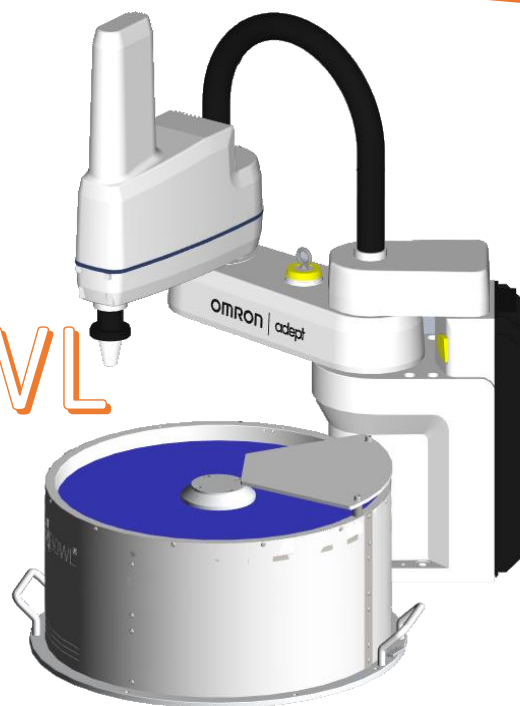


# OMRON FLEXIBOWL PLUGIN



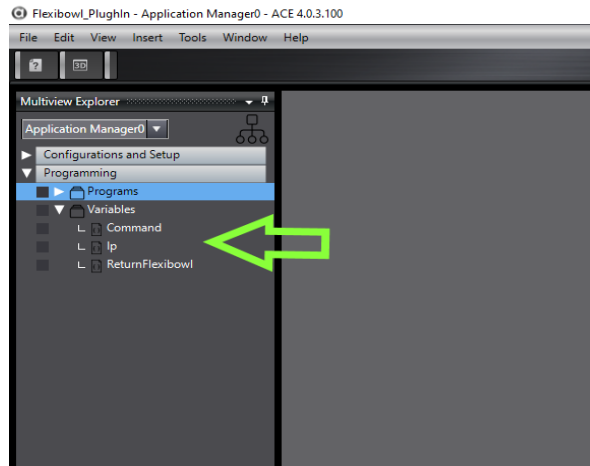
Questo Plugin è nato con l'idea di comunicare in maniera rapida e sicura con il flexibowl tramite i robot Omron, mediante il software Omron Ace versione 4 o versione 3. Il Plugin non necessita di licenze aggiuntive Omron.

# FlexiBowl®



(c) 2017-2019 Omron Robotics and Safety Technologies, Inc. All rights reserved.

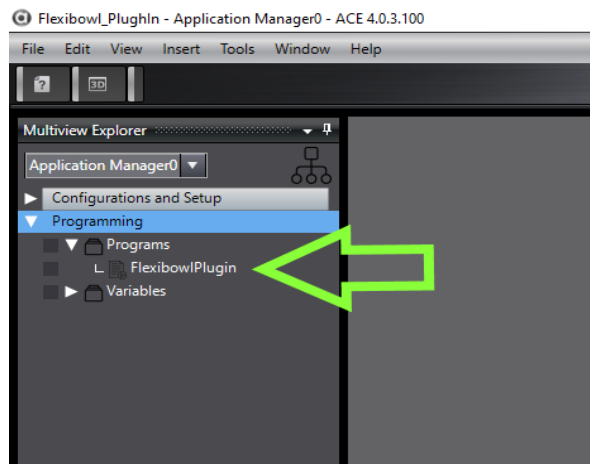
STEP 1:



Nel *Multiview Explorer*, selezionando l'*Application Manager* desiderata, è possibile aggiungere sotto l'albero *Variables*, mediante il tasto destro del mouse, tre *string variable* con nome:

- **Command**
- **Ip**
- **ReturnFlexibowl**

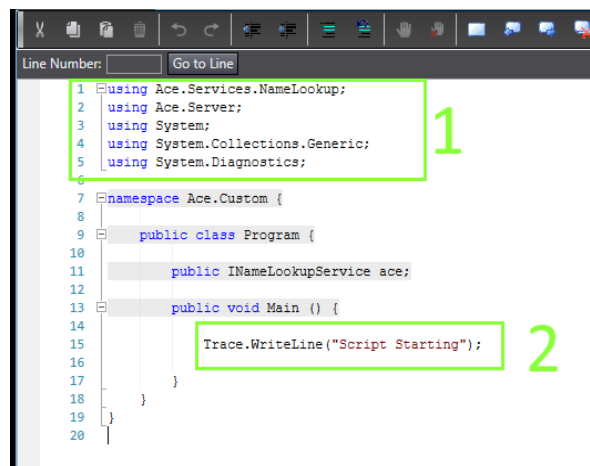
STEP 2:



Sempre avendo selezionato l'*Application Manager* desiderata, è possibile aggiungere sotto l'albero *Programs*, mediante il tasto destro del mouse, un nuovo task c#.

Tale task potrà essere richiamato da V+ per la movimentazione del flexibowl. Rinominare il task come "Flexibowl Plugin".

STEP 3:



Facendo doppio click sul programma appena creato sarà possibile editarlo. Nella zona 1 vi sono la dichiarazioni da usare, mentre nella parte 2 vi è il corpo dello script.

Ora andremo a modificare tale script.

# ACE versione 4



## STEP 4:

```
Line Number:  Go to Line
1 using Ace.Services.NameLookup;
2 using Ace.Server;
3 using System;
4 using System.Collections.Generic;
5 using System.Diagnostics;
6
7 using System.Net;
8 using System.Net.Sockets;
9 using System.Text;
10 using Ace.Server.Core.Scripting;
11 using Ace.Server.Core.Variable;
12
13 namespace Ace.Custom {
14
15     public class Program {
16
17         public INameLookupService ace;
18
19         public void Main() {
20
21
```

Aggiungere nella sezione 1 del codice le seguenti dipendenze:

```
using System.Net;
using System.Net.Sockets;
using System.Text;
using Ace.Server.Core.Scripting;
using Ace.Server.Core.Variable;
```

## STEP 5:

Nella sezione di codice 2 invece, cancellare e sostituire con tutto il codice che si trova a pagina seguente, nell'immagine sottostante diamo un'anteprima del risultato finale.

```
--
13 namespace Ace.Custom {
14
15     public class Program {
16
17         public INameLookupService ace;
18
19         public void Main() {
20
21             Trace.WriteLine("Flexibowl PlugIn " +DateTime.Now.ToString()+" Run");
22
23             string receiveString = "";
24             int byteSent = 0;
25
26             UdpClient m_udpClient= new UdpClient(7777);
27             //////////////////////////////////////////////////
28             /// To change/////////////////////////////////
29             IVariableString Command = (IVariableString) ace["/Application Manager0/Variables/Command"];
30             IVariableString Ip = (IVariableString) ace["/Application Manager0/Variables/Ip"];
31             IVariableString ReturnFlexibowl = (IVariableString) ace["/Application Manager0/Variables/ReturnFlexibowl"];
32             //////////////////////////////////////////////////
33             ReturnFlexibowl.CurrentValue="False";
34             IPEndPoint ep = new IPEndPoint(IPAddress.Parse(Ip.CurrentValue), 7775);
35             try {
36                 m_udpClient.Connect(ep);
37                 m_udpClient.Client.SendTimeout = 500;
38                 m_udpClient.Client.ReceiveTimeout = 500;
39             }
40             catch (ArgumentNullException ex) // dgram is null.
41             {
42                 Trace.WriteLine(ex.ToString());
43             }
44             string dataToSend = Command.CurrentValue.ToUpper();
45
46             try {
47
48                 Byte[] SCLstring = Encoding.ASCII.GetBytes(dataToSend);
```

```
Trace.WriteLine("Flexibowl PlugIn " + DateTime.Now.ToString() + " Run");

string receiveString = "";
int byteSent = 0;

UdpClient m_udpClient= new UdpClient(7777);
////////////////////
// To change////////////////
IVariableString Command = (IVariableString) ace["/Application Manager0/Variables/Command"];
IVariableString Ip = (IVariableString) ace["/Application Manager0/Variables/Ip"];
IVariableString ReturnFlexibowl = (IVariableString) ace["/Application Manager0/Variables/ReturnFlexibowl"];
////////////////////
ReturnFlexibowl.CurrentValue="False";
IPEndPoint ep = new IPEndPoint(IPAddress.Parse(Ip.CurrentValue), 7775);

try {
    m_udpClient.Connect(ep);
    m_udpClient.Client.SendTimeout = 500;
    m_udpClient.Client.ReceiveTimeout = 500;
}
catch (ArgumentNullException ex)
{
    Trace.WriteLine(ex.ToString());
}

string dataToSend = Command.CurrentValue.ToUpper();

try {
    Byte[] SCLstring = Encoding.ASCII.GetBytes(dataToSend);
    Byte[] sendBytes = new Byte[SCLstring.Length + 3];
    sendBytes[0] = 0;
    sendBytes[1] = 7;
    System.Array.Copy(SCLstring, 0, sendBytes, 2, SCLstring.Length);
    sendBytes[sendBytes.Length - 1] = 13; // CR
    byteSent = m_udpClient.Send(sendBytes, sendBytes.Length);
    var receivedData = m_udpClient.Receive(ref ep);
    receiveString = Encoding.ASCII.GetString(receivedData);
    if ((receiveString.Contains("%") && (dataToSend.Contains("Q")))) {
        bool moving = true;
        while (moving == true) {
            SCLstring = Encoding.ASCII.GetBytes("RS");
            sendBytes = new Byte[SCLstring.Length + 3];
            sendBytes[0] = 0;
            sendBytes[1] = 7;
            System.Array.Copy(SCLstring, 0, sendBytes, 2, SCLstring.Length);
            sendBytes[sendBytes.Length - 1] = 13; // CR
            byteSent = m_udpClient.Send(sendBytes, sendBytes.Length);
            receivedData = m_udpClient.Receive(ref ep);
            receiveString = Encoding.ASCII.GetString(receivedData);
            if (receiveString.Contains("F"))
                moving = true;
            else
                moving = false;
            System.Threading.Thread.Sleep(20);
        }
        ReturnFlexibowl.CurrentValue = "Done";
    }
    else {
        SCLstring = new Byte[receivedData.Length - 3];
        System.Array.Copy(receivedData, 2, SCLstring, 0, SCLstring.Length);
        receiveString = Encoding.ASCII.GetString(SCLstring);
        ReturnFlexibowl.CurrentValue = receiveString;
    }
}
m_udpClient.Dispose();
}
catch (ArgumentNullException ex)
{
    Trace.WriteLine(ex.ToString());
}
```

## STEP 6:

Una volta copiato e incollato il codice, bisogna verificare che i percorsi delle variabili create in precedenza siano corretti.

Per verificare ciò, bisogna controllare nel riquadro evidenziato nell'immagine, i percorsi delle tre variabili siano corretti.

Selezionare una delle tre variabili prima create, drag&drop nella pagina del codice.

In questo caso avete creato un riferimento alla vostra variabile, controllate il percorso corretto ed eliminate la riga creata.

Verificare che i percorsi delle tre variabili nel codice siano corretti.

### Esempio:

Originale

```
IVariableString Command = (IVariableString) ace["/Application Manager0/Variables/Command"];
```

Modificata

```
IVariableString Command = (IVariableString) ace["/Application Manager4/Variables/Command"];
```

The screenshot shows the Visual Studio IDE with the following details:

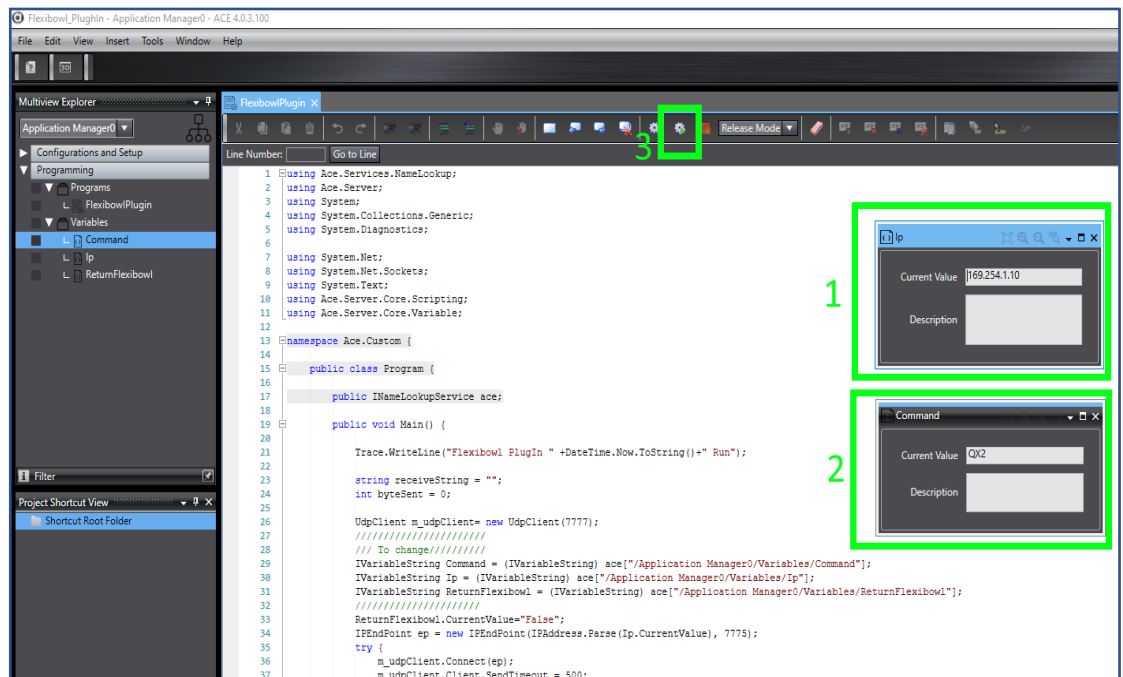
- Title Bar:** Flexibowl\_PluginIn - Application Manager0 - ACE 4.0.3.100
- Multiview Explorer (Left):** Shows a tree view with folders: Application Manager0, Configurations and Setup, Programming, Programs, FlexibowlPlugin, Variables, Command, Ip (highlighted with a blue arrow), and ReturnFlexibowl.
- Code Editor (Right):** Shows the source code for 'FlexibowlPlugin'. Lines 29-31 are highlighted in cyan:
 

```

29 // To change//////////
30 IVariableString Command = (IVariableString) ace["/Application Manager0/Variables/Command"];
31 IVariableString Ip = (IVariableString) ace["/Application Manager0/Variables/Ip"];
32 IVariableString ReturnFlexibowl = (IVariableString) ace["/Application Manager0/Variables/ReturnFlexibowl"];
      
```

## STEP 7:

Arrivati qui, è possibile testare il movimento del Flexibowl.  
Impostando l'Ip nella variabile IP (rif. 1), ed il comando da eseguire nella variabile Command (rif. 2), cliccando sul pulsante Run (rif. 3) verrà inviato il comando al Flexibowl con l'Ip impostato.



## STEP 8:

Ora vedremo come impostare le variabili e eseguire lo script da V+  
Creiamo un programma V+ con il codice della pagina successiva.  
Ricopiando il codice verificare che i percorsi delle variabili siano corretti, tipo:

*\$object = "/Application Manager0/Variables/Ip"*

Andando a settare l'ip e il comando, eseguendo lo script V+ il flexibowl effettuerà il comando

```
;insert the data
;////////////////////////////////////
$ip="169.254.1.10"
$command="QX3"
;////////////////////////////////////
```

*Al momento le variabili Ip,Command e return.flexibow in V+ sono locali (AUTO), per settarle da programmi esterni rendere queste variabili Globali, quindi non Auto. Eseguendo lo script V+ verrà eseguito lo script C# che azionerà il flexibowl*

```

.PROGRAM flbplugin()
  AUTO $object, $variable, $ip, $command, $return.flexibow, $method, $args[0]
  AUTO REAL status, is.alive
  ;insert the data
  ;////////////////////
  $ip="169.254.1.10"
  $command="QX3"
  ;////////////////////
  ;Set the data on c#
  ;IP
  $object = "/Application Manager0/Variables/Ip"
  $variable = "CurrentValue"
  CALL rm.write.str($object, $variable, 1, $ip, status)
  IF (status < 0) THEN
    TYPE "Unable To Write Value: ", status
    PAUSE
  END
  ;COMMAND
  $object = "/Application Manager0/Variables/Command"
  $variable = "CurrentValue"
  CALL rm.write.str($object, $variable, 1, $command, status)
  IF (status < 0) THEN
    TYPE "Unable To Write Value: ", status
    PAUSE
  END
  ;Execute the c#
  CALL rm.chk.server(is.alive)
  IF (is.alive == FALSE) THEN
    TYPE "Not Communicating"
    PAUSE
  END ; Execute a script on the server and wait for 3 seconds for it to complete
  $object = "/Application Manager0/Programs/FlexibowIPlugin"
  $method = "Execute"
  CALL rm.execute($object, $method, 0, $args[], 5, status)
  IF (status < 0) THEN
    TYPE "Problem executing script: ", status
    PAUSE
  END
  ;Read the Answer
  $object = "/Application Manager0/Variables/ReturnFlexibow1"
  $variable = "CurrentValue"
  ;Read the answer from flexibow1
  CALL rm.read.str($object, $variable, 1, $return.flexibow, status)
  IF (status < 0) THEN
    TYPE "Unable To Read the Value: ", status
    PAUSE
  END
.END

```

STEP 9:

Lista dei comandi e descrizioni da inviare al Flexibowl:

Action	Description
<b>MOVE</b>	Moves the feeder the current parameters.
<b>MOVE-FLIP</b>	Moves the feeder and activates Flip simultaneously
<b>MOVE-BLOW-FLIP</b>	Moves the feeder and activates Flip and blow simultaneously
<b>MOVE-BLOW</b>	Moves the feeder and activates Flip simultaneously
<b>SHAKE</b>	Shakes the feeder with the current parameters
<b>LIGHT ON</b>	Light on
<b>LIGHT OFF</b>	Light off
<b>FLIP</b>	Flip
<b>BLOW</b>	Blow
<b>QUICK_EMPTYING</b>	Quick Emptying Option
<b>RESET_ALARM</b>	Reset Alarm and enable the motor

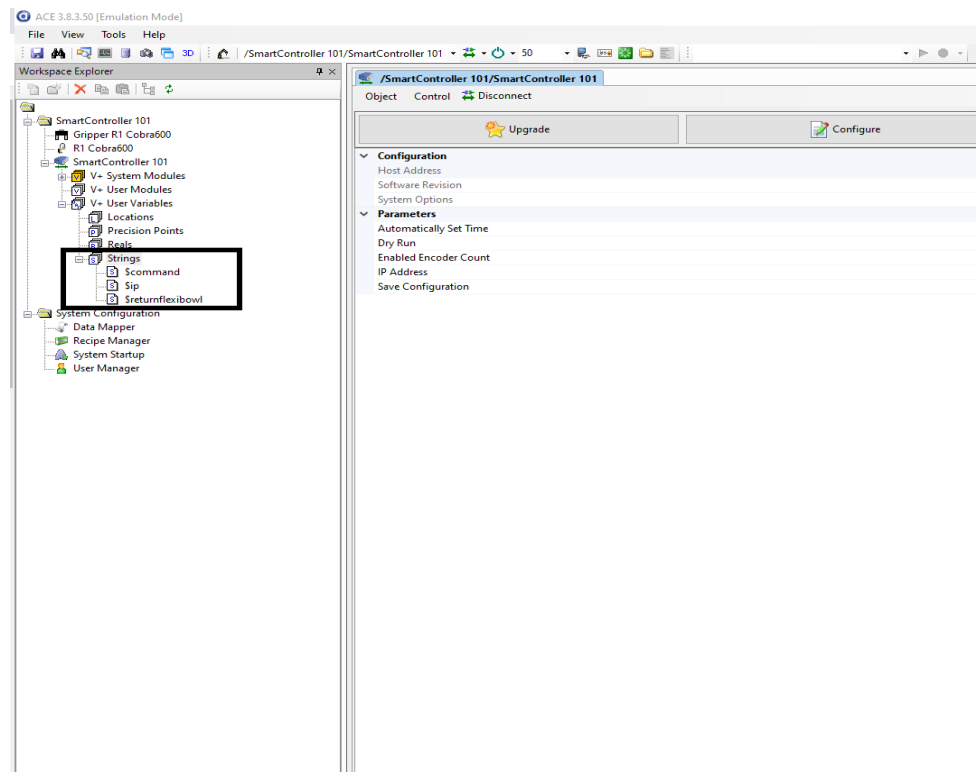
Command	Description
<b>QX2</b>	Move
<b>QX3</b>	Move - Flip
<b>QX4</b>	Move - Blow - Flip
<b>QX5</b>	Move - Blow
<b>QX5</b>	Shake
<b>QX7</b>	Light on
<b>QX8</b>	Light off
<b>QX9</b>	Blow
<b>QX10</b>	Flip
<b>QX11</b>	Quick Emptying Option
<b>QX12</b>	Reset Alarm



Da qui in poi vedremo come integrare il PluginFlexibowl in Ace 3.X o versioni precedenti.

STEP 1:

Creiamo TRE variabili V+ di tipo String,  
-Ip  
-Command  
-RerturnFlexibowl

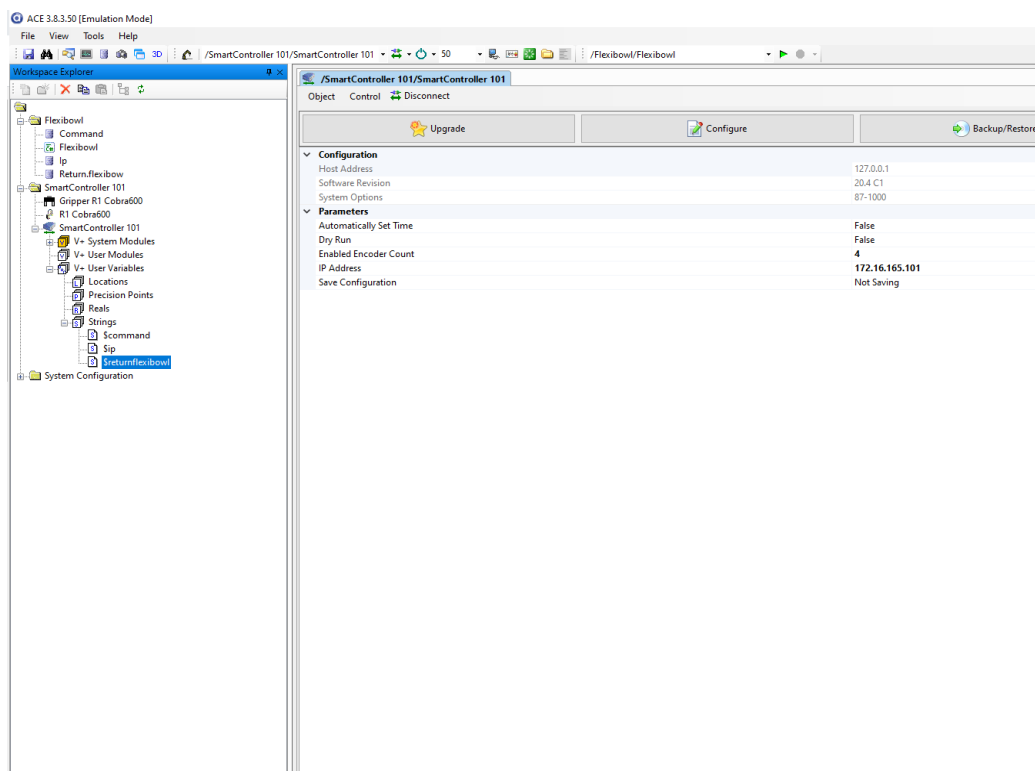


# ACE versione 3



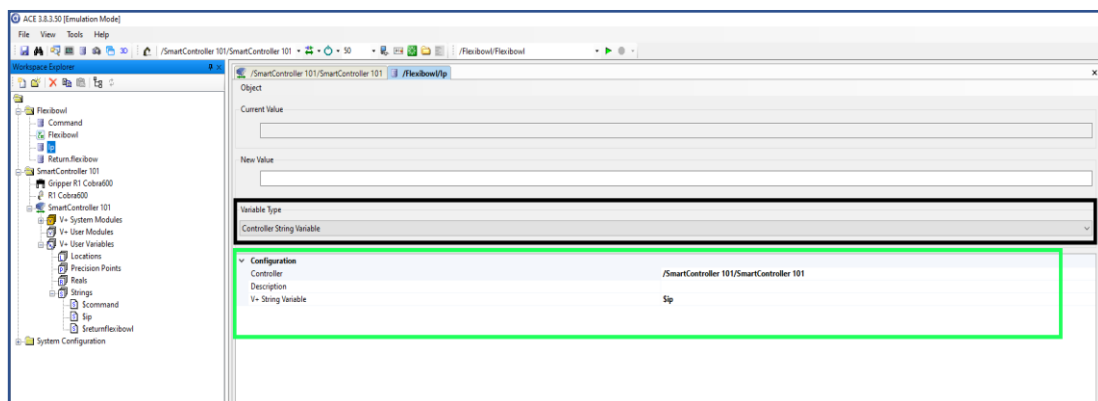
## STEP 2:

Creiamo una cartella nel WorkspaceExplorer e chiamiamola Flexibowl.  
A questo punto tasto destro sulla cartella appena creata, ImportWorkspaceFile, e carichiamo il file FlexibowlPlugin.awp da noi fornito.



## STEP 3:

Adesso andranno indicizzate le variabili V+ con le variabili C#.  
Effettuiamo un doppio clic ad esempio sulla variabile Ip C#. Impostando tale variabile come *ControllerStringVariabile* (riquadro nero) è possibile associarla alla nostra variabile V+ Ip (riquadro verde). Fare ciò per tutte e tre le variabili.



## STEP 4:

Bisogna verificare che i percorsi delle variabili C# siano corretti.

Per verificare ciò, bisogna controllare nel riquadro evidenziato nell'immagine, i percorsi delle tre variabili siano corretti.

Per fare ciò, selezionare una delle tre variabili C#, drag&drop nella pagina del codice, e controllare che il percorso sia corretto.

In questo caso avete creato un riferimento alla vostra variabile, controllate il percorso corretto ed eliminate la riga creata.

Verificare che i percorsi delle tre variabili nel codice siano corretti.

### Esempio:

Originale

```
IVariableString Command = (IVariableString) ace["/Application Manager0/Variables/Command"];
```

Modificata

```
IVariableString Command = (IVariableString) ace["/Application Manager4/Variables/Command"];
```

```

1  using Ace.Core.Server.Program;
2  using Ace.Core.Server;
3  using System;
4  using System.Collections.Generic;
5  using System.Diagnostics;
6
7  using System.Net;
8  using System.Net.Sockets;
9  using System.Text;
10
11
12 namespace Ace.Custom {
13
14     public class Program {
15
16         public AceServer ace;
17
18         public void Main () {
19
20             Trace.WriteLine("Flexibowl PlugIn " + DateTime.Now.ToString() + " Run");
21
22             string receiveString = "";
23             int bytesSent = 0;
24
25             UdpClient m_udpClient= new UdpClient(7777);
26
27             // To change!!!!!!!
28             IVariableString Command = (IVariableString) ace["/Flexibowl/Command"];
29             IVariableString Ip = (IVariableString) ace["/Flexibowl/Ip"];
30             IVariableString ReturnFlexibowl = (IVariableString) ace["/Flexibowl/Return.flexibowl"];
31
32             //!!!!!!!!!!!!!!!!!!!!!!
33             ReturnFlexibowl.CurrentValue="False";
34             IPEndPoint ep = new IPEndPoint(IPAddress.Parse(Ip.CurrentValue), 7775);
35
36             try {
37                 m_udpClient.Connect(ep);
38                 m_udpClient.Client.SendTimeout = 500;
39                 m_udpClient.Client.ReceiveTimeout = 500;
40             }
41             catch (ArgumentNullException ex)
42             {
43                 Trace.WriteLine(ex.ToString());
44             }
45         }
46     }
47 }

```

## STEP 5:

Ora vedremo come impostare le variabili e eseguire lo script da V+  
Creiamo un programma V+ con il codice della pagina successiva.  
Ricopiando il codice verificare che i percorsi delle variabili siano corretti, tipo:

```
$object = "/Application Manager0/Variables/Ip"
```

Andando a settare l'ip e il comando, eseguendo lo script V+ il flexibowl effettuerà il comando

```
;insert the data
;////////////////////////////////////
$ip="169.254.1.10"
$command="QX3"
;////////////////////////////////////

.PROGRAM flexibowlplg()

    AUTO $object, $variable, $ip, $command , $return.flexibow
, $method, $args[0]
    AUTO REAL status, is.alive

    ;insert the data
    ;////////////////////////////////////
    $ip="169.254.1.10"
    $command="QX3"
    ;////////////////////////////////////

    ;Execute the c#
    CALL rm.chk.server(is.alive)
    IF (is.alive == FALSE) THEN
        TYPE "Not Communicating"
        PAUSE
    END    ; Execute a script on the server and wait for 3
seconds for it to complete
    $object = "/Flexibowl/Flexibowl"
    $method = "Execute"
    CALL rm.execute($object, $method, 0, $args[], 5, status)
    IF (status < 0) THEN
        TYPE "Problem executing script: ", status
        PAUSE
    END

    ;the Answer
    ;$returnflexibowl

.END
```

*Al momento le variabili Ip,Command e return.flexibow in V+ sono locali (AUTO), per settarle da programmi esterni rendere queste variabili Globali, quindi non Auto.  
Eseguendo lo script V+ verrà eseguito lo script C# che azionerà il flexibowl*

