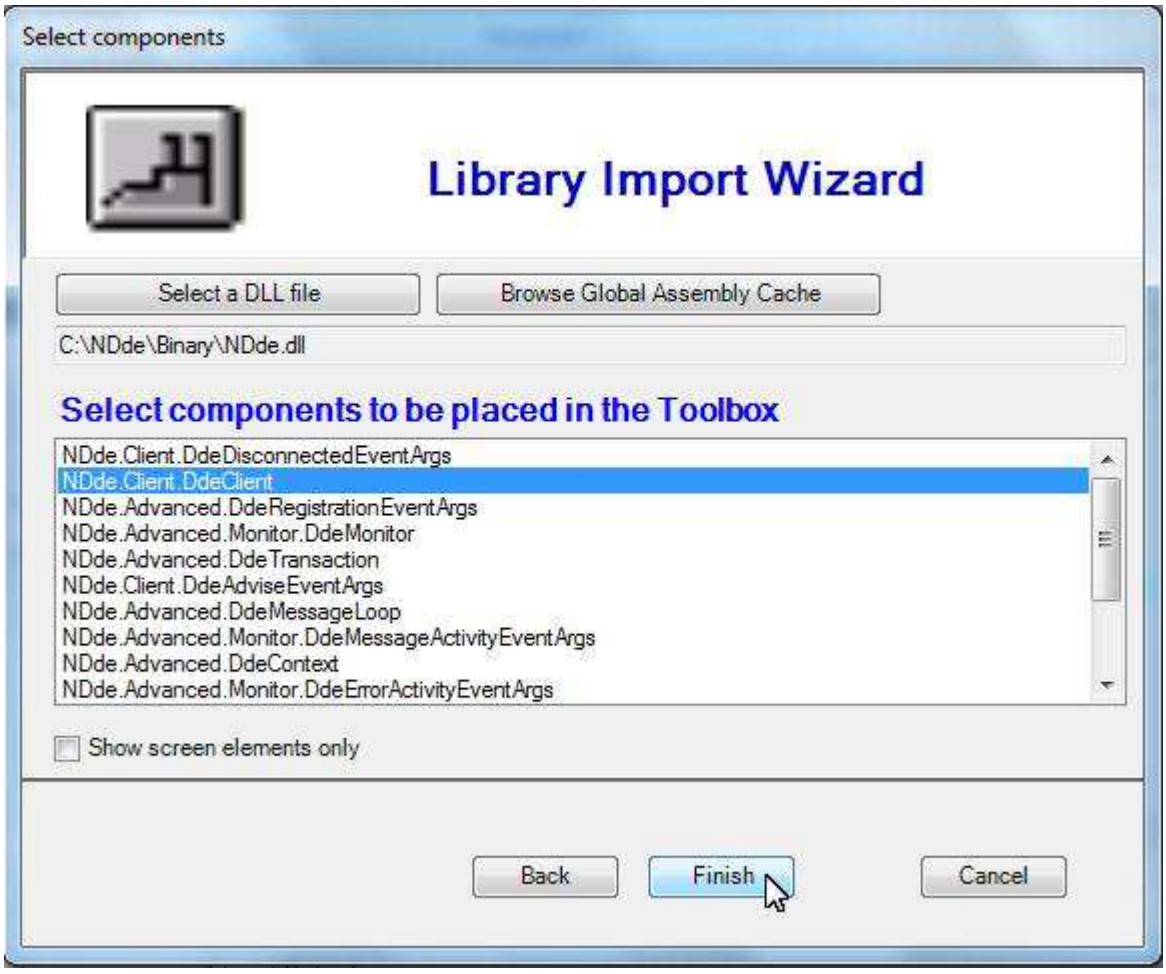
Add DdeClient to the Toolbox:



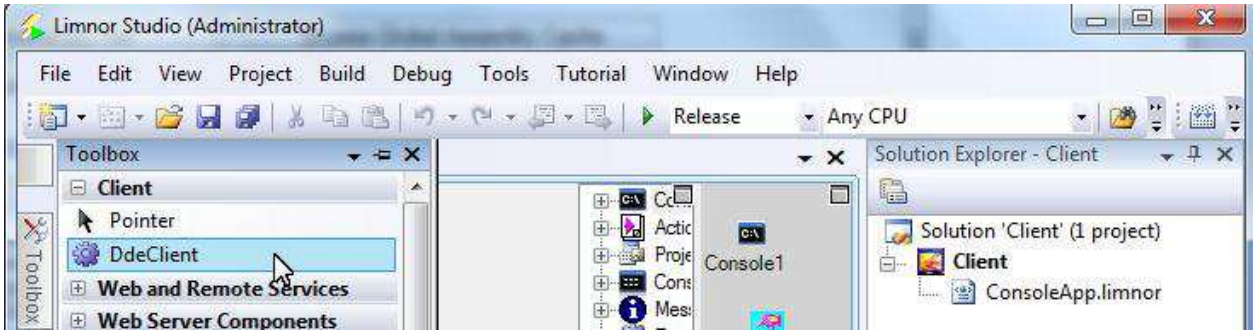

Select library type
Library Import Wizard
Import .NET classes
Import ActiveX components
Back
Next
Cancel

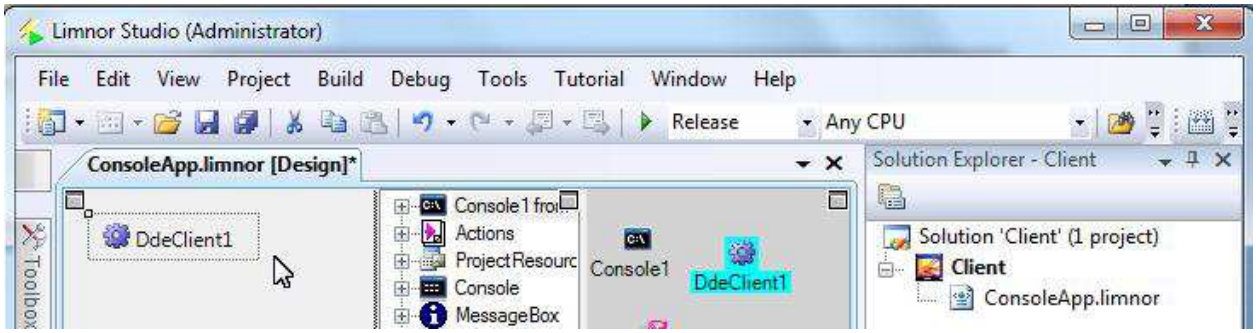

Select components
Library Import Wizard
Select a DLL file
Browse Global Assembly Cache



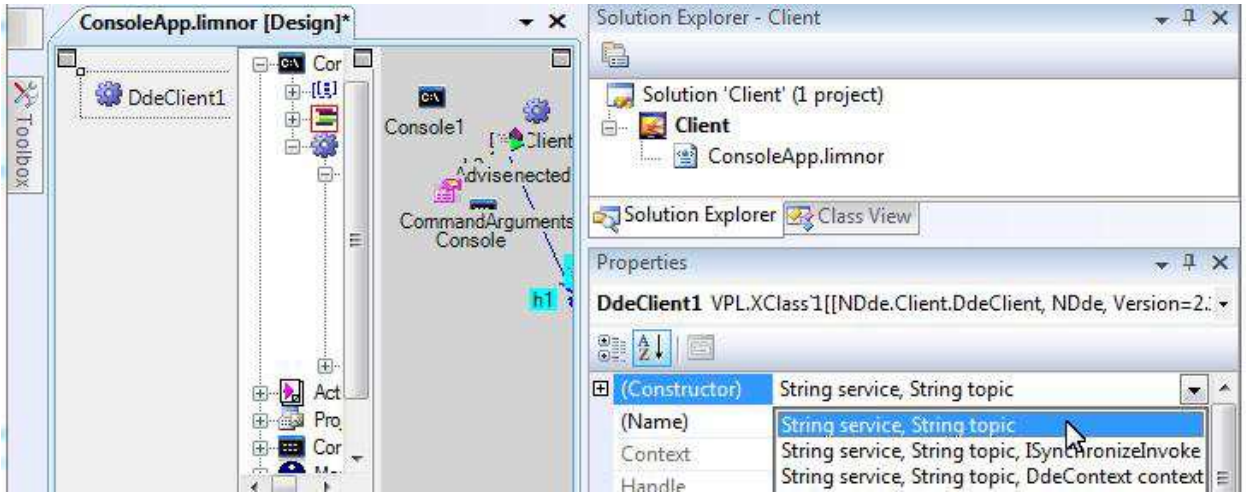


Drop DdeClient from the toolbox to the console application:

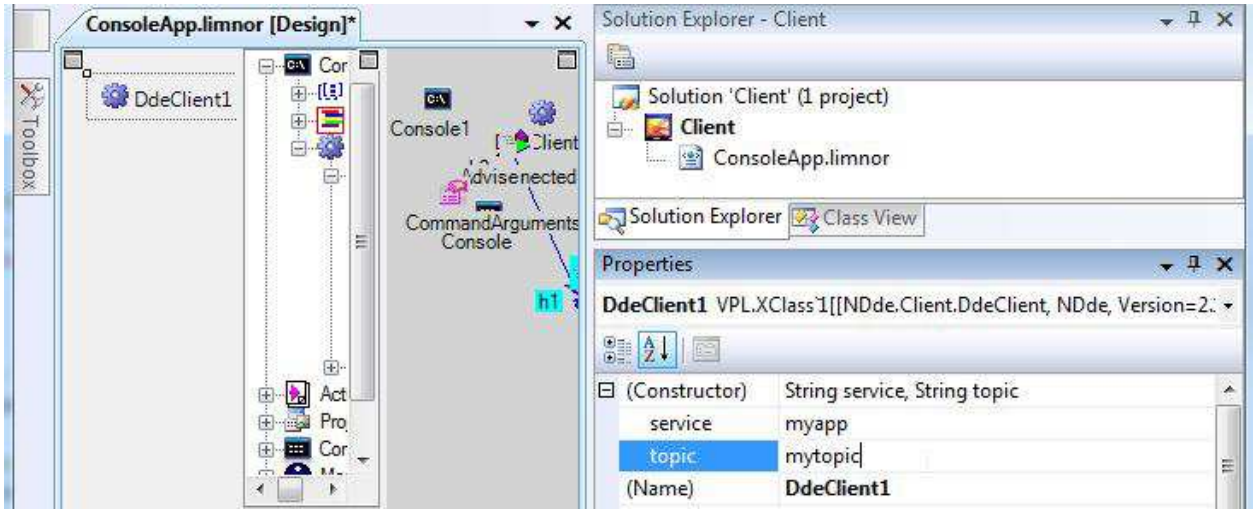




Select Constructor



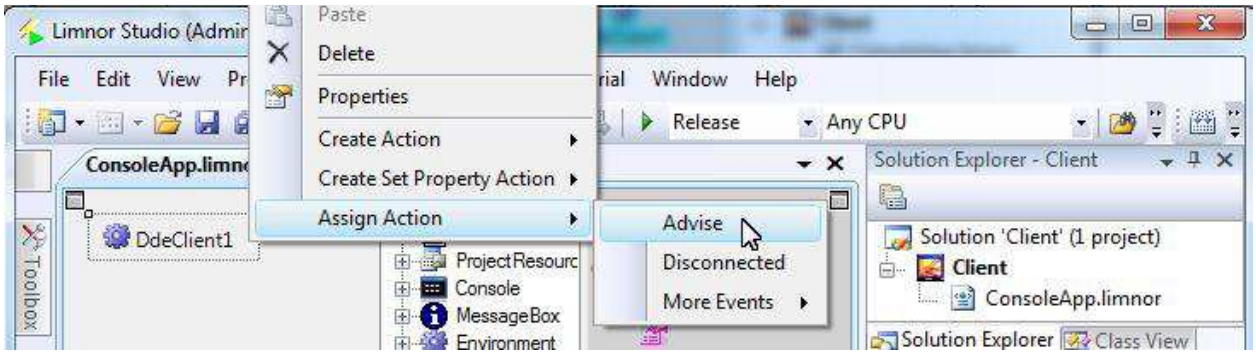
Set parameter values for selected constructor:

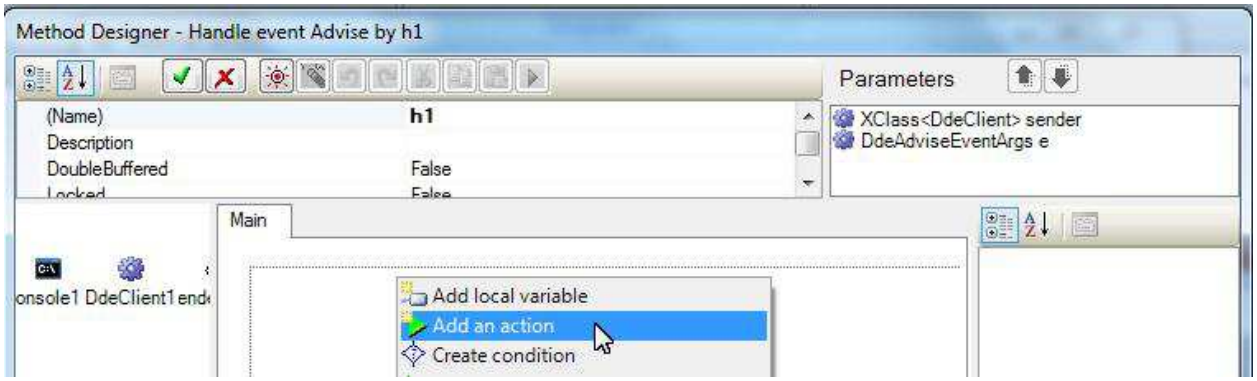


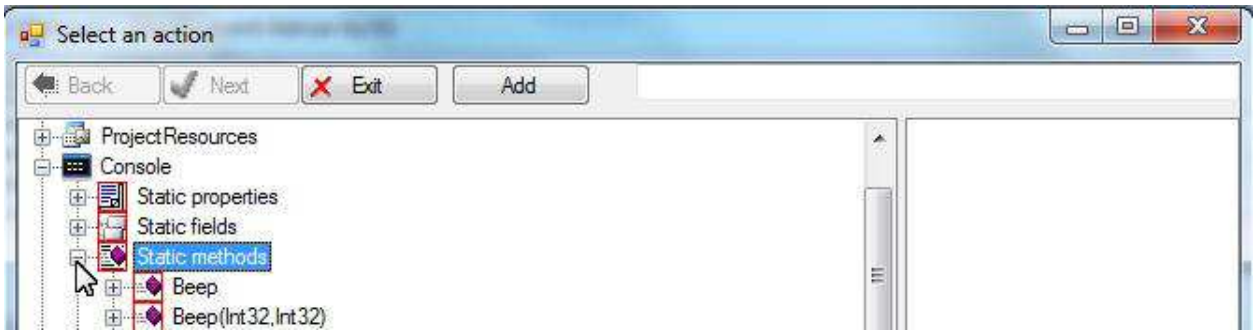
Handle DdeClient events

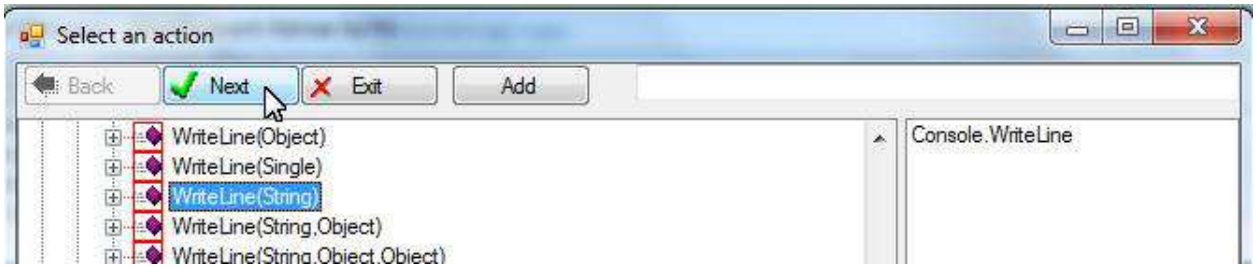
Handle Advice event

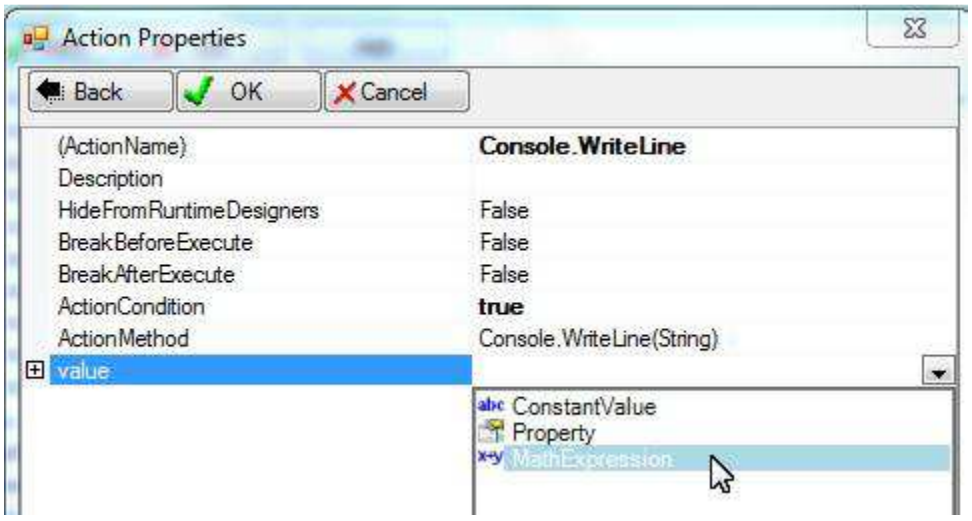
For this sample, we handle the events by simply writing information on the console.



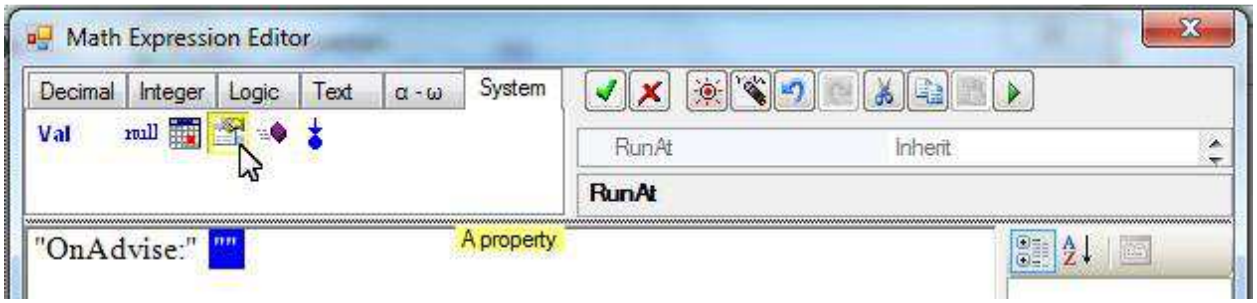

Method Designer - Handle event Advise by h1
Parameters
(Name)
h1
Description
DoubleBuffered
False
XClass<DdeClient> sender
DdeAdviseEventArgs e
Main
Add local variable
Add an action
Create condition

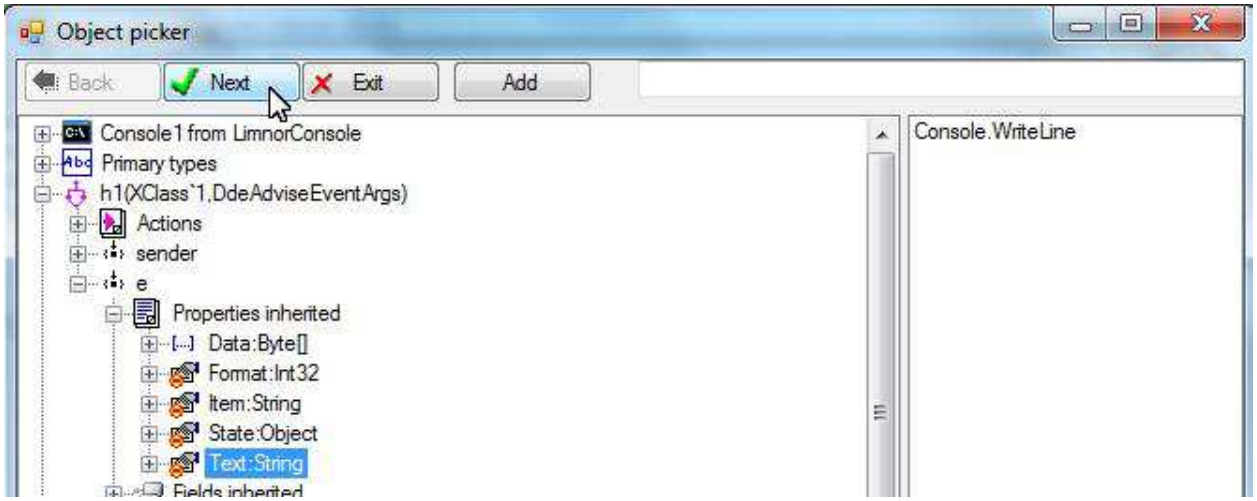

Select an action
Back
Next
Exit
Add
ProjectResources
Console
Static properties
Static fields
Static methods
Beep
Beep(Int32,Int32)


Select an action
Back
Next
Exit
Add
WriteLine(Object)
WriteLine(Single)
WriteLine(String)
WriteLine(String,Object)
Console.WriteLine

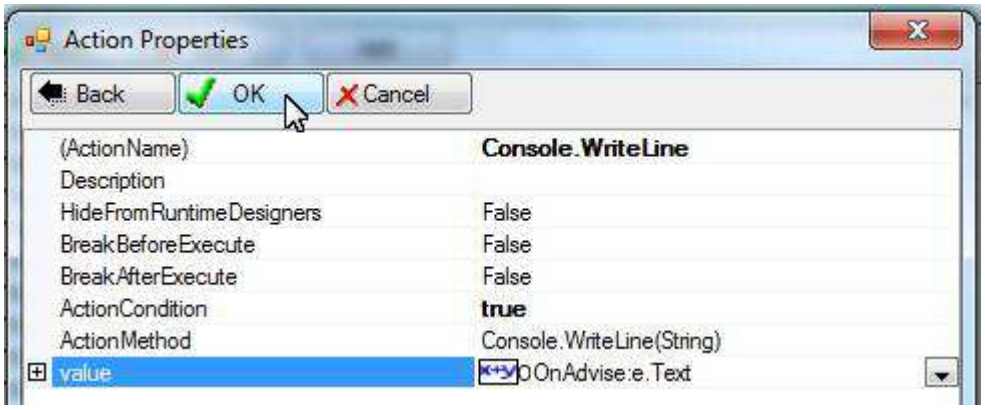

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(String)
value
ConstantValue
Property
MathExpression

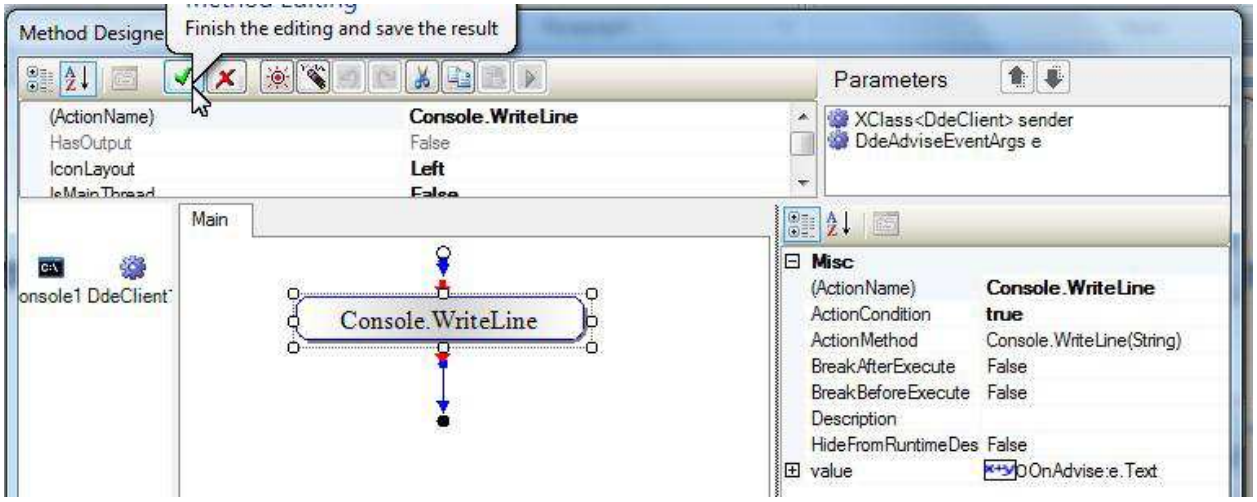

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
0x LF CR CL
A+
Concatenate two strings together to form a new string
0


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
RunAt
Inherit
RunAt
"OnAdvise:"
A property

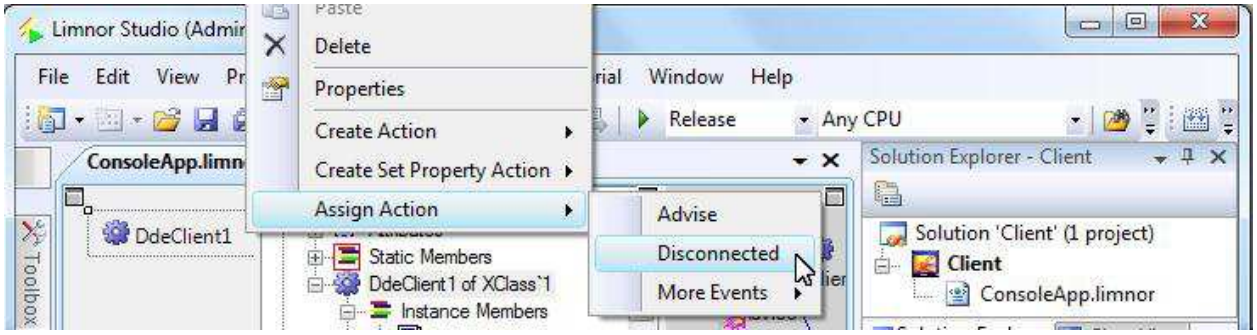

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
h1(XClass`1,DdeAdviseEventArgs)
Actions
sender
e
Properties inherited
Data:Byte[]
Format:Int32
Item:String
State:Object
Text:String
Console.WriteLine

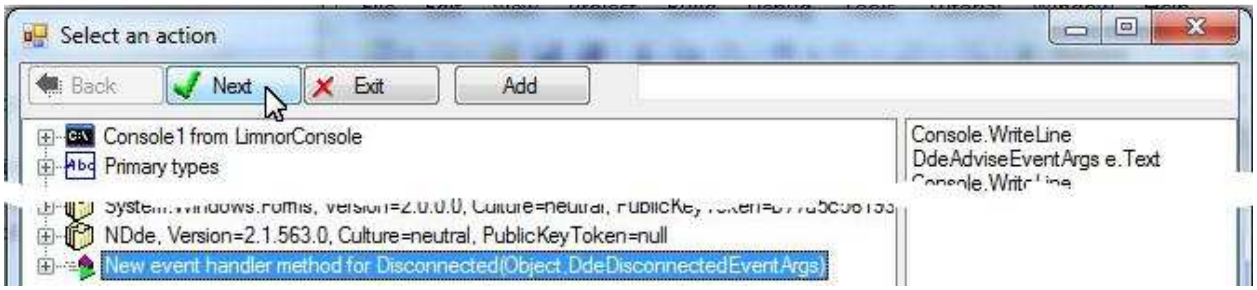

Finish the editing.
Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
RunAt
Inherit
RunAt
"OnAdvise:"
e.Text

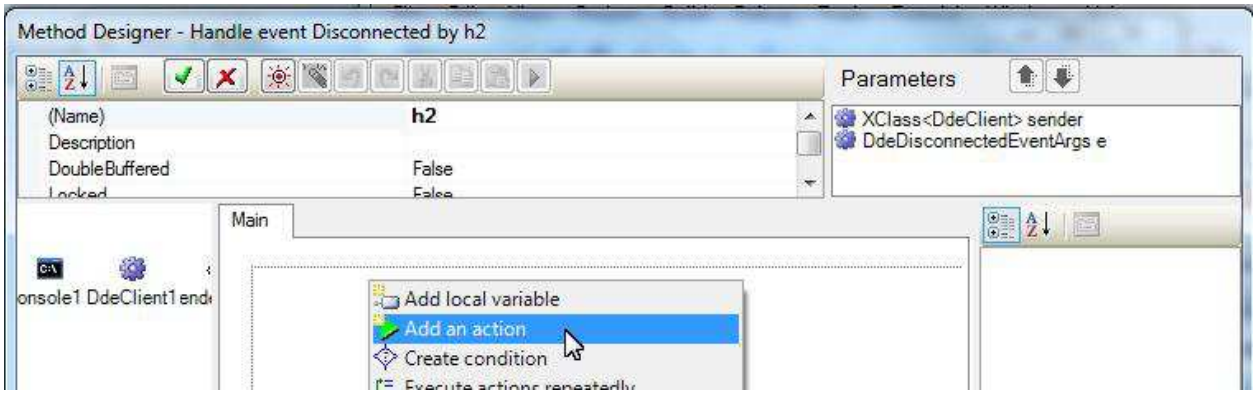


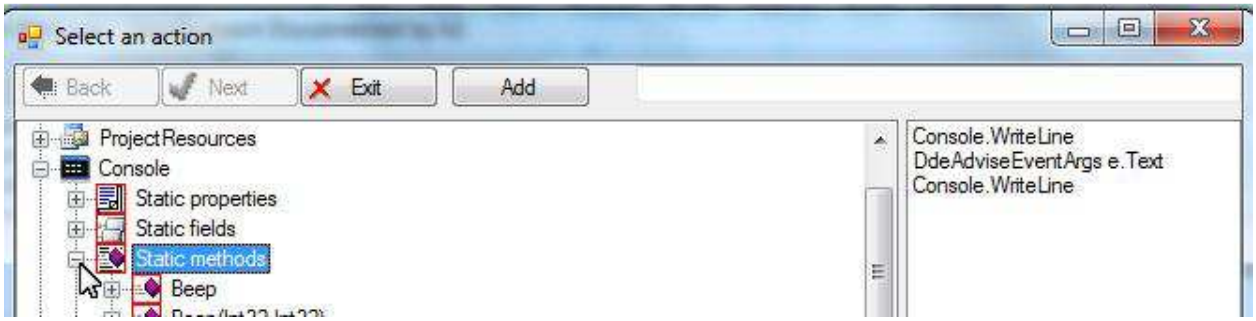


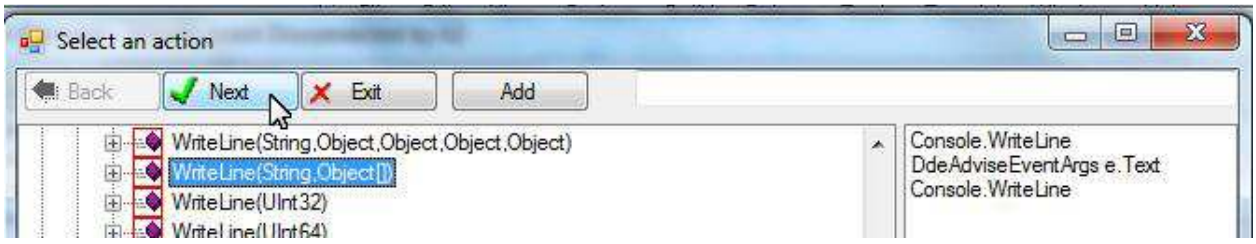
Handle Disconnected event

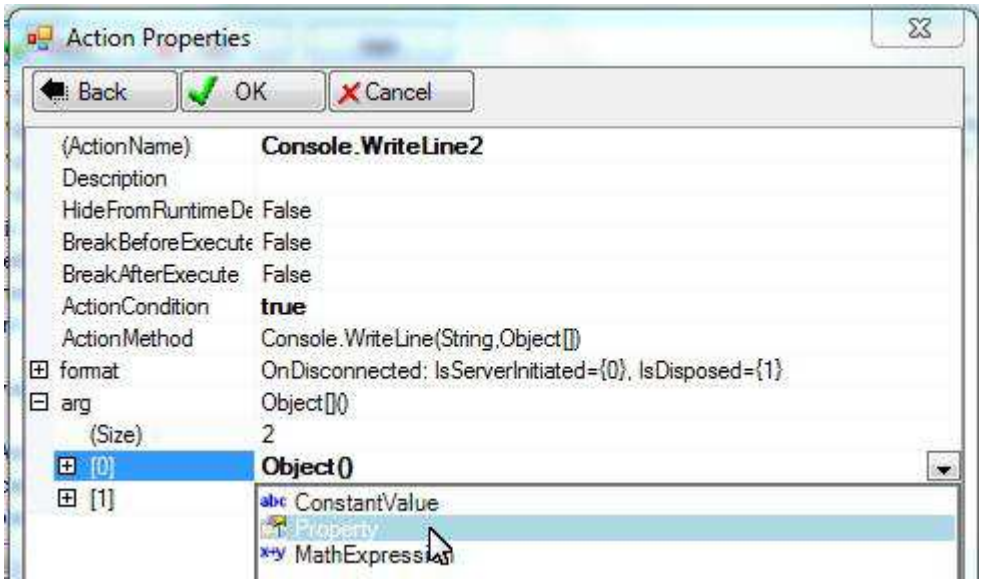


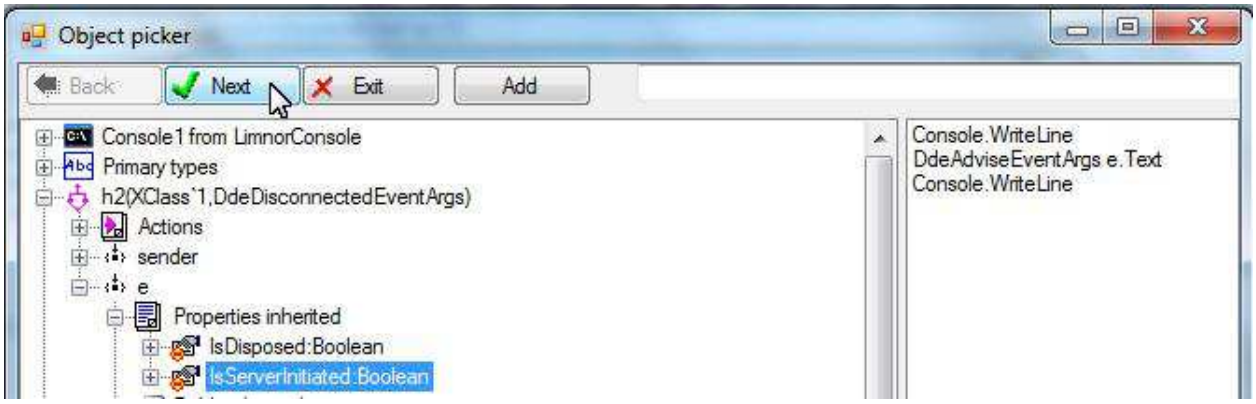


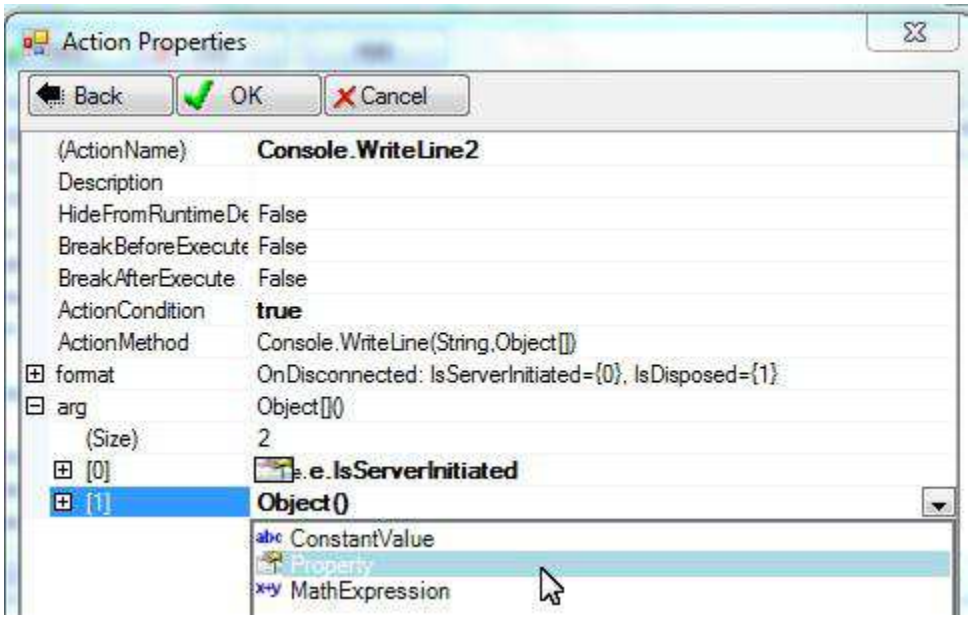

Method Designer - Handle event Disconnected by h2
Parameters
(Name)
h2
Description
DoubleBuffered
False
XClass<DdeClient> sender
DdeDisconnectedEventArgs e
Main
Add local variable
Add an action
Create condition

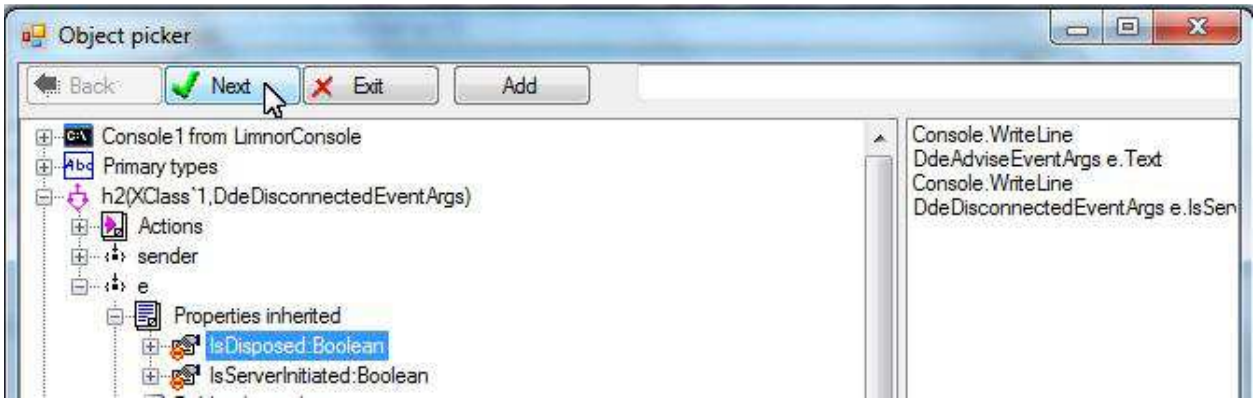

Select an action
Back
Next
Exit
Add
ProjectResources
Console
Static properties
Static fields
Static methods
Beep
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine

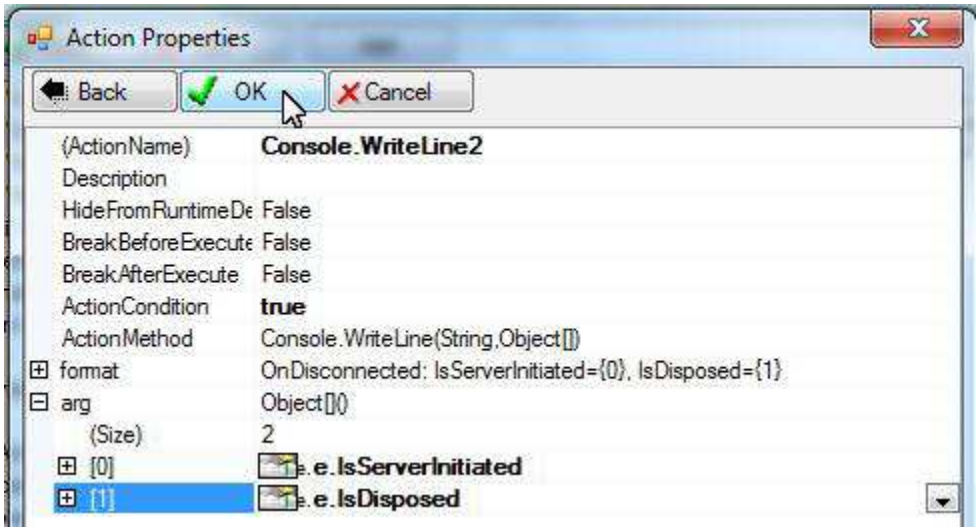

Select an action
Back
Next
Exit
Add
WriteLine(String,Object,Object,Object,Object)
WriteLine(String,Object[])
WriteLine(UInt32)
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine

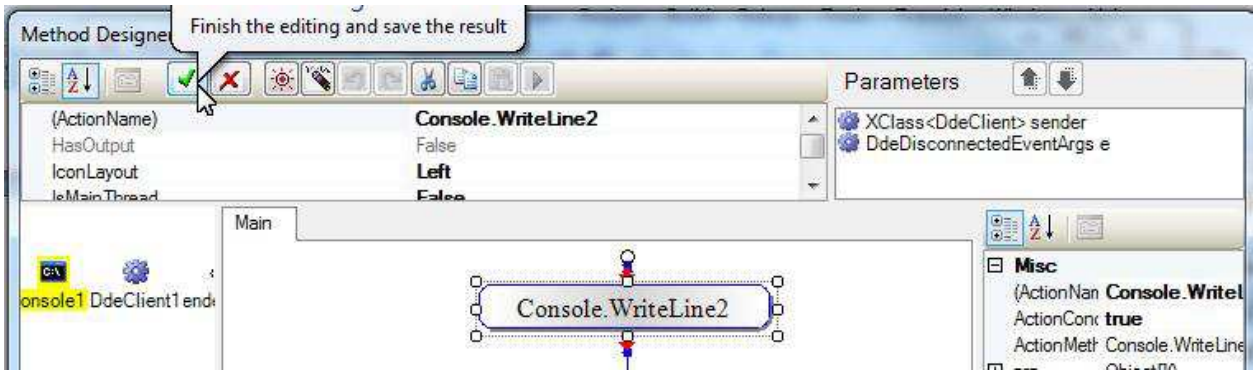

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine2
Description
HideFromRuntimeDe
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(String,Object[])
format
OnDisconnected: IsServerInitiated={0}, IsDisposed={1}
arg
Object[]()
(Size)
2
[0]
Object()
[1]
ConstantValue
Property
MathExpressi


Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
h2(XClass`1,DdeDisconnectedEventArgs)
Actions
sender
e
Properties inherited
IsDisposed:Boolean
IsServerInitiated:Boolean
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine


Action Properties
Back
OK
Cancel
(ActionName) Console.WriteLine2
Description
HideFromRuntimeDe False
BreakBeforeExecute False
BreakAfterExecute False
ActionCondition true
ActionMethod Console.WriteLine(String,Object[])
format OnDisconnected: IsServerInitiated={0}, IsDisposed={1}
arg Object[]()
(Size) 2
[0] e.IsServerInitiated
[1] Object()
ConstantValue
Property
MathExpression

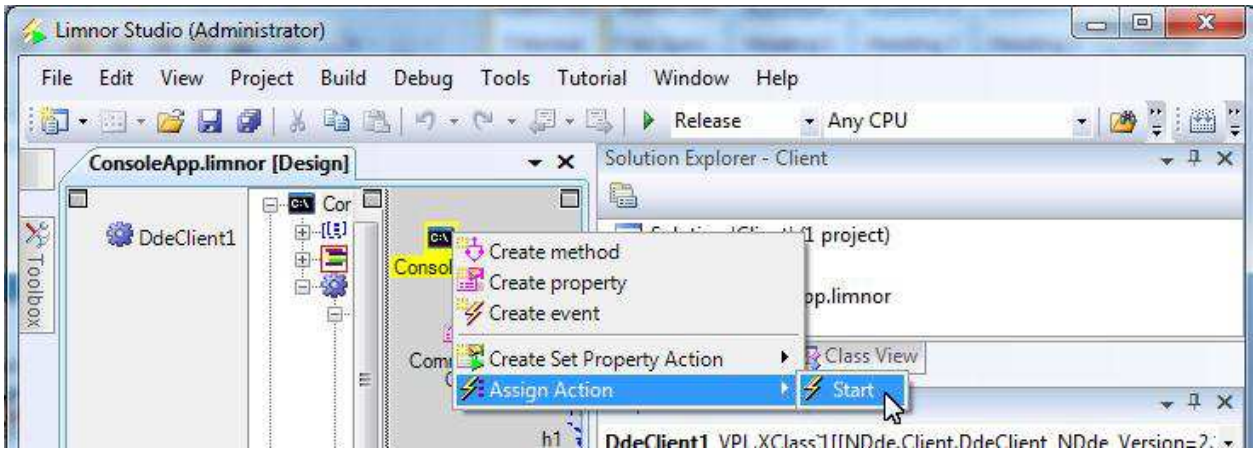

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
h2(XClass`1,DdeDisconnectedEventArgs)
Actions
sender
e
Properties inherited
IsDisposed:Boolean
IsServerInitiated:Boolean
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSer





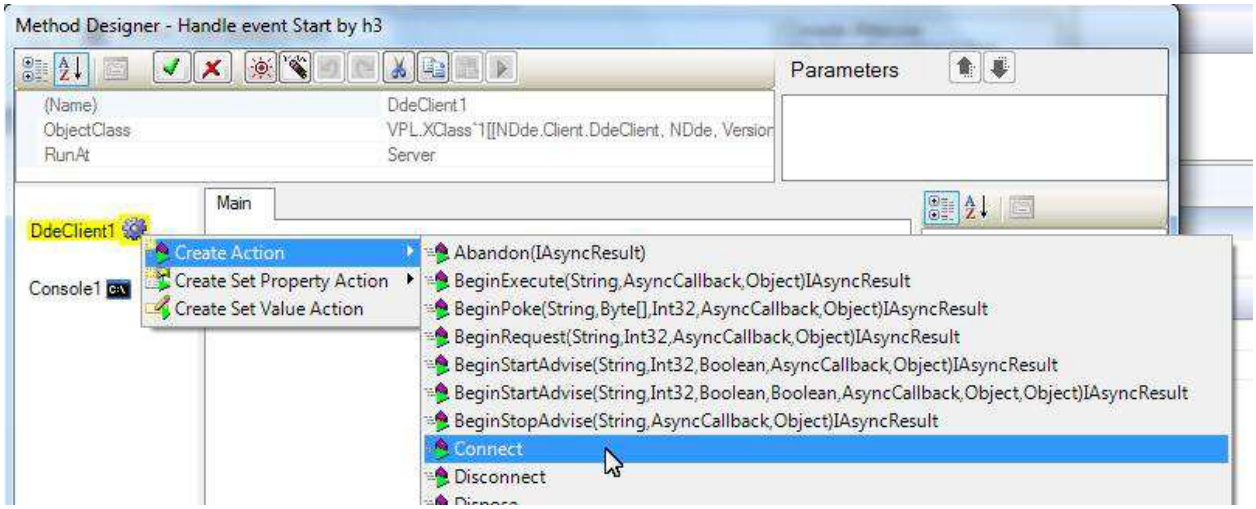
Execute DDE client actions

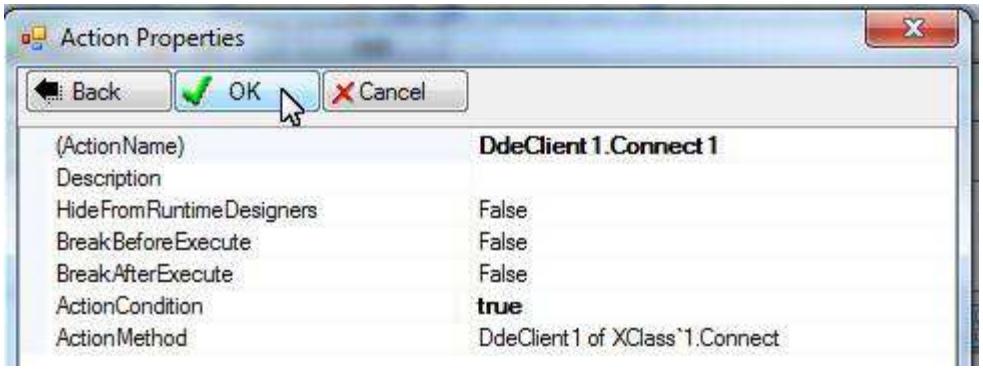
We handle the Start event of the console application to execute the DDE client commands.



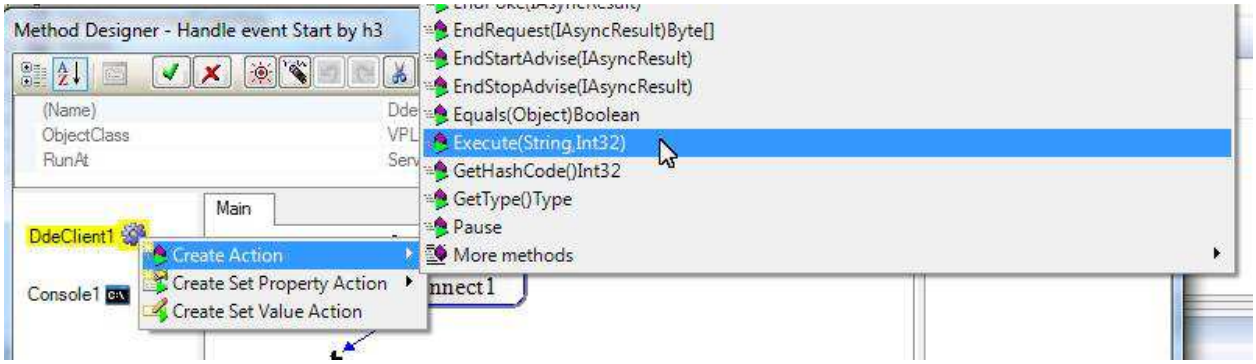
Connect to the DDE server

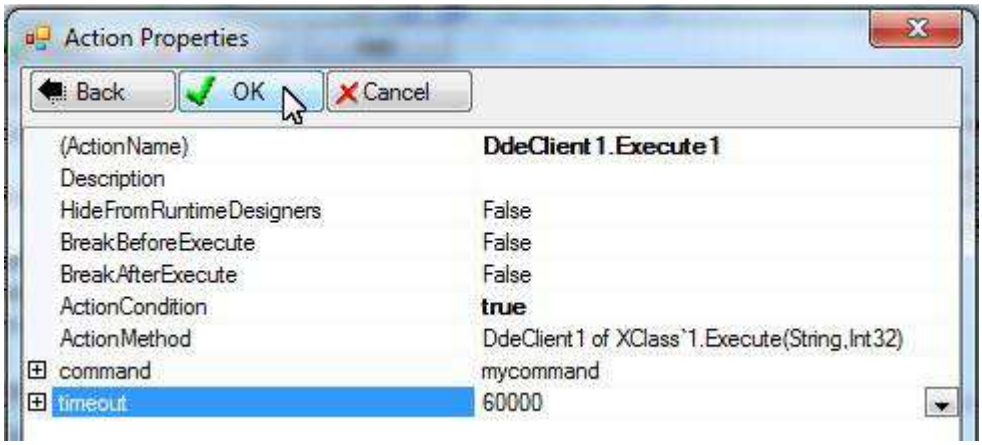
First we need to connect the DDE client.



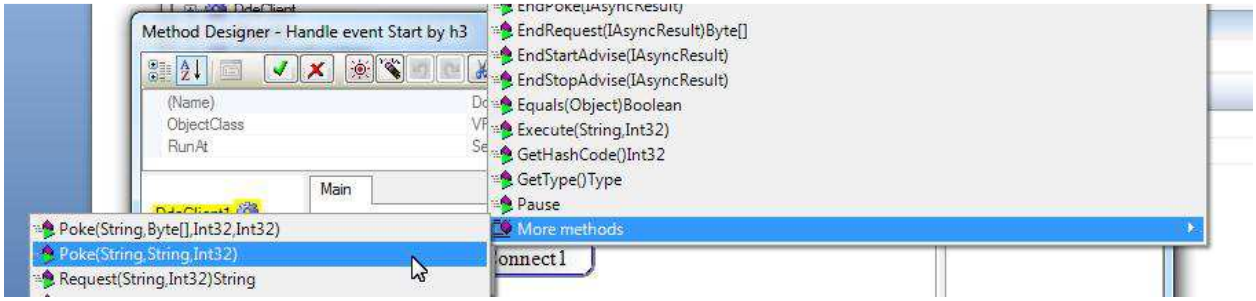


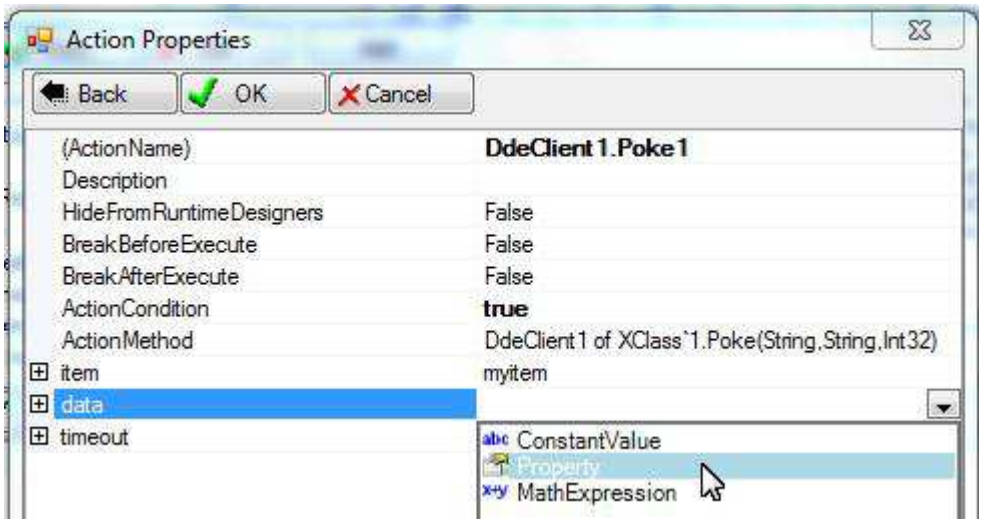
Synchronous Execute Operation

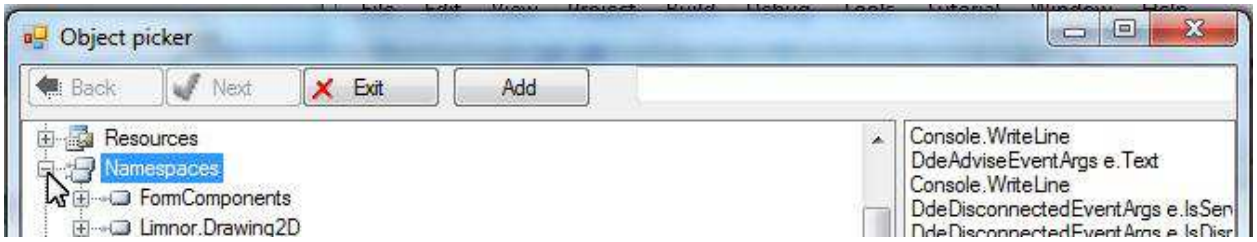


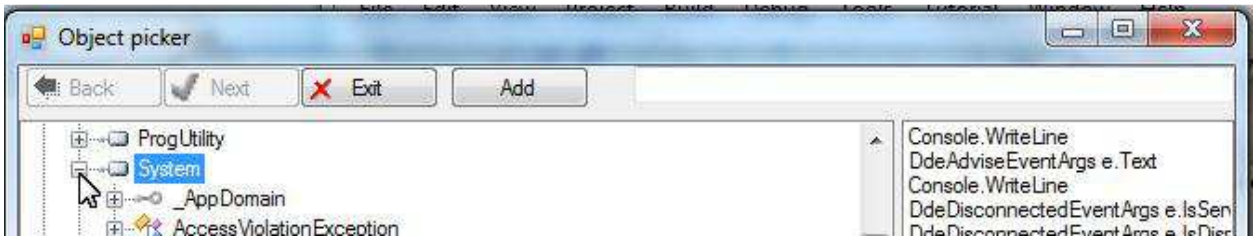


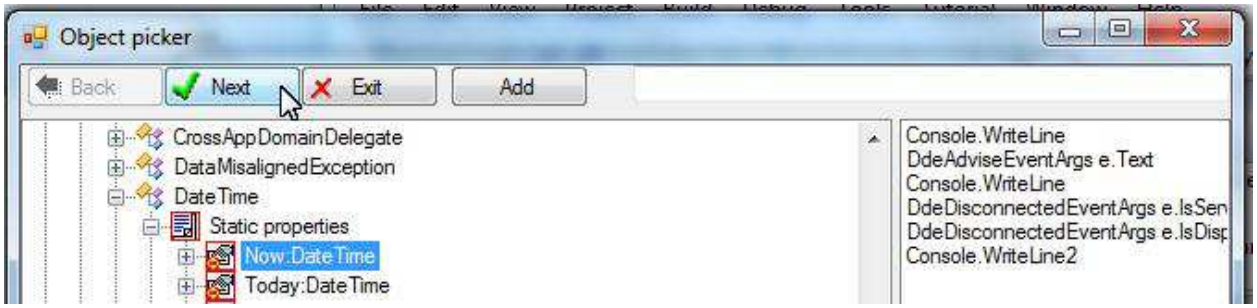
Synchronous Poke Operation

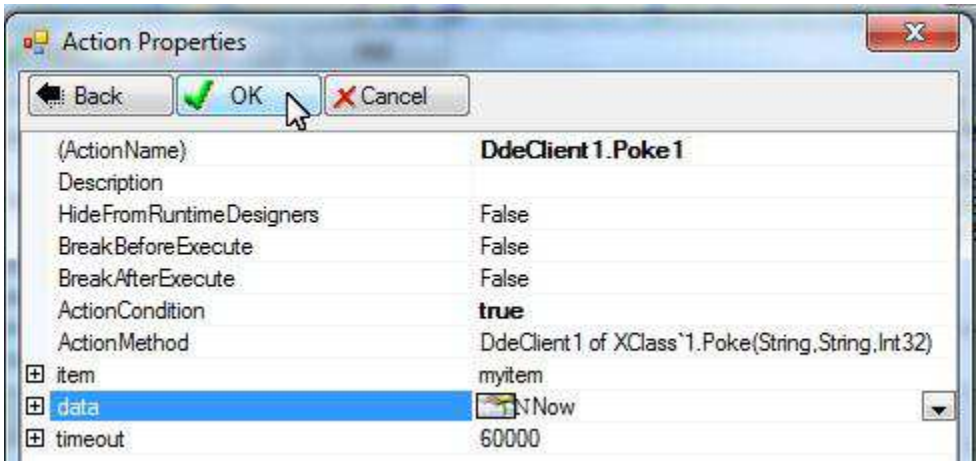



Action Properties
Back
OK
Cancel
(ActionName)
DdeClient1.Poke1
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
DdeClient1 of XClass`1.Poke(String,String,Int32)
item
myitem
data
timeout
ConstantValue
Property
MathExpression

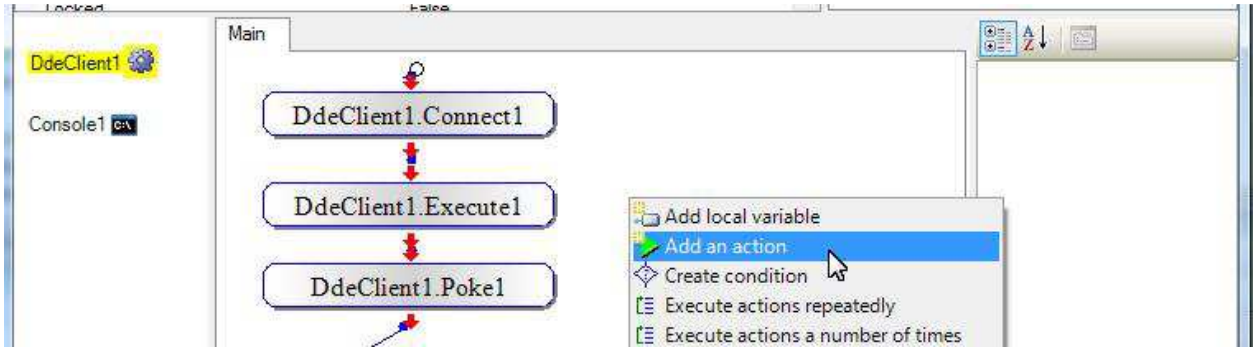

Object picker
Back
Next
Exit
Add
Resources
Namespaces
FormComponents
Limnor.Drawing2D
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSen

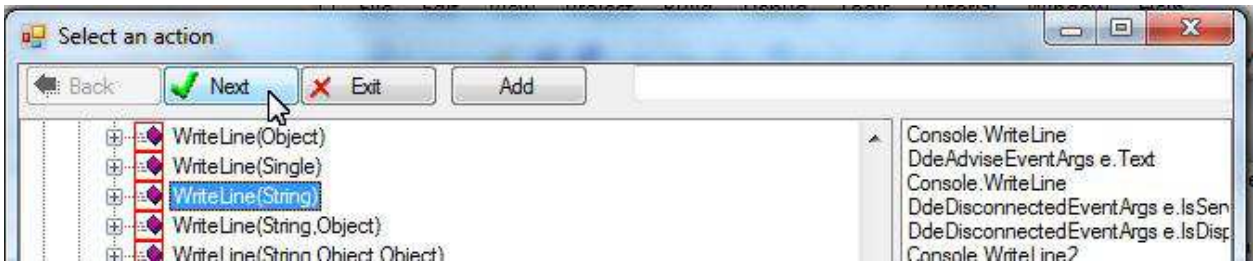

Object picker
Back
Next
Exit
Add
ProgUtility
System
_AppDomain
AccessViolationException
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSen

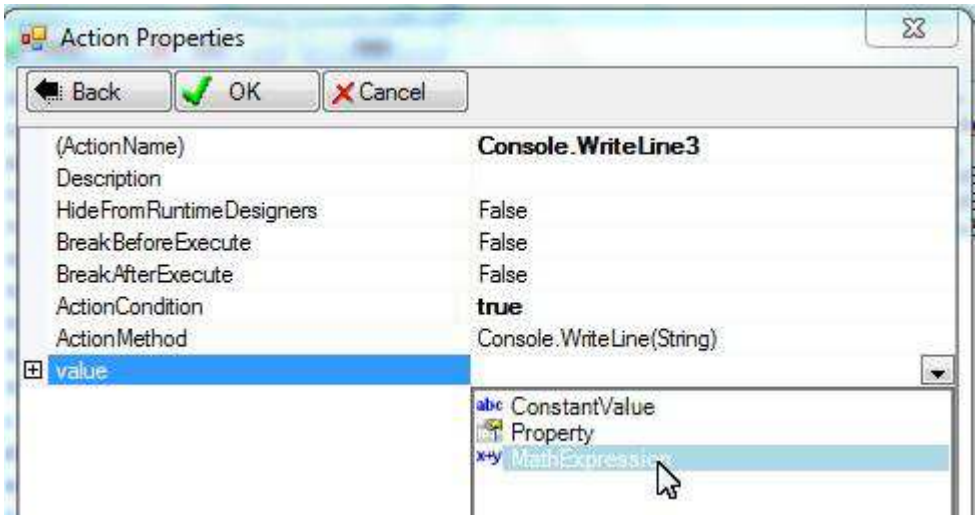

Object picker
Back
Next
Exit
Add
CrossAppDomainDelegate
DataMisalignedException
DateTime
Static properties
Now:DateTime
Today:DateTime
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSen
DdeDisconnectedEventArgs e.IsDisp
Console.WriteLine2



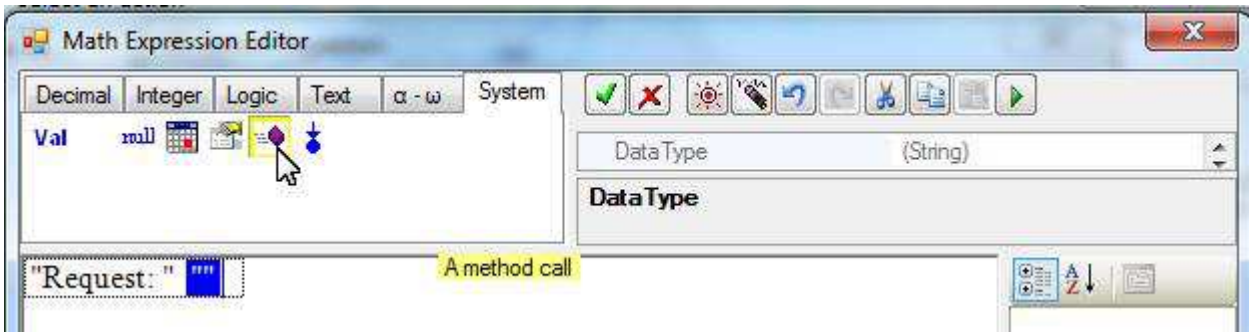
Synchronous Request Operation

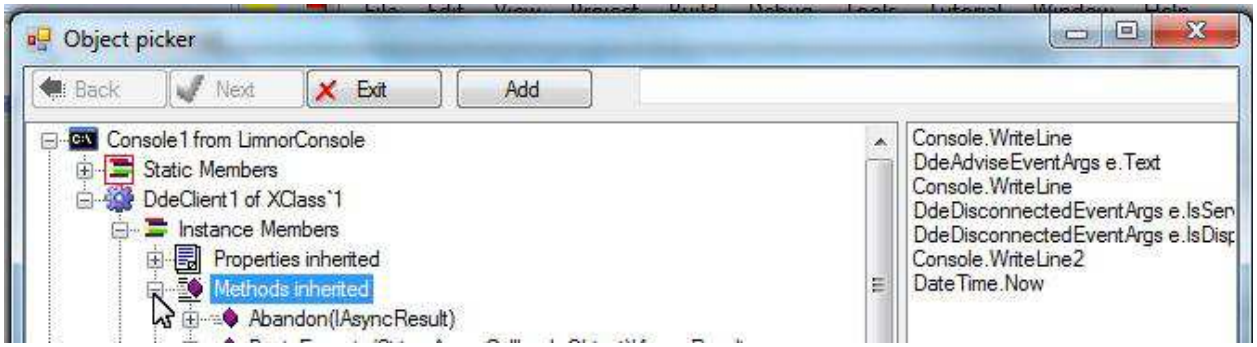


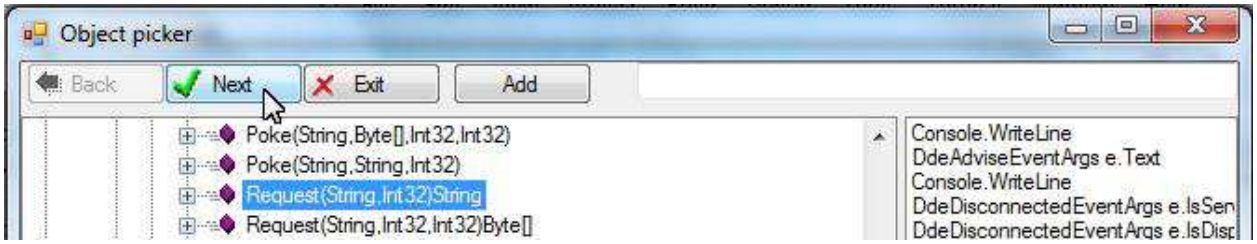


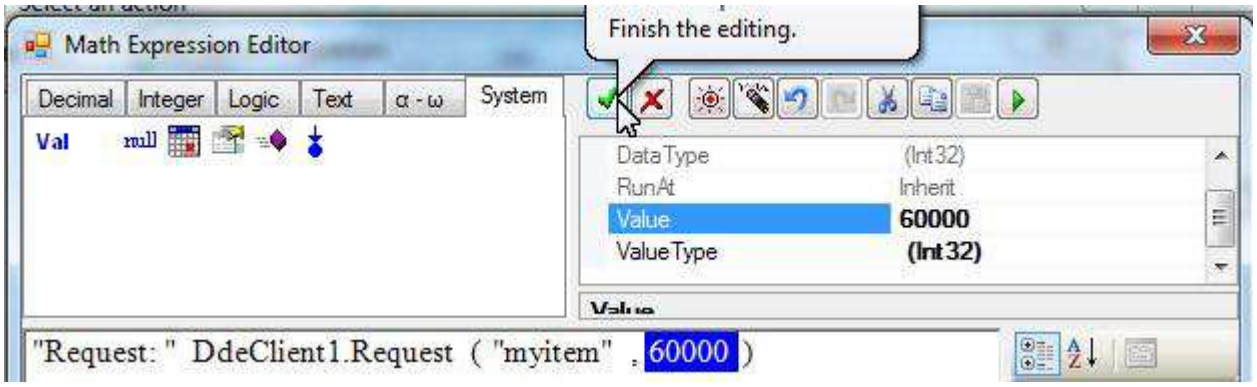

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine3
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(String)
value
ConstantValue
Property

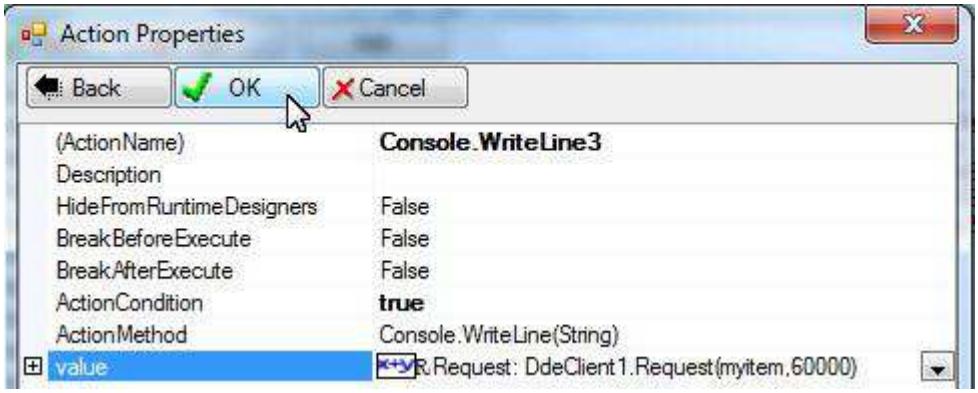

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Concatenate two strings together to form a new string


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Data Type
(String)
DataType
"Request: "
A method call


Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Static Members
DdeClient1 of XClass`1
Instance Members
Properties inherited
Methods inherited
Abandon(IAsyncResult)
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DateTime.Now

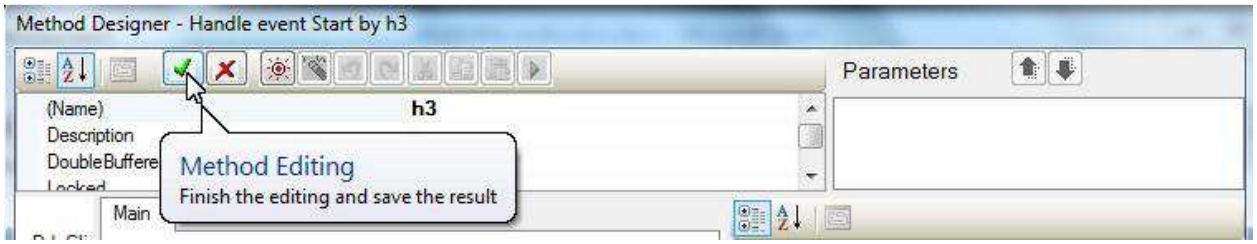




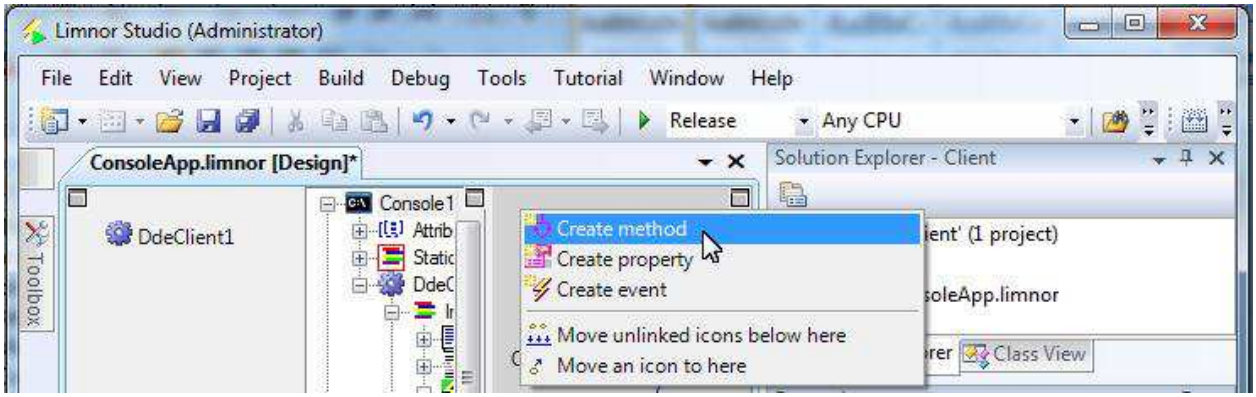


Asynchronous Execute Operation – Create callback function

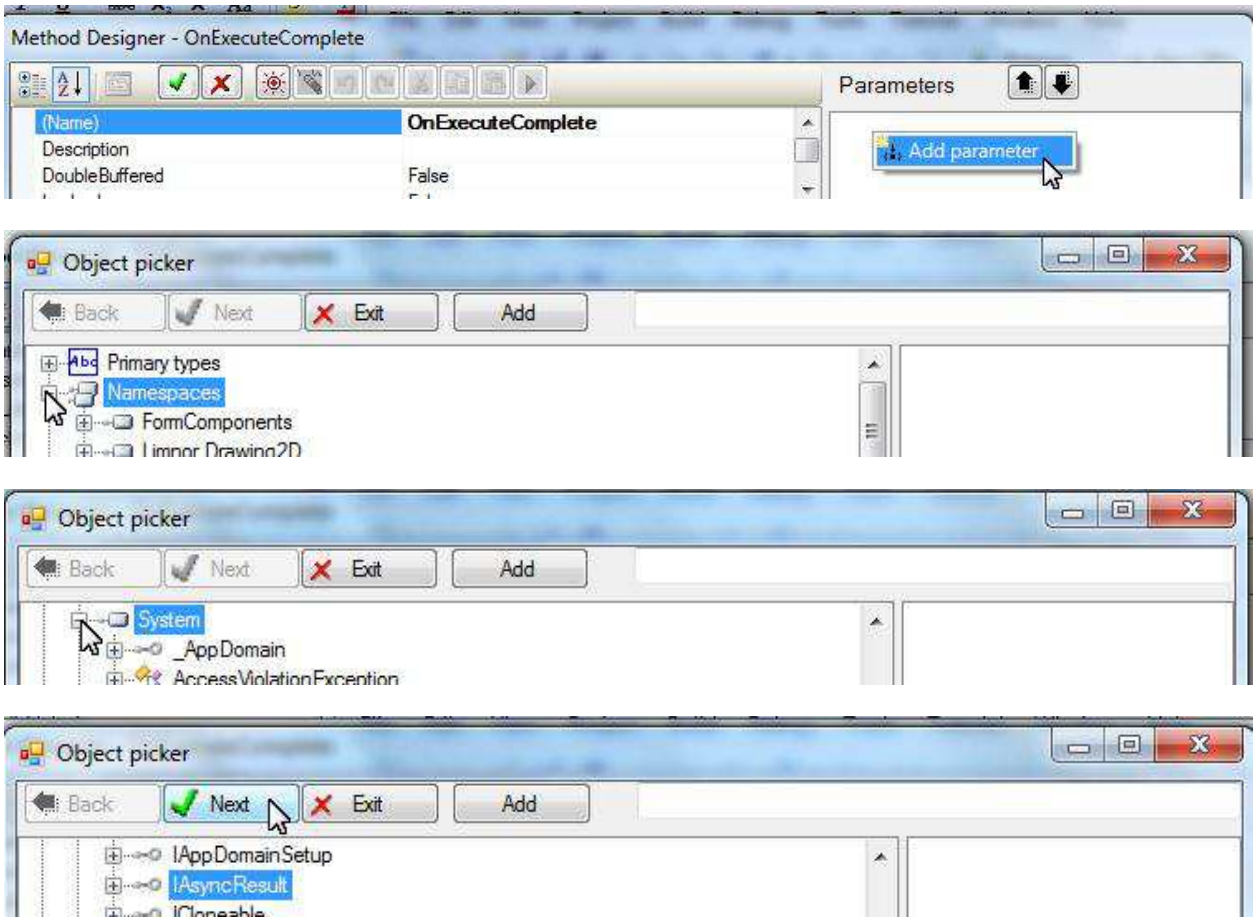
An asynchronous operation requires a callback function to be executed when the operation finishes. So, we create a method as the callback function. Close the current Method Editor so that we may create a new method:



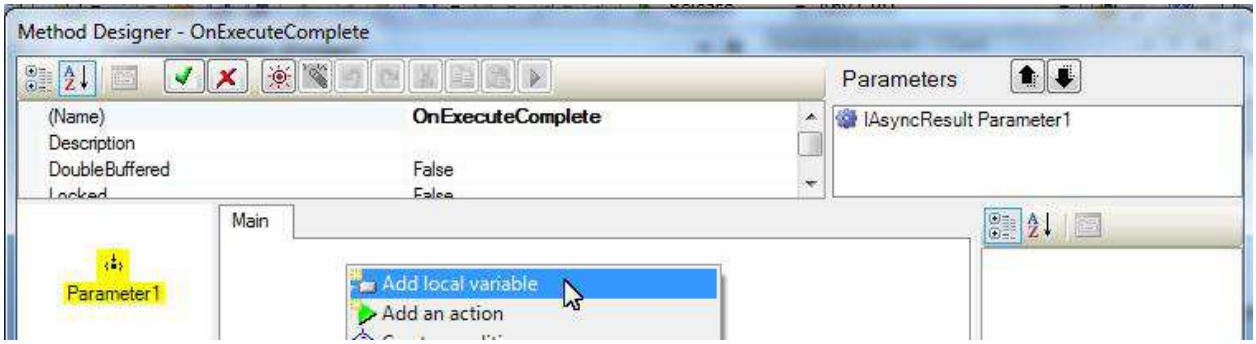
Create a new method:

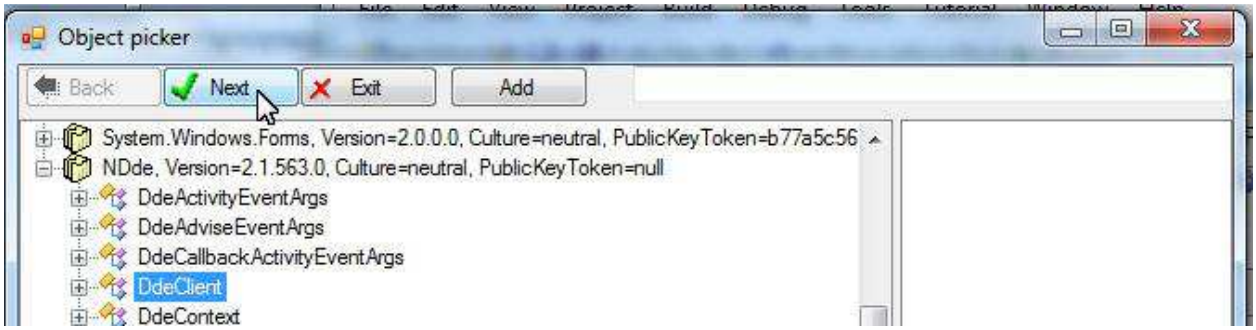


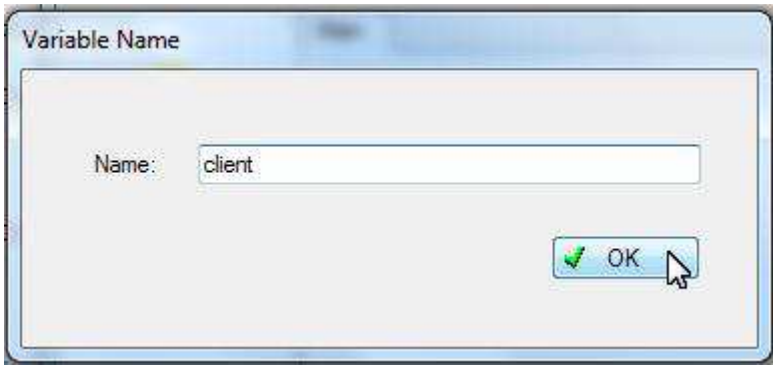
Rename the method to OnExecuteComplete. Add a parameter of type `IAsyncResult`.

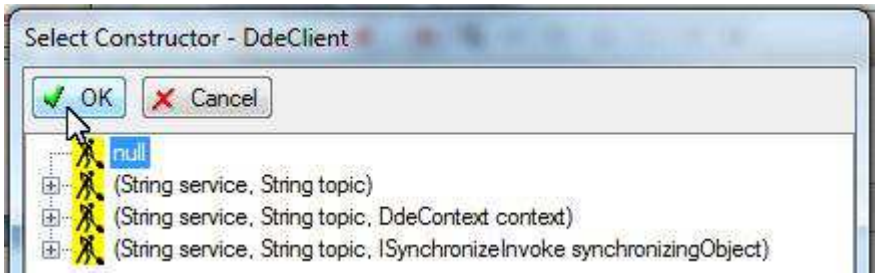


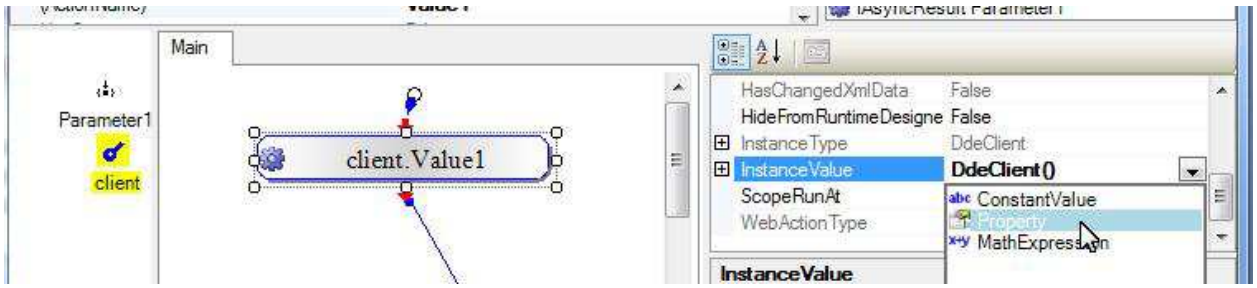
Get DdeClient from the method parameter:

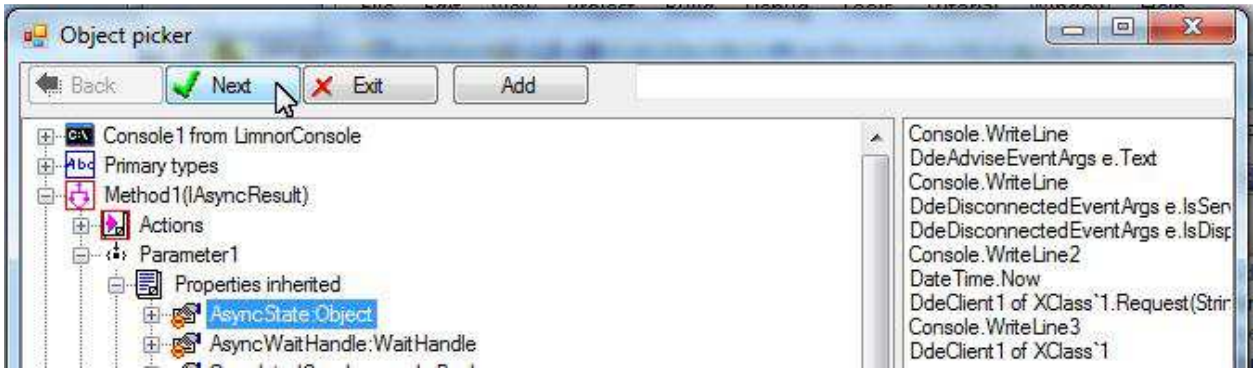

Method Designer - OnExecuteComplete
Parameters
(Name)
OnExecuteComplete
Description
DoubleBuffered
False
IAsyncResult Parameter1
Main
Parameter1
Add local variable
Add an action


Object picker
Back
Next
Exit
Add
System.Windows.Forms, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c56
NDde, Version=2.1.563.0, Culture=neutral, PublicKeyToken=null
DdeActivityEventArgs
DdeAdviseEventArgs
DdeCallbackActivityEventArgs
DdeClient
DdeContext

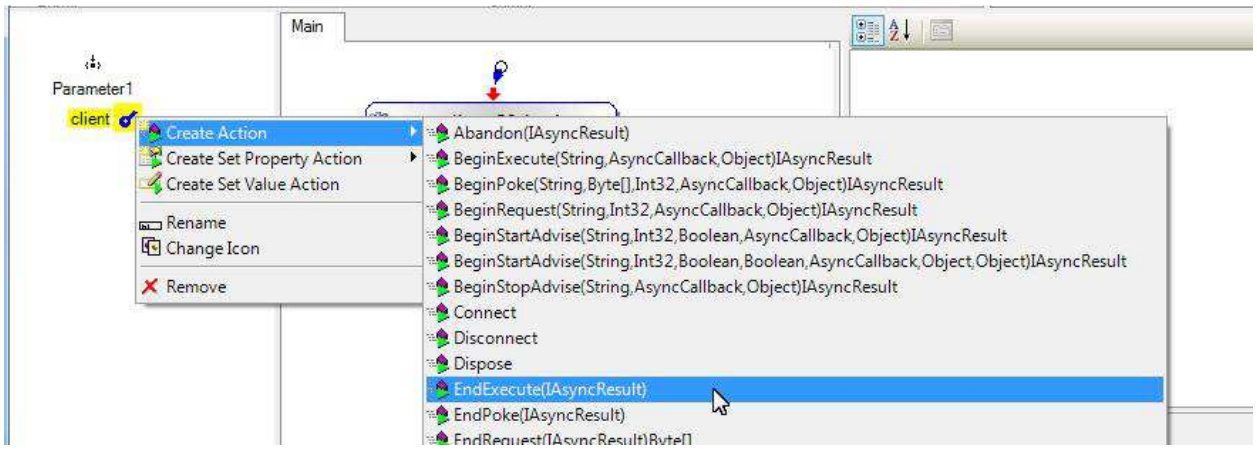

Variable Name
Name:
client
OK

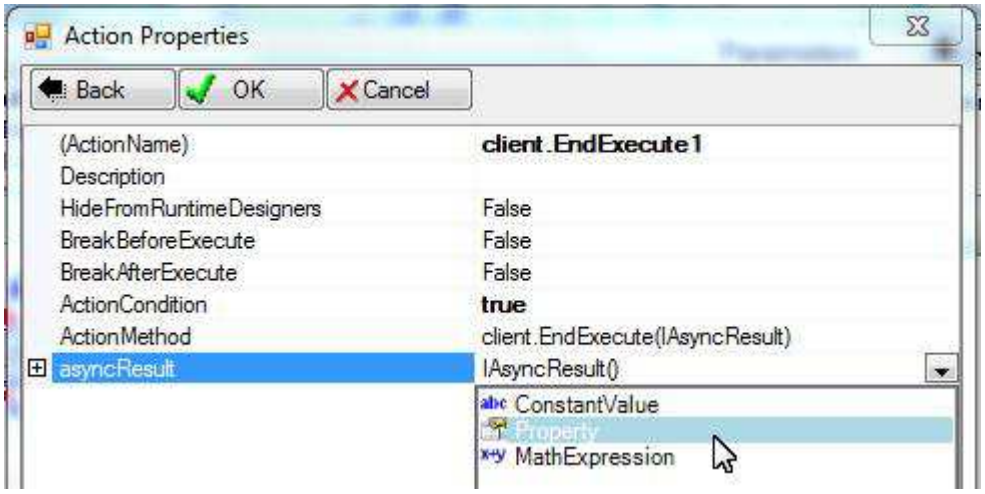

Select Constructor - DdeClient
OK
Cancel
null
(String service, String topic)
(String service, String topic, DdeContext context)
(String service, String topic, ISynchronizeInvoke synchronizingObject)

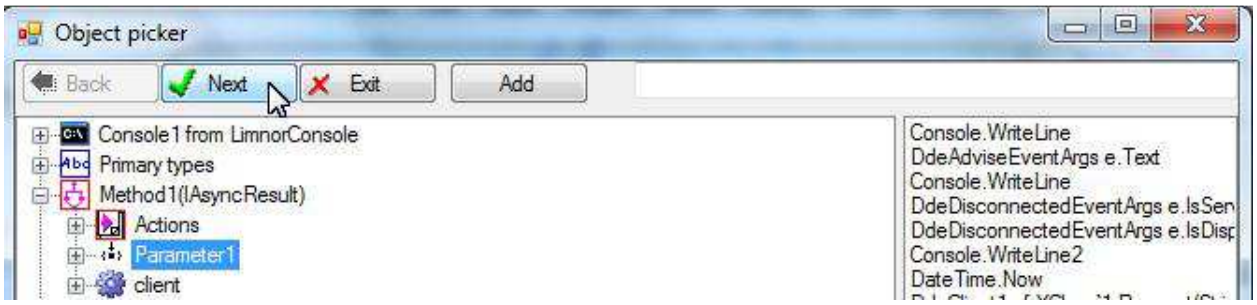


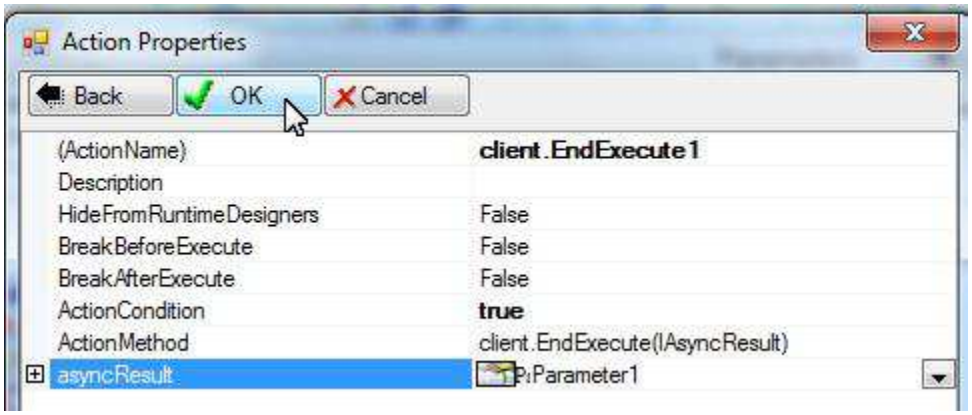


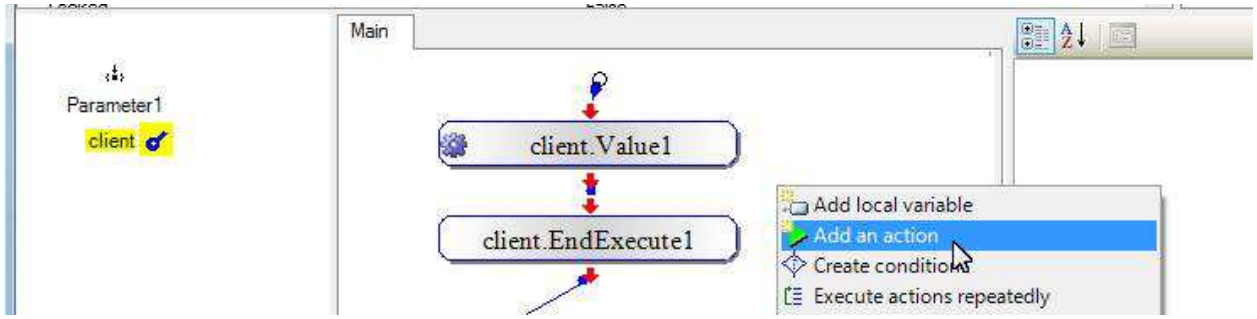
Execute EndExecute action:

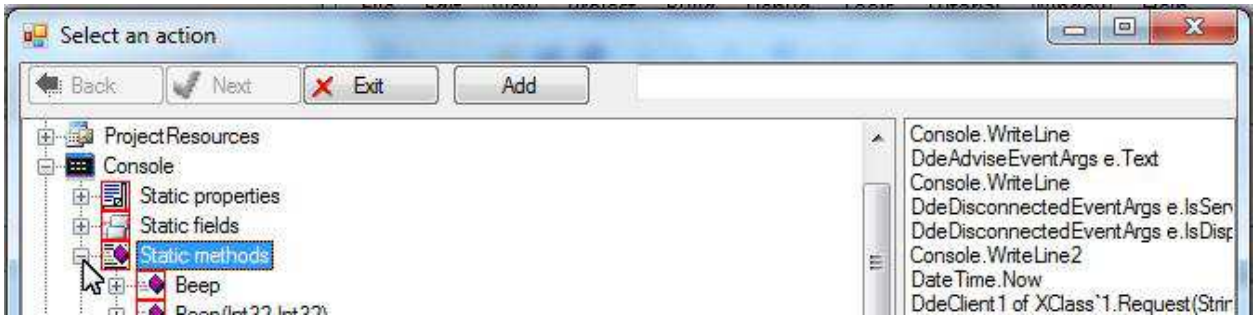


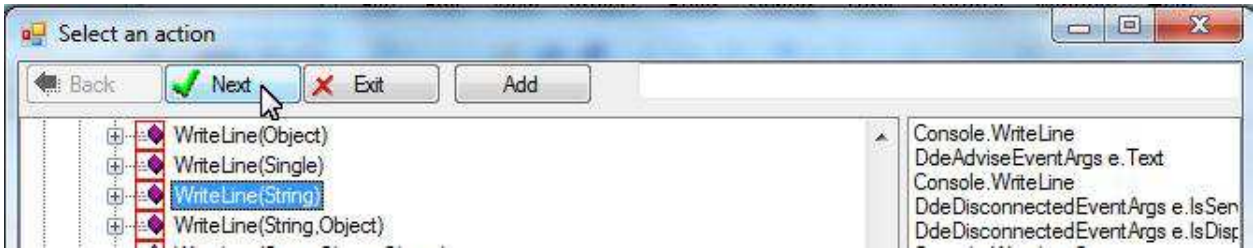

Action Properties
Back
OK
Cancel
(ActionName)
client.EndExecute1
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
client.EndExecute(IAsyncResult)
asyncResult
IAsyncResult()
ConstantValue
Property
MathExpression

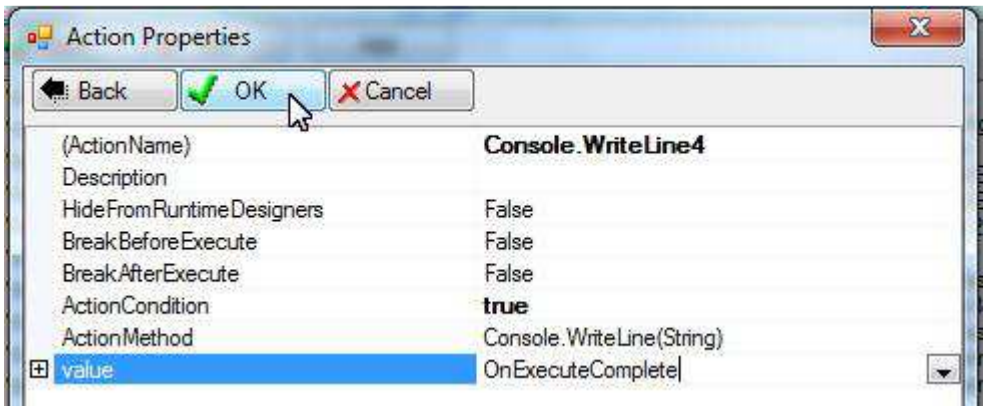

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
Method1(IAsyncResult)
Actions
Parameter1
client
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSen
DdeDisconnectedEventArgs e.IsDisp
Console.WriteLine2
DateTime.Now

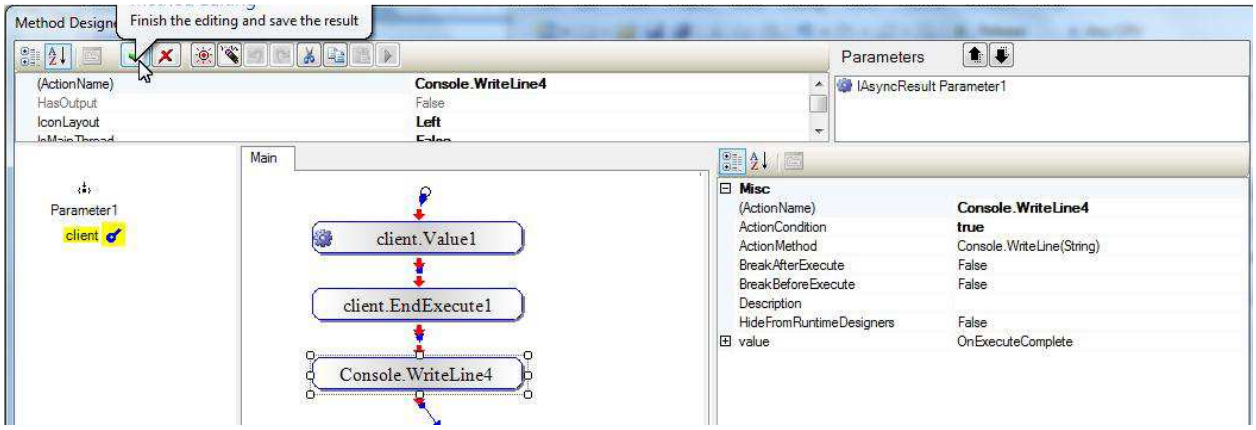

Action Properties
Back
OK
Cancel
(ActionName)
client.EndExecute1
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
client.EndExecute(IAsyncResult)
asyncResult
Parameter1


Parameter1
client
Main
client.Value1
client.EndExecute1
Add local variable
Add an action
Create conditions
Execute actions repeatedly

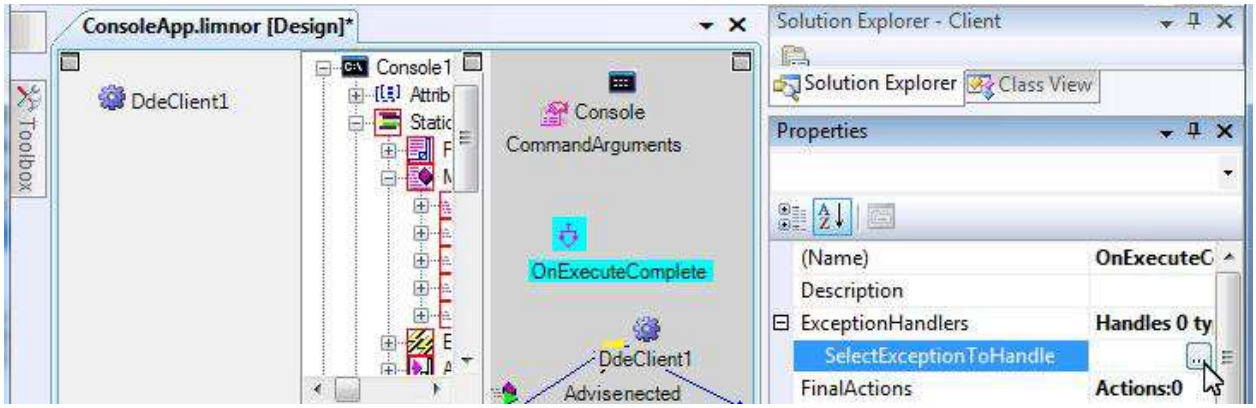

Select an action
Back
Next
Exit
Add
Project Resources
Console
Static properties
Static fields
Static methods
Beep
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSer
DdeDisconnectedEventArgs e.IsDisp
Console.WriteLine2
DateTime.Now
DdeClient1 of XClass`1.Request(Strir

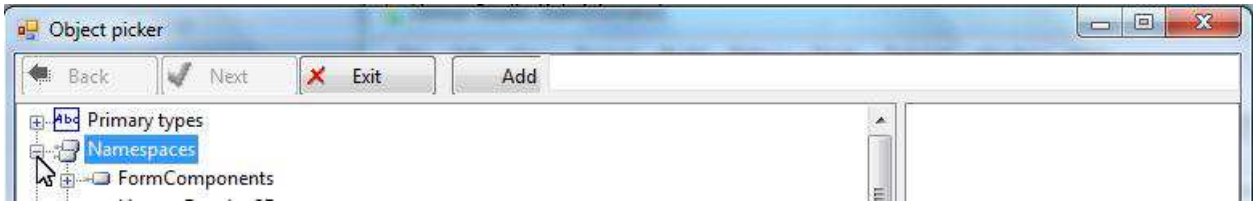

Select an action
Back
Next
Exit
Add
WriteLine(Object)
WriteLine(Single)
WriteLine(String)
WriteLine(String,Object)
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSen
DdeDisconnectedEventArgs e.IsDisp

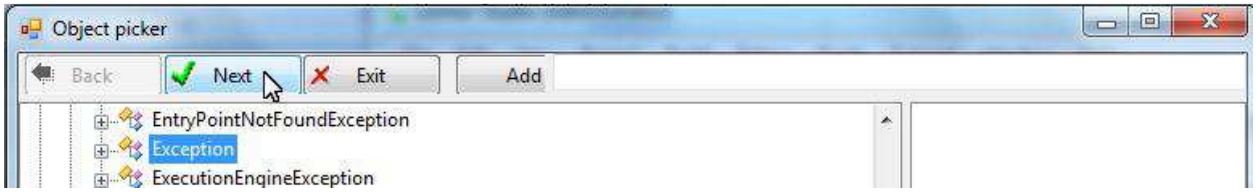

Action Properties
Back
OK
Cancel
(ActionName) Console.WriteLine4
Description
HideFromRuntimeDesigners False
BreakBeforeExecute False
BreakAfterExecute False
ActionCondition true
ActionMethod Console.WriteLine(String)
value OnExecuteComplete

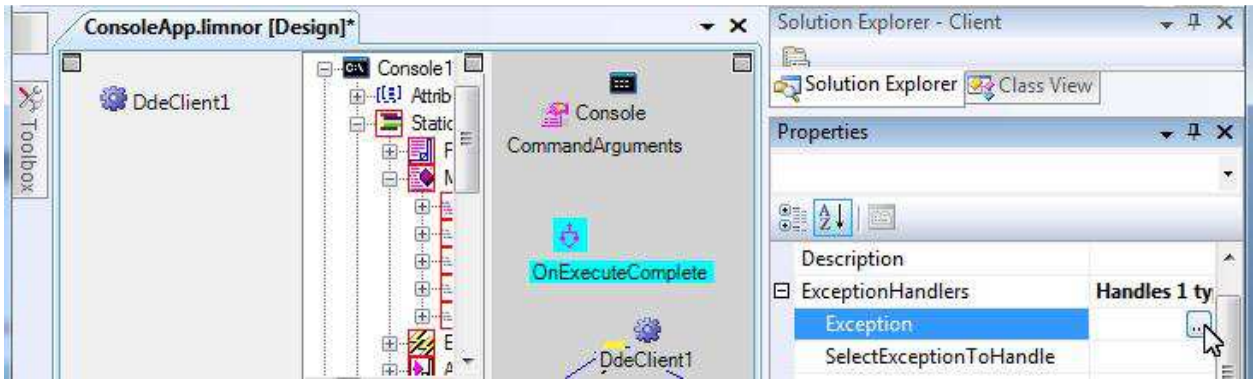

Finish the editing and save the result
Parameters
IAsyncResult Parameter1
(ActionName) Console.WriteLine4
HasOutput False
IconLayout Left
Main
Parameter1
client
client.Value1
client.EndExecute1
Console.WriteLine4
Misc
(ActionName) Console.WriteLine4
ActionCondition true
ActionMethod Console.WriteLine(String)
BreakAfterExecute False
BreakBeforeExecute False
Description
HideFromRuntimeDesigners False
value OnExecuteComplete

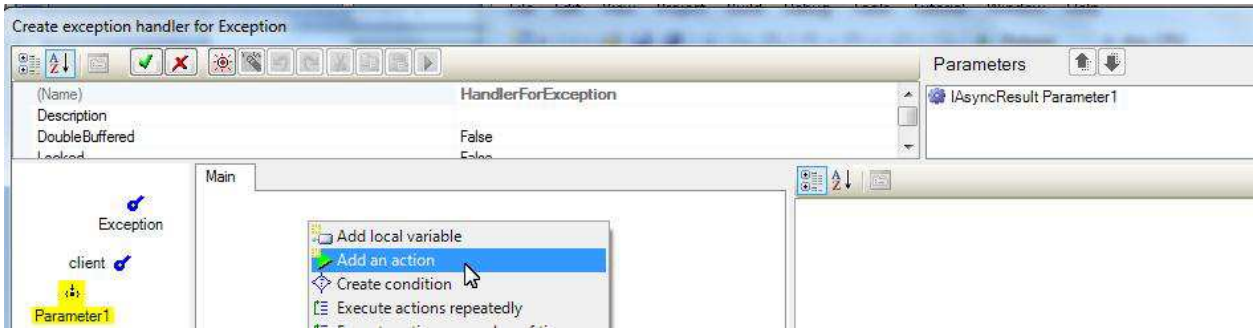
Asynchronous Execute Operation – Handle exception

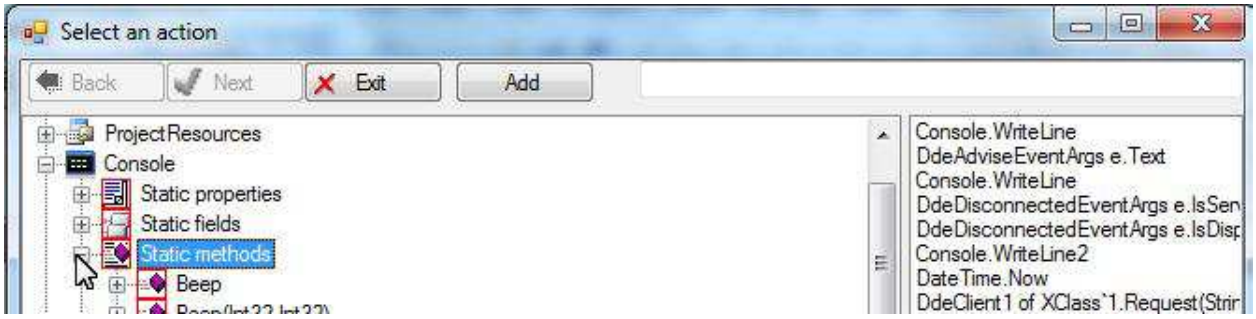


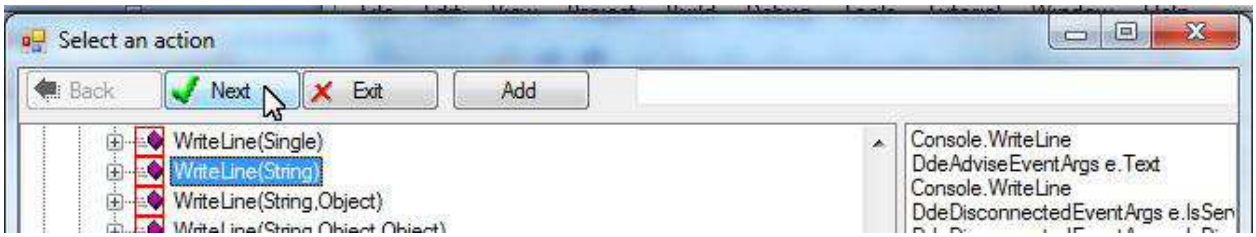


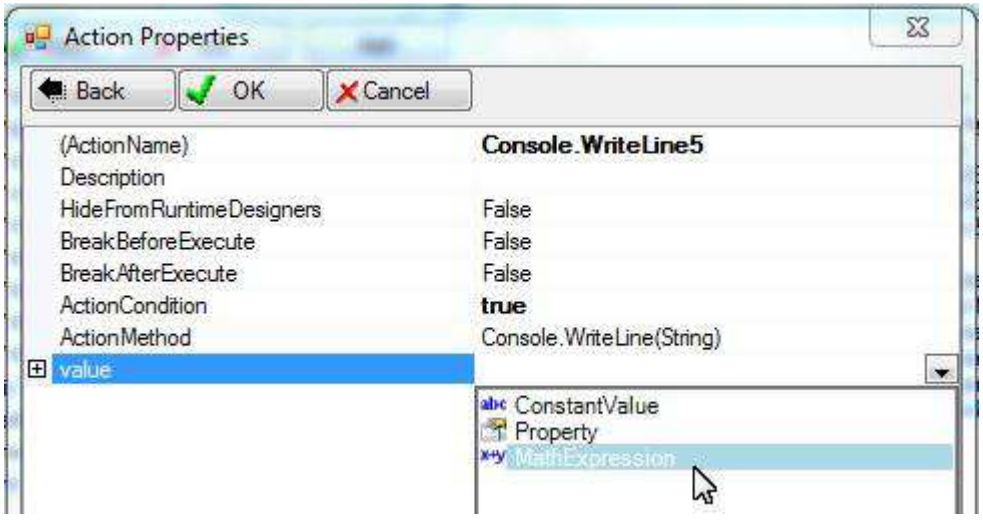





Create exception handler for Exception
Parameters
(Name)
HandlerForException
Description
DoubleBuffered
False
IAsyncResult Parameter1
Main
Exception
client
Parameter1
Add local variable
Add an action
Create condition
Execute actions repeatedly

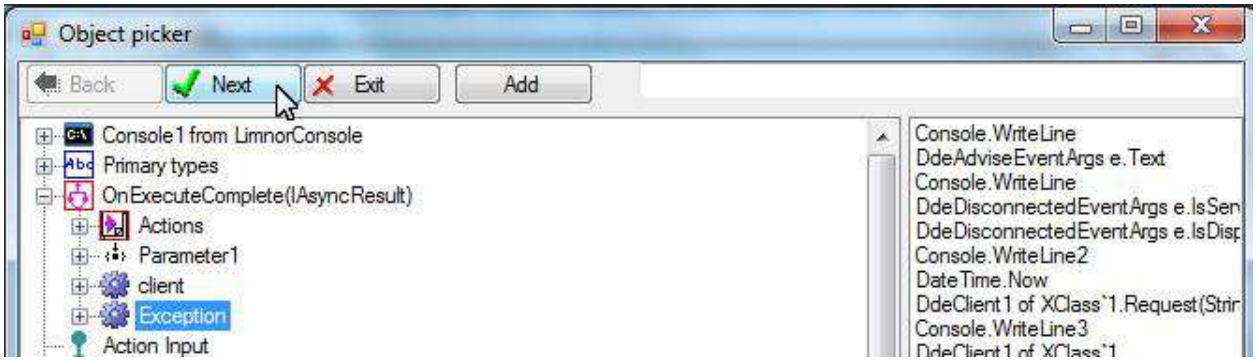

Select an action
Back
Next
Exit
Add
ProjectResources
Console
Static properties
Static fields
Static methods
Beep
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSen
DdeDisconnectedEventArgs e.IsDisp
Console.WriteLine2
DateTime.Now
DdeClient1 of XClass`1.Request(Strir

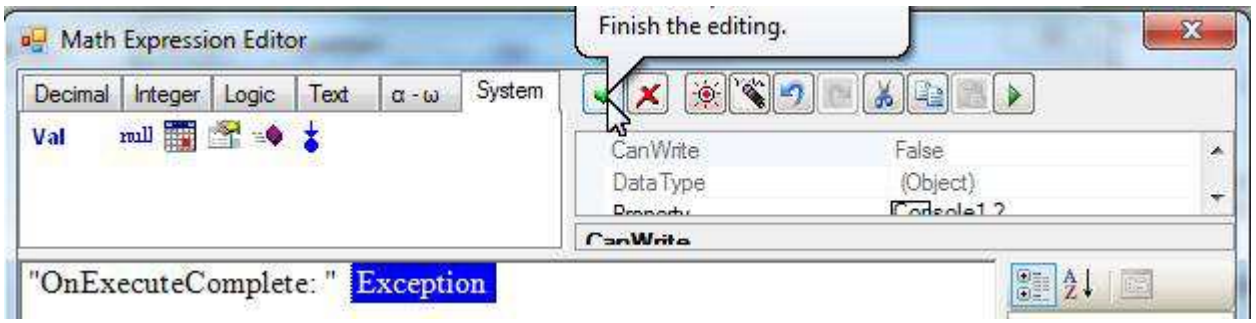

Select an action
Back
Next
Exit
Add
WriteLine(Single)
WriteLine(String)
WriteLine(String,Object)
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DdeDisconnectedEventArgs e.IsSen

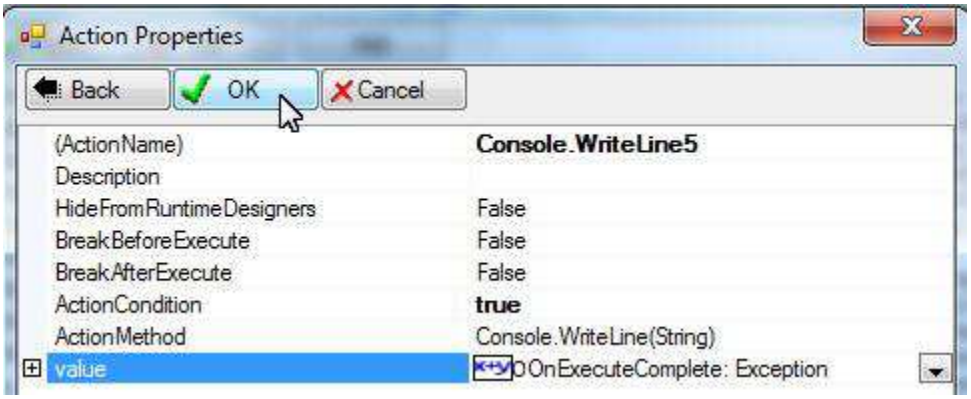

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine5
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(String)
value
ConstantValue
Property
MathExpression

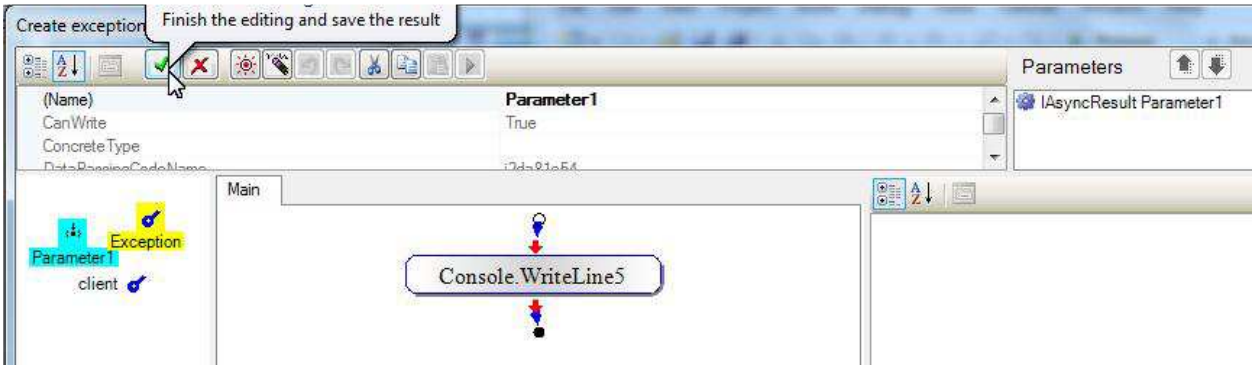

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
0x LF CR CL
A+
0
Concatenate two strings together to form a new string


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
RunAt
Inherit
Value
"OnExecuteComplete: "
A property


Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
OnExecuteComplete(IAsyncResult)
Actions
Parameter1
client
Exception
Action Input
Console.WriteLine
DdeAdviseEventArgs e.Text
Console.WriteLine
DateTime.Now
Console.WriteLine3

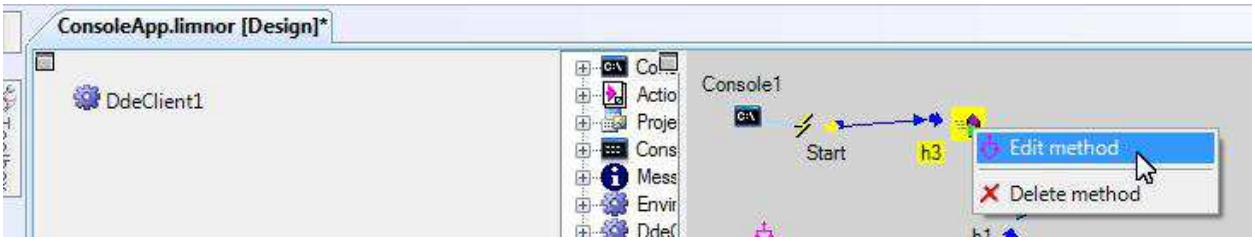

Finish the editing.
Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
CanWrite
False
DataType
(Object)
"OnExecuteComplete: "
Exception

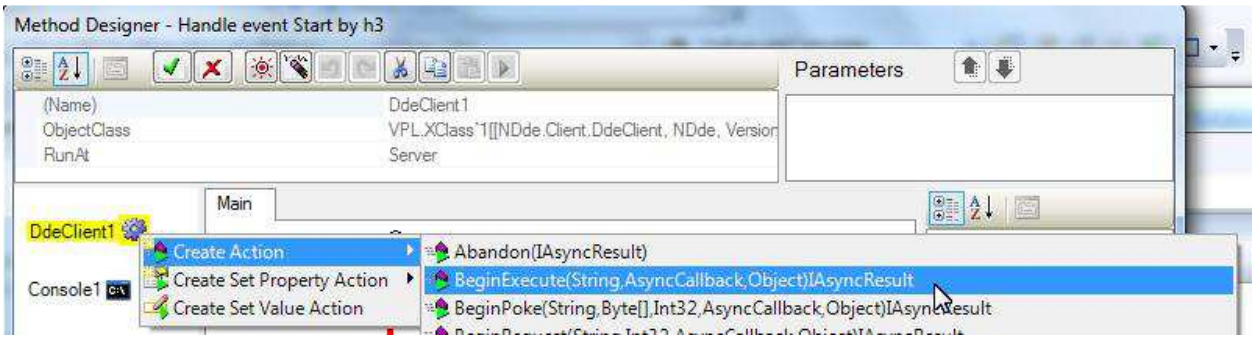


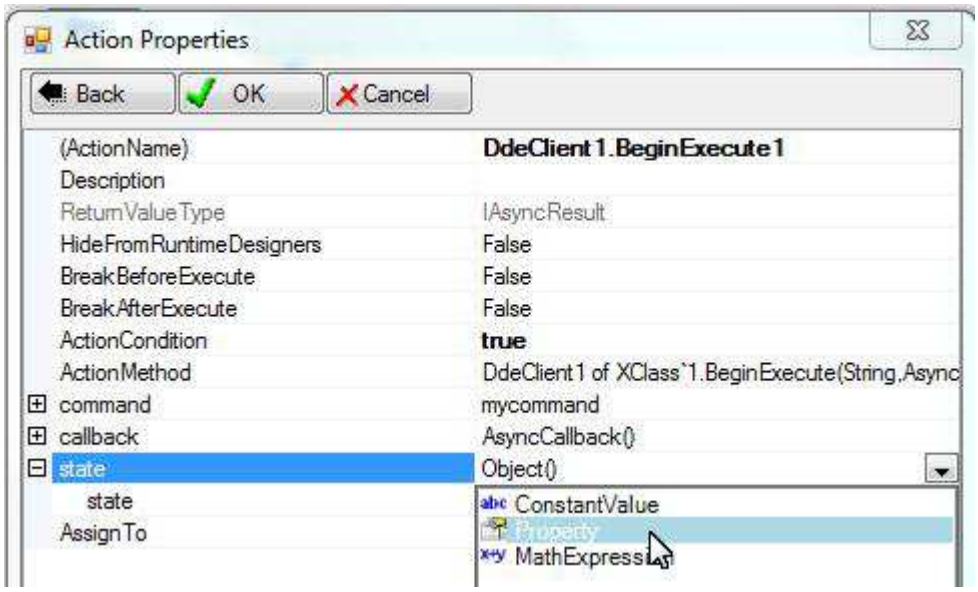


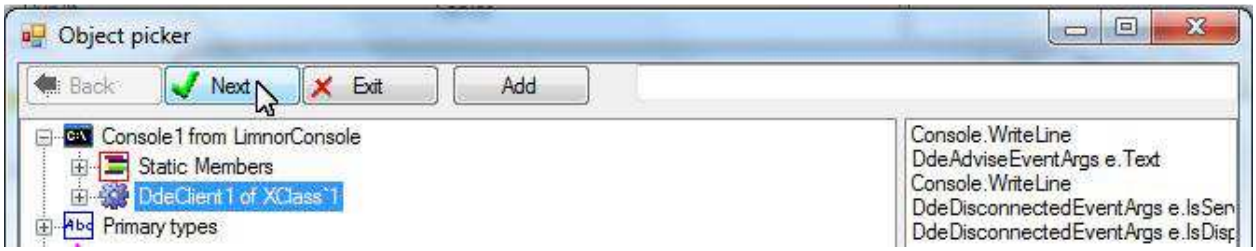
Asynchronous Execute Operation – Start Execute operation

Now we may create an asynchronous execute action.

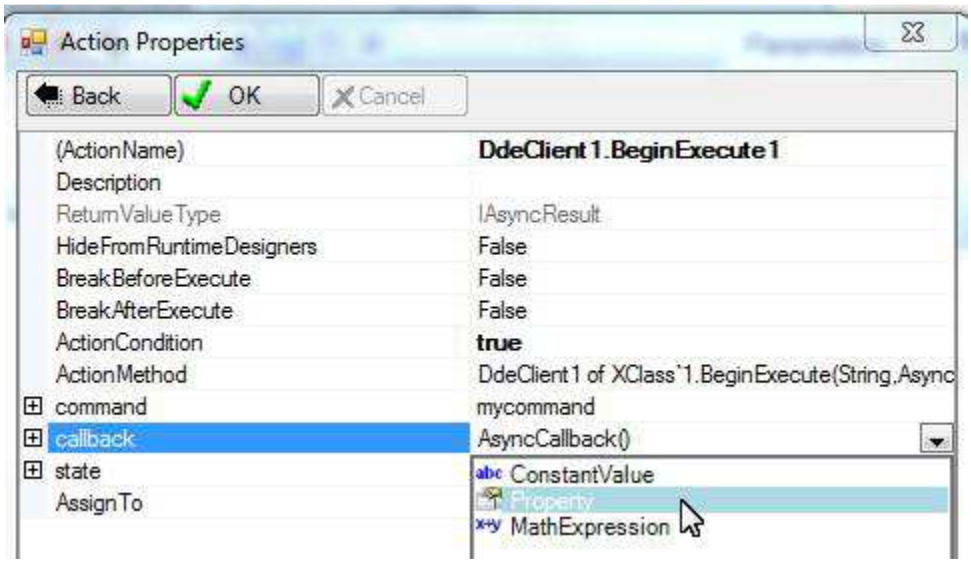


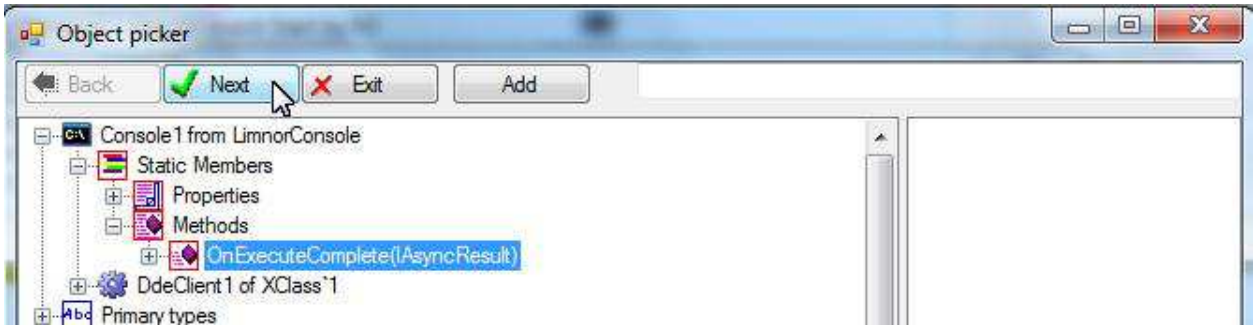


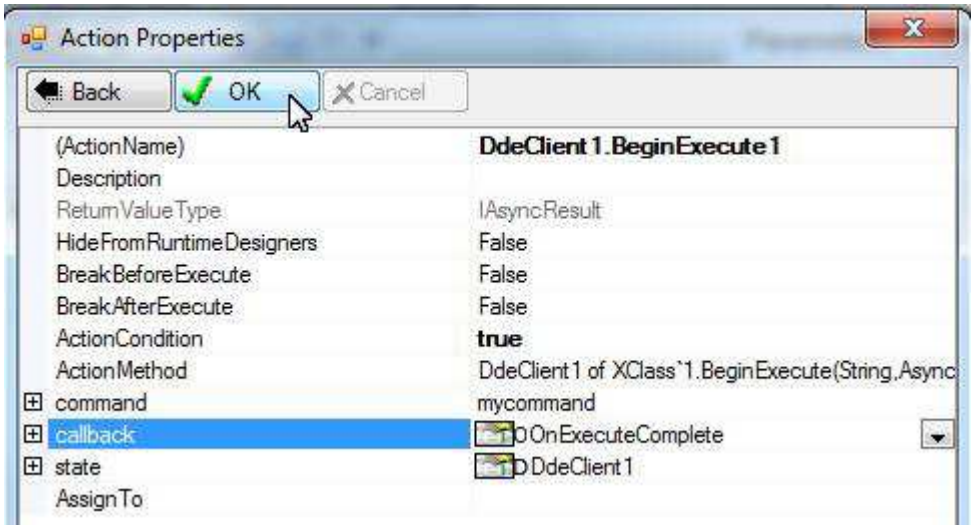




For “callback”, select OnExecuteComplete method we just created:

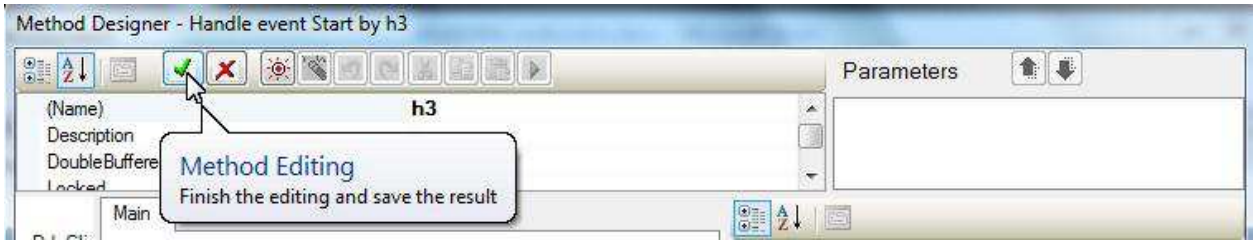




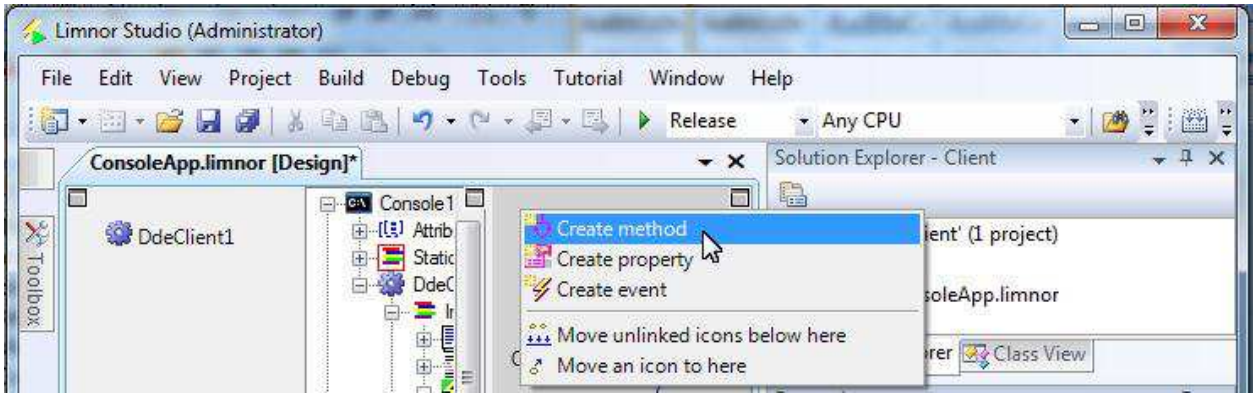


Asynchronous Poke Operation – Create callback function

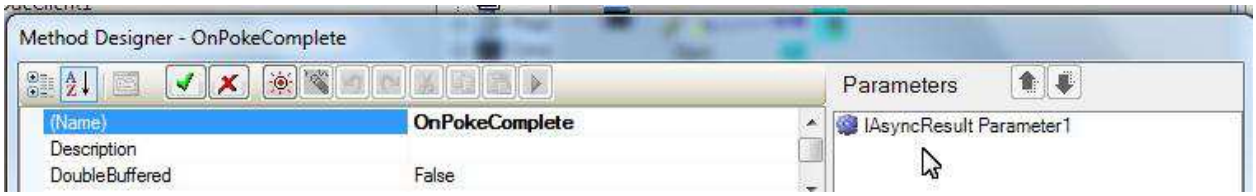
An asynchronous operation requires a callback function to be executed when the operation finishes. So, we create a method as the callback function. Close the current Method Editor so that we may create a new method:



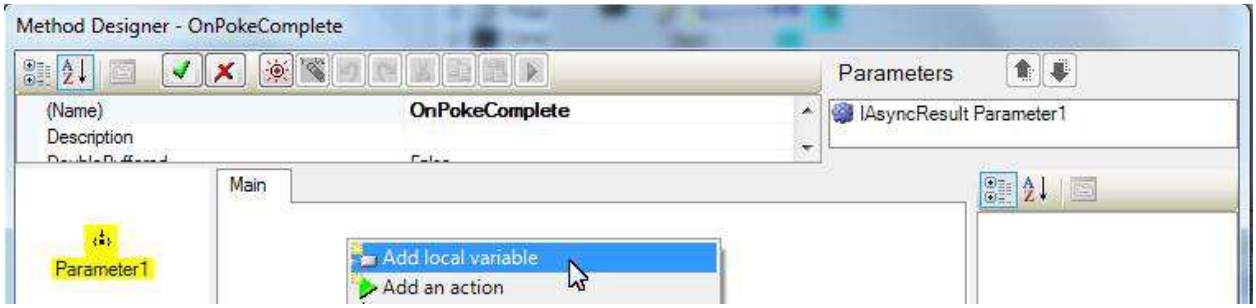
Create a new method:

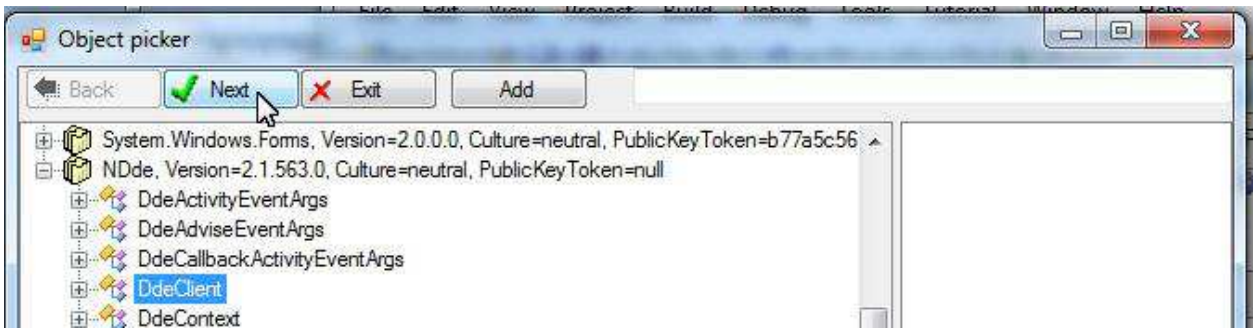


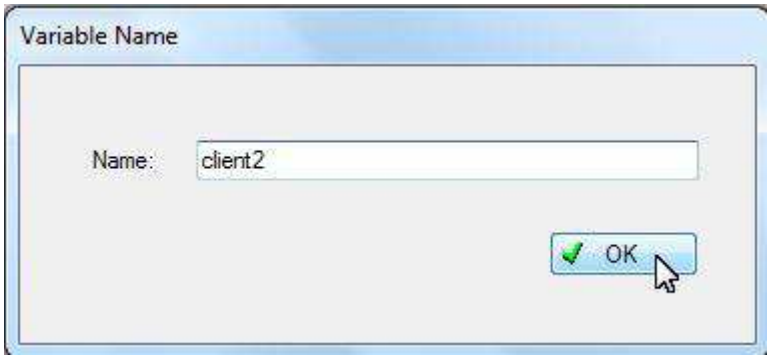
Rename the method to OnPokeComplete. Add a parameter of type `IAsyncResult`.

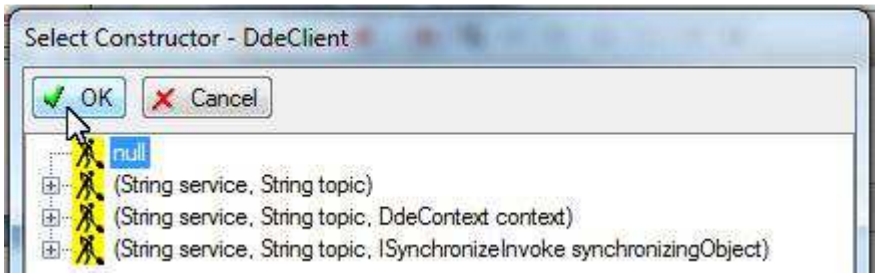


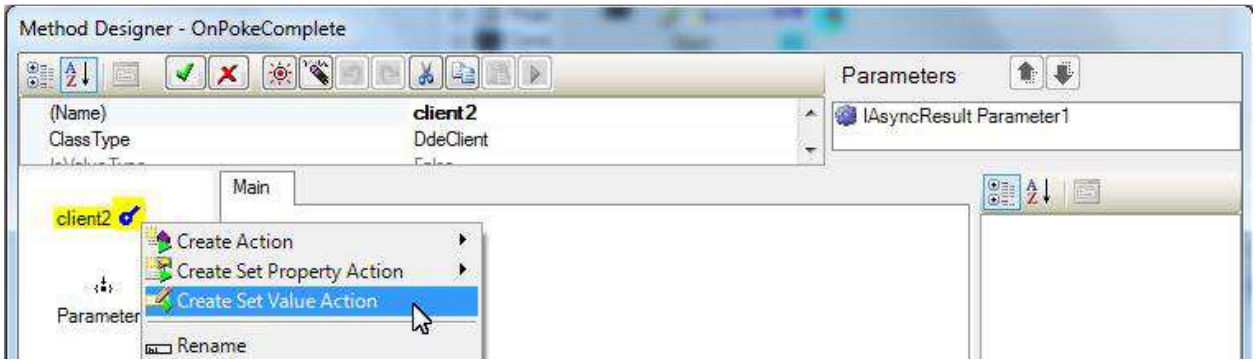
Get DdeClient from the method parameter:

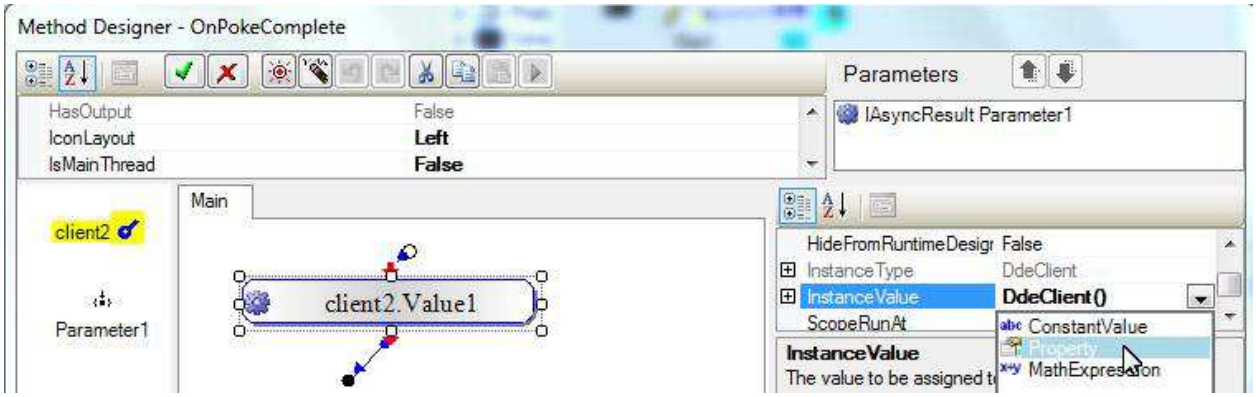




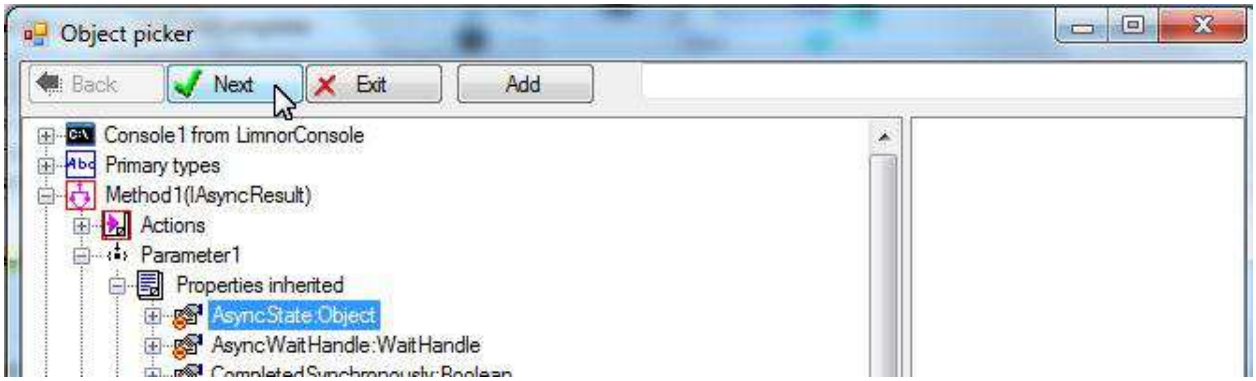


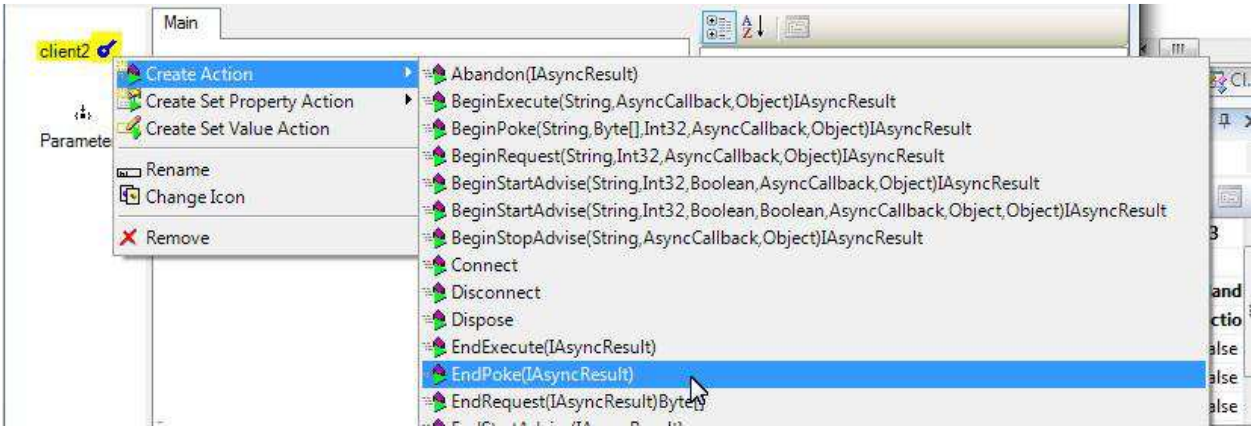


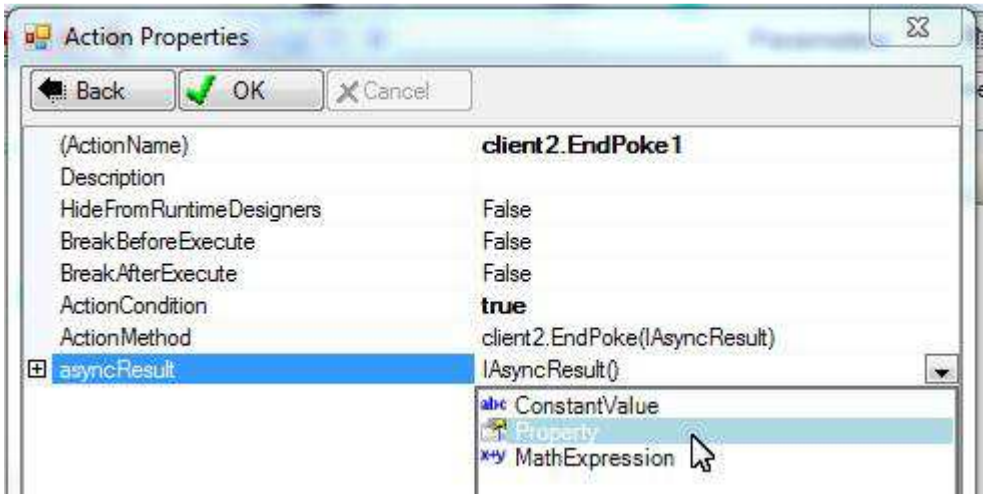


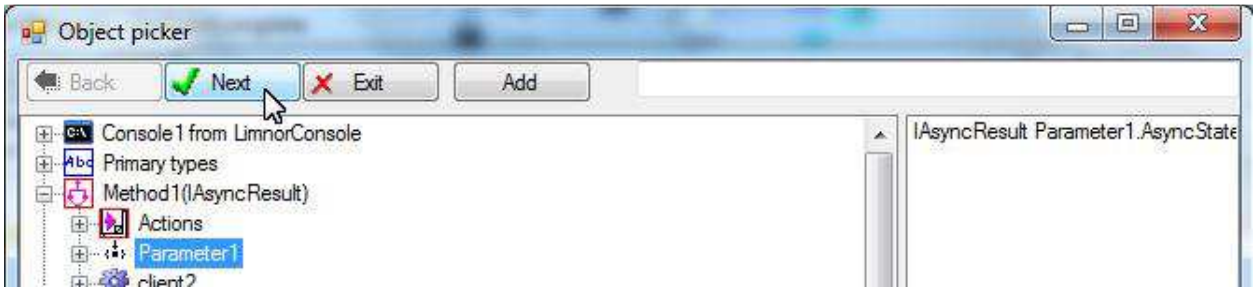


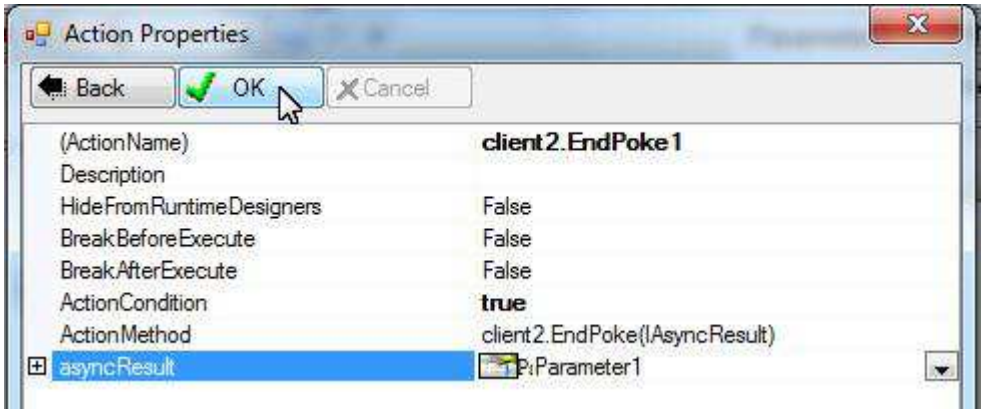
Execute EndPoke action:

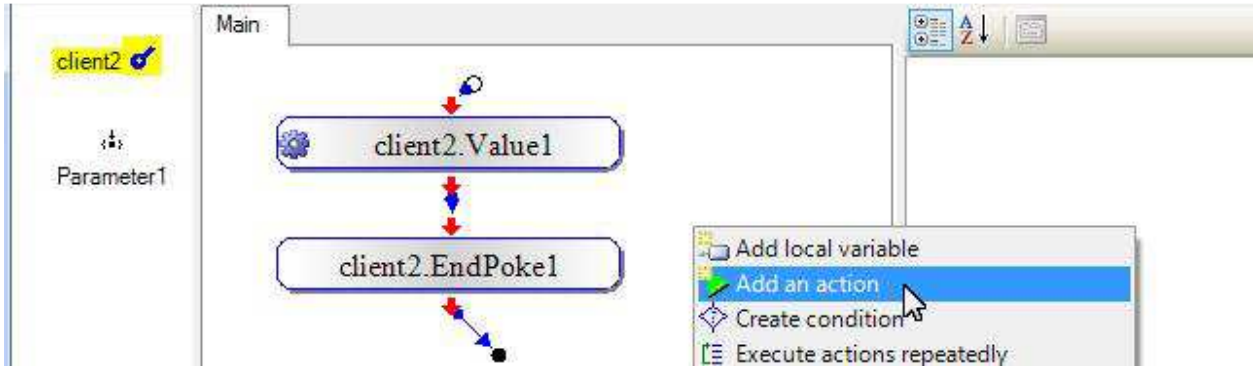

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
Method1(IAsyncResult)
Actions
Parameter1
Properties inherited
AsyncState:Object
AsyncWaitHandle:WaitHandle

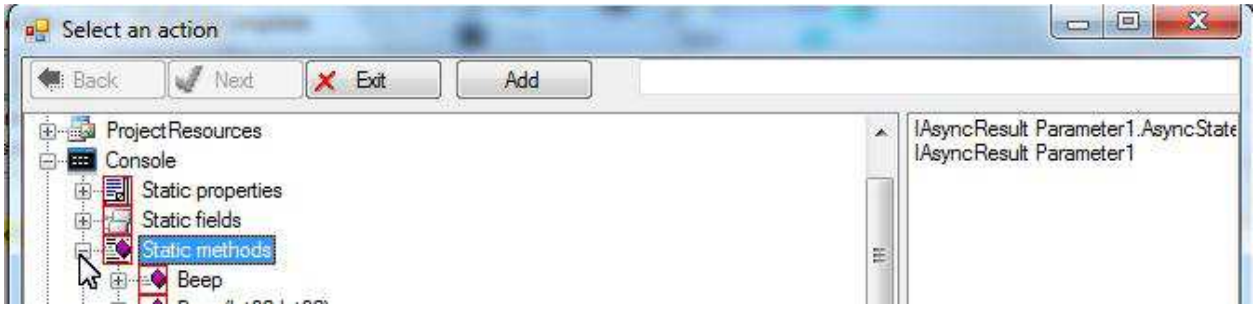

Main
client2
Create Action
Create Set Property Action
Create Set Value Action
Rename
Change Icon
Remove
Abandon(IAsyncResult)
BeginExecute(String,AsyncCallback,Object)IAsyncResult
BeginPoke(String,Byte[],Int32,AsyncCallback,Object)IAsyncResult
BeginRequest(String,Int32,AsyncCallback,Object)IAsyncResult
BeginStartAdvise(String,Int32,Boolean,AsyncCallback,Object)IAsyncResult
BeginStartAdvise(String,Int32,Boolean,Boolean,AsyncCallback,Object,Object)IAsyncResult
BeginStopAdvise(String,AsyncCallback,Object)IAsyncResult
Connect
Disconnect
Dispose
EndExecute(IAsyncResult)
EndPoke(IAsyncResult)

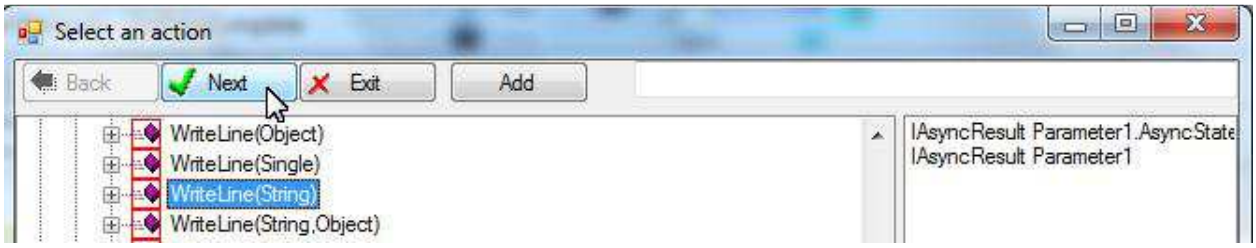

Action Properties
Back
OK
Cancel
(ActionName)
client2.EndPoke1
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
client2.EndPoke(IAsyncResult)
asyncResult
IAsyncResult()
ConstantValue
Property
MathExpression

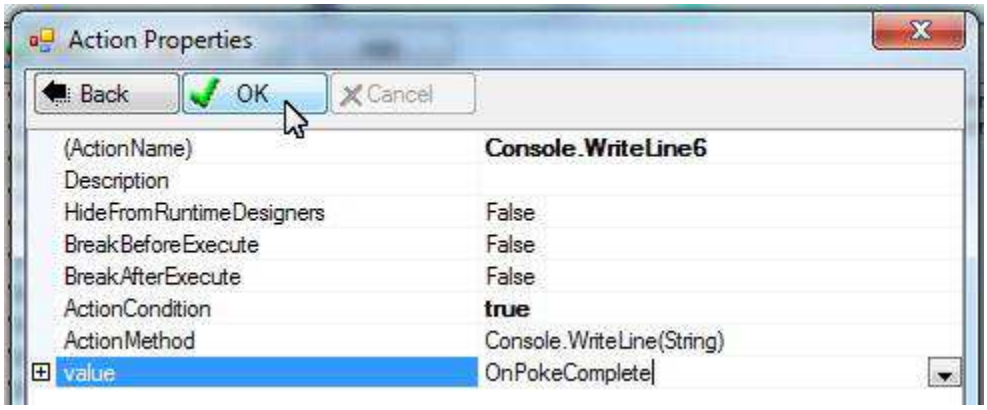

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
Method1(IAsyncResult)
Actions
Parameter1
IAsyncResult Parameter1.AsyncState

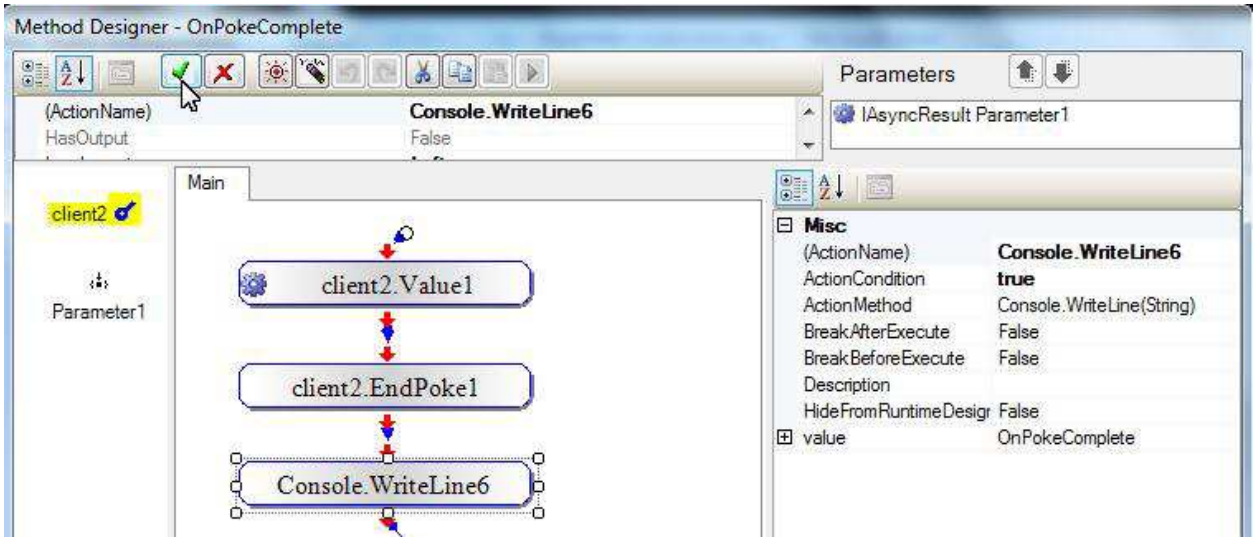

Action Properties
Back
OK
Cancel
(ActionName)
client2.EndPoke1
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
client2.EndPoke(IAsyncResult)
asyncResult
P:Parameter1


Main
client2
Parameter1
client2.Value1
client2.EndPoke1
Add local variable
Add an action
Create condition
Execute actions repeatedly

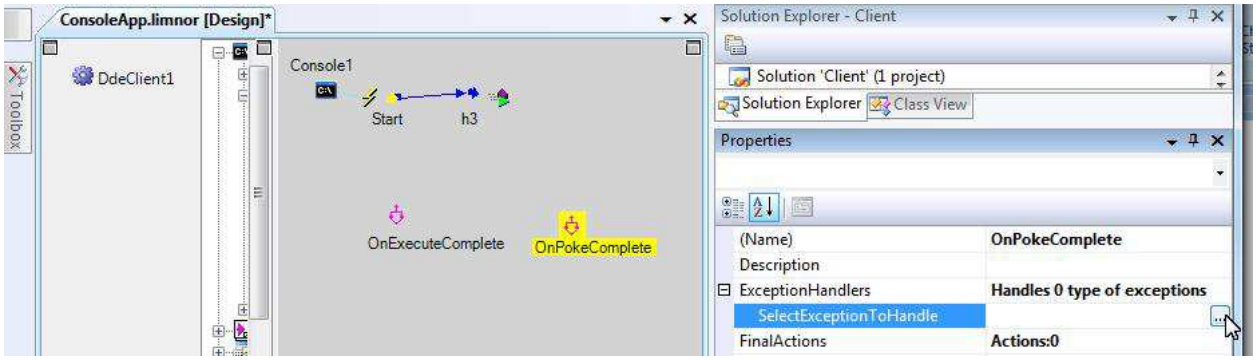

Select an action
Back
Next
Exit
Add
Project Resources
Console
Static properties
Static fields
Static methods
Beep
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1

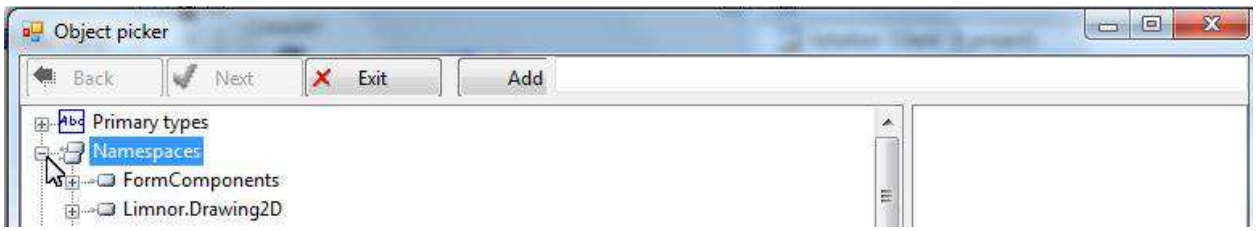

Select an action
Back
Next
Exit
Add
WriteLine(Object)
WriteLine(Single)
WriteLine(String)
WriteLine(String,Object)
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1





Asynchronous Poke Operation – Handler Exception

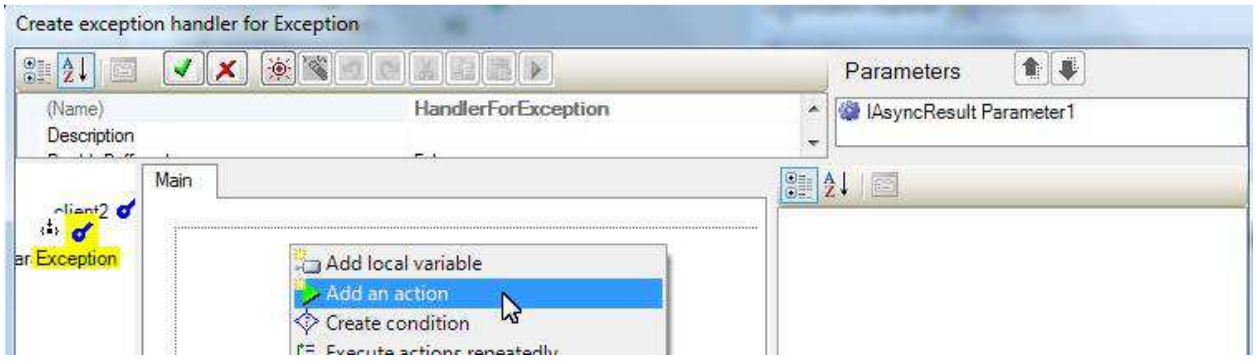


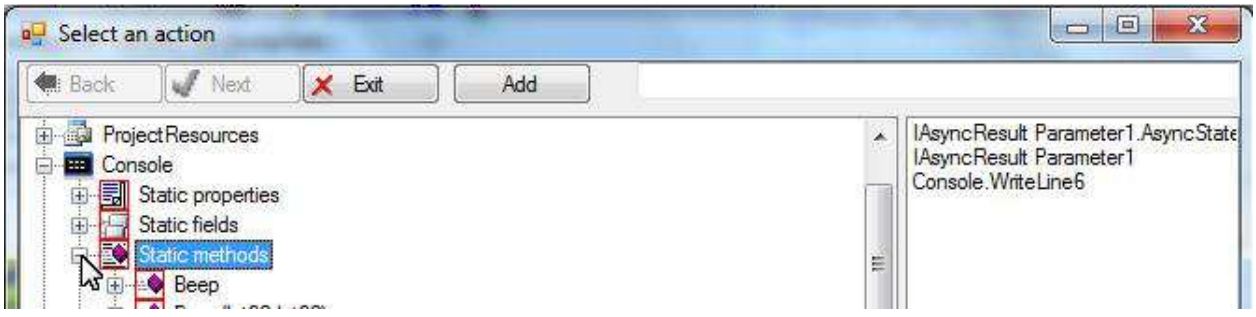


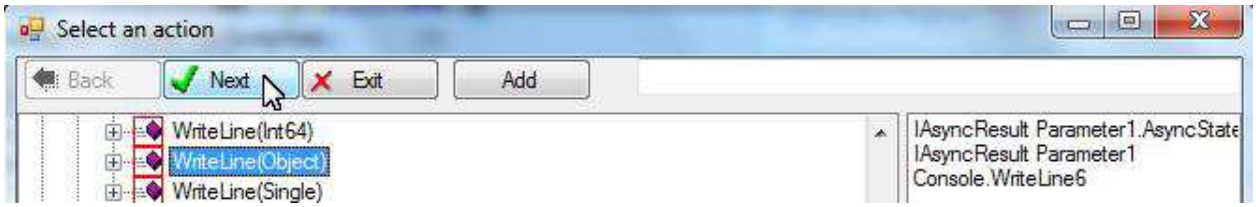

Object picker
Back
Next
Exit
Add
ProgUtility
System
AccessViolationException
AppDomainUnloadedException

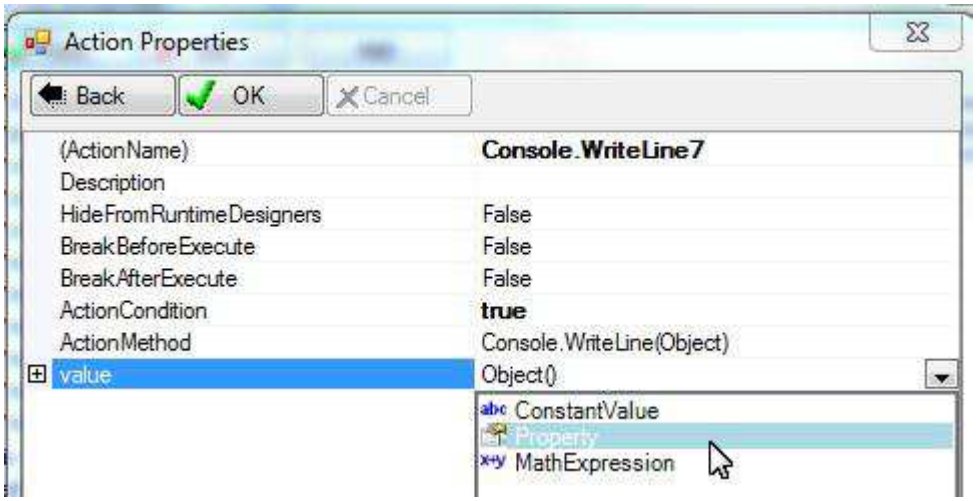

Object picker
Back
Next
Exit
Add
EntryPointNotFoundException
Exception
ExecutionEngineException

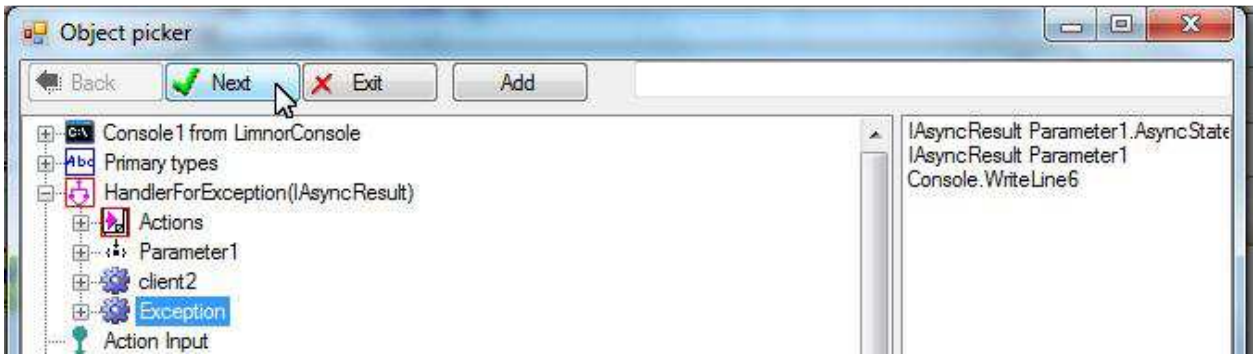

OnExecuteComplete
OnPokeComplete
(Name)
OnPokeComplete
Description
ExceptionHandlers
Handles 1 type of exceptions
Exception
SelectExceptionToHandle

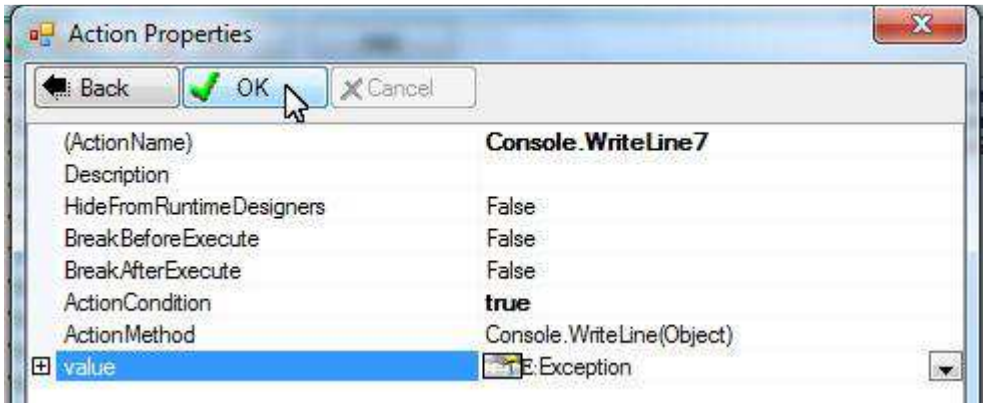

Create exception handler for Exception
Parameters
(Name)
HandlerForException
Description
IAsyncResult Parameter1
Main
Add local variable
Add an action
Create condition

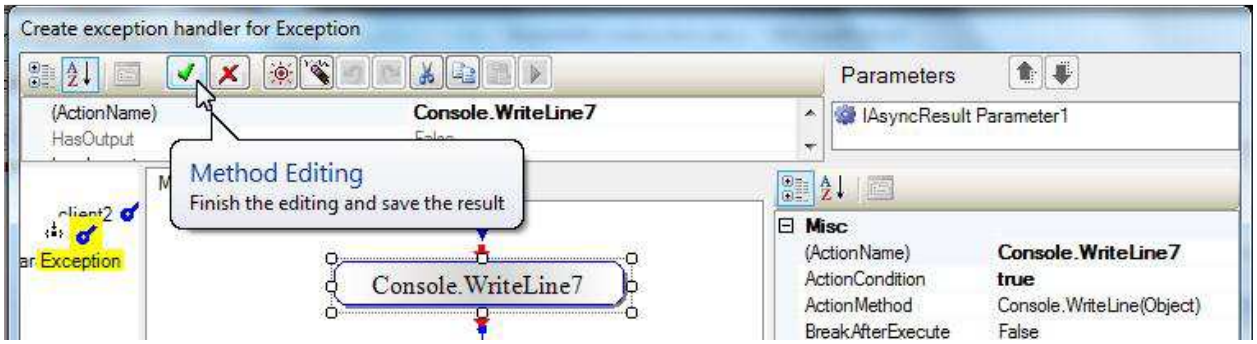

Select an action
Back
Next
Exit
Add
Project Resources
Console
Static properties
Static fields
Static methods
Beep
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6


Select an action
Back
Next
Exit
Add
WriteLine(Int64)
WriteLine(Object)
WriteLine(Single)
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6


Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine7
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(Object)
value
Object()
ConstantValue
Property
MathExpression

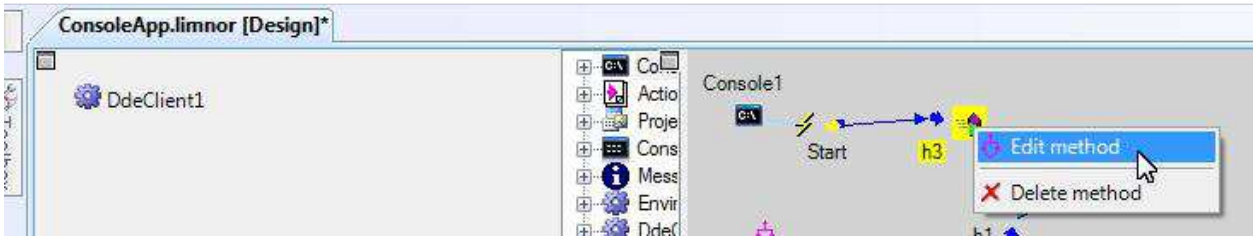

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
HandlerForException(IAsyncResult)
Actions
Parameter1
client2
Exception
Action Input
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6

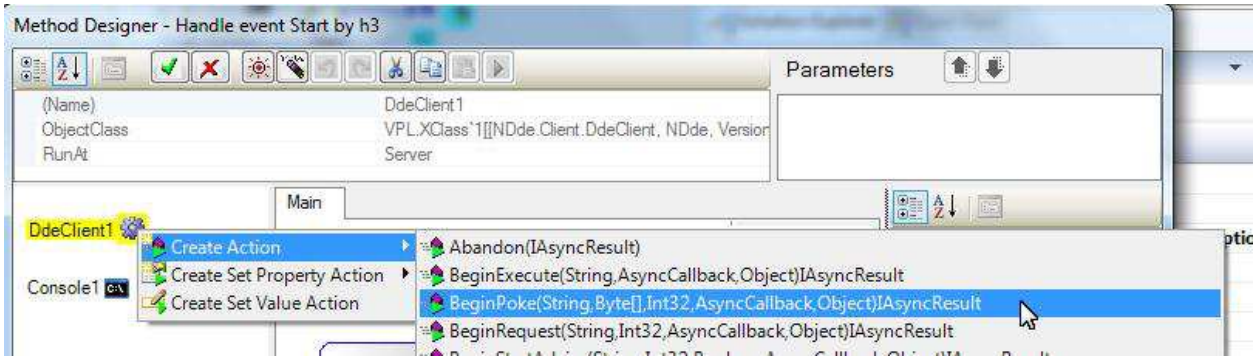

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine7
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(Object)
value
E:Exception

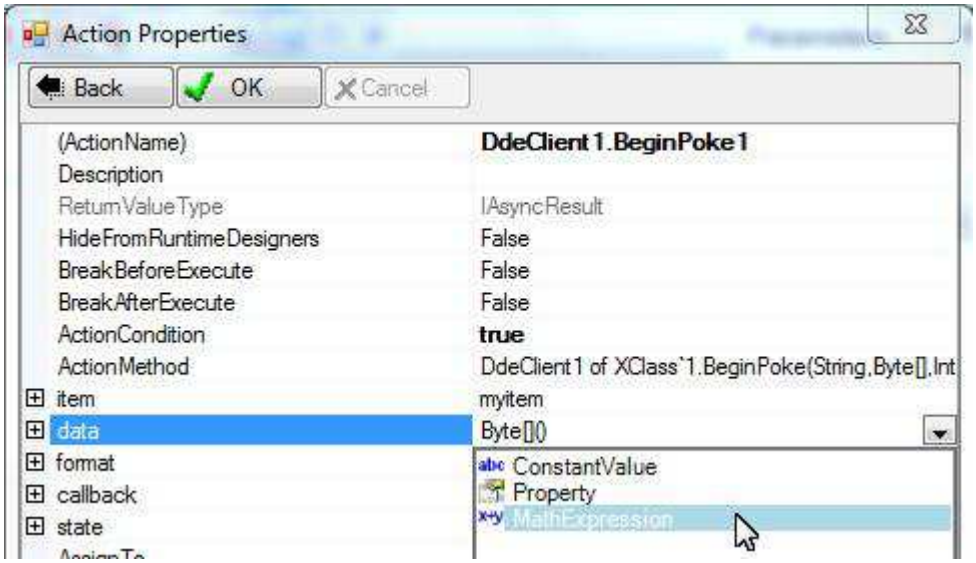

Create exception handler for Exception
Parameters
(ActionName)
Console.WriteLine7
HasOutput
IAsyncResult Parameter1
Method Editing
Finish the editing and save the result
Exception
Console.WriteLine7
Misc
(ActionName)
Console.WriteLine7
ActionCondition
true
ActionMethod
Console.WriteLine(Object)
BreakAfterExecute
False

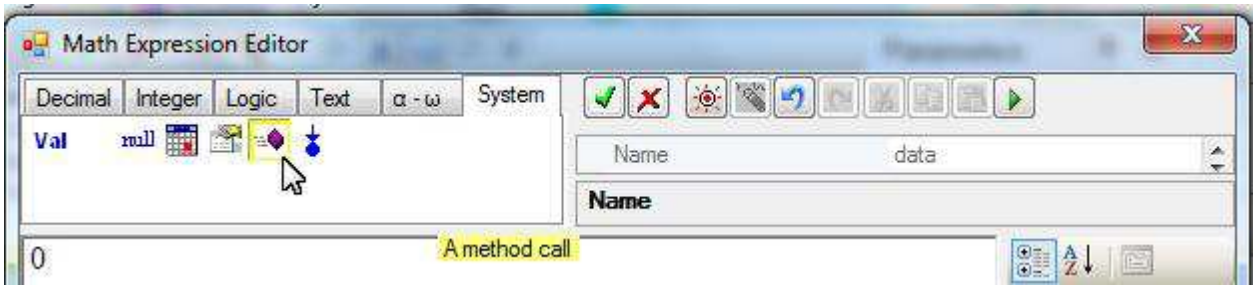
Asynchronous Poke Operation – Start Poke operation

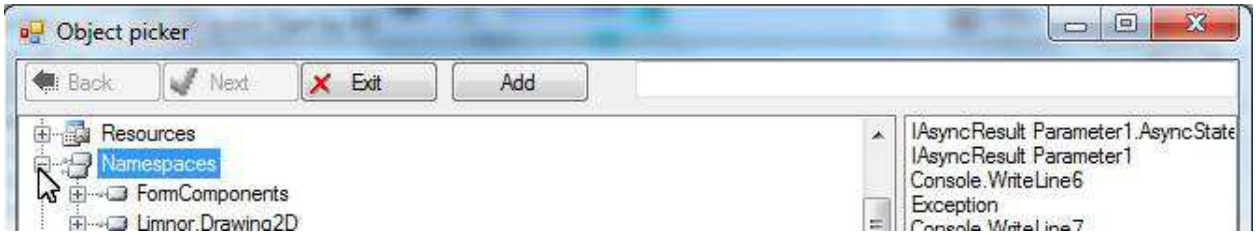
Now we may create an asynchronous poke action.

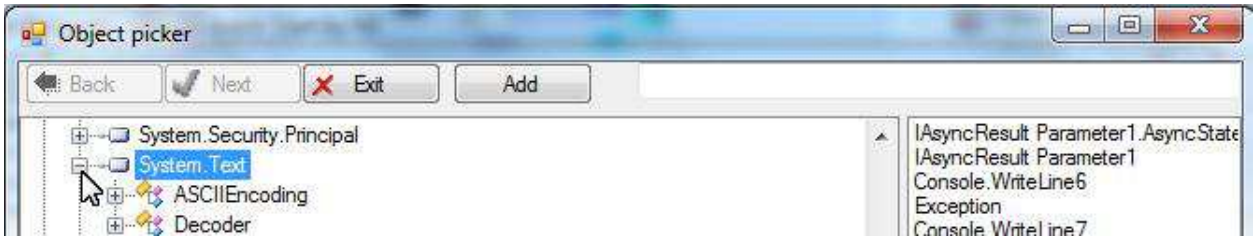


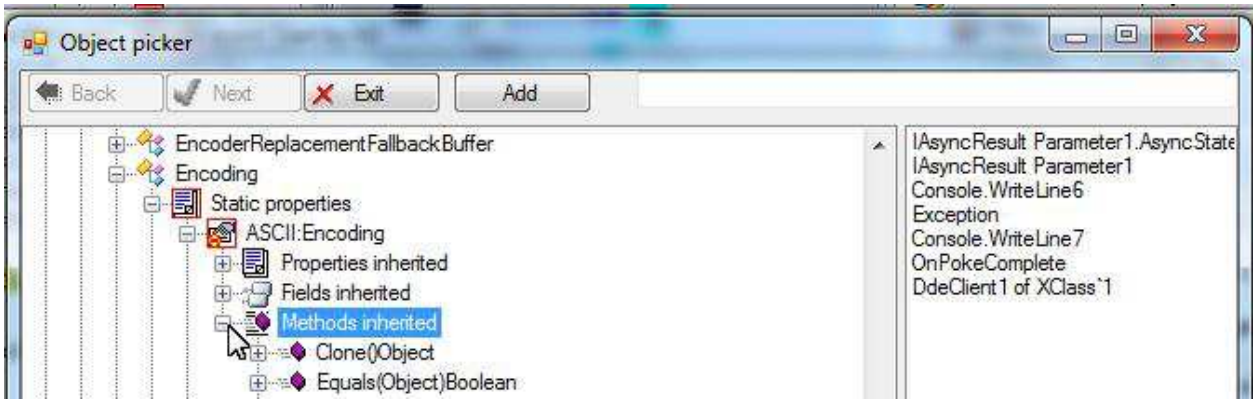


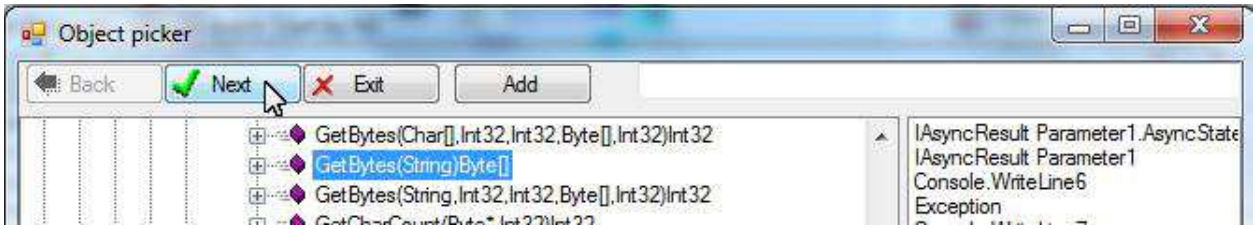


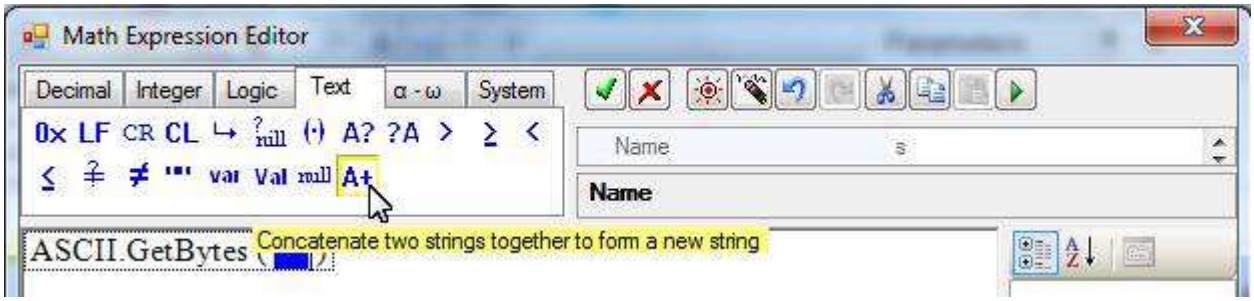

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
Name
data
Name
A method call
0

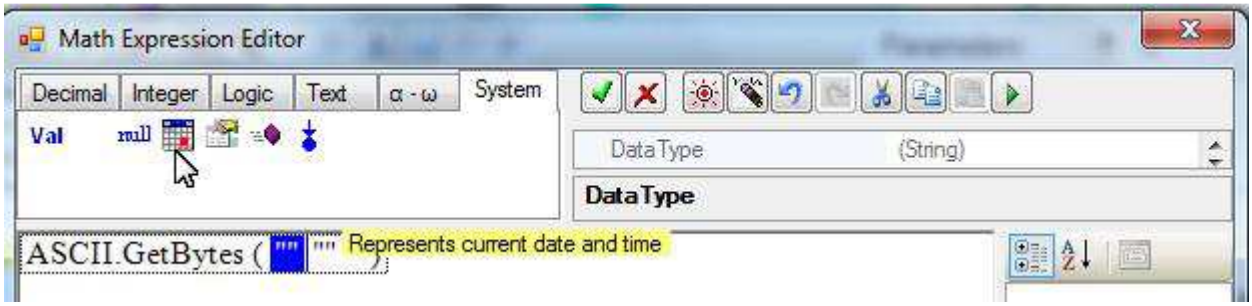

Object picker
Back
Next
Exit
Add
Resources
Namespaces
FormComponents
Limnor.Drawing2D
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception

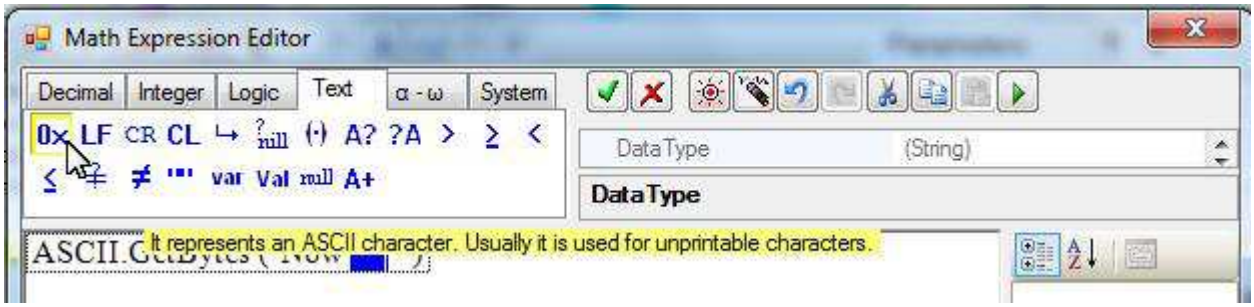

Object picker
Back
Next
Exit
Add
System.Security.Principal
System.Text
ASCIIEncoding
Decoder
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception

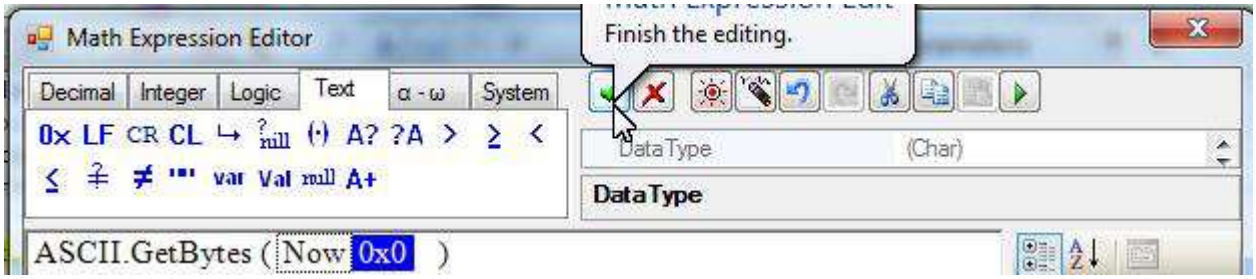

Object picker
Back
Next
Exit
Add
EncoderReplacementFallbackBuffer
Encoding
Static properties
ASCII:Encoding
Properties inherited
Fields inherited
Methods inherited
Clone()Object
Equals(Object)Boolean
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception
Console.WriteLine7
OnPokeComplete
DdeClient1 of XClass`1


Object picker
Back
Next
Exit
Add
GetBytes(Char[],Int32,Int32,Byte[],Int32)Int32
GetBytes(String)Byte[]
GetBytes(String,Int32,Int32,Byte[],Int32)Int32
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception

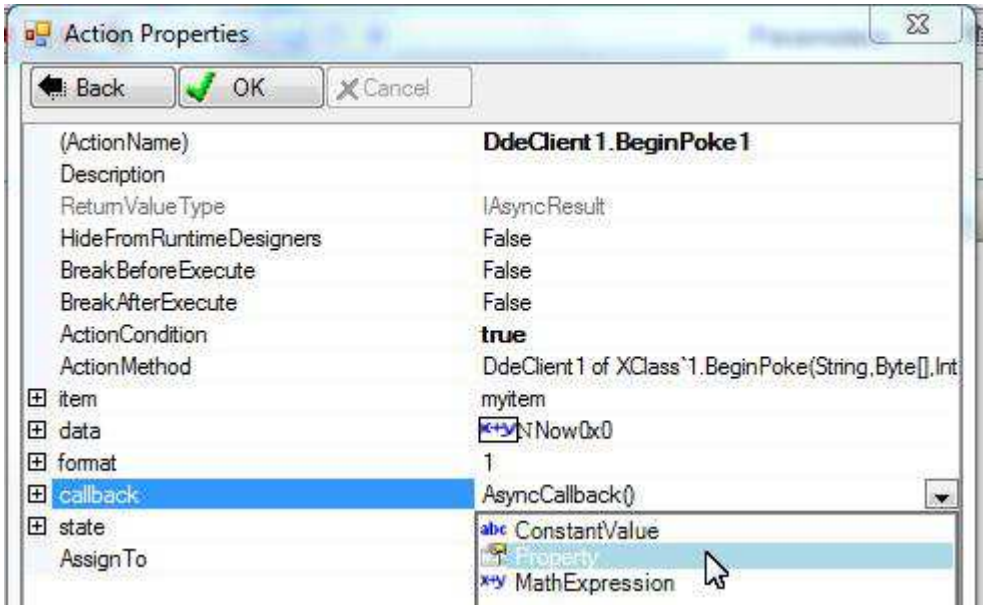


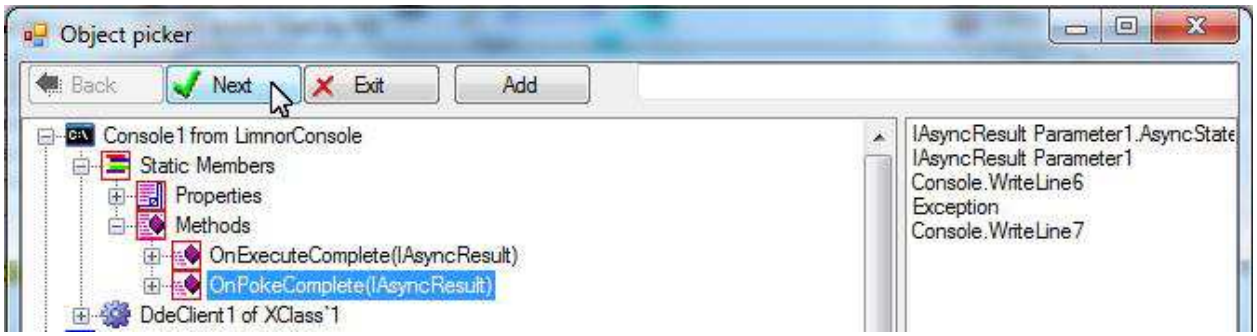


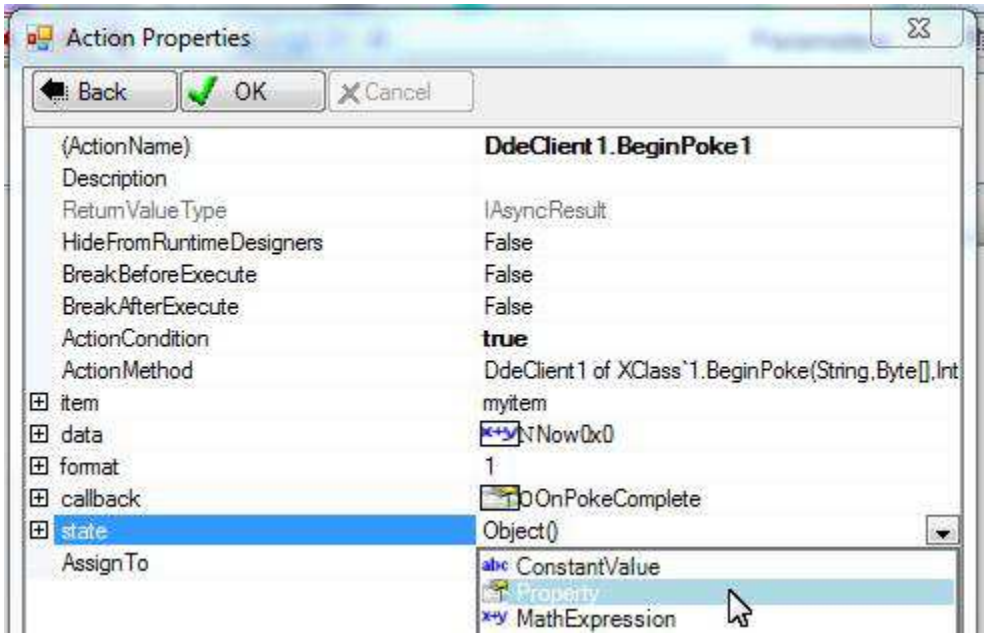


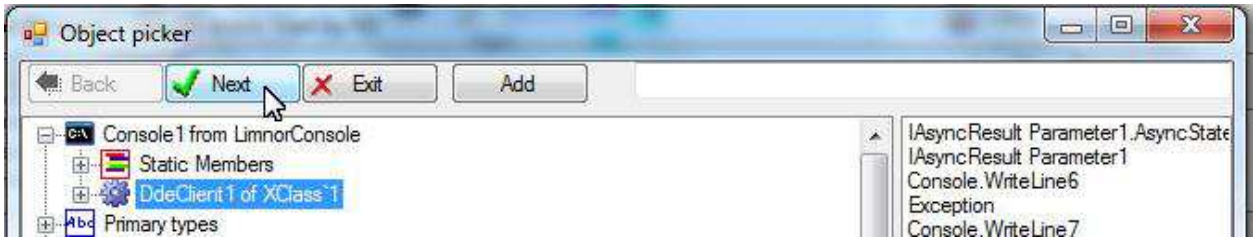


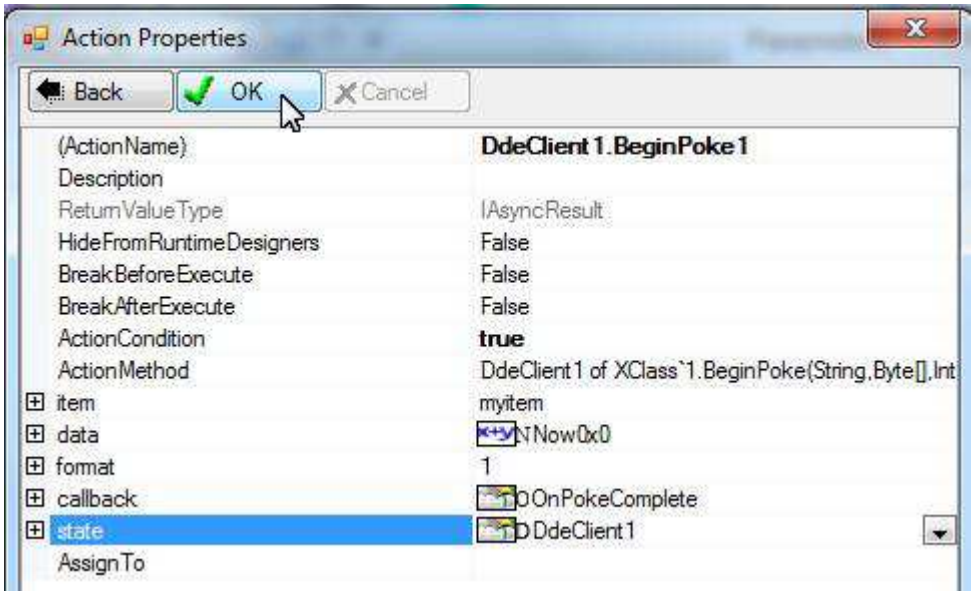
Provide the callback function:


Action Properties
Back
OK
Cancel
(ActionName)
DdeClient1.BeginPoke1
Description
ReturnValueType
IAsyncResult
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
DdeClient1 of XClass`1.BeginPoke(String,Byte[],Int
item
myitem
data
Now()x0
format
1
callback
AsyncCallback()
state
AssignTo
ConstantValue
Property
MathExpression


Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Static Members
Properties
Methods
OnExecuteComplete(IAsyncResult)
OnPokeComplete(IAsyncResult)
DdeClient1 of XClass`1
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception
Console.WriteLine7

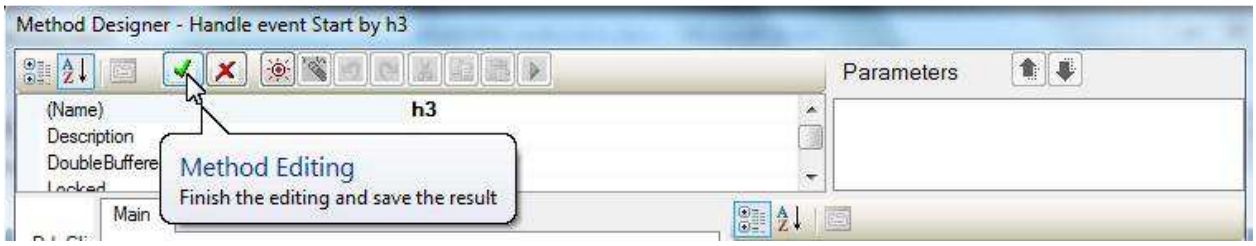

Action Properties
Back
OK
Cancel
(ActionName)
DdeClient1.BeginPoke1
Description
ReturnValueType
IAsyncResult
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
DdeClient1 of XClass`1.BeginPoke(String,Byte[],Int
item
myitem
data
Now()x0
format
1
callback
OnPokeComplete
state
Object()
AssignTo
ConstantValue
Property
MathExpression



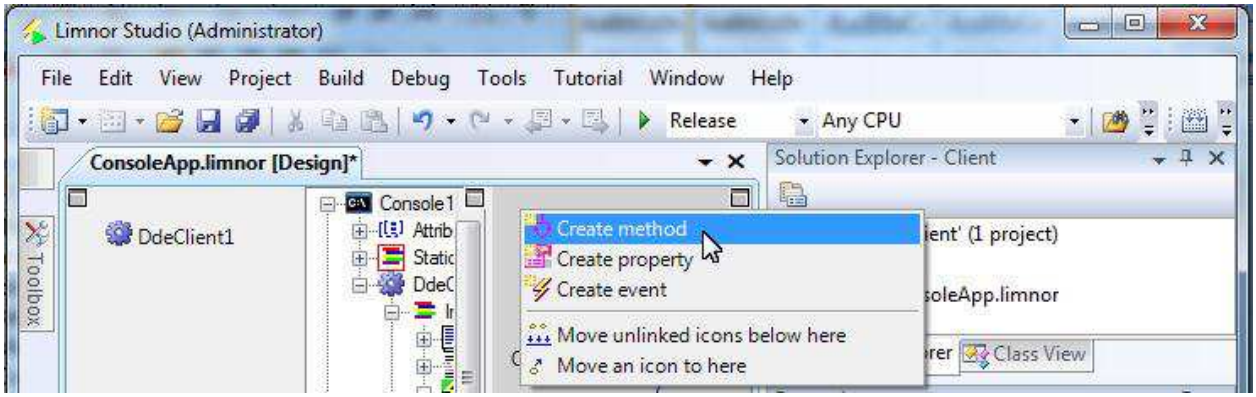


Asynchronous Request Operation – Create callback function

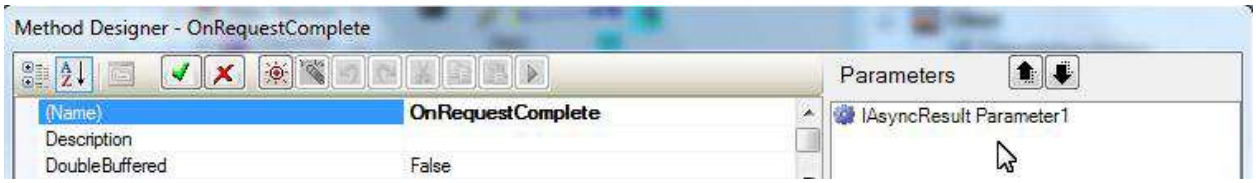
An asynchronous operation requires a callback function to be executed when the operation finishes. So, we create a method as the callback function. Close the current Method Editor so that we may create a new method:



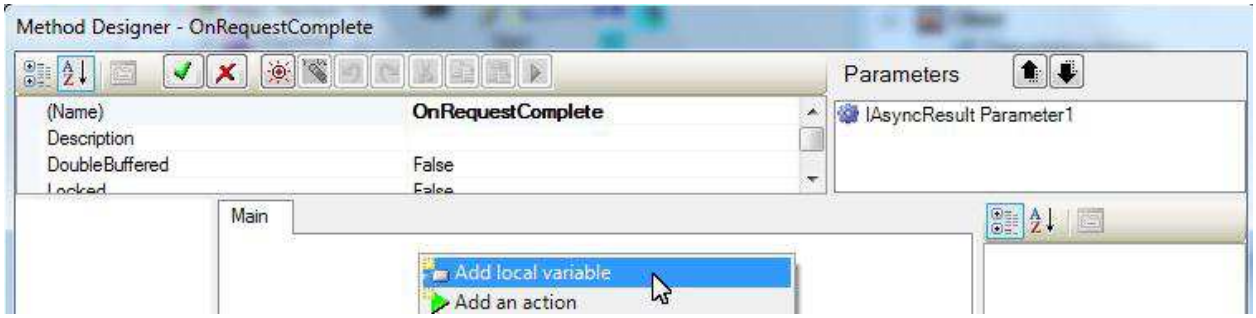
Create a new method:

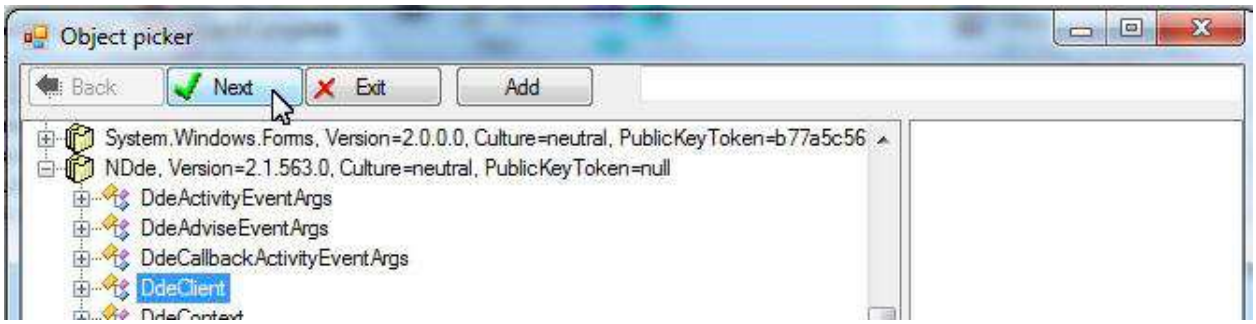


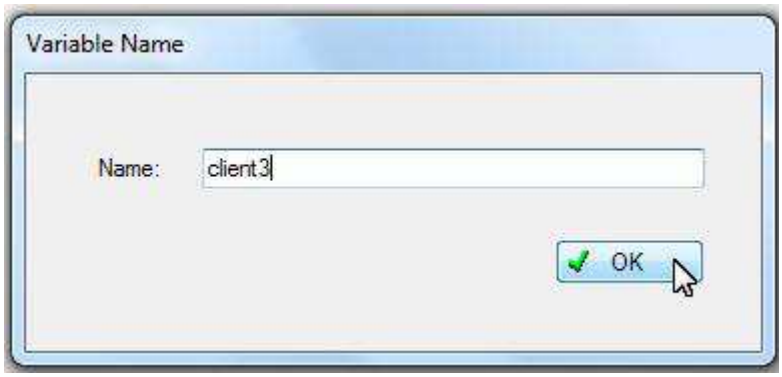
Rename the method to OnRequestComplete. Add a parameter of type `IAsyncResult`.

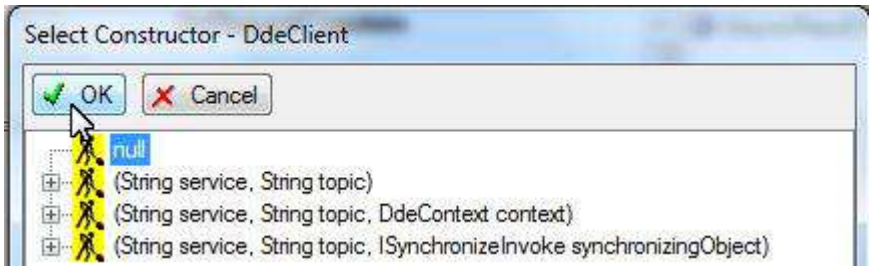


Get DdeClient from the method parameter:

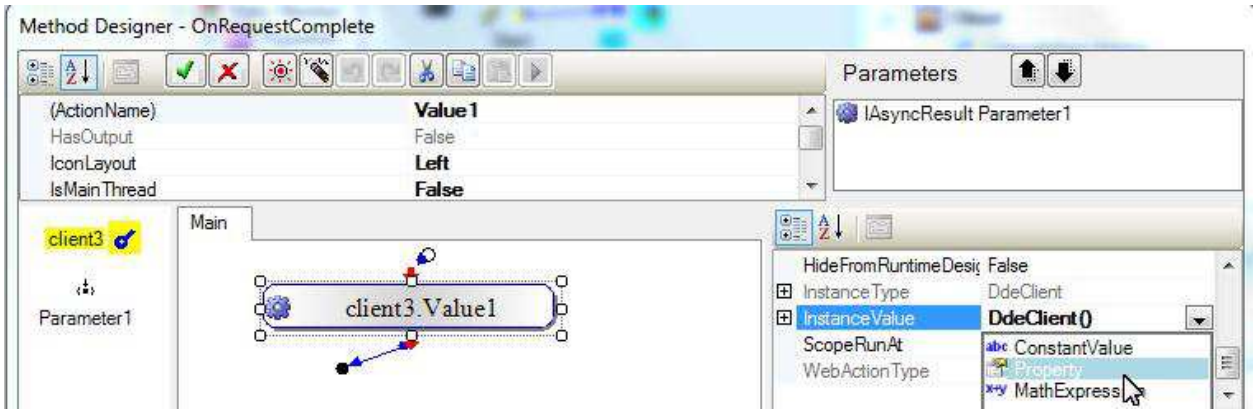


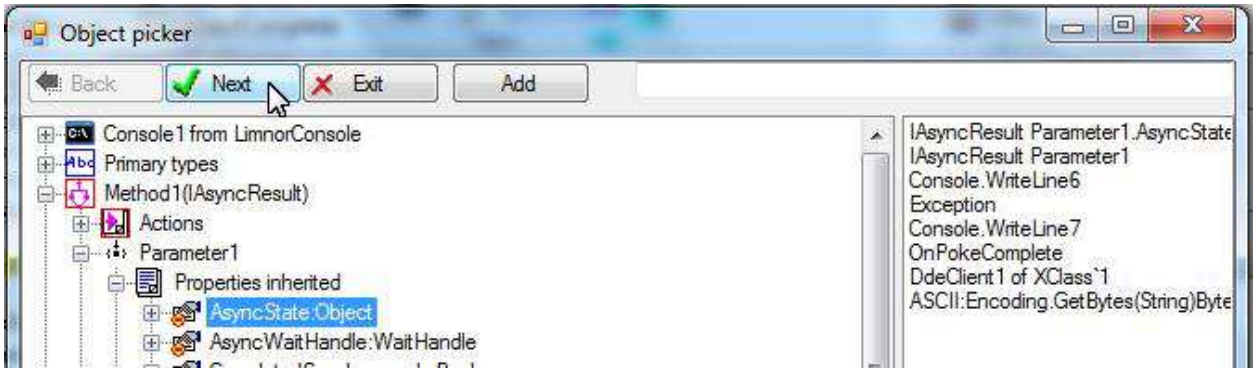



Variable Name
Name:
client3
OK

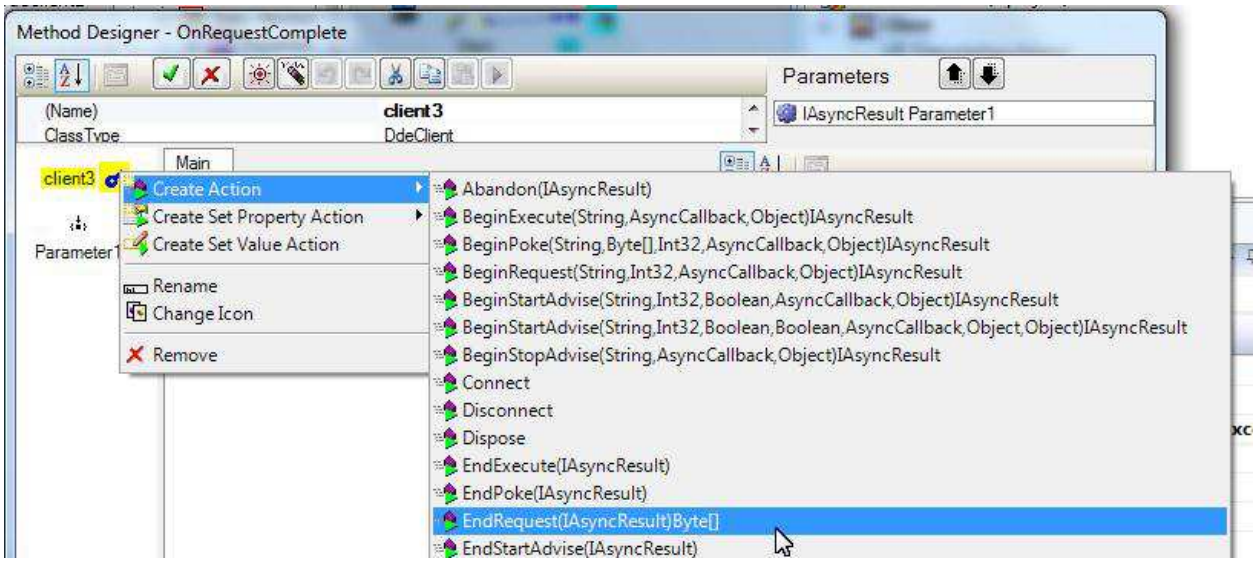

Select Constructor - DdeClient
OK
Cancel
null
(String service, String topic)
(String service, String topic, DdeContext context)
(String service, String topic, ISynchronizeInvoke synchronizingObject)

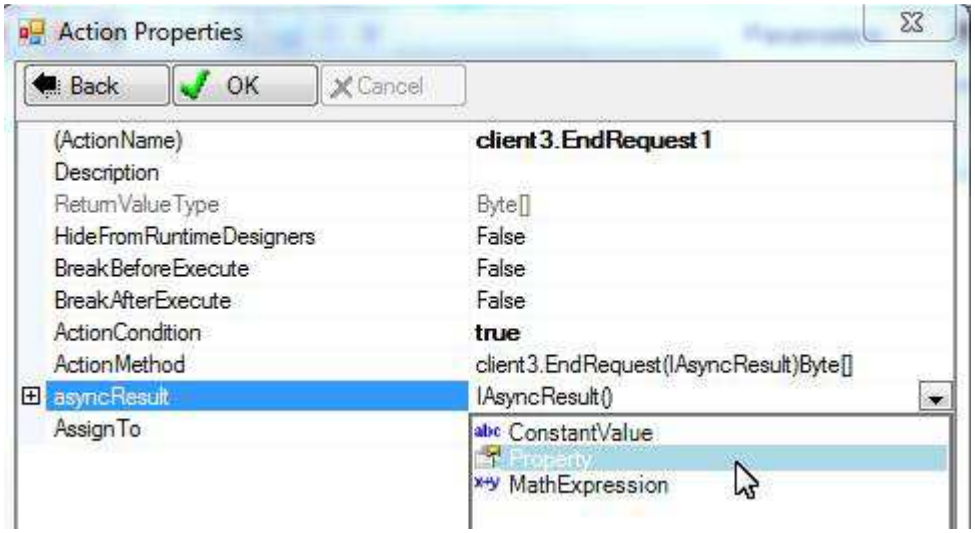

Method Designer - OnRequestComplete
Parameters
(Name)
client3
ClassType
DdeClient
IsValueType
False
IAsyncResult Parameter1
Main
client3
Parameter1
Create Action
Create Set Property Action
Create Set Value Action

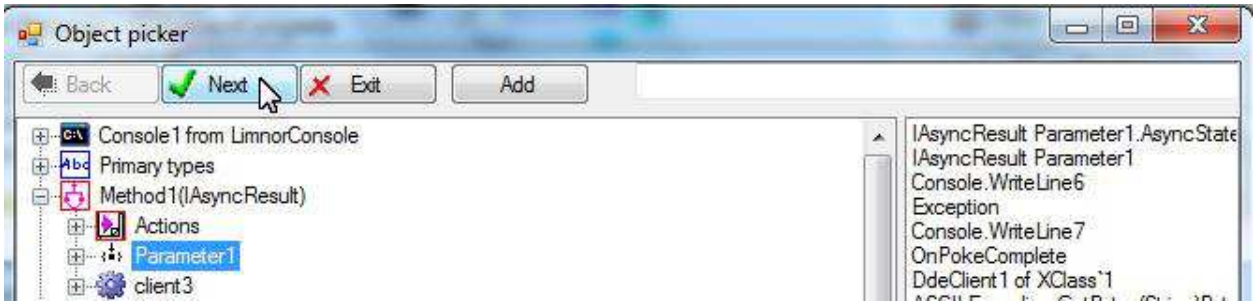

Method Designer - OnRequestComplete
Parameters
(ActionName)
Value1
HasOutput
False
IconLayout
Left
IsMainThread
False
IAsyncResult Parameter1
Main
client3
Parameter1
client3.Value1
InstanceType
DdeClient
InstanceValue
DdeClient()
ScopeRunAt
WebActionType
ConstantValue
Property
MathExpress



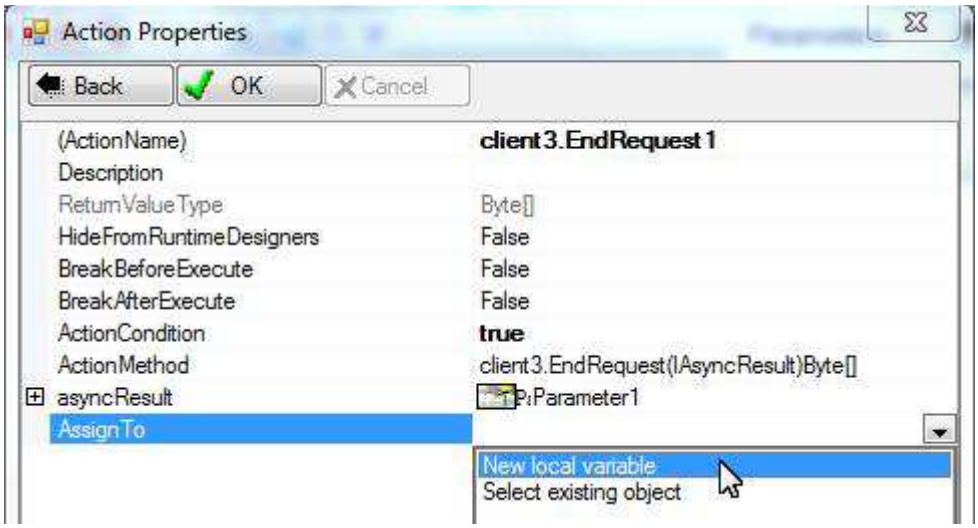
Execute EndRequest to get data:

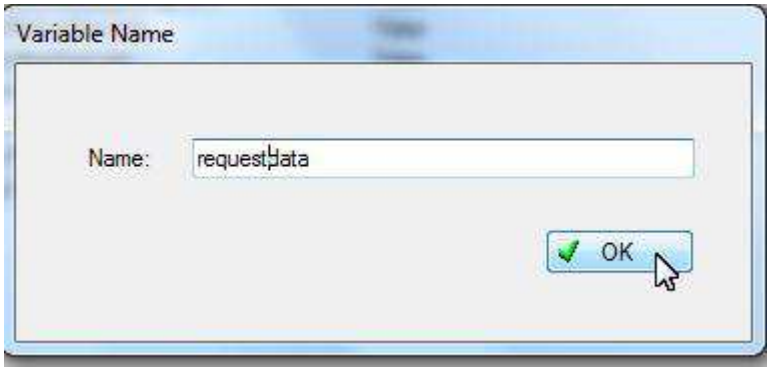


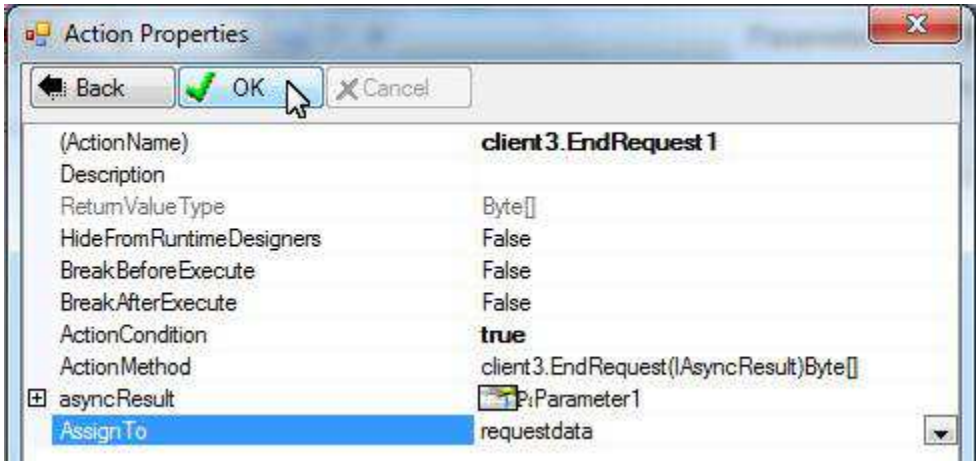




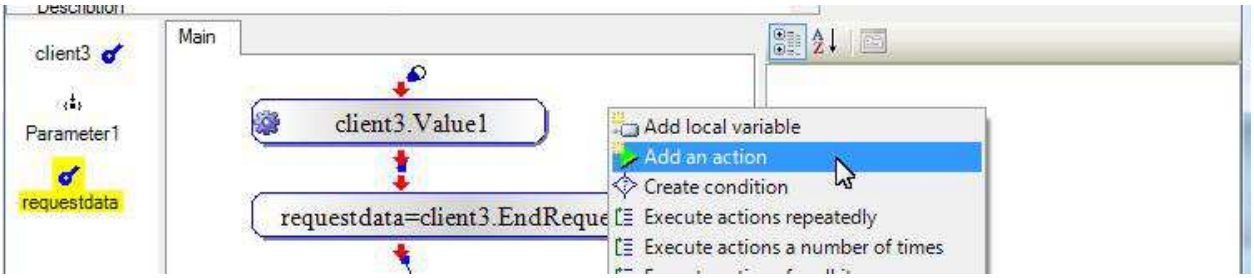
Use a new variable to get the request data:

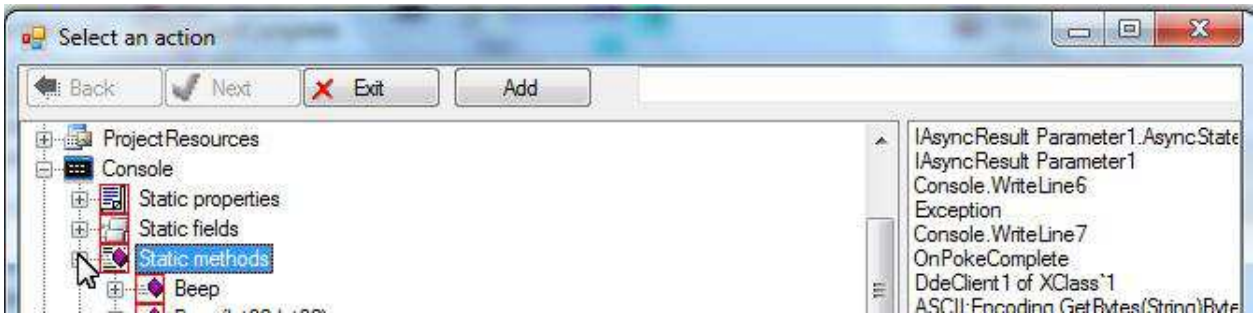


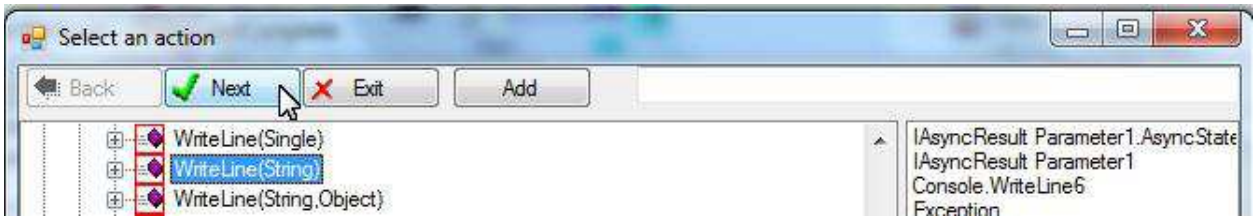


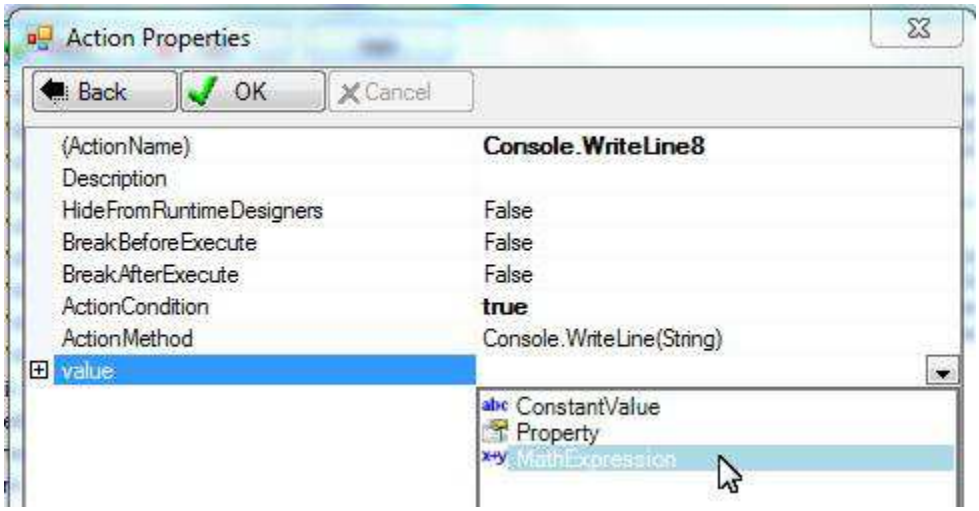


Show the data on the console:



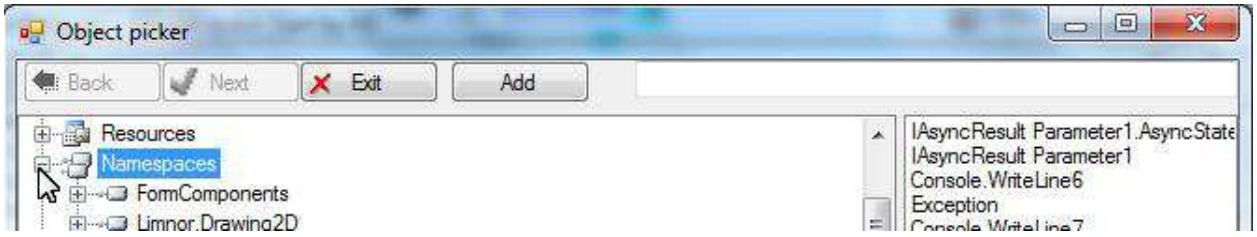


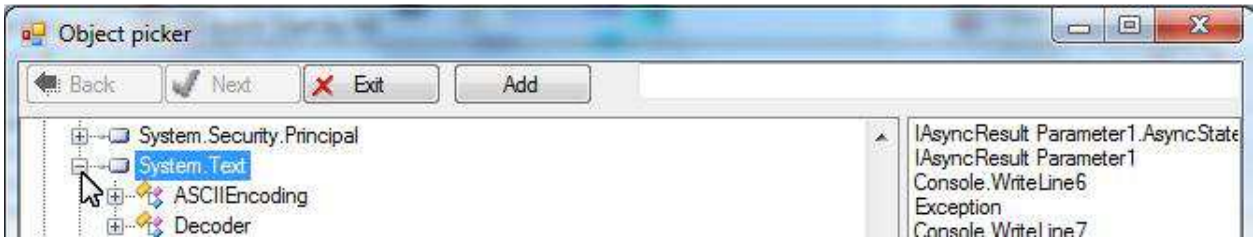


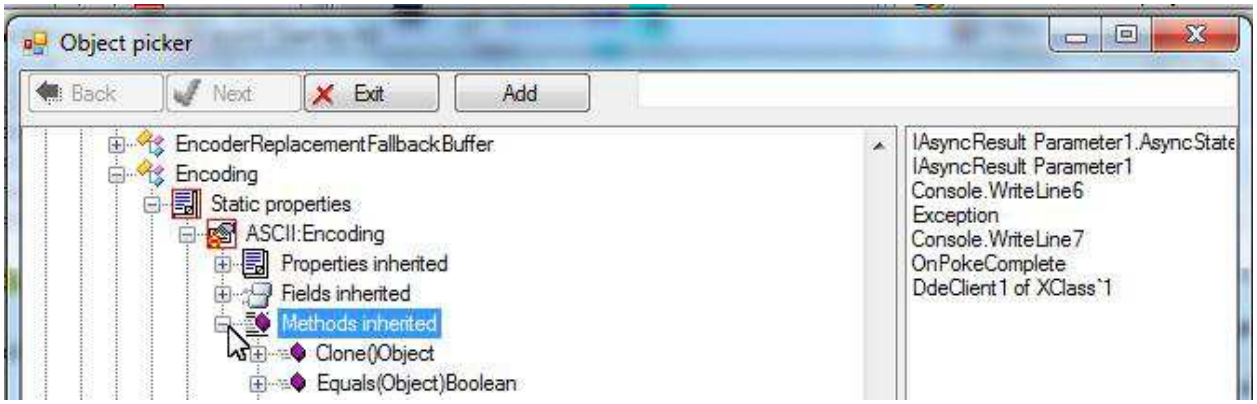

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine8
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(String)
value
ConstantValue
Property
MathExpression

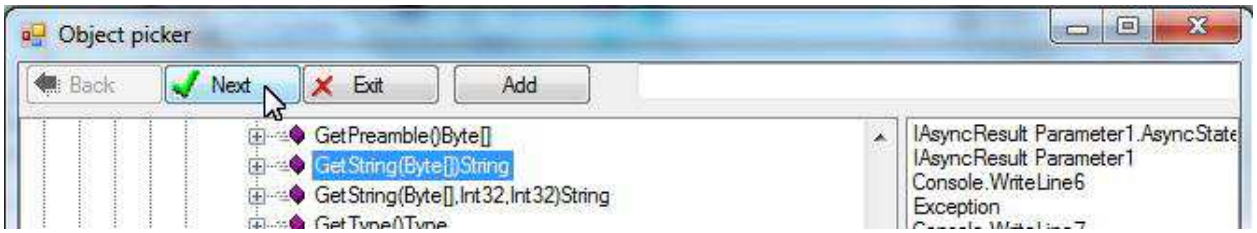

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Concatenate two strings together to form a new string

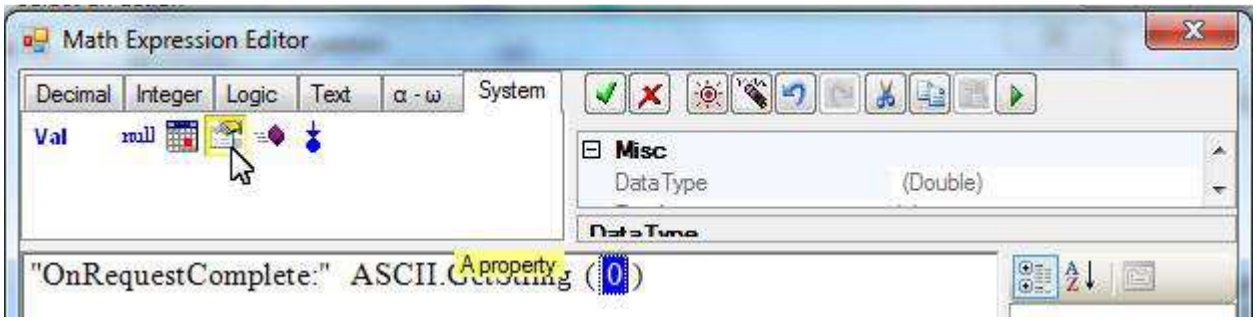

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Misc
Data Type
(String)
"OnRequestComplete:"
A method call

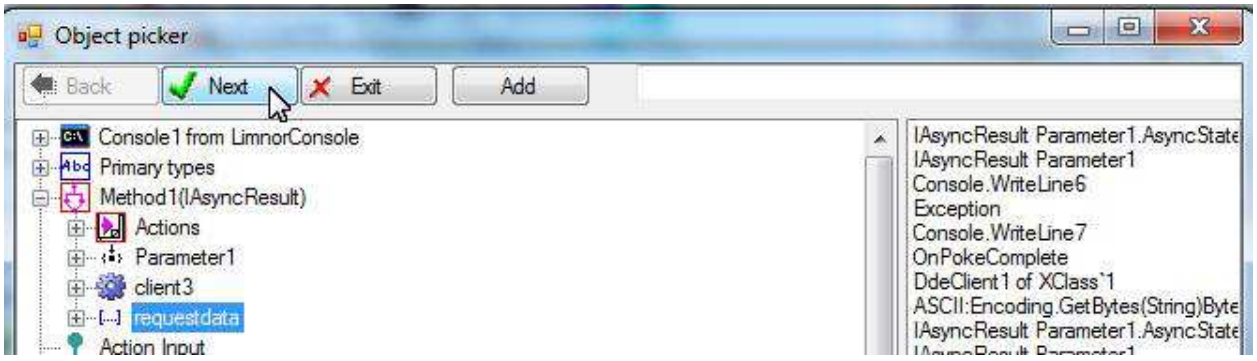

Object picker
Back
Next
Exit
Add
Resources
Namespaces
FormComponents
Limnor.Drawing2D
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception

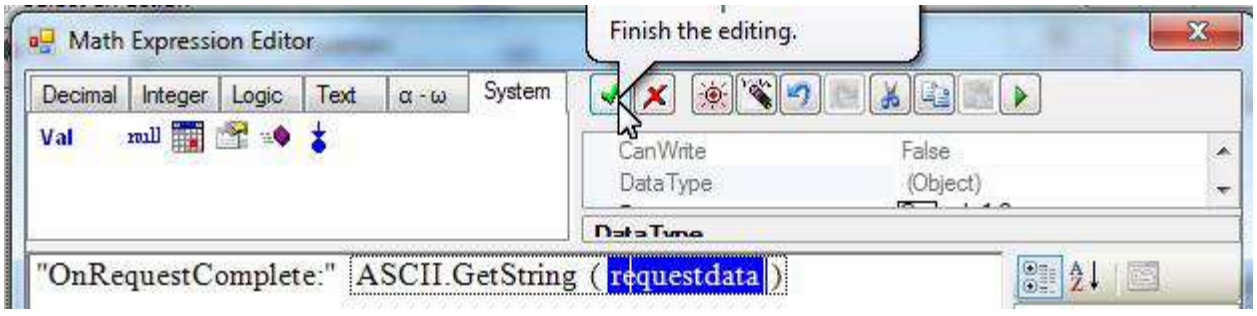

Object picker
Back
Next
Exit
Add
System.Security.Principal
System.Text
ASCIIEncoding
Decoder
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception

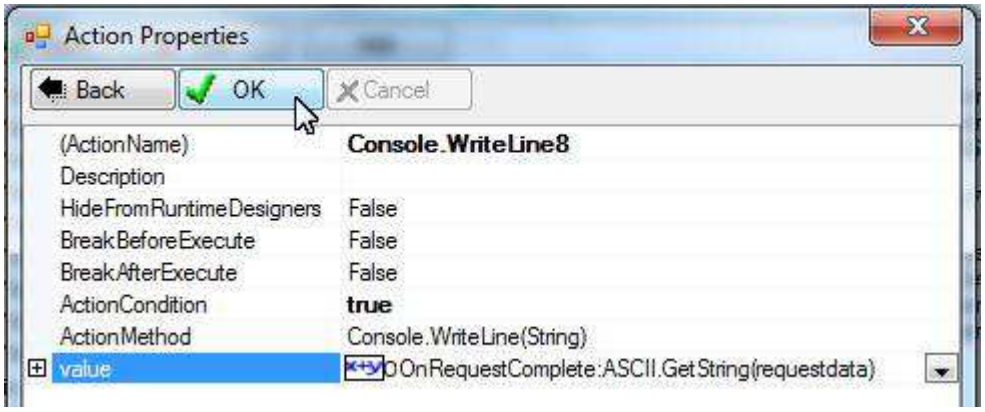

Object picker
Back
Next
Exit
Add
EncoderReplacementFallbackBuffer
Encoding
Static properties
ASCII:Encoding
Properties inherited
Fields inherited
Methods inherited
Clone()Object
Equals(Object)Boolean
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception
Console.WriteLine7
OnPokeComplete
DdeClient1 of XClass`1

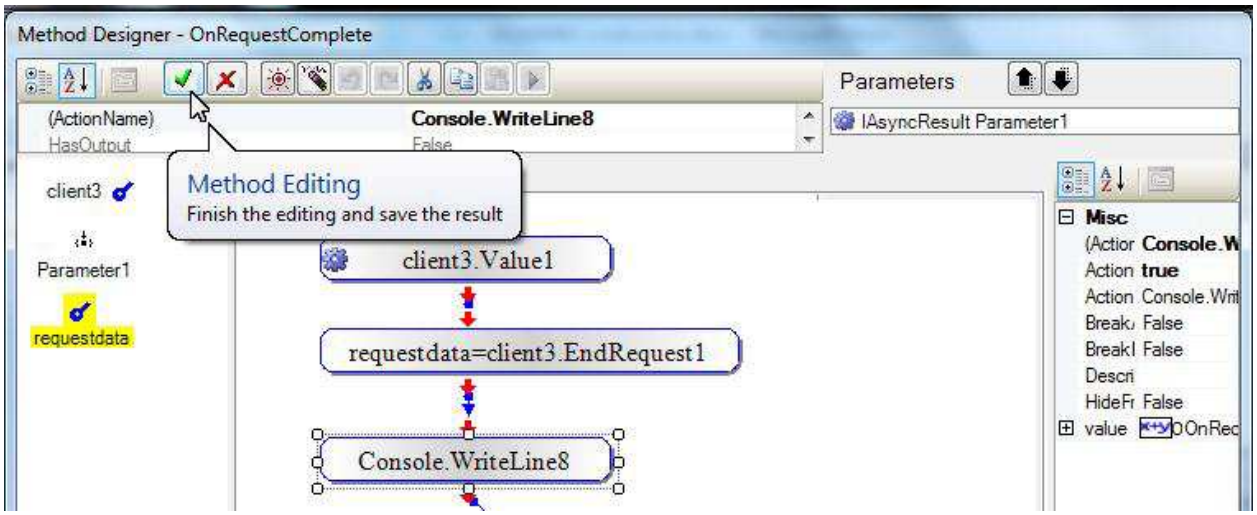

Object picker
Back
Next
Exit
Add
GetPreamble()Byte[]
GetString(Byte[])String
GetString(Byte[],Int32,Int32)String
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception


Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
Misc
Data Type
(Double)
A property
"OnRequestComplete:" ASCII.GetString (0)

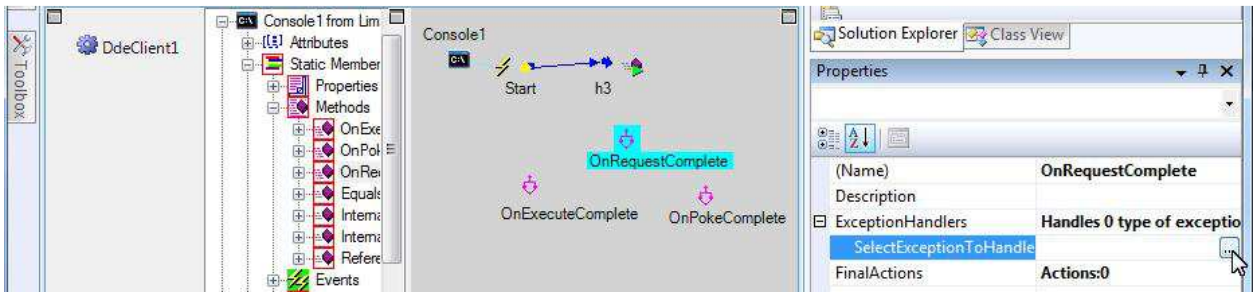

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
Method1(IAsyncResult)
Actions
Parameter1
client3
requestdata
Action Input
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception
Console.WriteLine7
OnPokeComplete
DdeClient1 of XClass`1
ASCII:Encoding.GetBytes(String)Byte
IAsyncResult Parameter1.AsyncState

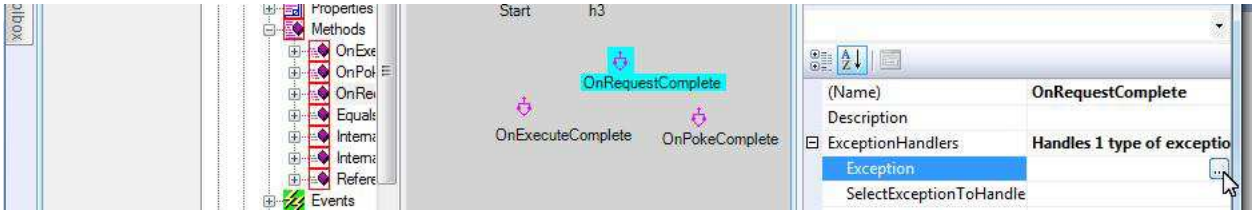

Finish the editing.
Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
CanWrite
False
Data Type
(Object)
"OnRequestComplete:" ASCII.GetString (requestdata)

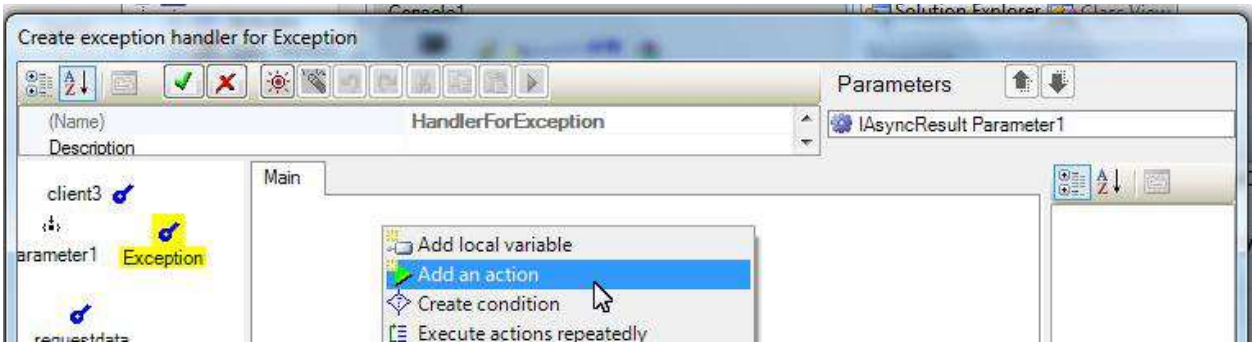


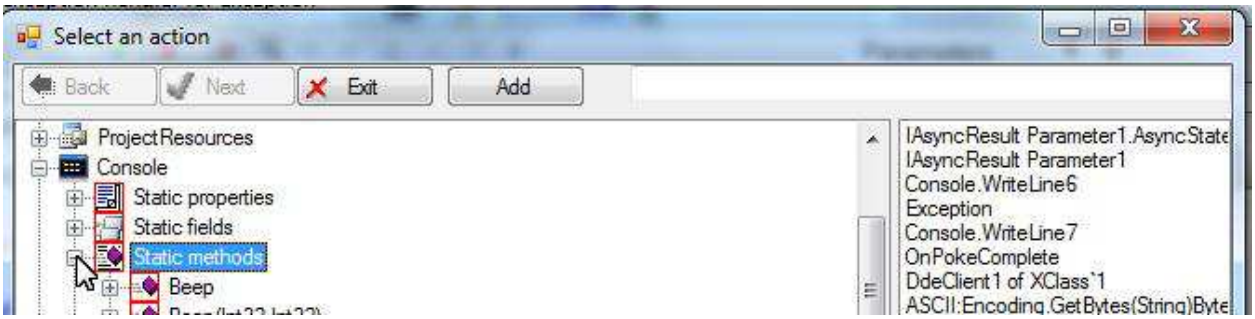


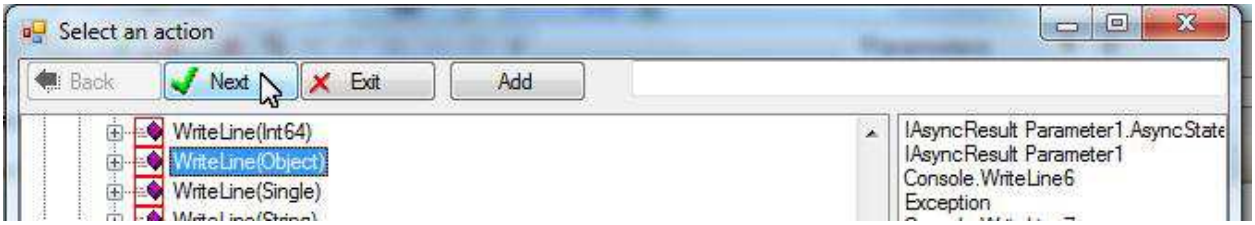
Asynchronous Request Operation – Handle exception

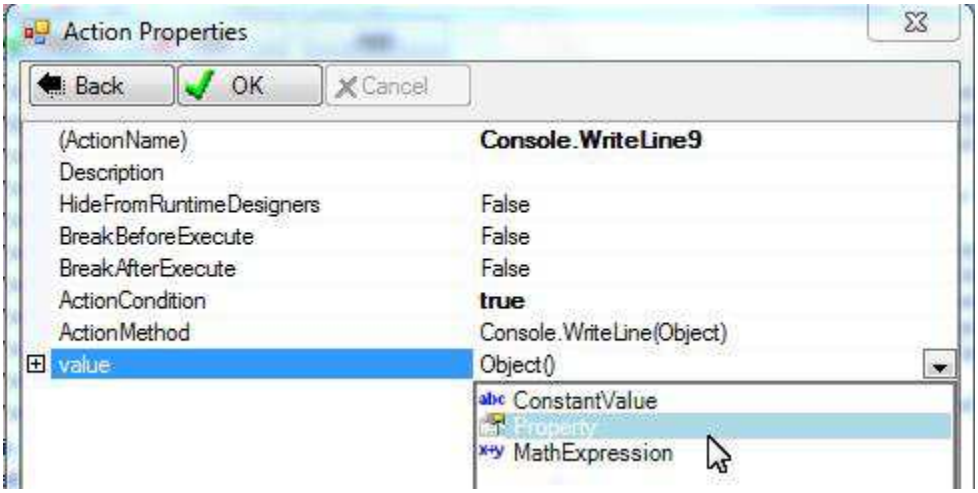


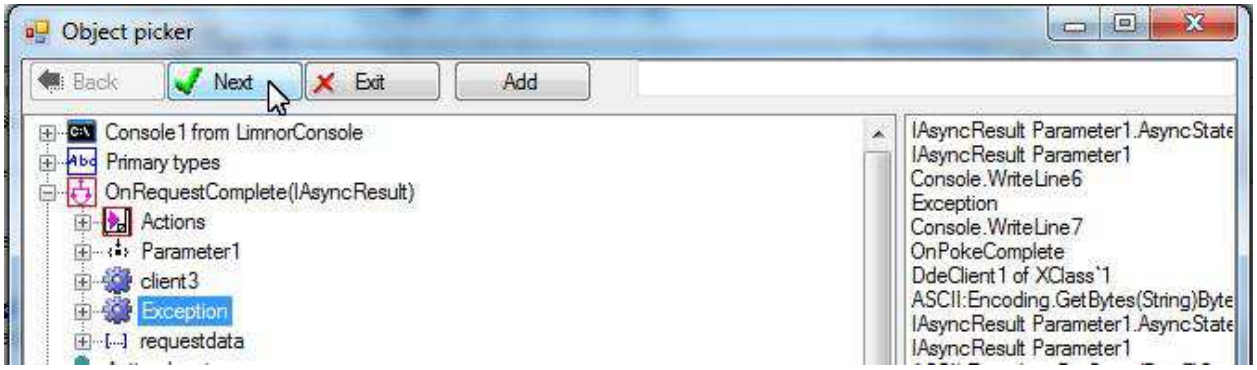

Properties
Methods
Events
Start
h3
OnRequestComplete
OnExecuteComplete
OnPokeComplete
(Name)
OnRequestComplete
Description
ExceptionHandlers
Handles 1 type of exceptio
Exception
SelectExceptionToHandle

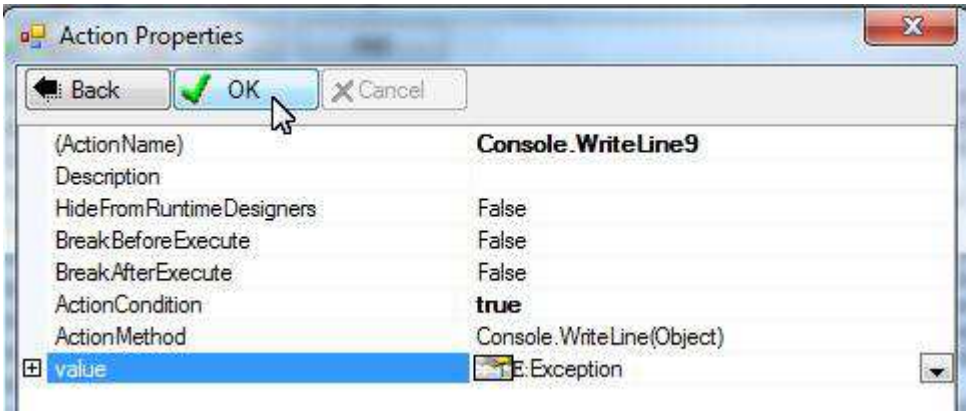

Create exception handler for Exception
Parameters
(Name)
HandlerForException
Description
IAsyncResult Parameter1
client3
Exception
Main
Add local variable
Add an action
Create condition
Execute actions repeatedly

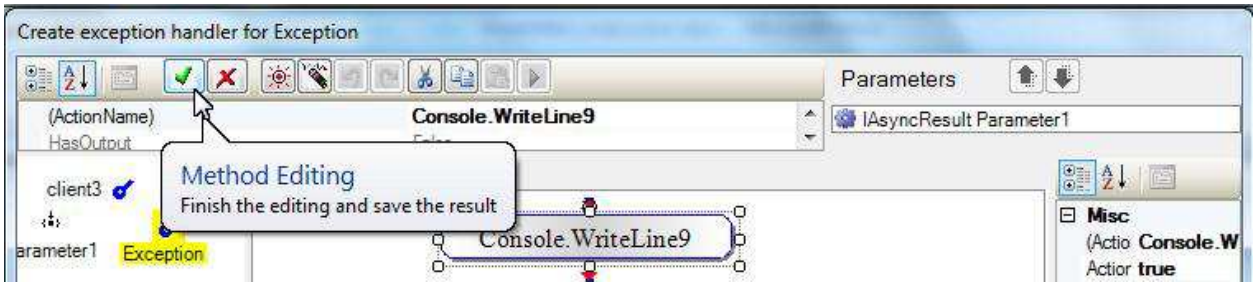

Select an action
Back
Next
Exit
Add
ProjectResources
Console
Static properties
Static fields
Static methods
Beep
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception
Console.WriteLine7
OnPokeComplete
DdeClient1 of XClass`1


Select an action
Back
Next
Exit
Add
WriteLine(Int64)
WriteLine(Object)
WriteLine(Single)
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception


Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine9
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(Object)
value
Object()
ConstantValue
Property
MathExpression

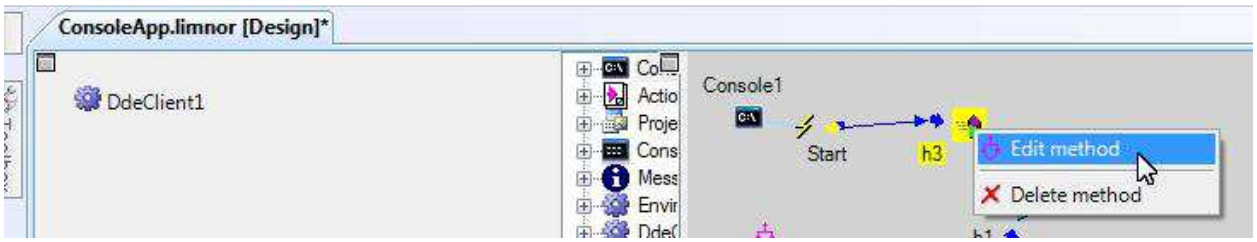


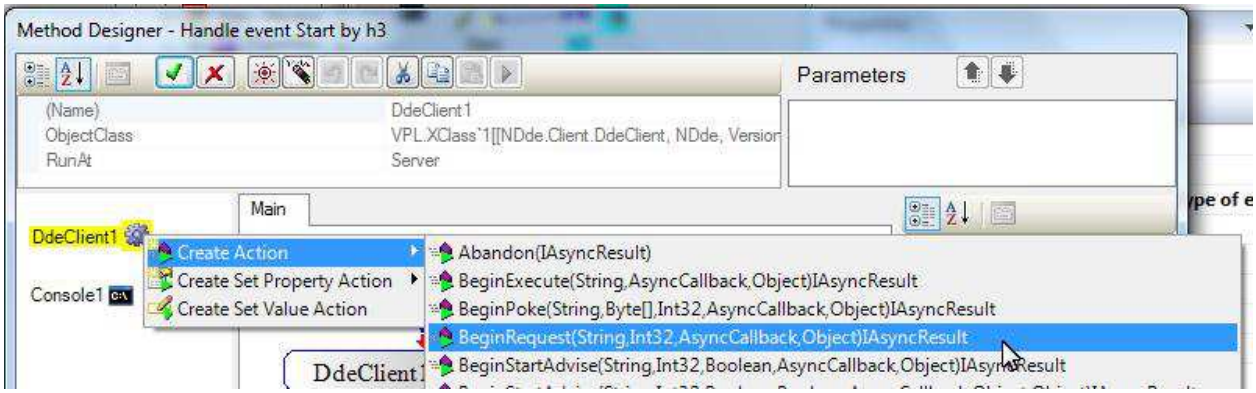




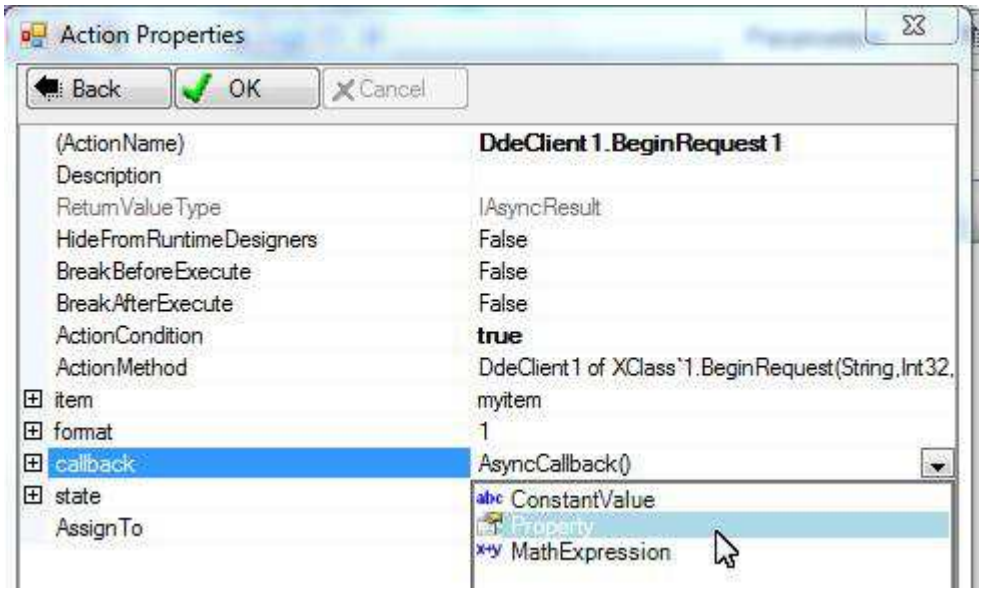
Asynchronous Request Operation – Start Request operation

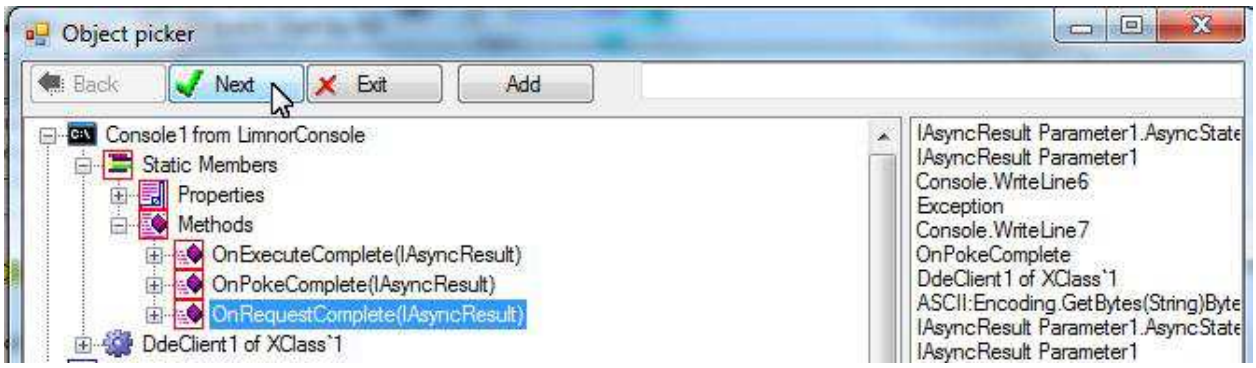
Now we may create an asynchronous request action.

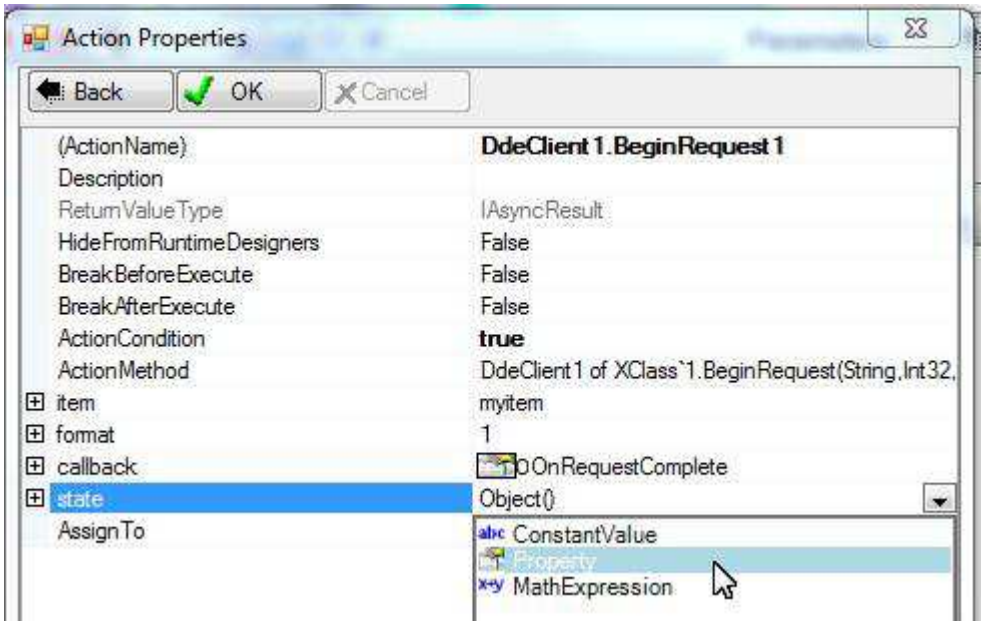


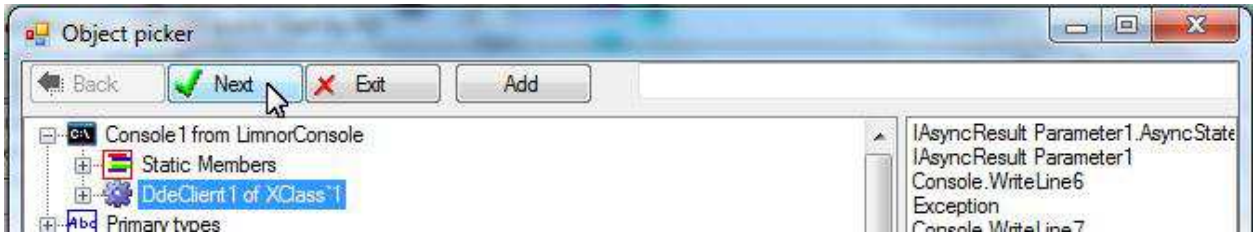


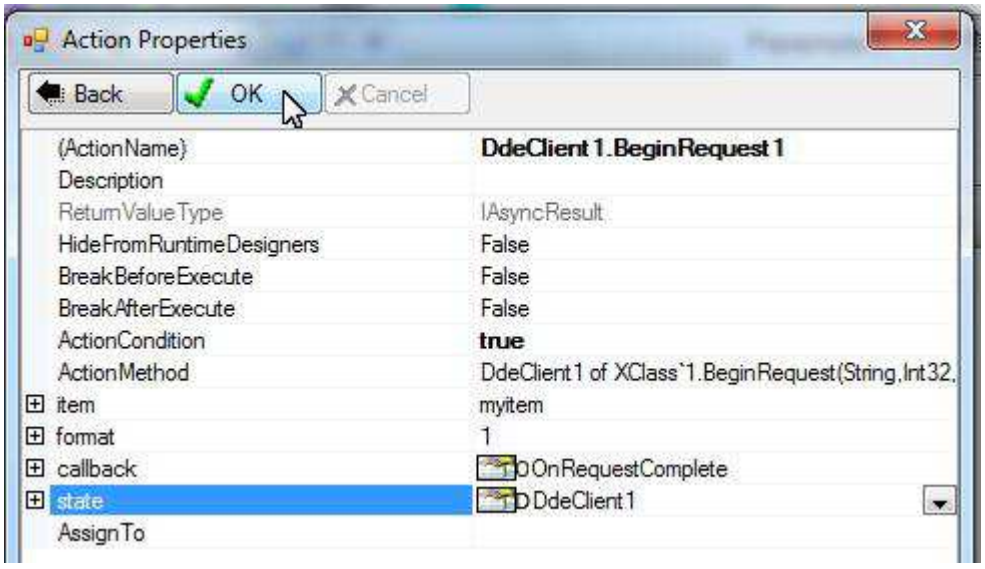
Provide callback function:



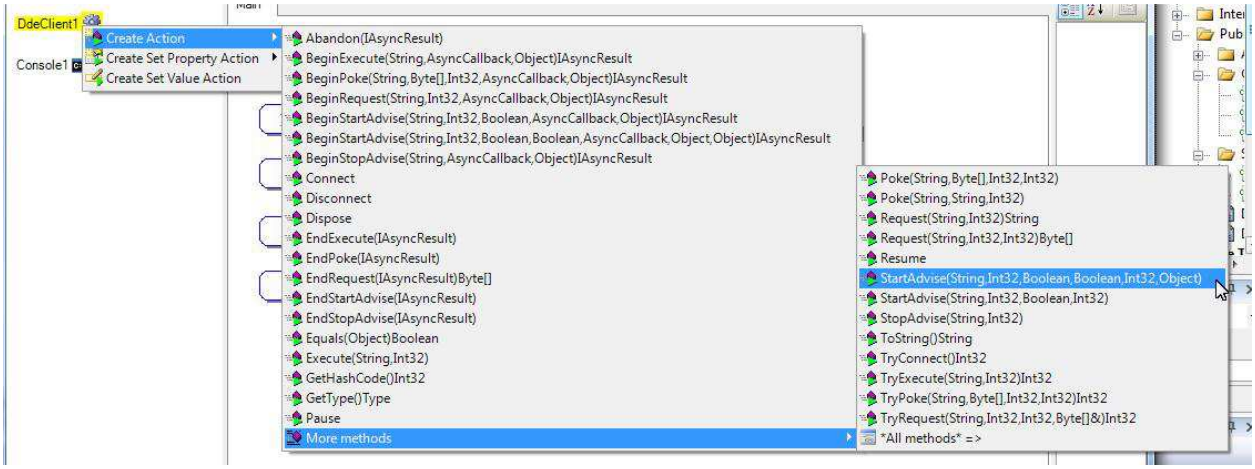


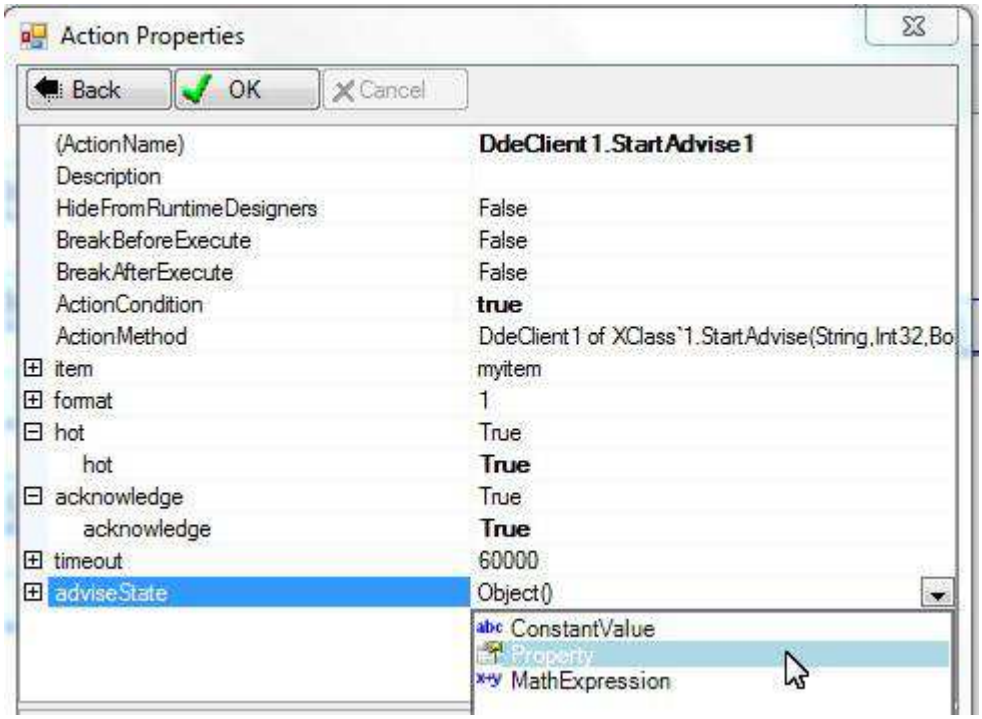

Action Properties
Back
OK
Cancel
(ActionName)
DdeClient1.BeginRequest1
Description
ReturnValueType
IAsyncResult
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
DdeClient1 of XClass`1.BeginRequest(String,Int32,
item
myitem
format
1
callback
OnRequestComplete
state
Object()
AssignTo
ConstantValue
Property
MathExpression

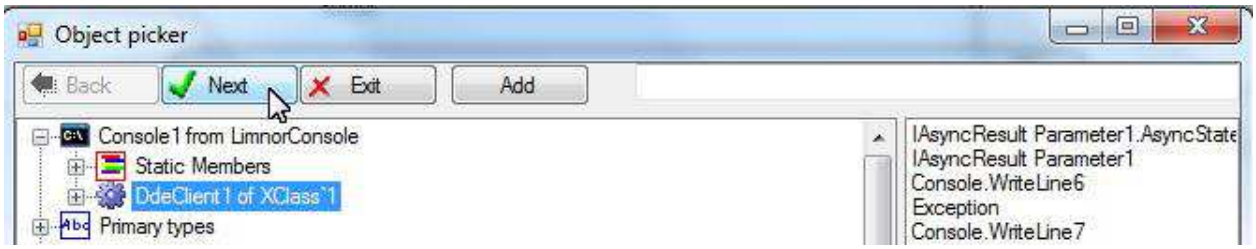

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Static Members
DdeClient1 of XClass`1
Primary types
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception

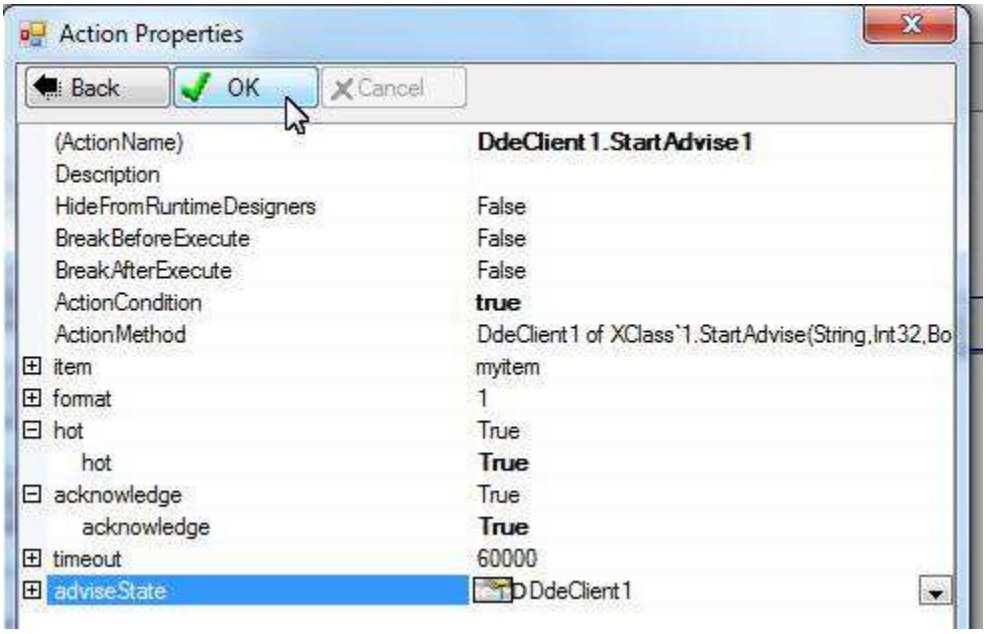

Action Properties
Back
OK
Cancel
(ActionName)
DdeClient1.BeginRequest1
Description
ReturnValueType
IAsyncResult
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
DdeClient1 of XClass`1.BeginRequest(String,Int32,
item
myitem
format
1
callback
OnRequestComplete
state
DdeClient1
AssignTo

Advise Loop


DdeClient1
Console1
Create Action
Create Set Property Action
Create Set Value Action
Abandon(IAsyncResult)
BeginExecute(String,AsyncCallback,Object)IAsyncResult
BeginPoke(String,Byte[],Int32,AsyncCallback,Object)IAsyncResult
BeginRequest(String,Int32,AsyncCallback,Object)IAsyncResult
BeginStartAdvise(String,Int32,Boolean,AsyncCallback,Object)IAsyncResult
BeginStartAdvise(String,Int32,Boolean,Boolean,AsyncCallback,Object,Object)IAsyncResult
BeginStopAdvise(String,AsyncCallback,Object)IAsyncResult
Connect
Disconnect
Dispose
EndExecute(IAsyncResult)
EndPoke(IAsyncResult)
EndRequest(IAsyncResult)Byte[]
EndStartAdvise(IAsyncResult)
EndStopAdvise(IAsyncResult)
Equals(Object)Boolean
Execute(String,Int32)
GetHashCode()Int32
GetType()Type
Pause
More methods
Poke(String,Byte[],Int32,Int32)
Poke(String,String,Int32)
Request(String,Int32)String
Request(String,Int32,Int32)Byte[]
Resume
StartAdvise(String,Int32,Boolean,Boolean,Int32,Object)
StartAdvise(String,Int32,Boolean,Int32)
StopAdvise(String,Int32)
ToString()String
TryConnect()Int32
TryExecute(String,Int32)Int32
TryPoke(String,Byte[],Int32,Int32)Int32
TryRequest(String,Int32,Int32,Byte[]&)Int32
All methods =>

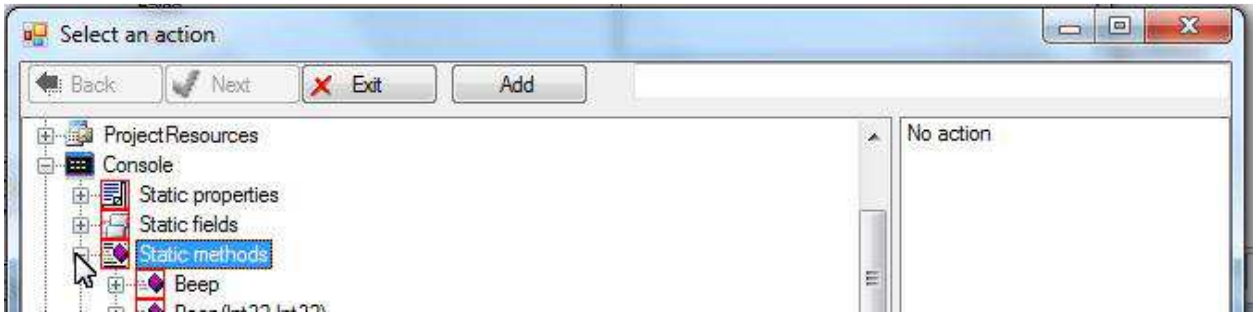

Action Properties
Back
OK
Cancel
(ActionName)
DdeClient1.StartAdvise1
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
DdeClient1 of XClass`1.StartAdvise(String,Int32,Bo
item
myitem
format
1
hot
True
hot
True
acknowledge
True
acknowledge
True
timeout
60000
adviseState
Object()
ConstantValue
Property
MathExpression

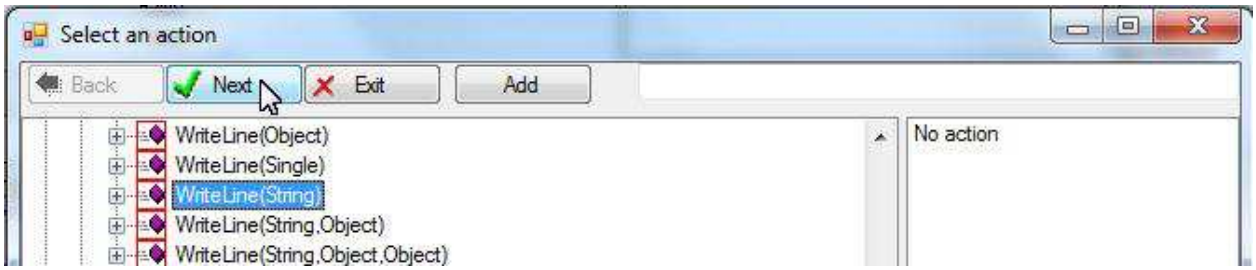

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Static Members
DdeClient1 of XClass`1
Primary types
IAsyncResult Parameter1.AsyncState
IAsyncResult Parameter1
Console.WriteLine6
Exception
Console.WriteLine7

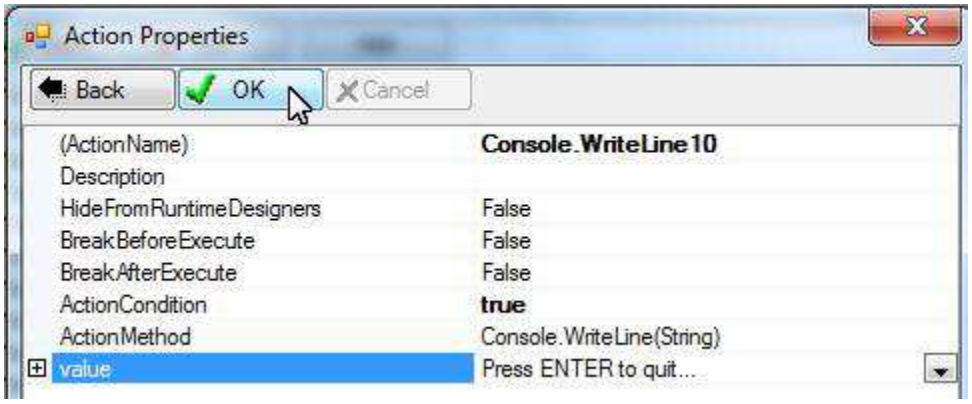


Console Wait

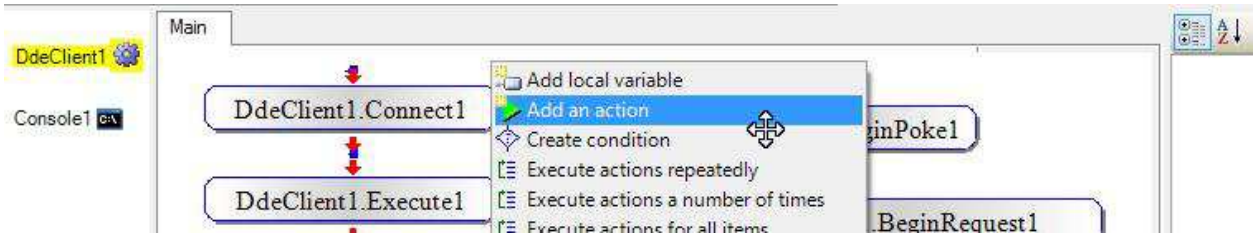
Show a prompt on the console:

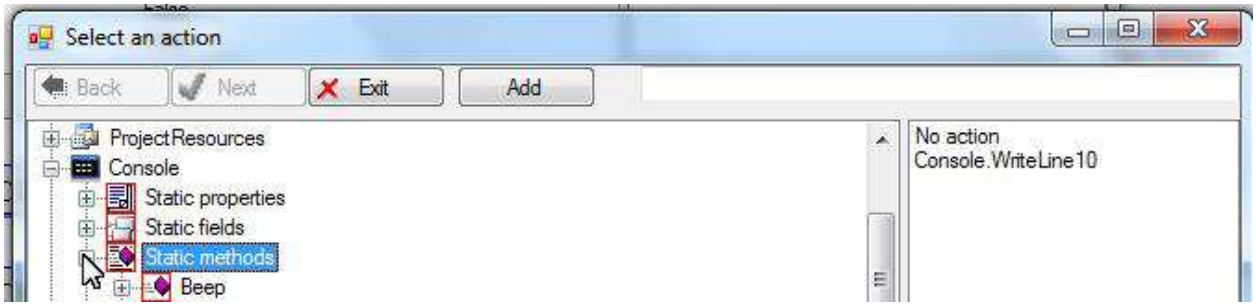


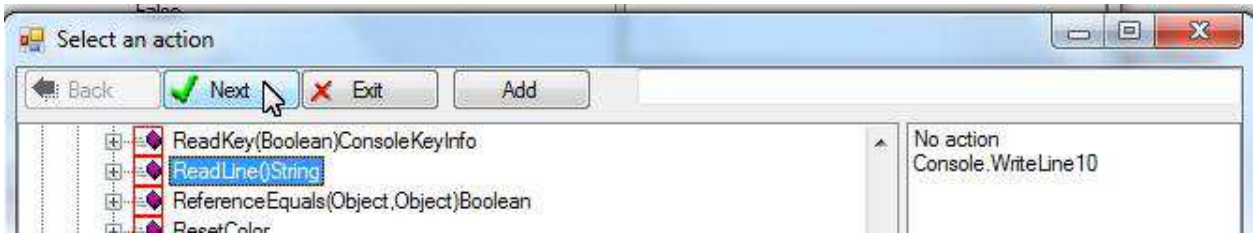


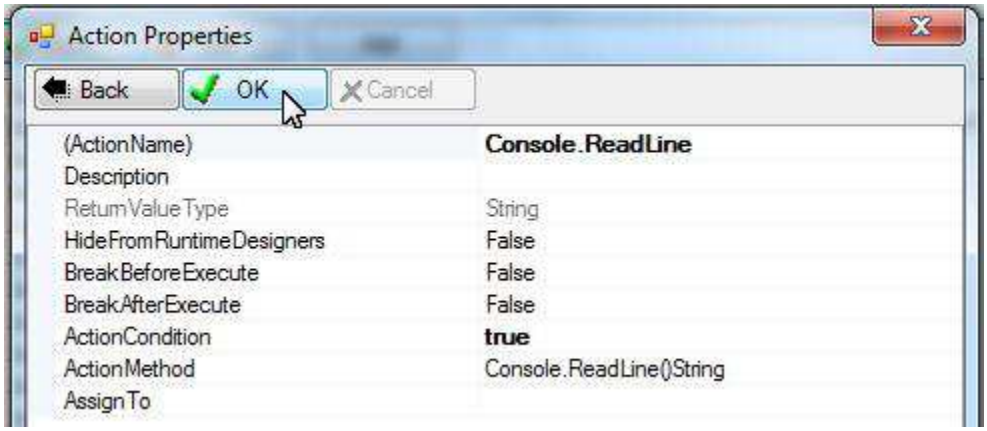


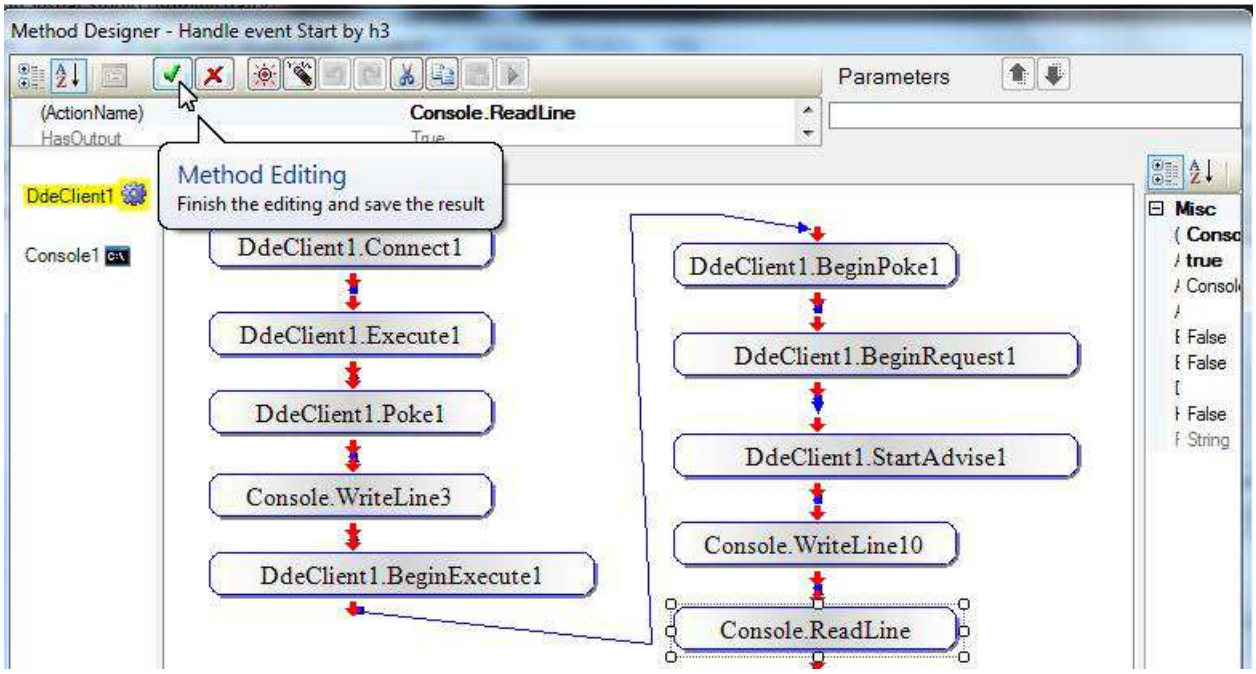
Wait for user pressing ENTER





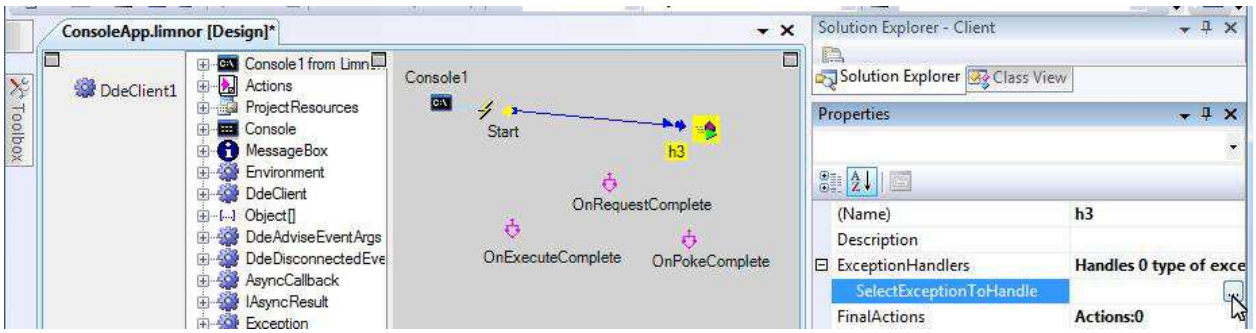






Handle Exceptions

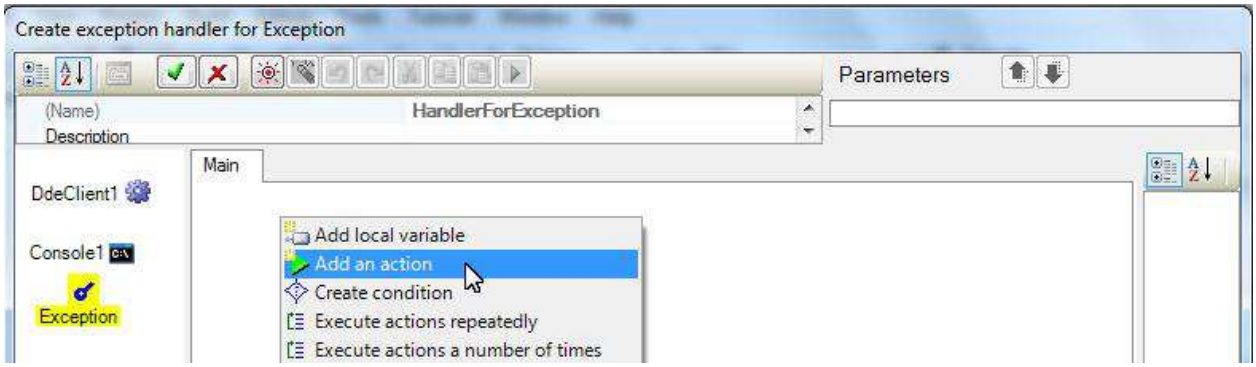
Select the Start event handler to add exception handling to it:

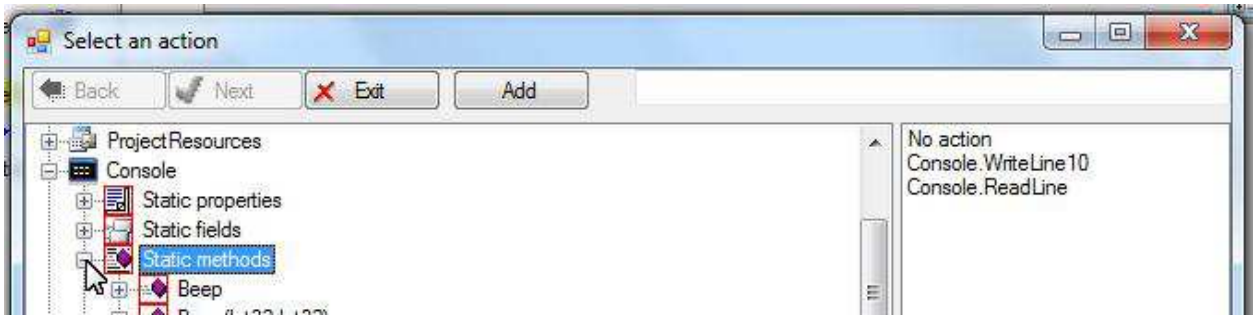


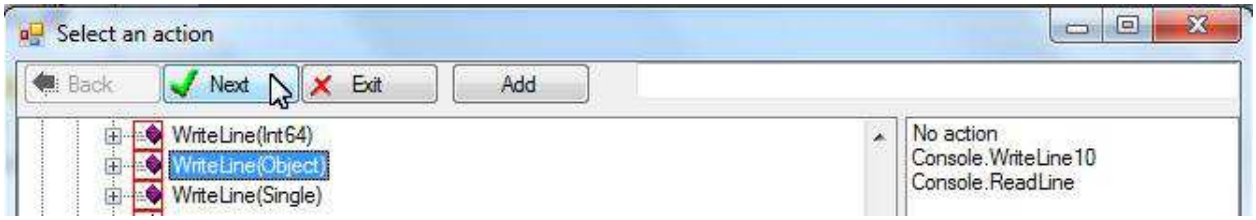
Select Exception class as the exception to be handled. The Exception class represents all kinds of the exceptions.

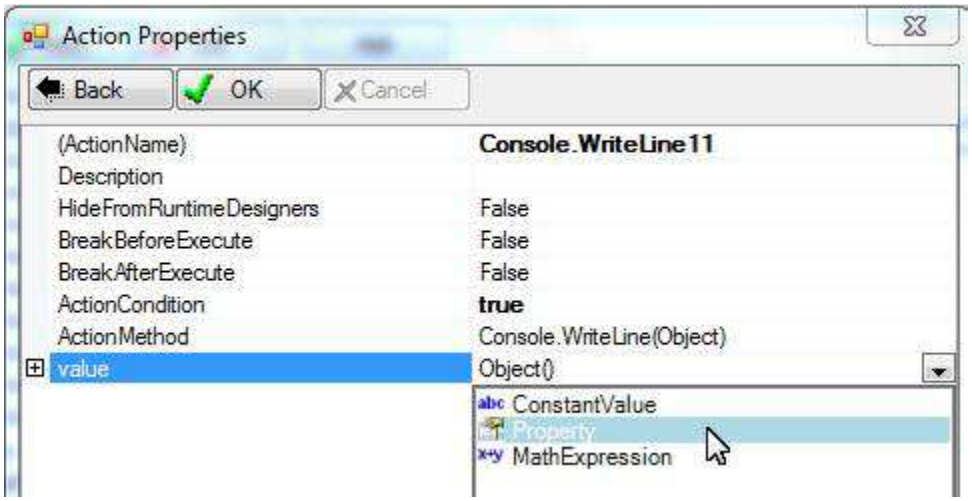
Add actions to handle exceptions:

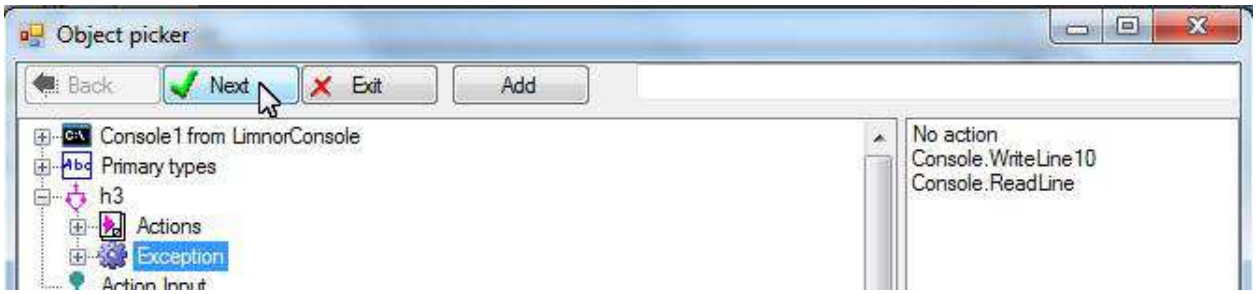
For this sample, we display the exception on the console:

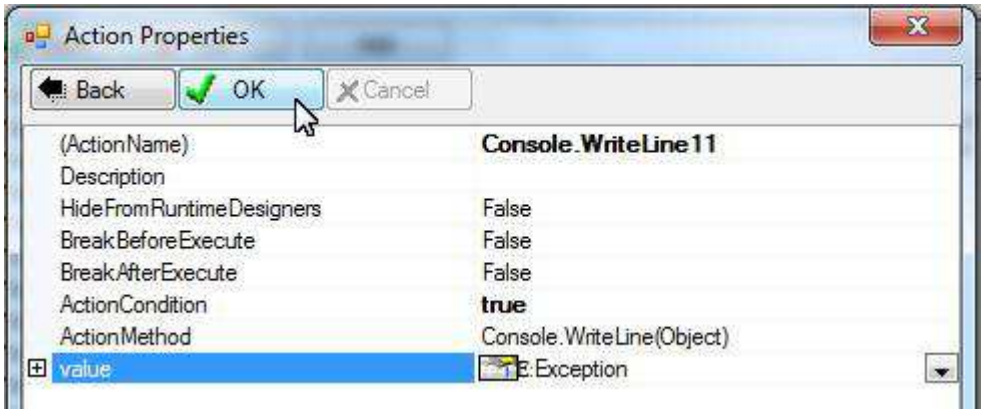



Select an action
Back
Next
Exit
Add
Project Resources
Console
Static properties
Static fields
Static methods
Beep
No action
Console.WriteLine10
Console.ReadLine

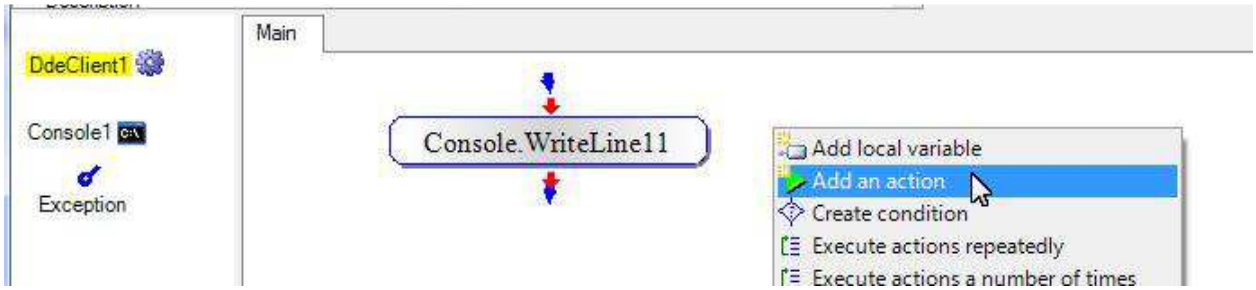

Select an action
Back
Next
Exit
Add
WriteLine(Int64)
WriteLine(Object)
WriteLine(Single)
No action
Console.WriteLine10
Console.ReadLine

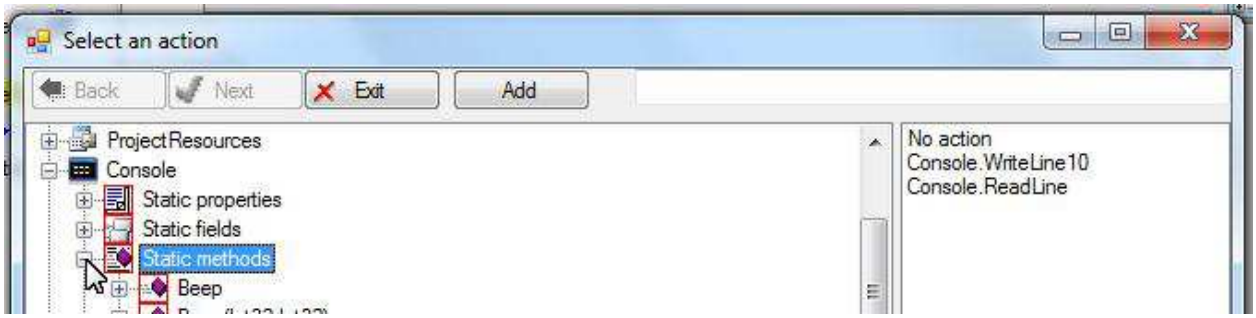

Action Properties
Back
OK
Cancel
(ActionName)
Console.WriteLine11
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
Console.WriteLine(Object)
value
Object()
ConstantValue
Property
MathExpression

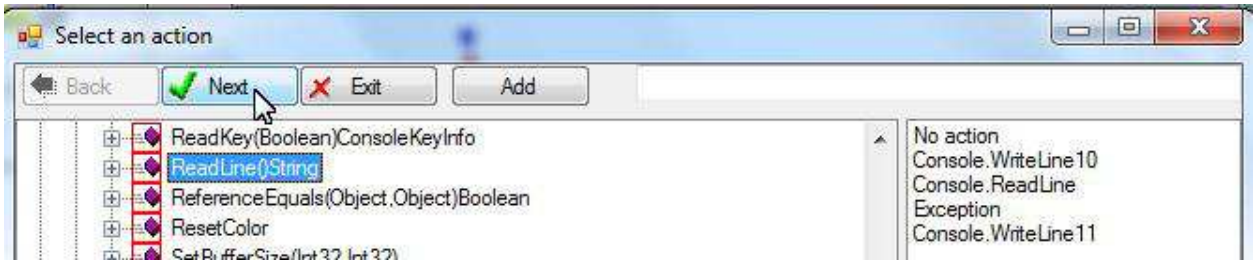

Object picker
Back
Next
Exit
Add
Console1 from LimnorConsole
Primary types
h3
Actions
Exception
No action
Console.WriteLine10
Console.ReadLine

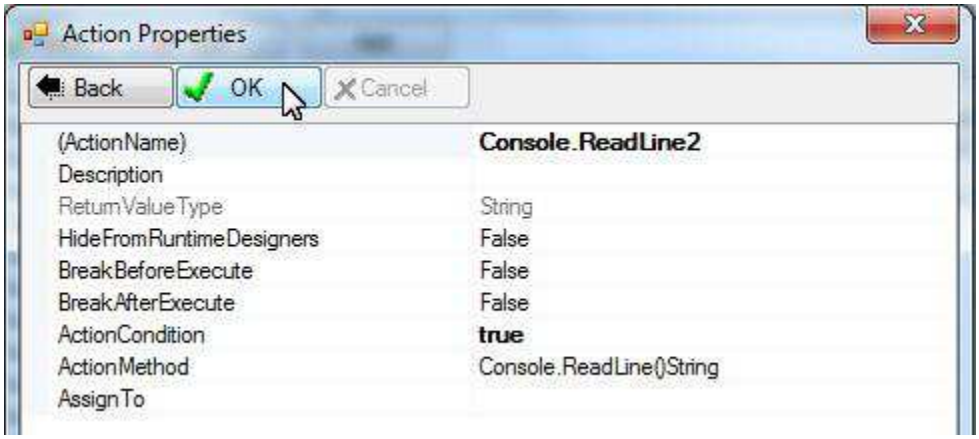


Wait for the user to press ENTER after showing the exception:

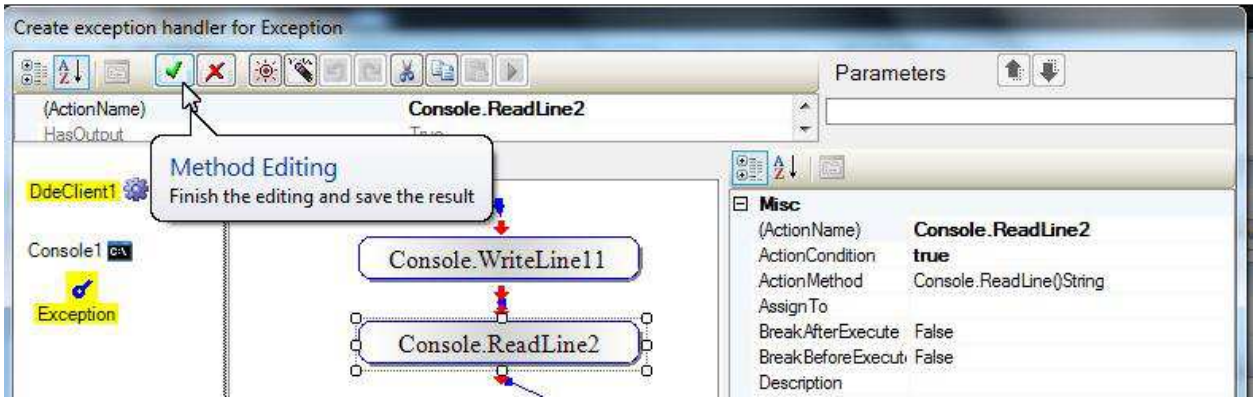






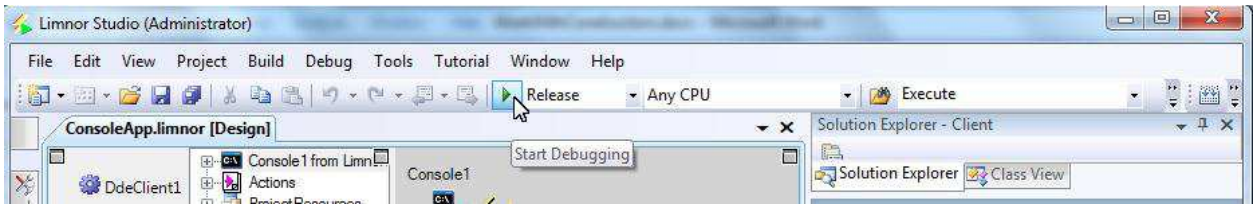


This is the sample exception handling:

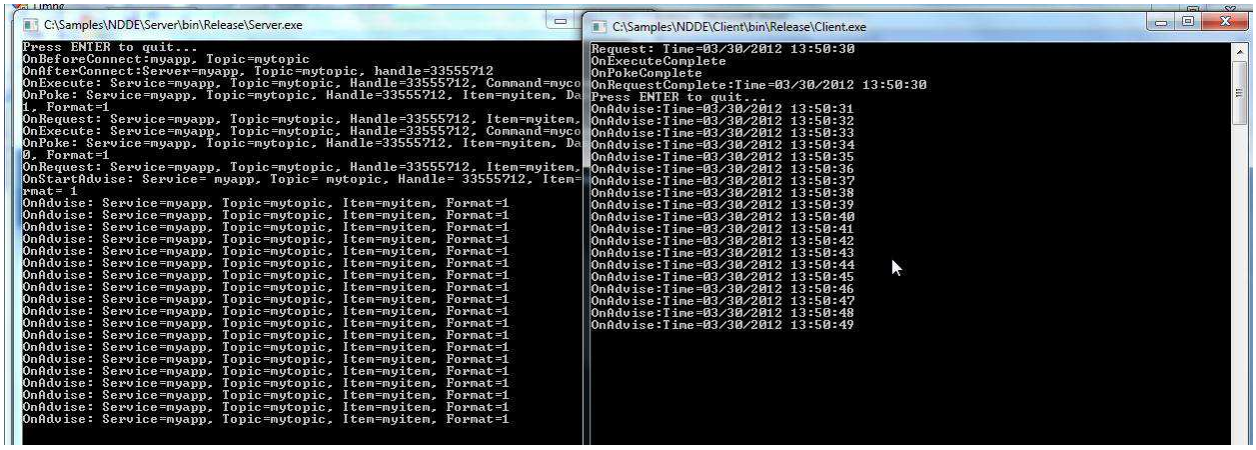


Start DDE Client

Before starting the DDE client console, be sure that the DDE server sample console is running.



We can see the communications between the DDE server and the DDE client:



Feedback

Please send your feedback to support@limnor.com