

moccasin	#FFE4B5
navajowhite	#FFDEAD
navy	#000080
oldlace	#FDF5E6
olive	#808000
olivedrab	#6B8E23
orange	#FFA500
orangered	#FF4500
orchid	#DA70D6
palegoldenrod	#EEE8AA
palegreen	#98FB98
paleturquoise	#AFEEEE
palevioletred	#DB7093
papayawhip	#FFEFD5
peachpuff	#FFDAB9
peru	#CD853F
pink	#FFC0CB
plum	#DDA0DD
powderblue	#B0E0E6
purple	#800080
red	#FF0000
rosybrown	#BC8F8F
royalblue	#4169E1
saddlebrown	#8B4513
salmon	#FA8072
sandybrown	#F4A460
seagreen	#2E8B57
seashell	#FFF5EE
sienna	#A0522D
silver	#C0C0C0
skyblue	#87CEEB
slateblue	#6A5ACD
slategray	#708090
snow	#FFFAFA
springgreen	#00FF7F
steelblue	#4682B4
tan	#D2B48C
teal	#008080
thistle	#D8BFD8
tomato	#FF6347
turquoise	#40E0D0
violet	#EE82EE
wheat	#F5DEB3
white	#FFFFFF
whitesmoke	#F5F5F5
yellow	#FFFF00
yellowgreen	#9ACD32

5. Plugins des Netscape und ActiveX-Controls des Internet Explorers für die Integration von Browsererweiterungen

Aufgrund der Desingunterschiede zwischen diesen Browsern muss für die Verwendung eines Plugins wie Shockwave oder Webspeech unterschiedlich programmiert werden.

Nur Netscape unterstützt das Objekt Plugin. Der Internet Explorer beruft sich auf ActiveX als Betriebssystemkomponente. Bezweckt wird allerdings dasselbe: Die Integration fremder Tools in den jeweiligen Browser.

5.1. *Plugins des Netscape für Browsererweiterungen durch Fremdanbieter*

Netscape unterstützt den <EMBED>-Tag (siehe HTML-Beschreibung).

5.2. *ActiveX des Internet Explorer für Browsererweiterungen (z.T. Fremdanbieter)*

Beispiel für prüfen auf ActiveX beim IE:

```
var ActiveXAktiv = false;

var NavigatorObjekt = window.navigator;           // Zeiger

var BrowserArt      = NavigatorObjekt.userAgent;   // String
var IEErkannt       = (BrowserArt.indexOf('IE') > -1); // true, so IE erkannt

var Plattform       = NavigatorObjekt.platform;    // String
var Win32Erkannt    = (Plattform === "Win32");     // true, so Win32-Bit erkannt
```



```
ActiveXAktiv    = (IEerkannt && Win32Erkannt);    // true, so IE und Win32-Bit erkannt,
                                                         // also ActiveX möglich
```

5.2.1. Datenbank im Internet Explorer ab 4.x

ActiveX muss aktiv sein !

Die Textdatenbank wird komplett in den Cache des Users übertragen und ist somit nicht geschützt !

Erst wenn die gesamte Datenbank im Cache geladen ist, wird sie angezeigt !

Die Domain der Datenbank-Datei und des HTML-Dokumentes, das die Datenbank verwendet, müssen identisch sein.

Datenbankverwaltung erfolgt mit dem TDC (Tabular Data Container) als ActiveX-Control

clsid:333C7BC4-460F-11D0-BC04-0080C7055A83

Hinweis: 0 ist Null und nicht Zeichen O

5.2.1.1. Aufbau der Datenbank

Datenbank im Textformat

Datenbank-Datei: Textdatei
satzweise

Datenbank-Datei-Satz: CSV-Format (Comma Separated Values)

endet standardgemäß mit Zeilenumbruch (üblich bei Textdateien \$0D\$0A also newline)
(Satztrenner newline ist abänderbar)

Folge von Feldwerten, die standardgemäß mit Komma zu trennen sind
(Feldtrenner Komma ist abänderbar)

Feldwert-Format: wert[:typ] :typ ist optional

Feldwert-Typen :typ sind zu kodieren im ersten Satz der Textdatei (Daten ab 2. Satz), wenn
es sich nicht um String-Typen handelt

Bsp.: Sind alle Daten im String-Typ, so kann 1. Satz bereits
Daten enthalten

Empfehlung: Wenn Datenbank sortiert werden soll, so immer
1. Satz mit Feldbezeichnern erzeugen

String ist Standard
kein " " oder ' ' kodieren (Ausnahme: siehe unten Feldtrenner
und Blank)

Datexxx mit xxx als Buchstabenfolge von D, M und Y in beliebiger
Reihenfolge

Boolean Yes oder No oder True oder False oder 0 oder not 0
wobei Yes identisch zu True
zu Wert > 0
No identisch zu False
zu Wert gleich 0

Int ganze Zahl

Float Gleitzahl

mit Dezimalpunkt laut Windowseinstellung

Deutsch: Punkt für Tausendertrenner

Komma für Dezimalkomma

Englisch (Standard bei JScript):

Komma für Tausendertrenner

Punkt für Dezimalkomma

Feldwert darf nicht enthalten

Feldtrenner und/oder Blank, es sei denn, der Wert des Feldes wird in " "
eingeschlossen und damit Blank bzw. Feldtrenner in ihrer Funktion
entwertet (Entwertungszeichen " ist abänderbar)
deutsche Umlaute und Sonderzeichen

Beispiel für Daten nur vom String-Typ:

ab 1. Satz Guericke,"Otto, von",0815/1234
Mueller,Anton,0815/6789

Beispiel für Daten nur vom String-Typ:

Satz 1 Feldbezeichner
vorname,name,telefon
ab 2. Satz Guericke,"Otto, von",0815/1234
Mueller,Anton,0815/6789

Beispiel für Daten mit Nicht-String-Typ:

1. Satz muss kann Feldbezeichnerliste sein mit Typangabe (außer bei String-Typ)
NahrungsmittelArt,Preis:Float,Kaufdatum:DateYMD,Menge:Int

ab 2. Satz Brot,1.57,97/5/12,30
Kaese,3.52,96/2/2,5
"Alter Wein",183.99,1905-1-1,1



5.2.1.2. HTML-Einbindung**5.2.1.2.1. Objekt-Deklaration**

<BODY>

```

<OBJECT ID="ID_Datenbank" CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83">
    <!-- // 0 ist Null und nicht Zeichen O //-->

    <PARAM NAME="DataURL" VALUE="adress.txt">
        <!-- Url und Name der Textdatei, also
              physischer Datenbankname wie z.B.
              file:// oder http:// oder data/elements.txt
              Achtung: Die Domain von Datenbank undHTML-Seite
              müssen identischsein !
              im Beispiel: adress.txt im gleichen Verzeichnis
        //-->

    <PARAM NAME="UseHeader" VALUE="True" oder "False" >
        <!-- True, so 1. Satz nicht anzeigen
              verwenden, wenn 1. Satz mit Feldbezeichner
              False, so 1. Satz anzeigen
              IE prüft nicht, ob 1. Satz mit Feldbezeichner,
              da dem IE die Dateninhalte egal sind.
        //-->

    // nachfolgender PARAM ist optional
    <PARAM NAME="FieldDelim" VALUE="&#09;">
        <!-- anstelle des Komma als Feldtrenner nun TAB verwenden
              ansonsten einfach das Zeichen kodieren z.B. ";"
        //-->

    // nachfolgender PARAM ist optional
    <PARAM NAME="Filter" VALUE="(logische_verknuepfung_von_feldbezeichnern)">
        <!-- Bsp. "(PhoneNum <> &quot;&quot;)"
              Achtung: ES MUSS geklammert werden.
              logische Verknüpfung wie bei JScript
        //-->

</OBJECT>
</BODY>

```

5.2.1.2.2. Datenfeld-Deklaration

im HTML-Tag, das das Datum verwenden möchte: Nicht alle Tags werden vom TDC unterstützt !
 per Anker auf die Datenbank

Beispiel für Tabelle:

Prinzip: jedes Feld wird als HTML-Tabellenspalte dargestellt:
 pro Spaltenzeile der Feldinhalt aus dem jeweiligen Satz
 Anzahl der Zeilen=Anzahl der Sätze (abgesehen von Feldbezeichnersatz also Satz 1)

Achtung: Nicht alle Tabellenattribute werden von TDC unterstützt !

unterstützt werden: DIV
 SPAN
 IMG

adress.txt:

1. Satz **vorname;name;telefon**
 ab 2. Satz Guericke;"Otto, von";0815
 Willmann;Theo;1234
 Xantippe;Isa;5678
 Arktin;Richard;1055

<BODY>

```

<OBJECT ID="ID_Datenbank"
  CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
>
  <PARAM NAME="DataURL" VALUE="adress.txt">
  <PARAM NAME="UseHeader" VALUE="True">
  <PARAM NAME="FieldDelim" VALUE=";">
</OBJECT>

<TABLE DATASRC=#ID_Datenbank>
<!-- 1. Satz als Kopf //-->
<THEAD>
<TR>

```



```

        <TH>Vorname</TH>
        <TH>Name</TH>
        <TH>Telefon</TH>

</TR>
</TR>
</THEAD>
<!-- genau 1 Datenzeile kodieren, die als Vorlage für jeden Satz der adress.txt verwendet wird:
        Jeder Satz erhält automatisch in der Tabelle eine eigene Zeile
        adress.txt wird angezeigt in der Satzfolge laut der Datei

//-->
<TBODY>
<TR>
        <TD><DIV DATAFLD="vorname"></DIV></TD>
        <TD><DIV DATAFLD="name"></DIV></TD>
        <TD><DIV DATAFLD="telefon"></DIV></TD>

</TR>
</TBODY>
</TABLE>

</BODY>

```

5.2.1.3. Operationen mit der Datenbank

5.2.1.3.1. Datenbank-Objekt

Syntax:

ID_Datenbank.eigenschaft
ID_Datenbank.methode()

Eigenschaften:

.AppendData	Boolean true , so neue Daten anhängen false, so neue Daten ersetzen alte Daten
.CaseSensitive	Boolean true, so Unterscheidung von Gross und Klein false, so keine Unterscheidung von Gross und Klein
.CharSet	Wert für Zeichensatz 20106 DIN 66003 IA5 German 1250 windows-1250 Central European Alphabet 1252 iso-8859-1 und windows-1252, Western Alphabet 50000 x-user-defined
.EscapeChar	Entwertungszeichen nicht in " " bzw. ' ' kodieren Standard ist Leerkette Bsp. für Zeichen innerhalb einer Kette, das identisch mit dem Ketten-Kennzeichnungs.Zeichen ID_Datenbank.EscapeChar=" ermöglicht die Kodierung von "Das ist \"echt\" cool!"
.FieldDelim	Feldtrennerzeichen Standard ist ",", Bsp.: "	" für TAB überschreibt <PARAM NAME=FieldDelim ... >
.Filter	für Satzselektion per Filter Wert ist "logische_verknuepfung_von_feldbezeichnen" analog zu <PARAM NAME="Filter" ...> überschreibt <PARAM NAME="Filter" ...> Verwendung von Wildcard * möglich z.B. vorname = O* Bsp: ID_Datenbank.Filter=(('vorname = Otto') & ('name =Test'));
.Language	HTML-Code-Sprache Bsp: "eng-us" (Standard)
.recordset	Zeiger auf Objekt der Satzselektion
.RowDelim	Satztrennzeichen Standard ist "newline"
.Sort	"folge_von_feldbezeichnen_mit_semikolontrennung" Bsp: ID_Datenbank.Sort="-vorname; +name" -vorname für absteigende Sortierung +name für aufsteigende Sortierung Sortierung von Text: lexikographisch, also im Alphabet mit Unterscheidung Gross und Klein



numerischen Werten: bei aufsteigend: von minus zu plus
 bei absteigend: von plus zu minus
 Datumswerten in der Form t/m/jjjj
 wenn t und m identisch sind dann erst nach Jahren
 logischen Werten bei aufsteigend: erst alle False, dann alle True
 bei absteigend; erst alle True, dann alle False

Abbruch der Sortierung sobald ein leeres Feld gefunden wird

Kodierung zur Sortierung:

Falls die Datenbank keine Feldbezeichner im 1. Satz besitzt, so müssen frei erfundenen Feld-Bezeichner als Platzhalter kodiert werden.

Hinweis: Es kann nicht folgender Fehler wie in MS-Excel passieren:

Markierung einer Spalte anstelle des gesamten Excel-Blattes
 dann Daten sortieren ---> alle Sätze sind **nur in der Spalte**
 sortiert und damit **inhaltlich komplett verändert**

.TextQualifier Zeichen der Stringkodierung für Kodierung von Blank bzw. Feldtrenner als Daten
 nicht in " " bzw. ' ' kodieren
 Standard ist "

Methoden:

.recordset() Referenz auf Wert eines Feldes des aktuellen Satzes liefern
 Syntax:
 [var ZeigerAufDatenFeldWert =] **ID_Datenbank**.recordset(Kette);

Kette String
 Feldbezeichner laut 1. Datensatz der Datenbank

.Reset(); Datenbank aktualisieren (auch in der Ansicht)
 z.B. nach Selektionseinstellungen
 Filtereinstellungen

5.2.1.3.2. Objekt der Satzselektion (recordset)

Recordset ist das Objekt der Satzselektion per Satzzeiger.

Syntax:

ID_Datenbank.recordset.eigenschaft

ID_Datenbank.recordset.methode()

Eigenschaften:

.AbsolutePosition liefert aktuelle Satzposition
 Integer, ab 1

.BOF Boolean
 true wenn Satzzeiger vor dem ersten Satz steht
 false wenn Satzzeiger nicht vor dem ersten Satz steht
 Anwendung in Schleifen

.EOF Boolean
 true, wenn Zeiger hinter dem letzten Satz steht
 false, wenn Zeiger nicht hinter dem letzten Satz steht
 Anwendung in Schleifen

RecordCount Gesamtanzahl der Sätze in der Datenbank
 beachte <PARAM NAME="UseHeader" ...>

Methoden:

.MoveFirst(); ersten Satz der Datenbank einstellen (Satzzeiger auf ersten Satz setzen)
 .MoveNext(); nächsten Satz einstellen und damit Satzzeiger verändern
 .MovePrevious(); vorhergehenden Satz einstellen und damit Satzzeiger verändern

5.2.1.4. Beispiele

5.2.1.4.1. Sortierung

adress.txt:
 1. Satz **vorname;name;telefon**
 ab 2. Satz Guericke;"Otto, von";0815
 Willmann;Theo;1234
 Xantippe;Isa;5678
 Arktin;Richard;1055
 <HTML>
 <HEAD>
 <SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
 <!--
 var SortierRichtung = true;

function Sortieren(FeldBezeichner)



```

    {
        // Sortierrichtung festlegen
        var Kette = "-"; // Annahme
        if (SortierRichtung) {Kette = "+";}

        // nächste Sortierung umgekehrt
        SortierRichtung = !SortierRichtung;

        // sortieren
        ID_Datenbank.Sort= Kette + FeldBezeichner;

        // und das Ergebnis anzeigen
        ID_Datenbank.Reset();
    }
// -->
</SCRIPT>
</HEAD>

<BODY>
<OBJECT ID="ID_Datenbank"
CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
>
    <PARAM NAME ="DataURL" VALUE="adress.txt">
    <PARAM NAME ="UseHeader" VALUE="True">
    <PARAM NAME ="FieldDelim" VALUE=";">
</OBJECT>

<TABLE DATASRC=#ID_Datenbank>
<THEAD>
<TR>
    <TH><A HREF="JavaScript:Sortieren('vorname')">Vorname </A></TH>
    <TH><A HREF="JavaScript:Sortieren('name')">Name</A></TH>
    <TH><A HREF="JavaScript:Sortieren('telefon')">Telefon</A></TH>
</TR>
</TR>
</THEAD>
<TBODY>
<TR>
    <TD><DIV DATAFLD="vorname"></DIV></TD>
    <TD><DIV DATAFLD="name"></DIV></TD>
    <TD><DIV DATAFLD="telefon"></DIV></TD>
</TR>
</TBODY>
</TABLE>
</BODY>
</HTML>

```

5.2.1.4.2. Blättern in Datenbank

adress.txt:

```

1. Satz   vorname;name;telefon
ab 2. Satz Guericke;"Otto, von";0815
          Willmann;Theo;1234
          Xantippe;Isa;5678
          Arktin;Richard;1055

```

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
<!--
    var SortierRichtung = true;

    function Sortieren(FeldBezeichner)
    {
        // Sortierrichtung festlegen
        var Kette = "-"; // Annahme
        if (SortierRichtung) {Kette = "+";}

        // nächste Sortierung umgekehrt
        SortierRichtung = !SortierRichtung;

        // sortieren
        ID_Datenbank.Sort= Kette + FeldBezeichner;

        // und das Ergebnis anzeigen
        ID_Datenbank.Reset();
    }

```



```

    }

    function vorwaerts(AnzahlSaetze)
    {
        // nächste Seite anzeigen
        document.all.ID_Tabelle.nextPage();

        // und Satzzeiger korrigieren
        for (var i = 0; i < AnzahlSaetze; i++)
        {
            if (ID_Datenbank.recordset.AbsolutePosition != ID_Datenbank.recordset.RecordCount)
            {ID_Datenbank.recordset.MoveNext();}

        }

        if (ID_Datenbank.recordset.AbsolutePosition == ID_Datenbank.recordset.RecordCount)
        {alert("Letzter Datensatz erreicht!");}
    }

    function rueckwaerts(AnzahlSaetze)
    {
        // vorhergehende Seite anzeigen
        document.all.ID_Tabelle.previousPage();

        // und Satzzeiger korrigieren
        for (var i = 0; i < AnzahlSaetze; i++)
        {
            if (ID_Datenbank.recordset.AbsolutePosition > 1)
            {ID_Datenbank.recordset.MovePrevious();}

        }

        if (ID_Datenbank.recordset.AbsolutePosition == 1)
        {alert("Erster Datensatz erreicht!");}
    }

// -->
</SCRIPT>
</HEAD>
<BODY>
<OBJECT ID="ID_Datenbank"
    CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
>
    <PARAM NAME="DataURL" VALUE="adress.txt">
    <PARAM NAME="UseHeader" VALUE="True">
    <PARAM NAME="FieldDelim" VALUE=";">
</OBJECT>

<TABLE ID="ID_Tabelle"
    DATASRC=#ID_Datenbank
    DATAPAGESIZE=1
>
<THEAD>
<TR>
    <TH><A HREF="JavaScript:Sortieren('vorname')">Vorname </A></TH>
    <TH><A HREF="JavaScript:Sortieren('name')">Name</A></TH>
    <TH><A HREF="JavaScript:Sortieren('telefon')">Telefon</A></TH>
</TR>
</THEAD>
<TBODY>
<TR>
    <TD><DIV DATAFLD="vorname"></DIV></TD>
    <TD><DIV DATAFLD="name"></DIV></TD>
    <TD><DIV DATAFLD="telefon"></DIV></TD>
</TR>
</TBODY>
</TABLE>
<BR>
<INPUT
    TYPE="button"
    VALUE="Zurueck"
    onclick=rueckwaerts(1)"
>
<!-- Anzahl der Sätze laut TABLE DATAPAGESIZE=1 -->

<INPUT
    TYPE="button"
    VALUE="Vorwaerts"
    onclick=vorwaerts(1)"
>
<!-- Anzahl der Sätze laut TABLE DATAPAGESIZE=1 -->

```



```
</BODY>
</HTML>
```

5.2.1.4.3. Satzselektion mit Filter

1. Satz **vorname,name,telefon**
2. Satz Guericke,"Otto, von",0815/1234

```
function filtern()
{
    ID_Datenbank.Filter='vorname=Otto';
    ID_Datenbank.Reset();
    return(false);
}

<INPUT TYPE="button"
VALUE="Filtern nach Vornamen Otto "
onclick="filtern();"
>
```

5.2.2. Direct Animation im Internet Explorer (Übersicht DA als DirectX-Komponente)

Direct Animation ist das Pendant z.B. zu Flash von Adobe (ursprünglich von Macromedia).

Direct Animation ist im Gegensatz zu Adobe- (Macromedia-) Produkten äußerst preiswert: Null Euro. Direct Animation ist Teil von Windows und dessen Konzeption.

Direct Animation arbeitet ohne Plugins und lässt sich ohne Fremdsoftware programmieren: Nachfolgend eine Übersicht zu Windows DirectAnimation (DA) ab IE 3.x auf Basis von ActiveX-Controls von DirectX am Beispiel der CLASS-ID B6FFC24C-7E13-11D0-9B47-00C04FC2F51D (DirectX-SDK von 1998

Warnung: CLASS-ID **B6FFC24C-7E13-11D0-9B47-00C04FC2F51D** kann bereits unter Windows XP SP2 abgeschaltet sein:

Das Objekt zur Class-ID lässt sich erzeugen, aber die Bibliothek MeterLibrary ist null-Zeiger sein.

Es ist zwingend auf Vorhandensein der Bibliothek zu prüfen !

Des Weiteren muss damit gerechnet werden, dass Microsoft auch andere CLASS-ID abschaltet (siehe unten), so dass einst funktionierende Scripte irgendwann nicht mehr funktionieren.

Unter Windows XP SP1 dürfte die Bibliothek MeterLibrary existieren, da Windows XP SP1 von Microsoft per Definition nicht mehr weitergepflegt wird (außer aus Sicht Microsoft kritischer Patches, wobei Microsoft bereits permanent bescheinigt, dass Win XP SP1 ohne SP2 nicht mehr sicher ist.)

DirectX wird permanent weiterentwickelt und zwar z.Z. nicht komplett abwärtskompatibel:

Ganze DirectX-Bibliotheken werden ersetzt, die dann älteres DirectX nicht mehr unterstützen - warum auch immer.

Da DirectX auf Hardwaretreiber, die von Microsoft abgenommen sind, basiert, wird mit jedem neuen Windows, das die Treiberschnittstellen ändert (Win XP SP2 und Win Vista) das Risiko erneuert, dass Webseiten, die DirectX benutzen, im DirectX-Teil nicht mehr laufen werden.

Man beachte Abschlaltungen von Windows-Komponenten im Rahmen der Sicherheitspatches von Microsoft (siehe auch unten).

Die Scriptmaschine des IE kann JScript-Code auf Basis von ActiveX-Controls verarbeiten, der Anweisungen von DirectAnimation (DA) enthält.

Die Kodierung des ActiveX-Controls erfolgt per üblichen OBJECT-Tag im BODY-Teil des HTML-Dokumentes.

Für das STYLE-Attribut des OBJECT-Tags ist **dringend anzuraten**, es **nicht** zu kodieren, sondern **alle** Style-Angaben ausschliesslich durch Referenz per `zeiger_auf_da_objekt.style.style_wert =;` zu erzeugen. Grund: Je nach DA-Objekt-Implementierung wird das STYLE-Attribut im OBJECT-Tag teilweise oder vollständig ignoriert und damit wirkungslos. Der Zeiger auf das DA-Objekt ist der Wert des ID-Attributes im OBJECT-Tag oder der Zeiger laut `createElement()` vom OBJECT.

Eine Integration von JScript mit DA ist nur z.T. möglich, da DA ein eigenes Laufzeitsystem besitzt, das grundsätzlich Objekte **von DA** erwartet .

Alle DA-Aktionen werden erst mit dem Start des Laufzeites der DA-Animation abgearbeitet. Dabei ist zu beachten, dass das Laufzeitsystem und seine DA-Aktionen **sehr intensiv** die Timer von Windows nutzen. Es ist daher empfehlenswert, möglichst keine parallelen Timer per Direct Animation und/oder z.B. per Javascript-Anweisung `setTimeout()` zu nutzen. Mit Ablauf der DA-Animation ist dringend das Stoppen des Laufzeites und damit der Timernutzung zu empfehlen ! Beispielsweise kann eine permanente Direct Animation des Hintergrundes das Timing des Browsers und von Windows (z.B. PC-Uhr) stark verzögern bzw. sogar lahmlegen, so dass ein Neustart des Rechnersystems nötig werden kann. DA-Aktionen sind also **sparsam** zu verwenden ! Direct Animation stammt - trotz der noch immer ansehnlichen Animationsmöglichkeiten - aus älteren Zeiten von Windows und dem Internet Explorer. Es ist weiterhin empfehlenswert, schnelle PC-Hardware und ein modernes Windows (z.B. Windows XP) zu verwenden, da dann Timerprobleme weniger auftreten. Allerdings hat nicht jeder User diese Möglichkeiten. Aber auch dann können Timerprobleme auftreten.

Nachteilig ist bezüglich der Direct Animation mit Soundwiedergabe der Umstand, dass in der eventuell windoweigenen Lautstärkeregelung die vom User als dauerhaft eingestellten Reglerstellungen zum Ärgernis des Users temporär oder sogar permanent verändert werden. Eine Synchronisierung mit Lautstärken von per JScript erzeugten Soundobjekten (z.B. `bgSound` Objekt) ist z.T. **nicht** möglich, so dass der User erneut Einstellungen der Regler (je nach Sound-Medium wie Wave oder Midi) als Korrektur treffen muss.

Sämtliche DA-Objekte sind Instanzen einer DA-Bibliothek, die alle je nach Zweck der DA in den JScript-Code eingebunden werden müssen (auch die DA-Bibliothek selbst).

