



WebSpeechSDK

Deutsche Sprachausgabe für Windows 95/98, NT, 2000, XP

Inhaltsverzeichnis

| | |
|--|---------------|
| Inhaltsverzeichnis | 2 |
| Allgemeiner Teil | 5 |
| 1 Copyright | 5 |
| 2 Dieses Buch | 6 |
| 3 Sprechende Internetseiten? | 7 |
| 4 Einführung | 8 |
| 4.1 Funktionsweise von WebSpeech | 8 |
| 4.2 Funktionsweise der Logox-Sprachausgabe | 8 |
| Abschnitt I | 9 |
| Die Installation von WebSpeech | 9 |
| 1 Download | 10 |
| 2 Installation | 11 |
| Abschnitt 2 und Internetseiten sprechen! | 13 |
| 1 WebSpeech starten | 14 |
| 2 Bedienung von WebSpeech | 15 |
| 3 WebSpeech Steuerfunktionen | 16 |
| 4 PopUp-Fenster | 18 |
| 5 Speechfont wählen... | 20 |
| 6 Einstellungen... | 22 |
| 7 Textbox... | 23 |
| Abschnitt 3 | 24 |
| Die WebSpeech-Programmierung | 24 |
| 1 Lizenzschlüssel | 24 |
| 2 Ein Text | 25 |
| 3 Automatisches Starten der Sprachausgabe | 25 |
| 4 Animation des Mundes | 25 |
| 5 Zusammenfassung | 26 |
| 6 Parameter | 27 |
| 6.1 AUTHKEY | 28 |
| 6.2 AUTOSTART | 29 |
| 6.3 IMMEDIATE | 30 |
| 6.4 TEXT | 31 |
| 6.5 URL | 32 |
| 6.6 MOUTHANIMATION | 33 |
| 6.7 MOUTHCOLOR | 34 |
| 6.8 TEXTANIMATION | 35 |
| 6.9 TEXTCOLOR | 36 |
| 6.10 TEXTPOSITION | 37 |
| 6.11 TEXTSIZE | 38 |
| 6.12 OPAQUE | 39 |
| 6.13 BACKGROUND COLOR | 40 |
| 6.14 CONTROLPOSITION | 41 |

| | |
|--|-----------|
| Abschnitt 4 | 42 |
| 1 Einbindung in Skriptsprachen | 42 |
| 2 Methoden | 44 |
| 2.1 SetText | 45 |
| 2.2 LoadText | 46 |
| 2.3 StartSpeaking | 47 |
| 2.4 StartSpeakingImmediate | 48 |
| 2.5 StopSpeaking | 49 |
| 2.6 PauseSpeaking | 50 |
| 2.7 ResumeSpeaking | 51 |
| 2.8 IsSpeaking | 52 |
| 2.9 IsPause | 53 |
| 2.10 IsMuting | 54 |
| 2.11 IsLoading | 55 |
| 2.12 SetMouthAnimation | 56 |
| 2.13 SetMouthColor | 57 |
| 2.14 SetTextAnimation | 59 |
| 2.15 SetTextColor | 60 |
| 2.16 SetTextPosition | 61 |
| 2.17 SetTextSize | 62 |
| 2.18 SetOpaque | 63 |
| 2.19 SetBackgroundColor | 64 |
| 2.20 SetControlPosition | 65 |
| 2.21 SetKey | 66 |
| 2.22 GetVersion | 67 |
| 2.23 GetInterfaceVersion | 68 |
| 2.24 GetUserName | 69 |
| 2.25 GetUserGreeting | 70 |
| 2.26 GetUserGender | 71 |
| 3 Events | 72 |
| 3.1 OnStartSpeaking | 74 |
| 3.2 OnStopSpeaking | 75 |
| 3.3 OnPauseSpeaking | 76 |
| 3.4 OnResumeSpeaking | 77 |
| 3.5 OnMark | 78 |
| 3.6 OnClick | 79 |
| 3.7 OnMouthOver | 80 |
| 3.8 OnWord | 81 |
| 3.9 OnLoad | 82 |
| Abschnitt 5 | 83 |
| Steuerung der Sprachausgabe in Texten und Textdateien | 83 |
| 1 LGX-Dateien - das erste WebSpeech Format | 84 |
| 2 Proprietäre 4 Speechtags | 85 |
| 3 Syntax und allgemeine Eigenschaften | 87 |
| 3.1 BSPD | 88 |
| 3.2 SPD | 89 |
| 3.3 BVOL | 90 |
| 3.4 VOL | 91 |
| 3.5 BPIT | 92 |
| 3.6 PIT | 93 |

| | | |
|--------------------|---------------------------------------|------------|
| 3.7 | MIDI | 94 |
| 3.8 | BINT | 95 |
| 3.9 | INT | 96 |
| 3.10 | BROUGH | 97 |
| 3.11 | ROUGH | 98 |
| 3.12 | RST | 99 |
| 3.13 | PAU | 100 |
| 3.14 | VOICE | 101 |
| 3.15 | EMPH | 102 |
| 3.16 | POS | 103 |
| 3.17 | FW | 104 |
| 3.18 | CTX | 105 |
| 3.19 | MARK | 106 |
| 3.20 | COM | 107 |
| 3.21 | Benutzer persönlich ansprechen | 108 |
| 3.22 | AFX | 109 |
| 4 | Ausführliche Beschreibung der Effekte | 110 |
| 4.1 | Balance | 110 |
| 4.2 | Chorus | 111 |
| 4.3 | Delay | 112 |
| 4.4 | Grafischer Equalizer | 113 |
| 4.5 | Flanger | 114 |
| 4.6 | Gain | 115 |
| 4.7 | Phaser | 116 |
| 4.8 | Reverb | 117 |
| 4.9 | Shelving Filter | 118 |
| 5 | SAMPA | 119 |
| 5.1 | Überblick | 119 |
| 5.2 | Ausführliche Beschreibung | 121 |
| 5.3 | Werte für die SAMPA-Umschrift | 122 |
| 5.4 | Beispiele | 131 |
| 5.5 | LASTACC | 132 |
| 5.6 | NOACC | 133 |
| 5.7 | NOIP | 134 |
| 5.8 | NOAP | 135 |
| 5.9 | NORP | 136 |
| 6 | Sprechmelodie | 137 |
| 6.1 | Einführung | 137 |
| Abschnitt 6 | | 141 |
| 1 | Tools | 141 |
| 2 | AudioFXEditor | 142 |
| 2.1 | Effekte erstellen und anordnen | 142 |
| 2.2 | Effekte editieren | 143 |
| 2.3 | Der resultierende AudioFX String | 143 |
| 3 | Anhang | 144 |
| 3.1 | Tabelle der Farbnamen | 144 |
| 3.2 | Index | 148 |

Allgemeiner Teil

I Copyright

Alle Rechte vorbehalten. Kein Teil dieses Buches darf ohne schriftliche Genehmigung der G DATA Software AG reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Die in diesem Buch verwendeten Markennamen sowie Software- und Hardwarebezeichnungen unterliegen im Allgemeinen dem warenzeichen-, marken- oder patentrechtlichen Schutz.

2 Dieses Buch

Vielen Dank für Ihr Interesse an WebSpeech Site Development Kit (SDK). Dieses Buch zeigt Ihnen in Abschnitt 2, wie Sie WebSpeech als Besucher einer Webseite bedienen können. Im dritten und vierten Abschnitt wird erklärt, wie Sie die Technologie von WebSpeech auf Ihren Internetseiten nutzen können. Sie erfahren, wie Sie Ihren HTML-Code aufbauen müssen, damit Ihre Webseite sprechen kann. Außerdem wird die Verwendung der Speechtags im zu lesenden Text erklärt. Damit kann man z.B. die Aussprache korrigieren, Sprecher bestimmen, Effekte ein- und ausschalten, Pausen einfügen, die Sprechmelodie verändern usw. Im Anhang, in den einzelnen Kapiteln und auf der beiliegenden CD sind die Tags und Funktionen durch Beispiele erläutert.

Folgende Kenntnisse werden vorausgesetzt:

- HTML
- Aktive Elemente auf HTML-Basis
- Grundwissen JavaScript

In diesem Handbuch kann keine Einführung in HTML und die Verwendung und Programmierung von Skriptsprachen erfolgen.

3 Sprechende Internetseiten?

WebSpeech ermöglicht es, Texte in Internetseiten sprechen zu lassen, die unabhängig vom dargestellten Inhalt der Seite sind. Dadurch wird ein neuer Kommunikationskanal bereitgestellt, der die grafische Gestaltung ergänzen oder ersetzen kann. Mit WebSpeech ist es möglich, die Besucher der Seite zu begrüßen, während sich die Seite aufbaut. Gerade mit moderner und allzeit präsenter Sprachausgabe ist die Interaktivität und Interaktion zwischen Anbietern und Interessenten realisierbar, die für eine allgemeine Akzeptanz des E-Commerce unverzichtbar ist. Man kann Newstickern, Chats und Avataren eine Stimme geben, Bilder und Tabellen kommentieren und Formularabfragen freundlicher gestalten. Es lassen sich ausführliche Anleitungen oder Kommentare geben, ohne das grafische Layout zu (zer-)stören. Es können Informationen (z.B. aus einer Datenbankabfrage) bereitgestellt werden, ohne Seiten neu aufzubauen. Last not least erhalten Sie die Möglichkeit, die Aufmerksamkeit Ihrer Besucher gezielt auf die Dinge zu lenken, die Ihnen wichtig sind - unüberhörbar.

Aber - gehen Sie sorgfältig mit dem neuen Kommunikationskanal um. Er sollte nicht überfrachtet werden. Auch eine mit Grafiken überladene Seite schreckt eher ab. Der Einsatz der gesprochenen Sprache sollte genauso bedacht werden wie die grafische Gestaltung der Webseite.

4 Einführung

Bei WebSpeech handelt es sich um ein PlugIn für Internet Explorer („IE“), das die Logoxtechnologie zur Sprachausgabe für Webseiten zur Verfügung stellt. Damit man durch WebSpeech unterstützte Seiten nutzen kann, muss WebSpeech installiert sein. Die Basisfunktionen von WebSpeech stehen ab Version 4.0 des Internet Explorers zur Verfügung.

4.1 Funktionsweise von WebSpeech

Im HTML-Code Ihrer Webseite bestimmen Sie einen Text, der gesprochen werden soll. Dieser Text wird entweder mit dem HTML-Dokument oder in einer separaten Textdatei vom Server zum Rechner des Besuchers der Webseite übertragen. Bestimmte Steuerzeichen im HTML-Code rufen das PlugIn „WebSpeech“ auf, das den übertragenen Text basierend auf der Technologie von Logox in Sprache umwandelt und über die Soundkarte ausgibt. Da die Textdateien sehr klein und daher schnell übertragen sind, kann die Webseite schon sprechen, bevor sie komplett aufgebaut ist.

4.2 Funktionsweise der Logox-Sprachausgabe

Logox ist eine deutsche Sprachausgabe nach dem konkatenativen Prinzip der Mikrosegmentsynthese. Dabei werden kleine Abschnitte von Sprachaufnahmen eines Sprechers so aneinander gehängt, dass man daraus jede beliebige Lautsequenz erzeugen kann. Diese Abschnitte sog. Mikrosegmente sind so ausgewählt, dass sie einen sparsamen Umgang mit Rechnerressourcen wie Speicherplatz und Rechenleistung ermöglichen.

Der Text wird zunächst in einzelne Wörter zerteilt und anschließend Wort für Wort weiterverarbeitet. Durch spezielle Lexika wird jedem Wort eine Aussprache zugewiesen. Weitere Informationen zur den Wörtern erlauben eine Analyse der Satzstruktur und es kann ein Betonungsverlauf erzeugt werden. Anschließend werden die „Ausspracheformen“ durch ein Regelwerk in eine Folge von Mikrosegmenten umgesetzt. Den Mikrosegmentfolgen werden die jeweiligen Mikrosegmentdateien zugeordnet und deren Dauer und Frequenz bestimmt. Das resultierende Audiosignal durchläuft die Echtzeiteffekte und wird anschließend ausgegeben.

Abschnitt I

Die Installation von WebSpeech

Damit der Anwender auf die Fähigkeiten von WebSpeech zurückgreifen kann, muss er es natürlich auch installieren. WebSpeech ist das kostenlose PlugIn für den Internet Explorer, welches dem Anwender ermöglicht, Webseiten zu hören, die mit WebSpeech SDK aufbereitet wurden.

WebSpeech findet sich als kostenlose Freeware auf diversen Beilage-CD-ROMs von Internet- und Computerzeitschriften, bei G DATA-Produkten und nicht zuletzt auf der WebSpeech Homepage

<http://www.webspeech.de/>

I Download

WebSpeech finden Sie als kostenlosen Download auf vielen Freeware-Sammlungen und auf „<http://www.webspeech.de/download.php>“. Die Download-Größe liegt bei ca. 5,7 MB.

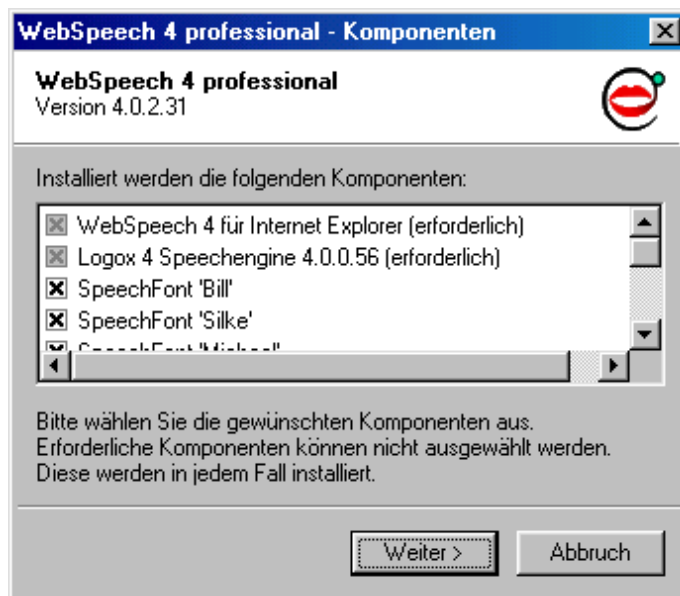
Die Download-Datei enthält den Sprachsynthesekern, alle Lexika und eine Reihe von SpeechFonts[®], die auf 4 Grundstimmen basieren. Weitere SpeechFonts[®] stehen zum Download bereit.

Die Datei, die Sie nach dem Download auf Ihrer Festplatte gespeichert haben, ist ein Archiv, welches sich selbst entpackt und den Setup-Vorgang für WebSpeech einleitet, sobald Sie es mit der Maus doppelklicken.

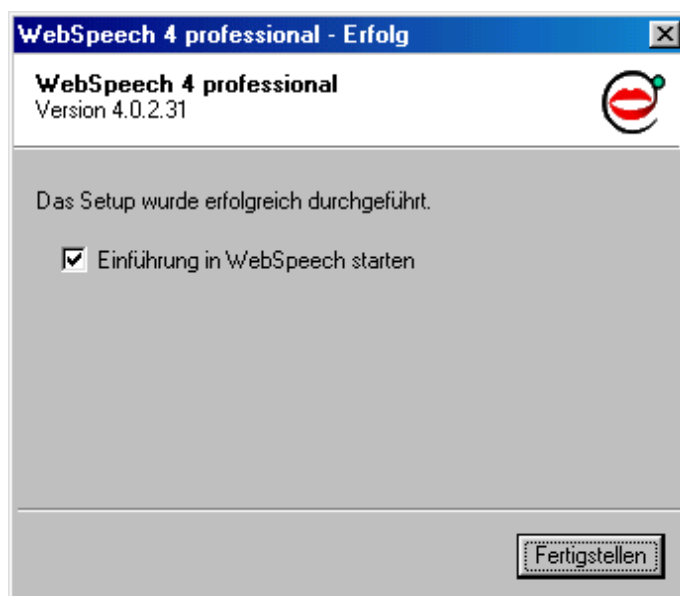
2 Installation

Doppelklicken Sie die Datei „webspeech4.exe“, um das Setup zu beginnen. Bevor das Setup startet, entpackt WebSpeech automatisch die Programmdateien, die zur Ersparnis von Speicherplatz und Downloadzeit komprimiert wurden.

Bitte beenden Sie alle Windows-Programme - insbesondere den Internet Explorer und den Explorer - , bevor Sie das Installationsprogramm ausführen.



WebSpeech sucht nun automatisch den Internet Explorer (ab Version 4) und installiert sich dort als PlugIn. Die PlugIn-Installation greift dabei nicht in den eigentlichen Programmaufbau der Browser ein und ist damit gefahrlos durchführbar. WebSpeech prüft, ob bereits SpeechFonts auf Ihrem Rechner vorhanden sind. Dieser Vorgang kann einige Minuten dauern.



WebSpeech steht Ihnen nach der Installation sofort zur Verfügung. Sollten einige Systemaktualisierungen nicht durchführbar gewesen sein (, da nicht alle Programme geschlossen waren), werden Sie zu einem Neustart aufgefordert. Nach dem Neustart stehen die aktualisierten Programmbibliotheken und Dateien von WebSpeech zur Verfügung.



WebSpeech ist nun voll einsatzbereit und bietet für Internet-Seiten, die mit WebSpeech SDK erstellt wurden, eine uralte und doch neue Dimension der Kommunikation an: die der menschlichen Sprache!

Abschnitt 2

.... und Internetseiten sprechen!

Für den Microsoft Internet Explorer ab Version 4.0 steht Ihnen ein zusätzliches Feature bei dem Logox® WebSpeech Plugin zur Verfügung:

- die Vorlesefunktion für Internetseiten.

Mit diesem Tool können Sie sich alle im Internet Explorer aufgerufenen Websites vorlesen lassen - komplett, ausgewählte Textabschnitte oder markierten Text.



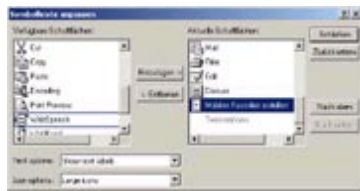
I WebSpeech starten

Um das WebSpeech - Fenster im unteren Teil des Browsers zu öffnen, stehen Ihnen folgende Möglichkeiten zur Auswahl:

1. Wählen Sie aus der Menüleiste des Internet Explorers den Menüpunkt "Ansicht" und dann unter "Explorerleiste" den Eintrag "WebSpeech Reader".
2. Drücken Sie die Funktionstaste "F8" (Point&Read) oder
3. Drücken Sie die Funktionstaste "F9" (Seite/Markierung vorlesen).



Ab dem Internet Explorer 5.0 erscheint zusätzlich das WebSpeech-Icon in der Symbolleiste "Standard-Schaltflächen" des Internet Explorers und der Menüeintrag "Seite/Markierung vorlesen (WebSpeech)" unter dem Menüpunkt "Extras".



Sollte das WebSpeech-Icon nicht in Ihrer Symbolleiste auftauchen, klicken Sie mit der rechten Maustaste in die Symbolleiste und wählen Sie "Anpassen...". Die linke Auswahl zeigt die verfügbaren Schaltflächen; markieren Sie das WebSpeech-Symbol und klicken Sie auf "Hinzufügen". Schließen Sie anschließend das Dialogfenster.

2 Bedienung von WebSpeech



Seite/Markierung vorlesen

Mit "F9" wird der Text der aktuell ausgewählten Internetseite vorgelesen. Hierbei wählt WebSpeech den größten Textblock aus. Wenn Sie nur einen bestimmten Bereich aus dieser Webseite vorgelesen haben möchten, markieren Sie diesen einfach mit der Maus und drücken dann auf "F9".



Point&Read

Mit Point&Read können Sie den zu sprechenden Text noch komfortabler auswählen. Klicken Sie dazu auf die Taste "F8". Es erscheint ein Mauszeiger mit einem kleinen Lautsprecher-Symbol. Mit diesem Zeiger klicken Sie nun auf den Textabschnitt der Webseite, den Sie hören möchten. Logox® WebSpeech liest Ihnen dann automatisch diesen Abschnitt vor. Mit dieser Funktion ist es auch möglich, sich einen Text vorlesen zu lassen, der noch gar nicht angezeigt wird: Wenn Sie mit dem modifizierten Mauszeiger auf einen beliebigen Hyperlink klicken, liest Logox® WebSpeech Ihnen den Inhalt der verlinkten Webseite vor, ohne die aktuelle Seite zu verlassen.

3 WebSpeech Steuerfunktionen



Mit dem Start-Button können Sie das Vorlesen der angezeigten Internet-Seite starten. Diese Funktion ist identisch mit der Funktionstaste "F9".



Beenden Sie mit einem Klick auf den Stop-Button die aktive Sprachausgabe. Alternativ können Sie die Sprachausgabe mit der ESC-Taste beenden.



Klicken Sie das Pause-Symbol an, wenn Sie eine längere Sprechpassage unterbrechen wollen. Ein erneuter Klick lässt Logox® WebSpeech später an derselben Stelle weiter lesen.



Drehen Sie an dem Rädchen, um die Lautstärke der Sprachausgabe zu regulieren.



Die Lautstärke können Sie auch einfach mit Drag & Drop an den blauen Balken links neben dem Startbutton einstellen.



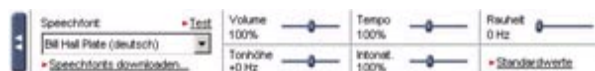
Point&Read-Funktion

Dieser Button entspricht der Point&Read-Funktion der Taste "F8". Wenn Sie auf diesen Button klicken, erscheint ein Mauszeiger mit einem kleinen Lautsprecher-Symbol. Mit diesem Zeiger klicken Sie nun auf den Textabschnitt der Webseite, den Sie hören wollen. Mit dieser Funktion ist es auch möglich, sich einen Text vorlesen zu lassen, der noch gar nicht angezeigt wird: Wenn Sie mit dem modifizierten Mauszeiger auf einen beliebigen Hyperlink klicken, liest der Logox® WebSpeech Ihnen den Inhalt der verlinkten Webseite vor, ohne die aktuelle Seite zu verlassen.



SpeechFont® Auswahl

Wenn Sie den Button zur Steuerung der Sprechweise anklicken, erscheint im rechten Teil des Fensters ein Dialog, der Ihnen die Einstellung von sprecherspezifischen Eigenschaften sowie des Sprechers selbst gestattet:



Unter "Speechfonts" sehen Sie alle installierten Stimmen und können die Stimme bestimmen, mit der WebSpeech arbeiten soll. Klicken Sie auf Test im Dialog, um sich die veränderte Stimme anzuhören. Durch einen Klick auf Standardwerte erhalten alle Eigenschaften Parameter ihren Ausgangswert. Möchten Sie weitere SpeechFonts aus dem Internet installieren, klicken Sie auf Speechfonts installieren. Mit dem Button links oder einem erneuten Klick auf das SpeechFont-Symbol im Bedienfeld schalten Sie zur Standard-Ansicht zurück.



WebSpeech Optionen

Wenn Sie den Button "Optionen" anklicken, erscheint im rechten Teil des Fensters ein Dialog, der Ihnen die Einstellung einiger Parameter von WebSpeech gestattet:



Die Vorlesemenge ist ein grober Richtwert, wieviel WebSpeech von einer Webseite vorliest. Es sollen ja keine Verknüpfungsmenüs, bzw. Werbeeinblendungen vorgelesen werden, sondern nur der Teil einer Webseite auf den es ankommt.



Online-Hilfe

Der Hilfe-Button öffnet im aktuellen Browserfenster die Online-Hilfe.

Popup-Menu

Weitere Einstellungen und Optionen, erreichen Sie durch einen Klick auf das WebSpeech-Logo. Wählen Sie in dem erscheinenden Popup-Fenster die gewünschte Funktionalität aus.



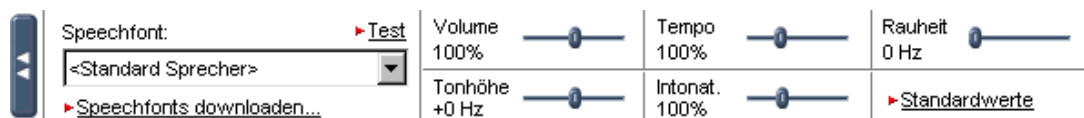
Gehen Sie jetzt ggf. online und geben Sie in die Adressleiste eine beliebige Internet-Adresse (z.B. <http://www.webspeech.de>) ein. Lassen Sie sich von nun an ganze Internetseiten oder ausgewählte Textabschnitte von WebSpeech vorlesen oder schauen Sie sich hier an, was mit WebSpeech alles möglich ist. Vielleicht besuchen Sie auch das WebSpeech Forum.

4 PopUp-Fenster

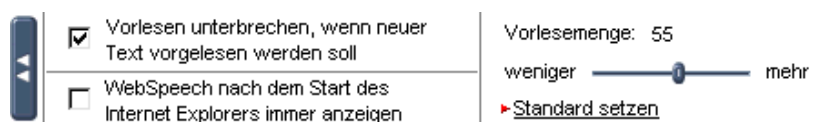
Im PopUp-Fenster stehen Ihnen folgende Funktionen zur Verfügung:



- **Seite/Markierung vorlesen**
Diese Funktion ist identisch mit der Vorlesefunktion, die Sie über die "F9"-Taste starten können. Lesen Sie hierzu bitte auch das Kapitel "Bedienung von WebSpeech".
- **Point & Read auswählen**
Diese Funktion ist identisch mit der Point & Read-Version, die Sie über die "F8"-Taste starten können. Lesen Sie hierzu bitte auch das Kapitel "Bedienung von WebSpeech".
- **Vorlesen beenden**
Über diese Funktion können Sie den Vorlesevorgang jederzeit abbrechen.
- **Speechfont wählen...**
Rufen Sie hiermit das Einstellungsmenü für die Speechfonts aus. Hier können Sie auch neue Speechfonts aus dem Internet herunterladen und vorhandene nach individuellen Wünschen modifizieren. Lesen Sie hierzu bitte auch das Kapitel "Speechfont wählen".




- **Einstellungen...**
Über den Einstellungen-Bereich können Sie grundlegende Vorleseigenschaften verändern. Lesen Sie hierzu bitte auch das Kapitel "Einstellungen".



- **Textbox...**
Über die Textbox können Sie beliebigen Text direkt im Browser eingeben und vorlesen lassen. Lesen Sie hierzu bitte auch das Kapitel "Textbox".



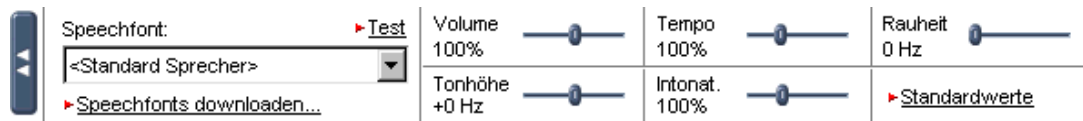
- **Hilfe...**
Rufen Sie hiermit die Online-Hilfe von WebSpeech im Browser auf.
- **Versionsbox...**
Hier erhalten Sie Informationen zur Versionsnummer von WebSpeech und seinen Bestandteilen.

| | | | |
|---|---|-------------------|----------|
|  | WebSpeech 4.0 | Webspeech Reader: | 4.0.2.31 |
| | (C) Copyright G DATA Software AG | Logox 4 Server: | 4.0.0.56 |
| | ► http://www.gdata.de | Logox 4 DLL: | 4.0.0.56 |

- **Schließen**
Hiermit schließen Sie das WebSpeech-Browserfenster. Die Vorlesefunktionen über die Tasten "F8" und "F9" stehen Ihnen jedoch weiterhin zur Verfügung.

5 Speechfont wählen...

Über das Einstellungsmenü für die Speechfonts können Sie auch neue Speechfonts aus dem Internet herunterladen und vorhandene nach individuellen Wünschen modifizieren.



- **Speechfont**

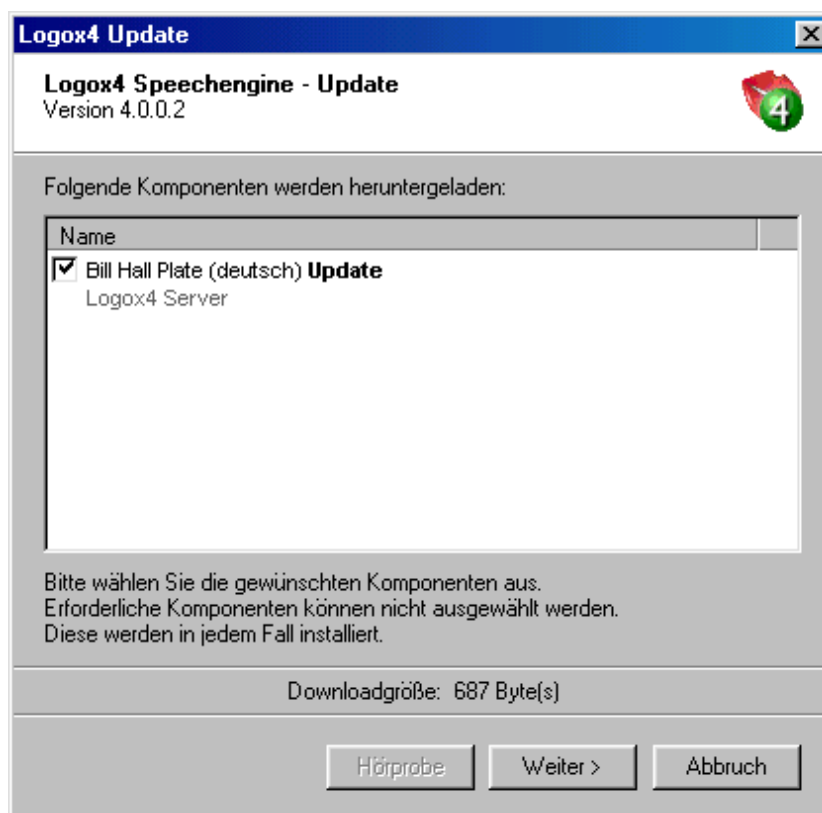
Wählen Sie hier einfach aus der Liste der vorhandenen Sprecher den aus, den Sie zum Vorlesen verwenden möchten.

- **Test**

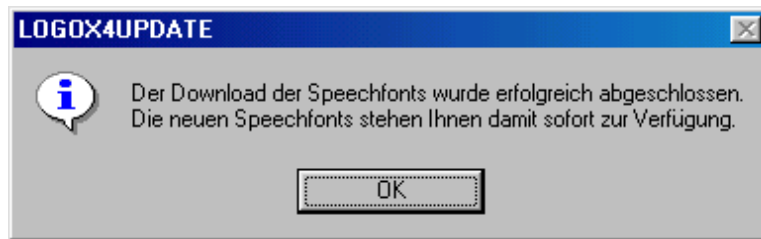
Über das Anklicken dieser Funktion können Sie sich schnell und einfach die ausgewählte Stimme mit einem Testsatz anhören, um zu entscheiden, ob Sie diese verwenden möchten oder nicht.

- **Speechfonts downloaden**

WebSpeech kann automatisch neue Speechfonts vom G DATA-Server herunterladen und in Ihr bestehendes WebSpeech-System integrieren. Sie müssen nur dafür sorgen, dass eine Online-Verbindung zur Verfügung steht. Nach Anklicken von "Speechfonts downloaden" erscheint ein Fenster mit neuen Komponenten, die Sie aus dem Internet herunterladen können. Wählen Sie die gewünschten Komponenten durch Anklicken des jeweiligen Häkchenfeldes aus und klicken dann auf "Weiter".



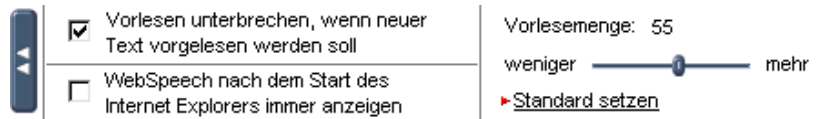
Der Download und die Installation verlaufen nun vollautomatisch und mit Anklicken des "OK"-Buttons in der erscheinenden Infobox ist der Vorgang schnell und unkompliziert abgeschlossen. Der neue Speechfont steht Ihnen sofort zur Verfügung.



- **Volume**
Legen Sie hier die relative Lautstärke der Stimme im Vergleich zu anderen Windows-Soundanwendungen fest. Sie haben hier die Auswahl zwischen 0 % (stumm) und 200 % (sehr laut).
- **Tonhöhe**
Variieren Sie hier die Tonhöhe des aktuell ausgewählten Speechfonts. Sie haben hier die Auswahl zwischen -100 Hz (tiefe Stimmlage) und + 100 Hz (hohe Stimmlage).
- **Tempo**
Verändern Sie hier das Sprechtempo. Sie haben hier die Auswahl zwischen 10 % (sehr langsam) und 300 % (sehr schnell).
- **Intonation**
Über die Intonation können Sie die ausgewählte Stimme mit einer Satzmelodie ausstatten, die gerade beim Vorlesen langer Textpassagen das Zuhören erleichtert. Sie haben hier die Auswahl zwischen 0 % (keine Intonation, monoton) und 300 % (starke Satzmelodie).
- **Rauheit**
Über die Rauheit akzentuieren Sie die Sprachausgabe und erzeugen sehr individuelle Stimmen. Sie haben dabei die Auswahl zwischen 0 Hz (keine Rauheit) und 100 Hz (starke Rauheit; Reibeisenstimme).
- **Standardwerte**
Über diese Funktion können Sie sämtliche veränderten Parameter eines Speechfonts wieder auf ihre Ausgangswerte zurückstellen.

6 Einstellungen...

Über den Einstellungen-Bereich können Sie grundlegende Vorleseigenschaften verändern.



- Vorlesen unterbrechen, wenn neuer Text vorgelesen werden soll**
 Über diese Funktion können Sie festlegen, ob WebSpeech den Vorlesevorgang unterbricht, wenn ein anderer Textblock zum Lesen markiert wurde oder ob der erste Vorlesevorgang erst abgeschlossen werden soll.
- WebSpeech nach dem Start des Internet Explorers immer anzeigen**
 Auf diese Weise wird der Internet Explorer immer automatisch mit der WebSpeech-Leiste gestartet. Wenn Sie diese Funktion nicht aktivieren, wird die WebSpeech-Leiste immer nur dann angezeigt, sobald Sie WebSpeech nutzen, z.B. über die "F8"- oder "F9"-Taste.
- Vorlesemenge**
 Über diese Funktion können Sie festlegen, wieviel Text vorgelesen werden soll, wenn Sie WebSpeech auf einer Website mit "F9" starten. Hier spielen Erfahrungswerte eine große Rolle. Während bei Vorlesemenge "0" gar kein Text vorgelesen wird, liest WebSpeech bei der Vorlesemenge "100" die gesamte Website inklusive sämtlicher Links und Werbeblöcke vor. Ein Mittelwert von 55 sorgt z.B. dafür, dass nur große Textblöcke einer Website vorgelesen werden.
- Standard setzen**
 Hiermit stellen Sie die Vorlesemenge wieder auf den Standardwert "55" ein, der in der Regel mit vielen textbasierten Websites gute Ergebnisse liefern sollte.

7 Textbox...

Über die Textbox können Sie beliebigen Text direkt im Browser eingeben und vorlesen lassen.



The screenshot shows a web interface for the WebSpeechSDK. On the left, there is a vertical blue button with two white arrows pointing right. To its right is a text input field with the label 'Texteingabe:' above it. Inside the input field is the placeholder text 'Geben Sie hier einen beliebigen Text ein.'. To the right of the input field is a small grey button with a right-pointing arrow. Further to the right is a link labeled 'Text vorlesen' with a red arrow icon.

Tippen Sie einfach den gewünschten Text ins Fenster ein und klicken dann auf "Text vorlesen". WebSpeech liest nun den eingegebenen Text automatisch vor.

Der Text darf maximal 200 Zeichen enthalten. Bei WebSpeech Pro ist die Textmenge nicht eingeschränkt.

Abschnitt 3

Die WebSpeech-Programmierung

In diesem Abschnitt wird ein erster Einblick in die Ansteuerung von WebSpeech in HTML gegeben. Die wichtigsten Parameter werden genannt und kurz besprochen. Ausführliche Beschreibungen der einzelnen Parameter finden Sie in diesem Abschnitt. Die Einbindung in Skriptsprachen wird in darauffolgenden Abschnitten beschrieben.

WebSpeech wird im Internet Explorer über das HTML-Tag <OBJECT> angesprochen. Der Netscape Communicator wird nicht unterstützt.

```
<OBJECT
  ID="WebSpeech"
  CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
  WIDTH="150"
  HEIGHT="125"
>
</OBJECT>
```

Diese Tags stehen im „BODY“ des HTML-Dokuments in Tabellen, in Layern oder in Frames. An der Stelle, wo die Tags stehen, erscheint - je nach Parametereinstellung - der Mund in der Größe, die durch „WIDTH“ und „HEIGHT“ bestimmt wurden. Durch das Attribut „ID“ im Tag <OBJECT> wird ein Objekt definiert, dessen Bezeichnung für die Ansteuerung durch Skriptsprachen wie z.B. JavaScript notwendig ist. Mit dem Attribut „CLASSID“ wird das ActiveX des Internet Explorer-Plugins angesprochen.

I Lizenzschlüssel

Jetzt fehlt noch der Lizenzschlüssel. Bevor WebSpeech aktiv wird, wird zuerst geprüft, ob die aufrufende Seite überhaupt berechtigt ist, WebSpeech zu benutzen. Das erfolgt mittels eines Lizenzschlüssels, den sie mit diesem Handbuch zusammen erhalten haben. Falls nicht wenden Sie sich bitte an

G DATA Software AG
Business-Vertrieb (B2B)
 Königsallee 178 b
 44799 Bochum
 Telefon: 0234 / 97 62 - 170
 E-Mail: b-vertrieb@gdata.de

Der Schlüssel funktioniert nur auf der lizenzierten Domäne (z.B. www.webspeech.de). Es ist nicht möglich den Schlüssel einer Webseite auf einer anderen zu benutzen.

Der Lizenzschlüssel wird über den Parameter AUTHKEY übergeben.

```
<OBJECT ID="WebSpeech"
  CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
  WIDTH="150" HEIGHT="125"
>
<PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
</OBJECT>
```


2 Ein Text

Dem oben angegebenen Code fehlt noch ein wichtiger Parameter - der zu sprechende Text. Zur Bestimmung eines Textes stehen 2 Möglichkeiten zur Verfügung:

- 1.) Kurze Texte können mit dem Parameter „TEXT“ übergeben werden.

```
<OBJECT ID="WebSpeech"
  CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
  WIDTH="150" HEIGHT="125"
>
<PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
<PARAM NAME="TEXT" VALUE="Herzlich willkommen bei WebSpeech.">
</OBJECT>
```

- 2.) Falls der gesprochene Text etwas länger ist oder Sonderzeichen („<“) und Zeilenumbrüche enthält, bietet es sich an, ihn in eine Datei zu schreiben. Im VALUE des Parameters „URL“ wird deren Adresse angegeben.

```
<OBJECT ID="WebSpeech"
  CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
  WIDTH="150" HEIGHT="125"
>
<PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
<PARAM NAME="URL" VALUE="http://www.webspeech.de/texte/text1.txt">
</OBJECT>
```

Beide Codes erzeugen eine Seite, in der der Mund in der angegebenen Größe angezeigt wird. Wenn der Besucher der Seite die Maus über den Mund bewegt, erscheint ein Fenster und durch einen Klick auf den Startbutton wird der angegebene Text gesprochen. Es gibt aber noch andere Möglichkeiten, die Sprachausgabe zu starten.

3 Automatisches Starten der Sprachausgabe

Die Sprachausgabe kann nicht nur durch den Benutzer ausgelöst werden, sondern auch automatisch gestartet werden. Dazu muss der Parameter „AUTOSTART“ auf „1“ gesetzt werden.

```
<OBJECT ID="WebSpeech"
  CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
  WIDTH="150" HEIGHT="125"
>
<PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
<PARAM NAME="TEXT" VALUE="Herzlich willkommen bei WebSpeech.">
<PARAM NAME="AUTOSTART" VALUE="1">
</OBJECT>
```

4 Animation des Mundes

Der Mund bewegt sich synchron zu den gesprochenen Lauten. Sie können die Animation des Mundes unterbinden, indem Sie den Parameter „MOUTHANIMATION“ auf „0“ setzen.

```
<OBJECT ID="WebSpeech"
  CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
  WIDTH="150" HEIGHT="125"
>
<PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
<PARAM NAME="TEXT" VALUE="Herzlich willkommen bei WebSpeech.">
<PARAM NAME="MOUTHANIMATION" VALUE="0">
</OBJECT>
```

5 Zusammenfassung

Nun sollte das Grundprinzip der HTML-Ansteuerung von WebSpeech klar geworden sein. Mit dem `<OBJECT>` -Tag wird das Plugin für den Internet Explorer angesprochen. Zusätzlich zur Objektdefinition können weitere Eigenschaften angegeben werden. Die OBJECT ID, die CLASSID und der Lizenzschlüssel in AUTHKEY müssen auf jeden Fall angegeben werden. Im folgenden Kapitel werden die einzelnen Parameter ausführlich beschrieben.

6 Parameter

Zunächst zur Übersicht ein Listing, das alle Parameter enthält, die zur Steuerung der Eigenschaften von WebSpeech zur Verfügung stehen. Die einzelnen Parameter sind untergliedert in die Bereiche Sprachparameter, Parameter zur grafischen Darstellung des Mundes und des Textes und weitere Parameter.

Beispiel

```
<OBJECT
  ID="WebSpeech"
  CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
  WIDTH="150"
  HEIGHT="125"
>
  <PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
  <PARAM NAME="AUTOSTART" VALUE="0">
  <PARAM NAME="IMMEDIATE" VALUE="0">
  <PARAM NAME="TEXT" VALUE="Ein netter Text">
  <PARAM NAME="URL" VALUE=" http://www.WebSpeech.de/mum/texte/mum03.lgx ">
  <PARAM NAME="AUTHKEY" VALUE="AUTHKEY">
  <PARAM NAME="SPECIAL" VALUE="SPECIAL">
  <PARAM NAME="MOUTHCOLOR" VALUE="black,TEXT,TEXT,TEXT,TEXT">
  <PARAM NAME="MOUTHANIMATION" VALUE="1">
  <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">
</OBJECT>
```

Folgend werden nun die einzelnen Parameter beschrieben. Die korrekte Syntax wird genannt und die Wertebereiche und Defaultwerte werden angegeben. Die Auswirkungen der Parameter können auf der HTML-Seite „testbox.htm“ im Verzeichnis „WebSpeech-CD://Dokumentation/Tutorial“ der beiliegenden CD ausprobiert werden.

6.1 AUTHKEY

```
<PARAM NAME="AUTHKEY" VALUE="value">
```

Standard

kein Standardwert

Beschreibung

Bevor WebSpeech aktiv wird, wird zuerst geprüft, ob die Seite berechtigt ist, WebSpeech anzusprechen. Dies erfolgt über einen Lizenzschlüssel, der für jede Domäne unterschiedlich aufgebaut ist. Die Seiten auf www.logox.de sprechen daher nicht mit dem Schlüssel von www.webspeech.de. Einen gültigen

Lizenzschlüssel erhalten Sie bei

G DATA Software AG

Business-Vertrieb (B2B)

Königsallee 178 b

44799 Bochum

Telefon: 0234 / 97 62 - 170

E-Mail: b-vertrieb@gdata.de

Der Lizenzschlüssel wird über den Parameter AUTHKEY übergeben.

Mögliche Werte

ein gültiger Lizenzschlüssel

Beispiel

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

Siehe auch

`SetKey()`

6.2 AUTOSTART

```
| <PARAM NAME="AUTOSTART" VALUE="value">
```

Standardwert

„0“ ausgeschaltet

Beschreibung

WebSpeech kann den Vorleseprozess durch eine Benutzeraktion starten oder automatisch beim Laden der Seite erfolgen lassen. Der Benutzer kann WebSpeech nur über das Menü des Mundes, den Wechsel auf eine andere Internetseite oder durch Klicken eines weiteren WebSpeech-Objektes stoppen.

„AUTOSTART“ setzt - wie alle anderen Parameter auch - voraus, dass ein zu sprechender Text entweder mit dem Parameter „TEXT“ oder mit dem Parameter „URL“ bestimmt wurde.

Mögliche Werte

„0“ manueller Start durch Benutzeraktion
„1“ automatischer Start

Beispiel

```
| <PARAM NAME="AUTOSTART" VALUE="1">
```

Siehe auch

TEXT, URL, IMMEDIATE, LoadText()

6.3 IMMEDIATE

```
| <PARAM NAME="IMMEDIATE" VALUE="value">
```

Standardwert

„0“ ausgeschaltet

Beschreibung

Es ist erforderlich, dass die Eigenschaft „AUTOSTART“ = „I“ gesetzt ist. Dann kann die Sprachausgabe des von den Eigenschaften „TEXT“ oder „URL“ bestimmten Textes unmittelbar beginnen, ohne das Ende eines zuvor gesprochenen Textes abzuwarten („IMMEDIATE“ = „I“).

Bei „IMMEDIATE“ = „0“ wird der neue Text erst gestartet, wenn der gerade aktive Text vorgelesen wurde. Wurde die Ausgabe eines Textes mit Pause unterbrochen, kann der neue Text nur mit „IMMEDIATE“ = „I“ gestartet werden.

Mögliche Werte

„0“ ein zuvor gestarteter Sprechvorgang wird vor der neuen Ausgabe vollständig ausgeführt
 „I“ ein zuvor gestarteter Sprechvorgang wird abgebrochen und die neue Ausgabe wird sofort gestartet

Beispiel

```
| <PARAM NAME="IMMEDIATE" VALUE="0">
```

Siehe auch

TEXT, URL, AUTOSTART, LoadText(), StartSpeakingImmediate()

6.4 TEXT

```
| <PARAM NAME="TEXT" VALUE="value">
```

Standard

kein Standardwert

Beschreibung

Mit dieser Eigenschaft kann der Text definiert werden, der vorgelesen werden soll, ohne eine Skriptsprache zu verwenden. Wenn kein Text bestimmt wird, erfolgt keine Sprachausgabe. Die Ausgabe erfolgt sofort, wenn „AUTOSTART“ auf „I“ gesetzt wurde. Andernfalls wird der Text erst nach einer Benutzeraktion (z.B. ein Klick auf das StartIcon im Menü des WebSpeech Mundes) vorgelesen.

Neben den Wörtern kann der Text auch Steuerzeichen zur Anpassung der Aussprache, zum Einfügen von Pausen, Sprecherwechseln und Soundeffekten und zur Variation der Lautstärke und der Sprechweise enthalten.

Mögliche Werte

Beliebig lange ASCII-Texte, auch mit WebSpeech-Steuertags (siehe Abschnitt 4 „Speechtags“), z.B.: „Herzlich willkommen auf unserer Homepage.“

Achtung: Manche Sonderzeichen verursachen Syntaxfehler im HTML-Code.

Beispiel

```
| <PARAM NAME="TEXT" VALUE="Hier spricht WebSpeech.">
```

Siehe auch

AUTOSTART, IMMEDIATE, URL, SetText() und Speechtags

6.5 URL

```
| <PARAM NAME="URL" VALUE="value">
```

Standard

kein Standardwert

Beschreibung

Diese Eigenschaft kann benutzt werden, um ohne die Verwendung eines Skripts eine Textdatei vorlesen zu lassen. Der Wert gibt eine HTTP-Adresse an, von wo der nächste Text geladen werden soll. Die Ausgabe erfolgt sofort, wenn „AUTOSTART“ auf „I“ gesetzt wurde. Wurde „IMMEDIATE“ auch auf „I“ gesetzt, wird die vorherige Ausgabe abgebrochen.

Der Verweis kann auf jeden Datentyp zeigen. Der Inhalt wird bis zum ersten Binärzeichen vorgelesen. Die erste Zeile in den „.ltx“-Dateien von WebSpeech werden korrekt interpretiert. Andere Datenformate werden als reine Textdateien behandelt und nicht weiter interpretiert.

Werden „TEXT“ und „URL“ gleichzeitig benutzt, wird zunächst der Inhalt von „TEXT“ gesprochen und dann der Inhalt der in „URL“ angegebenen Datei. Es ist auch möglich, die zu sprechenden Texte per Skriptsprachen zu definieren. Wird kein Text bestimmt, kann selbstverständlich nichts vorgelesen werden.

Mögliche Werte

Die Adresse der zu sprechenden Adresse im Internet, z.B.:

„http://www.WebSpeech.de/texte/willkommen.txt“

Beispiel

```
| <PARAM NAME="URL" VALUE="http://www.webspeech.de/texte/text1.txt">
```

Siehe auch

AUTOSTART, IMMEDIATE, TEXT, LoadText() und Speichtags

6.6 MOUTHANIMATION

```
| <PARAM NAME="MOUTHANIMATION" VALUE="value">
```

Standard

„1“

Beschreibung

schaltet die Lippenbewegungen des Mundes an bzw. aus

Mögliche Werte

„1“ der Mund ist animiert

„0“ der Mund ist nicht animiert

Beispiel

```
| <PARAM NAME="MOUTHANIMATION" VALUE="0">
```

Siehe auch

SetMouthAnimation()

6.7 MOUTHCOLOR

```
<PARAM NAME="MOUTHCOLOR" VALUE="value">
```

Standard

```
„black,TEXT,TEXT,TEXT,TEXT“
```

Beschreibung

Dieses Attribut definiert die Farben des Mundes. Die Farbangaben werden mit Kommata getrennt und bestimmen die Farben der folgenden Bereiche der Mundgrafik in der angegebenen Reihenfolge:

| | |
|-----------|-----------|
| Umrandung | (Outline) |
| Rachen | (Throat) |
| Zunge | (Tongue) |
| Zähne | (Teeth) |
| Lippen | (Lips) |

Mögliche Werte

Neben den HTML-Farben (siehe Tabelle der Farbnamen im Anhang) und den direkten Farbangaben („#RRGGBB“ hexadezimale Angabe der Rot-, Grün- und Blauanteile) sind auch folgende Angaben zulässig:

- „DEFAULT“: Die vordefinierten Farbwerte werden verwendet. Dabei werden „TRANSPARENT“ und „TEXTURE“ nicht berücksichtigt. „TRANSPARENT“ oder „TRANSP“: bedeutet Transparenz, der Hintergrund wird nicht überschrieben und er scheint durch. Man kann nur den Mund sehen (sofern er nicht auch transparent ist und der Browser transparente Farben unterstützt).
- „TEXTURE“ oder „TEXT“: Beim Anwählen der Textur wird eine Grafik in die aktuelle Form des Mundes gemorpht. Die Umrandung ist in der Textur enthalten und daher wird der Umrandungs-Farbwert bei „TEXT“ ignoriert.

Beispiel

```
<PARAM NAME="MOUTHCOLOR" VALUE="DEFAULT,TRANSP, #A1B2C3,TEXT, forestgreen">
```

Setzt die Farbe der Umrandung auf den vordefinierten Wert „black“, der Rachen ist transparent, die Zunge hat die RGB-Farbe „161, 178, 195“ (hexadezimal: „A1B2C3“), die Zähne werden mit der Textur gemorpht und die Lippen sind waldgrün (Laut Farbwertdefinition „RGB = 34, 139, 34“ (hexadezimal: „228B22“)).

Siehe auch

BACKGROUNDCOLOR, SetMouthColor(), Farbtabelle in Anhang I

6.8 TEXTANIMATION

```
| <PARAM NAME="TEXTANIMATION" VALUE="value">
```

Standard

„3“

Beschreibung

Das WebSpeech-Fenster mit dem Mund kann den gesprochenen Text anzeigen. Dieser Parameter (de-) aktiviert die Textanzeige und steuert deren Ausgabemodus.

Mögliche Werte

| | |
|-----|-----------------------------------|
| „0“ | aus |
| „1“ | Laufschrift von rechts nach links |
| „2“ | Einzelwortdarstellung |
| „3“ | Anzeige gemäß Benutzerauswahl |

Beispiel

```
| <PARAM NAME="TEXTANIMATION" VALUE="3">
```

Siehe auch

SetTextAnimation()

6.9 TEXTCOLOR

```
| <PARAM NAME="TEXTCOLOR" VALUE="value">
```

Standard

„black,gray“

Beschreibung

Hiermit wird die Textfarbe eingestellt, die für die Laufschrift der gerade gesprochenen Wörter verwendet wird, sofern die Textanzeige mit „TEXTANIMATION“ aktiviert wurde. Jeder Text wird mit einem Schatten angezeigt (der natürlich transparent gesetzt werden kann). Dadurch kann eine bessere Lesbarkeit erzielt werden. Das aktuell gesprochene Wort wird hierbei immer fett angezeigt.

Die Farben werden mit Kommata getrennt. Der erste Wert bezeichnet die Textfarbe, der zweite die Schattenfarbe.

Mögliche Werte

Siehe MOUTHCOLOR. Texturen sind für die Texte nicht vorgesehen.

Beispiel

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

Setzt die Farbe der Buchstaben auf den RGB-Farbwert „A1B2C3“ und den Schatten auf waldgrün

Siehe auch

TEXTANIMATION, MOUTHCOLOR, SetTextColor(), Farbtabelle in Anhang I

6.10 TEXTPOSITION

```
| <PARAM NAME="TEXTPOSITION" VALUE="value">
```

Standard

„0“

Beschreibung

Positioniert die Textausgabe in vertikaler Richtung. Dies funktioniert nur, wenn „TEXTANIMATION“ eingeschaltet ist.

Mögliche Werte

„0“ - „100“ als vertikale Ausrichtung des Feldes.
„0“ ganz oben,
„100“ ganz unten

Beispiel

```
| <PARAM NAME="TEXTPOSITION" VALUE="75">
```

Siehe auch

TEXTANIMATION, SetTextPosition()

6.11 TEXTSIZE

```
| <PARAM NAME="TEXTSIZE" VALUE="value">
```

Standard

„12“

Beschreibung

Bestimmt die Buchstabenhöhe der Textausgabe in Pixel, sofern die gesprochenen Texte angezeigt werden.

Mögliche Werte

Der Wert wird in Pixeln angegeben.

Beispiel

```
| <PARAM NAME="TEXTSIZE" VALUE="12">
```

Siehe auch

TEXTANIMATION, SetTextSize()

6.12 OPAQUE

```
| <PARAM NAME="OPAQUE" VALUE="value">
```

Beschreibung

Mit diesem Parameter kann eine deckende Ansicht des Objektes erzwungen werden. Dann wird das gesamte Fenster, in dem WebSpeech aktiv ist, sichtbar. Die Hintergrundfarbe wird durch „BACKGROUNDCOLOR“ bzw. „SetBackgroundColor“ bestimmt. Wenn eine transparente Darstellung des Hintergrunds erwünscht ist, sollte der Parameter Opaque angegeben und dessen Wert auf „0“ gesetzt werden.

Mögliche Werte

„0“ transparent
„1“ opak (undurchsichtig)

Beispiel

```
| <PARAM NAME="OPAQUE" VALUE="1">
```

Siehe auch

BACKGROUNDCOLOR, SetOpaque(), SetBackgroundColor()

6.13 BACKGROUND COLOR

```
| <PARAM NAME="BACKGROUND COLOR" VALUE="value">
```

Standard

„white“

Beschreibung

Das Objekt wird standardmäßig transparent gezeichnet. Der HTML-Hintergrund scheint also durch. Wenn eine transparente Darstellung nicht möglich oder nicht gewollt ist (siehe „OPAQUE“), wird der Hintergrund mit dieser Farbe gefüllt.

Mögliche Werte

Siehe „MOUTHCOLOR“. Allerdings sind „TRANSPARENT“ und „TEXTURE“ nicht definiert.

Beispiel

```
| <PARAM NAME="BACKGROUND COLOR" VALUE="#ffccff">
```

Siehe auch

OPAQUE, MOUTHCOLOR, SetBackgroundColor(), Farbtabelle in Anhang I

6.14 CONTROLPOSITION

```
| <PARAM NAME="CONTROLPOSITION" VALUE="value">
```

Standard

„75“

Beschreibung

Das Kontrollfeld, das angezeigt wird, sobald der Benutzer mit der Maus über das WebSpeech Fenster fährt, wird in vertikaler Richtung positioniert.

Mögliche Werte

„0“ - „100“ als vertikale Ausrichtung des Feldes.
„0“ ganz oben,
„100“ ganz unten

Beispiel

```
| <PARAM NAME="CONTROLPOSITION" VALUE="50">
```

Siehe auch

SetControlPosition()

Abschnitt 4

I Einbindung in Skriptsprachen

Hier sollen einige Grundprinzipien und Konventionen zur Einbindung von WebSpeech in Skriptsprachen erläutert werden.

JavaScript ist als Skriptsprache im Internet weit verbreitet. Es wird von den üblichsten Browsern interpretiert, wenn auch in unterschiedlichem Umfang. WebSpeech unterstützt zwar auch andere Skriptsprachen wie JScript und VBScript - diese werden aber nicht von allen Browsern verstanden. Da wir davon ausgehen, dass Webseiten in browserunabhängigem Code verfasst sind, werden alle Programmierbeispiele in JavaScript angegeben.

Bei JavaScript handelt es sich um eine objektorientierte Programmiersprache, deren Objekte im Bereich von HTML-Seiten liegen. Die Objekte haben Eigenschaften, Methoden und können mit bestimmten Ereignissen verknüpft werden. Die entsprechende Programmiersyntax entnehmen Sie bitte der Fachliteratur. Alle Objekte sind in einer Hierarchie angeordnet. WebSpeech steht in der Hierarchie unter „window.document.WebSpeech-Objektnamen.WebSpeech-Methoden/Eigenschaften.“

Für alle weiterhin angegebenen Beispiele gelten folgende Konventionen:

Der Objektnamen, der im Tag „OBJECT“ für den Internet Explorer durch das Attribut „ID“ bestimmt wird, ist in allen folgenden Beispielen „WebSpeech“.

Wenn Daten aus einem Formular gelesen werden, dann hat das Formular den Objektnamen „Formular“.

Wenn also WebSpeech Operationen durchgeführt werden, dann beginnen sie mit

```
| window.document.WebSpeech.
```

Diesen Ausdruck kann man in einer Funktion durch eine globale Variable ersetzen. Diese Funktion sollte im Tag <BODY> bei „OnLoad“ aufgerufen werden. Danach ersetzt das angegebene Kürzel - z.B. „MyWebSpeech“ - den umständlichen Ausdruck „window.document.WebSpeech“. Das folgende Beispiel zeigt dies anhand eines einfachen Skripts, in dem ein Text aus dem Textfeld eines Formulars vorgelesen wird, wenn auf „Sprechen“ geklickt wird.

```
<html><head>
<script language=JavaScript>
<!--
function WebSpeechInitialisieren()
{
    MyWebSpeech=window.document.WebSpeech;
}

function SagEsWS(Text)
{
    MyWebSpeech.SetText(Text);
    MyWebSpeech.StartSpeaking();
}
-->
</script></head>
<body onload="WebSpeechInitialisieren()">
<TABLE> <TR><TD ALIGN="CENTER">
<OBJECT
    ID="WebSpeech" CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
    WIDTH=120 HEIGHT=100>
    <PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
</OBJECT> <br>
<A href="javascript:SagEsWS(document.Formular.Textfeld.value);">Sprechen</A>
<FORM NAME="Formular">
</TD><TD><TEXTAREA name=Textfeld cols=40 rows=8 wrap=virtual></TEXTAREA></TD><TR>
</TABLE></body></html>
```

2 Methoden

Die Auflistung der Methoden erfolgt analog den Eigenschaften für HTML-Code zunächst für die Sprechparameter. Hinzu kommen Methoden über Statusmeldungen des Systems. Danach werden die grafischen Gestaltungsmöglichkeiten für die Darstellung des Mundes und des Textes angegeben, bevor weitere Methoden wie Benutzervariablen und Version beschrieben werden.

Die Benennung der Methoden erfolgt nach folgendem Muster:

- Typ des Rückgabewertes
- Name der Methode
- obligatorische Übergabeparameter in Klammern

Für jeden Übergabeparameter wird zuerst der Datentyp benannt und danach ein möglicher Name für den Parameter. Übergabeparameter sind durch „[in]“ gekennzeichnet und Rückgabewerte durch „[out]“.

Beispiel

```
| int LoadText(String strURL,Bool bAutoStart,Bool bImmediate);
```

2.1 SetText

```
| int SetText( String strText );
```

Rückgabewert

< „0“ Fehler

>= „0“ ok

Parameter

strText

[in] Der zu sprechende Text. Der Text darf beliebig lang sein und kann Speechtags beinhalten. Er kann direkt als String angegeben werden oder als Variable, die eine Zeichenkette enthält, oder als eine Kombination von allem gemäß der Syntax der verwendeten Skriptsprache.

Umfangreichere Texte sollten in eine (oder mehrere) Datei(en) geschrieben werden und mit der Methode „LoadText“ aufgerufen werden.

Bemerkungen

Setzt einen Text für die nächste Textausgabe. Die Funktion „SetText“ kann auch benutzt werden, wenn gerade ein anderer Text ausgegeben wird. Sie hat keinen unmittelbaren Einfluss auf den gerade gesprochenen Text. Erst nach einem Aufruf der Funktion „StartSpeaking“ wird der mit „SetText“ definierte Text gesprochen.

Beispiele

```
| window.document.WebSpeech.SetText(document.Formular.Textfeld.value);
```

oder

```
| window.document.WebSpeech.SetText("Willkommen bei WebSpeech");
```

Siehe auch

StartSpeaking(), StartSpeakingImmediate(), LoadText(), IsSpeaking(), TEXT

2.2 LoadText

```
int LoadText(String strURL, Bool bAutoStart, Bool bImmediate);
```

Rückgabewert

< „0“ Fehler

>= „0“ ok.

Parameter

strURL

[in] Gibt die Internetadresse an, von der ein Text geladen werden soll. Es kann ein vollständiger URL angegeben werden („http://www.xyz.com/sprich.txt“) oder eine relative Pfadangabe („texte/lies.txt“).

bAutoStart

[in] Gibt das Verhalten nach erfolgreicher Übertragung des Textes an. Ist „bAutoStart“ = „1“ wird der Text unmittelbar vorgelesen, ist „bAutoStart“ = „0“, kann der Text durch „StartSpeaking()“ oder „StartSpeakingImmediate“ vorgelesen werden.

blmmediate

[in] Gibt das Verhalten bei „bAutoStart“ = „1“ an. Ist „blmmediate“ = „1“ wird der Text vorgelesen und eine eventuell gerade aktive Textausgabe zuvor gestoppt. Ist „blmmediate“ = „0“, wird der neu geladene Text erst dann vorgelesen, nachdem die aktuelle Textausgabe endet.

Beschreibung

Von der übergebenen HTML-Adresse wird ein Text für die nächste Sprachausgabe geladen. Es kann immer nur ein Text geladen werden. Ist „bAutostart“ gesetzt (=„1“), wird der Text nach dem Laden sofort („blmmediate“=„1“) oder nach einer aktuellen Sprachausgabe („blmmediate“=„0“) gesprochen.

Der Datentyp von „AutoStart“ und „Immediate“ muss ggf. zuerst konvertiert werden (z.B. mit der JavaScript-Funktion „Number()“.

Die Initialisierung sehr großer Texte (> 100KB Sprechzeit ca. 90 Min.) stellt hohe Anforderungen an die Rechenleistung und kann einige Sekunden dauern.

Beispiel

```
<script language=JavaScript>
function LoadIt(URL,AutoStart,Immediate) {
    window.document.WebSpeech.LoadText(URL,AutoStart,Immediate);
}
</script>
...
<body onload="LoadIt('text1.txt',1,0)">
```

In dieser JavaScript-Funktion werden der Funktion „LoadIt“ die Parameter „URL“, „AutoStart“ und „Immediate“ übergeben. Im Body der Funktion wird damit ein Text geladen.

Siehe auch

SetText(), LoadText(), TEXT, URL, AUTOSTART, IMMEDIATE

2.3 StartSpeaking

```
| int StartSpeaking();
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Beschreibung

Ein zuvor gesetzter oder geladener Text wird gesprochen. Die Ausgabe erfolgt erst, nachdem eine aktuelle Sprachausgabe beendet wurde. Bei einer durch Pause unterbrochenen Ausgabe wird der Befehl erst ausgeführt, wenn der unterbrochene Text beendet wurde. Ein mit Pause unterbrochener Text kann mit „ResumeSpeaking()“ fortgesetzt oder mit „StartSpeakingImmediate()“ abgebrochen werden.

Beispiel

```
| <script language=JavaScript>  
|     window.document.WebSpeech.StartSpeaking();  
| </script>
```

Siehe auch:

SetText(), LoadText(), TEXT, URL, AUTOSTART

2.4 StartSpeakingImmediate

```
| int StartSpeakingImmediate();
```

Rückgabewert

< „0“ Fehler

>= „0“ ok

Beschreibung

Ein zuvor gesetzter oder geladener Text „(SetText()“, „LoadText(...)“, „TEXT=,,,,,,“ „URL=,,,,,,“) wird gesprochen. Eine aktuelle Sprachausgabe wird dabei abgebrochen.

Beispiel

```
| <script language=JavaScript>  
|     window.document.WebSpeech.StartSpeakingImmediate();  
| </script>
```

Siehe auch

SetText(), LoadText(), TEXT, URL, AUTOSTART, IMMEDIATE

2.5 StopSpeaking

```
| int StopSpeaking();
```

Rückgabewert

< „0“ Fehler

>= „0“ ok

Beschreibung

Die aktuelle Sprachausgabe wird abgebrochen.

Beispiel

```
| <script language=JavaScript>  
|     window.document.WebSpeech.StopSpeaking();  
| </script>
```

Siehe auch

StartSpeaking(), StartSpeakingImmediate(), LoadText(), PauseSpeaking()

2.6 PauseSpeaking

```
| int PauseSpeaking();
```

Rückgabewert

< „0“ Fehler

>= „0“ ok

Beschreibung

Eine aktuelle Sprachausgabe wird angehalten und kann mit „ResumeSpeaking()“ wieder aufgenommen werden. Der Befehl „StartSpeaking“ ist wirkungslos, solange die Pause nicht aufgehoben wurde. Ob die Sprachausgabe mit Pause unterbrochen wurde, kann man mit der Methode „IsPause()“ abfragen.

Beispiel

```
| <script language=JavaScript>  
|     window.document.WebSpeech.PauseSpeaking();  
| </script>
```

Siehe auch

ResumeSpeaking(), StartSpeaking(), StartSpeakingImmediate(), LoadText(), IsPause()

2.7 ResumeSpeaking

```
| int ResumeSpeaking();
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Beschreibung

Eine mit „Pause“ unterbrochene Sprachausgabe wird dort fortgesetzt, wo sie unterbrochen wurde.

Beispiel 1

```
| <script language=JavaScript>
|     window.document.WebSpeech.ResumeSpeaking();
| </script>
```

Beispiel 2

Basisfunktionen für die Steuerung von WebSpeech:

```
| <script language=JavaScript>
| function StartIt() {
| // im Body Tag mit Onload="StartIt()" aufrufen
|     MyWebSpeech=window.document.WebSpeech;
| }
|
| function LoadIt(URL,AutoStart,Immediate) {
|     MyWebSpeech.LoadText(URL,AutoStart,Immediate);
| }
|
| function SpeakIt(Text) {
|     MyWebSpeech.SetText(Text);
|     MyWebSpeech.StartSpeaking();
| }
|
| function SpeakItImm(Text) {
|     MyWebSpeech.SetText(Text);
|     MyWebSpeech.StartSpeakingImmediate();
| }
|
| function StopIt() {
|     MyWebSpeech.StopSpeaking(); }
|
| function PauseIt() {
|     MyWebSpeech.PauseSpeaking(); }
|
| function ResumeIt() {
|     MyWebSpeech.ResumeSpeaking(); }
|
| </script>
```

Siehe auch

PauseSpeaking(), StartSpeaking(), StartSpeakingImmediate()

2.8 IsSpeaking

```
| int IsSpeaking();
```

Rückgabewert

Hat entweder den Wert „1“, wenn gerade gesprochen wird, oder den Wert „0“.

Beschreibung

Fragt ab, ob gerade gesprochen wird.

Beispiel

```
| function CheckSpeaking()  
| {  
|   var SprechenAktiv=document.WebSpeech.IsSpeaking();  
|   if (SprechenAktiv==1) {  
|     document.Formular.Textfeld.value = "WebSpeech spricht.";  
|   }  
|   else {  
|     document.Formular.Textfeld.value = "WebSpeech ist stumm.";  
|   }  
| } // end of function CheckSpeaking()
```

Hier wird eine JavaScript-Funktion definiert, die abfragt, ob gesprochen wird. Je nach Ergebnis der Abfrage wird ein entsprechender Text in das Textfeld des Formulars „Formular“ ausgegeben.

Siehe auch

PauseSpeaking(), StartSpeaking(), StartSpeakingImmediate(), LoadText()

2.9 IsPause

```
| int IsPause();
```

Rückgabewert

Hat entweder den Wert „1“, wenn eine Ausgabe angehalten wurde, oder den Wert „0“.

Beschreibung

Fragt ab, ob eine Sprachausgabe angehalten wurde.

Beispiel

```
<script language=JavaScript>
function CheckPause()
{
    var Pause=document.WebSpeech.IsPause();
    window.document.WebSpeech.LoadText("text.txt",1,Pause);}
} // end of function CheckPause
</script>
```

Es wird eine JavaScript-Funktion definiert, die abfragt, ob die Sprachausgabe angehalten wurde. Wenn ja, wird mit „Pause“ der Wert für „Immediate“ auf „1“ gesetzt. Die Pause wird abgebrochen und WebSpeech beginnt sofort mit der Ausgabe von „text.txt“. Wenn nicht pausiert wird, wird - da „Pause“ und damit der Wert für „Immediate“ = „0“ ist - ein aktuell erklingender Text beendet und erst dann wird der „text.txt“ ausgegeben.

Siehe auch

PauseSpeaking(), ResumeSpeaking(), StartSpeaking(), StartSpeakingImmediate()

2.10 IsMuting

```
| int IsMuting();
```

Rückgabewert

„1“ wenn eine Ausgabe temporär oder permanent stummgeschaltet wurde
„0“ wenn die Sprachausgabe möglich ist

Beschreibung

Fragt ab, ob die Sprachausgabe stummgeschaltet wurde. Es ist nicht möglich, die Stummschaltung durch ein Skript zu umgehen. Der Endbenutzer muss die Stummschaltung explizit aufheben (durch einen Klick auf den Play-Button oder den Mute-Button des WebSpeech-Kontrollfelds).

Beispiel

```
<script language=JavaScript>
function CheckMuting()
{
    var MutingAktiv=document.WebSpeech.IsMuting();
    if (MutingAktiv==1)
    {
        window.location.href="Seite-ohne-Sprachunterstützung.htm";
    }
    else {
        window.location.href="Seite-mit-Sprachunterstützung.htm";
    }
} // end of function CheckMuting()
</script>
```

Es wird eine JavaScript-Funktion definiert, die, je nachdem ob WebSpeech stummgeschaltet wurde, verschiedene HTML-Seiten aufruft.

2.11 IsLoading

```
| int IsLoading();
```

Rückgabewert

Hat entweder den Wert „1“, wenn gerade geladen wird, oder den Wert „0“.

Beschreibung

Fragt ab, ob gerade ein Text von einer HTTP-Adresse geladen wird.

Beispiel

```
<script language=JavaScript>
function CheckLoading()
{
    var LoadingAktiv=document.WebSpeech.IsLoading();
    if (LoadingAktiv==1)
    {
        window.setTimeout( 'CheckLoading()',1000);
    }
    else {
        window.document.WebSpeech.StartSpeaking();
    }
} // end of function CheckLoading()
</script>
```

Wenn ein Text von einer URL geladen wird, der nicht sofort gesprochen werden soll („bAutostart=0“), dann kann mit dieser JavaScript-Funktion geprüft werden, ob der Ladevorgang abgeschlossen ist. Wenn der Ladevorgang noch läuft, wird eine Sekunde später erneut geprüft, ob der Ladevorgang abgeschlossen ist. Wenn er abgeschlossen ist, wird der Text vorgelesen.

Siehe auch

LoadText(), URL

2.12 SetMouthAnimation

```
| int SetMouthAnimation(int nMouthAnimation)
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

iMouthAnimation

[in] Animation einschalten = 1, Animation ausschalten = 0

Beschreibung

Hiermit wird die Animation des WebSpeech-Mundes ein- bzw. ausgeschaltet. Ist die Animation ausgeschaltet, wird auch kein Mund gezeichnet. Standardmäßig ist die Animation eingeschaltet.

Beispiel

```
| <script language=JavaScript>  
|     window.document.WebSpeech.SetMouthAnimation(3);  
| </script>
```

Siehe auch

MOUTHANIMATION

2.13 SetMouthColor

```
| int SetMouthColor(String strMouthColor)
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

strMouthColor

[in] Definiert die Farben des Logox-Mundes. Die Farben werden als Zeichenkette übergeben. Insgesamt sind 5 Farben zu setzen. Die Farbangaben werden mit Kommata getrennt und haben folgende Reihenfolge:

| | |
|-----------|-----------|
| Umrandung | (Outline) |
| Rachen | (Throat) |
| Zunge | (Tongue) |
| Zähne | (Teeth) |
| Lippen | (Lips) |

Neben den HTML-Farben (siehe Farbtabelle im Anhang) und den direkten Farbangaben in der Form „#RRGGBB“ mit hexadezimalen Angaben der Rot-, Grün- und Blauanteile, sind auch folgende Angaben zulässig:

DEFAULT

Die von uns vordefinierten Farbwerte werden übernommen. Dabei werden keine Texturen und transparente Farben berücksichtigt. Default unterscheidet sich daher vom Standardwert.

TRANSPARENT oder TRANSP

Bedeutet Transparenz, der Hintergrund wird nicht überschrieben und scheint durch.

TEXTURE oder TEXT

Bedeutet Textur, eine Textur ist eine Grafik die in die aktuelle Form gemorpht wird. Wenn „TEXTURE“ gesetzt ist, wird der Parameter für die Umrandung („Outline“) wirkungslos.

Beschreibung

Hiermit wird die Darstellung des WebSpeech Mundes verändert. Als Standardwert wird „black,text,text,text,text“ gewählt.

Beispiel

```
<script language=JavaScript>
function BunterMund()
{
    Umrandung = window.document.Formular.Outline.value;
    Rachen = window.document.Formular.Throat.value;
    Zunge = window.document.Formular.Tongue.value;
    Zaehne = window.document.Formular.Teeth.value;
    Lippen = window.document.Formular.Lips.value;
    window.document.WebSpeech.SetMouthColor(Umrandung+", "+Rachen+", "+Zunge+", "+Zaehne+",
"+Lippen);
} // end of function BunterMund()
</script>
```

In dieser JavaScript-Funktion werden aus den Zellen „Outline“, „Throat“, „Tongue“, „Teeth“ und „Lips“ des Formulars „Formular“ die Werte für die Farben des Mundes gelesen und danach mit „SetMouthColor“ gesetzt.

Siehe auch

MOUTHCOLOR, Farbtabelle in Anhang I

2.14 SetTextAnimation

```
| int SetTextAnimation(int nTextAnimation)
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

iTextAnimation

[in] bestimmt den Darstellungsmodus der Laufschrift:

| | |
|-----|-----------------------------------|
| „0“ | aus |
| „1“ | Laufschrift von rechts nach links |
| „2“ | Einzelworte |
| „3“ | Anzeige gemäß Benutzerauswahl |

Beschreibung

Hiermit wird die Form der Darstellung der gesprochenen Wörter im Lippenfenster bestimmt. Der Standardwert ist „3“, also „gemäß Benutzerwunsch“.

Beispiel

```
| <script language=JavaScript>
|     window.document.WebSpeech.SetTextAnimation(0);
| </script>
```

Siehe auch

TEXTANIMATION

2.15 SetTextColor

```
| int SetTextColor(String strTextColor);
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

strTextColor

[in] Definiert die Farbe des Textes und seines „Schattens“. Die zwei Farben werden mit Kommata getrennt und haben die Reihenfolge „Textfarbe“, „Schattenfarbe“.

Neben den HTML-Farben (siehe Farbtabelle im Anhang) und den direkten Farbangaben in der Form „#RRGGBB“ mit hexadezimaler Angabe der Rot-, Grün- und Blauanteile, sind auch folgende Angaben zulässig:

DEFAULT

Bedeutet vordefinierte Farbwerte („black“, „gray“)

TRANSPARENT oder TRANSP

Bedeutet Transparenz, der Hintergrund wird nicht überschrieben und scheint durch

Beschreibung

Hiermit wird die Textfarbe eingestellt, die für die Laufschrift der gerade gesprochenen Wörter (sofern in „SetTextAnimation“ aktiviert) verwendet wird. Jeder Text wird mit einem Schatten angezeigt (der natürlich auf „TRANSPARENT“ gesetzt werden kann). Dadurch wird eine bessere Lesbarkeit erzielt. Das aktuell gesprochene Wort wird hierbei immer fett angezeigt.

Der Standardwert für den Text ist „black, gray“.

Beispiel

```
| <script language=JavaScript>
|   window.document.WebSpeech.SetTextColor("fuchsia,#00ff00");
| </script>
```

Setzt den Text auf die durch „fuchsia“ definierten RGB-Werte (#FF00FF) und den Schatten auf grün.

Siehe auch

TEXTCOLOR, TEXTANIMATION, SetTextAnimation, Farbtabelle in Anhang I

2.16 SetTextPosition

```
| int SetTextPosition(int nTextPosition);
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

nTextPosition

[in] Definiert die vertikale Ausrichtung der Textausgabe des WebSpeech-Fensters im Bereich „0 – 100“, „0“ bedeutet dabei oben, „100“ bedeutet unten. Standardmäßig befindet sich der Text oben („= 0“).

Beschreibung

Hiermit wird die Textposition eingestellt, die für die Laufschrift der gerade gesprochenen Wörter verwendet wird (sofern mit „SetTextAnimation“ aktiviert).

Beispiel

```
| <script language=JavaScript>
|     window.document.WebSpeech.SetTextPosition(25);
| </script>
```

Siehe auch

TEXTPOSITION, TEXTANIMATION, SetTextAnimation

2.17 SetTextSize

```
| int SetTextSize(int nTextSize);
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

nTextSize

[in] Definiert die Buchstabenhöhe der Textausgabe des WebSpeech Fensters in Pixeln.

Beschreibung

Hiermit wird die Textgröße eingestellt, die für die Laufschrift der gerade gesprochenen Wörter verwendet wird (sofern mit „SetTextAnimation“ aktiviert). Der Standardwert ist 12 Pixel.

Beispiel

```
| <script language=JavaScript>  
|     window.document.WebSpeech.SetTextSize(8);  
| </script>
```

Siehe auch

TEXTSIZE, TEXTANIMATION, SetTextAnimation

2.18 SetOpaque

```
| int SetOpaque(int nOpaque)
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

nOpaque

[in] Bestimmt, ob das Fenster, das für WebSpeech bestimmt wurde, transparent gezeichnet wird („nOpaque“ = „0“) oder stets deckend („nOpaque“ = „1“).

Beschreibung

Mit diesem Parameter kann eine deckende Ansicht des WebSpeech-Fensters erzwungen werden (Opaque=„1“). Hierfür wird die Hintergrundfarbe („BACKGROUND_COLOR“, bzw. „SetBackgroundcolor“) verwendet. Wenn eine transparente Darstellung des Hintergrunds erwünscht ist, sollte der Parameter Opaque angegeben und dessen Wert auf „0“ gesetzt werden.

Beispiel

```
| <script language=JavaScript>
|     window.document.WebSpeech.SetOpaque(0);
| </script>
```

Siehe auch

OPAQUE, BACKGROUND, SetBackgroundColor()

2.19 setBackgroundColor

```
| int setBackgroundColor(String strBackgroundColor);
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

strBackgroundColor

[in] Hiermit wird die Hintergrundfarbe des gesamten Objektes definiert. Neben den HTML-Farben (siehe Farbtabelle im Anhang) sind auch Farbangaben in der Form „#RRGGBB“ mit hexadezimaler Angabe der Rot-, Grün- und Blauanteile möglich.

Beschreibung

Das Objekt wird standardmäßig transparent gezeichnet. Der HTML-Hintergrund scheint also durch. Wenn eine transparente Darstellung nicht möglich ist oder nicht gewollt ist (vgl. „OPAQUE“ und „SetOpaque“), wird der Hintergrund mit dieser Farbe gefüllt.

Beispiel

```
| <script language=JavaScript>
|   window.document.WebSpeech.SetOpaque(1);
|   window.document.WebSpeech.SetBackgroundColor("forestgreen");
| </script>
```

Macht das WebSpeech-Fenster sichtbar und setzt die Hintergrundfarbe auf „waldgrün“.

Siehe auch

BACKGROUNDCOLOR, OPAQUE, SetOpaque()

2.20 SetControlPosition

```
| int SetControlPosition(int nControlPosition);
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

nControlPosition

[in] Definiert die vertikale Ausrichtung des Kontrollfensters im WebSpeech Fenster im Bereich „0 – 100“.

„0“ bedeutet dabei oben, „100“ bedeutet unten. Standardwert ist „75“.

Beschreibung

Hiermit wird die Position des Kontrollfelds eingestellt, das beim Überfahren mit der Maus erscheint.

Beispiel

```
| <script language=JavaScript>  
|     window.document.WebSpeech.SetControlPosition(25);  
| </script>
```

2.21 SetKey

```
| int SetKey(String strAuthKey);
```

Rückgabewert

< „0“ bedeutet Fehler

>= „0“ ok

Parameter

strLicenceKey

[in] der mitgelieferte Lizenzschlüssel

Beschreibung

Diese Funktion setzt den Lizenzschlüssel, der notwendig ist, um WebSpeech zu aktivieren. Mit dieser Funktion können Betreiber großer Domänen mehrere Lizenzschlüssel verwalten. Einen gültigen Lizenzschlüssel erhalten Sie bei

G DATA Software AG

Business-Vertrieb (B2B)

Königsallee 178 b

44799 Bochum

Telefon: 0234 / 97 62 - 170

E-Mail: b-vertrieb@gdata.de

Beispiel

```
| <script language=JavaScript>
|     window.document.WebSpeech.SetKey( "www.webspeech.de@RGVUA-CSN52-22B98-KU3WD" );
| </script>
```

Siehe auch

AUTHKEY

2.22 GetVersion

```
| String GetVersion();
```

Rückgabewert

Versionsstring, z.B. „4.0.2.27“.

Beschreibung

Liefert die Version des Objektes in Form einer Zeichenkette.

Beispiel

```
| <script language=JavaScript>  
|     var Version = window.document.WebSpeech.GetVersion();  
|     alert("Sie haben WebSpeech Version " + Version);  
| </script>
```

2.23 GetInterfaceVersion

```
| int GetInterfaceVersion();
```

Rückgabewert

Derzeit wird „4“ zurückgeliefert, zukünftige Interfaceversionen können höhere Werte liefern.

Beschreibung

Liefert die Version des Interfaces. Hiermit können Sie feststellen, welche Eigenschaften, Funktionen und Events implementiert sind. Diese Dokumentation bezieht sich auf Interface Version „4“.

Beispiel

```
| <script language=JavaScript>  
|     var Version = window.document.WebSpeech.GetInterfaceVersion();  
|     alert("Sie haben WebSpeech Interface-Version " + Version);  
| </script>
```

2.24 GetUserName

```
| String GetUserName();
```

Rückgabewert

Liefert den Namen des Anwenders.

Beschreibung

Sofern der Benutzer seine persönlichen Informationen nicht gesperrt hat, gibt diese Funktion den in den Einstellungen angegebenen Namen zurück. Ist diese Angabe gesperrt, wird ein leerer String zurückgeliefert.

Mit „GetUserGender()“ können Sie abfragen, ob der Anwender die Informationen gesperrt hat. In diesem Fall können Sie den Benutzer auch über das Speechtag „#WebSpeech=name#“ im zu sprechenden Text ansprechen.

Beispiel

Siehe „GetUserGender“ in Kapitel 5.27.

Siehe auch

Speechtags: #WebSpeech=name#

2.25 GetUserGreeting

```
| String GetUserGreeting();
```

Rückgabewert

Liefert die Anrede des Anwenders.

Beschreibung

Sofern der Benutzer seine persönlichen Informationen nicht gesperrt hat, gibt diese Funktion die in den Einstellungen von WebSpeech gewählte Anrede zurück. Ist diese Angabe gesperrt, wird ein leerer String zurückgeliefert.

Mit „GetUserGender()“ können Sie abfragen, ob der Anwender die Informationen gesperrt hat. In diesem Fall können Sie den Benutzer auch über das Speechtag „#WebSpeech=greeting#“ ansprechen.

Beispiel

Siehe „GetUserGender“ in Kapitel 5.27.

Siehe auch

Speechtags: #WebSpeech=greeting#

2.26 GetUserGender

```
| int GetUserGender();
```

Rückgabewert

„0“ für weiblich
 „1“ für männlich
 „-1“ für gesperrte Information

Beschreibung

Liefert das Geschlecht des Anwenders, sofern die Benutzung der persönlichen Informationen nicht gesperrt wurde. Mit „GetUserGender()“ können Sie auch abfragen, ob der Anwender die Informationen gesperrt hat.

Beispiel

```
<script language=JavaScript>
function Welcome() {
    var UserName=document.WebSpeech.GetUserName();
    var UserGreeting=document.WebSpeech.GetUserGreeting();
    var UserGender=document.WebSpeech.GetUserGender();
    MyWebSpeech=window.document.WebSpeech;
    // Info nicht gesperrt
    if (UserGender!=-1){

    // weiblich
    if (UserGender==0) {
        MyWebSpeech.SetText(UserGreeting + " liebe " + UserName);
        MyWebSpeech.StartSpeaking(); }

    // männlich
    if (UserGender==1) {
        MyWebSpeech.SetText(UserGreeting + " lieber " + UserName);
        MyWebSpeech.StartSpeaking(); }
    } // endof if (Info frei)

    // Ersatztext bei gesperrter Info
    else {
        MyWebSpeech.SetText("#WebSpeech=greeting#\# \#WebSpeech=name\#");
        MyWebSpeech.StartSpeaking(); }
    }
}</script>
```

Beim Aufruf dieser JavaScript-Funktion wird für männliche und weibliche Besucher ein unterschiedlicher Begrüßungstext gesprochen, der die Variablen für Benutzername und Benutzerbegrüßung enthält. Falls die Information gesperrt ist, wird ein neutraler Text gesprochen, der die Speechtags für Begrüßung und Namen enthält. Die Raute für die Speechtags sind mit einem Rückstrich auskommentiert.

Siehe auch

GetUserName, GetUserGreeting, Speechtags: #WebSpeech=name#, #WebSpeech=greeting#

3 Events

Durch die Events werden der HTML-Seite weitere Informationen über Zustandsänderungen mitgeteilt. Diese Events werden von WebSpeech aufgerufen, wenn sie in einer Scriptsprache in der HTML-Seite definiert sind. Die WebSpeech-Events sind keine Event-Handler (z.B. „OnClick“) denen in HTML-Tags (z.B. „<FORM>“) eine Funktion zugewiesen wird. Sie sind vielmehr Funktionen, die automatisch aufgerufen werden, wenn dieses Ereignis in der Sprachausgabe vorkommt. Sie müssen daher nur als Funktionen angegeben werden und brauchen später im „<BODY>“ nicht mehr aufgerufen zu werden.

Unter Internet Explorer werden Events regulär als echte ActiveX-Events ausgeführt und können entsprechend durch ein Script abgefangen werden.

Beispiel

Trägt das PlugIn den Namen „WebSpeech“ wäre z.B. der folgende Code denkbar:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function DoOnLoad(URL,Prozent)
{
    // Skriptbefehle wie zB:
    alert("Ladevorgang läuft");
}
</SCRIPT>

<SCRIPT LANGUAGE="VBScript">
Sub WebSpeech_OnLoad(URL,Prozent)
    DoOnLoad URL,Prozent
End Sub
</SCRIPT>
</HEAD>

<BODY>
<OBJECT
    ID="WebSpeech"
    CLASSID="CLSID:B38FEBBF-B2FD-11D3-BEC1-00500445FAEC"
    WIDTH="150" HEIGHT="125">
<PARAM NAME="AUTHKEY" VALUE="www.webspeech.de@RGVUA-CSN52-22B98-KU3WD">
</OBJECT>
</BODY>
</HTML>
```


Um Events kompatibel zu anderen Browsern zu halten, empfiehlt sich folgende Vorgehensweise. Es wird eine Eventfunktion in JavaScript geschrieben. Diese Funktion bestimmt, was bei dem Auftreten des entsprechenden Events passieren soll. Der Name dieser Funktion kann beliebig sein. Wir empfehlen den Event-Typ mit einem vorgesetzten „DoOn“. Dieser Konvention folgen auch die angegebenen Beispiele. Eine zweite Funktion für Internet Explorer wird in VBScript geschrieben. Ihr Name setzt sich zusammen aus dem Namen des Objektes, einem Unterstrich „_“ und dem Namen des Events. Im Body dieser Funktion wird nur die bereits in Javascript definierte Eventfunktion aufgerufen. Dabei ist auf die unterschiedliche Syntax für die Übergabe mehrerer Parameter in VBScript zu achten.

Konventionen für die Beschreibung der Events

In allen hier abgedruckten Beispielen ist der Objektname, der mit dem Attribut "ID" definiert wird, „WebSpeech“. Nach dem Unterstrich „_“ steht der Name des Events und in Klammern folgt der Datentyp des Parameters und ein möglicher Parametername. „[out]“ in der Beschreibung der Parameter symbolisiert, dass es sich um einen Rückgabewert handelt.

3.1 OnStartSpeaking

```
| object_OnStartSpeaking(int nReason)
```

Rückgabewerte

nReason
 „[out]“ = gibt an, warum es zu diesem Ereignis kam
 „0“ = normaler oder unbekannter Grund

Beschreibung

Eine Sprachausgabe wurde begonnen. Die angegebenen Anweisungen werden ausgeführt, wenn die Sprachausgabe begonnen wurde. Der Rückgabewert ist immer 0.

Beispiel

```
<script language="JavaScript">
function DoOnStartSpeaking(StartReason) {
// Funktionskern z.B.
    document.images[0].src="bild01.jpg";
}
</script>

<script language="VBScript">
sub WebSpeech_OnStartSpeaking(StartReason)
    DoOnStartSpeaking(StartReason)
end sub
</script>
```

Durch das VBScript wird lediglich die in JavaScript definierte Funktion aufgerufen. Dort wird ein Bild bestimmt, sobald die Sprachausgabe erfolgt.

3.2 OnStopSpeaking

```
| object_OnStopSpeaking(int nReason)
```

Rückgabewerte

nReason
 „[out]“ = gibt an, warum es zu diesem Ereignis kam
 „0“ = normaler oder unbekannter Grund

Beschreibung

Eine Sprachausgabe wurde beendet. Die angegebenen Anweisungen werden ausgeführt, wenn die Sprachausgabe gestoppt wurde. Der Rückgabewert ist immer 0.

Beispiel

```
<script language=JavaScript>
function DoOnStopSpeaking(StopReason) {
    // Funktionskern zB:
    win1.close();
}
</script>

<script language=VBScript>
sub WebSpeech_OnStopSpeaking(StopReason)
    DoOnStopSpeaking(StopReason)
end sub
</script>
```

Dieses Skript schließt das Fenster „win1“. So können Fenster geöffnet und nach dem Ende der Ansage automatisch wieder geschlossen werden.

3.3 OnPauseSpeaking

```
| object_OnPauseSpeaking(int nReason)
```

Rückgabewerte

nReason
 „[out]“ = gibt an, warum es zu diesem Ereignis kam
 „0“ = normaler oder unbekannter Grund

Beschreibung

Die Sprachausgabe wurde angehalten. Die angegebenen Anweisungen werden ausgeführt, wenn die Sprachausgabe unterbrochen wurde.

Beispiel

```
<script language=JavaScript>
function DoOnPauseSpeaking(PauseReason) {
    // Funktionskern zB:
    if (PauseReason!=1) {
        setTimeout('Erinnerung()', 60000);
    }
}
</script>

<script language=VBScript>
sub WebSpeech_OnPauseSpeaking(PauseReason)
    DoOnPauseSpeaking(PauseReason)
end sub
</script>
```

In dieser Funktion wird nach einer Minute die Funktion „Erinnerung()“ gestartet, wenn die Pause nicht von einer anderen Methode verursacht wurde. Der Aufruf erfolgt über das VBScript, wo lediglich die durch JavaScript definierte Funktion aufgerufen wird.

3.4 OnResumeSpeaking

```
| object_OnResumeSpeaking(int nReason)
```

Rückgabewerte

nReason

„[out]“ = gibt an, warum es zu diesem Ereignis kam

„0“ = normaler oder unbekannter Grund

„1“ = durch den Aufruf einer Methode

„2“ = durch den Anwender

Beschreibung

Eine Sprachausgabe wurde wieder aufgenommen. Die angegebenen Anweisungen werden ausgeführt, wenn die Sprachausgabe wieder aufgenommen wurde.

Beispiel

```
<script language=JavaScript>
function DoOnResumeSpeaking(Reason) {
    // Funktionskern z.B:
    window.status = 'Endlich darf ich weitersprechen.';
    return true;
}
</script>

<script language=VBScript>
sub WebSpeech_OnResumeSpeaking(Reason)
    DoOnResumeSpeaking(Reason)
end sub
</script>
```

3.5 OnMark

```
| object_OnMark(int nMarker)
```

Rückgabewerte

nMarker

[out] Identifikationsnummer der Markierung durch das „#mark=nMarker#“-Speechtag.

Beschreibung

Mit dem Speechtag MARK kann man Markierungen im Text setzen. Die Funktion OnMark wird aufgerufen, wenn eine Markierung im Text erreicht wurde. Damit ist es möglich, Aktionen mit den gesprochenen Texten zu synchronisieren. Dieses Event sollte „OnWord“ vorgezogen werden, um bestimmte Ereignisse mit dem gesprochenen Text zu synchronisieren.

Beispiel

```
<script language=JavaScript>
MyWebSpeech=window.document.WebSpeech;
MyWebSpeech.SetText("Beim Erreichen der Markierung wechselt die Seite. #mark=1#");
MyWebSpeech.StartSpeaking();

function DoOnMark(Marker) {
    if (Marker==1) {
        window.location.href="NextPage.htm";
    }
} // end of function WebSpeechEvent_OnMark
</script>

<script language=VBScript>
sub WebSpeech_OnMark(Marker)
    DoOnMark(Marker)
end sub
</script>
```

Siehe auch

Speechtags #MARK=nMarker#

3.6 OnClick

```
| object_OnClick(int nReason)
```

Rückgabewerte

nReason
„[out]“ = dieser Wert ist reserviert und ist immer „0“

Beschreibung

Dieses Ereignis wird ausgelöst, wenn der animierte Mund von WebSpeech mit der linken Maustaste angeklickt wird. Klickt der Benutzer gezielt auf eine Funktion des Controlpanels innerhalb des Mundfensters wird „OnClick“ nicht ausgelöst.

Beispiel

```
<script language=JavaScript>
function DoOnClick(Reason) {
    window.document.WebSpeech.SetText("Bitte nicht hier klicken.");
    window.document.WebSpeech.StartSpeaking();
}
</script>

<script language=VBScript>
sub WebSpeech_OnClick(Reason)
    DoOnClick(Reason)
end sub
</script>
```

3.7 OnMouthOver

```
| object_OnMouthOver(int iOutIn)
```

Rückgabewert

iOutIn

[out] ist der Wert von „iOutIn“ = „0“, ist die Mundgrafik mit dem Mauszeiger verlassen worden.
Ist der Wert = „1“ wurde der Mauszeiger auf den Mund bewegt.

Beschreibung

Wird der WebSpeech Mund mit dem Mauszeiger überfahren, wird dieses Ereignis mit „iOutIn=1“ ausgelöst. Solange der Mauszeiger innerhalb des Mundes bleibt, wird kein weiteres Ereignis ausgelöst, erst wenn dieser Bereich wieder verlassen wurde, wird „OnMouthOver“ ein weiteres Mal aufgerufen mit „iOutIn=0“.

Beispiel

```
<script language=JavaScript>
function SpeakIt(Text) {
    window.document.WebSpeech.SetText(Text);
    window.document.WebSpeech.StartSpeaking();
}
function DoOnMouthOver(OutIn) {
    if (OutIn == 1) {SpeakIt("bin drin.")}
    if (OutIn == 0) {SpeakIt("bin raus.")}
}
</script>

<script language=VBScript>
sub WebSpeech_OnMouthOver(OutIn)
    DoOnMouthOver(OutIn)
end sub
</script>
```


3.8 OnWord

```
| object_OnWord(String strWord,int nPos,int nLen,int nWordType)
```

Rückgabewerte

strWord

[out] Das gerade gesprochene Wort.

nPos

[out] Beim wievielten Zeichen eines Textes das Wort beginnt. Die Zählung beginnt mit „0“.

nLen

[out] Die Länge des gerade gesprochenen Wortes in Anzahl der Zeichen.

nWordType

[out] wird nicht mehr unterstützt und liefert den Wert "0" zurück.

Beschreibung

Das Wort „strWord“ wird gerade gesprochen.

„nPosition“ benennt die Position des Wortanfangs als Anzahl der Zeichen seit Beginn des Textes mit der Basis „0“. Die Länge „nLength“ gibt die Anzahl der Zeichen des Wortes an.

Beispiel

```
<script language=JavaScript>
i=0;
function DoOnWord(Wort,Position,Laenge,Wortart) {
    // Funktionskern zB:
    window.status="Wort " + (++i); return true;
}
</script>

<script language=VBScript>
sub WebSpeech_OnWord(Wort,Position,Laenge,Wortart)
    DoOnWord Wort,Position,Laenge,Wortart
end sub
</script>
```

Die globale Variabel „i“ wird in der Funktion „WebSpeechEvent_OnWord“ inkrementiert und bei jedem neuen Wort wird in der Statuszeile die Anzahl der gesprochenen Wörter angegeben.

3.9 OnLoad

```
| object_ OnLoad(String strURL, int nPercent)
```

Rückgabewerte

strURL

[out] Internet Adresse, die diesen Event ausgelöst hat.

nPercent

[out] Status des Textdownloads in Prozent.

Beschreibung

Es wird ein Text von einer HTML-Adresse geladen. Neben der URL „strURL“ wird auch der Status des Ladevorgangs „nPercent“ übergeben („0“ = Ladevorgang beginnt, „100“ = Ladevorgang fertig).

Beispiel

```
<SCRIPT LANGUAGE="JavaScript">
function DoOnLoad(URL,Prozent) {
    // Skriptbefehle wie zB:
    alert("Ladevorgang von " + URL + " gestartet.");
}
</SCRIPT>

<SCRIPT LANGUAGE="VBScript">
Sub WebSpeech_OnLoad(URL,Prozent)
    DoOnLoad URL,Prozent
End Sub
```

Abschnitt 5

Steuerung der Sprachausgabe in Texten und Textdateien

Die zu sprechenden Texte können entweder in den entsprechenden HTML-Code oder in Dateien geschrieben werden. Für die Textdateien bietet sich die Endung „.txt“ an. Dateien mit anderen Endungen, wie z.B. „.dat“, „.bat“ und „.htm“ werden ebenfalls als Textdateien angesehen, jedoch nicht weiter interpretiert.

I LGX-Dateien - das erste WebSpeech Format

WebSpeech unterstützt auch weiterhin das WebSpeech I Dateiformat. Bei der ersten Version von WebSpeech mussten die gesprochenen Textdateien die Endung „.lgx“ und einen Schlüssel zur Authentifizierung enthalten.

LGX-Dateien werden durch WebSpeech vom Internet Server selbständig nachgeladen. Im Augenblick sind dies reine Textdateien, die den zu sprechenden Text enthalten.

Eine typische LGX-Datei sah wie folgt aus:

```
C132K063-EE147S4Q-KIMHE8C5-GB32P5LC-VAJGCT4N
Ab der zweiten Zeile wird WebSpeech den Text vorlesen.
Der Text muss nicht speziell formatiert werden.
Er darf auch Steuercodes (Speechtags) enthalten.
```

Die erste Zeile enthält einen Lizenzierungsschlüssel, der bei WebSpeech nicht mehr abgefragt wird. Diese Zeile wird von WebSpeech ignoriert. Erst ab der zweiten Zeile wird der Text vorgelesen. Bei „.txt“-Dateien wird der Text direkt ab der ersten Zeile gelesen. Eine Ausnahme bilden URLs, die auf lokale Festplatten zugreifen. In diesem Fall findet keine Überprüfung des Schlüssels statt.

Die alten „.lgx“-Dateien können also weiterhin verwendet werden. Neue Dateien sollten jedoch als „.txt“-Dateien abgespeichert werden. Alle anderen Dateitypen werden direkt ab der 1. Zeile als ASCII-Text interpretiert.

2 Proprietäre Logox 4 Speichtags

Es ist auf vielfältige Weise möglich, die Sprachausgabe innerhalb eines Textes zu steuern. Hier lernen Sie, wie Sie die Sprechweise bestimmen, SpeechFonts wechseln, die Besucher Ihrer Webseite persönlich ansprechen können, wie Sie die Echtzeiteffekte von WebSpeech einsetzen und die Aussprache verändern können.

Speichtags können in den zu sprechenden Text integriert werden. Sie werden nicht mitgesprochen sondern von Logox interpretiert. Alle Tags werden von zwei Rauten („#“) umgeben. Vor der detaillierten Beschreibung der einzelnen Funktionen soll zunächst ein Überblick gegeben werden.

Die folgenden Speichtags sind proprietäre (nur für Logox 4 gültige) Steuerzeichen, die es erlauben, die Sprachausgabe über das von SAPI 5 vorgesehene Maß hinaus zu beeinflussen.

Übersicht

| Speechtag | Beschreibung | Wertebereich |
|-----------|--|---|
| BSPD | Grundlegende Sprechgeschwindigkeit eines Sprechers. | 10 - 400% |
| SPD | Sprechgeschwindigkeit relativ zu BSPD | 10 - 400% |
| BVOL | Grundlegende Lautstärke eines Sprechers | -100dB - +12dB 0 - 400 % |
| VOL | Lautstärke relativ zu BVOL | -100dB - +12dB 0 - 400 % |
| BPIT | Grundlegende Basistönhöhe der Stimme | 10 - 800 Hz |
| PIT | Basistönhöhe relativ zu BPIT | 10 - 800 Hz |
| MIDI | MIDI Notenwert als Notenname einer Oktave oder als Zahl | e-2 - g#4 10 - 800 Hz |
| BINT | Grundlegender Stimmumfang für einen Sprecher | 0 - 400% |
| INT | Stimmumfang relativ zu BINT | 0 - 400% |
| BROUGH | Grundlegende Rauheit einer Stimme | 0 - 100 Hz |
| ROUGH | Rauheit relativ zu BROUGH | 0 - 100 Hz |
| RST | Setzt die Werte von SPD, VOL, PIT, INT und ROUGH zurück | keine |
| PAU | Setzt eine Pause | 0 - 5000 ms |
| VOICE | Wechselt den SpeechFont [®] inkl. Stimme und allen zugehörigen Einstellungen. | SpeechFont [®] |
| POS | Bestimmt die Wortart des folgenden Wortes. | NOUN, NAME, VERB, ADJ, NUMB, PRON, ART, PREP, ADV, CONJ, PART, INTJ |
| FW | Das folgende Wort wird als Funktionswort betrachtet und abgeschwächt. | keine |
| CTX | Bestimmt, wie ein Text interpretiert wird. | SPELL |
| MARK | Marker, der eine Notification schickt, wenn die Position im Text erreicht ist. | 0 - 65535 |
| COM | Kommentar, dieser Text wird nicht vorgelesen. | beliebiger Text außer "#" |
| AFX | Parametersatz für die Audioeffekte | Control Code |
| SAMPA | Aussprache gemäß der SAMPA-Konvention. | ASCII Zeichen |
| EMPH | Das nächste Wort ist hervorgehoben (emphatisch). | keine |
| LASTACC | Das nächste Wort trägt den letzten Akzent im Satz. | keine |
| NOACC | Das nächste Wort ist nicht akzentuiert. | keine |
| NOIP | Unterbindet das Setzen einer großen melodischen Grenze an dieser Stelle im Text. | keine |
| NOAP | Unterbindet das Setzen einer kleinen melodischen Grenze an dieser Stelle im Text. | keine |
| NORP | Unterbindet das Setzen einer rhythmische Grenze an dieser Stelle im Text. | keine |

3 Syntax und allgemeine Eigenschaften

Während die SAPI 5 Tags dem XML-Standard folgen und oft mit Bereichen arbeiten, sind die Logox 4 spezifischen SpeechoTags eher punktuell wirksam. Sie gelten ab der Stelle, wo sie stehen, und wirken auf ein umliegendes Wort oder in manchen Fällen so lange, bis eine neue Anweisung folgt.

Ein SpeechoTag beginnt unmittelbar nach einem Rautenzeichen "#" und endet mit einem Rautezeichen "#" direkt nach dem letzten Argument. Nach der ersten Raute steht direkt der Tagname (ohne Leerzeichen). Groß- und Kleinschreibung der Tagbezeichnungen sowie deren Attribute und Werte wird in zwei Formaten akzeptiert: Alles klein oder alles groß. Die einzige Ausnahme bilden die Einheiten Hertz und Dezibel, die auch als "Hz" und "dB" akzeptiert werden.

Die Werte stehen nach einem Gleichheitszeichen "=", das auf den Tagnamen folgt. Die Werte können entweder absolut oder relativ angegeben werden. Absolut gilt als Default und wird nicht weiter gekennzeichnet. Wenn ein Wert relativ zu lesen ist, wird der Zahl ein "R" vorangestellt.

Die Einheit des angegebenen Wertes wird durch ein nachgestelltes Kürzel bestimmt:

- % = Prozent
- Hz = Frequenz in Hertz
- ST = Frequenz in Halbtönen (engl. semi-tone)
- dB = Lautstärke in Dezibel

Prozentangaben erfolgen in 1% Schritten. Die Werte für Hz, st und dB können auf eine Nachkommastelle genau angegeben werden. Bei Werten mit einer Nachkommastelle wird diese durch einen Punkt "." getrennt (kein Komma!!).

Unbekannte oder falsch aufgebaute Tags werden von Logox 4 ignoriert und im schlimmsten Fall vorgelesen.

Beispiele:

```
#PAU=500#
#pau=500#
#PIT=120Hz#
#PIT=R-20.4Hz#
#VOL=87%dB#
```

Die Beschreibung der SpeechoTags hat für jedes SpeechoTag den gleichen Aufbau. Zunächst wird dessen Syntax beschrieben. Dabei gelten folgende Konventionen:

Anstelle eines Wertes steht dessen Datentyp mit einer entsprechenden Bezeichnung. Mögliche Einheiten für ein SpeechoTag stehen in eckigen Klammern und sind ggf. durch Kommata getrennt.

Beispiel:

```
#SPD = short signedSpeed [%]#
```

Nach einer Beschreibung folgen Angaben über die möglichen Werte, Beispiele und Verweise auf ähnliche SpeechoTags.

3.1 BSPD

```
| #BSPD = short signedBaseSpeed [%]#
```

Jede Stimme hat einen voreingestellten Wert für die Sprechgeschwindigkeit. Mit BSPD kann dieser grundlegende Wert geändert werden.

BSPD ist nur ein Bestandteil des endgültigen Sprechgeschwindigkeitswerts. Außer BSPD beeinflusst auch der Wert des Speechoags SPD und ggf. der in den WebSpeech-Einstellungen gesetzte Wert die Sprechgeschwindigkeit. Alle drei Werte werden miteinander multipliziert und durch 10000 dividiert. Das Ergebnis ist die tatsächlich umgesetzte Sprechgeschwindigkeit. Ein Beispiel:

```
| BSPD = 120%; SPD = 80%; SetSPD = 150%
| Resultierende Sprechgeschwindigkeit: 120x80x150/10000 = 144
```

BSPD sollte nur verändert werden, wenn die Voreinstellungen für Ihre Zwecke nicht adäquat sind. Das variabelere Speechoag, das sich auch für häufige Wechsel der Sprechgeschwindigkeit eignet, ist SPD.

Werte

Die Werte für die Sprechgeschwindigkeit können nur prozentual angegeben werden. Ein Wert von 200% verdoppelt die Sprechgeschwindigkeit, ein Wert von 50 % halbiert sie. Zulässige Werte liegen zwischen 10% und 400%. Sinnvolle Werte liegen zwischen 66% und 200%.

Beispiele

```
| #BSPD=120%#
```

Siehe auch

SPD, BPIT, BVOL

3.2 SPD

```
#SPD = short signedSpeed [%]#
#SPD = R short signedSpeed [%]#
```

SPD ist das Speeachtag, das normalerweise für Veränderungen der Sprechgeschwindigkeit benutzt werden sollte. SPD bezieht sich auf den Wert von BSPD. Dabei wird der Wert von SPD mit BSPD multipliziert. Dadurch wirken sich die Änderungen mittels SPD auch bei unterschiedlichen Sprechern immer gleich aus.

Die Werte können aber nicht nur absolut angegeben werden, sondern auch relativ zum vorhergehenden Wert. Dem Wert wird dann ein R vorangestellt. Mit relativen Einstellungen wird die Sprechgeschwindigkeit verlangsamt oder beschleunigt unabhängig von den gerade aktuellen Einstellungen des Benutzers. Der aktuelle Wert wird um den angegebenen Prozentsatz variiert. Der aktuelle Wert ($BSPD \times \text{Benutzereinstellung} / 100$) wird um den relativen Wert von SPD subtrahiert. Ein Beispiel:

```
BSPD = 120%; SetSPD = 100%; SPD = R-50%
Aktuelle Sprechgeschwindigkeit: 120 x 100 / 100 = 120
Resultierende Sprechgeschwindigkeit: 120 - 50 = 70
```

Werte

Die Werte werden prozentual angegeben. 200 % führt zu einer Verdoppelung der Sprechgeschwindigkeit und 50% zu einer Halbierung. Zulässige Werte liegen zwischen 10% und 400%. Sinnvolle Werte liegen zwischen 66% und 200%.

Beispiele

```
#SPD=80%#
```

setzt die neue Sprechgeschwindigkeit auf 80% der Geschwindigkeit von BSPD.

```
#SPD=R40%#
```

setzt die neue Sprechgeschwindigkeit um 40% schneller.

```
#SPD=R-20%#
```

verringert die Sprechgeschwindigkeit um 20%

Siehe auch

BSPD, RST

3.3 BVOL

```
#BVOL = short signedBaseVol [dB,%]#
```

Für jeden Sprecher ist eine Grundlautstärke vordefiniert. BaseVolume definiert die Grundlautstärke für einen Sprecher neu. Normalerweise sollte für Änderungen der Lautstärke das Speechoag VOL verwendet werden. Mit beiden sollte die Lautstärke vornehmlich abgesenkt werden. Die Stimmen sind so optimiert, dass sie möglichst laut sind. Wenn die Lautstärke übermäßig angehoben wird, kann es zu Verzerrungen des Audiosignals kommen. Auch die Lautstärke errechnet sich aus 3 Parametern: Einstellungen durch den Benutzer, der Grundlautstärke BVOL und der relativen Lautstärke aus VOL. Die Prozentwerte werden miteinander multipliziert und durch 10000 geteilt (vgl. BSPD), die Dezibelwerte der einzelnen Komponenten werden addiert. Ein Beispiel:

```
SetVOL = -1.5 dB; BVOL = 1.2 dB; VOL = -6 dB  
Resultierende Lautstärke: -1.5 + 1.2 -4.7 = -5 dB
```

Werte

Die Werte können in Dezibel (dB) oder prozentual (%) angegeben werden. Intern wird mit Dezibelwerten gerechnet. Der Wertebereich liegt zwischen -100 dB und +12 dB in 0.1 dB Schritten. Eine Absenkung um 6 dB führt zu einer Halbierung der Energie des Signals. Eine Anhebung um 6 dB zu einer Verdoppelung der Energie. Sinnvolle Werte liegen zwischen -6dB und +1 dB. Entsprechend ist die Prozentskala aufgeteilt. 100% entsprechen 0dB, 200% + 6db und 50% -6dB. Die Prozentwerte können in 1%-Schritten angegeben werden. Gültige Werte liegen zwischen 10% und 400%. Sinnvolle Werte liegen im Bereich von 50% - 120%.

Beispiele

```
#BVOL=-3.2dB#  
#BVOL=50%#
```

Siehe auch

VOL, BSPD

3.4 VOL

```
#VOL = short signedVol [dB,%]#
#VOL = R short signedVol [dB,%]#
```

VOL setzt die Lautstärke bezogen auf die aktuelle Lautstärke, die sich aus der gesetzten Lautstärke (Einstellungen) und der Basislautstärke (BVOL) zusammensetzt. Lautstärkeänderungen sollten vornehmlich mit VOL vorgenommen werden. Die Änderungen können nicht nur absolut vorgenommen werden, sondern auch relativ. Relative Änderungen werden durch ein R vor dem Lautstärkewert gekennzeichnet.

Werte

Die Werte können in Dezibel (dB) oder prozentual (%) angegeben werden.

Intern wird mit Dezibelwerten gerechnet. Der Wertebereich liegt zwischen -100 dB und +12 dB in 0.1 dB Schritten. Eine Absenkung um 6 dB führt zu einer Halbierung der Energie des Signals. Eine Anhebung um 6 dB zu einer Verdoppelung der Energie. Sinnvolle Werte liegen zwischen -6dB und +1 dB. Entsprechend ist die Prozentskala aufgeteilt. 100% entsprechen 0dB, 200% + 6db und 50% -6dB. Die Prozentwerte können in 1%-Schritten angegeben werden. Gültige Werte liegen zwischen 10% und 400%. Sinnvolle Werte liegen im Bereich von 50% - 120%.

Die Lautstärke lässt sich nicht nur absolut setzen, sondern auch ausgehend vom aktuellen Lautstärkewert. Absolute Werte berechnen sich nach folgendem Schema:

```
SetVOL = -1.5 dB; BVOL = 1.2 dB; VOL = -6 dB
Resultierende Lautstärke: -1.5 + 1.2 - 4.7 = -5 dB
```

Relative Werte werden ausgehend von der aktuellen Lautstärke berechnet. Ein Prozentwert bezieht sich dann auf die Summe der dB-Werte von SetVOL und BVOL und berechnet gemäß der Prozentskala den neuen Wert. Dieser Wert wird wiederum mit VOL verrechnet. Ein Beispiel:

```
SetVOL = -2.4 dB; BVOL = 1.2 dB; VOL = -50%
```

Aktuelle Lautstärke: $-2.4 + 1.2 = -1.2$ dB

Umrechnung: 50% entspricht 6 dB

Resultierende Lautstärke: $-1.2 - 6\text{dB} = -7.2$ dB

Beispiele

```
#VOL=2.4dB#
#VOL=R-20%
```

Siehe auch

BVOL, RST, BSPD, BPIT

3.5 BPIT

```
| #BPIT = short signedBasePitch [Hz]#
```

Genauso wie für Lautstärke und Sprechgeschwindigkeit gibt es auch für die Basistönhöhe Voreinstellungen für jeden Sprecher. Diese vordefinierten Werte sind die Basis für die interne Berechnung der Sprechmelodie. Dieser Wert sollte also nur mit Bedacht geändert werden. Normalerweise sollten Änderungen der Tonhöhe mit dem Speehtag PIT vorgenommen werden.

Werte

Die Basistönhöhe kann nur in der Hertz-Skala angegeben werden.

Hertz Hz-Werte sind die interne Berechnungsgrundlage. Zulässige Werte liegen zwischen 10 Hz und 800 Hz. Sie sind zugänglich in 0.1 Hz Schritten.
Da die Hz-Skala ein lineares Maß für die Frequenz darstellt, kann der gleiche Unterschied in Hz in verschiedenen Frequenzbereichen zu einem unterschiedlichen Höreindruck führen. Sinnvolle Werte hängen vom Sprecher ab. Für Männerstimmen liegen sie zwischen 70 und 130 Hertz, für Frauenstimmen zwischen 140 und 240 Hertz.

Beispiele

```
| #BPIT=150Hz#
```

Siehe auch

PIT, MIDI, BVOL, RST, BSPD

3.6 PIT

```
#PIT = short signedPitch [Hz,ST,%]#
```

Die mittlere Tonhöhe für einen Sprecher berechnet sich aus den gesetzten Werten, der Basistonhöhe (BPIT) und dem relativen Wert PIT. Die Werte für PIT können nicht nur absolut gesetzt werden, sondern auch relativ zum gerade aktuellen Wert.

PIT sollte für Änderungen der Tonhöhe verwendet werden, nicht zuletzt deshalb, weil die vorherigen Werte wieder hergestellt werden können, indem man PIT auf 100% setzt.

Werte

Die Werte für PIT können in verschiedenen Skalen angegeben werden:

- | | |
|-------|---|
| Hertz | Hz-Werte sind die interne Berechnungsgrundlage. Zulässige Werte liegen zwischen 10 Hz und 800 Hz. Sie sind zugänglich in 0.1 Hz Schritten. Die resultierende Frequenz muss in diesem Bereich liegen. Da die Hz-Skala ein lineares Maß für die Frequenz darstellt, kann der gleiche Unterschied in Hz in verschiedenen Frequenzbereichen zu einem unterschiedlichen Höreindruck führen. Sinnvolle Werte hängen vom Sprecher ab. Für Männerstimmen liegen sie zwischen 70 und 130 Hertz, für Frauenstimmen zwischen 140 und 240 Hertz. |
| ST | Halbtöne (semitones) sind ein logarithmisches Maß für die Frequenz. Werte in Halbtönen lassen sich auch auf unterschiedliche Tonlagen übertragen. Die Halbtöne brauchen immer einen Bezugspunkt, von dem aus sie in Hertzwerte umgerechnet werden. Zulässige Werte liegen zwischen -24 und +24 Halbtönen in 0.1 ST Schritten. Sinnvolle Werte liegen zwischen -4 und +4 Halbtönen. Größere Änderungen führen zu klanglichen Einbußen. |

Beispiele

```
#PIT=25Hz#
Hallo, #PIT=R-2.5st# Hallo, #PIT=R-2.5st# Hallo.
Hallo, #PIT=-2.5st# Hallo, #PIT=-5st# Hallo.
```

Relative Werte beziehen sich auf den aktuellen Wert der Tonhöhe, auch wenn er gerade erst verändert wurde. Bei gleicher Ausgangsfrequenz führen die beiden letzten Beispiele zum gleichen Ergebnis.

Siehe auch

BPIT, MIDI, RST, BVOL

3.7 MIDI

```
#MIDI = short signedMIDINote short signedOktav#
#MIDI = short unsignedMIDICode#
```

Die Tonhöhe kann in Logox auch über MIDI Notenwerte bestimmt werden. Das ist besonders sinnvoll, wenn Lieder gesungen werden sollen.

Die MIDI-Notenwerte können in zwei Formaten angegeben werden:
Notenwert im Format Note-Oktav

```
#MIDI=<note><oktav>#
```

Notenwert im MIDI-Zahlenwert der Note

```
#MIDI=<code>#
```

Werte

<note> kann folgende Werte haben:

"c", "c#", "db", "d", "d#", "eb", "e", "f", "f#", "gb", "g", "g#", "ab", "a", "a#", "bb", "b", "c", "cis", "des", "d", "dis", "es", "e", "f", "fis", "ges", "g", "gis", "as", "a", "ais", "bb", "h"

<oktav> ist eine Oktavnummer zwischen -2 und 8.

Es gilt die Beschränkung auf Notenwerte zwischen 10 Hz und 800 Hz. Das ist der in MIDI definierte Notenbereich zwischen f-2 und g#4. Werte unterhalb von f-2 werden mit 10 Hz abgespielt. Werte, die höher als g#4 liegen, mit einer Frequenz von 800 Hz.

<code> ist ein Wert zwischen 5 und 80, der den durch MIDI spezifizierten Notenbereich (F-2 - G#4) abdeckt. Werte unterhalb von f-2 werden mit 10 Hz abgespielt. Werte, die höher als g#4 liegen mit einer Frequenz von 800 Hz.

Hinweis: In diesem Tag können innerhalb der Rautenzeichen des Tags auch die Rautenzeichen der Notennamen vorkommen. Das Tag endet in diesem Fall an der dritten Raute.

Beispiele

```
#MIDI=c#-l#
#MIDI=Bb3#
#MIDI=h2#
#MIDI=fis3#
#MIDI=58#
```

Siehe auch

BPIT, PIT

3.8 BINT

```
| #BINT = signed shortBaseInt [%]#
```

Der Umfang der Sprechmelodie kann mit BINT und INT bestimmt werden. BINT bezieht sich auf den voreingestellten Basiswert, der zusammen mit dem im IE eingestellten Wert für Stimmumfang und INT für den endgültigen Stimmumfang verantwortlich ist (vgl. BSPD).

Der Stimmumfang trägt zur Emotion bei. Ein kleiner Stimmumfang wird mit Trauer und Langeweile assoziiert. Ein erweiterter Stimmumfang wird als lebhaft und erregt wahrgenommen.

Änderungen des Stimmumfangs sollten normalerweise mit INT erfolgen.

Werte

Die Werte werden prozentual angegeben. Bei 100% findet keine Veränderung statt. Bei Werten unterhalb von 100% wird der Stimmumfang verringert, bei Werten darüber wird er erweitert (hoch wird höher und tief wird tiefer).

Gültige Werte liegen zwischen 0% und 400%. 200% verdoppelt den Stimmumfang, 50% halbiert ihn. Sinnvolle Werte liegen zwischen 30% und 150%.

Beispiele

```
| #BINT=60%#
```

Siehe auch

INT, BSPD, BPIT, PIT, BVOL

3.9 INT

```
| #INT = signed shortIntRange [%]#
```

Der Umfang der Sprechmelodie kann mit BINT und INT bestimmt werden. INT ist ein relativer Wert, der sich auf BINT und den per Einstellungen gesetzten Wert bezieht.

Der Stimmumfang trägt zur Emotion bei. Ein kleiner Stimmumfang wird mit Trauer und Langeweile assoziiert. Ein erweiterter Stimmumfang wird als lebhaft und erregt wahrgenommen.

Änderungen des Stimmumfangs sollten mit INT erfolgen, da hier auch relative Werte angegeben werden können und so die Original-Einstellungen wieder hergestellt werden können.

Werte

Die Werte werden prozentual angegeben. Bei 100% findet keine Veränderung statt. Bei Werten unterhalb von 100% wird der Stimmumfang verringert und bei Werten darüber wird er erweitert (hoch wird höher und tief wird tiefer).

Gültige Werte liegen zwischen 0% und 400%. 200% verdoppelt den Stimmumfang, 50% halbiert ihn. Sinnvolle Werte liegen zwischen 30% und 150%.

Die Werte können aber nicht nur absolut angegeben werden, sondern auch relativ zum vorhergehenden Wert. Dem Wert wird dann ein R vorangestellt. Die endgültigen Werte für den Stimmumfang berechnen sich nach dem gleichen Muster, wie bei BSPD und SPD.

Beispiele

```
| #INT=150%#
```

Siehe auch

BINT, RST, BSPD, SPD

3.10 BROUGH

```
| #BROUGH = unsigned shortBaseRough [Hz]#
```

Dieser Parameter macht die Stimme rau. Das kann am Ende eines Satzes aber manchmal beabsichtigt sein (John Wayne).

Wie bei BPIT und BVOL setzt sich der Wert für die Rauheit aus 3 Werten zusammen: BROUGH, ROUGH und SetROUGH (per Einstellungen gesetzt). BROUGH ist der voreingestellte Wert für eine Stimme. Änderungen der Rauheit sollten mit ROUGH erfolgen.

Werte

Der Wertebereich ist 0 bis 100 Hz in 0.1 Hz Schritten. Die Variation zwischen den einzelnen Perioden eines Sprechers liegt im Bereich von 1 - 3 Hz. Raue Stimmen werden durch Werte zwischen 20 und 100 Hz erzeugt. Die Natürlichkeit des Effektes kann noch erhöht werden, wenn zusätzlich in extrem rauen Passagen die mittlere Tonhöhe mit PIT abgesenkt wird.

Beispiele

```
| #BROUGH=1.5Hz#  
| #BROUGH=R15Hz#
```

Siehe auch

ROUGH, BVOL

3.11 ROUGH

```
| #ROUGH = unsigned shortRough [%]#
```

Dieser Parameter macht die Stimme rau. Das kann am Ende eines Satzes aber manchmal beabsichtigt sein (John Wayne).

Werte

Der Wertebereich ist 0 bis 100 Hz in 0.1 Hz Schritten. Die natürliche Variation liegt im Bereich von 1 - 3 Hz. Raue Stimmen werden durch Werte zwischen 20 und 100 Hz erzeugt.

Beispiele

```
| #ROUGH=90Hz#  
| #ROUGH=R-0.5Hz#
```

Siehe auch

BROUGH, RST , VOL

3.12 RST

```
#RST strType strType#  
#RST#
```

Setzt die Werte von VOL, PIT, SPD, INT und ROUGH zurück. Wenn keine Argumente übergeben werden, wird alles zurückgesetzt. Wenn nur einzelne akustische Merkmale zurückgesetzt werden sollen, stehen sie hinter einem Gleichheitszeichen durch Leerzeichen voneinander getrennt.

Werte

| | |
|-------|---|
| VOL | setzt die Lautstärke zurück auf den Wert von BVOL |
| PIT | setzt die Tonhöhe zurück auf den Wert von BPIT |
| SPD | setzt die Sprechgeschwindigkeit auf den Wert von BSPD |
| INT | setzt den Sprechstimmumfang auf den Wert von BINT |
| ROUGH | setzt die Rauheit der Stimme auf den Wert von BROUGH |

Beispiele

```
#RST#  
#RST=PIT VOL#
```

Siehe auch

VOL, PIT, SPD, INT, ROUGH

3.13 PAU

| #PAU = unsigned longDauer#

Fügt an der gesetzten Stelle im Text eine Pause ein. Dieses Speehtag setzt zusätzliche Pausen. Pausen innerhalb und am Ende von Sätzen sind unabhängig davon.

Werte

Dauer der Pause in Millisekunden (ms). Gültige Werte liegen zwischen 0 und 5000 ms. Für längere Pausen müssen mehrere Speehtags gesetzt werden.

Beispiele

| #PAU=500#

Fügt eine Pause von 500 ms ein.

3.14 VOICE

```
| #VOICE = NAME:strVoiceName#
```

Sprecherwechsel erfolgen mit dem Speechoption VOICE. Dazu können bestimmte SpeechFonts[®] angefordert werden.

Werte

NAME Der Sprechername ist für jeden SpeechFont[®] festgelegt und setzt sich aus drei Bestandteilen zusammen:

CompanyID: 3 Zeichen, z.B.: "LGX"

SprachID: 2 Zeichen (nach ISO 639), z.B.: "DE"

SprecherID: bis zu 8 Zeichen, z.B.: "BILL"

Während die ersten beiden Bestandteile festgelegt sind, darf die SprecherID beliebige Buchstaben, Ziffern, sowie die Sonderzeichen "-", "_" und "." enthalten.

Folgende Sprecheramen stehen momentan zur Verfügung:

LGX.DE.BILL

LGX.DE.SABINE

LGX.DE.MANDY

LGX.DE.KATHI

LGX.DE.SILKE

LGX.DE.MICHAEL

LGX.DE.MARTIN

Beispiele

```
| #VOICE=NAME:LGX.DE.BILL#
```

3.15 EMPH

#EMPH#

Das folgende Wort ist hervorgehoben (emphatisch). Dies wird erreicht, indem die Dauer der Laute des Wortes etwas verlängert wird, und das Wort einen Akzent bekommt.

Werte

Dieses Tag nimmt keine Werte.

Beispiele

| #EMPH#

Siehe auch

LASTACC, NOACC

3.16 POS

```
| #POS = strWortart#
```

Für die Berechnung der Aussprache, der Sprechmelodie und des Sprechrhythmus ist es hilfreich, wenn die Wortart eines Wortes bekannt ist. Sie kann mit dem Speechtag POS bestimmt werden.

Werte

| Wortart | Bedeutung | Beispiel |
|---------|------------------------------|-------------------|
| NOUN | Substantiv, Nomen, Hauptwort | Hase |
| NAME | Eigenname | Aida |
| VERB | Verb, Zeitwort, Tuwort | laufen |
| ADJ | Adjektiv, Eigenschaftswort | grün |
| NUMB | Zahlwort | zwölf |
| PRON | Pronomen | sie |
| ART | Artikel | der |
| PREP | Präposition | nach |
| ADV | Adverb | damals |
| CONJ | Konjunktion | weil, und |
| PART | Partikel | vielleicht, schon |
| INTJ | Interjektion | Ach! |

Beispiele

```
| #pos=NOUN#
```

Siehe auch

FW, SAMPA

3.17 FW

| #FW#

Eine sehr grobe Untergliederung von Wortarten ist die Unterscheidung zwischen Inhaltswörtern, die wichtige Informationen tragen, und inhaltsleeren Funktionswörtern, die oft reduziert ausgesprochen werden. Inhaltswörter sind Substantive, Adjektive, Verben und Adverbien. Zu den Funktionswörtern zählen Präpositionen, Artikel und Konjunktionen. Ein Funktionswort wird reduzierter ausgesprochen, und die Dauer seiner Laute wird verkürzt.

FW bezieht sich auf das nachfolgende Wort.

Werte

FW nimmt keine Werte.

Beispiele

| #FW#

Siehe auch

POS, SAMPA

3.18 CTX

```
#CTX = strContext#  
#CTX#
```

CTX beschreibt Kontexte und Formate, die zur Interpretation des Textes benutzt werden. Momentan wird nur ein Format unterstützt: SPELL (buchstabieren). Damit ist es möglich, Teile eines Textes buchstabieren zu lassen.

Werte

CTX wirkt ab der Position, wo es gesetzt wurde, so lange bis ein erneutes CTX Speechtag gesetzt wird. CTX ohne Argumente führt zurück zur normalen Textinterpretation.

SPELL Buchstabieren von Wörtern; Zahlen werden in einzelnen Ziffern vorgelesen.

Beispiele

```
#CTX=SPELL# Hallo. #CTX# Das ist ein Test.
```

Das Wort "Hallo" wird buchstabiert. Der darauf folgende Satz wird normal gesprochen.

3.19 MARK

```
| #MARK = longMark#
```

MARK setzt einen Bookmark, der synchron durch ITTSBufNotifySink zurückgemeldet wird. Dadurch ist es möglich, Markierungen im Text zu setzen. Wenn sie erreicht werden, erfolgt eine Rückmeldung, und es können entsprechende Aktionen gestartet werden. Bei ausgiebigem Gebrauch von MARK empfiehlt sich die Verwendung von Kommentaren mit COM.

Werte

MARK nimmt einen Zahlenwert. Gültige Werte liegen zwischen 0 und 65535.

Beispiele

```
| #MARK=1#
```

Siehe auch

COM

3.20 COM

```
| #COM = strComment#
```

COM erlaubt, in den zu sprechenden Text Kommentare zu schreiben, die nicht vorgelesen werden. Dies ist z.B. sinnvoll, um Markierungen zu kommentieren.

Werte

Alle ASCII-Zeichen außer "#"

Beispiele

```
| #COM=Grafik Jahresbilanz einblenden#
```

Siehe auch

MARK

3.21 Benutzer persönlich ansprechen

```
| #WEBSPEECH=[strParameter]#
```

Mit WebSpeech ist es möglich, die Benutzer der Webseite persönlich anzusprechen. Der Nutzer kann WebSpeech über die Registerkarte „Begrüßung“ seinen Namen, eine Begrüßung und sein Geschlecht mitteilen. Die Variablen für Begrüßung und Name können mit dem Speechtag WEBSPEECH von der Sprachausgabe genutzt werden, ohne auf die JavaScript-Variablen zuzugreifen, die per Default nicht zugänglich sind.

Mögliche Werte

„#WEBSPEECH=NAME#“ setzt den Namen des Benutzers ein

„#WEBSPEECH=GREETING#“ setzt die vom Benutzer gewählte Begrüßung ein

Als Default wird der Name „Benutzer“ und die Begrüßung „Hallo“ gesetzt.

Beispiel

```
| #WebSpeech=greeting# #WebSpeech=name#, willkommen bei G-Data.
```

Dieser Text könnte je nach den Angaben des Benutzers z.B. so klingen:

„Hallo Benutzer, willkommen bei G-Data.“ (Standard)

„Grüß Gott, Hartmut, willkommen bei G-Data.“

„Sehr geehrter Herr Professor Brinckmeyer, willkommen bei G-Data.“

„Lieber Schnuckelhase, willkommen bei G-Data.“

Siehe auch

GetUserName, GetUserGreeting

3.22 AFX

| #AFX = strAudioEffekt#

Logox 4 bietet eine Reihe von Audioeffekten:

| | |
|-----------|---|
| Balance | Positioniert die Stimme im Stereopanorama |
| Chorus | Zufällige kurze Echos, die die Stimme vervielfachen |
| Delay | Echo |
| Equalizer | Anhebung und Absenkung in verschiedenen Frequenzbereichen |
| Flanger | Moduliertes Echo zur Verfremdung von Stimmen |
| Gain | Anheben und Absenken der Lautstärke |
| Phaser | Phasenverschiebung des Signals |
| Reverb | Halleffekt |
| Shelving | Anheben und Absenken von hohen und tiefen Frequenzen |

Diese Audioeffekte können umfassend per Steuerzeichen angesteuert werden. Die resultierenden Parametersätze sind allerdings sehr komplex.

Werte

Die Menge der Steuerzeichen ist sehr umfangreich und deren Anordnung äußerst komplex. Daher können Steuerzeichen mit dem AudioFXEditor erstellt und über die Zwischenablage in den Text kopiert werden. Die Beschreibung dieses Tools finden Sie in "Abschnitt 6 Tools"

WICHTIG: Audioeffekte ausschalten!!

Bei der Verwendung von SpeechoTags zur Ansteuerung von Audioeffekten müssen am Ende des Textes die Audioeffekte wieder ausgeschaltet werden, sonst bleibt der Effekt bis zur Wahl eines neuen Sprechers auch für andere Texte erhalten. Das geschieht, indem man #AFX# ans Ende des Textes schreibt.

4 Ausführliche Beschreibung der Effekte

4.1 Balance

Der Balance-Effekt positioniert den Sprecher im Stereopanorama.

Werte

Alle Werte werden als Prozentwerte angegeben. 0% entspricht ganz links, 100% ist ganz rechts und 50% genau in der Mitte.

| Parameter | Min. Wert | Max. Wert | Einheit |
|-----------|-----------|-----------|----------------------|
| Balance | -1 | 1 | -1 linker Kanal; |
| | | | 1 rechter Kanal; |
| | | | 0 beide Kanäle; |

Beispiel

```
| #AFX = BA120=1:$121=1;# Hier spricht Logox 4 ganz rechts #AFX#.
```

4.2 Chorus

Der Choruseffekt erzeugt mehrere unabhängige, zufällig generierte Echos, deren Tonhöhe moduliert wird. Das erzeugt den Eindruck, als ob mehrere Stimmen aus einer einzigen Signalquelle erzeugt werden. Der Parameter "Delay" beschreibt die Verzögerungszeit der Echos und "Sweep" den Variationsbereich der Verstimmung. Die Dauer der Echos und deren Verstimmung wird in unterschiedlichen Zeitabständen aktualisiert, was mit dem Parameter "Rate" gesteuert wird. Die Stärke des Effekts wird mit "Depth" angegeben.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt.

| Parameter | Min. Wert | Max. Wert | Einheit |
|-----------|-----------|-----------|---------|
| Delay | 0 | 40 | ms |
| Sweep | 0 | 40 | ms |
| Rate | 0.1 | 10 | Hz |
| Depth | 0 | 1 | |

Beispiel

```
#AFX = CH115=1:$116=40.0,$117=20.0,$118=0.3,$119=0.7,0;# Hier spricht Logox 4 mit Chorus #AFX#.
```

4.3 Delay

Durch den Delay-Effekt wird ein Signal nach einem bestimmten Zeitintervall verzögert ausgegeben (Parameter "Delay"). Das Effektsignal kann gegenüber dem Originalsignal in der Lautstärke abgeschwächt werden. Wenn "Mix" auf 100% steht, ist das Effektsignal so laut wie das Original. Über den Parameter "Feedback" kann die Anzahl der Wiederholungen des Echos eingestellt werden. Allerdings ist der Effekt auf ca. 2 Sekunden Dauer beschränkt. Danach wird er ausgeblendet.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt.

| Parameter | Min. Wert | Max. Wert | Einheit |
|------------|-----------|-----------|---------|
| Delay | 0 | 1000 | ms |
| Mix | 0 | 1 | |
| Feedback | 0 | 1 | |

Beispiel

```
#AFX = DL122=1:$123=100.0,$124=0.4,$125=0.0;#
Hier spricht Logox 4 mit Echo #AFX#.
```


4.4 Grafischer Equalizer

Ein grafischer Equalizer verändert die Lautstärke eines Audiosignals in bestimmten Frequenzbereichen. Hier ist der Frequenzbereich in 8 Bänder aufgeteilt, die um 12 dB angehoben oder abgesenkt werden können. Eine Absenkung um 6 dB entspricht einer Halbierung der Energie. Eine Anhebung um 6 dB verdoppelt die Energie.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt.

| Parameter | Min. Wert | Max. Wert | Einheit |
|-----------|-----------|-----------|---------|
| 80 Hz | -12 | 12 | dB |
| 160 Hz | -12 | 12 | dB |
| 315 Hz | -12 | 12 | dB |
| 630 Hz | -12 | 12 | dB |
| 1.25 kHz | -12 | 12 | dB |
| 2.5 kHz | -12 | 12 | dB |
| 5 kHz | -12 | 12 | dB |
| 10 kHz | -12 | 12 | dB |

Beispiel

```
#AFX = EQ126=1:$127=-12,$128=-12,$129=-1,$130=0,$131=00,$132=0,$133=-12,$134=-12;#
Hier spricht Logox 4 über Telefon #AFX#.
```

4.5 Flanger

Der Flanger-Effekt verwendet zwei Delays mit zwei Abgriffen. Der erste Abgriff ist zeitlich fixiert ("Delay"), während der zweite Abgriff kontinuierlich verändert wird. "Sweep" bestimmt dabei den Abstand des zweiten Delays zum ersten; "Rate" die Modulationsgeschwindigkeit und "Depth" die Modulationstiefe des zweiten Delays. Beim Zusammenmischen der beiden Delays entstehen die charakteristischen schwebenden, sich auslöschenden Klangverfärbungen.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt.

| Parameter | Min. Wert | Max. Wert | Einheit |
|--------------|-----------|-----------|---------|
| Delay | 0 | 10 | ms |
| Sweep | 0 | 10 | ms |
| Rate | 0.1 | 10 | Hz |
| Depth | 0 | 1 | |
| Feedback | 0 | 1 | |

Beispiel

```
#AFX = CH135=1:$136=2.0,$137=2.0,$138=0.3,$139=0.7,$140=0.7;# Hier spricht Logox 4 mit
Flanger #AFX#.
```

4.6 Gain

Gain verändert den Lautstärkepegel des Signals. Dies ist sinnvoll, um Lautstärkeeinbußen auszugleichen. Gain ist besonders nützlich, wenn mit dem Equalizer Frequenzbereiche angehoben werden sollen. Um Verzerrungen zu verhindern, kann die Eingangslautstärke reduziert werden.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt. 50% entspricht keiner Veränderung. 0% senkt den Pegel um 6 dB ab und 100% hebt den Pegel um 6 dB an.

| Parameter | Min. Wert | Max. Wert | Einheit |
|-----------|-----------|-----------|---------|
| Gain | -6 | 6 | dB |

Beispiel

```
| #AFX = GA141=1:$142=-2;# Hier spricht Logox 4 etwas leiser als vorher #AFX#.
```

4.7 Phaser

Bei einem Phaser wird das Eingangssignal mit einer phasenverschobenen Kopie kombiniert. "Depth" bestimmt den Umfang der Modulation, "Range" deren Bereich und "Rate" deren Geschwindigkeit. "Feedback" bestimmt, wie oft eine Phasenverschiebung vorgenommen wird. Die dadurch entstehenden Filtereffekte verwischen den Klang ähnlich wie ein Flanger.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt.

| Parameter | Min. Wert | Max. Wert | Einheit |
|--------------|-----------|-----------|---------|
| Depth | 0 | 1 | |
| Range | 0 | 1 | |
| Rate | 0.1 | 10 | Hz |
| Feedback | 0 | 1 | |

Beispiel

```
#AFX = PH0=1:$1=+0.7,$2=+0.66666666,$3=+0.88749959,$4=+0.4;# Hier spricht Logox 4 mit
Phaser #AFX#.
```

4.8 Reverb

Ein Reverb- oder Halleffekt simuliert die akustischen Eigenschaften eines Raums. Ein akustisches Signal wird in einem Raum reflektiert. Neben der zeitlichen Verzögerung der ersten Reflexionen ("Predelay") sind dabei die Eigenschaften der diffusen, reflektierten Echos wichtig. Deren Frequenzgang kann eingeschränkt sein ("Bandwidth"). Auch deren Anzahl kann je nach Wandbeschaffenheit variieren. Bei einem hohen Wert für "Damping" werden nur wenige Echos reflektiert - wie in einem gut gedämpften Raum. "Decay" bestimmt die Länge des Ausklangs der Echos. Mit "Mix" wird der Anteil des Originalsignal mit dem Effektsignal bestimmt.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt.

| Parameter | Min. Wert | Max. Wert | Einheit |
|-----------|-----------|-----------|---------|
| Predelay | 0 | 100 | ms |
| Bandwidth | 0 | 1 | |
| Damping | 0 | 1 | |
| Decay | 0 | 1 | |
| Mix | 0 | 1 | |

Beispiel

```
#AFX = RV148=1:$149=67.6,$150=1.0,$151=0.4,$152=0.7,$153=0.15;# Hier spricht Logox 4
mit Hall #AFX#.
```

4.9 Shelving Filter

Ein Shelving Filter hebt die Frequenzbereiche oberhalb bzw. unterhalb einer bestimmten Grenzfrequenz an oder senkt sie ab. Dieses Filter ist weniger rechenintensiv als ein Equalizer und führt oft zum gleichen Ergebnis.

Werte

Die Wertebereiche sind in der folgenden Tabelle aufgelistet. Im AudioFXEditor werden die Werte in Prozent angegeben. Die Skalen sind entsprechend aufgeteilt.

| Parameter | Min. Wert | Max. Wert | Einheit |
|-----------|-----------|-----------|---------|
| Bass | -6 | 6 | dB |
| Treble | -6 | 6 | dB |

Beispiel

```
#AFX = SH154=1:$155=2,$156=2;# Hier spricht Logox 4 mit angehobenen Bässen und Höhen  
#AFX#.
```

5 SAMPA

5.1 Überblick

```
#SAMPA = strSAMPA;strWort;strWortart#
#SAMPA = strSAMPA;strWort;#
#SAMPA = strSAMPA;;strWortart#
#SAMPA = strSAMPA;;#
```

Im SAMPA-Speechtag sind die folgenden Bestandteile verpflichtend: je eine Raute (#) am Anfang und am Ende des Tags, die Bezeichnung des Tags (SAMPa), gefolgt von einem Gleichheitszeichen, die SAMPA-Umschrift und zwei Semikolons.

Das SAMPA Speechtag umfasst drei Bereiche.

1. Die Aussprache gemäß dem SAMPA Standard,
2. Das Wort, das in der WebSpeech Textanzeige dargestellt wird.
3. Die Wortart, die zur Bestimmung der Sprechweise verwendet werden soll.

Werte Aussprache

Die Lautschrift muss nach den oben beschriebenen Konventionen des SAMPA-Alphabets gebildet werden – ein ungültiges Zeichen führt zum Abbruch der Ausgabe für die restlichen Zeichen in der Lautschrift für dieses Wort – und sie muss Silbengrenzzeichen sowie Akzentmarkierung(en) enthalten. Das gilt insbesondere für Leerzeichen. Da sie kein Bestandteil der SAMPA Notation sind, führen sie zu einer fehlerhaften Ausgabe. Der Laut "Stille" wird durch einen Punkt ./ gekennzeichnet.

Wenn die Aussprache mit SAMPA bestimmt wird, müssen neben der Aussprache auch Silbengrenzen und Wortakzente gesetzt werden, sonst können Fehler in der Sprechmelodie und beim Sprechrhythmus auftreten.

Hier werden nur die Werte aufgelistet. Eine ausführliche Beschreibung des SAMPA-Alphabets finden Sie in den folgenden Ausführungen zum SAMPA-Speechtag

| | |
|--------------------------|---|
| Vokale | a, A, {, E, 8, e, l, i, O, o, U, u, 9, 2, Y, y, @, 6, a~, O~, E~, 9~, (ai),(au), (oy), (E6), (ei), (ou) |
| Unsilbische Vokale | i=, o=, 6= u= |
| Konsonanten | f, v, s, z, T, D, C, S, x, m, n, N, l, r, P_, p, B_, b, T_, t, D_, d, K_, k, G_, g, (ts), (tS), (dZ), (pf), h, ?, Q |
| Silbische Konsonanten | m=, n=, N=, l= |
| Diakritika | ., ' , " , - |

Werte Wort

beliebige ASCII-Zeichen außer "#" und ";

Werte Wortart

Es gelten die gleichen Werte, wie beim Speechtag POS.

NOUN, NAME, VERB, ADJ, NUMB, PRON, ART, PREP, ADV, CONJ, PART, INTJ

Beispiele

```
#SAMP=v'Aks-tu:-b@;Wachstube;NOUN#  
#SAMP=v'Aks-tu:-b@;Wachstube;#  
#SAMP=v'Aks-tu:-b@;;#
```

Die Felder für Wort und Wortart können auch leer sein

Siehe auch

POS, SAMPA-ausführlich, Sprechmelodie

5.2 Ausführliche Beschreibung

```
#sampa=strSAMPA;;#
#sampa=strSAMPA;strPOS#
#sampa=strSAMPA;strWORT;#
#sampa=strSAMPA;strWORT;strPOS#
```

Mittels des Speechtags "SAMPA" ist es möglich, in der Benutzereingabe die lautliche Repräsentation eines Wortes direkt an die Sprachausgabe zu übergeben. Der Lexikonzugriff von Logox wird mit diesem Speechtag umgangen. Mehrdeutige oder unsauber ausgesprochene Wörter können so korrigiert werden, oder es können Eigenheiten bei der Artikulation nachempfunden werden.

Mit dem SAMPA-Speechtag wird ein Wort in den Eingabetext eingefügt, das in phonetischer Umschrift mit dem SAMPA-Alphabet angegeben ist. Diese Lautschrift ist ein phonetisches Alphabet, das aus computerlesbaren Zeichen besteht. Jedes ASCII Zeichen dieses Alphabets ist einem bestimmten Sprachlaut zugeordnet. Einige Laute werden auch durch eine Folge von Zeichen repräsentiert. Die ausführliche Beschreibung des SAMPA-Alphabets erfolgt in Werte für die SAMPA-Umschrift.

Das SAMPA-Tag muss mindestens aus der SAMPA-Umschrift bestehen (strSAMPA). Diese Angabe steht direkt nach dem Gleichheitszeichen. Verpflichtend für ein gültiges SAMPA-Tag sind zudem zwei Semikolons und die das Tag abschließende Raute. Durch die Semikolons werden zwei weitere, optionale Parameter abgegrenzt. Zum Einen kann das gesprochene Wort in der Form angegeben werden, wie es gegebenenfalls als Text dargestellt werden soll (z.B. in Form einer Laufschrift, die mit der Sprachausgabe läuft) (strWORT). Zum Anderen kann eine Wortart angegeben werden, die dann in der Zuweisung der Intonation berücksichtigt wird (strPOS). Die Angabe der Wortform folgt der Angabe der SAMPA-Umschrift nach dem ersten Semikolon, die Angabe der Wortart nach dem zweiten. Wenn nur eine dieser Angaben gemacht wird, muss ihre Position nach dem ersten bzw. zweiten Semikolon dennoch gewahrt bleiben. Werden beide Parameter offen gelassen, so müssen die Semikolons trotzdem angegeben werden.

Werte

Die möglichen Werte für die Lautschrift richten sich nach den Konventionen, die im SAMPA-Alphabet festgeschrieben sind (vgl. [Werte für die SAMPA-Umschrift](#)). Neben den eigentlichen Lauten müssen auch Silbengrenzen und Wortbetonungen angegeben werden, da sonst die Sprechmelodie und der Sprechrhythmus leiden können. Ist in der Zeichenkette ein nicht definierter Wert enthalten, bricht die Sprachausgabe das Wort an der Stelle mit dem unbekannten Zeichen ab. Unter Umständen wird das Tag sogar ungültig und es wird ignoriert oder im schlimmsten Fall vorgelesen. In [Tricks zur Verbesserung der Aussprache](#) werden Hilfestellungen für eine möglichst natürlich klingende Umschreibung von Wörtern gegeben.

Für die Schriftform des Wortes wird eine an orthografischen Konventionen orientierte Eingabe erwartet. Satzzeichen sind zwar zulässig (mit Ausnahme des Semikolons), werden jedoch intonatorisch nur berücksichtigt, wenn sie außerhalb des Tags stehen.

Die Angabe der Wortart kann aus beliebigen Kombinationen von bis zu vier Großbuchstaben bestehen. Zwölf gültige Werte werden berücksichtigt, um Lautdauer und Intonation zu berechnen (vgl. [Werte zur Spezifizierung der Wortarten](#)). Ebenso wie eine fehlende Wortart-Angabe, werden unbekannte Werte auf den Default-Wert "NOUN" abgebildet.

Keine der Angaben der drei Parameter des SAMPA-Tags, Lautschrift, Orthografie oder Wortart, darf ein Semikolon oder eine Raute enthalten, da diese unweigerlich als Begrenzer des Parameters bzw. Ende des Tags interpretiert werden.

5.3 Werte für die SAMPA-Umschrift

Die Angabe der SAMPA-Umschrift innerhalb des SAMPA-Tags muss aus den Zeichen gebildet werden, die im SAMPA-Alphabet definiert sind. Es umfasst für die verschiedenen im Deutschen verwendeten Laute jeweils ein eigenes Zeichen oder eine Zeichenkombination. Darüber hinaus umfasst es einige Zusatzzeichen (sogenannte Diakritika), die keinem Lautwert entsprechen. Sie markieren Lautdauer oder Nasalisierung, unsilbische Vokale bzw. silbische Konsonanten und kennzeichnen Silbengrenzen und akzentuierte Silben. Wird ein Zeichen oder eine Zeichenkombination benutzt, die nicht den beschriebenen SAMPA-Zeichen entspricht, wird das Wort ab dem fehlerhaften Zeichen nicht mehr gesprochen. Womöglich wird sogar das gesamte Tag ungültig.

Vokale

SAMPA-Zeichen für Vokale unterscheiden zwischen gespannten (meist Kleinbuchstaben) und ungespannten Vokalen (meist Großbuchstaben). Gespannte Vokale werden auch als Langvokale bezeichnet, da sie in betonten Silben lang sind. Ungespannte Vokale sind immer kurz (außer in einigen Dialekten und in Fremdwörtern). Zudem sind einige Zusatzzeichen sog. Diakritika definiert, die Dauer, Nasalisierung und unsilbische Vokale markieren. Ein Doppelpunkt nach einem Laut verlängert dessen Dauer. So können alle Laute - nicht nur Vokale - verlängert werden. Langvokale brauchen kein gesondertes Dauerzeichen, da ihre Dauer durch die Betontheit bestimmt wird. Nur wenn eine zusätzliche Dehnung gewünscht ist, sollte ein Doppelpunkt folgen (vgl. "Albright", "Oeuvre"). Mit Logox 4 können auch die Nasalvokale aus französischen Fremdwörtern verwendet werden. Nasalvokal sind durch eine Tilde "~" gekennzeichnet (z.B. a~; s.u. [Nasalvokale](#)). Ein nachgestelltes Gleichheitszeichen kennzeichnet Vokale, die keine eigene Silbe bilden (i=; s.u. [Unsilbische Vokale](#)).

| Beispiel | SAMPA | Umschrift |
|-----------|-------|-------------------|
| Biene | 'i | B_b'i-n@ |
| vital | i | vi-T_t'al |
| bist | 'I | B_b'Ist |
| Leben | 'e | l'eB_b@n |
| Methan | 8 | m8-T_t'an |
| Räder | 'E | r'E-D_d6 |
| präzis | E | P_prE-T_(ts)'is |
| Heft | '{ | h'{fT_t |
| Staat | 'a | SD_d'aT_t |
| Stadt | 'A | SD_d'AT_t |
| Hof | 'o | h'of |
| Moral | o | mo-r'al |
| Post | 'O | P_p'OstT_t |
| Albright | 'O: | ? 'O:lB_br(ai)T_t |
| Bude | 'u | B_b'uD_d@ |
| kulturell | u | K_kU1-T_tu-r'{l |
| Pult | 'U | P_p'U1T_t |
| Ökonom | 2 | ?2-K_ko-n'om |
| Öl | '2 | ? '2l |
| göttlich | '9 | G_g'9T_t-lIC |
| Oeuvre | '9: | ? '9:vr@ |
| Mykene | y | my-K_k'e-n@ |

| | | |
|--------|----|----------|
| Bücher | 'y | B_b'y-C6 |
| füllt | 'Y | f'YlT_t |

Diphthonge

Obwohl Diphthonge aus einer Folge von Vokalen bestehen, gelten sie als ein Laut. Um Verwechslungen mit echten Vokal-Vokal Abfolgen zu vermeiden, werden die beiden Bestandteile eines Diphthongs in Klammern eingeschlossen.

| Beispiel | SAMPA | Umschrift |
|----------|-------|-----------------|
| Weit | (ai) | v'(ai)T_t |
| Haut | (au) | h'(au)T_t |
| Heu | (oy) | h'(oy) |
| Verband | (E6) | f(E6)-B_b'AnD_d |
| Lady | (ei) | l'(ei)-D_di |
| Windows | (ou) | v'In-D_d(ou)s |

Nasalvokale

Nasalierte Vokale, die häufig in französischen Fremdwörtern vorkommen, werden mit der Tilde (~) markiert.

| Beispiel | SAMPA | Umschrift |
|------------|-------|--------------|
| Timbre | E~ | T_t'E~-B_br@ |
| Restaurant | a~ | r{sT_tor'a~ |
| Fondue | O~ | fO~D_d'y |
| Parfum | 9~ | P_parf'9~ |

Schwache Vokale

Schwache Vokale sind der Mummellaut "Schwa" (/@/) und sein Pendant vor "r", das r-Schwa (oder "Schwar") /6/. Auch der Diphthong (E6) in den Vorsilben "ver-", "zer-" und "er-" ist ein reduzierter Vokal.

| Beispiel | SAMPA | Umschrift |
|----------|-------|-----------------|
| halte | @ | h'Al-T_t@ |
| Ober | 6 | ? 'o-B_b6 |
| Verband | (E6) | f(E6)-B_b'AnD_d |

Unsilbische Vokale

Normalerweise bilden Vokale den Kern einer Silbe. In bestimmten Kontexten können Vokale sich aber an andere Silben anhängen und werden "unsilbisch". Unsilbische Vokale sind Teil derselben Silbe wie der vorangehende bzw. folgende Vokal.

| Beispiel | SAMPA | Umschrift |
|----------|-------|----------------|
| Nation | i= | nA-T_(ts)i='on |

```

Francois    o=      fra~-so='A
akuell      u=      QAK_k-T_tu=' {l
Wirt        6=      ?v'i6=T_t

```

Konsonanten

Im SAMPA-Alphabet sind Konsonanten teilweise durch den Kleinbuchstaben repräsentiert, der auch orthografisch diesen Laut abbildet (z.B. f, h, l, m und n). Für andere Konsonanten stimmen SAMPA-Zeichen und orthografische Schreibweise nicht überein (besonders C, Z, x, D, T). Für verlängerte Konsonanten kann, wie auch für Langvokale, der Doppelpunkt (:) als Längezeichen verwendet werden. Lange Konsonanten (z.B. /n:/) treten beispielsweise in französischen Lehnwörtern auf.

| Beispiel | SAMPA | Umschrift |
|----------|-------|------------|
| Fass | f | f'As |
| was | v | v'As |
| Skat | s | sK_k'aT_t |
| Sonne | z | z'O-n@ |
| Schal | S | S'al |
| Genie | Z | Z8n'i |
| ich | C | ? 'IC |
| Bach | x | B_b'Ax |
| hat | h | h'AT_t |
| ja | j | j'a |
| Last | l | l'AsT_t |
| Rast | r | r'AsT_t |
| Mast | m | m'AsT_t |
| Naht | n | n'aT_t |
| lang | N | l'AN |
| this | D | D'Is |
| Thing | T | T'IN |
| Garonne | n: | G_gA-r'On: |

Plosive und Affrikaten

Plosive (p,t,k,b,d,g) und Affrikaten (pf, ts, tS, dZ) werden in mehrere Laute aufgeteilt und aus einer Abfolge von Lauten gebildet. In der SAMPA-Umschrift in Logox wird jeder ihrer Bestandteile als eigener Laut repräsentiert. Plosive und Affrikaten werden durch eine Verschlussphase und eine Lösungsphase gebildet. Die Verschlussphase wird durch einen Großbuchstaben mit Unterstrich notiert. Der zweite Bestandteil eines Plosivs ist die Lösung des Verschlusses und das damit verbundene Plosionsgeräusch. Es wird durch den entsprechenden Kleinbuchstaben repräsentiert. Bei Affrikaten geht die Lösung des Verschlusses in ein Reibegeräusch über. Obwohl dies als Abfolge von Konsonanten aufgefasst werden könnte, zählen Affrikaten als ein Laut. Ähnlich wie Diphthonge werden die beiden Lautbestandteile in Klammern eingeschlossen, um Verwechslungen mit echten Lautfolgen zu verhindern.

| Beispiel | SAMPA | Umschrift |
|----------|-------|-------------|
| Ball | B_b | B_b'Al |
| Pakt | P_p | P_p'AK_kT_t |

| | | |
|-----------|--------|--------------|
| dann | D_d | D_d'An |
| Tal | T_t | T_t'al |
| Gast | G_g | G_g'AsT_t |
| kalt | K_k | K_k'AlT_t |
| Zahl | T_(ts) | T_(ts)'al |
| Matsch | T_(tS) | m'AT_(tS) |
| Pfahl | P_(pf) | P_(pf)'al |
| Dschungel | D_(dZ) | D_(dZ)'u-N@l |

Glottalverschluss

Ein Glottalverschluss steht im Deutschen zumeist vor Vokalen am Wortanfang oder vor betonten Silben, die mit einem Vokal anfangen (z.B. "am" in "Beamte"). Er wird in Logox in zwei Varianten realisiert. In der Regel steht vor betonten Vokalen ein stärkerer Verschluss (/ʔ/), die abgeschwächte Variante (/Q/) wird vor unbetonten Vokalen benutzt.

| Beispiel | SAMPA | Umschrift |
|-----------|-------|------------------|
| Amt | ʔ | ʔ'AmT_t |
| offiziell | Q | QO-fi-T_(ts)i='l |

Silbische Konsonanten

Normalerweise gruppieren sich Konsonanten um Vokale zu einer Silbe. Unter bestimmten Bedingungen können Konsonanten aber selbst eine Silbe bilden. Sie werden "silbisch". Silbische Konsonanten werden durch Anhängen des Gleichheitszeichens (=) markiert. Das Gleichheitszeichen kennzeichnet den Wechsel der Silbizität.

| Beispiel | SAMPA | Umschrift |
|----------|-------|-------------|
| Nabel | l= | n'a-B_bl= |
| großem | m= | G_gr'o-sm= |
| baden | n= | B_b'a-D_dn= |
| legen | N= | l'e-G_gN= |

Der Laut "Stille"

Wenn man verhindern möchte, dass ein Laut an den nächsten überbunden wird, kann man das Lautsymbol für Stille einfügen. Es ist ein Punkt /. Der Punkt sollte aber nicht dazu dienen lange Pausen einzufügen.

| Beispiel | SAMPA | Umschrift |
|---------------|-------|------------------|
| Wurst Fischer | | v'U6=sT_t.f'I-S6 |

Diakritika

Neben den Zeichen, die selbst Laute repräsentieren, gibt es im SAMPA-Alphabet Zusatzzeichen (sogenannte Diakritika), die diese Laute modifizieren. Ein Doppelpunkt nach einem Laut zeigt an, dass

dessen Dauer verlängert wird (z.B. /O:/). Weiterhin werden Silbengrenzen mit dem Minuszeichen (-) gekennzeichnet und Akzente mit dem Apostroph (') oder doppelten Anführungszeichen ("). Es ist wichtig, dass Silbengrenzen und Betonungszeichen korrekt gesetzt sind, da es sonst zu Fehlern im Sprechrhythmus und in der Sprechmelodie kommen kann.

| Bedeutung | SAMPA | Beispiel |
|-----------------|-------|-------------------|
| Hauptakzent | ' | v 'AlD_d |
| Nebenakzent | " | v 'AlD_d-r "AnD_d |
| Silbengrenze | - | z 'O-n@n-S" (ai)n |
| Dehnungszeichen | : | v 'a:l |

Wichtiger Hinweis

Da das SAMPA-Tag den Lexikonaufruf umgeht, muss die SAMPA-Umschrift in diesem Tag alle Zeichen enthalten, die im Lexikon innerhalb der Phonemkette erzeugt werden. Das sind zum einen die Lautzeichen selber, zum anderen aber auch die Silbengrenzen sowie die Markierung der betonten Silben. Silbengrenzen und Betonungen sollten in der Umschrift im SAMPA-Speechtag mitangegeben werden, da sie in der weiteren Verarbeitung für die Berechnung der Lautdauer und für die Intonation relevant sind.

Eine Silbengrenze wird zwischen zwei Silben markiert. Das Minuszeichen steht nach dem letzten Laut der vorherigen und vor dem ersten Laut der folgenden Silbe. Wortanfang und –ende werden automatisch als Silbengrenze interpretiert. In einsilbigen Wörtern muss keine Silbengrenze angegeben werden.

In einem Wort werden bestimmte Silben betont. Diese Silben tragen die Zielpunkte der Sprechmelodie. Daher sollte in der betonten Silbe eines Wortes ein Betonungszeichen gesetzt (Apostroph bzw. Anführungszeichen) werden, und zwar jeweils direkt vor den ersten Vokal der Silbe, bei Diphthongen außerhalb der Klammer. Bei Folgen von Glottalverschluss und Vokal steht das Betonungszeichen zwischen beiden Zeichen, in Kombinationen aus einem Vokal und einem unsilbischen Vokal trägt der Vollvokal die Markierung der Betonung. Der Apostroph steht in der Silbe, die den Hauptakzent trägt. In einem Wort, das nur aus einer einzigen Silbe besteht, muss in dieser Silbe immer ein Hauptakzent markiert werden. In langen, zusammengesetzten Wörtern können neben dem Hauptakzent noch mehrere Nebenakzente vorkommen. Diese Silben werden durch das doppelte Anführungszeichen markiert.

Tricks zur Verbesserung der Aussprache

Die Aussprache von Wörtern kann man in Aussprachewörterbüchern (z.B. Duden Band 6 "Das Aussprachewörterbuch") nachschlagen. Die dort gemachten Angaben sind jedoch auf die Aussprache einzelner Wörter ausgelegt und daher im Satzzusammenhang oft überpräzise. In einigen Fällen ist es von Vorteil, von der optimalen Wortaussprache abzuweichen. Manchmal muss die Aussprache an den Satzkontext angepasst werden, da Logox ja meist Sätze vorliest, manchmal klingt auch die "richtige" Aussprache unnatürlicher als die, die Sprecher im normalen Sprachgebrauch tatsächlich produzieren.

Statt /St/ und /Sp/ eher /Sd/ und /Sb/

In den Konsonantenfolgen "st" und "sp" steht gemäß der Standardaussprache nach /S/ der stimmlose Plosiv. Die Sprenglaute /T_t/ bzw. /P_p/ klingen jedoch in diesem Kontext zu stark. Besser klingen diese Folgen, wenn der entsprechende stimmhafte Plosiv gewählt wird, also /SD_d/ bzw. /SB_b/.

Beispiel

"st" in "Stadt" /**ST**_t'aT_t/ besser klingt /**SD**_d'aT_t/
 "sp" in "Spiel" /**SP**_p'il/ besser klingt /**SB**_b'il/.

Der unsilbische Vokal /6=/und die Realisierung von /r/ als Vokal

Der unsilbische schwache Vokal /6=/ steht in der Regeln in Kontexten, in denen ein Vokal von "r" gefolgt wird. Wenn /r/ im Silbenauslaut steht oder das Silbenende bildet, wird es in der SAMPA-Umschrift vokalisiert und in /6=/ umgewandelt. Wird das /r/ nach einem Vokal zur nächsten Silbe gezählt, so bilden der Vokal und /6=/ den Auslaut der einen und /r/ den Anlaut der nächsten Silbe.

Die silbische Variante des schwachen Vokals /6/ ersetzt normalerweise am Wortende Folgen des schwachen Vokals /@/ vor /r/. Wenn die Silbe am Wortende auf /r/ endet, wird /@r/ zu /6/, wenn /r/ zur nächsten Silbe gehört, wird daraus /6-r/.

Beispiel

| SAMPA | Beispiel | Umschrift /6=/ | Umschrift/6=-r/ oder /6-r/ |
|-------|-------------------|------------------|----------------------------|
| i:6= | Tier(e) | T_t'i6= | T_t'i6=-r@ |
| I6= | Wirt | v'I6=T_t | |
| y:6= | Tür(en) | T_t'y6= | T_t'y6=-r@n |
| Y6= | Türke | T_tY6=-K_k@ | |
| e:6= | schwer(e) | Sve6= | Sve6=-r@ |
| {6= | Berge | b{6=-G_g@ | |
| E:6= | Bär(en) | bE6= | bE6=-r@n |
| 2:6= | Föhr(en) | f26= | f26=-r@n |
| 96= | Wörter | v'96=-T_t6 | |
| a:6= | Haar(e) | ha6= | ha6=-r@n |
| a6= | hart | ha6=T_t | |
| u:6= | Kur(en) | K_ku6= | K_ku6=-r@n |
| U6= | kurz | K_kU6=T_ts | |
| o:6= | Ohr(en) | o6= | o6=-r@n |
| O6= | dort | D_dO6=T_t | |
| 6 | besser (e) | B_b'E-s 6 | B_b'E-s 6 -r@ |

Glottalverschlüsse

Vor Vokalen am Wortanfang steht ein Glottalverschluss, und zwar auch dann, wenn ein Wort in Kombination mit anderen Wörtern ein komplexes Wort bildet. In Logox gibt es eine starke und eine abgeschwächte Variante des Glottalverschlusses. Die starke Variante (/ʔ/) steht vor dem Anfangsvokal in Funktionswörtern (z.B. auf, als, ein) und vor betonten Vokalen am Anfang von Inhaltswörtern (z.B. Ufer, albern, eng). Vor unbetonten Vokalen steht die schwächere Variante des Glottalverschlusses (/Q/).

Beispiel

/ʔ/ /ʔ'(au)f/ auf

| | | |
|-----|-------------------------------|------------|
| | /?'(ai)n/ | ein |
| | /?'uf6/ | Ufer |
| | /?'A6=-B_b(ai)T_(ts)-?'AmT_t/ | Arbeitsamt |
| /Q/ | /Q(E6)-f'A6=-rUN/ | Erfahrung |
| | /Qy-B_b6-h'o-l@n/ | überholen |

Verschluckte Endsilben – Silbische Konsonanten

Am Wortende werden Silben, die den Mummellaut /@/ ("Schwa") enthalten, oft verschluckt. Das Schwa fällt weg und es bleibt nur der darauffolgende Konsonant übrig. Statt des schwachen Vokals wird dadurch der Konsonant zum Silbenträger. Solche Konsonanten werden silbisch genannt und durch ein angehängtes /=/ gekennzeichnet.

Beispiel

| | | | |
|---------|-------------|------|--------------|
| "haben" | /h'a-B_b@n/ | wird | /h'a-B_bn= / |
| "rotem" | /r'o-T_t@m/ | wird | /r'o-T_tm= / |
| "legen" | /l'e-G_g@n/ | wird | /l'e-G_gn= / |

Die Aussprache mit dem silbischen Konsonanten ist zwar undeutlicher, klingt aber im Satz oft natürlicher als die genormte.

Darüber hinaus wird am Wortende der silbische Konsonant oft an die Artikulationsstelle des vorhergehenden Konsonanten angepasst (zum Beispiel /P_p/, /B_b/ und /m/ werden alle mit verschlossenen Lippen gebildet, und /K_k/, /G_g/ und /N/ werden in gleicher Weise am Gaumen artikuliert). Diese Aussprache klingt ebenfalls natürlicher, kann aber auf Kosten der Deutlichkeit gehen.

Beispiel

| | | | |
|---------|--------------|------|---------------|
| "haben" | /h'a-B_b@n/ | wird | /h'a-B_bn= / |
| "legen" | /l'e:-G_g@n/ | wird | /l'e:-G_gN= / |

Vokal am Wortende

Vokale am Wortende sind in der Regel länger als im Wort, obwohl sie nicht in betonten Silben stehen. In der SAMPA-Umschrift kann nach solchen Vokalen ein Doppelpunkt (:) stehen.

Beispiel

| | |
|----------|-----------------|
| "Eskimo" | /?'{s-K_kI-mo:/ |
| "Villa" | /v'I-lɑ:/ |

Langvokale im Englischen

Ungespannte Langvokale in englischen Wörtern klingen in der normalen Umschrift, die sich an deutschen Phonemen orientiert, manchmal zu kurz. Wenn der Doppelpunkt zur Dehnung nicht ausreicht, kann dasselbe Zeichen in der Umschrift zweimal notiert werden (z.B. /OO/).

Beispiel

"Call" /K_k'OOl/

Fremdwörter

In Fremdwörtern wird in der Standardaussprache meist davon ausgegangen, dass nur gespannte Vokale verwendet werden (das sind die sog. Langvokale, die in SAMPA klein geschrieben werden). Für häufig verwendete Fremdwörter klingt oft die Aussprache mit den ungespannten Vokalen (groß geschrieben in SAMPA) vertrauter, da dies eher der Umgangssprache entspricht. Welche Umschrift man in diesen Fällen wählt, ist allerdings eine Frage des Stils und ist daher dem Verfasser des zu sprechenden Textes freigestellt.

Beispiel

"Politik" /P_poliT_t'IK_k/ (Standardumschreibung)
 /P_pOlIT_t'IK_k/ (umgangssprachliche Umschrift)

Abkürzungen

Abkürzungen werden manchmal falsch ausgesprochen, zum Beispiel dann, wenn sie mit einem Wort (oder Wortteil) übereinstimmen. Die gewünschte Aussprache kann dann im SAMPA-Tag in der Umschrift angegeben werden. In buchstabierten Abkürzungen ist in der Regel nur die letzte Silbe (also die, die dem letzten Buchstaben entspricht) betont. Vokalbuchstaben werden durch gespannte Langvokale repräsentiert. Vor den Vokalen steht ebenso wie vor Vokalen, die beim Buchstabieren vor Konsonanten eingefügt werden (z.B. "s" wird /{s/ oder "n" wird /{n/ buchstabiert), ein Glottalverschluss. Es steht /?/ in betonter und /Q/ in unbetonter Position.

Beispiel

"UN" /Qu:-?'{n/
 "ISS" /Q(ai)-Q{s-?'{s/

Werte zur Spezifizierung der Wortarten

Für ein Wort, das in einem SAMPA-Tag übergeben wird, kann in diesem Tag eine Wortart angegeben werden. Die folgenden Wortarten können verwendet werden. Wird keine Wortart angegeben, wird das Wort als Nomen (NOUN) interpretiert.

| Wortart | Bedeutung | Beispiel |
|---------|-------------------|-----------|
| NOUN | Substantiv, Nomen | Hase |
| NAME | Eigennamen | Aida |
| VERB | Verb | laufen |
| ADJ | Adjektiv | grün |
| NUMB | Zahlwort | zwölf |
| PRON | Pronomen | sie |
| ART | Artikel | der |
| PREP | Präposition | nach |
| ADV | Adverb | damals |
| CONJ | Konjunktion | weil, und |
| PART | Partikel | ja, noch |
| INTJ | Interjektion | ähm, tja |

5.4 Beispiele

Im SAMPA-Speechtag sind die folgenden Bestandteile verpflichtend: je eine Raute (#) am Anfang und am Ende des Tags, die Bezeichnung des Tags (SAMPa), gefolgt von einem Gleichheitszeichen, die SAMPA-Umschrift und zwei Semikolons.

Die Lautschrift muss nach den oben beschriebenen Konventionen des SAMPA-Alphabets gebildet werden – ein ungültiges Zeichen führt zum Abbruch der Ausgabe für die restlichen Zeichen in der Lautschrift für dieses Wort – und sie muss Silbengrenzzeichen sowie Akzentmarkierung(en) enthalten. Das gilt insbesondere für Leerzeichen. Da sie kein Bestandteil der SAMPA Notation sind, führen sie zu einer fehlerhaften Ausgabe. Der Laut "Stille" wird durch einen Punkt ./ gekennzeichnet.

```
| #SAMPa=h'A-lo:;#
```

Zusätzlich zur SAMPA-Umschrift kann für das Wort eine Wortart angegeben werden. Diese Angabe steht immer an letzter Stelle des Tags, zwischen dem zweiten Semikolon und der Endraute.

```
| #SAMPa=h'A-lo:;NOUN#
```

Eine andere Angabe, die zusätzlich zur Lautschrift gemacht werden kann, ist die orthografische Schreibweise des Wortes. Diese Angabe wird als Textmarkierung übergeben und wird beispielsweise in einem laufenden Text parallel zur Sprachausgabe angezeigt.

```
| #SAMPa=h'A-lo;Hallo;#
```

Ein vollständiges SAMPA-Speechtag enthält alle drei Angaben, Lautschrift, Orthografie und Wortart.

```
| #SAMPa=h'A-lo:;Hallo;NOUN#
```

Siehe auch

POS, ACC, Beschreibung Intonation

5.5 LASTACC

| #LASTACC#

Manchmal werden nicht alle Wörter in einem Satz akzentuiert. Dabei kommt es oft vor, dass eine Gruppe von Wörtern am Satzende keine Akzente mehr bekommt, da das wichtigste Wort des Satzes relativ früh im Satz steht, wie zum Beispiel in:

| "Wer ißt gerne süße Bonbons?"
| "Peter ißt gerne süße Bonbons."

In der Antwort auf die Frage sind alle Wörter nach "Peter" deakzentuiert. Das kann man mit "LASTACC" erwirken:

| "Wer ißt gerne süße Bonbons?"
| "Peter #LASTACC# ißt gerne süße Bonbons."

Alle Wörter nach dem Speechtag bis zum Ende der Intonationsphrase sind deakzentuiert.

Werte

LASTACC nimmt keine Parameter.

Beispiele

| #LASTACC#

Siehe auch

ACC, NOACC, EMPH

5.6 NOACC

| #NOACC#

Das folgende Wort ist nicht akzentuiert. Eventuell gesetzte Akzente werden gelöscht.

Werte

NOACC nimmt keine Parameter.

Beispiele

| #NOACC#

Siehe auch

ACC, LASTACC, EMPH

5.7 NOIP

| #NOIP#

Unterdrückt das Setzen einer IP Grenze an dieser Position.

Werte

NOIP nimmt keine Parameter.

Beispiele

| #NOIP#

Siehe auch

NOAP, NORP

5.8 NOAP

| #NOAP#

Unterdrückt das Setzen einer AP-Grenze nach dem vorhergehenden Wort. Die Akzentphrase (AP) ist die kleinste intonatorisch relevante Einheit.

Werte

NOAP nimmt keine Parameter.

Beispiele

| #NOAP#

Siehe auch

NOIP, NORP

5.9 NORP

| #NORP#

Unterdrückt das Setzen einer RP-Grenze nach dem vorhergehenden Wort. Dadurch wird die Bildung einer rhythmischen Phrase, der kleinsten satzinternen Einheit, unterdrückt, und die rhythmische Struktur des Satzes verändert.

Werte

NORP nimmt keine Parameter.

Beispiele

| #NORP#

Siehe auch

NOIP, NOAP

6 Sprechmelodie

6.1 Einführung

Die Sprechmelodie oder Intonation ist ein ziemlich komplexes Thema. Mit Logox 4 können Sie den Melodieverlauf eines Satzes selbst beeinflussen. Dazu wird hier ein kleiner Einblick in die zugrundeliegenden Konzepte der Berechnung und Umsetzung der Sprechmelodie gegeben.

Grundlegendes

Um die Sprechmelodie zu erzeugen, werden im Satz an bestimmten Stellen hohe oder tiefe Zielpunkte gesetzt. Die Zielpunkte werden mit L (low - tief) und H (high - hoch) bezeichnet. Ausgehend von einem Startwert werden diese Zielpunkte miteinander verbunden. So entsteht der Melodieverlauf oder die Intonationskontur. Die Zielpunkte werden innerhalb eines Satzes an zwei Positionen gesetzt:

- In oder um akzentuierte Silben von hervorgehobenen Wörtern
- An Grenzen von intonatorischen Einheiten (Phrasengrenzen)

Akzente

Die wichtigsten Wörter eines Satzes werden akzentuiert. In der am stärksten betonten Silbe eines wichtigen Wortes wird ein Zielpunkt erreicht. Dieser Zielpunkt wird mit einem Stern markiert. H* setzt also einen hohen Zielpunkt in der hauptbetonten Silbe eines akzentuierten Wortes. In einem Satz können mehrere Wörter akzentuiert sein. Ein Beispiel:

| Sie hat eine Zeitung gelesen.

"Zeitung" ist das wichtigste Wort dieses Satzes und wird akzentuiert. Die hauptbetonte Silbe ist "Zei". In dieser Silbe wird ein Zielpunkt gesetzt. Bei H* ist es ein hoher Zielpunkt. Bei L* ein tiefer. Zielpunkte können aber auch um akzentuierte Silben herum auftreten. D.h. auch in der vorherigen oder der folgende Silbe kann ein Zielpunkt gesetzt werden. Diese Zielpunkte werden mit einem "+" an den Ton der Akzentsilbe angehängt (z.B.: "L+H*" oder "L*+H"). Im ersten Fall (L+H*) wird ein tiefer Zielpunkt in der Silbe vor der Akzentsilbe gesetzt. Im Beispiel oben wäre das in der Silbe "ne" von "eine". Im zweiten Fall (L*+H) wird in der Silbe nach der Akzentsilbe ein hoher Zielpunkt gesetzt (im obigen Beispiel wäre das "tung" aus dem Wort "Zeitung").

Daraus ergibt sich das folgende Basisinventar für Akzenttöne:

| | |
|-------|---|
| H* | Hoher Zielpunkt in der Akzentsilbe |
| L* | Tiefer Zielpunkt in der Akzentsilbe |
| L*+H | Tiefer Zielpunkt in der Akzentsilbe Hoher Zielpunkt nach der Akzentsilbe |
| L+H* | Tiefer Zielpunkt vor der Akzentsilbe Hoher Zielpunkt in der Akzentsilbe |
| H+L* | Hoher Zielpunkt vor der Akzentsilbe Tiefer Zielpunkt in der Akzentsilbe |
| H*+L | Hoher Zielpunkt in der Akzentsilbe Tiefer Zielpunkt nach der Akzentsilbe |
| H+!H* | Hoher Zielpunkt vor der Akzentsilbe |

Mittelhoher Zielpunkt in der Akzentsilbe (vgl. [Downstep](#))

Wenn nicht genügend Silben zur Verfügung stehen, um die Zielpunkte in benachbarten Silben zu setzen, werden sie in die akzentuierte Silbe verschoben. Ein Beispiel:

| Ist das ein #ACC=L*+H# Test?

Der hohe Zielpunkt, der nach Test ausgeführt werden soll, kann nicht auf die nächste Silbe fallen. Daher wird er in der Silbe "Test" nach dem L Zielpunkt gesetzt.

Tonverläufe an Grenzen von intonatorischen Einheiten

Ein Satz ist eine intonatorische Einheit. Aber - insbesondere wenn er lang ist - wird er intern weiter gegliedert. In Logox 4 werden 3 Gliederungsebenen innerhalb eines Satzes unterschieden:

- IP - Intonationsphrase
- AP - Akzentphrase
- RP - Rhythmische Phrase

Diese Gliederungsebenen sind hierarchisch geordnet. An einer Satzgrenze ist automatisch auch eine IP-Grenze, eine AP-Grenze und eine RP-Grenze. Analog erzeugt eine IP-Grenze eine AP- und eine RP-Grenze, und eine AP-Grenze eine RP-Grenze.

Die Unterteilung eines Satzes (einer IP, einer AP) in rhythmische Phrasen wirkt sich nicht auf die Sprechmelodie aus. Sie dient der sprechrhythmischen Gliederung.

Die kleinste intonatorische relevante Einheit ist die Akzentphrase. Sie wird z.B. nach längeren Sinneinheiten gesetzt (z.B. "die Mutter des Kindes", "Bundesverteidigungsminister Rudolf Scharping"). Am Ende einer Akzentphrase wird ein Zielpunkt gesetzt. Diese Zielpunkte werden durch ein Minus "-" markiert. Daraus ergibt sich das folgende Basisinventar für AP-Grenztöne:

H- Mittelhoher Zielpunkt am Ende der Akzentphrase.

L- Tiefer Zielpunkt am Ende der Akzentphrase.

Eine Intonationsphrase entspricht ungefähr einem Nebensatz.

| Ich habe gesehen, dass er mit dem Wagen gekommen ist.

Dieser Satz enthält (bei normaler Sprechweise) zwei Intonationsphrasen.

Da eine Intonationsphrase immer mit einer Akzentphrase zusammenfällt, wird an dieser Stelle auch immer ein AP-Grenztone gesetzt. Die beiden Grenztöne zusammen können daher auch komplexe Tonbewegungen erzeugen. Der Grenztone der Intonationsphrase ist durch ein Prozentzeichen "%" gekennzeichnet. Die beiden Grenztöne werden nicht getrennt geschrieben, sondern sie werden in einem Symbol zusammengefasst. In einem zusammengesetzten IP-Grenztone setzt L% keinen eigenen Zielpunkt. Daher wirkt sich L% auch nicht auf die Bewegungsrichtung der Sprechmelodie aus. Daher wird "L%" durch "%" ersetzt. Daraus ergibt sich folgendes Basisinventar für IP-Grenztöne:

L-%, L-L% Tiefer Zielpunkt am Ende der IP.
Wie L- nur etwas tiefer

L-H% Tiefer Zielpunkt nach der letzten Akzentsilbe
Mittelhoher Anstieg am Ende der IP

H-%, H-L% Mittelhoher Zielpunkt am Ende der IP (wie H-)

H-H% Mittelhoher Zielpunkt nach der letzten Akzentsilbe
 Hoher Anstieg am Ende der IP

Auch hier gilt, dass Zielpunkte, für die nicht genügend Silben zur Verfügung stehen, zusammengeschoben werden und im Extremfall alle auf einer einzigen Silbe realisiert werden.

Akzente und Stimmumfang

Die Tonhöhe der hohen Zielpunkte von Akzenten und Grenztönen wird manchmal von vorhergehenden Akzenten beeinflusst. Ein Akzentton oder Grenzton (mit Einschränkungen bei H%) kann den Stimmumfang für nachfolgende Zielpunkte stufenweise verringern (Downstep) oder erweitern (Upstep). Während eine Verringerung des Stimmumfangs häufig zum Ende eines Satzes vorkommt, tritt die Erweiterung nur in expressiver Sprechweise auf. Hohen Zielpunkten, die den Stimmumfang verringern, wird ein Ausrufezeichen "!" vorangestellt. Hohe Zielpunkte, die den Stimmumfang erweitern, werden durch ein "^" markiert. Abgesehen vom Stimmumfang ändert sich aber nichts an der Charakteristik des zugrundeliegenden Akzent- oder Grenztons. Mit folgender Ausnahme: H+!H* ist ein Akzentton mit eigener Charakteristik. Vor der akzentuierten Silbe wird ein hoher Zielpunkt erreicht. Dann findet die Reduktion des Stimmumfangs statt, und der neue hohe Zielpunkt ist etwas tiefer als vorher, aber nicht tief. Alle folgenden hohen Zielpunkte sind im Stimmumfang reduziert. Die Reduktion bzw. Expansion des Stimmumfangs kann mehrmals erfolgen. Durch diese Modifikationen erweitert sich das Inventar auf folgende Werte.

| Basiston | Modifikationen | Zielpunkte und deren Position |
|--------------------|----------------|--|
| Akzentttöne | | |
| H* | !H*, ^H* | Hoher Zielpunkt in der Akzentsilbe |
| L* | | Tiefer Zielpunkt in der Akzentsilbe |
| L*+H | L*+!H, L*+^H | Tiefer Zielpunkt in der Akzentsilbe Hoher Zielpunkt nach der Akzentsilbe |
| L+H* | L+!H*, L+^H* | Tiefer Zielpunkt vor der Akzentsilbe Hoher Zielpunkt in der Akzentsilbe |
| H+L* | !H+L*, ^H+L* | Hoher Zielpunkt vor der Akzentsilbe Tiefer Zielpunkt in der Akzentsilbe |
| H*+L | !H*+L, ^H*+L | Hoher Zielpunkt in der Akzentsilbe Tiefer Zielpunkt nach der Akzentsilbe |
| H+!H* | !H+!H*, ^H+!H* | Hoher Zielpunkt vor der Akzentsilbe Mittelhoher Zielpunkt in der Akzentsilbe (vgl. Downstep) |
| Grenztöne | | |
| H- | !H-, ^H- | Mittelhoher Zielpunkt am Ende der Akzentphrase. |
| L- | | Tiefer Zielpunkt am Ende der Akzentphrase. |
| L-% | | Tiefer Zielpunkt am Ende der IP. Wie L- nur etwas tiefer |
| L-H% | L-!H% | Tiefer Zielpunkt nach der letzten Akzentsilbe Mittelhoher Anstieg am Ende der IP |
| H-% | !H-%, ^H-% | Mittelhoher Zielpunkt am Ende der IP (wie H-) |

H-^H%

!H-^H%, ^H-^H%

Mittelhoher Zielpunkt nach der letzten Akzentsilbe
Hoher Anstieg am Ende der IP

Abschnitt 6

I Tools

Im Lieferumfang von WebSpeech SDK sind auch einige Tools zum Erzeugen von HTML-Code , Standardsprechfunktionen in JavaScript und zur Einbindung von Speechtags enthalten.

Die WS-Objektbox erstellt den HTML-Code für ein WebSpeech Objekt mit den angegebenen Eigenschaften.

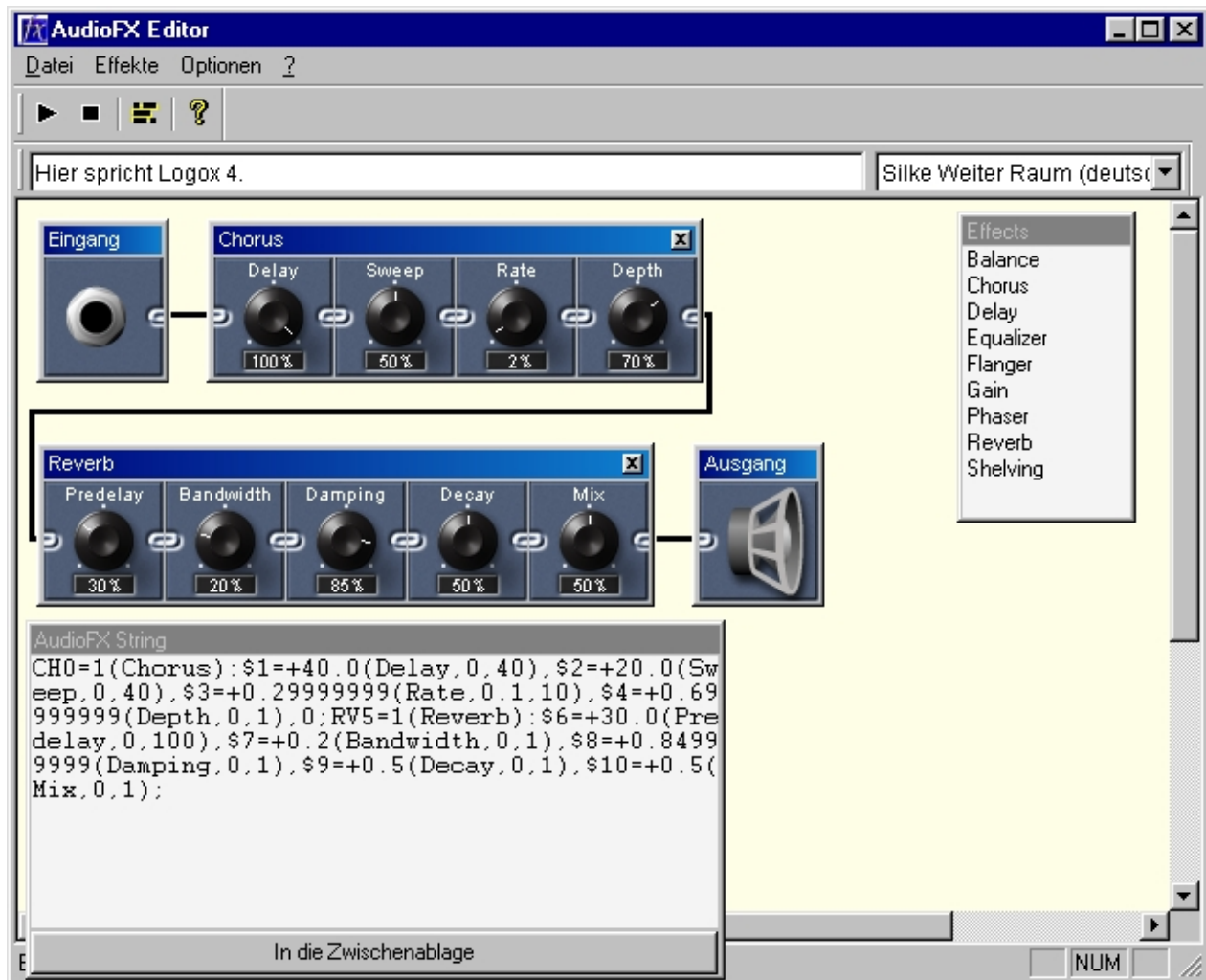
Die JS-Box erzeugt auf Knopfdruck den Code einer sprechenden HTML-Seite mit den angewählten WebSpeech Funktionen in JavaScript. Man kann eine WebSpeech-Abfrage, Standardsprechfunktionen, sowie Abfragen von Status und Events auswählen. Der jeweilige JavaScript Code wird getrennt für die Bereiche HEAD und BODY oder als komplette HTML-Seite ausgegeben. Der resultierende Code kann entweder in bestehende Seiten kopiert werden oder die komplette HTML-Seite kann als Ausgangsbasis für weitere Dateien dienen.

Der Audioeffekt-Editor erstellt die komplexen Parameter, die im Speechtag AFX verwendet werden. Mit diesem Steuerzeichen, können im Text die Parameter der integrierten Audioeffekte bestimmt werden.

2 AudioFXEditor

Nach dem Öffnen von audiofxeditor.exe sehen Sie 3 Fenster:

- Das Hauptfenster mit einem Eingangs- und einem Ausgangssymbol, die miteinander verbunden sind.
- Das Verzeichnis "Effects" mit den vorhandenen Audioeffekten
- Ein (noch) leeres Fenster, das den Steuercode für einen Effekt oder eine Kombination von Effekten enthält. Per Klick auf den Button wird der Steuercode in die Zwischenablage kopiert.



2.1 Effekte erstellen und anordnen

Um einen Effekt zu erstellen, ziehen Sie ihn einfach aus der Liste "Effects" in das Hauptfenster. Sie können auch mehrere Effekte hintereinander schalten. Ziehen Sie dazu einfach weitere Effekte (oder den gleichen Typ nochmal) in das Hauptfenster. Die Effekte werden automatisch in der Reihenfolge der Anordnung miteinander verknüpft. Die Reihenfolge können sie auch manuell ändern, indem sie die Verbindungslinien löschen und neu verbinden. Zum Löschen ziehen Sie einfach die gewünschte Verbindung ins Leere. Meist reicht auch ein kurzer Klick. Die neue Verbindung erstellen Sie, indem sie die Maus mit gedrückter linker Maustaste von der grauen Output-Fläche des ersten Moduls zur grauen Input-Fläche des zweiten Moduls bewegen und dann loslassen (Drag&Drop). Die Effekte werden automatisch angeordnet, wenn Sie mit der rechten Maustaste ins Hauptfenster klicken und "Symbole anordnen" wählen.

2.2 Effekte editieren

Bei gedrückter Maustaste können Sie an den Reglern der Effekte drehen. Die Funktion der einzelnen Parameter ist in den folgenden Abschnitten beschrieben. Die Auswirkungen können Sie hören, wenn Sie den Startbutton drücken. Den zu sprechenden Text können Sie in der Textzeile darunter festlegen. In der Liste rechts daneben können Sie einen der vorhandenen Sprecher auswählen.

2.3 Der resultierende AudioFX String

Der aus Ihren Einstellungen resultierende Parametersatz wird im Feld "AudioFX String" angezeigt. Durch einen Klick auf "In die Zwischenablage" wird der Steuerstring in die Zwischenablage kopiert. Von dort können Sie ihn in Ihren Text einfügen.

Beispiele

```
#AFX=DL70=1:$71=100.000000,$72=0.400000,$73=0.000000;#  
Hier spricht Logox 4 mit Echo. #AFX#
```

Hiermit wird ein Echoeffekt eingefügt, und am Ende des Textes wird der Effekt wieder ausgeschaltet.

3 Anhang

3.1 Tabelle der Farbnamen

In der folgenden Tabelle sind die für die Verwendung in HTML definierten Farbnamen mit ihren RGB-Werten angegeben.

| Farbname | R | G | B |
|----------------|-----|-----|-----|
| Aliceblue | 240 | 248 | 255 |
| Antiquewhite | 250 | 235 | 215 |
| Aqua | 0 | 255 | 255 |
| Aquamarine | 127 | 255 | 212 |
| Azure | 240 | 255 | 255 |
| Beige | 245 | 245 | 220 |
| Bisque | 255 | 228 | 196 |
| Black | 0 | 0 | 0 |
| blanchedalmond | 255 | 235 | 205 |
| Blue | 0 | 0 | 255 |
| Blueviolet | 138 | 43 | 226 |
| Brown | 165 | 42 | 42 |
| Burlywood | 222 | 184 | 135 |
| Cadetblue | 95 | 158 | 160 |
| Chartreuse | 127 | 255 | 0 |
| Chocolate | 210 | 105 | 30 |
| Coral | 255 | 127 | 80 |
| Cornflower | 100 | 149 | 237 |
| Cornsilk | 255 | 248 | 220 |
| Crimson | 220 | 20 | 60 |
| Cyan | 0 | 255 | 255 |
| Darkblue | 0 | 0 | 139 |
| Darkcyan | 0 | 139 | 139 |
| Darkgoldenrod | 184 | 134 | 11 |
| Darkgray | 169 | 169 | 169 |
| Darkgreen | 0 | 100 | 0 |
| Darkkhaki | 189 | 183 | 107 |
| Darkmagenta | 139 | 0 | 139 |
| Darkolivegreen | 85 | 107 | 47 |
| Darkorange | 255 | 140 | 0 |
| Darkorchid | 153 | 50 | 204 |
| Darkred | 139 | 0 | 0 |
| Darksalmon | 233 | 150 | 122 |
| Darkseagreen | 143 | 188 | 139 |
| Darkslateblue | 72 | 61 | 139 |
| Darkslategray | 47 | 79 | 79 |
| Darkturquoise | 0 | 206 | 209 |
| Darkviolet | 148 | 0 | 211 |
| Deeppink | 255 | 20 | 147 |
| deepskyblue | 0 | 191 | 255 |

| Farbname (fortgesetzt) | R | G | B |
|---------------------------|-----|-----|-----|
| Dimgray | 105 | 105 | 105 |
| dodgerblue | 30 | 144 | 255 |
| Firebrick | 178 | 34 | 34 |
| Floralwhite | 255 | 250 | 240 |
| Forestgreen | 34 | 139 | 34 |
| Fuchsia | 255 | 0 | 255 |
| Gainsboro | 220 | 220 | 220 |
| Ghostwhite | 248 | 248 | 255 |
| Gold | 255 | 215 | 0 |
| Goldenrod | 218 | 165 | 32 |
| Gray | 128 | 128 | 128 |
| Green | 0 | 128 | 0 |
| Greenyellow | 173 | 255 | 47 |
| Honeydew | 240 | 255 | 240 |
| Hotpink | 255 | 105 | 180 |
| Indianred | 205 | 92 | 92 |
| Indigo | 75 | 0 | 130 |
| Ivory | 255 | 255 | 240 |
| Khaki | 240 | 230 | 140 |
| Lavender | 230 | 230 | 250 |
| lavenderblush | 255 | 240 | 245 |
| Lawngreen | 124 | 252 | 0 |
| lemonchiffon | 255 | 250 | 205 |
| Lightblue | 173 | 216 | 230 |
| Lightcoral | 240 | 128 | 128 |
| Lightcyan | 224 | 255 | 255 |
| Lightgoldenrodyellow | 250 | 250 | 210 |
| Lightgreen | 144 | 238 | 144 |
| Lightgray | 211 | 211 | 211 |
| Lightpink | 255 | 182 | 193 |
| Lightsalmon | 255 | 160 | 122 |
| lightseagreen | 32 | 178 | 170 |
| Lightskyblue | 135 | 206 | 250 |
| lightslategray | 119 | 136 | 153 |
| lightsteelblue | 176 | 196 | 222 |
| Lightyellow | 255 | 255 | 224 |
| Lime | 0 | 255 | 0 |
| Limegreen | 50 | 205 | 50 |
| Linen | 250 | 240 | 230 |
| Magenta | 255 | 0 | 255 |
| Maroon | 128 | 0 | 0 |
| Mediumaquamarine | 102 | 205 | 170 |
| Mediumblue | 0 | 0 | 205 |
| mediumorchid | 186 | 85 | 211 |
| mediumpurple | 147 | 112 | 219 |
| mediumseagreen | 60 | 179 | 113 |
| mediumslateblue | 123 | 104 | 238 |

| Farbname (fortgesetzt) | R | G | B |
|---------------------------|-----|-----|-----|
| Mediumspringgreen | 0 | 250 | 154 |
| mediumturquoise | 72 | 209 | 204 |
| mediumvioletred | 199 | 21 | 133 |
| midnightblue | 25 | 25 | 112 |
| Mintcream | 245 | 255 | 250 |
| Mistyrose | 255 | 228 | 225 |
| Moccasin | 255 | 228 | 181 |
| Navajowhite | 255 | 222 | 173 |
| Navy | 0 | 0 | 128 |
| Oldlace | 253 | 245 | 230 |
| Olive | 128 | 128 | 0 |
| Olivedrab | 107 | 142 | 35 |
| Orange | 255 | 165 | 0 |
| Orangered | 255 | 69 | 0 |
| Orchid | 218 | 112 | 214 |
| palegoldenrod | 238 | 232 | 170 |
| Palegreen | 152 | 251 | 152 |
| paleturquoise | 175 | 238 | 238 |
| palevioletred | 219 | 112 | 147 |
| Papayawhip | 255 | 239 | 213 |
| Peachpuff | 255 | 218 | 185 |
| Peru | 205 | 133 | 63 |
| Pink | 255 | 192 | 203 |
| Plum | 221 | 160 | 221 |
| Powderblue | 176 | 224 | 230 |
| Purple | 128 | 0 | 128 |
| Red | 255 | 0 | 0 |
| Rosybrown | 188 | 143 | 143 |
| Royalblue | 65 | 105 | 225 |
| saddlebrown | 139 | 69 | 19 |
| Salmon | 250 | 128 | 114 |
| Sandybrown | 244 | 164 | 96 |
| Seagreen | 46 | 139 | 87 |
| Seashell | 255 | 245 | 238 |
| Sienna | 160 | 82 | 45 |
| Silver | 192 | 192 | 192 |
| Skyblue | 135 | 206 | 235 |
| Slateblue | 106 | 90 | 205 |
| Slategray | 112 | 128 | 144 |
| Snow | 255 | 250 | 250 |
| Springgreen | 0 | 255 | 127 |
| Steelblue | 70 | 130 | 180 |
| Tan | 210 | 180 | 140 |
| Teal | 0 | 128 | 128 |
| Thistle | 216 | 191 | 216 |

| Farbname (fortgesetzt) | R | G | B |
|---------------------------|-----|-----|-----|
| Tomato | 255 | 99 | 71 |
| Turquoise | 64 | 224 | 208 |
| Violet | 238 | 130 | 238 |
| Wheat | 245 | 222 | 179 |
| White | 255 | 255 | 255 |
| Whitesmoke | 245 | 245 | 245 |
| Yellow | 255 | 255 | 0 |
| Yellowgreen | 154 | 205 | 50 |

3.2 Index

| | | | |
|--------------------------------|--------|------------------------------|--------|
| AUTHKEY | 28 | OnStartSpeaking..... | 74 |
| AUTOSTART | 25, 29 | OnStopSpeaking..... | 75 |
| BACKGROUNDColor..... | 40 | OnWord..... | 81 |
| Benutzergruß..... | 108 | OPAQUE..... | 39 |
| CLASSID | 24 | Parameter | 27 |
| CONTROLPOSITION..... | 41 | PauseSpeaking..... | 50 |
| Download | 10 | PlugIn | 8 |
| Events | 72 | ResumeSpeaking..... | 51 |
| Funktionen..... | 72 | SetBackgroundColor..... | 64 |
| Geschlecht des Anwenders | 71 | SetControlPosition..... | 65 |
| GetInterfaceVersion..... | 68 | SetKey | 66 |
| GetUserGender | 71 | SetMouthAnimation..... | 56 |
| GetUserGreeting..... | 70 | SetMouthColor | 57 |
| GetUserName..... | 69 | SetOpaque..... | 63 |
| GetVersion | 67 | SetText..... | 45 |
| HEIGHT | 24 | SetTextAnimation..... | 59 |
| IMMEDIATE | 30 | SetTextColor | 60 |
| Installation..... | 11 | SetTextPosition | 61 |
| IsLoading | 55 | SetTextSize..... | 62 |
| IsMuting..... | 54 | Skriptsprachen..... | 42 |
| IsPause | 53 | SpeechFont..... | 85 |
| IsSpeaking..... | 52 | Speechtags | 85 |
| JavaScript..... | 42 | Sprachausgabe..... | 8, 83 |
| Lexika..... | 8 | Sprechweise | 85 |
| LGX-Dateien..... | 84 | StartSpeaking..... | 47 |
| LoadText..... | 46 | StartSpeakingImmediate | 48 |
| Markierung..... | 78 | StopSpeaking..... | 49 |
| Methoden..... | 44 | Stummschaltung..... | 54 |
| Mikrosegmente..... | 8 | TEXT | 25, 31 |
| Mikrosegmentsynthese | 8 | TEXTANIMATION..... | 35 |
| MOUTHANIMATION | 25, 33 | TEXTCOLOR..... | 36 |
| MOUTHCOLOR | 34 | Textdateien | 83 |
| OBJECT..... | 24 | TEXTPOSITION | 37 |
| OnClick..... | 79 | TEXTSIZE | 38 |
| OnLoad | 82 | TEXTURE | 57 |
| OnMark..... | 78 | TRANSPARENT..... | 57 |
| OnMouthOver..... | 80 | URL | 25, 32 |
| OnPauseSpeaking..... | 76 | WIDTH..... | 24 |
| OnResumeSpeaking..... | 77 | | |