

System Manual

ESEMPI INTEGRAZIONE PRODOTTI

**Utilizzo delle funzionalità SOAP nei moduli
FactoryCast V3.1 in Excel, Internet Explorer
e VBasic**

SM_soap



SOMMARIO

1 - OGGETTO

2 - ARCHITETTURA

2.1 - Hardware

2.2 - Software

2.3 - Comunicazione

3 - IMPLEMENTAZIONE

3.1 - Funzionalità

3.2 - Hmi

3.3 - Plc

3.4 - Device

4 - COMPONENTI

1 - OGGETTO

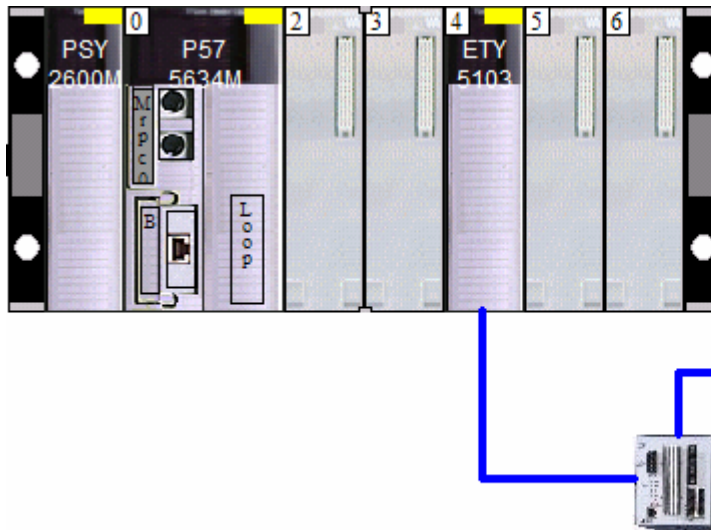
Scopo della prova è quello di realizzare una connessione tra un modulo FactoryCast ed un PC utilizzando la tecnologia SOAP xml.

Verrà analizzata l'implementazione in applicativi quali Excel, Internet Explorer e VisualBasic (dotNet).

Si vuole porre in risalto la semplicità di implementazione di questa tecnologia con un minimo impatto informatico (non sono richieste forti conoscenze informatiche di base) e senza l'ausilio di driver di comunicazione specifici (OPC) o fatti ad arte per ciascuna applicazione.

2.1 - ARCHITETTURA: Hardware

- Microsoft Office XP or later
- Microsoft Office XP Web Services Toolkit 2.0 Plugin
- Modicon FactoryCast module
 - 140NOE77111
 - TSXETY5103



2.2 - ARCHITETTURA: Software

PLC:

Non occorre alcuna specifica configurazione se non l'impostazione dell'indirizzo IP(Subnet e Gateway).

PC:

Il sistema operativo può essere 2K o XP(in merito alle prove in oggetto), ma può essere anche Linux.

2.3 - ARCHITETTURA: Comunicazione

Naturalmente il PC ed il PLC devono vedersi tra loro e quindi o appartengono allo stesso dominio di collisione(LAN) o sono correttamente configurati per attraversare Bridge e Router.

La cosa più semplice per accertarsi di ciò, è eseguire da PC un 'ping' della stazione PLC.

3.1 - IMPLEMENTAZIONE: Funzionalità

PC:

Nel caso della prova in oggetto, dove il sistema operativo è Microsoft, occorre installare il WebService 2.0 ed il SoapToolkit 3.0; entrambi si scaricano dal sito Microsoft. Contestualmente al WebService, prestare attenzione alla lingua di installazione di Office, ed installare quindi il WebService nella lingua di installazione stessa del pacchetto Office; il rischio è quello di avere installata una funzione che apparentemente funziona ma non è in grado di integrarsi e quindi funzionare correttamente.

Per Office installato in Italiano installare i seguenti add-on:

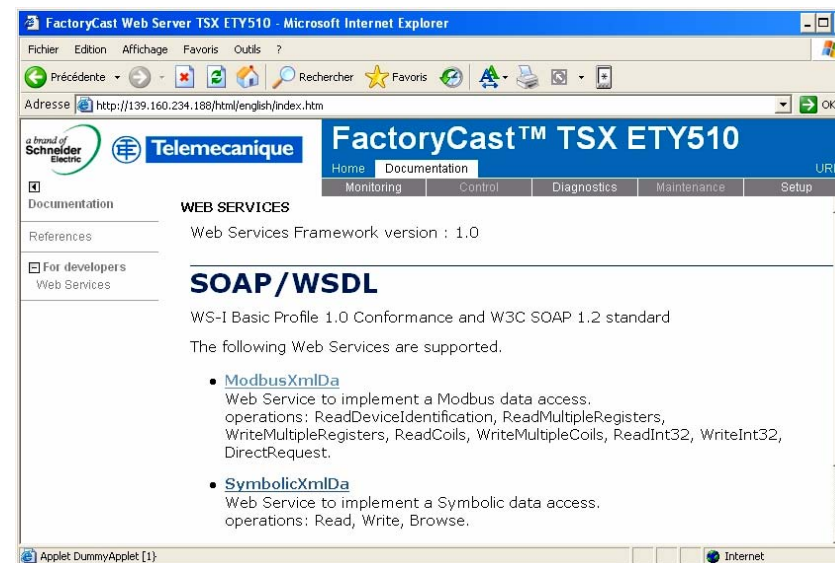
- [Web Service ToolKit 2.0](#)
- [SoapSDK 3.0](#)

3.1 - IMPLEMENTAZIONE: Funzionalità

Modulo FactoryCast:

- FactoryCast™ SOAP/XML Web services sono basati su standard W3C quali:
 - **WSDL** (Web Services Description Language) il linguaggio di descrizione Web Services, in formato XML.
 - **XML** (eXtensible Markup Language): lo standard universale di scambio dati.
 - **SOAP** (Single Object Access Protocol), il protocollo di scambio trasportato dal canale HTTP (HyperText Transfer Protocol) .

3.1 - IMPLEMENTAZIONE: Funzionalità



❑ Il server SOAP fornisce accesso ai dati del PLC

- ❑ Dati disponibili : Variabili PLC

❑ 2 Livelli di accesso ai dati del PLC

- ❑ ModbusXMLDA Web service: Accesso diretto alle variabili mediante indirizzo fisico(es:%MW100).
- ❑ SymbolicXMLDA Web service : Accesso alle variabili mediante nome del simbolo(**attualmente non sono supportate le variabili non allocate!**)

3.2 - IMPLEMENTAZIONE: Hmi

- **Interfaccia SOAP/XML 'server'**

Questa implementazione abilita un'applicazione client SOAP (ad esempio MES,ERP,ect.) a comunicare direttamente con i moduli WEB Server integrati nei sistemi PLC. Lo scambio è generato dall'applicazione SOAP client(il PC) al quale risponde la scheda FactoryCast.

1 – Creazione dell'applicazione client:

con apprendimento automatico del 'Web Service' disponibile.

L'ambiente di sviluppo(ad esempio Visual Studio .NET) cerca nel FactoryCast server il quale risponde con la lista dei servizi disponibili e con l'interfaccia standard WSDL implementata nel modulo.

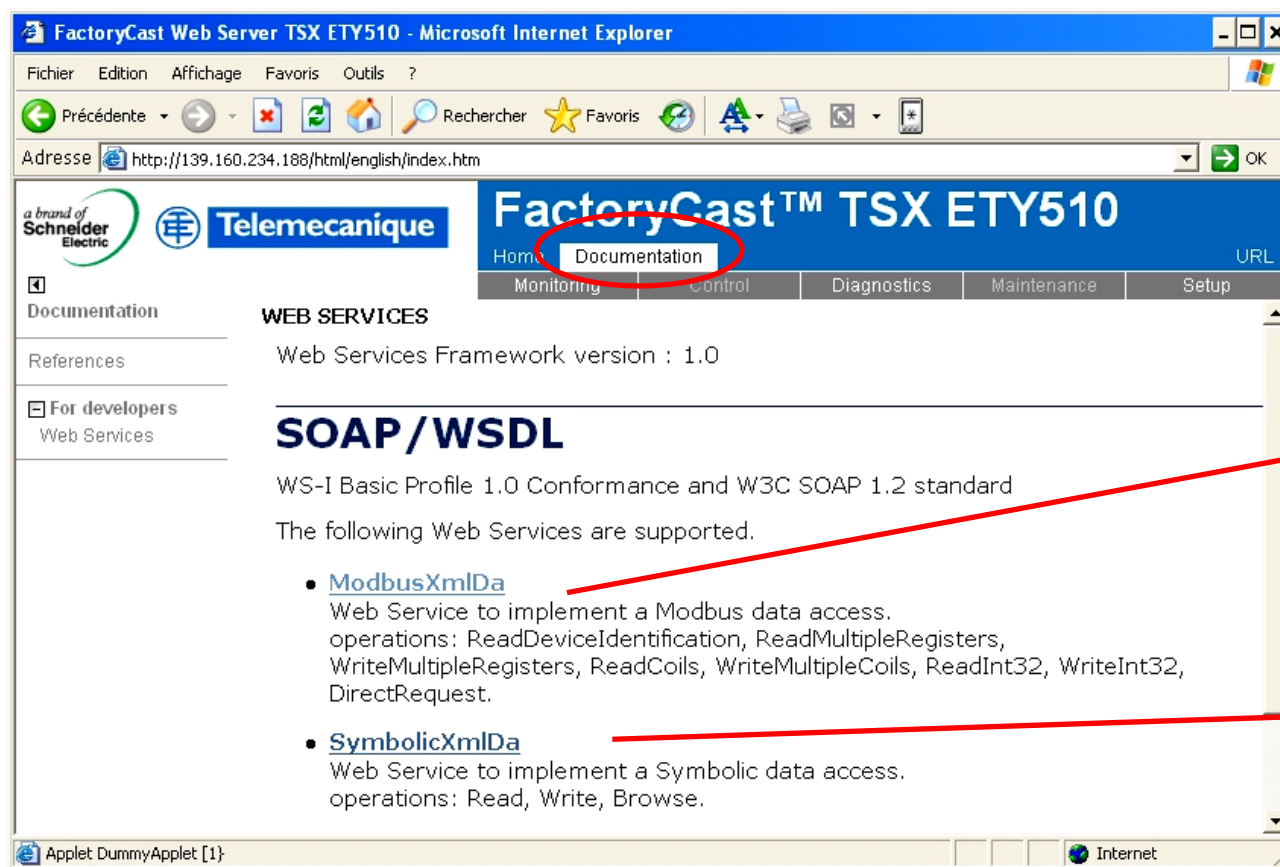
2 – Sviluppo dell'applicazione client:

Lo sviluppatore integra i servizi Web utilizzando il codice recuperato dal modulo FactoryCast stesso al momento dell'autoapprendimento(punto precedente).

3 – Esecuzione dell'applicazione client:

L'applicazione client comunica in real-time con il modulo web server FactoryCast mediante l'ausilio del protocollo SOAP.

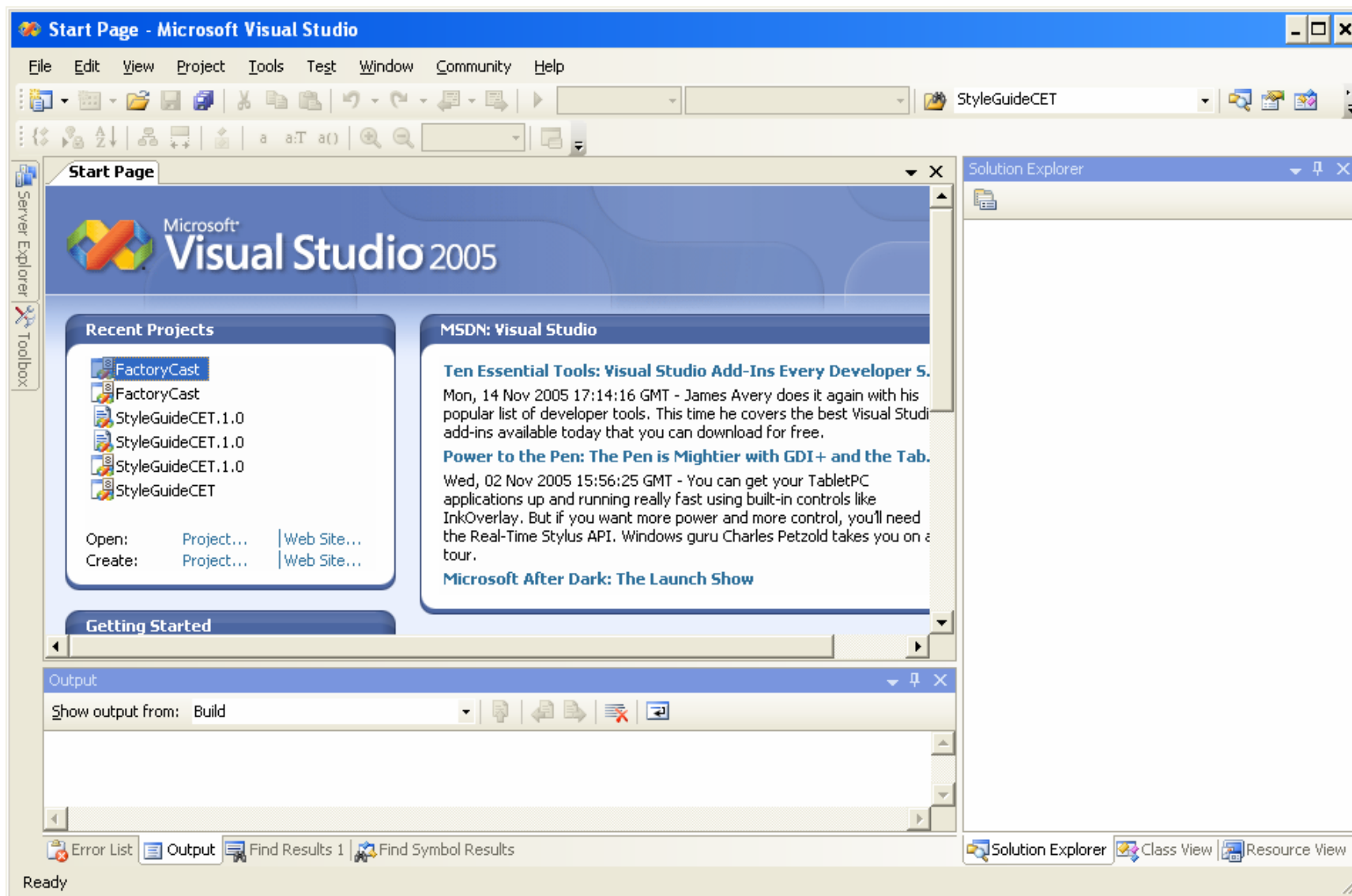
3.2 - IMPLEMENTAZIONE: Hmi – Interfaccia WSDL in FactoryCast



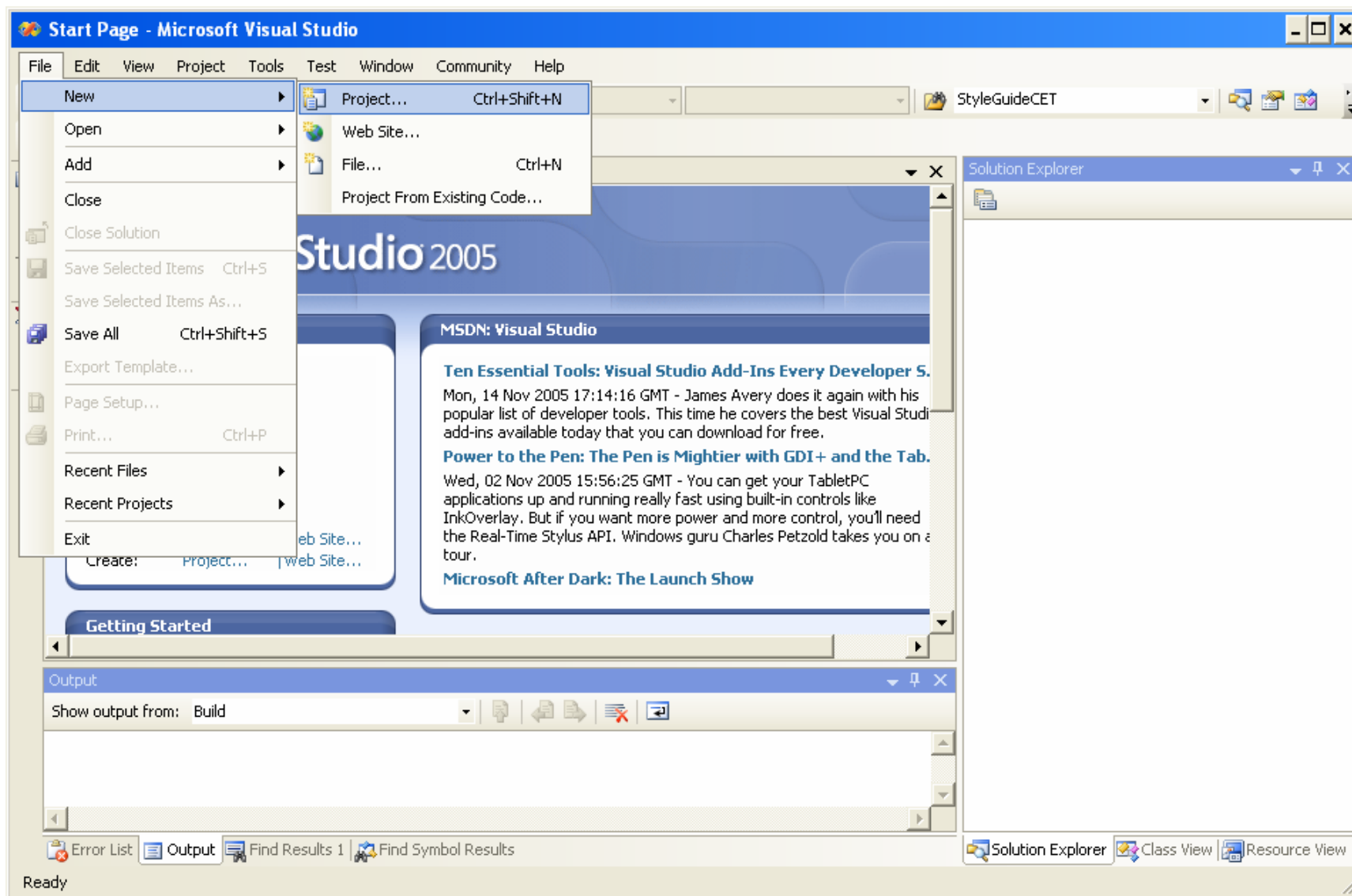
ModbusXmlDa :
ReadDeviceIdentification
ReadMultipleRegister
WriteMultipleRegister
ReadCoils
WriteMultipleCoils
ReadDiscreteInputs

SymbolicXmlDa :
Read
Write
Browse

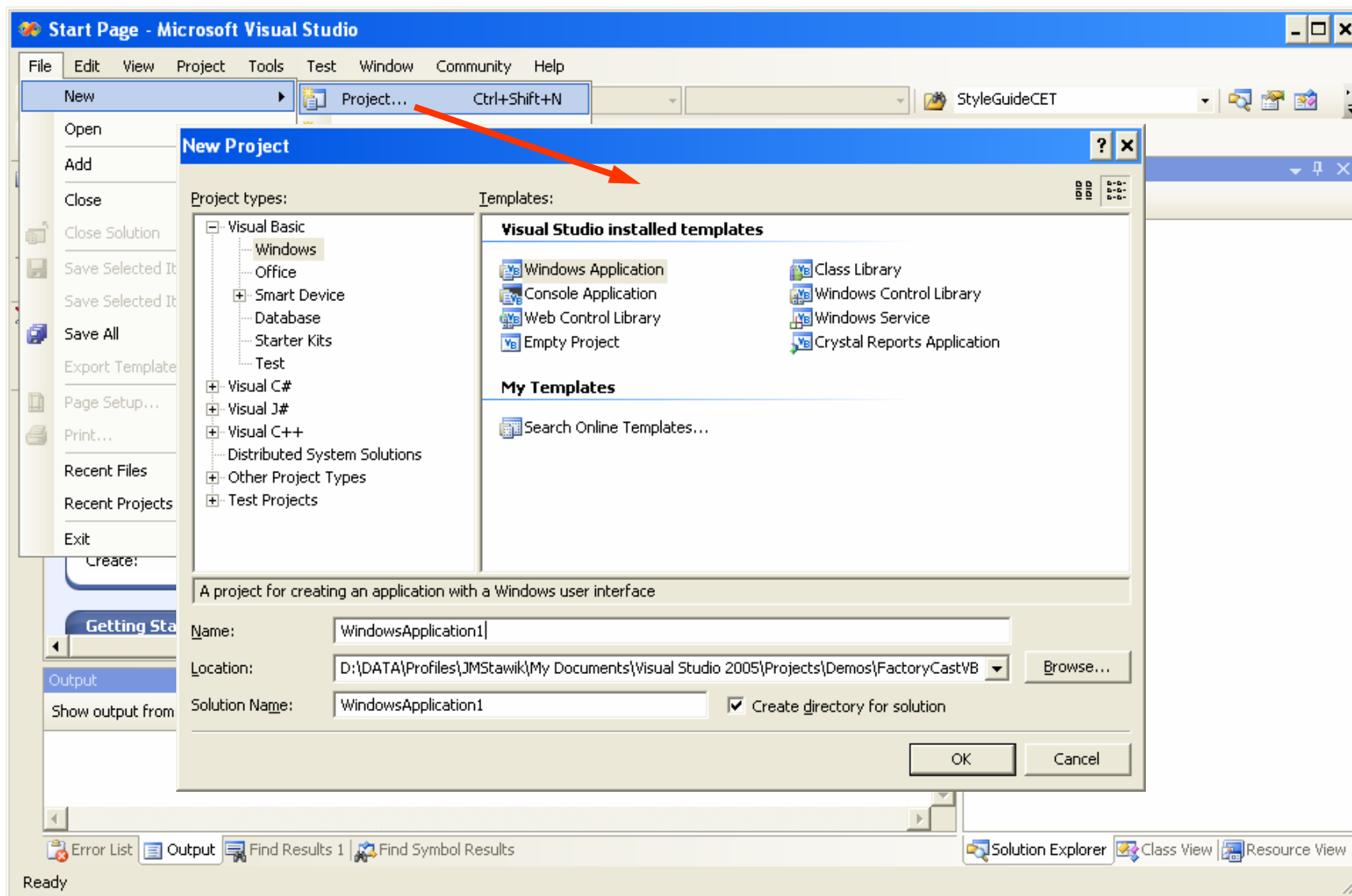
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



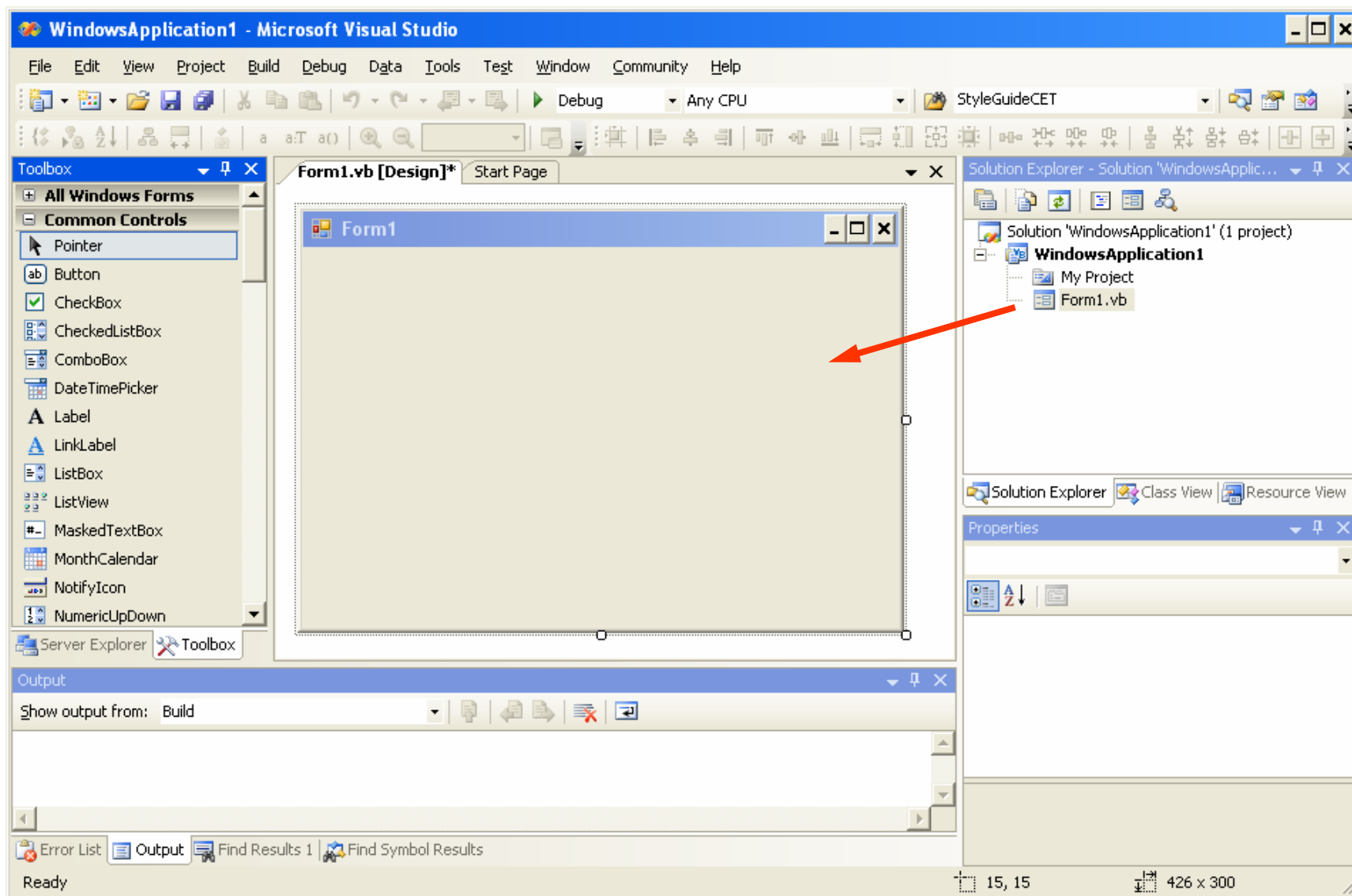
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



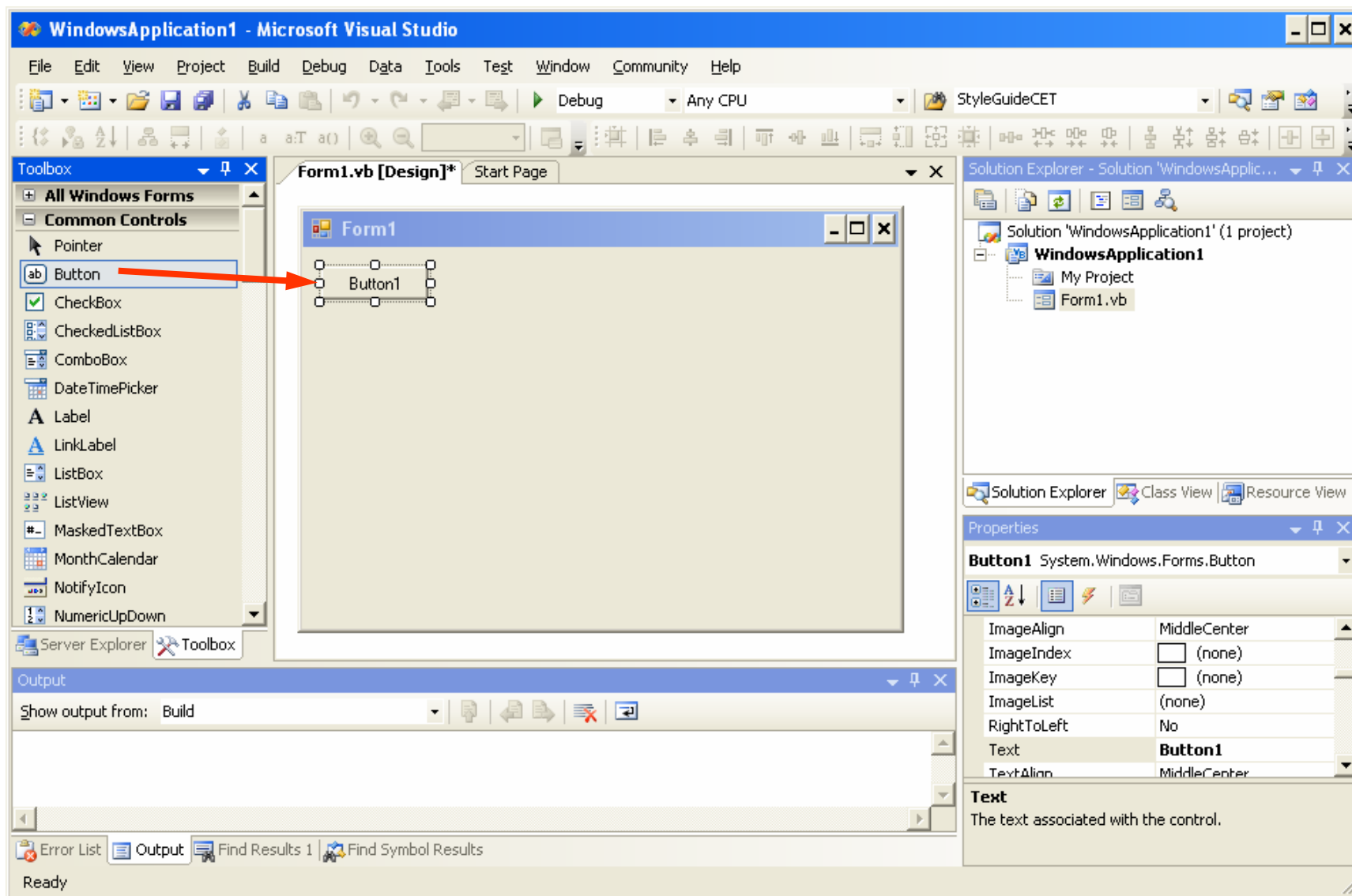
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



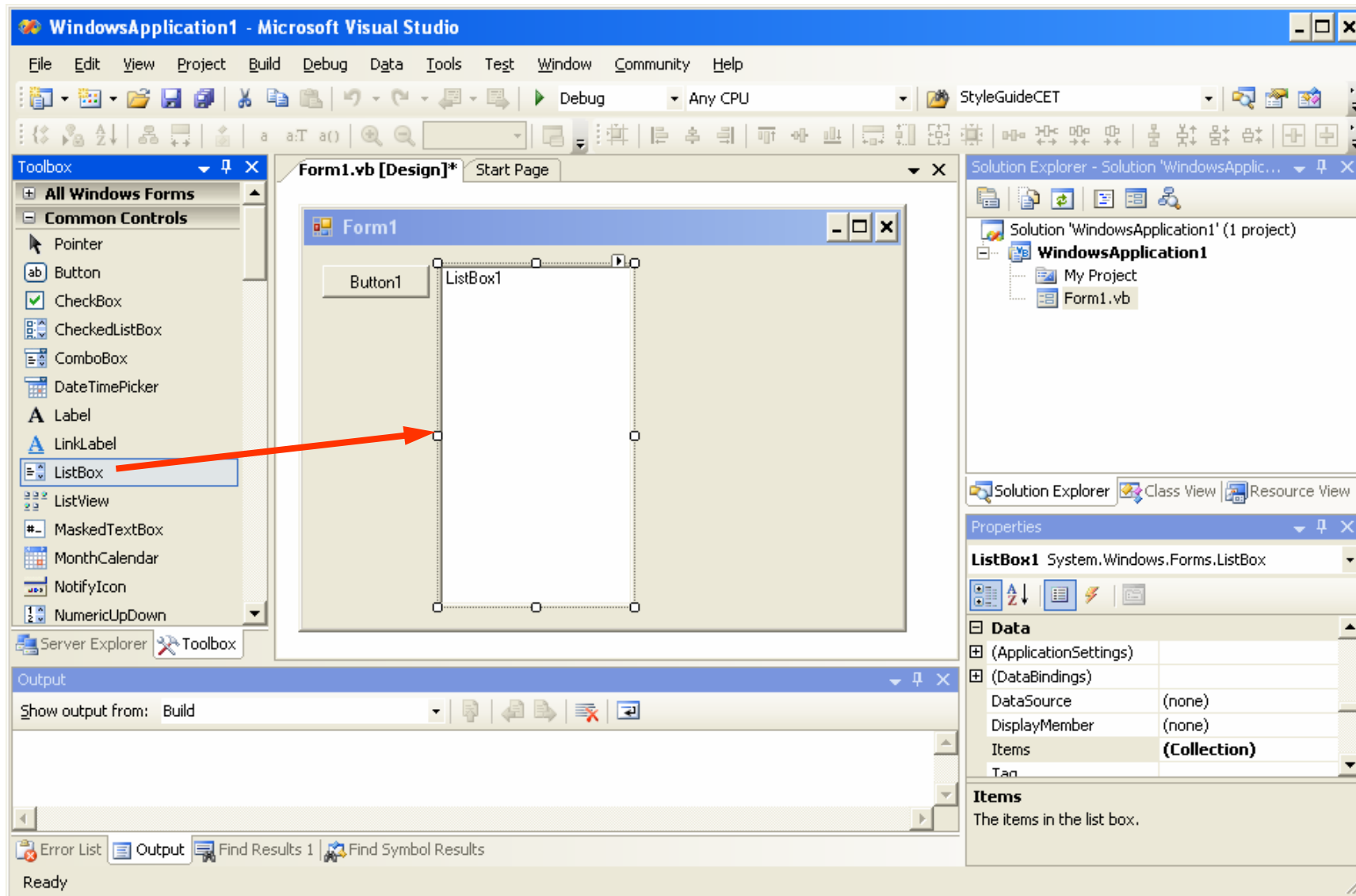
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



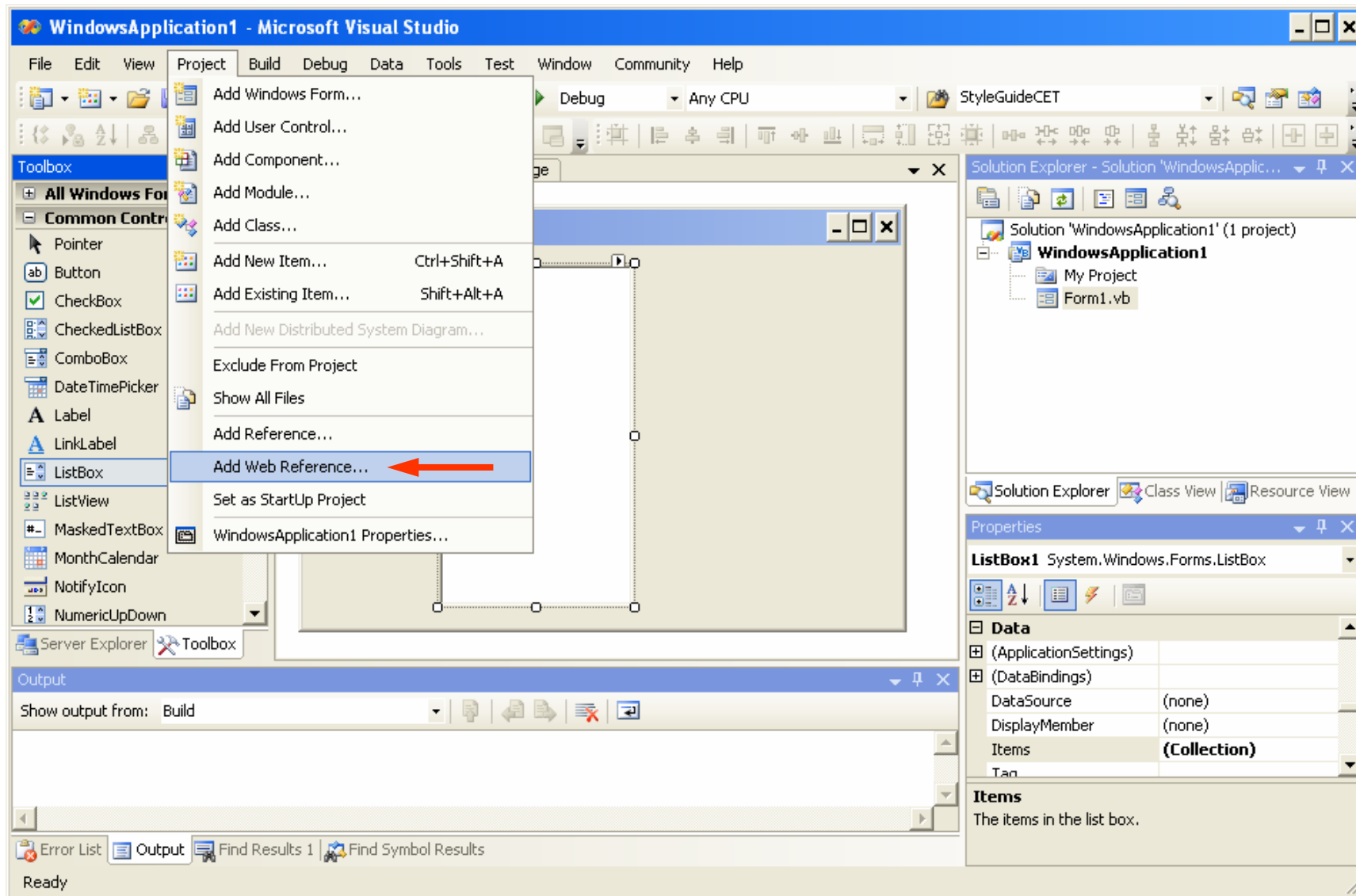
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



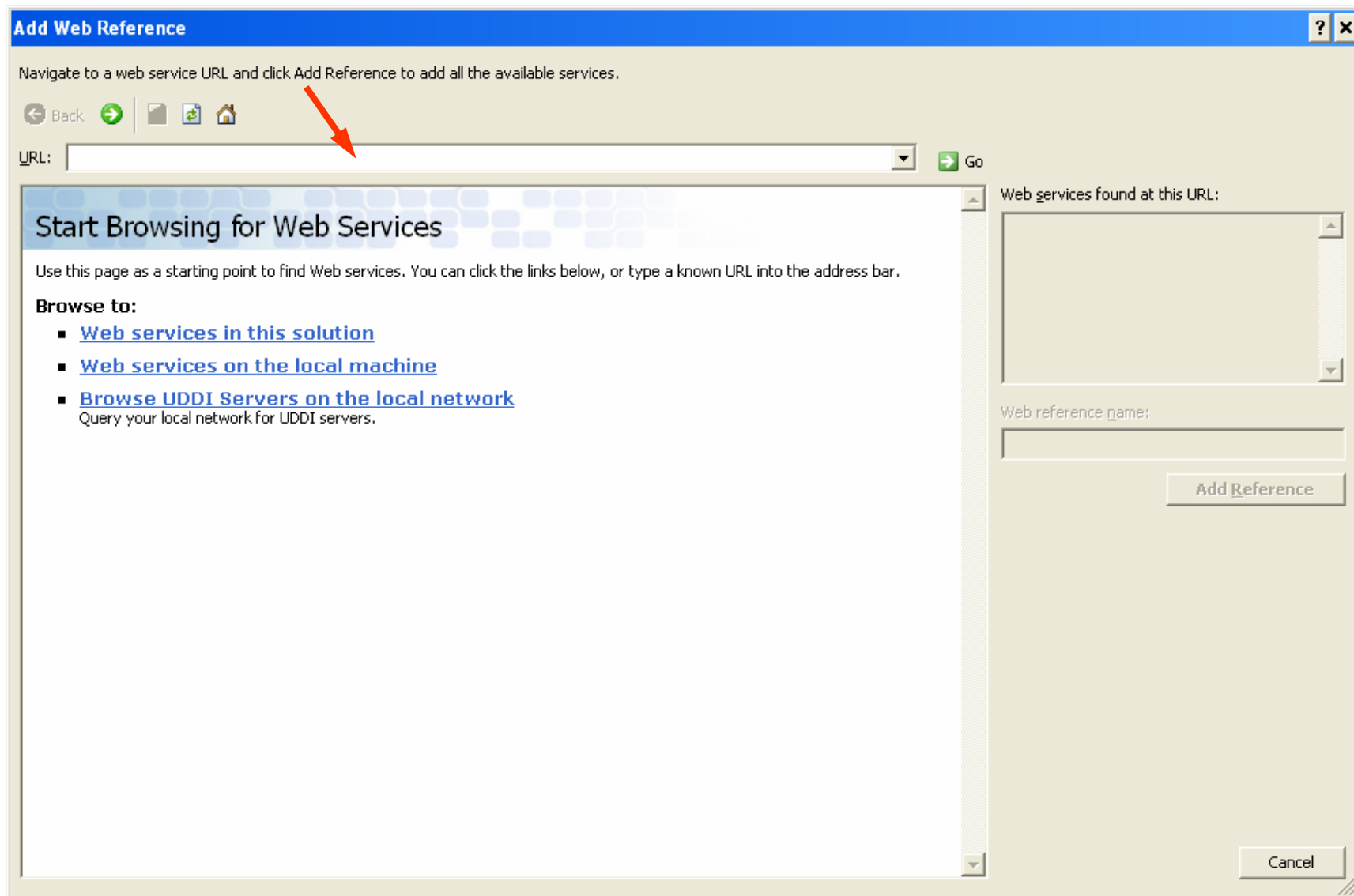
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



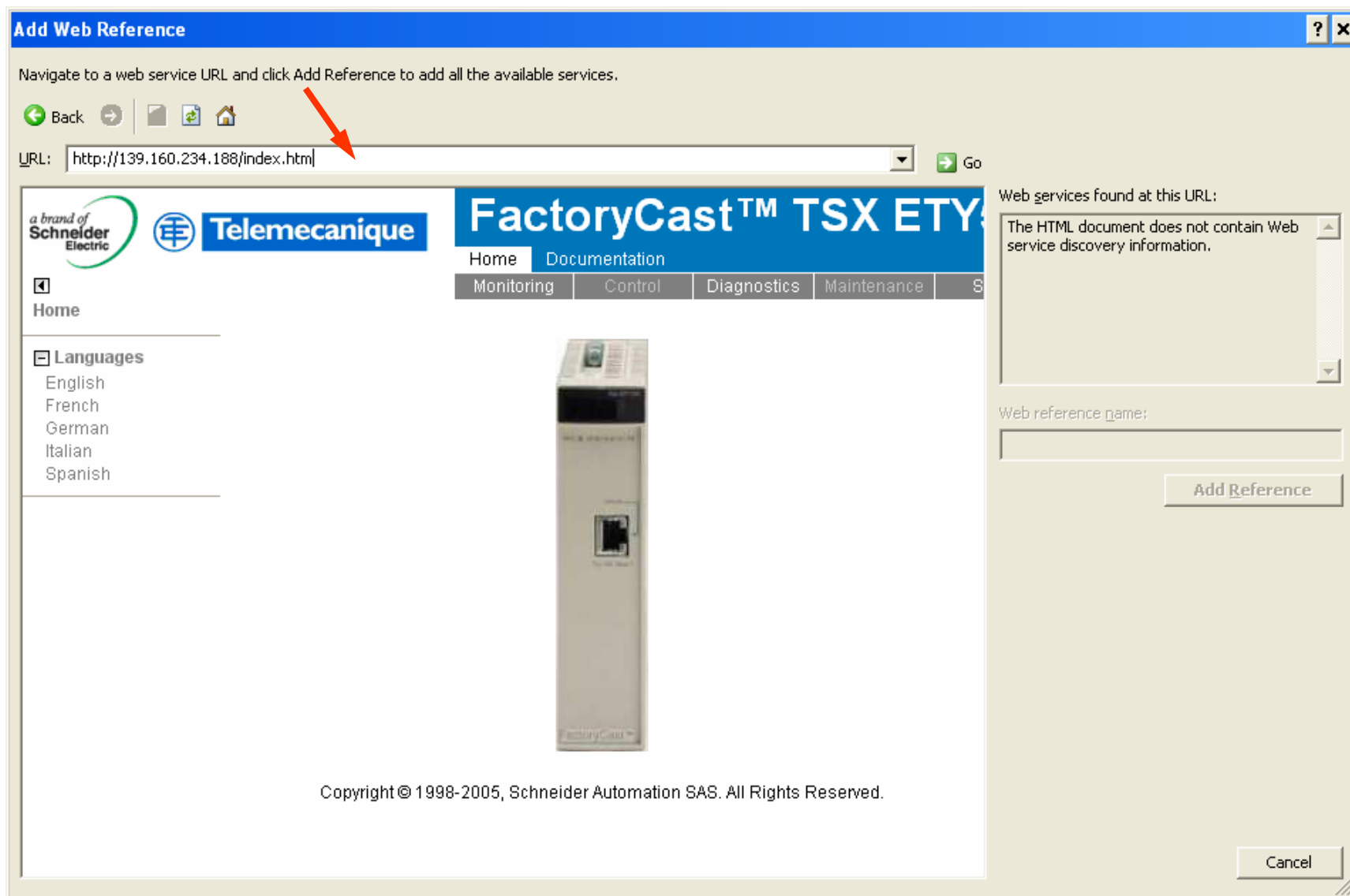
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



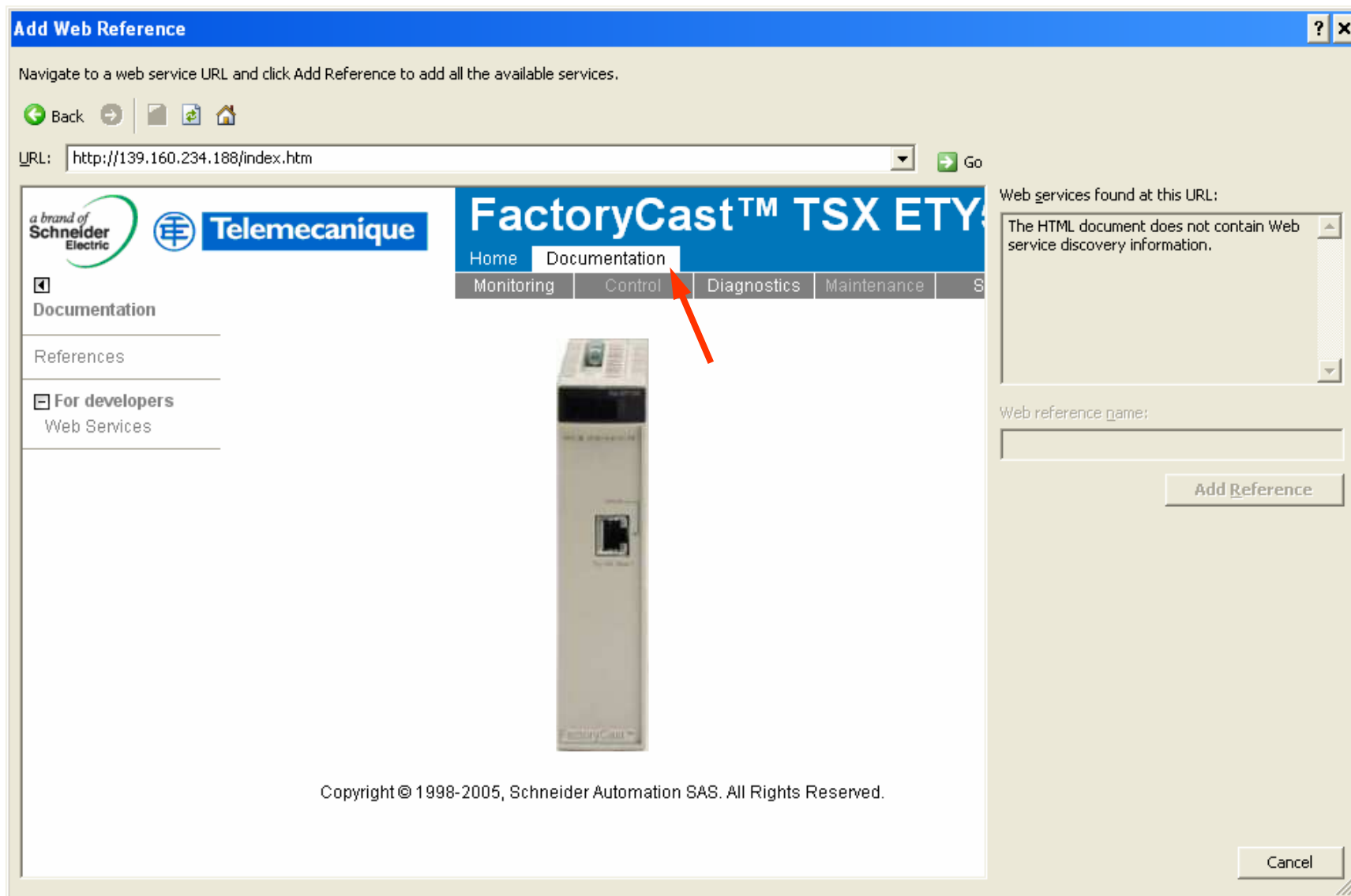
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET

Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available services.

Back | | | |

URL: Go

FactoryCast™ TSX ETY

Home | Documentation

Monitoring | Control | Diagnostics | Maintenance | S

Documentation

References

For developers

Web Services

WEB SERVICES

Web Services Framework version : 1.0

SOAP/WSDL

WS-I Basic Profile 1.0 Conformance and W3C SOAP 1.2 standard

The following Web Services are supported.

- [ModbusXmlDa](#)
Web Service to implement a Modbus data access.
operations: ReadDeviceIdentification, ReadMultipleRegisters, WriteMultipleRegisters, ReadCoils, WriteMultipleCoils, ReadInt32, WriteInt32.
- [SymbolicXmlDa](#)
Web Service to implement a Symbolic data access.
operations: Read, Write, Browse.

Copyright © 2000-2005 Schneider AUTOMATION SAS. All Rights Reserved.

Web services found at this URL:
The HTML document does not contain Web service discovery information.

Web reference name:

Add Reference

Cancel

3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET

The screenshot shows the 'Add Web Reference' dialog box in Visual Studio. The dialog has a title bar 'Add Web Reference' and a close button. Below the title bar, there is a navigation bar with 'Back', 'Forward', and 'Home' icons. The 'URL' field contains 'http://139.160.234.188/index.htm' and a 'Go' button. The main content area displays a web page from 'FactoryCast™ TSX ETY' with a navigation menu and a section titled 'ModbusXmlDa Web Service'. The page content includes a description of the web service and a list of supported operations: 'ReadDeviceIdentification', 'ReadMultipleRegisters', and 'WriteMultipleRegisters'. On the right side of the dialog, there is a list of 'Web services found at this URL:' containing '1 Service Found:' and '- ModbusXmlDa'. Below this list is a text box for 'Web reference name:' containing 'WebReference'. At the bottom right of the dialog, there is an 'Add Reference' button, which is highlighted with a red arrow, and a 'Cancel' button.

3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET

The screenshot displays the Microsoft Visual Studio IDE. The main window shows the design view of 'Form1.vb [Design]*'. The form contains two controls: 'Button1' and 'ListBox1'. The 'Toolbox' on the left lists various Windows Forms controls, with 'ListBox' selected. The 'Solution Explorer' on the right shows the project structure for 'WindowsApplication1', with a 'WebReference' folder highlighted by a red arrow. The 'Properties' window at the bottom right shows the 'WebReference' folder properties, including the 'Web Reference URL' set to 'http://139.160.234.188/ws/1'. The 'Output' window at the bottom left shows the message 'Web reference update complete.'

3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET

The screenshot displays the Microsoft Visual Studio IDE in design mode for a Windows application. The main window shows a form titled 'Form1' containing a 'Button1' and a 'ListBox1'. A red arrow points to 'Button1'. The 'Toolbox' on the left lists various Windows Forms controls, with 'ListBox' selected. The 'Solution Explorer' on the right shows the project structure, including a 'WebReference' folder. The 'Properties' window at the bottom right shows the 'WebReference' folder properties, including the 'Web Reference URL'.

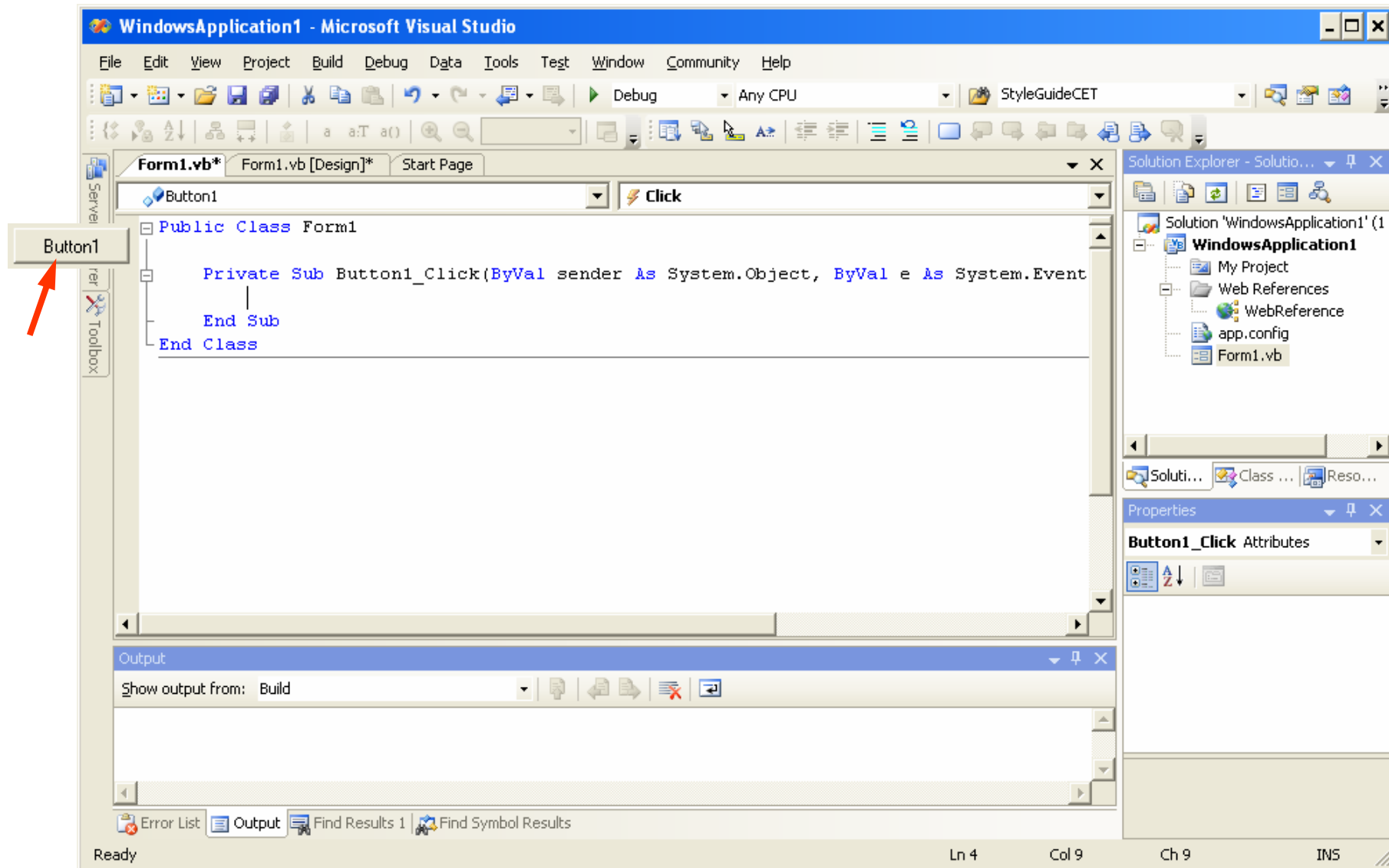
WebReference Folder Properties

Property	Value
Folder Name	WebReference
URL Behavior	Dynamic
Web Reference URL	http://139.160.234.188/ws/...

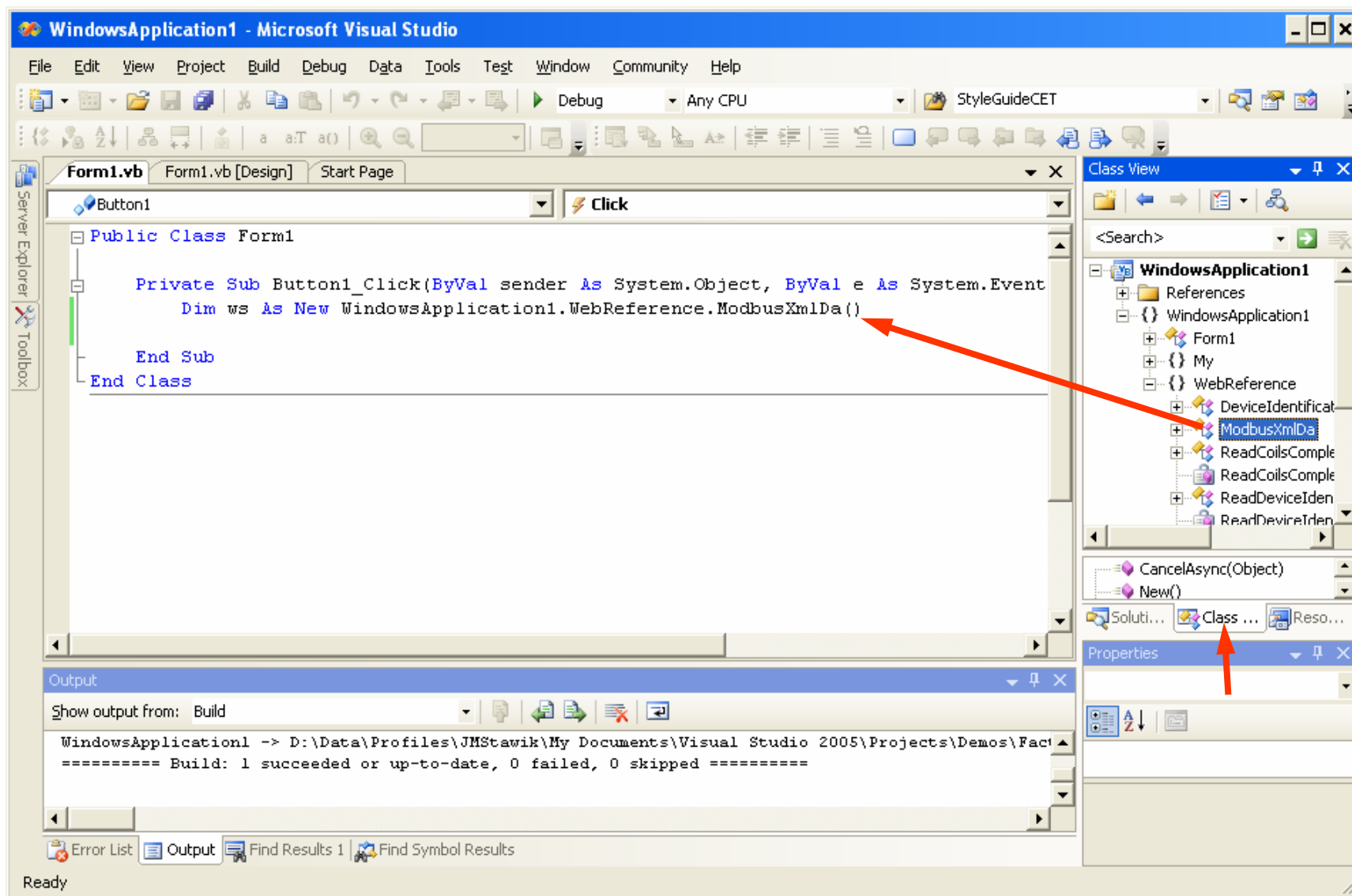
Misc

Web reference update complete.

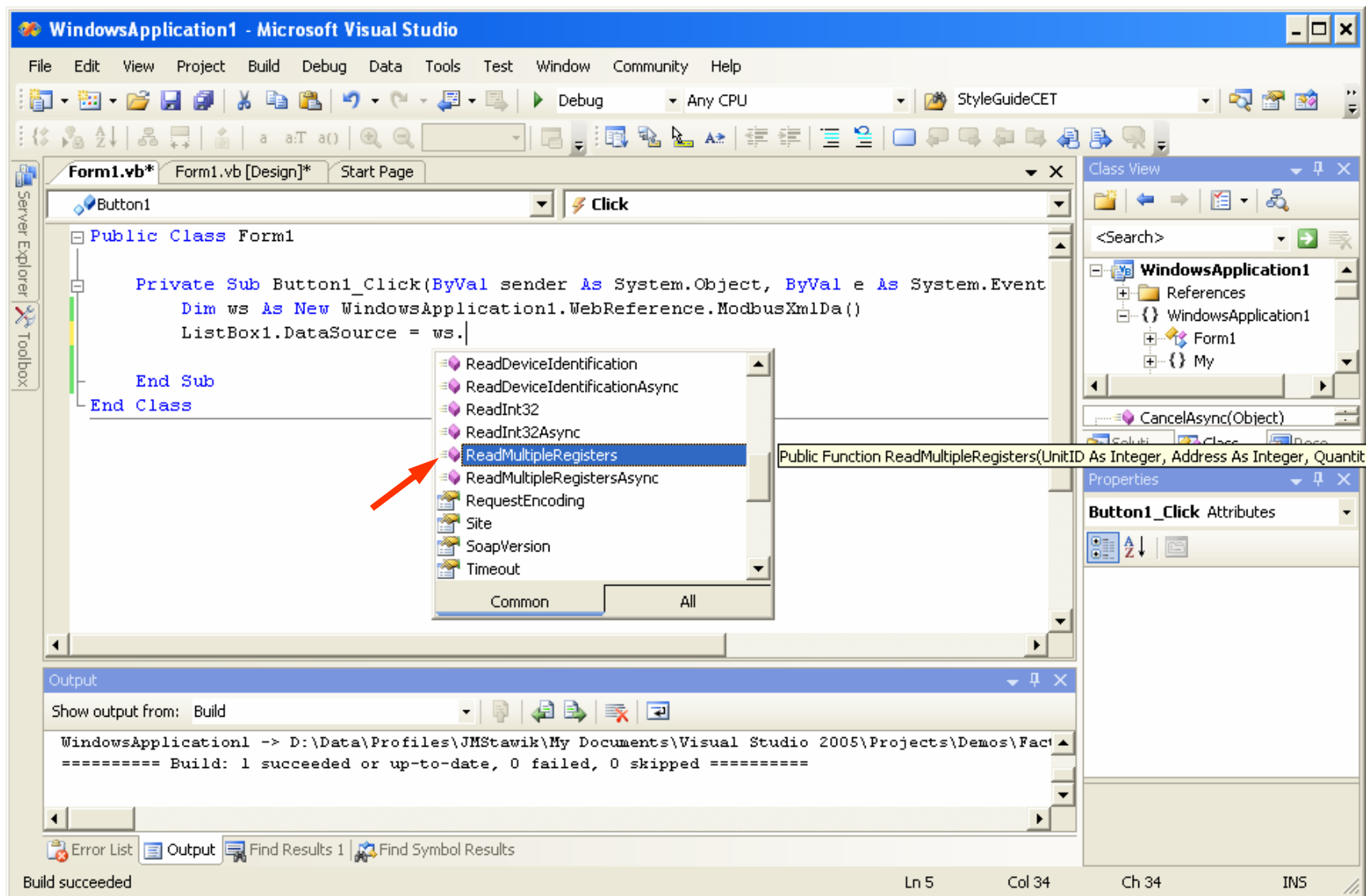
3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET

The screenshot shows the Microsoft Visual Studio IDE. The main window displays the code for `Form1` in `Form1.vb`. The code is as follows:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Dim ws As New WindowsApplication1.WebReference.ModbusXmlDa()
        ListBox1.DataSource = ws.ReadMultipleRegisters(
            ReadMultipleRegisters (UnitID As Integer, Address As Integer, Quantity As Integer) As Integer() {} My
    End Sub
End Class
```

A tooltip is visible over the `ReadMultipleRegisters` method call, showing its signature: `ReadMultipleRegisters (UnitID As Integer, Address As Integer, Quantity As Integer) As Integer() {} My`. A red arrow points to the `UnitID As Integer` parameter in the tooltip.

The right-hand side of the IDE shows the `Class View` pane with a search bar and a tree view containing `WindowsApplication1`, `References`, `WindowsApplication1`, and `Form1`. Below that is the `Properties` pane for `Button1_Click` attributes.

The bottom pane shows the `Output` window with the following text:

```
Show output from: Build
WindowsApplication1 -> D:\Data\Profiles\JMStawik\My Documents\Visual Studio 2005\Projects\Demos\Fac...
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
```

The status bar at the bottom indicates `Build succeeded`, `Ln 5`, `Col 56`, `Ch 56`, and `INS`.

3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET

The screenshot displays the Microsoft Visual Studio IDE with the following details:

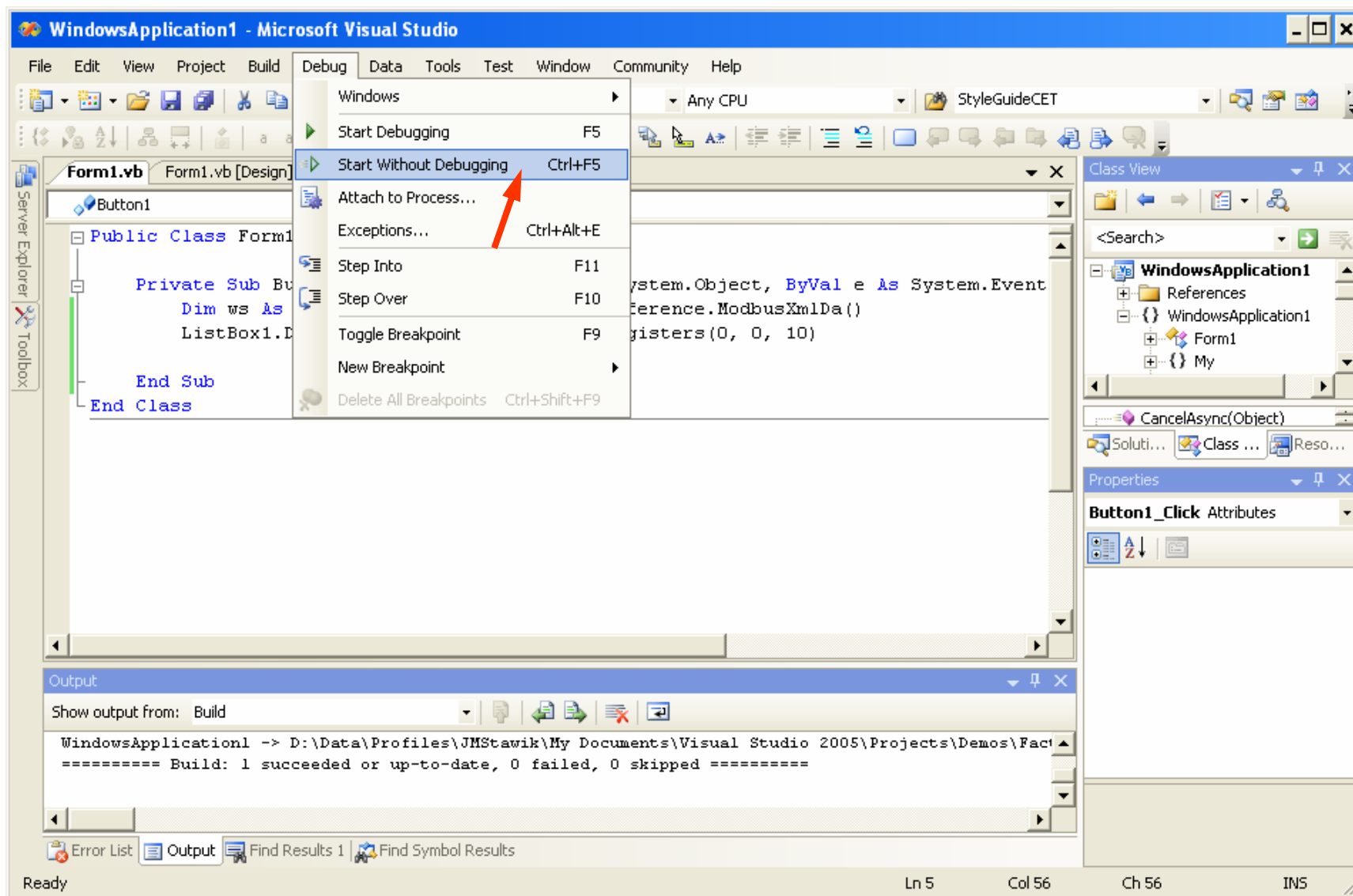
- Project Name:** WindowsApplication1 - Microsoft Visual Studio
- File Name:** Form1.vb*
- Code:**

```

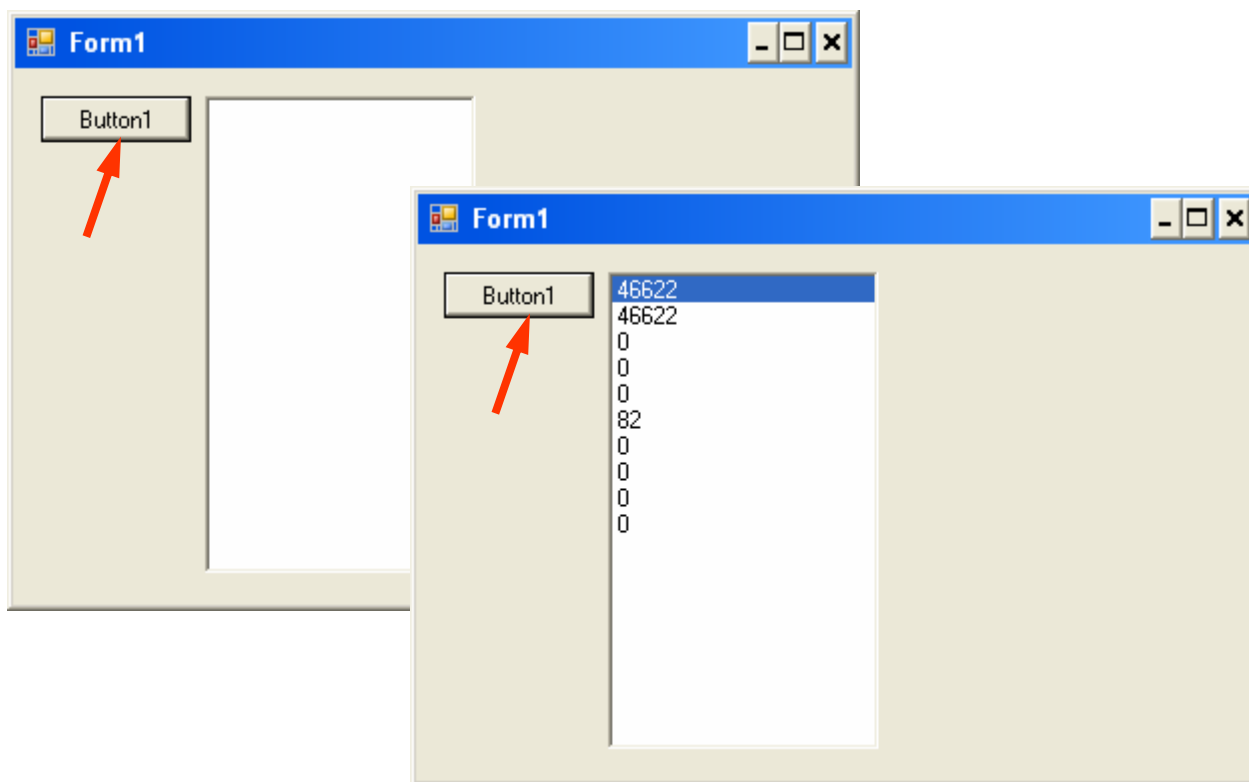
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Dim ws As New WindowsApplication1.WebReference.ModbusXmlDa()
        ListBox1.DataSource = ws.ReadMultipleRegisters(0, 0, 10)
    End Sub
End Class

```
- Tooltip:** ReadMultipleRegisters (UnitID As Integer, Address As Integer, Quantity As Integer) As Integer() {} My
- Class View:** Shows the project structure including WindowsApplication1, References, and Form1.
- Properties:** Shows Button1_Click Attributes.
- Output Window:** Shows the build output: "Build succeeded" and "WindowsApplication1 -> D:\Data\Profiles\JMStawik\My Documents\Visual Studio 2005\Projects\Demos\Fac...".
- Status Bar:** Shows "Build succeeded", "Ln 5", "Col 56", "Ch 56", and "INS".

3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi – VisualBasic .NET



3.2 - IMPLEMENTAZIONE: Hmi - EXCEL

The image displays two overlapping screenshots of the FactoryCast™ NOE 771 11 web interface, accessed via Microsoft Internet Explorer. The browser address bar shows the URL `http://192.168.1.75/html/english/index.htm`.

Left Screenshot (http://192.168.1.75/index.htm):

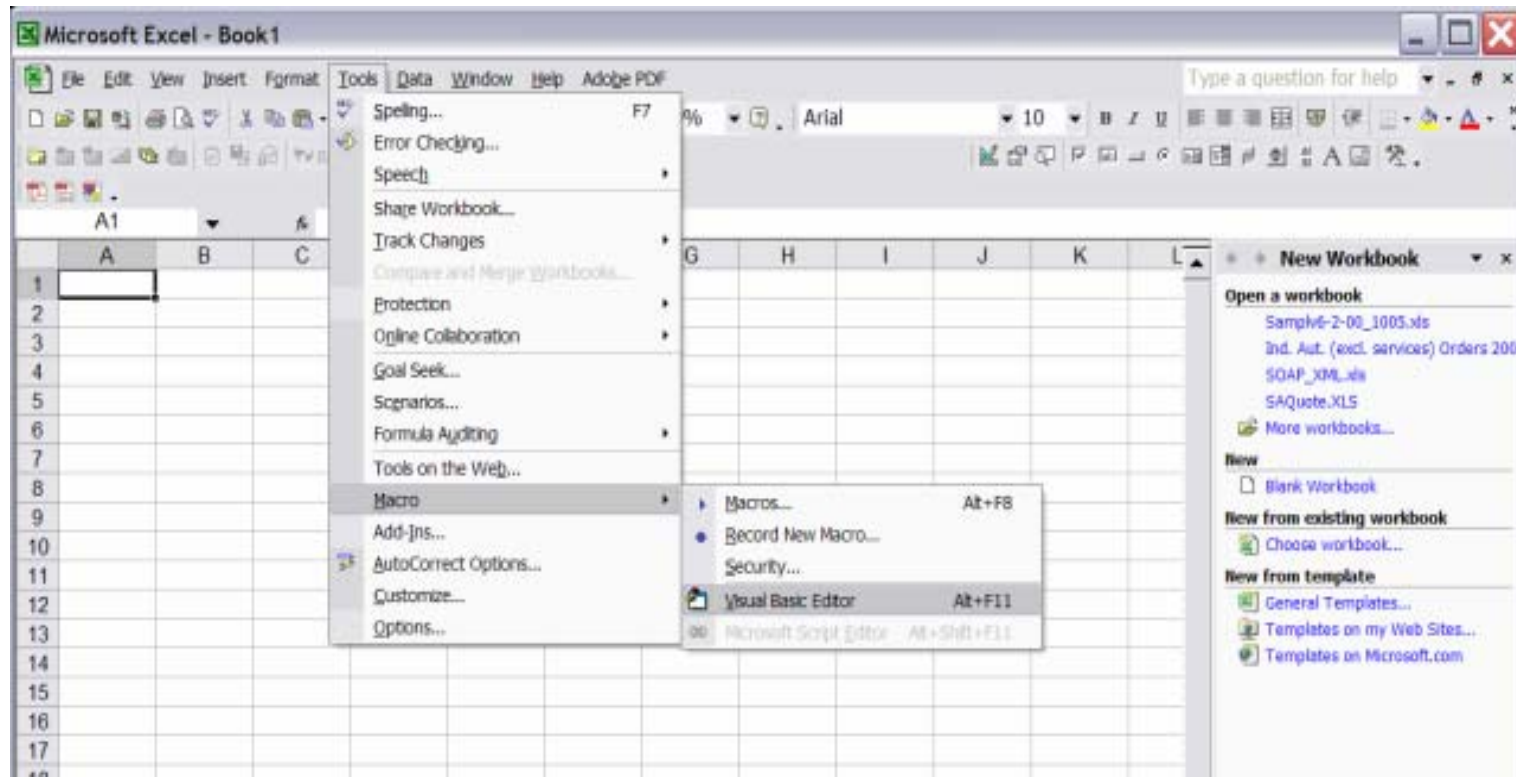
- Navigation:** Home, Documentation, Monitoring, Control.
- WEB SERVICES:** Web Services Framework version : 1.0
- SOAP/WSDL:** WS-I Basic Profile 1.0 Conformance and W3C SOAP 1.2 st. The following Web Services are supported.
 - ModbusXmlDa:** Web Service to implement a Modbus data access. operations: ReadDeviceIdentification, ReadMultipleReq, ReadInt32, WriteInt32, DirectRequest.
 - SymbolicXmlDa:** Web Service to implement a Symbolic data access. operations: Read, Write, Browse.
- Copyright:** © 2000-2005 Schneider AUTOMATION SAS. All Rights Reserved.

Right Screenshot (http://192.168.1.75/html/english/index.htm):

- SymbolicXmlDa Web Service:** Click [here](#) for a complete list of Web Services.
- Operations:** Web Service to implement a Symbolic data access. The following operations are supported. For a formal definition, please review the [WSDL 1.1](#) document, or review the [WSDL 1.1-SOAP 1.1](#) and [WSDL 1.1-SOAP 1.2](#) documents for a SOAP binding specific.
 - Read:** Operation to read item list value. inputs: **string[]** Symbol. outputs: **Items[]** ReadResult.
 - Write:** Operation to write item list value. inputs: **Items[]** ItemList. outputs:
 - Browse:** Operation to browse item list. inputs: outputs: **Descriptions[]** BrowseResult.
- Copyright:** © 2000-2005 Schneider AUTOMATION SAS. All Rights Reserved.

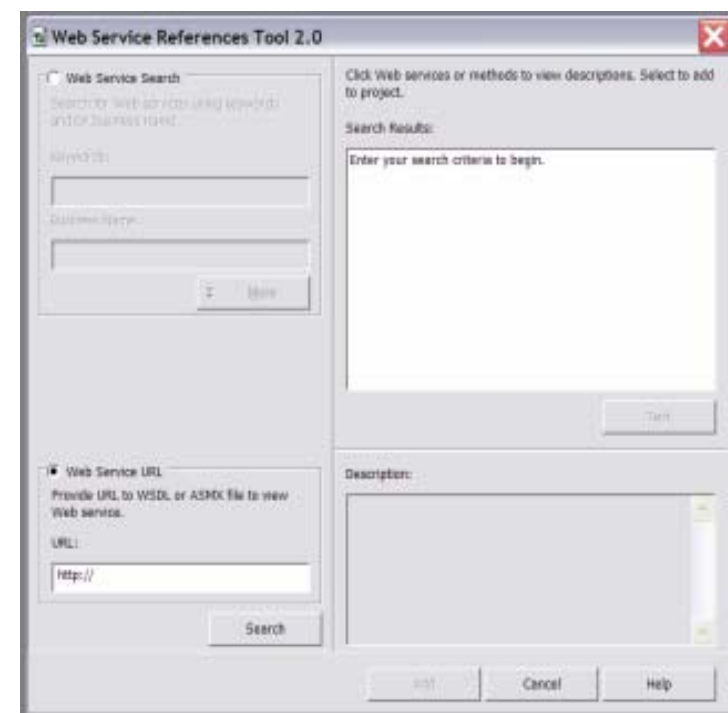
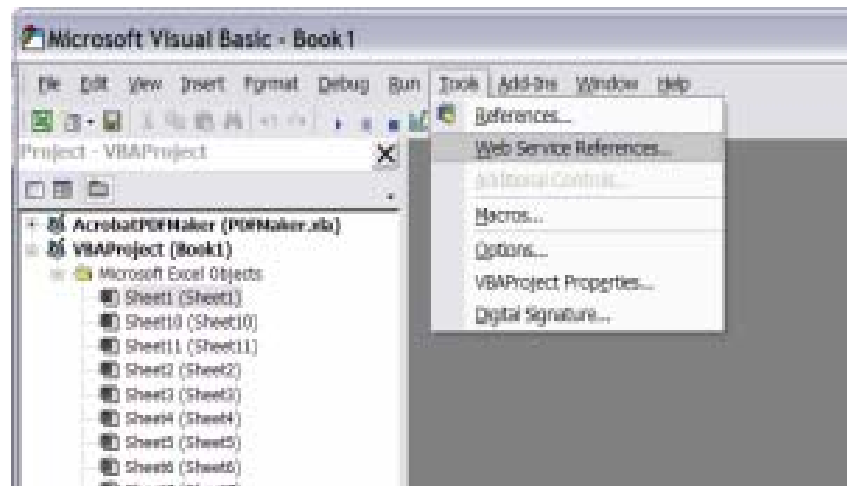
3.2 - IMPLEMENTAZIONE: Hmi - EXCEL

Aprire Excel e lanciare il VisualBasic Editor



3.2 - IMPLEMENTAZIONE: Hmi - EXCEL

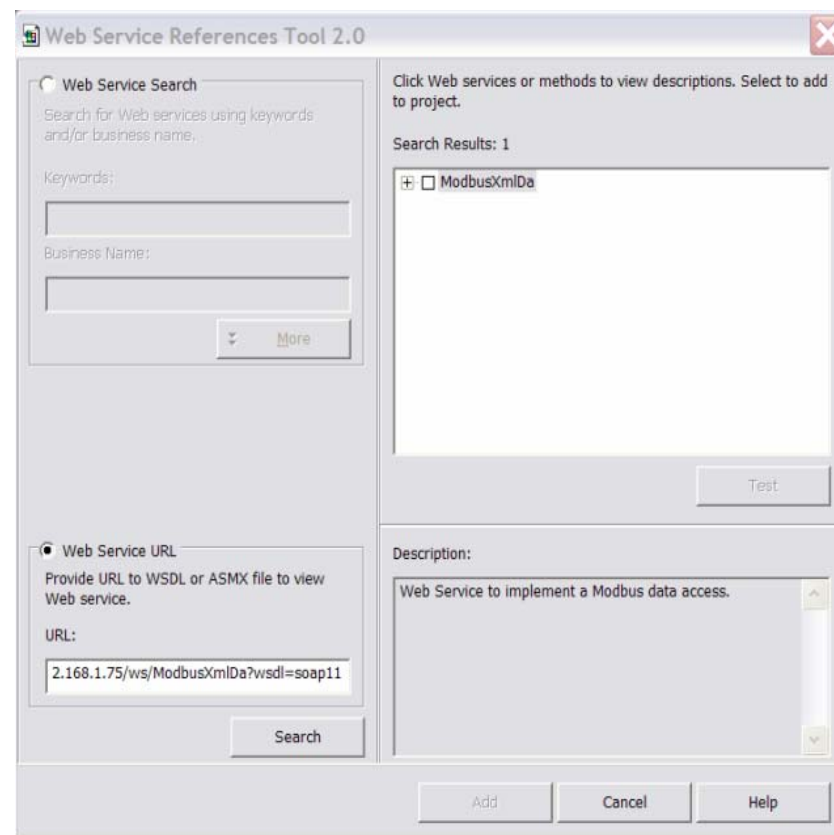
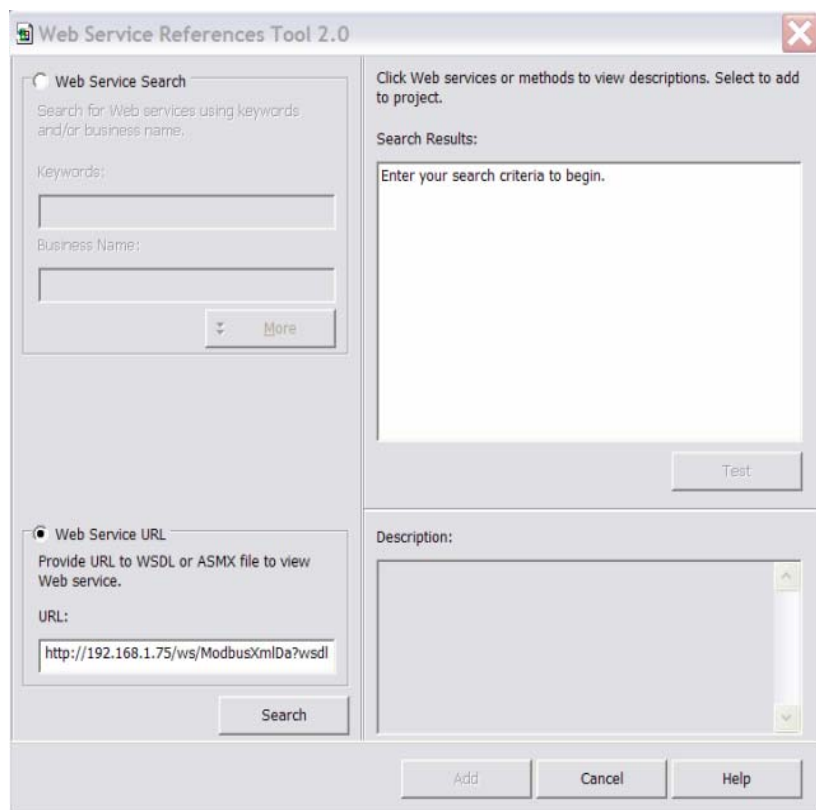
- In VBA comparirà una nuova voce nel menu degli Strumenti
 - Web Services References



- Selezionare Web Services References dal menu Strumenti in VBA

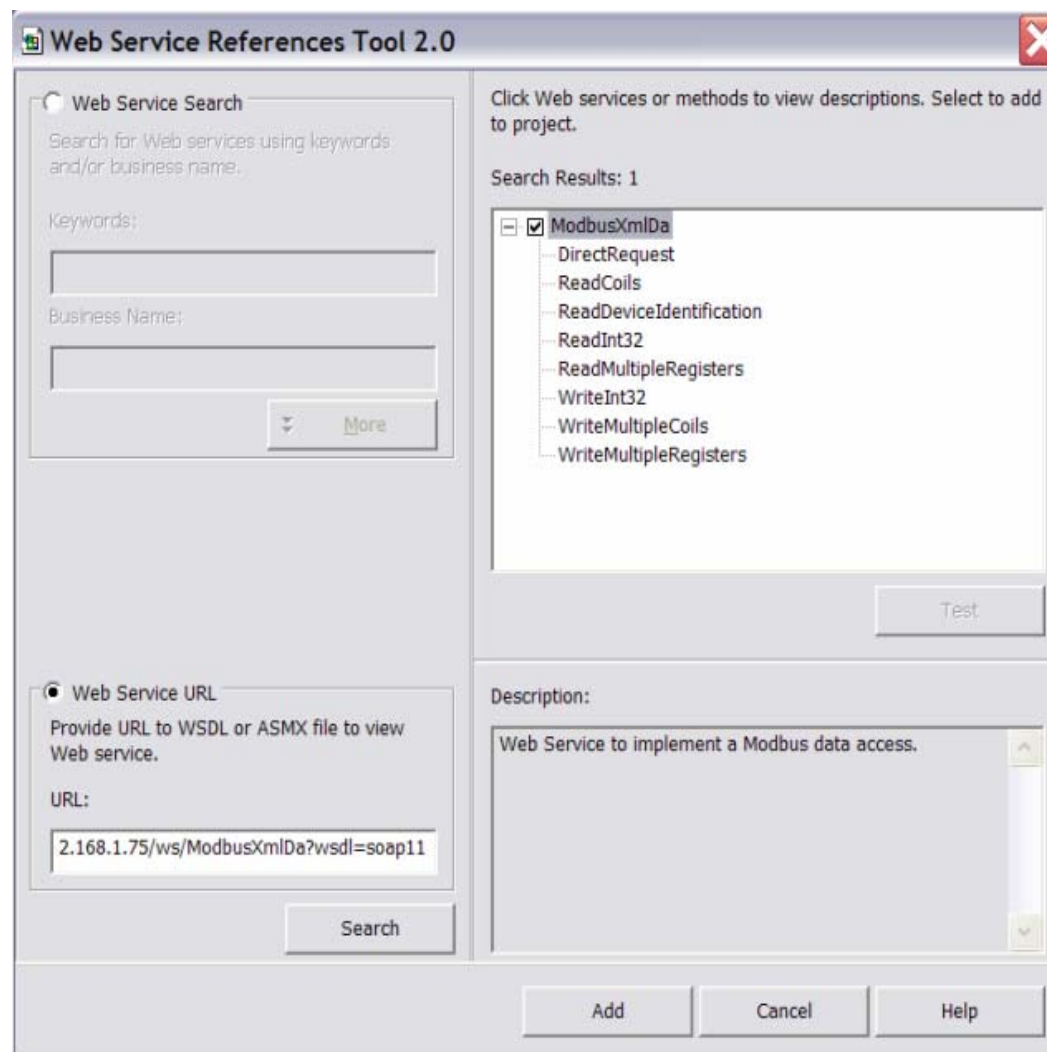
3.2 - IMPLEMENTAZIONE: Hmi - EXCEL

- Selezionare l'URL del Web Services e premere Invio
 - <http://192.168.1.75/ws/ModbusXmlDa?wsdl=soap11>
 - Dove l'IP è l'IP del modulo ETY/NOE, poi fare click su cerca



3.2 - IMPLEMENTAZIONE: Hmi - EXCEL

- Selezionare il box ModbusXmlDa per selezionare il Web Service e poi cliccare su Add



3.2 - IMPLEMENTAZIONE: Hmi - EXCEL

- Il 'Modulo Classe' verrà aggiunta al progetto

The screenshot shows the Microsoft Visual Basic IDE with the following components:

- Project Explorer:** Shows a project named 'VBAProject (Book1)' containing several sheets (Sheet1 to Sheet9) and a 'Class Modules' folder with 'clsws_ModbusXmlDa' selected.
- Properties Window:** Shows the 'General' tab for the 'wsm_DirectRequest' property page. The 'Name' is 'clsws_ModbusXmlDa' and it is a 'Private' class module.
- Code Editor:** Contains the following VBA code:


```
Option Explicit

'*****
'This class was created by the Web Service References Tool 2.0.
'
'Created: 11/4/2005 04:26:39 PM
'
'Description:
'This class is a Visual Basic for Applications class representation of the Web service
'as defined by http://192.168.1.75/ws/ModbusXmlDa?wsdl=soap11.
'
'To Use:
'Dimension a variable as new clsws_ModbusXmlDa, and then write code to
'use the methods provided by the class.
'Example:
' Dim ExampleVar as New clsws_ModbusXmlDa
' debug.print ExampleVar.wsm_ReadDeviceIdentification("Sample Input")
'
'For more information, see Complex Types in Web Service References
'Tool 2.0 Help.
'
'Changes to the code in this class may result in incorrect behavior.
'
'*****

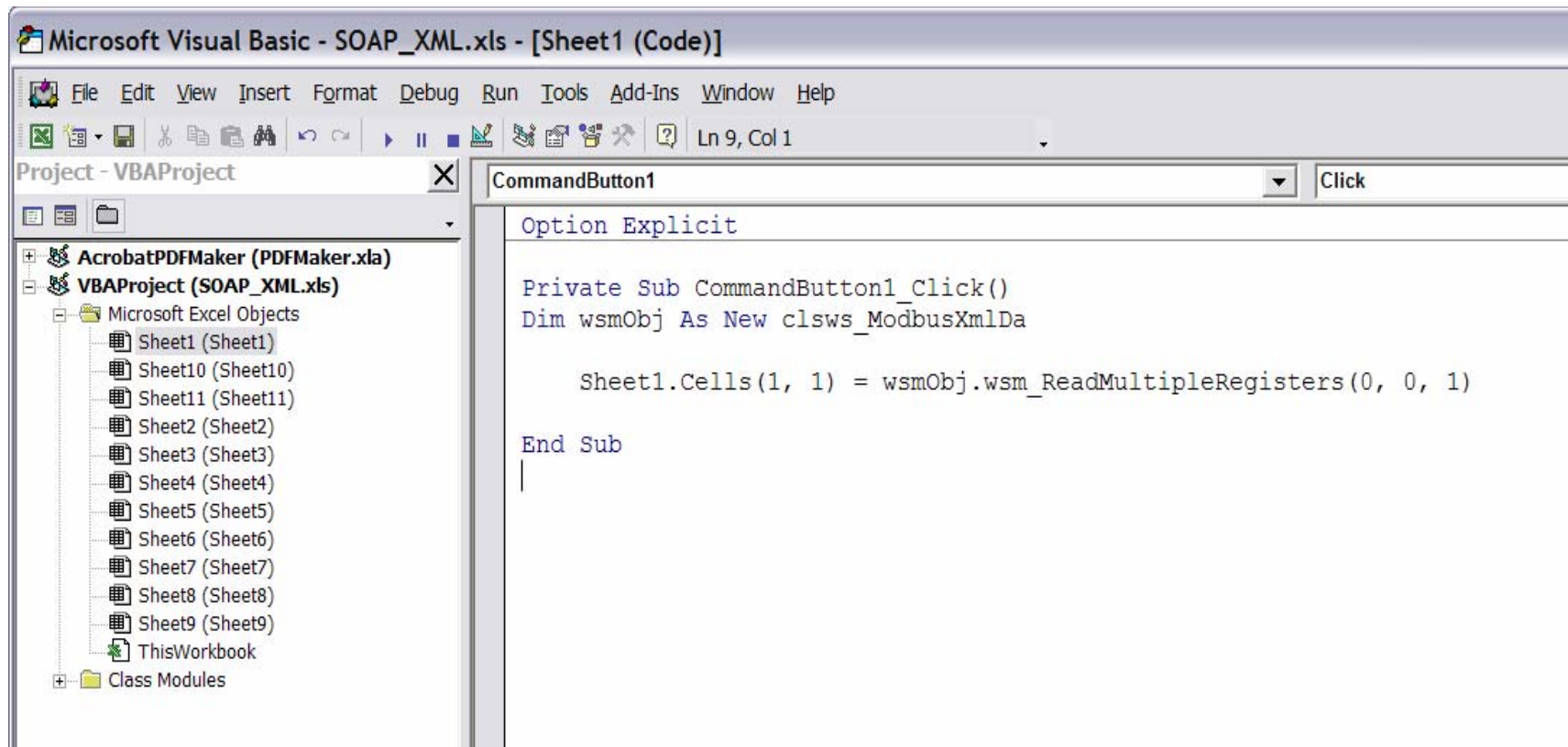
'Dimensioning private class variables.
Private sc_ModbusXmlDa As SoapClient30
Private Const c_WSDL_URL As String = "http://192.168.1.75/ws/ModbusXmlDa?wsdl=soap11"
Private Const c_SERVICE As String = "ModbusXmlDa"
Private Const c_PORT As String = "ModbusXmlDaSoap"
Private Const c_SERVICE_NAMESPACE As String = "http://www.schneider-electric.com/ws/ModbusXmlDa/"

Private Sub Class_Initialize()
'*****
'This subroutine will be called each time the class is instantiated.
'Creates sc_ComplexTypes as new SoapClient30, and then
'initializes sc_ComplexTypes.msoapinit2 with WSDL file found in
'http://192.168.1.75/ws/ModbusXmlDa?wsdl=soap11.
'*****

Dim str_WSML As String
str_WSML = "<servicemapping>"
str_WSML = str_WSML & "<service name='ModbusXmlDa'>"
str_WSML = str_WSML & "<using PROgid='MSSOAP.GenericCustomTypeMapper30' cachable='0' ID='GCTM' />"
str_WSML = str_WSML & "<types>"
str_WSML = str_WSML & "<type name='DeviceIdentification' targetNamespace='http://www.schneider-electric.com/ws/ModbusXmlDa/'>"
```

3.2 - IMPLEMENTAZIONE: Hmi - EXCEL

■ Codice base per leggere una Word



The screenshot shows the Microsoft Visual Basic editor for a VBA project named 'SOAP_XML.xls'. The interface includes a menu bar (File, Edit, View, Insert, Format, Debug, Run, Tools, Add-Ins, Window, Help), a toolbar, and a Project Explorer on the left. The Project Explorer shows the 'VBAPROJECT (SOAP_XML.xls)' containing 'Microsoft Excel Objects' with sheets Sheet1 through Sheet9 and 'ThisWorkbook', and 'Class Modules'. The main window displays the code for the 'Click' event of 'CommandButton1'.

```
Option Explicit

Private Sub CommandButton1_Click()
    Dim wsmObj As New clsWS_ModbusXmlDa

    Sheet1.Cells(1, 1) = wsmObj.wsm_ReadMultipleRegisters(0, 0, 1)
End Sub
```

3.2 - IMPLEMENTAZIONE: Hmi – Internet Explorer

- Collegarsi con il configuratore FactoryCast al modulo Ethernet
- Selezionare la cartella wwwroot/unsecure/user
- Creare una cartella 'demo'
- Copiare I file allgati nel file allegato [ws.zip](#)
- Collegarsi con InternetExplorer alla pagina scaricata
-provare.....

3.2 - IMPLEMENTAZIONE: Hmi - Riepilogo

■ Esempi applicativi a scopo didattico:

- [Applicazione per Excel 97](#)
- [Applicazione in VisualBasic .NET](#)
- [Applicazione HTML per InternetExplorer](#)

3.3 - IMPLEMENTAZIONE: Plc

3.4 - IMPLEMENTAZIONE: Device

4 - COMPONENTI

Oggetto	Utilizzo delle funzionalità SOAP nei moduli FactoryCast V3.1 in Excel, Internet Explorer e VBasic	
Materiali	Codice	Versione
	TSXETY5103	V:3.3
	WEB SERVICE	V:2.0 (Attenzione alla lingua di installazione!)
	SOAP TOOLKIT 3.0	V:3.0
	Microsoft Excel 2000	V:9.0.3821 SR1
	VisualBasic .NET	V:2005 .NetFramework V:2.0.50727
	Internet Explorer	V:6.0 SP2
Link	Sito Microsoft per Download	http://www.microsoft.com/downloads
	Specifiche SOAP	http://www.w3.org/TR/soap/
Varie	Versione di questo documento	V:1.0
	Redatto da:	Pronto Contatto: +390112281203
	Ultima Revisione	Data: 17 Feb 2006