.mute Stummschaltung für Audio bzw. Video mit Ton (true für stumm, false für nicht stumm)
.playCount Anzahl der Wiedergaben (nicht der Wiederholungen) ab 1
.rate Faktor für Wiedergaberate laut Media-Datei
Veränderung der Wiedergabe:
reale Wiedergaberate = Wiedergaberate laut Medium * Faktor
Floating point
immer 1 bei Audio
< 0 und > 0 bei Medium mit Datenstrom bei ASF oder WMV
< 0 für Rückwärtswiedergabe
> 0 für Vorwärtswiedergabe
> 0 und bei Medium, das nicht Audio und nicht ASF und nicht WMV
< 1 so geringere Rate als laut Medium
> 1 so höhere Rate als laut Medium
Standard ist 1.0 (Rate laut Medium)
aber bei ID_Player.controls.fastForward 5.0
ID_Player.controls.fastReverse -5.0
.volume Lautstärke für Ton
Achtung: Verändert den Regler **zur Media-Datei-Art** in der Windows-Lautstärke-Regelung und nicht den Master-Regler !
Bps.: Wiedergabe einer MID-Datei und Volume-Veränderung, so Regler der MID-Datei verändert
Unbedingt pro Mediumart vor der Volume-Veränderung den aktuellen Wert von Volume per Script sichern und nach der Wiedergabe rückspeichern !
Integer
0 bis 100
0 für stumm
100 für maximal
Standard laut aktueller Windows Lautstärke-Regelung zum Medium-Typ

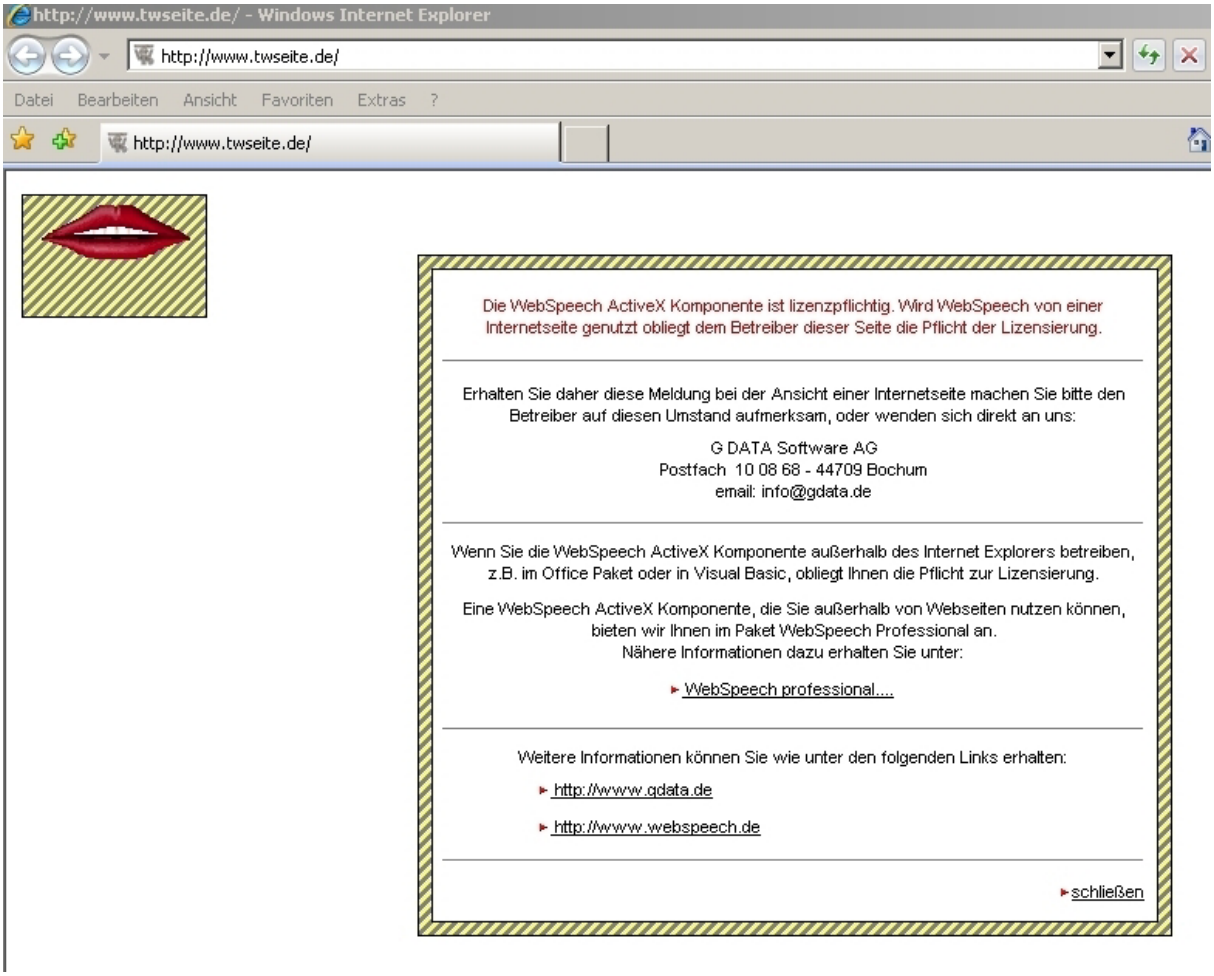
Methoden:
.getMode() Modus der Wiedergabe von Tracks (Modus des Players) ermitteln (zufällig, endlos)
siehe Event ID_Player.modeChange
.isAvailable() Veränderbarkeit von ID_Player.settings.rate prüfen
.setMode() Modus der Wiedergabe von Tracks (Modus des Players) setzen (zufällig, endlos)
siehe Event ID_Player.modeChange

5.3. Webspeech von Logox im Internet Explorer und Netscape

Die Sprachausgabe in einer Webseite ist mit der Software Logox WebSpeech auf Basis der Windows Speech Technology des Unternehmens GDATA möglich (www.gdata.de oder www.logox.de oder www.webspeech.de). Diese Software ist z.T. Freeware und vorrangig auf den Internet Explorer spezialisiert. Es können sprachanimierte Webseiten z..B. behinderten-gerecht erzeugt werden.

Es gibt verschiedene Versionen von Webspeech, die mit dem jeweiligen Plugin bzw. AktiveX.-Control im Browser installiert sein muss. Der Hersteller bietet das Plugin bzw. Active-X-Control für die aktuelle Version von Webspeech an. Ab IE 6.x wird kein Plugin mehr unterstützt. Es muss das Active-X-Control installiert werden. Höhere Versionen sind bezüglich der Steuertags z.B. für Sprachsteuerung nicht kompatibel.

Ab Version 4 wird die Benutzung von Webspeech in einer HTML-Seite auf einem Server (Domain) lizenzpflichtig, wobei die Kosten einer Lizenz
(pro Domain 1 Lizenz) beim Hersteller zu erfragen sind und eventuell mit dem Kauf der Software Logox abgedeckt sind, wenn die Software beim Hersteller registriert wurde und danach auf User-Anfrage nach Ausstellung eines Lizenzschlüssels der Hersteller für die in der Anfrage genannte Domain einen Schlüssel erzeugt hat, der den Domain-Namen enthält. **Für gewerbliche Zwecke sieht der Hersteller andere Regelungen vor, die beim Hersteller zu erfragen sind.** Nur mit der Domainlizenz sind Daten einer Webseite im Internet (auf Webserver) durch den Besucher der Webseite mit seinem Computer (Client) per Active-X-Control des Internet Explorers als Sprache hörbar. (Für lokale Webseiten ohne HTTP-Server bzw. ohne Virtual Host eines lokalen HTTP-Server ist die Domain-Lizenz unerheblich.) Ist der Lizenzschlüssel im HTML-Script bzw. JScript nicht kodiert aber die Webseite auf HTTP-Server gehostet, dann erscheint eine Fehlermeldung - hier ein Beispiel für die Domain www.twseite.de unter dem IE 7:



Wie diese Abrage funktioniert, konnte der technische Telefon-Support (ca 5 Cent pro Minute) nicht sagen, außer, dass die Schlüssel keine Dongel für Zwangskontakt zu Servern von GData sind. Da aber ein HTTP-Server Routinen des Internetzuganges aktiviert, könnte Webspeech nach Registrierung dieses Status eine per Schlüssel im Scriptcode hinterlegte Zeichenkette per Prüfsumme vergleichen mit zulässigen Werten, wobei das Active-X-Control diesen Algorithmus kennen muss. (Es ist auch so, dass bereits Webspeech 2 einen Parameter hatte für einen Schlüssel hatte, der aber nicht benutzbar war.) - Schlaumeier denken jetzt, dass dann Webspeech 2 diese Domain-Zwangslizenz nicht kennt: Korrekt, aber Webspeech 2 läuft nicht unter Windows XP (Tonerzeugung versagt, es liegt nicht am Active-X-Control selbst).

Fatal ist übrigens:

Diese Meldung erscheint auch, wenn der kodierte Domain-Schlüssel zwar von GData gekauft aber vom Active-X-Control nicht akzeptiert wird.
Es ist also nach Erwerb des Domain-Schlüssels zu prüfen, ob dieser akzeptiert wird. Für diese Prüfung muss die Test-Webseite auf
einem HTTP-Server gehostet sein. Die Testwebseite index.html hat folgenden Inhalt:

```
<HTML>
<HEAD>
<script language="JavaScript">
<!--
// Browser und WebSpeech Plugin abfragen
//WEBSPEECH FOR NAV DETECTION (MUST BE 1ST)
var IsWebSpeech = 0;
var IsNavigator = 0;
var IsExplorer = 0;
	if(navigator.appName=="Netscape" && navigator.plugins && navigator.javaEnabled)
	{
		IsNavigator=1;

		for(i=0;i<navigator.plugins.length;i++)
		{
			if(navigator.plugins[i].description.indexOf("WebSpeech") != -1 )
			{IsWebSpeech=1;}
		}
	}
```

```
// -->
</script>
<!-- WEBSPEECH FOR IE DETECTION (MUST BE 2ND)-->
<script language="VBScript">
on error resume next
if(navigator.javaEnabled) then
IsExplorer=1
IsWebSpeech=IsObject(CreateObject("WebSpeech.WebSpeech"))
end if
</script>
</HEAD>
<BODY>
<script language=Javascript>
// dieser Zweig bestimmt, was WebSpeech tun soll, wenn es installiert ist.
        if(IsWebSpeech==1)
        {
                document.write("<OBJECT ID='WebSpeech1'\n");
                document.write("CLASSID='CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC'\n");
                document.write("WIDTH=120 HEIGHT=80>\n");
                document.write("<PARAM NAME='OPAQUE' VALUE='0'>\n");
                document.write("<PARAM NAME='AUTHKEY' VALUE=' www.xxxx.yy @aaaaa-bbbbb-ccccc-ddddd '>\n");
                                        // Domainschlüssel .www.xxxx.yy @aaaaa-bbbbb-ccccc-ddddd
                document.write("</OBJECT>\n");
        }
        else
        {
                // hier geben Sie an, was passieren soll, wenn WebSpeech nicht installiert ist
                document.write("<a href='http://www.webspeech.de/download.php3'> ");
                document.write("<img src='pix/schweigen.gif' width='120' height='80' ");
                document.write("border='0' alt='Bitte laden Sie WebSpeech!!'></a>\n");
        }

        if(IsWebSpeech==1)
        {alert(window.document.WebSpeech1!=null);
         window.document.WebSpeech1.SetText('Ich bin bereit.');
        window.document.WebSpeech1.StartSpeakingImmediate();
        }
</Script>
</BODY>
</HTML>
```

Der Vertrieb und Support wurde allerdings von GData eingestellt: Das Produkt ist nicht mehr kaufbar. Selbst wenn man es bei Ebay kauft, muss für den Internet-Einsatz eine Domain-Lizenz beschafft werden. Die Webseiten zu der Sprachausgabe offerieren immer noch das Produkt, ohne dass es je in der Produktübersicht zu finden, geschweige im Onlineshop zu finden ist. Das WebSpeech-Forum ist abgeschaltet (toter Link - Webseite nicht gefunden). Emailsupport hüllt sich auch in Ignoranz. - Die Webseiten sind nichts anderes als Vergraulen von Kunden bzw. Interessenten, dafür bieten sie noch den Download der Free-Versionen und freien Dokumentationen
an, welche für Webseiten, die nicht HTTP-Server gebunden werden, nutzbar sind. Unangenehm ist der Umstand, dass GData die Zwangs-Domain-Lizenz nicht abgeschafft hat, wenn GData das Produkt nicht mehr vertreibt und supportet: Da nur GData die Lizenz vergeben konnte, wird es kritisch bei Problemen mit dem Domainschlüssel, so dass der Kauf der Sprachausgabesoftware nicht
lohnt bei Ebay etc.. Das wird allerdings dafür sorgen, dass das Internet langsamer lebendig wird: Eine Alternative zu Webspeech ist z.B. Microsoft Text To Speech, deren Software unter Win XP bereits als Grundkomponente enthalten ist, aber - im Gegensatz zu Webspeech - nicht mit Soundblaster-kompatiblen Soundkarten (Creative Labs-Standard) funktioniert: Z.B. nicht mit der Soundkarte
Creative Labs Xtreme (Modell SB 04060 mit EAX ab Version 4).

Nachfolgende Beschreibung der Programmierung zu Webspeech in HTML und Script (Javascript und VBScript) gilt ausschliesslich für den privaten und nicht kommerziellen Gebrauch.

Die Programmierung der Versionen 2 bis unter 4 und 4 wird nachfolgend skizziert. Die wenigen Unterschiede von Webspeech 4 zu Vorgänger-Versionen werden explizit genannt.

Bei Nutzung von Webspeech auf einer Domain ist eine Verlinkung auf den Hersteller eventuell notwendig (Hersteller fragen für die Form der Verlinkung) und zwar nicht nur für den Download des Plugins bzw. Active-X-Controls zum Browsers.

Das Software-Paket Logox 4 Professional enthält alle notwendigen Regelungen zur Lizensierung und Art der Verlinkung und bietet weiteren Support, der eventuell über den Rahmen des kostenlosen Online-Supports des Herstellers hinausgeht.

Für gewerbliche Zwecke sieht der Hersteller andere Regelungen vor, die beim Hersteller zu erfragen sind.

5.3.1. Webspeech-Objekt erzeugen

Unbedingt obige Hinweise zum Domain-Schlüssel beachten !

Wegen Patentwahrung hat Microsoft ein freiwilliges Patch herausgegeben, dass bei ActiveX-Control

per APPLET, EMBED oder OBJECT, die auf dem Bildschirm rendern (mit oder ohne Userschnittstelle), dafür sorgt, dass bei mouseover über das Control eine Sprechblase erscheint, die darauf hinweist, dass das Objekt als ActiveX-Control klickbar ist.

Diese Sprechblase erscheint auch, wenn das Control keine Userschnittstelle hat, also diese gar nicht klickbar ist.

Es werden alle Events solange unterdrückt, bis der User die Sprechblase geklickt hat.

Das Klicken muss auf das Objekt im Sprechblasenrahmen erfolgen, der so groß ist, wie die Dimension, in der gerendert wurde.

Es muss also ERST per Mausklick das Control aktiviert werden, ehe das Control klickbar und damit die Eventsteuerung aktiviert ist.

Ein Control, dass programmtechnisch zwar was rendert, aber ansonsten ohne sichtbare programmtechnisch startet, muss ebenfalls geklickt werden, obwohl es bereits läuft und es nichts zu klicken gäbe (wenn keine Eventsteuerung eingebaut wurde).

DOM der Webseite wird nicht verändert: Nur Blockierung der gesamten Eventsteuerung.
Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.

Die Eventauslösung per nicht-objekteigenen Eventhandler, der für das Objekt per fireEvent() ein Event auslöst, ist solange blockiert, bis der User die Sprechblase geklickt hat.

style.visibility='hidden' WIRD IGNORIERT:
Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert: Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.

Diese Sprechblase erscheint nur DANN NICHT, wenn die Userschnittstelle mit Breite == Höhe == 0 gerendert wird. Sollte die Userschnittstelle in einem Container liegen, z.B. DIV, dann wird der Container, wenn er in der Dimension kleiner ist, also die Userschnittstelle, angepasst. Daher muss der Container ebenfalls mit Breite == Höhe == 0 gerendert werden. Wegen Dimensionierung auf 0 sollte style.visivility="hidden" sein. Im Falle eines Containers reicht es, den style des Containers zu ändern, da visibility normalerweise vererbt wird an Kinder, also auch an das Control.

Microsoft ändert fortlaufend die Active-X-Eigenschaften von Windows und somit auch des Internet Explorers

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.
Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.
Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Linzensierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks
Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.
Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.
Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2).
Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.
Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

Hinweis: Man suche doch mal im Internet nach einem kostenlosen HTTP-Server vom Microsoft, um IE-Seite testen zu können, die JScript nutzen (inklusive Debugger). Denn sollte kein kostenloses Angebot findbar sein, kommen die Kosten von Entwicklungssoftware zum IE hinzu. Ein Blick auf Konkurrenz-HTTP-Server klärt die Sachlage: Apache-HTTP-Server ist kostenlos, allerdings nicht einfach einzurichten (Hinweis: Der HTTP-Server sollte virtuelle Hosts einrichten können und korrekt mit der Firewall des Users zusammenarbeiten können).

Abänderungen wegen Sicherheitspatches der jeweiligen Windows-Versionen

Abschaltungen von Active-X-Controls erfolgen auch im Rahmen der Sicherheitspatches zu Windows-Versionen.
Es ist auch möglich, dass wegen Sicherheitslücken abgeschaltet wird und somit Komponenten einer Webseite je nach Windowsversion nicht mehr laufen.

Im Rahmen der Sicherheitspatches ist es Microsoft sogar gelungen, Webseiten, die den MS-Encoder zur Komprimierung von HTML- und JScript-Code nutzen, schlagartig unnutzbar zu machen: Ein Bug in einem Patch zu Windows XP - Q918899
Das Patch verusacht IE-Browser-Absturz bei per MS ScriptEncoder gepacktem JScript unter SP1 und 2 wenn HTTP 1.1 mit Kompression genutzt wird z.B. bei
onlick-Handler auf IMG
klick ins Fenster per aktivem Popup

Der Absturz ist "read" -Fehler von immer ein und derselben Speicherstelle.
User, die dieses Patch installiert haben, können ab sofort keine IE-Seiten mit codiertem Script mehr ansehen.
Microsoft stellt Abhilfe nach geraumer Zeit zur Verfügung, jedoch spezifisch nach Windows XP-Version:
Patch Q918899 für
Windows XP SP1Download für jedermann bereitgestellt
SP2 nur auf kostenpflichte telefonische Anfrage des Users per Downloadlink bereitgestellt, da Microsoft explizit die User registriert haben will, bei denen das Patchproblem auftritt (User muss sich Telefonnummer besorgen)
Solange also das Patch zum fehlerhatem Patch vom User nicht installiert wird, z.B. weil der User keine Ahnung hat, dass und wo er sich die Telefonnummer von Microsoft besorgen muss bzw. zu besorgen hat, wird der User IE-Seiten mit komprimierten Code dauerhaft nicht nutzen können. (Microsoft-Support ist z.T. nur in Englisch).

Abänderungen wegen Browser-Inkompatibilität

Popupblocker-Fehler

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()
Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.
Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.
Der Fehler tritt nicht auf, wenn ein Fenster per window.open() erzeugt wurde.
Bedingung:
Scripfehleranzeige ist erlaubt im IE 7
Popupblocker ist im IE abgeschaltet
ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 window.popup per .show()erzeugt.
ein weiteres Fenster (Register) z.B. leere Seite (about:blank)
beide (Register) liegen in einer gemeinsamen IE-Instanz
Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per .show() erzeugt, bricht der Browser das Dokument mit .show() ab (Scriptfehler).
Der Popupblocker für die leere Seite verursacht den Programmfehler im Dokument mit .show(). Es wird folgende Meldung angezeigt (in der Informationsleiste):
'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'
Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:
Der Popupblocker hat ein Popupfenster geblockt. Sie können den Popupblocker deaktivieren oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.
Die Realität zur obigen Meldung ist völlig anders:
Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter
Popupblocker einschalten
weitere Informationen
jedoch keine Möglichkeit wie laut Bedeutung
Damit gilt: Der abgeschaltete Popupblocker ist in Wirklichkeit aktiv.
Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster einer Windowsanwendung z.B. einer anderen IE-Instanz)
Der Popupblocker des IE bemeckert aber NUR Webseite, die das Popup erzeugt.
Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popupblocker verwaltet wird.
Der Popupblocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen Webseiten (die nicht das Popup erzeugt haben).
Der Popupblocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.
Der Popupblockerfehler verändert die Eventverwaltung:
Es werden u.a. ignoriert
onfocus
onblur
onfocusin
onfocusout
und viele andere, so dass trotz Events z.B. des Body der Popupblockerfehler entsteht.

```
// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell
window.focus();
window.document.focus();
if(document.body!=null)
{if(document.body.style!='hidden')          // wenn  hidden so focus() nicht möglich (Scriptfehler erzeugt)
 {document.body.focus();}
}
// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt
popupzeiger.show(...);
```

Hinweis: Der Popupfehler ist so elementar, dass die vielen Beta-Testphasen des IE mehr als fragwürdig erscheinen, wie die Angabe von Microsoft, dass Code neu programmiert wurde, um den IE sicherer zu machen.

focus-Methode beim IE 7

windows.focus() document.focus() und body.focus() funktionieren NICHT
zwischen Register in einem IE-Fenster
zwischen Fensters z.B. in Taskleiste

Hinweis:
.focus() setzt Element aktiv, gibt dem Element den Focus und feuert dann onfocus
.setActive() ist Teilmenge von .focus(): nur das aktiv setzen
funktioniert nicht mit allen Elementen, mit denen .focus() funktioniert

animierte Gif (mit Timer)

Animierte Gifs (mit Timer), die unter IE 6 korrekt laufen, müssen unter IE 7 im Timer nicht mehr laufen:
z.B. garnicht mehr sichtbar, oder Timer nicht verwendet.
Dann müssen animierte Gif-Bilder nach IE-Version bereitgestellt werden.

Abänderungen wegen Rechtstreitigkeiten von Microsoft mit Fremdanbietern

Ein sehr bekanntes Beispiel ist die nachträglich eingeführte Einschränkung von Active-X-Controls wegen Patentwahrung durch Microsoft, wobei für den JScript-Programmierer massive Änderungen eintreten.

Wegen Patentwahrung hat Microsoft ein zunächst freiwilliges Patch herausgegeben, dass bei ActiveX-Control per APPLET, EMBED oder OBJECT, die auf dem Bildschirm rendern (mit oder ohne Userschnittstelle), dafür sorgt, dass bei mouseover über das Control eine Sprechblase erscheint, die darauf hinweist, dass das Objekt als ActiveX-Control klickbar ist.
Diese Sprechblase erscheint auch, wenn das Control keine Userschnittstelle hat, also diese gar nicht klickbar ist.

Es wurde das Eventmodell gleichzeitig geändert:

Es werden alle Events solange unterdrückt, bis der User die Sprechblase geklickt hat.
Das Klicken muss auf das Objekt im Sprechblasenrahmen erfolgen, der so groß ist, wie die Dimension, in der gerendert wurde.
Es muss also ERST per Mausklick das Control aktiviert werden, ehe das Control klickbar und damit die Eventsteuerung aktiviert ist.
Ein Control, dass programmtechnisch zwar was rendert, aber ansonsten ohne sichtbare programmtechnisch startet, muss ebenfalls geklickt werden, obwohl es bereits läuft und es nichts zu klicken gäbe (wenn keine Eventsteuerung eingebaut wurde).
Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.
Die Eventauslösung per nicht-objekteigenen Eventhandler, der für das Objekt per fireEvent() ein Event auslöst, ist solange blockiert, bis der User die Sprechblase geklickt hat.

style.visibility='hidden' wird ignoriert

Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert:
Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.

Diese Sprechblase erscheint nur DANN NICHT, wenn die Userschnittstelle mit Breite == Höhe == 0 gerendert wird. Sollte die Userschnittstelle in einem Container liegen, z.B. DIV, dann wird der Container, wenn er in der Dimension kleiner ist, also die Userschnittstelle, angepasst. Daher muss der Container ebenfalls mit Breite == Höhe == 0 gerendert werden. Wegen Dimensionierung auf 0 sollte style.visivility="hidden" sein. Im Falle eines Containers reicht es, den style des Containers zu ändern, da visibility normalerweise vererbt wird an Kinder, also auch an das Control.

Abänderung wegen Abschaltungen

DirectX ist wegen Abschaltung von Active-X--Controls nicht mehr abwärtskompatibel:
Z.B. wurde bei Win XP SP2 Direct Animation aus DirectX schlagartig durch Abschaltung von Bibliotheken dezimiert, die es bei Win XP SP1 aber noch gibt.

Hier ein Beispiel aus dem Jahr 2004: Abschaltungen von Active-X-Controls

ActiveX-Controls und Unterstützung/Verbot 20041215

erlaubt sind noch

Tabular Data-Steuerelement{333C7BC4-460F-11D0-BC04-0080C7055A83} Das TDC (Tablular Data-Steuerelement) ermöglicht die Weiterverarbeitung von Daten, die nur im Textformatvorliegen, beispielsweise durch Darstellung in einer Tabelle oder Sortierung. Weitere Informationen:• http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp(http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp)

Microsoft Agent Control - Version 2.0 {D45FD31B-5C6E-11D1-9EC1-00C04FD7081F} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kannauch in Webseiten verwendet werden. Weitere Informationen:• http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm(http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm)

Microsoft MSChat-Steuerelement-Objekt 2.0 - 2.5 {D6526FE0-E651-11CF-99CB-00C04FD64497} Dieses Steuerelement wird von Webautoren verwendet, um text- und graphisch basierte Chatgemeinden für Echtzeitkonversationen im Web zu erstellen.

Microsoft ActiveX Upload-Steuerelement, Version 1.5 {886e7bf0-c867-11cf-b1ae-00aa00a3f2c3} Dieses Steuerelement kann auf vielerlei Art genutzt werden, um auf einfache Weise Webinhalte via Drag and Drop zuveröffentlichen. Weitere Informationen:• 230298 (http://support.microsoft.com/kb/230298/DE/) - Posting Acceptor Release Notes
• http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp (http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp)

verboten sind

Datenbindung RDS {BD96C556-65A3-11D0-983A-00C04FC29E36} {BD96C556-65A3-11D0-983A-00C04FC29E33} Die RDS (Remote Data Service) Steuerelemente ermöglichen dem Browser, client-basierte SQL Abfragen an einen Webserver zu stellen. Inzwischen wurde RDS jedoch durch neuere Standards wie SOAP abgelöst, von einer weiteren Verwendung von RDS wird daher abgeraten. Weitere Informationen:• 184375 (http://support.microsoft.com/kb/184375/DE/) - Sicherheitsaspekte bei RDS 1.5, IIS 3.0 oder 4.0 und ODBC
http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp (http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp)
http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp (http://msdn.microsoft.com/library/enus/dnmdac/html/data_mdacroadmap.asp)

XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer {550dda30-0541-11d2-9ca9-0060b0ec3d39} {CFC399AF-D876-11d0-9C10-00C04FC99C8E} {e54941b2-7756-11d1-bc2a-00c04fb925f3}{7108ECB4-AFDC-11D1-ADC1-00805FC752D8}XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer ermöglichen die Verarbeitung von XML Daten,etwa die Bindung von HTML Elementen an einen XML Datensatz, oder das Einlesen, Manipulieren, und Zurückschreiben von XML Daten.

Die Steuerelemente DOMDocument und XMLIslandPeer bzw. die dazugehörigen ClassIDs sind nicht mehr aktuell, so dass von einer generellen Freigabe dieser Steuerelementgruppe abgeraten wird. Weitere Informationen:• http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp(http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp)

Internet Explorer

Active Setup / IE Active Setup-Steuerelement {F72A7B0E-0DD8-11D1-BD6E-00AA00B92AF1}Dieses Steuerelement enthält die in Microsoft Security Bulletin MS99-037 beschriebene Sicherheitsanfälligkeit. Umeine weitere Ausführung zu verhindern wurde im Rahmen dieses Security Bulletins ein Kill-Bit gesetzt, so dass selbst bei einer Freigabe dieses Controls eine Ausführung blockiert wird. Weitere Informationen:• http://www.microsoft.com/technet/security/bulletin/ms99-037.mspx(http://www.microsoft.com/technet/security/bulletin/ms99-037.mspx)
http://www.microsoft.com/technet/security/bulletin/fq99-037.mspx (http://www.microsoft.com/technet/security/bulletin/fq99-037.mspx)
240797 (http://support.microsoft.com/kb/240797/DE/) - So verhindern Sie die Ausführung von ActiveX-Steuerelementen in Internet Explorer

Media Player / Active Movie Runtime {A4001DE0-7075-11d0-89AB-00A0C9054129} Die Funktionalität dieses Steuerelements wird wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Runtime Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wirdabgeraten.

Media Player / ActiveMovie-Steuerelement{05589FA1-C356-11CE-BF01-00AA0055595A} Die Funktionalität dieses Steuerelements wird wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Microsoft NetShow Player {2179C5D3-EBFF-11CF-B6FD-00AA00B4E220} Die Funktionalität dieses Steuerelements wird wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das NetShow Player Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Windows Media Player {22D6F312-B0F6-11D0-94AB-0080C74C7E95} Dies ist das Steuerelement für Windows Media Player version 6.4 und war Installationsbestandteil bis einschließlich Windows Media Player Version 8. Ab Windows Media Player 9 wurde diese ClassID durch die neue ClassID {6BF52A52-394A-11D3-B153-00C04F79FAA6} abgelöst, deren Verwendung stattdessen empfohlen wird. Ab Windows Media Player Version 9 wirdferner die alte ClassID anhand eines Wrappers automatisch auf die neue ClassID umgeleitet. Die ClassID für Windows Media Player Version 9 ist jedoch nicht in der Liste der vom Administrator genehmigten Steuerelemente enthalten, undmuss bei Bedarf manuell hinzugefügt werden.

Animierte Schaltflächen{0482B100-739C-11CF-A3A9-00A0C9034920} Dieses Steuerelement erlaubte in frühen Versionen des Internet Explorer die Verwendung animierter Schaltfächen auf Webseiten. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von derFreigabe des Steuerelements wird daher abgeraten.

IE Label-Steuerelement {99B42120-6EC7-11CF-A6C7-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 auch kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen:• 190045 (http://support.microsoft.com/kb/190045/DE/) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

IE Menu-Steuerelement {74701400-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ermöglicht die Handhabung von Menüstrukturen in Webseiten, wird jedoch nicht mehr unterstützt und dürfte nur noch selten Verwendung finden. Von einer Freigabe des Steuerelements wird daher abgeraten.

IE Preloader-Steuerelement {16E349E0-702C-11CF-A3A9-00A0C9034920} Dieses Steuerelement ermöglichte das Vorladen von Webseiten, ist jedoch inzwischen nicht mehr aktuell, wird nicht mehr unterstützt und dürfte nicht mehr im Einsatz sein. Aufgrund einer potentiellen Sicherheitsanfälligkeit in diesem Steuerelement wird von einer Freigabe abgeraten. Weitere Informationen:• 231452 (http://support.microsoft.com/kb/231452/DE/) - Update Available for "Legacy ActiveX Control" Issue

IE Timer-Steuerelement {59CCB4A0-727D-11CF-AC36-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements

wird daher abgeraten. Weitere Informationen:• 190045 (http://support.microsoft.com/kb/190045/DE/) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

MCSiMenü {275E2FE0-7486-11D0-89D6-00A0C90C9B67} Dieses Steuerelement dient der Anpassung von Popupmenüs, ist jedoch nicht mehr aktuell und wurde nach Windows 98 nicht mehr ausgeliefert. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten.

Popupmenüobjekt {7823A620-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen:• 190045 (http://support.microsoft.com/kb/190045/DE/) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

Microsoft Agent Control - Version 1.5 {F5BE8BD2-7DE6-11D0-91FE-00C04FD701A5} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Diese Version des Steuerelements ist jedoch nicht mehr aktuell und wird nicht mehr unterstützt. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen:• http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm (http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm)

5.3.1.1. Webspeech-Objekt in HTML erzeugen

Erzeugung in HTML:

beim IE anhand des OBJECT-Tags
beim pluginfähigem NS anhand des EMBED-Tags

zu OBJECT und EMBED:
es sind kodierbar
die üblichen Attribute wie WIDTH, HEIGHT, STYLE etc..
Empfehlung: Angaben zu WIDTH und HEIGHT immer kodieren
Eventhandler, die aber in **keinem Zusammenhang** mit den Events von Webspeech stehen, da diese per VBScript zu kodieren sind hier nicht beschrieben werden.

Beispiel:

```
<OBJECT         ID="freies_id"
                CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
                WIDTH="150"
                HEIGHT="125"
>               <! -- freies_id muss identisch sein mit der in der EMBED-Kodierung Attribut NAME //-->

        <!-- nur beim IE //-->
        <PARAM NAME="AUTHKEY'"          VALUE="lizenz_schluessel">
        <PARAM NAME="TEXT" VALUE="Ein netter Text">
        <!-- oder // -->
                <PARAM NAME="URL" VALUE=" http://www.test.de /test.txt">
        <PARAM NAME="AUTOSTART"         VALUE="0">
        <PARAM NAME="IMMEDIATE"         VALUE="0">
        <PARAM NAME="MOUTHANIMATION"    VALUE="1">
        <PARAM NAME="MOUTHCOLOR"        VALUE="black,TEXT,TEXT,TEXT,TEXT">
        <PARAM NAME="CONTROLPOSITION"   VALUE="75">
        <PARAM NAME="TEXTANIMATION"     VALUE="3">
        <PARAM NAME="TEXTCOLOR"         VALUE="black,silver">
        <PARAM NAME="TEXTPOSITION"      VALUE="0">
        <PARAM NAME="TEXTSIZE"          VALUE="12">
        <PARAM NAME="BACKGROUNDCOLOR"   VALUE="black">
        <PARAM NAME="OPAQUE"            VALUE="0">


        <!-- nur beim NS //-->
        <! -- freies_id muss identisch sein mit der in der OBJECT-Kodierung Attribut ID //-->
        <! -- PARAM_TEXT und PARAM-URL sind alternativ zu kodieren //-->
        <EMBED NAME="freies_id"
                TYPE="application/x-WebSpeech"
                WIDTH=150
                HEIGHT=125
```

```
                    PARAM_AUTHKEY="lizenz_schluessel"
                    PARAM_TEXT="Ein netter Text."
                    PARAM_URL="http://www.test.de /test.txt"
                    PARAM_AUTOSTART="0"
                    PARAM_IMMEDIATE="0"
                    PARAM_MOUTHANIMATION="1"
                    PARAM_MOUTHCOLOR="black,TEXT,TEXT,TEXT,TEXT"
                    PARAM_CONTROLPOSITION="75"
                    PARAM_TEXTANIMATION="3"
                    PARAM_TEXTCOLOR="black,silver"
                    PARAM_TEXTPOSITION="0"
                    PARAM_TEXTSIZE="12"
                    PARAM_BACKGROUNDCOLOR="black"
                    PARAM_OPAQUE="0"
          >
          </EMBED>
</OBJECT>
```

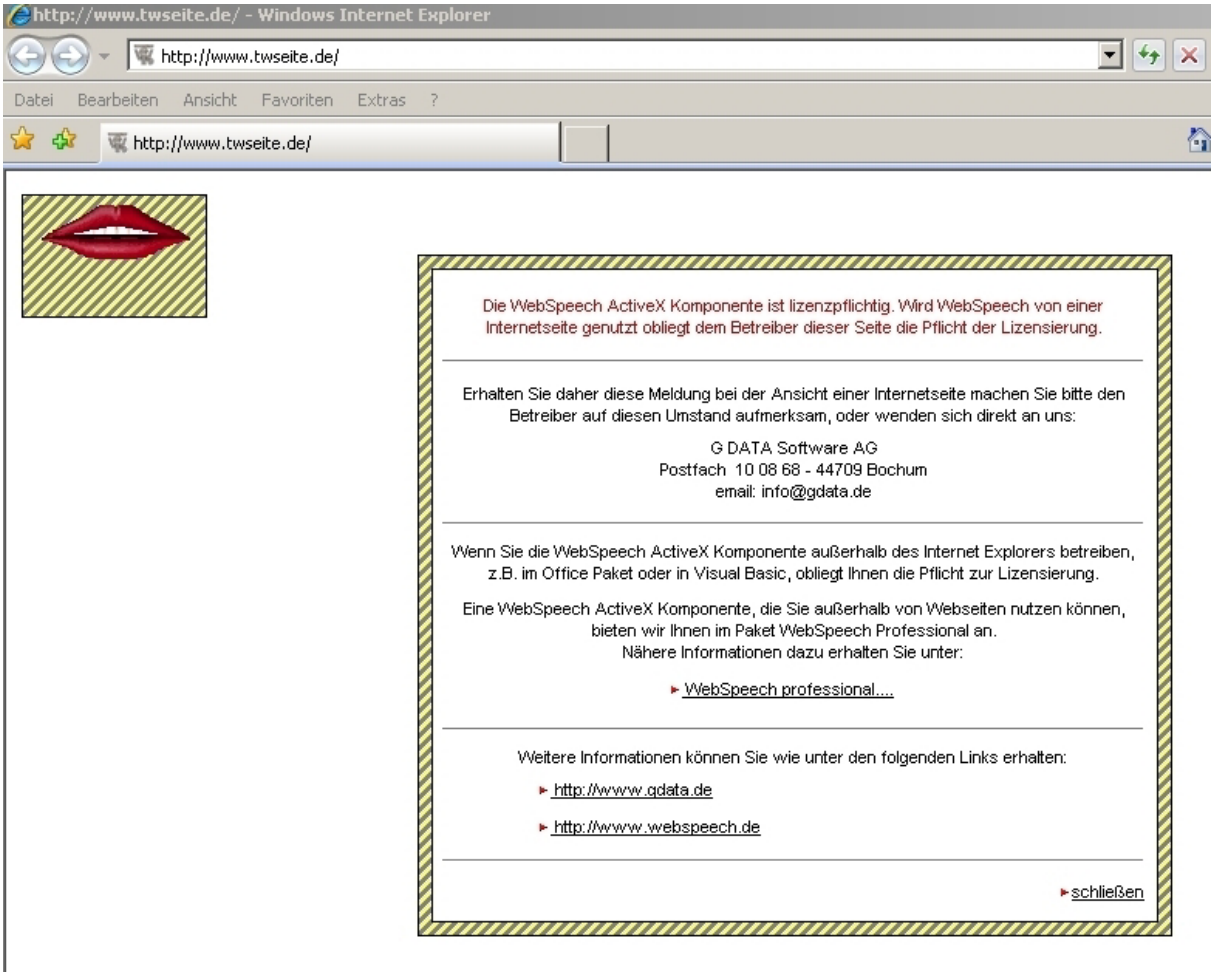
Parameter mit ungültigem Wert müssen keinen Fehler erzeugen.

5.3.1.1.1. Parameter AUTHKEY (ab Webspeech 4)

Domain-Lizenz-Schlüssel
kann nicht entfallen, da kein Standard
Solange die Webseite lokal auf dem PC getestet wird, ist keine Lizenz nötig und man kann daher den Schlüssel des Herstellers verwenden:

```
IE    <PARAM NAME="AUTHKEY'" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
NS    PARAM_AUTHKEY="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD"
```

Ab Version 4 wird die Benutzung von Webspeech in einer HTML-Seite auf einem Server (Domain) lizenzpflichtig, wobei die Kosten einer Lizenz
(pro Domain 1 Lizenz) beim Hersteller zu erfragen sind und eventuell mit dem Kauf der Software Logox abgedeckt sind, wenn die Software beim Hersteller registriert wurde und danach auf User-Anfrage nach Ausstellung eines Lizenzschlüssels der Hersteller für die in der Anfrage genannte Domain einen Schlüssel erzeugt hat, der den Domain-Namen enthält. **Für gewerbliche Zwecke sieht der Hersteller andere Regelungen vor, die beim Hersteller zu erfragen sind.** Nur mit der Domainlizenz sind Daten einer Webseite im Internet (auf Webserver) durch den Besucher der Webseite mit seinem Computer (Client) per Active-X-Control des Internet Explorers als Sprache hörbar. (Für lokale Webseiten ohne HTTP-Server bzw. ohne Virtual Host eines lokalen HTTP-Server ist die Domain-Lizenz unerheblich.) Ist der Lizenzschlüssel im HTML-Script bzw. JScript nicht kodiert aber die Webseite auf HTTP-Server gehostet, dann erscheint eine Fehlermeldung - hier ein Beispiel für die Domain www.twseite.de unter dem IE 7:



Wie diese Abrage funktioniert, konnte der technische Telefon-Support (ca 5 Cent pro Minute) nicht sagen, außer, dass die Schlüssel keine Dongel für Zwangskontakt zu Servern von GData sind. Da aber ein HTTP-Server Routinen des Internetzuganges aktiviert, könnte Webspeech nach Registrierung dieses Status eine per Schlüssel im Scriptcode hinterlegte Zeichenkette per Prüfsumme vergleichen mit zulässigen Werten, wobei das Active-X-Control diesen Algorithmus kennen muss. (Es ist auch so, dass bereits Webspeech 2 einen Parameter hatte für einen Schlüssel hatte, der aber nicht benutzbar war.) - Schlaumeier denken jetzt, dass dann Webspeech 2 diese Domain-Zwangslizenz nicht kennt: Korrekt, aber Webspeech 2 läuft nicht unter Windows XP (Tonerzeugung versagt, es liegt nicht am Active-X-Control selbst).

Fatal ist übrigens:

Diese Meldung erscheint auch, wenn der kodierte Domain-Schlüssel zwar von GData gekauft aber vom Active-X-Control nicht akzeptiert wird.
Es ist also nach Erwerb des Domain-Schlüssels zu prüfen, ob dieser akzeptiert wird. Für diese Prüfung muss die Test-Webseite auf
einem HTTP-Server gehostet sein. Die Testwebseite index.html hat folgenden Inhalt:

```
<HTML>
<HEAD>
<script language="JavaScript">
<!--
// Browser und WebSpeech Plugin abfragen
//WEBSPEECH FOR NAV DETECTION (MUST BE 1ST)
var IsWebSpeech = 0;
var IsNavigator = 0;
var IsExplorer = 0;
	if(navigator.appName=="Netscape" && navigator.plugins && navigator.javaEnabled)
	{
		IsNavigator=1;

		for(i=0;i<navigator.plugins.length;i++)
		{
			if(navigator.plugins[i].description.indexOf("WebSpeech") != -1 )
			{IsWebSpeech=1;}
		}
	}
```

```
// -->
</script>
<!-- WEBSPEECH FOR IE DETECTION (MUST BE 2ND)-->
<script language="VBScript">
on error resume next
if(navigator.javaEnabled) then
IsExplorer=1
IsWebSpeech=IsObject(CreateObject("WebSpeech.WebSpeech"))
end if
</script>
</HEAD>
<BODY>
<script language=Javascript>
// dieser Zweig bestimmt, was WebSpeech tun soll, wenn es installiert ist.
        if(IsWebSpeech==1)
        {
                document.write("<OBJECT ID='WebSpeech1'\n");
                document.write("CLASSID='CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC'\n");
                document.write("WIDTH=120 HEIGHT=80>\n");
                document.write("<PARAM NAME='OPAQUE' VALUE='0'>\n");
                document.write("<PARAM NAME='AUTHKEY' VALUE=' www.xxxx.yy @aaaaa-bbbbb-ccccc-ddddd '>\n");
                                        // Domainschlüssel .www.xxxx.yy @aaaaa-bbbbb-ccccc-ddddd
                document.write("</OBJECT>\n");
        }
        else
        {
                // hier geben Sie an, was passieren soll, wenn WebSpeech nicht installiert ist
                document.write("<a href='http://www.webspeech.de/download.php3'> ");
                document.write("<img src='pix/schweigen.gif' width='120' height='80' ");
                document.write("border='0' alt='Bitte laden Sie WebSpeech!!'></a>\n");
        }

        if(IsWebSpeech==1)
        {alert(window.document.WebSpeech1!=null);
         window.document.WebSpeech1.SetText('Ich bin bereit.');
        window.document.WebSpeech1.StartSpeakingImmediate();
        }
</Script>
</BODY>
</HTML>
```

Der Vertrieb und Support wurde allerdings von GData eingestellt: Das Produkt ist nicht mehr kaufbar. Selbst wenn man es bei Ebay kauft, muss für den Internet-Einsatz eine Domain-Lizenz beschafft werden. Die Webseiten zu der Sprachausgabe offerieren immer noch das Produkt, ohne dass es je in der Produktübersicht zu finden, geschweige im Onlineshop zu finden ist. Das WebSpeech-Forum ist abgeschaltet (toter Link - Webseite nicht gefunden). Emailsupport hüllt sich auch in Ignoranz. - Die Webseiten sind nichts anderes als Vergraulen von Kunden bzw. Interessenten, dafür bieten sie noch den Download der Free-Versionen und freien Dokumentationen
an, welche für Webseiten, die nicht HTTP-Server gebunden werden, nutzbar sind. Unangenehm ist der Umstand, dass GData die Zwangs-Domain-Lizenz nicht abgeschafft hat, wenn GData das Produkt nicht mehr vertreibt und supportet: Da nur GData die Lizenz vergeben konnte, wird es kritisch bei Problemen mit dem Domainschlüssel, so dass der Kauf der Sprachausgabesoftware nicht
lohnt bei Ebay etc.. Das wird allerdings dafür sorgen, dass das Internet langsamer lebendig wird: Eine Alternative zu Webspeech ist z.B. Microsoft Text To Speech, deren Software unter Win XP bereits als Grundkomponente enthalten ist, aber - im Gegensatz zu Webspeech - nicht mit Soundblaster-kompatiblen Soundkarten (Creative Labs-Standard) funktioniert: Z.B. nicht mit der Soundkarte
Creative Labs Xtreme (Modell SB 04060 mit EAX ab Version 4).

5.3.1.1.2. Parameter TEXT und URL

Text der Sprachausgabe bei Erzeugung des Webspeech-Objektes
kann entfallen, aber kein Standard
es ist entweder TEXT oder URL zu kodieren

TEXT Wert ist String also in " " bzw. ' ' zu kodieren
kann Sprechtags enthalten: Achtung Sprechtags vor Webspeech 4 sind in der Regel nicht kompatibel zu Webspeech 4

Beispiel für Webspeech 2: "Herzlich willkommen bei \PAU=500\ WebSpeech."

Beispiel für Webspeech 4: "Herzlich willkommen bei #PAU=500# WebSpeech."

URL Pfad und Name der Datei
Datei: ASCII-Textdatei mit Suffix .TXT
LOGOX-Datei mit Suffix LGX
Datei ist mit Tools zu Webspeech zu erstellen
vor Webspeech 2 reine ASCII-Datei mit Versionsnummer
in

Zeile 1
ab Webspeech 4 kompilierte Datei
Datei kann im Internet liegen Beispiel: "http://www.test.de/text.txt"
lokal liegen Beispiel: "file:///C:/Texte/readme.txt"

5.3.1.1.3. Parameter AUTOSTART

:automatische Sprachausgabe mit Vollendung der Erzeugung des Webspeech-Objektes
Parameter TEXT oder URL muss mit gültigem Wert ungleich Leerkette kodiert worden sein

"0" Standard
keine automatische Sprachausgabe: User muss Play-Button des Webspeech-Control klicken
Achtung: User weiss nicht, ob ein Text beim Erzeugen des Webspeech-Objektes erzeugt wurde oder nicht !

1 für Autostart

5.3.1.1.4. Parameter IMMEDIATE

es muss Autostart aktiv sein
Text sofort sprechen und dabei eine eventuell bereits aktive Sprachausgabe beachten

"0" Standard
warten bis bereits aktives Sprechen beendet ist

"1" bereits aktives Sprechen sofort beenden

5.3.1.1.5. Parameter MOUTHANIMATION

"0" keine Mundanimation bei der Sprachausgabe
"1" Standard
Mundanimation bei der Sprachausgabe

5.3.1.1.6. Parameter MOUTHCOLOR

Farbgestaltung der Mundregion

Farben sind HTML-Farben browserspezifisch
"#rrggbb" mit rr für Rotanteil, gg für Grünanteil, bb für Blauanteil
jeweils hexadezimale Kodierung
"DEFAULT" für Standardfarben
"TRANSPARENT" oder "TRANSP" für Transparenz, also Hintergrund nicht abdecken, also Farbe nicht anzeigen
"TEXTURE" oder "TEXT" für eine vordefinierte Textur-Grafik
wenn TEXTURE verwendet, so wird Angabe der Umrandung ignoriert

Mundbereiche sind Umrandung
Rachen
Zunge
Zähne
Lippen

Wert String aus komma-getrennten Farbangaben in folgender Reihenfolge
Farbe der Umrandung
Farbe des Rachens
Farbe der Zunge
Farbe der Zähne
Farbe der Lippen
Komma sind Teil des String

Standard ist "black, text, text, text, text"

5.3.1.1.6. Parameter TEXTANIMATION

Anzeige des Textes während der Sprachausgabe

"0" keine Textanzeige
"1" Lauftext von rechts nach links
"2" wortweise
"3" Standard
laut Usereinstellung

5.3.1.1.7. Parameter TEXTCOLOR

Textfarbe

<PARAM NAME="TEXTCOLOR" VALUE="#A1B2C3, forestgreen ">

Text	black ist Standard
Schatten	gray ist Standard

Farben sind HTML-Farben browserspezifisch
"#rrggbb" mit rr für Rotanteil, gg für Grünanteil, bb für Blauanteil
jeweils hexadezimale Kodierung
"DEFAULT" für Standardfarben

"TRANSPARENT" oder "TRANSP" für Transparenz, also Hintergrund nicht abdecken, also Farbe nicht anzeigen

Wert String aus komma-getrennten Farbangaben in folgender Reihenfolge
Farbe des Textes
Farbe des Schattens
Komma sind Teil des String

Standard ist "black, gray"

5.3.1.1.8. Parameter TEXTPOSITION

vertikale Textposition relativ zur Lage des OBJECT-Tags bzw. EMBED-Tags
"0" bis "100"
"0" Standard
ganz oben
"100" ganz unten.

5.3.1.1.9. Parameter TEXTSIZE

Textfonthöhe in Pixel
ab "0"
ohne Einheit kodieren
Standard ist "12"

5.3.1.1.10. Parameter BACKGROUNDCOLOR

Hintergrundfarbe des Webspeech-Objektes innerhalb der Dimension laut OBJECT-Tag bzw. EMBEd-Tag (Width und Height), wobei alle sichtbaren Komponenten des Objektes auf der Hintergrundfarbe liegen, also diese abdecken, falls Komponenten nicht transparent sind

Farben sind	HTML-Farben	browserspezifisch
	"#rrggbb"	mit rr für Rotanteil, gg für Grünanteil, bb für Blauanteil jeweils hexadezimale Kodierung
	"DEFAULT"	für Standardfarbe white

Hinweis: Transparenz siehe Parameter OPAQUE

5.3.1.1.11. Parameter OPAQUE

Deckkraft der per Parameter BACKGROUNDCOLOR definierten Hintergrundfarbe
"0" transparent, also Hintergrundfarbe nicht angezeigen
"1" deckend, also Hintergrundfarbe anzeigen

5.3.1.1.12. Parameter CONTROLPOSITION

vertikale Control-Elemente-Leiste relativ zur Lage des OBJECT-Tags bzw. EMBED-Tags
"0" bis "100"
"0" ganz oben
"100" ganz unten.
"75" Standard

5.3.1.2. Webspeech-Objekt in Javascript erzeugen

per Funktion document.write()

5.3.2. Webspeech-Objekt in Javascript verwalten

5.3.2.1. Webspeech-Objekt in Javascript erkennen

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<!------------------------ WebSpeech Erkennung ---------------------------------------->
function NS_Pruefen()
{
	var Wert = 0;

	// prüfen auf NS mit Plugin-Fähigkeit
	if(		(navigator.appName=="Netscape")
		&&	(navigator.plugins)
		&&	(navigator.javaEnabled)
	  )
	{
		// pluginfähiger NS gefunden

		// prüfen auf Webspeech2-Plugin
		var PluginFeld	= navigator.plugins
		var AnzahlPlugins = PluginFeld.length;

		for( var i=0; i < AnzahlPlugins; i++)
	{
			var PluginBeschreibung = PluginFeld[i].description;
```

```
                        if(PluginBeschreibung.indexOf("WebSpeech.2") != -1 )
                        {Wert=1;} // Plugin gefunden
                }
        }

        return Wert;
}

<!-- Schritt 1: Webspeech-Erkennung für NS mit Pluginfähigkeit //-->
        var IsWebSpeech = NS_Pruefen();         // IsWebSpeech muss global sein
</SCRIPT>

<!-- Schritt 2: Webspeech-Erkennung für IE nur per Visual Basic möglich //-->
<SCRIPT LANGUAGE="VBScript">
on error resume next
  if(navigator.javaEnabled) then
  IsWebSpeech=IsObject(CreateObject("WebSpeech.WebSpeech"))
end if
</SCRIPT>

<!-- Schritt 3: Webspeech-Erkennung für Opera //-->
<SCRIPT LANGUAGE="JavaScript">
        // Opera kann kein Webspeech (kann kein ActiveX noch NAV-Plugin
        //                            und Opera-Plugin existiert nicht)
 if (navigator.userAgent.indexOf("Opera") != -1)
{IsWebSpeech = 0;}
</SCRIPT>

<!------------------------ WebSpeech Routinen ---------------------------------------->

<SCRIPT LANGUAGE="Javascript">

        <!------------------------ vorlesen ---------------------------------------->

        function vorlesen()
        {
                if(IsWebSpeech==1)
                {
                        var Version = WebSpeechObjekt.GetVersion();
                        {alert(Version); } // z.B. 4. 0. 2. 34 bei Webspeech 4

                        HauptVersion = Version.slice(0,1);
                        var Text = "Webspeech " + HauptVersion + " erkannt."; // z.B. 4  bei Webspeech 4

                        WebSpeechObjekt.SetText(Text);
                        WebSpeechObjekt.StartSpeaking();
                }
        }

        <!------------------------ WebSpeech Init ---------------------------------------->

        var WebSpeechObjekt;            // muss global sein

        function init()
        {
                if(IsWebSpeech==1)
                {
                        document.writeln("<OBJECT       ID='ID_WebSpeech'
                                                CLASSID='CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC'
                                                WIDTH=110
                                                HEIGHT=100
                                >");  // alles 1 Zeile
                        document.writeln("<PARAM        NAME='AUTHKEY'
                                                VALUE='www.webspeech.de@RGVUA-CSN52-22B98-KU3WD'
                                >"); // alles 1 Zeile
                        document.writeln("<PARAM NAME='AUTOSTART' VALUE='1'>");
                        document.write("<PARAM NAME='TEXTSIZE' VALUE='8'>\n");
                        document.write("<PARAM NAME='TEXTANIMATION' VALUE='2'>\n");
                        document.write("<PARAM NAME='TEXTCOLOR' VALUE='black,transparent'>\n");

                        // NS
                        document.write("<EMBED\n");
                        document.write("        NAME='ID_WebSpeech '
                                        TYPE='application/x-WebSpeech' WIDTH='110' HEIGHT='100'\n"
```

```
                                        );      // alles 1 Zeile
                                document.write("PARAM_AUTOSTART='1'\n");
                                document.write("PARAM_TEXTSIZE='8'\n");
                                document.write("PARAM_TEXTANIMATION='2'\n");
                                document.write("PARAM_TEXTCOLOR='black,transparent'\n");
                                document.write(">\n");
                                document.write("</EMBED>\n");

                                document.write("</OBJECT>\n");

                                WebSpeechObjekt = window.document.ID_WebSpeech; // NS und IE
                        }
                        else
                        {alert('Es ist kein WebSpeech installiert');}
                }
//-->
</SCRIPT>
</HEAD>
<BODY onLoad="init();vorlesen();">
</BODY>
</HTML>
```

5.3.2.2. Webspeech-Objekt in Javascript programmieren

5.3.2.2.1. Webspeech-Objekt und seine Methoden

Die Methoden lassen die vollständige Verwaltung der Webspeech-Objektes bezüglich der Eigenschaften der Sprachausgabe zu und das auch nach der Erzeugung des Webspeech-Objektes.

Parameter aus der HTML-Kodierung des Webspeech-Objektes sind als Javascript-Methoden entweder direkt oder indirekt ansprechbar. Es ist möglich, in der HTML-Kodierung keine Parameter anzugeben und diese durch Javascript zu ersetzen.

Hinweis: Entgegen den Beschreibungen von Logox liefern Funktionen, die einen Rückkehrcode besitzen sollen, diesen z.T. **nicht**:
Unter der kostenpflichtigen Version des Active-X-Controls von WebSpeech 4 in Verbindung mit Logox 4 **Professional** mit Serverupdate 1 (fehlerfreie Installation),
sowie bei Nutzung des IE 6.0 SP1 in W9x (Netscape wird von WebSpeech 4 nicht mehr unterstützt),
sowie mit Kodierung sämtlicher Scriptanweisungen für WebSpeech (inklusive der Erzeugung des WebSpeech-Objektes) per JScript (nicht JavaScript) im HEAD des HTML-Dokumentes,
wird für die Abfrage des Rückkehrcodes o.g. Funktionen **undefinded** geliefert.
Desweiteren wird die Zeigerabfrage auf eine WebSpeech-Funktion per
zeiger_auf_webspeech_objekt.webspeech_funktion != null
vom Browser mit der Meldung quittiert, dass das Objekt die Funktion **nicht unterstützt**, obwohl das Objekt existiert. (Anstelle zeiger_auf .... ist natürlich der konkrete Zeiger und anstelle webspeech_funktion die konkrete zulässige Funktion zu kodieren). Die trotzdem durchgeführte Ausführung der o.g. WebSpeech-Funktionen, die keinen Rückkehrcode liefern, kann man an den **manuellen** Steuerungselementen des WebSpeech-Objektes testen.

Unter JScript in Verbindung mit einer Rückkehrcode-Abfrage nutzbare Funktionen sind IsMuting()), IsSpeaking(), IsPause(), IsLoading(), jedoch nicht z.B. SetText():

In der Kodierung if (zeiger_auf_webspeech_objekt.SetText('Testtext')){....}else{...} liefert if den Wert undefinded, so dass die if-Steuerung hinfällig ist.

Die Kodierung zeiger_auf_webspeech_objekt.SetText('Testtext'); führt die Funktion aus (wenn das Objekt existiert), gibt aber keine Auskunft, dass die Funktion ausgeführt wurde (Erfolg oder Misserfolg). SetText() kann also nicht für kontrolliertes Laden eines Textes benutzt werden. Nur das manuelle Klicken auf das Sprech-Symbol in der Steuerungsleiste zum WebSpeech-Objekt gibt Auskunft. Wenn der Test des HTML-Dokumentes mit WebSpeech erfolgreich war, heisst das aber nicht, dass die Funktion SetText() auch immer im Onlinezustand des HTML-Dokumentes funktioniert. Damit hat der Programmierer das Nachsehen.: Die Steuerung des WebSpeech-Objektes ist schlichtweg **unvollständig** implementiert.

Die Kodierung if (zeiger_auf_webspeech_objekt.SetText != null) liefert die Browsermeldung, dass SetText nicht vom Objekt unterstützt wird, obwohl die Kodierung zeiger_auf_webspeech_objekt.SetText('Testtext'); die Funktion ausführt, wenn das Objekt existiert.

Diese Situation ist äußerst bedauerlich ! Aufgrund eigener Erfahrungen des Autors dieser JavaScript-Dokumentation (der in Besitz mehrerer gekaufter Lizenzen von WebSpeech 4 ist) bezüglich des (für deutsche Verhältnisse typischen) Kundenservices des deutschen Logox-Software-Anbieters (schwerfälliger bzw. z.T. unterlassener Email-Service (selbst bei Beantragung eines Autoren-Schlüssels aufgrund gekaufter und registrierter Lizenz), Alternativ-Service auf Basis einer 0190-Telefonnummer mit Kosten in Höhe von 1,86 Euro pro Minute, unzureichende Pflege bzw. dürftiger Inhalt des Internet-Forums) hat der Autor dieser JavaScript-Dokumentation eine Supportanfrage beim o.g. Softwareanbieter unterlassen. Der (weitere bzw. zukünftige) Einsatz von Webspeech will also aus kosten- und programmtechnischen Gründen genau überlegt sein. Wer z.B. den Versuch scheut, o.g. fehlende Rückkercode durch eigenen Programmierungsaufwand per JScript zu umgehen, sollte o.g. WebSpeech nicht oder nur unter Duldung o.g. Nachteile einsetzen.

.GetInterfaceVersion() Haupt-Version von Webspeech ermitteln, das auf dem PC des Users installiert ist
Syntax:

[var Kette =] zeiger_auf_webspeech_objekt..GetVersion()

Kette String z.B. Webspeech 4 "4"
Webspeech 2 "1"

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

Beispiel:

```
var Kette = zeiger_auf_webspeech_objekt.GetInterfaceVersion(); //  "4"
```

.GetVersion() Version von Webspeech ermitteln, das auf dem PC des Users installiert ist

Syntax:

[var Kette =] zeiger_auf_webspeech_objekt..GetVersion()

Kette String

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

Beispiel:

```
var Kette = zeiger_auf_webspeech_objekt.GetVersion(); //Webspeech 4  z.B. "4. 0. 2. 34"
                                                       // Webspeech 2 z.B. "2.0.0.2. IE"
HauptVersionsNumerAlsKette = Kette.slice(0,1);
alert(Kette + "\n" + "Webspeech " + HauptVersionsNumerAlsKette + " erkannt.");
                                   // 2 oder oder oder 4 etc.
```

.isLoading() prüfen auf aktive Pufferung eines Textes
siehe .SetText() und .LoadText()

Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.isLoading();

Wert Integer
1 so Text wird gepuffert
0 so Text ist komplett gepuffert oder wird noch nicht gepuffert

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

Beispiel:

```
var AnzahlSekunden = 1;

function TextPufferungAbwarten()
{
        var LoadingAktiv =document.WebSpeech.IsLoading();

        if (LoadingAktiv==1)
        {
                AnzahlSekunden += 1;
                window.setTimeout(TextPufferungAbwarten()',1000);
        }
        else
        {alert("Ladezeit maximal "+  AnzahlSekunden  + " Sekunden");}
}
```

.isMuting() prüfen auf stummgeschaltete Sprachausgabe (egal ob aktiv oder ob sie pausiert)

Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.isMuting();

Wert Integer
1 so Sprachausgabe stummgeschaltet
0 so Sprachausgabe nicht stummgeschaltet

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.isPause() prüfen auf aktive und pausierte Sprachausgabe
siehe .isSpeaking()

Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.isPause();

Wert Integer
1 so aktive Sprachausgabe und pausiert
0 so aktive Sprachausgabe und pausiert nicht

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.isSpeaking()

prüfen auf aktive Sprachausgabe (egal ob diese pausiert oder nicht)
siehe .isPause()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.isSpeaking();

Wert Integer
1 so aktive Sprachausgabe
0 so keine aktive Sprachausgabe

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.LoadText()

entspricht Parameter URL
zu sprechenden Text aus Datei puffern für die **nächste** Sprachausgabe (Sprechtags möglich im Text)
je größer die Textdatei um so länger der Zeitbedarf für die Umwandlung
(anhängig von der PC-CPU)
bereits gesprochener Text wird nicht beeinflusst
Sprachausgabe starten mit StartSpeaking()
siehe .SetText()
Syntax:

[var Wert3 =] zeiger_auf_webspeech_objekt.LoadText(Kette, Wert2, Wert1);

Kette Pfad und Name der Datei
Datei: ASCII-Textdatei mit Suffix .TXT
LOGOX-Datei mit Suffix LGX
ist mit Tools zu Webspeech zu erstellen
vor Webspeech 2reine ASCII-Datei mit Versionsnummer in Zeile 1
ab Webspeech 4 kompilierte Datei
kann im Internet liegen Beispiel: "http://www.test.de/text.txt"
lokal liegen Beispiel: "file:///C:/Texte/readme.txt"

Wert1 Integer
1 so Autostart des Sprachausgabe (siehe Wert 2)
0 so Sprachausgabe starten per StartSpeaking() oder
StartSpeakingImmediate()

Wert 2 Integer
Wert1 muss 1 sein
0 warten bis bereits aktives Sprechen beendet ist
1 bereits aktives Sprechen sofort beenden (auch wenn diese pausiert)

Wert3 Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.PauseSpeaking()

aktive Sprachausgabe pausieren, es sei denn, sie pausiert bereits
siehe .ResumeSpeaking()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.PauseSpeaking();

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.ResumeSpeaking()

aktive und pausierte Sprachausgabe fortsetzen (Pause beenden)
siehe .PauseSpeaking()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.ResumeSpeaking( );

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetBackgroundColor()

entspricht BACKGROUNDCOLOR
Hintergrundfarbe des Webspeech-Objektes einstellen
Deckkraft der Hintergrundfarbe einstellen per .SetOpaque()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.SetBackgroundColor(Kette)

Kette String
Farbangabe
HTML-Farben
#RRGGBB
DEFAULT (Standardfarben),

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetControlPosition()

entspricht Parameter CONTROLPOSITION
vertikale Position der Steuerungselemente für User relativ zur Lage des Webspeech-Ojektes
Syntax:

[var Wert2 =] zeiger_auf_webspeech_objekt.SetControlPosition(Wert1);

Wert1 Integer
0 bis 100
0 ganz oben
100 ganz unten 100

Wert2 Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetKey()

ab Webspeech 4
entspricht Parameter AUTHKEY
Lizenz-Schlüssel festlegen
für lokale Test des HTML-Dokumentes ohne Internetzugriff auf die Domain kann der herstellereigene Schlüssel verwendet werden:
"www.webspeech.de@RGVUA-CSN52-22B98-KU3WD"
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt..SetKey(Kette)

Kette String
Author-Schlüssel laut Lizenz

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetMouthAnimation()

entspricht Parameter MOUTHANIMATION
Mundanimation einstellen
Syntax:

[var Wert2 =] zeiger_auf_webspeech_objekt..SetMouthAnimation(Wert1);

Wert1 Integer
1 Mund wird mitbewegt
0 Mund wird nicht mitbewegt

Wert2 Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetMouthColor()

entspricht Parameter MOUTHCOLOR
Mundfarben einstellen

Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.SetMouthColor(Kette)

Kette Liste von kommagetrennten Farbangaben
als String (inklusive Kommas)
Listenelemente mit nachfolgender Reihenfolge
Farbe der Umrandung
Farbe des Rachen
Farbe der Zunge
Farbe der Zähne
Farbe der Lippen

HTML-Farben
#RRGGBB
DEFAULT (Standardfarben),
TRANSPARENT oder TRANSP (Hintergrundtransparenz)
TEXTURE oder TEXT Grafik in die aktuelle Form des Mundes
wenn TEXTURE verwendet, so wird Angabe der Umrandung ignoriert

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetOpaque() entspricht Parameter OPAQUE
Deckkraft der Hintergerundfarbe des Webspeech-Objektes einstellen
Hintergrundfarbe definiert per .SetBackgroundColor()
Syntax:

[var Wert2 =] zeiger_auf_webspeech_objekt.SetOpaque(Wert1);

Wert1 Integer
0 transparent, also Hintergrundfarbe des Webspeech-Objektes nicht sichtbar
dafür die Hintergrundfarbe des Containers vom Webspeech-Objekt
1 Hintergrundfarbe deckt ab, also nur Hintergrundfarbe des
vom Webspeech-Objekt sichtbar und nicht die des Containers

Wert2 Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetText() entspricht Parameter TEXT
zu sprechenden Text puffern für die **nächste** Sprachausgabe (Sprechtags möglich im Text)
bereits gesprochener Text wird nicht beeinflusst
Sprachausgabe starten mit StartSpeaking()
siehe .LoadText()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.SetText(Kette);

Kette String, auch Literal
kann Sprechtags enthalten: Achtung Sprechtags vor Webspeech 4 sind in
der Regel nicht kompatibel zu Webspeech 4

Beispiel für Webspeech 2:
"Herzlich willkommen bei \pau=500\ WebSpeech."

Beispiel für Webspeech 4:
"Herzlich willkommen bei #PAU=500# WebSpeech."

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetTextAnimation() entspricht Parameter TEXTANIMATION
Anzeige des Textes während der Sprachausgabe einstellen
Syntax:

[var Wert2 =] zeiger_auf_webspeech_objekt..SetTextAnimation(Wert1)

Wert1 Integer
0 bis 3
0 keine Anzeige
1 als Lauftext von rechts nach links
2 einzelne Wörter erscheinen nacheinander
3 laut Usereinstellung

Wert2 Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetTextColor() entspricht Parameter TEXTCOLOR
Textfarben einstellen
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.SetTextColor(Kette)

Kette Liste von kommagetrennten Farbangaben
als String (inklusive Kommas)
Listenelemente mit nachfolgender Reihenfolge
Farbe des Textes
Farbe des Schattens
HTML-Farben
#RRGGBB
DEFAULT (Standardfarben),
TRANSPARENT oder TRANSP (Hintergrundtransparenz)

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetTextPosition() entspricht Parameter TEXTPOSITION
vertikale Position der Textanzeige während der Sprachausgabe relativ zur Lage des Webspeech-Ojektes
Syntax:

[var Wert2 =] zeiger_auf_webspeech_objekt.SetTextPosition(Wert1);

Wert1 Integer
0 bis 100
0 ganz oben
100 ganz unten 100

Wert2 Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.SetTextSize() entspricht Parameter TEXTSIZE
Fonthöhe der Textanzeige während der Sprachausgabe
Syntax:

[var Wert2 =] zeiger_auf_webspeech_objekt.SetTextSize(Wert1);

Wert1 Integer, in Pixel
>0

Wert2 Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.StartSpeaking() nächste Sprachausgabe starten mit Abwarten auf das Ende einer bereits aktiven Sprachausgabe
wiederholbar ohne erneute Pufferung von Text
es muss ein Text gepuffert sein (siehe SetText() und LoadText()), sonst erfolgt keine Sprachausgabe
siehe .StopSpeaking() und .PauseSpeaking()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.StartSpeaking();

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.StartSpeakingImmediate() nächste Sprachausgabe starten ohne Abwarten auf das Ende einer bereits aktiven Sprachausgabe, die sofort beendet wird (auch wenn diese pausiert)
wiederholbar ohne erneute Pufferung von Text
es muss ein Text gepuffert sein (siehe SetText() und LoadText()), sonst erfolgt keine Sprachausgabe
siehe .StopSpeaking() und .PauseSpeaking()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.StartSpeaking();

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

.StopSpeaking() aktive Sprachausgabe beenden (auch wenn diese pausiert)
siehe .StartSpeaking() und .StartSpeakingImmediate()
Syntax:

[var Wert =] zeiger_auf_webspeech_objekt.StopSpeaking();

Wert Integer
>= 0 so erfolgreich
< 0 so nicht erfolgreich

zeiger_auf_webspeech_objekt laut ID des OBJECT-Tags
NAME des EMBED-Tags

5.3.2.2.2. Webspeech-Objekt und Events

Events in Javascript exisitieren nicht. Alternativ sind die Methoden isXXX() zu verwenden
Events sind nur über VBScript realisierbar .

5.3.2.2.3. Beispiel zur Javscript-Programmierung zu Webspeech 2 für IE und NS 4.x

Hinweis: Die Programmierung für die WebSpeech-Version höher als 2 ist
nur z.T. abwärtskompatibel zur WebSpeech-Version 2
nicht im selben Umfang möglich wie wie WebSpeech-Version 2 (siehe obige Bemerkungen)

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
        // ############ WebSpeech-Prüfung auf ActiveX des IE
        var IE_ActivexInstalliert=false;        //  nur IE: Annahme: ActiveX ist nicht installiert

//-->
</SCRIPT>

<SCRIPT LANGUAGE="VBScript">
on error resume next

if (navigator.javaEnabled) then
IE_ActivexInstalliert = IsObject(CreateObject("WebSpeech.WebSpeech.2"))
end if
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript1.2">
<!--
function WebSpeechInstallationPruefen()
{
        var ReturnWert=0;                       //  0 für Webspeech-Plugin ist nicht installiert
                                                //        im Browser (egal ob NS oder IE)
                                                //     oder IE  bzw. NS nicht aktiv
                                                //  >0  für Webspeech-Plugin ist installiert im NS
                                                //                  und NS aktiv
                                                //  <0  für Webspeech-Plugin ist installiert im IE
                                                //                  und IE aktiv
        if (ns)
```

```
	{
		if (	(navigator.plugins)		//  Plugins-Objekt existiert
		    && (navigator.javaEnabled)		// Java ist aktiv im Browser
		  )
		{
			//  prüfen auf eingebundenes Plugin "Webspeech.2"
			//  Objekt navigator.plugins[] ist Feld
			//		mit Anzahl der installierten Plugins lauf navigator.plugins.lenght
			//   ist leer, wenn kein Plugin (egal welcher Hersteller) installiert ist
			//					im Browser, also Anzahl = 0
			//  Achung: Feld-Index immer ab 0 !!

			var Zahler=0;					// Index ab 0
			while (	(Zahler < navigator.plugins.lenght)		// Länge ab 1
				&&  (ReturnWert = 0)			// laut Deklaration
				)
			{
				if (navigator.plugins[Zahler].description.indexOf("WebSpeech.2") != -1)
				{ReturnWert=1;}		//  Webspeech-Plugin ist installiert

				Zahler++;
			}
		}
	}
	else
	{
		if (ie)
		{
			if (IE_ActivexInstalliert)		// siehe oben
			{
				//  IE aktiv und ActiveX-Control installiert
				ReturnWert=-1;
			}
		}
	}

	return ReturnWert;
}

// +++++++++++ Routinen zur WebSpeech-Ausgabe
function WebSpeech_TextStringSprechen (Text, Warten)
//	benötigt erzeugtes Standard-WebSpeech-Objekt
//	Text	: nur String, Kann leer sein,
//		also nur zum LÖSCHEN des Objekt-Text-Puffers, damit Useraktion
//		auf Control-Leiste keine Ausgabe auslöst
//	Warten :	0	Sprachausgabe sofort starten und bereits aktive abbrechen
//		1	Sprachausgabe erst starten, wenn aktive beendet ist
{
	if (WebSpeechInstalliert != 0)
	{
		parent.info.WebSpeech_TextTemporaerLadenUndSprechen(	WebSpeechStandardObjektNr,
		0,
		Text,
		Warten
		);
	}
}

// + + + + + + + + + + + + auf erzeugtes Objekt prüfen
function WebSpeech_AufErzeugtesHTMLObjektPruefen(ObjektNr)
// liefert true, wenn Objekt erzeugt ist
//		sonst false
// ObjektNr muss korrekt sein und wird nicht geprüft !
// nur ein erzeugtes Objekt ist auch sichtbar
{
	return (WebSpeechObjektZeigerFeld[ObjektNr] != null);
}

// + + + + + + + + + + + + Zustand eines erzeugtes WebSpeech-Objekt ermitteln
//		benötigt WebSpeechInstalliert
function WebSpeech_HTMLObjektZustandErmitteln(ObjektNr)
// aktualisiert WebSpeechObjektZustandFeld[]
// liefert -1 Fehler weil Objektnummer falsch und oder Objekt nicht installiert
//		0	unbekannter Zustand
```

```
//                  1       Ausgabe gerade begonnen worden
//                  2       gestartete Ausgabe komplett beendet
//                  3       gestartete Ausgabe pausiert
//                  4       gestoppte Ausgabe gerade erneut gestartet
//                  5       Ausgabe ist vom User stummgeschaltet (muted)
//                  6       Text wird gerade geladen
{
        var ReturnWert=-1;              // Annahme: Objekt nicht installiert oder ObjektNr ist falsch

        if (WebSpeechObjektZeigerFeld[ObjektNr])
        {
                //      erst Ereignisse abfragen
                ReturnWert=WebSpeechObjektZustandFeld[ObjektNr];

                if (ReturnWert == 0)
                {
                        // unbekannter Zustand also Is-Routinen bemühen
                        //      auf Muten prüfen
                        if (WebSpeechObjektZeigerFeld[ObjektNr].IsMuting())
                        {
                                ReturnWert=5;

                                // und WebSpeechObjektZustandFeld korrigieren
                                WebSpeechObjektZustandFeld[ObjektNr]=5;
                        }
                        else
                        {
                                // auf gerade sprechen prüfen
                                if (WebSpeechObjektZeigerFeld[ObjektNr].IsSpeaking())
                                {
                                        ReturnWert=1;

                                        // und WebSpeechObjektZustandFeld korrigieren
                                        WebSpeechObjektZustandFeld[ObjektNr]=1;
                                }
                                else
                                {
                                        // auf gerade pausieren prüfen
                                        if (WebSpeechObjektZeigerFeld[ObjektNr].IsPause())
                                        {
                                                ReturnWert=3;

                                                // und WebSpeechObjektZustandFeld korrigieren
                                                WebSpeechObjektZustandFeld[ObjektNr]=3;
                                        }
                                        else
                                        {
                                                // auf gerade Laden des Textes prüfen
                                                if (WebSpeechObjektZeigerFeld[ObjektNr].IsLoading())
                                                {
                                                        ReturnWert=6;

                                                        // und WebSpeechObjektZustandFeld korrigieren
                                                        WebSpeechObjektZustandFeld[ObjektNr]=6;
                                                }
                                        }

                                        // mehr Möglichkeiten gibts nicht im SDK !
                                }
                        }
                }
        }

        return ReturnWert;
}

// + + + + + + + + + + + WebSpeech-Objekt erzeugen
//      benötigt WebSpeechInstalliert
function WebSpeech_HTMLObjektErzeugen(          ObjektNr,
                                                AbstandVomLinkenFensterrandInPixel,
                                                AbstandVomOberenFensterrandInPixel,
                                                BreiteInPixel,
                                                HoeheInPixel,
                                                HintergrundAbdecken,
```

```
                                        HintergrundFarbe,
                                        Mund_Bewegen,
                                        Mund_Umrandung_Farbe,
                                        Mund_Rachen_Farbe,
                                        Mund_Zunge_Farbe,
                                        Mund_Zaehne_Farbe,
                                        Mund_Lippen_Farbe,
                                        AutoStart,
                                        Text_Bewegen,
                                        Text_Farbe,
                                        Text_Schatten_Farbe,
                                        Text_Position,
                                        Text_Hoehe,
                                        Text                    // nur String und keine Url !!
                                )
{
        var ReturnWert=false;           // Annahme : Objekt nicht installiert
        var ObjektZustandWert=0;

        if (!WebSpeech_AufErzeugtesHTMLObjektPruefen(ObjektNr))
        {
                // Objekt existiert noch nicht, also erzeugen

                var EvalKommando='';

                var MundFarbKette =     Mund_Umrandung_Farbe + ','
                                + Mund_Rachen_Farbe     + ','
                                + Mund_Zunge_Farbe      + ','
                                + Mund_Zaehne_Farbe     + ','
                                + Mund_Lippen_Farbe;

                var HintergrundFarbKette = HintergrundFarbe;

                var TextFarbKette = Text_Farbe + ',' + Text_Schatten_Farbe;

                if (WebSpeechInstalliert > 0)
                {
                        //  Netscape aktiv und Plugin ist installiert

                        //  WebSpeech-Objekt im DIV erzeugen mit Ereignissteuerung aktivieren
                        //              wobei die EVENT-Angaben identsich sein müssen mit den
                        //              JavaScript-Ereignis-Routinen
                        document.write(         '<DIV ID="DIV_ID_WebSpeech' + ObjektNr.toString() + '"'
                                        +  ' STYLE="position: absolute;'
                                        +               'left:'     + AbstandVomLinkenFensterrandInPixel            + 'px;'
                                        +               'top:'      + AbstandVomOberenFensterrandInPixel            + 'px;'
                                        +               'width:'    + BreiteInPixel                                 + 'px;'
                                        +               'height:'   + HoeheInPixel                                  + 'px;'
                                        +               ''''
                                        + '>\n'
                                        +  '<EMBED'
                                        +               ' NAME="EMBED_ID_WebSpeech'  + ObjektNr.toString()          + '"\n'
                                        +               ' TYPE="application/x-WebSpeech"\n'
                                        +               ' WIDTH="'  + BreiteInPixel                                 + '"\n'
                                        +               ' HEIGHT="'  + HoeheInPixel                                 + '"\n'
                                        +               ' PARAM _AUTOSTART="'  + AutoStart                          + '"\n'
                                        +               ' PARAM _IMMEDIATE="1\n'
                                        +               ' PARAM _TEXT="'  + Text                                    + '"\n'
                                        +               ' PARAM _MOUTHCOLOR=" + MundFarbKette                       + '"\n'
                                        +               ' PARAM _MOUTHANIMATION="' + Mund_Bewegen                   + '"\n'
                                        +               ' PARAM _TEXTANIMATION="'  +  Text_Bewegen                  + '"\n'
                                        +               ' PARAM _TEXTCOLOR="' + TextFarbKette                       + '"\n'
                                        +               ' PARAM _TEXTPOSITION="' + Text_Position                    + '"\n'
                                        +               ' PARAM _TEXTSIZE="'  + Text_Hoehe                          + '"\n'
                                        +               ' PARAM _BACKGROUNDCOLOR="'
                                        +                       HintergrundFarbKette                                + '"\n'
                                        +               ' PARAM _OPAQUE="'+ HintergrundAbdecken                     + '"\n'
                                        +               'EVENT_ONSTARTSPEAKING="DoOnStartSpeaking'
                                        +                       ObjektNr.toString() + '(Reason)"\n'
                                        +               'EVENT_ONSTOPSPEAKING="DoOnStopSpeaking'
                                        +                       ObjektNr.toString() + '(Reason)"\n'
                                        +               'EVENT_ONPAUSESPEAKING="DoOnPauseSpeaking'
                                        +                       ObjektNr.toString() + '(Reason)"\n'
                                        +               'EVENT_ONRESUMESPEAKING="DoOnResumeSpeaking'
```

```
                                        +           ObjektNr.toString() + '(Reason)'"
                                        +   '>\n'
                                        +   '</EMBED>\n'
                                        +   '</DIV>\n'
                                );

                // und Zeiger merken
                WebSpeechObjektZeigerFeld[ObjektNr]=
                                        eval('window.document.EMBED_ID_WebSpeech' + ObjektNr.toString());

                ReturnWert=true;
        }
        else
        {
                if (WebSpeechInstalliert < 0)
                {
                        //  IE aktiv und ActiveX ist installiert

                        //   WebSpeech-Objekt im DIV erzeugen
                        //          anstelle von EVENT muss VisualBasicScript verwendet werden "
                        document.write(          '<DIV ID="DIV_ID_WebSpeech' + ObjektNr.toString() + '"'
                                        +           ' STYLE="position: absolute;'
                                        +                   'left:' + AbstandVomLinkenFensterrandInPixel  + 'px;'
                                        +                   'top:'  + AbstandVomOberenFensterrandInPixel + 'px;'
                                        +                   'width:'  + BreiteInPixel                    + 'px;'
                                        +                   'height:' + HoeheInPixel                     + 'px;'
                                        +                         '"'
                                        +  '>\n'
                                        +   '<OBJECT ID="OBJECT_ID_WebSpeech' +ObjektNr.toString() + '"\n'
                                        +               ' CLASSID="'          + CLASS_ID                    + '"\n'
                                        +               ' WIDTH="'            + BreiteInPixel               + '"\n'
                                        +               ' HEIGHT="'           + HoeheInPixel                + '"\n'
                                        +    '>\n'
                                        +           '<PARAM NAME="AUTOSTART"
                                        +                       " VALUE="' +  AutoStart                   + '">\n'
                                        +           '<PARAM NAME="IMMEDIATE"
                                        +                       " VALUE="1">\n'
                                        +           '<PARAM NAME="TEXT"
                                        +                       " VALUE="' + Text                         + '">\n'
                                        +           '<PARAM NAME="MOUTHCOLOR"
                                        +                       " VALUE="' + MundFarbKette                + '">\n'
                                        +           '<PARAM NAME="MOUTHANIMATION"
                                        +                       " VALUE="' + Mund_Bewegen                 + '">\n'
                                        +           '<PARAM NAME="TEXTANIMATION"
                                        +                       " VALUE="' + Text_Bewegen                 + '">\n'
                                        +           '<PARAM NAME="TEXTCOLOR"
                                        +                       " VALUE="' + TextFarbKette                + '">\n'
                                        +           '<PARAM NAME="TEXTPOSITION"
                                        +                       " VALUE="' + Text_Position                + '">\n'
                                        +           '<PARAM NAME="TEXTSIZE"
                                        +                       " VALUE="' + Text_Hoehe                   + '">\n'
                                        +           '<PARAM NAME="BACKGROUNDCOLOR"
                                        +                        "VALUE="' + HintergrundFarbKette         + '">\n'
                                        +           '<PARAM NAME="OPAQUE"
                                        +                       " VALUE="' + HintergrundAbdecken          + '">\n'
                                        +   '</OBJECT>\n'
                                        +   '</DIV>\n'
                                );

                        // und Zeiger merken
                        WebSpeechObjektZeigerFeld[ObjektNr]=
                                        eval('document.OBJECT_ID_WebSpeech' + ObjektNr.toString());

                        ReturnWert=true;
                }
        }

        if (ReturnWert)
        {
                // Dimension des Objektes merken
                WebSpeechObjektBreiteFeld[ObjektNr]                = BreiteInPixel;
                WebSpeechObjektHoeheFeld[ObjektNr]                 = HoeheInPixel;

                // Positionen des Objektes merken zum eventuellen Löschen auf Bildschirm
```

```
                        WebSpeechObjektLeftFeld[ObjektNr]           = AbstandVomLinkenFensterrandInPixel;
                        WebSpeechObjektTopFeld[ObjektNr]            = AbstandVomOberenFensterrandInPixel;

                        // und Textquelle "String" und Text merken
                        WebSpeechObjektTextQuelleFeld[ObjektNr]     = true;     // true = String, false = Url
                        WebSpeechObjektTextOderUrlFeld[ObjektNr]    = Text;

                        // Objektzustand ermitteln: aktualisiert WebSpeechObjektZustandFeld[]
                        //          Funktionswert egal, da nur das Zustandfeld gefüllt werden muss
                        ObjektZustandWert=WebSpeech_HTMLObjektZustandErmitteln(ObjektNr);
                }
        }

        return ReturnWert;
}

// + + + + + + + + + + + WebSpeech-Objekt vom Bildschirm verschwinden lassen
function WebSpeech_HTMLObjektEinAusblenden(ObjektNr, AnzeigeArt)
//      liefert     true, wenn Objekt ausgeblendet wurde
//                  sonst false, auch wenn Parameter falsch
//      ObjektNr  >=0
//      AnzeigeArt  true, so einblenden
//                  false, so ausblenden
{
        var ReturnWert=false;               // Annahme : Objekt nicht gelöscht
        var EvalKommando="";

        if (WebSpeechObjektZeigerFeld[ObjektNr])
        {
                var Faktor=0;

                if (AnzeigeArt)
                {
                        // einblenden
                        Faktor=1;
                }
                else
                {
                        // ausblenden
                        Faktor=-1;
                }

                if (WebSpeechInstalliert > 0)
                {
                        //  Netscape aktiv und Plugin ist installiert
                        //          Objekt komplett aus dem Bildschirm schieben

                        EvalKommando=       'document.layers.DIV_ID_WebSpeech' + ObjektNr.toString()
                                            + '.left='
                                            + ( Faktor * WebSpeechObjektLeftFeld[ObjektNr])
                                            + ';'
                        eval(EvalKommando);

                        EvalKommando=       'document.layers.DIV_ID_WebSpeech' + ObjektNr.toString()
                                            + '.top='
                                            + ( Faktor * WebSpeechObjektTopFeld[ObjektNr])
                                            + ';'
                        eval(EvalKommando);

                        ReturnWert=true;
                }
                else
                {
                        if (WebSpeechInstalliert < 0)
                        {
                                //  IE aktiv und ActiveX ist installiert
                                //          Objekt komplett aus dem Bildschirm schieben
                                EvalKommando=       'document.all.DIV_ID_WebSpeech' + ObjektNr.toString()
                                                    + '.style.left='
                                                    +  ( Faktor * WebSpeechObjektLeftFeld[ObjektNr])
                                                    + ';'
                                eval(EvalKommando);

                                EvalKommando=       'document.all.DIV_ID_WebSpeech' + ObjektNr.toString()
```

```
                                                + '.style.top='
                                                + ( Faktor * WebSpeechObjektTopFeld[ObjektNr])
                                                + ';'
                                        eval(EvalKommando);

                                        ReturnWert=true;
                                }
                        }
                }

                return ReturnWert;
}

// + + + + + + + + + + + WebSpeech-Objekt löschen und vom Bildschirm verschwinden lassen
function WebSpeech_HTMLObjektLoeschen(ObjektNr)
//              liefert         true, wenn Objekt erzeugt wurde
//                                      sonst false, auch wenn Parameter falsch
// Es ist egal, welchen Zustand das Objekt hat
//              aber ein gerade gesprochener Text (auch gestapelter) wird zu Ende gesprochen,
//                      auch wenn das Objekt bereits gelöscht ist
{
                var ReturnWert=false;                   // Annahme : Objekt nicht gelöscht

                // Objekt ausblenden auf BS
                ReturnWert = WebSpeech_HTMLObjektEinAusblenden(ObjektNr, false);

                if (ReturnWert)
                {
                        // und DANACH die Instanz löschen
                        WebSpeechObjektZeigerFeld[ObjektNr]             = null;
                        WebSpeechObjektBreiteFeld[ObjektNr]             = null;
                        WebSpeechObjektHoeheFeld[ObjektNr]              = null;
                        WebSpeechObjektLeftFeld[ObjektNr]               = null;
                        WebSpeechObjektTopFeld[ObjektNr]                = null;
                        WebSpeechObjektTextOderUrlFeld[ObjektNr]        = null;
                        WebSpeechObjektTextQuelleFeld[ObjektNr]         = null;     // true = String, false = Url
                }

                return ReturnWert;
}

// + + + + + + + + + + + WebSpeech-Objekt-Text laden als String oder aus Datei,  OHNE Sprechen
//                                      benötigt erzeugtes Objekt
function WebSpeech_TextLaden(ObjektNr, Quelle, TextOderUrl)
// Text aus Datei oder Textstring in das Objekt laden
//              liefert         true, wenn Objekt geladen wurde
//              sonst false, auch wenn Parameter falsch oder Objekt nicht installiert oder ein Text bereits geladen wird

// vor dem Laden wird eine bereits aktive Sprachausgabe nicht verändert
//                                      gemutete   Sprachausgabe nicht gestartet und nicht zu Ende geführt
//                                                              da NUR der User demuten kann

//              Es gilt für das Laden eines Textes Direkt nach Erzeugung des Objektes per
//              WebSpeech_HTMLObjektErzeugen
//                      also per Javascript-Folge

//                                      WebSpeech_HTMLObjektErzeugen(....);
//                                      WebSpeech_TextLaden(....);              aus einer DATEI

//                                              Sprachausgabe MIT dem Erzeugen erfolgt NICHT !!!!

//                                      bzw.    WebSpeech_TextLaden(....);              aus einer String

//                                      Sprachausgabe MIT dem Erzeugen erfolgt ANHAND des TEXTES laut
//                                              WebSpeech_TextLaden() !!

//                      Abhilfe kommt nur durch Einfügen eines Tastaturereignisses vor WebSpeech_TexLaden
//                                      z.B. alert();

//              ObjektNr                                        :       >=0
//              Quelle                                          :       0 so String in TextOderUrl
//                                                                      1 so Url    in TextOderUrl
//              TextOderUrl                                     :       siehe Quelle
//                                                                      Für String gilt: Kann leer sein, also nur zum
```

```
//                                                                  LÖSCHEN des
//                                                                  Objekt-Text-Puffers, damit Useraktion
//                                                                  auf Control-Leiste keine Ausgabe auslöst
//                                                                  Für Url gilt: darf nicht leer sein !!!
//                                                                          Der Dateiname ist beliebig, darf
//                                                                          aber nicht mit # beginnen
//                                                                           (wird nicht geprüft !)
//                                                                  Als Url sind z.B. möglich
//                                                                          www.test.de/test.txt
//                                                                          file:///c:/test.txt
//                                                                          test.txt
//                                                                          im gleichen Verzeichnis wie das
//                                                                                  HTML-Dokument
//                                                                          text/test.txt
//                                                                  im Unterverzeichnis zum Verzeichnis
//                                                                          des HTML-Dokumentes

{
        var Wert=-1;            // Annahme: Laden ohne Erfolg
        var ObjektZustandWert=0;

        if (WebSpeechObjektZeigerFeld[ObjektNr])
        {
                // Objektzustand ermitteln: aktualisiert WebSpeechObjektZustandFeld[], dann auslesen
                ObjektZustandWert=WebSpeech_HTMLObjektZustandErmitteln(ObjektNr);

                if (ObjektZustandWert != 6)
                {
                        // kein Text wird bereits geladen

                        // Laden aus Url oder Kette
                        if (Quelle == 1)
                        {
                                if (TextOderUrl.length > 0)
                                {
                                        // und Text aus Datei laden
                                        Wert=WebSpeechObjektZeigerFeld[ObjektNr].LoadText(
                                                        TextOderUrl, 0, 0);
                                                        // erste  0 für kein Autostart
                                                        // zweite 0 für Nichtwarten, aber Wert ist egal
                                                        //      da nur für Autostart = 1 wichtig wäre
                                        if (Wert >= 0)
                                        {
                                                // und Textquelle merken
                                                WebSpeechObjektTextQuelleFeld[ObjektNr] = false;
                                                // true = String, false = Url
                                        }
                                }
                        }
                        else
                        {
                                // Textstring laden, auch Leerkette zum Pufferlöschen
                                Wert=WebSpeechObjektZeigerFeld[ObjektNr].SetText(TextOderUrl);

                                if (Wert >= 0)
                                {
                                        // und Textquelle merken
                                        WebSpeechObjektTextQuelleFeld[ObjektNr] = true;
                                         // true = String, false = Url
                                }
                        }

                        if (Wert >= 0)
                        {
                                // und Text bzw. Url merken
                                WebSpeechObjektTextOderUrlFeld[ObjektNr] = TextOderUrl;
                        }
                }
        }

        // Objektzustand ermitteln: aktualisiert WebSpeechObjektZustandFeld[]
        //      Funktionswert egal, da nur das Zustandfeld gefüllt werden muss
        ObjektZustandWert=WebSpeech_HTMLObjektZustandErmitteln(ObjektNr);
```

```
	return (Wert >= 0);
}

// + + + + + + + + + + + WebSpeech-Objekt-Text Sprechen
//	benötigt erzeugtes Objekt und geladenen Text
function WebSpeech_AktuellGeladenenTextSprechen(ObjektNr, Warten)
// Die Funktion hat nur einen Effekt, wenn sie ein vorhergehendes Tastatur- oder Mausereignis erzeugt wurde
// eine stummtgeschaltete Sprachausgabe blockiert das Sprechen des Textes, da NUR der User demuten kann !!
// vor dem Sprechen wird eine pausierte Sprachausgabe weitergeführt, dann Verhalten laut Warten

//	liefert	true, wenn Objekt gesprochen wird
//		sonst false, wenn Parameter falsch oder Objekt nicht installiert oder stummgeschaltet
//		oder Text gerade geladen wird
//
//	ObjektNr		: >=0
//	Warten		: 0	Sprachausgabe sofort starten und bereits aktive
//		abbrechen
//		1	Sprachausgabe erst starten, wenn aktive beendet ist
{
	var Wert=-1;		// Annahme: Sprechen ohne Erfolg
	var ObjektZustandWert=0;

	if (WebSpeechObjektZeigerFeld[ObjektNr])		// Objekt hat Instanz
	{
		// Objektzustand ermitteln: aktualisiert WebSpeechObjektZustandFeld[], dann auslesen
		ObjektZustandWert=WebSpeech_HTMLObjektZustandErmitteln(ObjektNr);

		if (ObjektZustandWert < 5)
		{
			//	Sprachausgabe zum Objekt ist
			//		nicht stummgeschaltet, also neue Sprachausgabe möglich
			//		Hinweis: Aufheben der Stummschaltung nicht nur durch Javascript
			//		sondern nur durch Useraktion möglich
			//		UND  kein Text ist gerade am Laden

			// Objektzustand ermitteln: aktualisiert WebSpeechObjektZustandFeld[], dann auslesen
			ObjektZustandWert=WebSpeech_HTMLObjektZustandErmitteln(ObjektNr);

			if (ObjektZustandWert == 3)
			{
				//	pausierende Sprachausgabe weiterführen
				WebSpeechObjektZeigerFeld[ObjektNr].ResumeSpeaking();
			}

			if ( Warten == 0 )
			{
				// kein Warten, also aktive Sprache abbrechen und neuen Text sprechen
				Wert = WebSpeechObjektZeigerFeld[ObjektNr].StartSpeakingImmediate();
			}
			else
			{
				// erst warten dann Text sprechen
				Wert = WebSpeechObjektZeigerFeld[ObjektNr].StartSpeaking();
			}
		}
	}

	// Objektzustand ermitteln: aktualisiert WebSpeechObjektZustandFeld[]
	//	Funktionswert egal, da nur das Zustandfeld gefüllt werden muss
	ObjektZustandWert=WebSpeech_HTMLObjektZustandErmitteln(ObjektNr);

	return (Wert >= 0);
}

// + + + + + + + + + + + WebSpeech-Objekt-Text temporär laden und sprechen
//	alten Text retten, neuen Text laden und sprechen, alten Text laden
function WebSpeech_TextTemporaerLadenUndSprechen(ObjektNr, Quelle, TextOderUrl, Warten)
//	benötigt erzeugtes Objekt
//	ObjektNr	:	>=0
//	Quelle	:	0 so String in TextOderUrl
//		1 so Url	in TextOderUrl
//	TextOderUrl	:	siehe Quelle
//		Für String gilt: Kann leer sein, also nur zum LÖSCHEN des
//		Objekt-Text-Puffers, damit Useraktion
```

```
//                                                                    auf Control-Leiste keine Ausgabe auslöst
//                                                        Für Url gilt: darf nicht leer sein !!!
//                                                                Der Dateiname ist beliebig, darf aber nicht mit #
//                                                                beginnen
//                                                                                (wird nicht geprüft !)
//                                                        Als Url sind z.B. möglich
//                                                                www.test.de/test.txt
//                                                                file:///c:/test.txt
//                                                                test.txt
//                                                                        im gleichen Verzeichnis wie das
//                                                                                        HTML-Dokument
//                                                                text/test.txt           im Unterverzeichnis zum
//                                                                                        Verzeichnis
//                                                                                        des HTML-Dokumentes
//          Warten                                  : 0     Sprachausgabe sofort starten und bereits aktive
//                                                                  abbrechen
//                                                    1     Sprachausgabe erst starten, wenn aktive beendet ist
{
        var ReturnWert=false;

        // retten der aktuellen Textquelle und des Textes
        var RetteTextQuelle  = WebSpeechObjektTextQuelleFeld[ObjektNr];
        var RetteTextOderUrl = WebSpeechObjektTextOderUrlFeld[ObjektNr];

        // Text laden
        ReturnWert= WebSpeech_TextLaden(ObjektNr, Quelle, TextOderUrl);
        if (ReturnWert)
        {
                // und sprechen
                ReturnWert=WebSpeech_AktuellGeladenenTextSprechen(ObjektNr, Warten)
                if (ReturnWert)
                {
                        // und geretten Text wieder laden, danach ist leider Text nicht sprechbar
                        ReturnWert=WebSpeech_TextLaden(ObjektNr, RetteTextQuelle, RetteTextOderUrl);
                }
        }

        return ReturnWert;
}

// +++++++++++++ Browser-Typ ermitteln
// Dieser Quellcode muss VOR allen anderen Routinen codiert sein, damit zuerst abgearbeitet
var ns = document.layers ? true : false; // NS 4.x
var ie = document.all ? true : false;

// +++++++++++++++++++ auf Webspeech-Objekt erzeugen prüfen
var WebSpeechInstalliert = WebSpeechInstallationPruefen();
        //          0  für Webspeech-Plugin ist nicht installiert
        //                          im Browser (egal ob NS oder IE)
        //                  oder IE  bzw. NS nicht aktiv
        //          > 0  für Webspeech-Plugin ist installiert im NS 4.x
        //                          und NS 4.x aktiv
        //          < 0  für Webspeech-Plugin ist installiert im IE
        //                          und IE aktiv

var WebSpeech_MenuAnsage_Text =  'das ist ein test';

if (WebSpeechInstalliert !=0)
{
        WebSpeech_TextTemporaerLadenUndSprechen(        WebSpeechStandardObjektNr,
                                                        0,
                                                                WebSpeech_MenuAnsage_Text,
                                                        1
                                                );
}
```

5.3.3. Webspeech-Sprechtags (Auswahl)

5.3.3.1. Webspeech-Sprechtags unter Webspeech 4

Tags tauchen im Text auf, der gesprochen werden soll
werden in \ \ kodiert
Fettmarkiertes sind Plichtkodierungen

Lautstärke: relativ zur aktuellen Lautstärkeeinstellung laut Windows-Lautstärke-Regelung
VOL=value\ value Integer

			0 bis 65535 0 stumm 65535 maximal
Pause	**\PAU=**value\	value	Integer in Millisekunden ab 0
Reset	von Sprechgeschwindigkeit und Lautstärke **\RST**		
Sprechgeschwindigkeit	**\SPD=**value\	value	Integer Anzahl der Worte pro Minute 80 bis 160
Sprecherwechsel	**MOD=**value\	value	Integer 0 Johanna 1 Martin 2 Bill 3 Silke ab 4 nur wenn weitere installierte Stimmen vorhanden sind

5.3.3.2. Webspeech-Sprechtags in Webspeech 4

Sprechtags aus Webspeech-Versionen unter 4 werden dann nicht verarbeitet, wenn
- Webspeech 4 einen anderen Syntax hat (außer \, das als # interpretiert wird)
- Wert nicht in der Schnittmenge der Wertebereiche von Webpseech 4 und der Vorgängerversionen liegt
- Einheit des Wertes nicht in Webspeech 4 bekannt ist bzw. fehlt, aber in Webspeech 4 kodiert werden muss

Tags tauchen im Text auf, der gesprochen werden soll
- werden in # # kodiert

Fettmarkiertes sind Plichtkodierungen

Lautstärke relativ zur aktuellen Lautsärke laut Stimmvorgabe

	#VOL=value**dB#**	value	Floating Point in Dezibel -100 bis +12 Kleinste einheit ist 0.1
	#VOL=value**%#**	value	Integer in Prozent 10 bis 400 100 entspricht 0dB 200 entspricht +6db 50 entspricht -6dB
Pause	**#PAU=**value**#**	value	Integer in Millieksunden 0 bis 5000

Rücksetzen von Sprechgeschwindigkeit und Lautstärke

#RST#

Sprechgeschwindigkeit relativ zur aktuellen Geschwindigkeit laut Stimmenvorgabe

	#SPD=value**%#**	value	Integer in Prozent 10 bis 400
Stimmwechsel	**#VOICE=**xxx**.**yy**.**zzzzzzzz**#**	xxx	Herstellerkürzel 3 Zeichen z.B. LGX
		yy	Sprache 2 Zeichen z.B. DE
		zzzzzzzz	Stimmname max. 8 Zeichen z.B. BILL

Punkte müssen kodiert werden

Standardstimmen sind

LGX.DE.BILL
LGX.DE.SABINE
LGX.DE.MANDY
LGX.DE.KATHI
LGX.DE.SILKE
LGX.DE.MICHAEL
LGX.DE.MARTIN

Text Text buchstabieren (Zahlen ziffernweise)

#CTX=SPELL# Einschalten des Buchstabierens
#CTX# Abschalten des Buchstabierens

Wort Betonung für Wort abschalten
#NOACC# vor dem Wort kodieren

Wort fremdes Wort der Sprachausgabe als Wortart bekanntgeben Variante 1

#POS=kette# vor dem Wort kodieren

kette	NOUN	Substantiv	
	NAME	Eigenname	
	VERB	Verb	
	ADJ	Adjektiv	
	NUMB	Zahlwort	
	PRON	Pronomen	z.B. sie
	ART	Artikel	
	PREP	Präposition	z.B. nach
	ADV	Adverb	z.B. damals
	CONJ	Konjunktion	z.B. weil, und
	PART	Partikel	z.B. vielleicht, schon
	INTJ	Interjektion	z.B. Ach! als Ausruf

Wort fremdes Wort der Sprachausgabe als Wortart bekanntgeben Variante 2

#FW# vor dem Wort kodieren

Dauer der Laute verkürzen wenn Wort ist

Präposition	z.B. nach
Artikel	
Konjunktion	z.B. weil, und

Wort Vokal eines nachfolgenden Wortes länger sprechen und Wort mit Betonung sprechen

#EMPH# vor dem Wort kodieren

Wortfolge Betonung abschalten bis Satzende

#LASTACC# ab Wort dahinter bis Satzende

5.4. Beispiel für windowseigenes Active-X-Control – Analoge Uhr

Microsoft ändert fortlaufend die Active-X-Eigenschaften von Windows und somit auch des Internet Explorers

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.
Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.
Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Linzensierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks
Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.
Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.
Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2).
Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.
Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

Hinweis: Man suche doch mal im Internet nach einem kostenlosen HTTP-Server vom Microsoft, um IE-Seite testen zu können, die JScript nutzen (inklusive Debugger). Denn sollte kein kostenloses Angebot findbar sein, kommen die Kosten von Entwicklungssoftware zum IE hinzu. Ein Blick auf Konkurrenz-HTTP-Server klärt die Sachlage: Apache-HTTP-