



## .NET Telephony

## Product Introduction

Voice Elements is a software tool that brings telephony to the wider development community of Microsoft .NET. The tool enables .NET developers to develop IVRs, conference bridges, dialers, call centers, gateways or any telephony application. Supporting all .NET languages, such as VB.NET and C#, Voice Elements includes pre-made voice application modules, sample code tutorials, and reporting features such as call monitoring and logging.

The standard Voice Elements toolkit opens up the .NET architecture for telephony. A typical Visual Studio developer can easily learn the Voice Elements classes and create voice applications. Connection to all of the voice resources that a .NET application requires can be accomplished with the Voice Elements Server or the Telephony Bank.

## Deployment Options

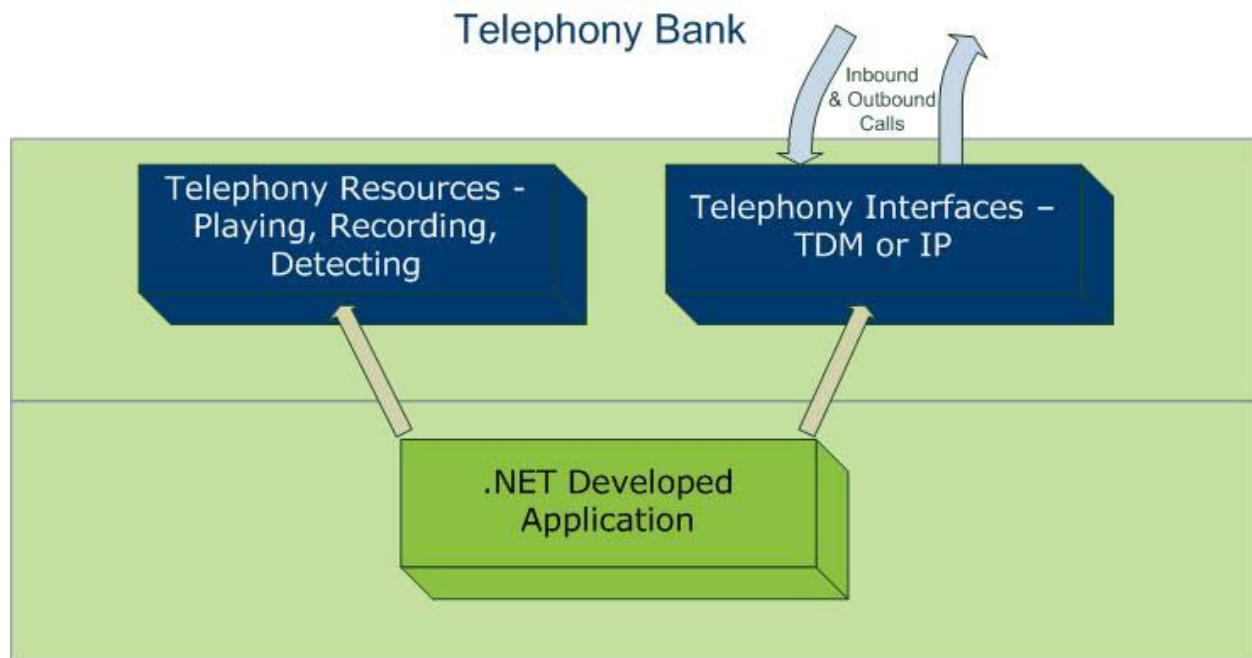
### Voice Elements Server (Option #1)

Voice Elements Server allows developers to use their own servers for telephony resources or distribute applications for deployment on server(s) at customers' sites. Solutions built with Voice Elements can run on unlimited ports on a single server for a one-time fee. This option enables full control over deployed systems and software.



### Telephony Bank (Option #2)

The Inventive Labs Telephony Bank is a group of resource servers that allow a Voice Elements-based application to run at a customer site with internet access (using the resources remotely) or the complete hosting of the voice solution at the Inventive Labs' location. The Telephony Bank is a simple monthly service based on the resource usage of an application. This service not only provides an easier route to deployment, but also provides piece of mind that all equipment, servers, etc. are fully supported and maintained.

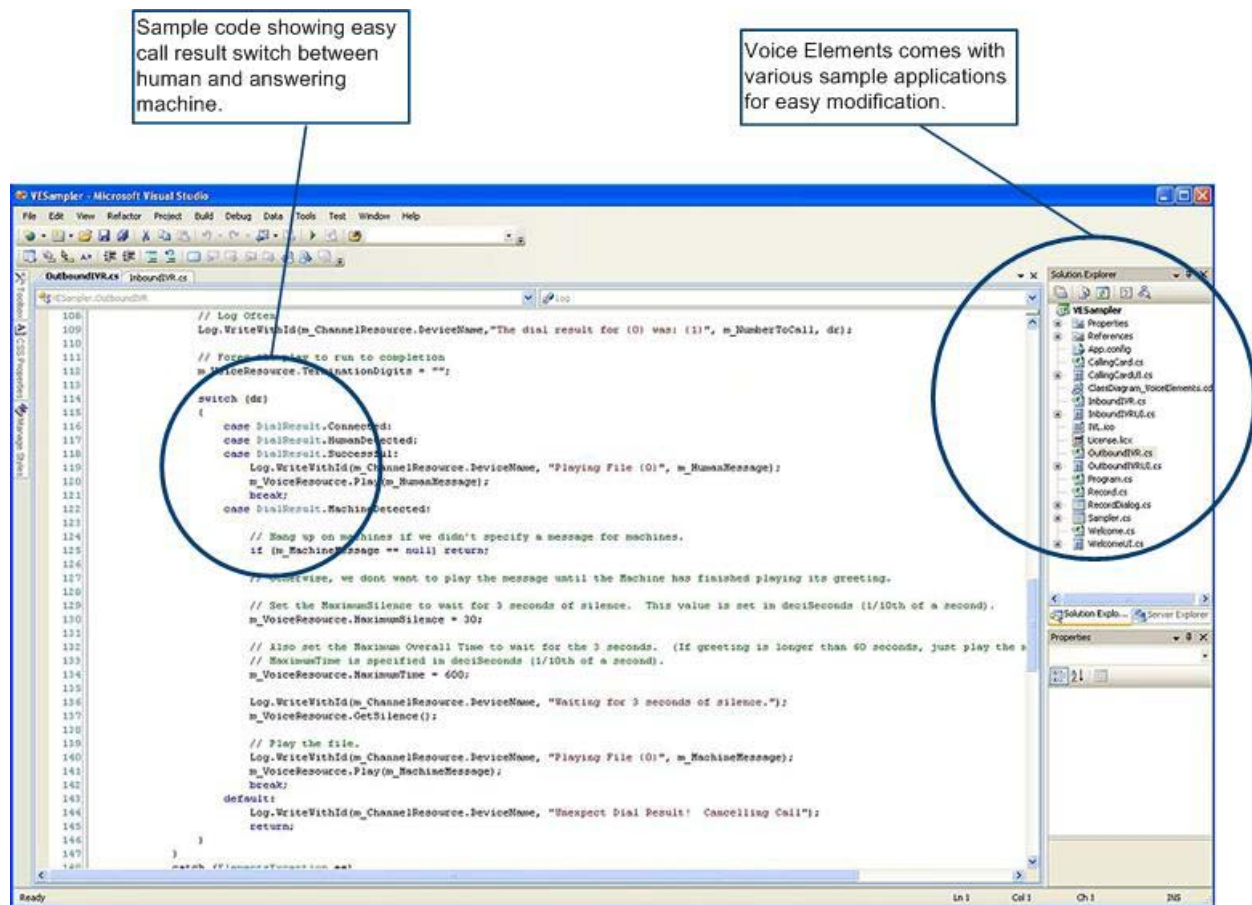


### Supported Technologies

In addition to the Microsoft .NET Framework, Voice Elements supports speech recognition, text-to-speech, faxing, Dialogic® TDM and VoIP Hardware, and Dialogic® HMP software. A list of supported Dialogic Interfaces is included at the end of this document.

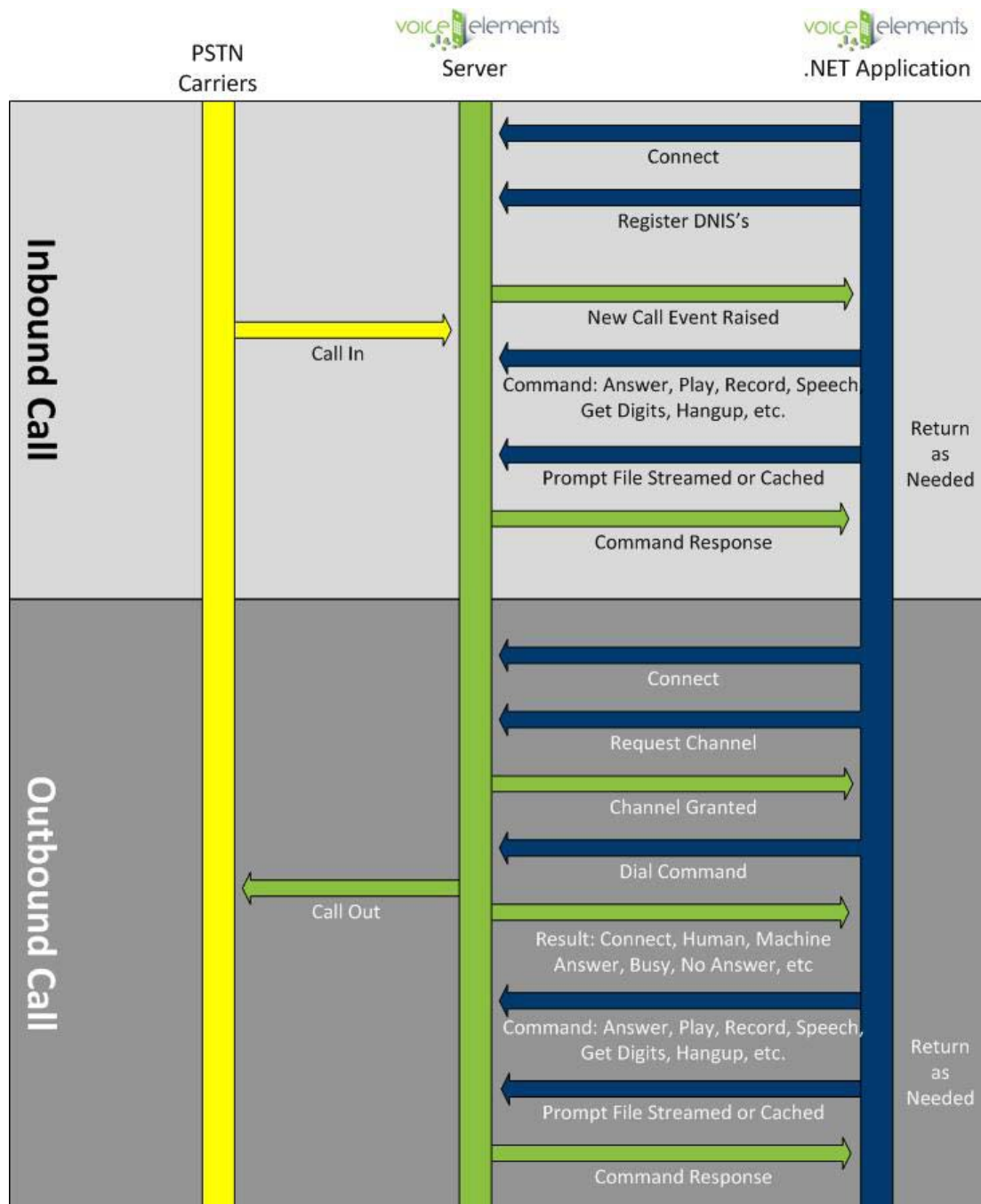
## Code Example

Voice Elements comes equipped with a sampler pack which includes sample applications, such as an inbound and outbound IVR and calling card solution. It also includes a skeleton application for quickly starting your specific application. These can be used as base applications to easily modify by switching out prompts and call flow. In addition, the code example below illustrates how easy it is within Visual Studio to switch actions based on the detection of an answering machine or a human.



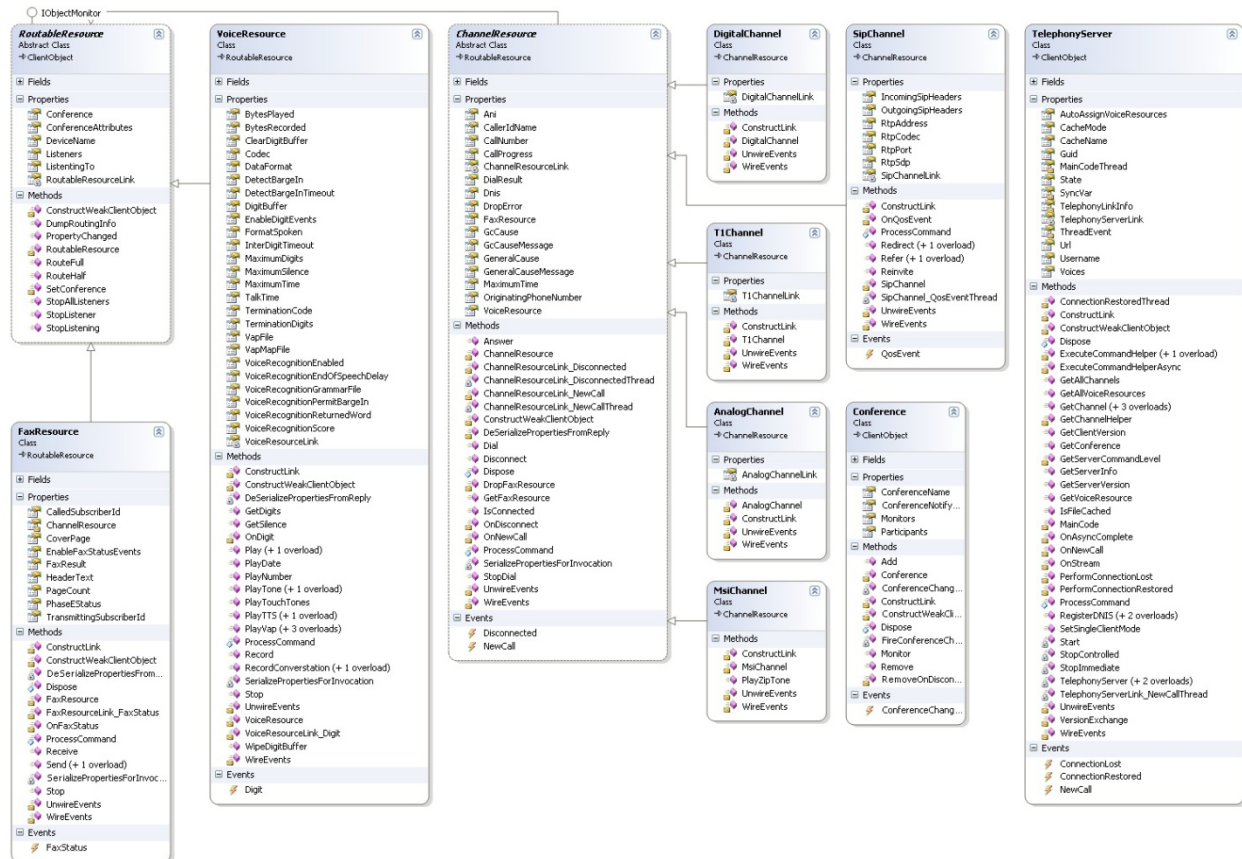
The next diagram illustrates a typical call flow sequence between Voice Elements Server and a .NET application. The .NET application provides commands such as answer, play, record and "get digits", while the Voice Elements Server gathers and passes data from the caller interaction, such as busy, no answer, and other results to the application.

## Call Flow Sequence



## Voice Elements in Detail

For those of you already familiar with programming in .NET, check out the simplistic class diagram:



Simply add these references to your project: VoiceElementsClient, VoiceElementsCommon, and VoiceElementsInterface. Add license.licx to your project. You are ready to go.

It all starts with a statement similar to this:

```
myTelephonyServer = new TelephonyServer(sIpaddress, "myusername",
"mypassword");
```

For an inbound application, do this:

```
//Method to call on new phone call
myTelephonyServer.NewCall += new
VoiceElements.Client.NewCall(myTelephonyServer_NewCall);
and
```

```
//Point all Phone numbers assigned to you to this application
myTelephonyServer.RegisterDNIS();
```



When new call comes in you get a ChannelResource and within the ChannelResource you will find a voice resource. For example you could do the following:

```
myChannelResource.Answer();  
myVoiceResource.Play("Welcome.wav");
```

For an outbound application, do this:

```
myOutboundChannel = myTelephonyServer.GetChannel();  
DialResult dr = myOutboundChannel.Dial(sNumberToDial);
```

## Dialogic Interfaces

The Voice Elements native Microsoft .NET telephony toolkit supports the following Dialogic hardware and software interfaces for both IP and TDM deployments.

## Currently Available Boards and Interfaces



**HMP:** This includes media, call control (SIP only) and FAX

### All DNI HMP Interface Boards

**All J-Series Boards:** This includes FAX and speech support.

**All DM/V & DM/F boards:** This includes fax, conference and speech.

**D/4PCIU4S and D/4PCIUF:** Analog interface boards.

**D42JCT / D82JCT:** PBX integration boards.

**All DISI and HDSI:** Station and interface boards.

**DI/0408-LS:** Combination switching boards.

### **Discontinued (but Supported)**

**D/240SC-T1 (also D/480SC-2T1)**

**All MSI Station Boards**

**DCB Conferencing Boards**

**D/41E Analog Boards**

## **Key Voice Elements Benefits**

### **Built on Dialogic**

#### **Time to Market**

Application is developed first - before the need to install drivers, boards, software or telephone circuits

“Show the boss” - Prove application before investing in the telephony resources.

#### **Provides for “telephony in the cloud” (i.e. “cloud computing”)**

Applications and telephony resources are separate

Ease of development (do it anywhere)

Ease of debugging (step through your voice application using the Visual Studio debugger)

No restarting of services to implement new versions

#### **Targets Enterprises using Visual Studio and .NET**

Ease of integration into existing business systems and rules

#### **Easy transition from TDM to VoIP (SIP)**

Deploy using our pre-built “Telephony Bank” or install your own server – your choice

Switching from “Telephony Bank” to your own Voice Elements server requires no code changes

#### **NO PORT ROYALTIES – sold by the server**

Design and implement multiple applications without interfering with currently deployed applications

#### **Bring state of the art .NET development environment to telephony**

Intellisense

Get all the benefits of developing in Visual Studio .NET



## Conclusion

Voice Elements is an enabler, allowing .NET-skilled programmers to use the environment most familiar to them to build telephony solutions. Potential telephony projects can be realized at a much faster pace than previous toolkits. Lastly, the various deployment options, with Voice Elements Server and the Inventive Labs Telephony Bank, provide inexpensive and quick opportunities to market.

## About Inventive Labs

Inventive Labs is a leading provider of royalty-free telephony application solutions. The company's software products and related services, including Voice Elements, dramatically reduce the time, cost and complexity of creating voice solutions. By rapidly developing applications in native .NET, enterprises and service providers can significantly reduce operating costs and quickly respond to new revenue opportunities. Inventive Labs is headquartered in Greenwood Village, Colorado.

[www.voiceelements.com](http://www.voiceelements.com)

(866) 923-5290

[sales@inventivelabs.com](mailto:sales@inventivelabs.com)

**Inventive Labs Corporation  
4955 E. Preserve Court  
Greenwood Village, CO 80121**