



**Especificaciones para la utilización de
la voz en FlexBrowser.
El ActiveX voiXtreme.**

wfb_voice_es.doc

20080226 Versión 1.0 ES	
-------------------------	--

Introducción

Resumen.

Este documento presenta el modo de utilización del sistema de reconocimiento de la voz (*ASR – Automatic Speech Recognition*) y de la síntesis del habla (*TTS – Text to Speech*), en una aplicación web que va a ser lanzada a través de *FlexBrowser*.

El mismo está orientado al programador de aplicaciones web, presentando al mismo tiempo: una introducción, la sintaxis para el conjunto de funciones y ejemplos.

Presentación del sistema

Denominaremos “*FlexBrowser Voice*” o simplemente “*Voice*” al ensamble de dos poderosos sistemas relacionados con el habla y los sistemas electrónicos: *ASR (Automatic Speech Recognition)*, *TTS (Text to Speech)*, junto al navegador *FlexBrowser*.

Básicamente, *ASR* permite escuchar una determinada alocución, procesarla sobre la base de una gramática predeterminada y devolver la cadena de texto -perteneciente a esa gramática- más cercana a la alocución pronunciada.

TTS realiza el proceso inverso. Hace las veces de “lector” de texto, de modo de poder escuchar un determinado texto que viene dado por una cadena de caracteres.

En ambos casos, todo el procesamiento se hace en la terminal sin necesidad de conexión alguna con el servidor.

FlexBrowser utiliza ambos sistemas a través de un control ActiveX propio llamado *voiXtreme*, cuyas características se presentan aquí. Se supone que para que una página HTML (con javascript) pueda utilizar *Voice*, debe seguir las especificaciones presentadas.

voiXtreme

voiXtreme.ocx es el nombre del ActiveX que se instala independientemente de *FlexBrowser* y que permite la utilización del *Voice*. Hay una versión diferente para cada lenguaje y en cada caso, se instala el ocx junto al conjunto de datos y librerías necesarias para su funcionamiento.

Debido al tamaño de los componentes de *voiXtreme* y al hecho de que los sistemas Windows Mobile 2005 contienen su sistema de archivos en memoria persistente, el *Voice* se instala como resistente a warmboot, coldboot, pero no a cleanboot.

Funciones definidas, parámetros, valores devueltos y sintaxis

El objeto a instanciar tiene el CLSID `5FF0CA2F-A640-4F62-8EDF-C97F30B562FB`, y expone los siguientes métodos:

```
int StgVoiceInit( ) ;
```

Inicializa la librería del *Voice* (ASR+TTS). Es necesario llamar a esta función antes de una llamada a cualquier otra función. Una vez inicializada, resta activa para todas las páginas, no es necesario reinicializarla.

```
int StgVoiceDeInit( ) ;
```

Contraparte de `StgVoiceInit`. Se debe llamar a esta función cuando no se va a utilizar más el sistema de la *Voice*.

```
int StgVoiceSay( strToSay ) ;
```

Lee la cadena de caracteres pasada por parámetro y devuelve un código de error, 0 si ok.

```
int StgVoiceListen( ) ;
```

Lanza el reconocedor de la voz del usuario. La función es asincrónica, en el sentido que el sistema devuelve inmediatamente el control, permitiendo seguir con la ejecución de las sentencias sucesivas. Es deber del usuario esperar el final del reconocimiento testeando la función `StgVoiceIsListening`, hasta que devuelva `false`, momento en el que la cadena resultante se puede obtener mediante la función `StgVoiceGetString`.

```
str StgVoiceListenSync( ) ;
```

Devuelve una cadena de caracteres, producto del reconocimiento de la voz del usuario. La función es sincrónica, en el sentido que el sistema queda esperando el resultado de esta función para seguir con la ejecución de las sentencias sucesivas.

```
int StgVoiceActivateRule( strRule ) ;
```

Activa una regla de la gramática. Esto *permite* que la función `Listen` pueda devolver una cadena de caracteres generada con esta regla.

```
int StgVoiceDeactivateRule( strRule ) ;
```

Desactiva una regla de la gramática. Esto *impide* que la función `Listen` pueda devolver una cadena de caracteres generada con esta regla.

```
int StgVoiceSetDebuggingMode(ULONG p_bShowMessageBoxes) ;
```

Permite la activación del modo de debugging. En el modo debugging, cada error de ejecución lanza un mensaje de texto informando del error. Por defecto está desactivado.

```
int StgVoiceIsSpeaking( ) ;
```

Devuelve un valor booleano indicando si el TTS está activo pronunciando un texto.

```
str StgVoiceGetString( ) ;
```

Devuelve la última cadena de texto reconocida por el ASR.

```
int StgVoiceIsListening( ) ;  
Devuelve un valor booleano indicando si el ASR está activo recibiendo  
input del sistema de audio.
```

```
int StgVoiceSpell( strToSpell ) ;  
Deletrea la cadena de caracteres numérica pasada por parámetro y  
devuelve un código de error, 0 si ok.
```

```
int StgVoiceStopListening( ) ;  
Finaliza abruptamente el reconocedor de la voz sin devolver un  
resultado.
```

```
int StgVoiceStopSpeaking( ) ;  
Finaliza abruptamente la pronunciación de una frase.
```

Los parámetros del sistema vocal están preconfigurados con opciones por defecto. Los métodos y propiedades que permiten una configuración más detallada se irán agregando en sucesivas versiones.

Un ejemplo de utilización

En el siguiente ejemplo instanciaremos un objeto como VoiceObj a través del cual se accederá al sistema vocal.

Archivo voixtremel.htm

```
<HTML>
<HEAD>
<TITLE>Test page for Voice Object voiXtreme in FlexTools</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<META NAME="MobileOptimized" content="240">
<SCRIPT LANGUAGE="javascript">
  function Init()
  {
    VoiceObj.StgVoicedInit() ;
    document.forms[0][0].disabled = true ;
    document.forms[0][1].disabled = false ;
  }
  function DeInit()
  {
    VoiceObj.StgVoiceDeInit() ;
    document.forms[0][0].disabled = false ;
    document.forms[0][1].disabled = true ;
  }
</SCRIPT>
</HEAD>

<BODY>
<OBJECT ID="VoiceObj" CLASSID="CLSID:5FF0CA2F-A640-4F62-8EDF-C97F30B562FB">
  <PARAM NAME="_Version" VALUE="65536">
</OBJECT>
<P>Demo Voice with FlexTools - voiXtreme</P>
<FORM id="form1">
  <P><INPUT type="button" id="init" name="Init" onclick="javascript:Init();"
value="Init"></P>
  <P><INPUT type="button" id="deinit" name="DeInit"
onclick="javascript:DeInit();" value="DeInit"></P>
</FORM>
<A HREF="voixtreme2.htm">Go</A>
</BODY>
</HTML>
```

Archivo voixtreme2.htm

```
<HTML>
<HEAD>
<TITLE>Test page for Voice Object voiXtreme in FlexTools</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<META NAME="MobileOptimized" content="240">
<SCRIPT LANGUAGE="javascript">
    function OnSubmit()
    {
        VoiceObj.StgVoiceSpellSync( document.forms[0][0].value );
        VoiceObj.StgVoiceSaySync( "¿Aceptar, anular o terminar" );

        VoiceObj.StgVoiceActivateRule("<commande>") ;
        VoiceObj.StgVoiceDeactivateRule("<digits>") ;
        VoiceObj.StgVoiceListenSync() ;
        commandsaid = VoiceObj.StgVoiceGetResultString() ;
        if ( commandsaid == "acceptar" )
        {
            document.forms[0].submit();
        }
        else
        {
            document.forms[0][0].value = VoiceObj.StgVoiceGetResultString() ;
            if ( commandsaid != "terminar" )
            {
                Listen() ;
            }
            else
            {
                VoiceObj.StgVoiceSaySync( "Gracias y hasta pronto." ) ;
            }
        }
    }
    function Listen()
    {
        VoiceObj.StgVoiceSaySync("cantidad") ;
        VoiceObj.StgVoiceActivateRule("<digits>") ;
        VoiceObj.StgVoiceDeactivateRule("<commande>") ;
        VoiceObj.StgVoiceListenSync( ) ;
        document.forms[0][0].value = VoiceObj.StgVoiceGetResultString() ;
        OnSubmit() ;
    }
</SCRIPT>
</HEAD>

<BODY onload="javascript:Listen();" >
<OBJECT ID="VoiceObj" CLASSID="CLSID:5FF0CA2F-A640-4F62-8EDF-C97F30B562FB">
    <PARAM NAME="_Version" VALUE="65536">
</OBJECT>
    <P>Demo Voice with FlexTools</P>
    <FORM id="form1">
    <P>Quantity said: <INPUT type="text" id="qty" name="qty"
action="voixtreme2.htm"></P>
    <P><INPUT type="submit" onclick="javascript:OnSubmit();" value="Enviar"></P>
    </FORM>
    <A HREF="voixtremel.htm">Return</A>
</BODY>
</HTML>
```

Se supone que el programador conoce la gramática instalada, que para este ejemplo es la siguiente:

```
!grammar Digit;
!start <Speech>;

<Speech>: !repeat(<digits>,1,*) | <commande> ;

<digits>: 0 !id(0) |
          1 !id(1) |
          2 !id(2) |
          3 !id(3) |
          4 !id(4) |
          5 !id(5) |
          6 !id(6) |
          7 !id(7) |
          8 !id(8) |
          9 !id(9) ;

<commande> : aceptar |
            anular   |
            comenzar |
            terminar ;
```