

```

    }
  }
}
else
{
  // +++++ mindesten 1 Schlüssel fehlt, also Script in Registry eintragen
  Registry_ScriptRegistratur(true);
}

```

5.2.3.2.6.7. **Beispiel: Zugriff auf PATH-Variable**

Zugriff auf PATH-Variable

```

// WinScriptHost-Objekte
var WinScriptHost_LaufzeitUmgebung = WScript.CreateObject("WScript.Shell");
var WinScriptHost_PATHVariable = WinScriptHost_LaufzeitUmgebung.ExpandEnvironmentStrings("%path%");

// JScript-Dateisystem
var JScript_DateiSystem = new ActiveXObject("Scripting.FileSystemObject");

var PfadTrenner = ",";
var TeilPfad;
var TeilPfade_Anzahl = 0; // ab 1
var TeilPfade_Ungueltige = "";

var Zahler = 0,

// \" ersetzen durch Leerkette
WinScriptHost_PATHVariable = WinScriptHost_PATHVariable.replace(/\"/g, "");

// Semikolon zählen (Pfad-Trenner) durch zeichenweises abklappern
for (Zahler = 0; Zahler < WinScriptHost_PATHVariable.length; Zahler++)
{
  if (WinScriptHost_PATHVariable.charAt(Zahler) == PfadTrenner) {TeilPfade_Anzahl++;}
}

// Feld der Teilpfade erzeugen
var TeilPfade_Feld = new Array(TeilPfade_Anzahl-1); // Index ab 0

// Path-Kette nach Feld: Feldelement liegt zwischen Pfadtrenner
TeilPfade_Feld = WinScriptHost_PATHVariable.split(PfadTrenner);

// Teilpfad auf realen Ordner prüfen
for (Zahler = 0; Zahler < TeilPfade_Feld.length; Zahler++)
{
  // nächster Teilpfad
  TeilPfad=TeilPfade_Feld[Zahler];

  // prüfen auf realen Ordner
  if (JScript_DateiSystem.FolderExists(TeilPfad) == false)
  {
    // kein realer Ordner vorhanden
    TeilPfade_Ungueltige = TeilPfade_Ungueltige + TeilPfad + "\n";
  }
}

// Anzeige aller ungültigen Teilpfade im Path
if (TeilPfade_Ungueltige == "")
{WScript.Echo("Teilpfade existieren real !");}
else
WScript.Echo("Ungültige Teilpfade sind \n\n" + TeilPfade_Ungueltige);

```

5.2.4. **Windows Media Player 7.1 und Internet Explorer**

Der Windows Media Player dient als Addon u.a. zur Wiedergabe diverser Medien anhand von Medien-Dateien oder einer Playliste von Medien-Dateien.

Der Mediaplayer 7.1 wurde exemplarisch gewählt, da im SDK die Scriptprogrammierung offeriert wurde.

Es besteht die Möglichkeit, dass Nachfolgeversionen des Mediaplayer 7.1 ebenfalls diese Scriptprogrammierung unterstützen.

Vermutlich wurde aber ab MediaPlayer 10 Inkompatibilität hergestellt: Man muss es eben austesten.

Der Windows Media Player ist im HTML-Dokument per HTML und optional per JScript ansteuerbar, wobei der Programmierungsaufwand in JScript erheblich ist. Eine sehr einfache, aber wenig dynamische Programmierungsvariante ist beim Objekt bgound zu finden (siehe dort).

Der Windows Media Player 7.1 wird für den Netscape im Gegensatz zum Windows Media Player 6.4 als Plugin **nicht** mehr unterstützt. Ab dem IE 6.x werden generell keine Plugins mehr unterstützt. Microsoft favorisiert ihr eigenes ActiveX-Konzept und überlässt z.T.

Fremdherstellern die Programmierung von ActiveX-Controls z.B. als Addon zum Internet Explorer. Vorteil ist die Normung per Active-X-



Schnittstelle und somit die Komplettintegration z.B. des Players und IE in Windows. Nachteil ist das Ausbieten von Konkurrenzprodukten, wenn Microsoft die notwendigen Schnittstellen nicht komplett offengelegt hat und damit eine Weiterentwicklung durch die Konkurrenz vor allem für Anwender erschwert, die nicht unbedingt mit der Microsoft-Player-Software arbeiten wollen bzw. andere Features erwarten, also sie Microsoft bisher implementiert hat.

Eine etwas veraltete und ebenfalls programmtechnisch aufwändige Alternative ist die Medienwiedergabe per Microsoft Direct Animation (DA) als einer der Vorgänger bzw. Komponenten des Windows Media Player.

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

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.

Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.

Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Lizensierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks

Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.

Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.

Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2).

Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.

Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

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

Abänderungen wegen Sicherheitspatches der jeweiligen Windows-Versionen

Abschaltungen von Active-X-Controls erfolgen auch im Rahmen der Sicherheitspatches zu Windows-Versionen.

Es ist auch möglich, dass wegen Sicherheitslücken abgeschaltet wird und somit Komponenten einer Webseite je nach Windowsversion nicht mehr laufen.

Im Rahmen der Sicherheitspatches ist es Microsoft sogar gelungen, Webseiten, die den MS-Encoder zur Komprimierung von

HTML- und JScript-Code nutzen, schlagartig unnutzbar zu machen: Ein Bug in einem Patch zu Windows XP - Q918899

Das Patch verursacht IE-Browser-Absturz bei per MS ScriptEncoder gepacktem JScript unter SP1 und 2 wenn HTTP 1.1 mit Kompression genutzt wird z.B. bei onlick-Handler auf IMG klick ins Fenster per aktivem Popup

Der Absturz ist "read" -Fehler von immer ein und derselben Speicherstelle.

User, die dieses Patch installiert haben, können ab sofort keine IE-Seiten mit codiertem Script mehr ansehen.

Microsoft stellt Abhilfe nach geraumer Zeit zur Verfügung, jedoch spezifisch nach Windows XP-Version:

Patch Q918899 für

Windows XP SP1Download für jedermann bereitgestellt

SP2 nur auf kostenpflichtige telefonische Anfrage des Users per Downloadlink bereitgestellt, da

Microsoft explizit die User registriert haben will, bei denen das

Patchproblem auftritt (User muss sich Telefonnummer besorgen)

Solange also das Patch zum fehlerhaften Patch vom User nicht installiert wird,

z.B. weil der User keine Ahnung hat, dass und wo er sich die Telefonnummer

von Microsoft besorgen muss bzw. zu besorgen hat, wird der User

IE-Seiten mit komprimierten Code dauerhaft nicht nutzen können.

(Microsoft-Support ist z.T. nur in Englisch).

Abänderungen wegen Browser-Inkompatibilität

Popupblocker-Fehler

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()

Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.

Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.

Der Fehler tritt nicht auf, wenn ein Fenster per window.open() erzeugt wurde.

Bedingung:

Scriptfehleranzeige ist erlaubt im IE 7

Popupblocker ist im IE abgeschaltet

ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 window.popup per .show()erzeugt.

ein weiteres Fenster (Register) z.B. leere Seite (about:blank)

beide (Register) liegen in einer gemeinsamen IE-Instanz



Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per .show() erzeugt, bricht der Browser das Dokument mit .show() ab (Scriptfehler).

Der Popupblocker für die leere Seite verursacht den Programmfehler im Dokument mit .show(). Es wird folgende Meldung angezeigt (in der Informationsleiste):

'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'

Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:

Der Popupblocker hat ein Pop-up-Fenster geblockt. Sie können den Popupblocker deaktivieren oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.

Die Realität zur obigen Meldung ist völlig anders:

Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter
 Popupblocker einschalten
 weitere Informationen

jedoch keine Möglichkeit wie laut Bedeutung

Damit gilt: Der abgeschaltete Popupblocker ist in Wirklichkeit aktiv.

Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster einer Windowsanwendung z.B. einer anderen IE-Instanz)

Der Popupblocker des IE bemerkt aber NUR Webseite, die das Popup erzeugt.

Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popupblocker verwaltet wird.

Der Popupblocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen Webseiten (die nicht das Popup erzeugt haben).

Der Popupblocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.

Der Popupblockerfehler verändert die Eventverwaltung:

Es werden u.a. ignoriert

onfocus

onblur

onfocusin

onfocusout

und viele andere, so dass trotz Events z.B. des Body der Popupblockerfehler entsteht.

// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell

window.focus();

window.document.focus();

if(document.body!=null)

{if(document.body.style!='hidden') // wenn hidden so focus() nicht möglich (Scriptfehler erzeugt)

{document.body.focus();}

}

// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt

popupzeiger.show(...);

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

focus-Methode beim IE 7

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

Hinweis:

.focus() setzt Element aktiv, gibt dem Element den Focus und feuert dann onfocus

.setActive() ist Teilmenge von .focus(): nur das aktiv setzen

funktioniert nicht mit allen Elementen, mit denen .focus() funktioniert

animierte Gif (mit Timer)

Animierte Gifs (mit Timer), die unter IE 6 korrekt laufen, müssen unter IE 7 im Timer nicht mehr laufen:

z.B. garnicht mehr sichtbar, oder Timer nicht verwendet.

Dann müssen animierte Gif-Bilder nach IE-Version bereitgestellt werden.

Abänderungen wegen Rechtstreitigkeiten von Microsoft mit Fremdanbietern

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

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

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

Es wurde das Eventmodell gleichzeitig geändert:

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

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



wurde.

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

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

Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.

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

style.visibility='hidden' wird ignoriert

Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert:

Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.

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

Abänderung wegen Abschaltungen

DirectX ist wegen Abschaltung von Active-X--Controls nicht mehr abwärtskompatibel:

Z.B. wurde bei Win XP SP2 Direct Animation aus DirectX schlagartig durch Abschaltung von Bibliotheken dezimiert, die es bei Win XP SP1 aber noch gibt.

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

ActiveX-Controls und Unterstützung/Verbot 20041215

erlaubt sind noch

Tabular Data-Steuerelement {333C7BC4-460F-11D0-BC04-0080C7055A83} Das TDC (Tabular Data-Steuerelement) ermöglicht die Weiterverarbeitung von Daten, die nur im Textformat vorliegen, beispielsweise durch Darstellung in einer Tabelle oder Sortierung. Weitere Informationen:•

http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp(http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp)

Microsoft Agent Control - Version 2.0 {D45FD31B-5C6E-11D1-9EC1-00C04FD7081F} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Weitere Informationen:•

<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>(<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Microsoft MSChat-Steuerelement-Objekt 2.0 - 2.5 {D6526FE0-E651-11CF-99CB-00C04FD64497}

Dieses Steuerelement wird von Webautoren verwendet, um text- und graphisch basierte Chatgemeinden für Echtzeitkonversationen im Web zu erstellen.

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

• http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp (http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp)

verboten sind



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

<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>
<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>
http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp
http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp

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

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

Internet Explorer

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

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

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

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

Media Player / Windows Media Player {22D6F312-B0F6-11D0-94AB-0080C74C7E95} Dies ist das Steuerelement für Windows Media Player version 6.4 und war Installationsbestandteil bis einschließlich Windows Media Player Version 8. Ab Windows Media Player 9 wurde diese ClassID durch die neue ClassID {6BF52A52-394A-11D3-B153-00C04F79FAA6} abgelöst, deren Verwendung stattdessen empfohlen wird. Ab Windows Media Player Version 9 wird ferner die alte ClassID anhand eines Wrappers automatisch auf die neue



ClassID umgeleitet. Die ClassID für Windows Media Player Version 9 ist jedoch nicht in der Liste der vom Administrator genehmigten Steuerelemente enthalten, und muss bei Bedarf manuell hinzugefügt werden.

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

IE Label-Steuerelement

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

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

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

IE Timer-Steuerelement {59CCB4A0-727D-11CF-AC36-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

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

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

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



<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>
 (http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm)

Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden. Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS

exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Wenn der IE 6.x aktive Inhalte blockiert, wird NOSCRIPT-Tag aktiviert, Ausnahme: Frameset

FRAMESET ist ein aktiver Inhalt:

Da der Frameset anstelle <BODY> kodiert sein muss, gilt:

Alle Tags, die für BODY zulässig sind, werden ignoriert, auch NOSCRIPT.

Wird neben Frameset noch BODY kodiert, so wird Frameset ignoriert.

Die Freigabe der Scriptblockierung erzeugt Ausführung aller Script-Teile inklusive der Eventauslösungen

Bsp.: Folgendes funktioniert vom Dokument, das window.open() hat im geöffneten Dokument (Quelltext im Dokument das window.open() verwendet):

```
function Y_unload(X00){X85[X00].close();}

var X85=new Array();var X86=new Array();

X85[0]=window.open(...);

var X87=parent.Y_unload(0); X86[0]=new Function("X87);

X85[0].document.body.onunload=X86[0];
```

Wird die Scriptblockierung im geöffneten Fenster abgeschaltet,
so wird das Fenster geschlossen, weil onunload ausgelöst wird.

Achtung: document.body.onunload funktioniert ev. nicht mehr
wenn z.B. mit attachevent() aktiviert wurde

Folgende Metatags sind für den IE 6.x aktiver Inhalt:

```
<META HTTP-EQUIV="imagetoolbar" CONTENT="no">
  unterdrückt NICHT IE-Kontextmenü rechte Maus auf Bild

<META HTTP-EQUIV="site-enter" CONTENT="revealtrans(duration=0.3, transition=12)">
<META HTTP-EQUIV="site-exit" CONTENT="revealtrans(duration=0.3, transition=12)">
```

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.

Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein,
innerhalb

dessen dann die neuen HTML-Elemente erzeugt werden.

Beispiel für Prüfung 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

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

Beispiel für Browser-Erkennung:

```
var UserAgent        = navigator.userAgent;
var Browser_Version_Haupt = 0;
var Browser_Version_Unter = 0;

// Browserversion ermitteln
```



```

if (UserAgent != "")
{
    // Hauptversion des Browsers holen
    for (var i=0; i < 10; i++)
    {
        eval(      'if (UserAgent.indexOf("'
                    + i
                    + '") != -1)'
                    + '{Browser_Version_Haupt = '
                    + i
                    + '};}'
                );
    }

    // Unterversion des Browsers holen
    for (var i=0; i < 100; i++)
    {
        eval(      'if (UserAgent.indexOf("'
                    + i
                    + '") != -1)'
                    + '{Browser_Version_Unter = '
                    + i
                    + '};}'
                );
    }

    // wenn Unterversion des Browsers am Ende eine Null hat, so diese eliminieren z.B. 50 zu 5
    if ((Browser_Version_Unter % 10) == 0)
    {Browser_Version_Unter = Browser_Version_Unter / 10;}    // Bsp.: 50 % 10 = 5 Rest 0, also 5
}

var NSunter6= (      (!document.all)                // kein NS kennt document.all
                || (document.layers)                // bis NS unter 6.x
            );

var NSab6= (      (!document.all)                // kein NS kennt document.all
                && (document.getElementById)        // ab NS 6.x ist getElementById implementiert
                // theoretisch wäre zusätzlich (!document.layers) kodierbar, aber mit Implementation
                // von getElementById wurde gleichzeitig document.layers abgeschafft
            );

var NS4x = (      (NSunter6)
                && (Browser_Version_Haupt >= 4)
                && (Browser_Version_Haupt < 5)
            );

var NS6x = NSab6;

// theoretisch kann noch auf die Hauptversionsnummer geprüft werden
// var NS6x = (      (NSab6)
//                  && (Browser_Version_Haupt >= 6)
//                  );

var IE = (document.all) ? true : false;    // if-Anweisung ist wichtig, sonst erfolgt Zeigerzuweisung

var IEab5 = (      (IE)
                && (document.getElementById)        // ab IE 5.x ist getElementById
                implementiert
            );

var IE4x = (      (IE)                // NS-Browser setzt IE auf false
                && (!IEab5)           // NS-Browser setzt !IEab5 auf true
                // Es muss (IE) && (!IEab5) abgefragt
                // Würde nur (!IEab5) kodiert
                // dann würde der NS-Browser
                // IE4x auf true setzen
                && (Browser_Version_Haupt >= 4)
                && (Browser_Version_Haupt < 5)
            );

var IE5x = (      (IEab5)
                && (Browser_Version_Haupt >= 5)
            );

```




```

        && (Browser_Version_Haupt < 6)
    );

    var IE55 = (
        (IEab5)
        && (Browser_Version_Haupt == 5)
        && (Browser_Version_Unter == 5)
    );

    var IE6x = (
        (IEab5)
        && (Browser_Version_Haupt >= 5)
    );

```

Wiedergabe von Medien mit dem Windows Media Player und HTML-Navigation im Dokument:

Die aktive Wiedergabe eines Mediums wird mit Wechsel des Dokumentes z.B. per <A HREF....> automatisch gestoppt. Der Player ist also nur im Dokument verwendbar, in das er eingebettet ist und das nur solange, wie das Dokument geladen ist.

Vergleich .style.time2 Behavior und Windows Media Player ab 7.1:

Die Objekte und Collectionen zum Player und zum Behavior .style.time2 sind konzeptionell ähnlich:

Der Player und .style.time2 basieren auf gemeinsamen Prinzipien, nur dass sich die referenzierten Objekte und Collectionen anders nennen bzw. verschieden aufgebaut sind.

Die Eigenschaften zum .style.time2 Behavior und zum Windows Media Player sind sehr ähnlich, wenn nicht gar identisch.

Während der Windows Media Player alle Windows-kompatiblen Medienarten unter einem gemeinsamen Dach verwalten kann, muss beim .style.time2 Behavior das richtige Behavior-Objekt gewählt werden, um entsprechende spezielle Methoden zum Typ des Mediums ansprechen zu können.

Das Objektmodell des Windows Media Players ist wesentlich ganzheitlicher, dadurch übersichtlicher, und hat einen größeren Umfang als das des Behaviors. Z.B. kann nur der Player CD-Laufwerke als Systemkomponente verwalten.

Die Programmierung von .style.time2 ist durch die Verwendung des Players vielleicht unnötig, setzt aber voraus, dass der Player auch verwendet werden **soll**.

Per .style.time2 Behavior kann der Windows Media Player eingebunden werden, muss aber nicht. Wird der Player eingebunden, so ist er per Behavior ansprechbar. Die Programmierung des Players setzt **aber** eine permanente Player-Instanz im HTML-Dokument voraus und damit ein permanentes Playerfenster. Außerdem wird der Player im **HTML-Dokument** nicht mit allen seine Objekteigenschaften unterstützt, was vielleicht doch wieder für die Nutzung des Behavior .style.time2 spricht.

Es ist zu vermuten, dass

der Behavior ebenfalls das Active-X-Control benutzt, wenn der Windows Media Player in den Behavior eingebunden ist.

die Verwendung des Behaviors ressourcenschonender ist, denn es werden nur Eigenschaften und Methoden instanziiert, die für das Behavior-Objekt nötig sind.

Behavior .style.time2 wird per XML-Tag in das HTML-Dokument eingebunden. Der Player wird in das HTML-Dokument per Active-X-Control im OBJECT-Tag eingebunden.

Beispiele zum Windows Media Player:

Alle Beispiele dienen nur zum Verständnis der Programmierung unter JScript.

5.2.4.1. Begriffe

5.2.4.1.1. Media Datei

Eine Media Datei

kann lokal auf dem User-PC oder im Netzwerk (z.B. Internet) liegen

ist Video und Audio in diversen Formen z.B.:

nicht window-spezifisch:	*.AVI
	*.MID
	*.MP3
	*.MPEG
	*.WAV
window-spezifisch	*.WM
	*.WMA
	*.WMV
	*.ASF

Datei-Suffix	MIME type	Windows-Media-Datei
*.wma	audio/x-ms-wma	nur Audio
*.wmv	video/x-ms-wmv	Audio/Video
*.asf	video/x-ms-asf	Audio/Video

hat Media-Typ z.B. "Audio" oder "Video"



kann Attribute folgende Attribute besitzen

"Album"	nur bei Media Item
"Artist"	nicht für Playlist, die per Methode ID_Player.mediaCollection.getByXXX() oder ID_Player.mediaCollection.getAll() erzeugt wurde
"Author"	
"Bitrate"	Bitrate, nur bei Element aus Media-Bibliothek
"Copyright"	nur bei Playlist-Eintrag nicht Playlist von CD nicht Media-Item
"CreationDate"	Datum des Hinzufügen des Elementes zur Media-Bibliothek nicht Playlist und Media Item
"DigitallySecure"	Datum des Schutzes des Elementes in der Media-Bibliothek Schutz = Digital Rights Management nicht Playlist und Media Item
"Genre"	nur Playlist-Eintrag nicht Playlist von CD
"MediaType"	Media-Typ (audio oder video)
"Name"	Name des Playlist-Eintrages
"PlayCount"	Anzahl der Wiederholungen eines Elementes aus der Media-Bibliothek nicht Playlist und Media Item
"SourceURL"	Url oder Pfad und Dateiname der Media-Datei als Element in der Media Bibliothek nicht Playlist und Media Item
"TOC"	CD Table of Contents Identifier nur für Playlist von CD nicht für Playlist, die per Methode ID_Player.mediaCollection.getByXXX() oder ID_Player.mediaCollection.getAll() erzeugt wurde

Die Attribute haben natürlich je nach Media-Datei einen konkreten Inhalt.

5.2.4.1.1.1. **Media Datei als nicht window-spezifische Media Datei**

Die Media Datei enthält **nur** Daten des Medium, das nicht nur unter Windows verwaltet werden kann.

z.B. Sound wie MP3 etc.
Media Clip als Video mit Sound.

Diese Art von Media Dateien existiert z.T. schon seit DOS-Zeiten, also vor Windows.

5.2.4.1.1.2. **Media Datei als window-spezifische Media Datei**

5.2.4.1.1.2.1. **Media Datei als Windows Media Datei**

Die Media Datei enthält **nur** Daten des Medium, das nur unter Windows verwaltet werden kann.

Typisches Kennzeichen (leider nicht immer) dieser Media Datei ist der Suffix *.Wxx mit **W** für Windows

z.B. Dateien mit Suffix *.wma
*.wmv
aber auch *.asf

5.2.4.1.1.2.2. **Media Datei als Windows Meta Datei**

Eine Meta Datei ist eine Script-Datei (z.B. ASX-Datei), die

spezifische Meta-Elemente zur Verwaltung von Media Dateien und Media Daten besitzt
folgende Media Dateien laden kann:

nicht-window-spezifische	Media Datei(en) wie MP3-Sound oder Videoclip
window-spezifische	Media Datei(en), aber nur dann, wenn die Windows-Kompressionstechnologie zur Medienart auch in der Verwaltung implementiert ist
window-spezifische	Meta Datei(en), aber nur teilweise

Übersicht zu Windows Meta Dateien und ladbaren Windows Media Dateien:

Meta-Datei-Suffix	Windows-Media-Datei-Suffixe
	*.asf *.wma *.wmv



*.wvx	X	X	X
*.wax	X	X	
*.asx	X		

X = ladbar

Übersicht zu Windows Meta Dateien:

Mime-Typ der	Windows Meta	Datei
ladbare	Windows Media	Datei
ladbare	Windows Meta	Datei

Meta Datei-Suffix	MIME Type	Windows Media-Datei-Suffix bzw. Meta-Datei-Suffix
*.wax	audio/x-ms-wax	Media-Datei mit Suffix *.asf Meta-Datei mit Suffix *.wma *.wax
*.wvx	video/x-ms-wvx	Media-Datei mit Suffix Meta-Datei mit Suffix *.wma *.wmv *.wvx
*.asx	video/x-ms-asf	Media-Datei mit Suffix *.asf Meta-Datei mit Suffix *.wma *.wmv *.wvx

5.2.4.1.2. Media Bibliothek

Die Media-Bibliothek ist eine Sammlung

von media Objekten: auf Basis der Collection mediaCollection

von Playlisten: Playlist besteht aus mindestens 2 Media Items als gelistete media Objekte
Jede Playliste hat einen eindeutigen Namen.

als Datenbank: Namensindex (Namen der Objekte)
Datenbank wird vom Player automatisch verwaltet und gespeichert.

als Datei-Pool: als Menge physischer Verknüpfungen bei lokalen Media Dateien auf der Festplatte
als Menge logischer Verknüpfungen bei nicht-lokalen Media Dateien (z.B. im Netz)

5.2.4.1.3. Playliste

Eine Playliste ist Liste von mindestens 2 media Objekten aus der Media-Bibliothek.

In der Playliste wird ein media Objekt als Media Item Objekt bezeichnet

Eine Playliste kann

lokal oder im Netzwerk liegen
Datenbank oder Textdatei sein

5.2.4.1.4. JScript: media Objekt, Media Item Objekt und Collection mediaCollection

Ein media Objekt ist

eine media-typ-gerecht instanzierte Media-Datei mit den Attributen der Media-Datei
ein Media Item Objekt als Playlisten-Eintrag, der einer Playliste zugeordnet ist.

Alle media Objekte werden in der Collection mediaCollection verwaltet (jedes Element ist ein media Objekt)

Das **aktuelle** media Objekt wird gerade im Player wiedergegeben.

Hinweis: Es kann zu jedem Zeitpunkt nur 1 Objekt im Player wiedergegeben werden, also aktuell sein.

5.2.4.2. Varianten des Windows Media Player (der ActiveX-Controls)

Variante	Wert des CLASSID-Attributes im OBJECT-Tag
Windows Media Player in minimaler Version der Control-Elemente 00AA0055595A	05589FA1-C356-11CE-BF01-00AA0055595A
Windows Media Player in erweiterter Version der Control-Elemente Media Bar Player, also Windows Media Player	22d6f312-b0f6-11d0-94ab-0080c74c7e95 52ca3bcf-3b9b-419e-a3d6-5d28c0b0b50c
Achtung: CLASSID wird inzwischen nicht mehr sicher unterstützt ! Windows Media Player 7.1	6BF52A52-394A-11D3-B153-00C04F79FAA6

Hinweis: Groß-Kleinschreibung des Wertes vom CLASSID-Attributes ist egal.

Beispiele für Active-X-Control des Mediaplayers zum Abspielen eines Videos oder Sounds:

Achtung: CLASSID wird inzwischen nicht mehr sicher unterstützt (siehe unten Microsoft ändert die Active-X-Eigenschaften)

```
<BODY>
<OBJECT ID="ActiveMovie1"
        WIDTH=356
        HEIGHT=328
        CLASSID="CLSID:05589FA1-C356-11CE-BF01-00AA0055595A"
>
    <PARAM NAME="MovieWindowWidth" VALUE="352">
    <PARAM NAME="MovieWindowHeight" VALUE="247">
    <PARAM NAME="FileName" VALUE="0.wmv">
</OBJECT>
```

Das Active-X-Control umfasst folgende PARAM



```

<PARAM NAME="Appearance" VALUE="0">
<PARAM NAME="AutoStart" VALUE="0">
<PARAM NAME="AllowChangeDisplayMode" VALUE="-1">
<PARAM NAME="AllowHideDisplay" VALUE="0">
<PARAM NAME="AllowHideControls" VALUE="-1">
<PARAM NAME="AutoRewind" VALUE="-1">
<PARAM NAME="Balance" VALUE="0">
<PARAM NAME="CurrentPosition" VALUE="0">
<PARAM NAME="DisplayBackColor" VALUE="0">
<PARAM NAME="DisplayForeColor" VALUE="16777215">
<PARAM NAME="DisplayMode" VALUE="0">
<PARAM NAME="Enabled" VALUE="-1">
<PARAM NAME="EnableContextMenu" VALUE="-1">
<PARAM NAME="EnablePositionControls" VALUE="-1">
<PARAM NAME="EnableSelectionControls" VALUE="0">
<PARAM NAME="EnableTracker" VALUE="-1">
<PARAM NAME="Filename" VALUE="">
<PARAM NAME="FullScreenMode" VALUE="0">
<PARAM NAME="MovieWindowSize" VALUE="0">
<PARAM NAME="PlayCount" VALUE="1">
<PARAM NAME="Rate" VALUE="1">
<PARAM NAME="SelectionStart" VALUE="-1">
<PARAM NAME="SelectionEnd" VALUE="-1">
<PARAM NAME="ShowControls" VALUE="-1">
<PARAM NAME="ShowDisplay" VALUE="-1">
<PARAM NAME="ShowPositionControls" VALUE="0">
<PARAM NAME="ShowTracker" VALUE="-1">
<PARAM NAME="Volume" VALUE="-600">

```

VALUE-Angaben sind nur Beispiele

Bezeichner der Eigenschaften für JScript sind IDENTISCH mit den Werten aus NAME-Attribut von PARAM !
Gross-Kleinschreibung beachten !

User	AllowChangeDisplayMode	Änderung des Anzeigemodus Zeitanzeige bzw. Framesanzeige durch User boolean true oder false
	AllowHideControls	Änderung des Anzeigemodus der Steuerungstasten während Wiedergabe durch User boolean true oder false false (0) für nicht erlauben
	AllowHideDisplay	Änderung des Anzeigemodus der Wiedergabe-Anzeige während Wiedergabe durch boolean true oder false false (0) für nicht erlauben
	AutoRewind	Automatisches Rückspulen nach Stop boolean true oder false false (0) für nicht rückspulen
	AutoStart	Autostart der Wiedergabe ein oder aus boolean true oder false false (0) für nicht automatische Wiedergabe: siehe Aktionen zum Player
	Balance	numerisch 0 für zentriert default -1000 ganz links +1000 ganz rechts
	CurrentPosition	unklar
	DisplayBackColor	Hintergrundfarbe der Control-Elemente numerischer Farbwert, kein Hexastring 0 Default für Black 16777215 für Weiss
	DisplayForeColor	Hintergrundfarbe der Control-Elemente numerischer Farbwert, kein Hexastring 0 Default für Black 16777215 für Weiss
	DisplayMode	Anzeigeart der Position 0 für Time



	1 für Frames
Enabled	Control-Anzeige ein, aus boolean true oder false false (0) für nicht erlauben
EnableContextMenu	Rechte-Maus-Kontextmenü ein, aus boolean true oder false false (0) für nicht erlauben
EnablePositionControls	Positionierungs-Button im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
EnableSelectionControls	Selektions-Button im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
EnableTracker	Tracker-Balken im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
FileName	Mediendatei mit Pfad oder Url
FullScreenMode	Vollbildschirm ein, aus boolean true oder false false (0) für kein Vollbildmodus
MovieWindowSize	unklar, Grösse des Anzeigefensters
PlayCount	Anzahl der Wiedergaben der Mediendatei ab 1
Rate	Wiedergabefaktor numerisch 1 für 1-fache Geschwindigkeit 10 für 20-fache Geschwindigkeit
SelectionEnd	Bis-Position, bis zu der wiedergegeben werden soll in Sekunden bezüglich Anfang der Mediendatei (0)
SelectionStart	Von-Position, bis zu der wiedergegeben werden soll in Sekunden bezüglich Anfang der Mediendatei (0)
ShowControls	Start Stop buttons and tracker controls sichtbar ein, aus boolean true oder false false (0) für aus
ShowDisplay	Status-Anzeige-Panel sichtbar ein, aus boolean true oder false false (0) für aus
ShowPositionControls	Previous, Rewind, Forward and Next buttons sichtbar ein, aus boolean true oder false false (0) für aus
ShowSelectionControls	Selektions-Button im Control-Panel sichtbar ein, aus boolean true oder false false (0) für aus
ShowTracker	Trackerbar im Controlpanel sichtbar ein, aus: nur lesen boolean true oder false false (0) für aus
uiMode	Sichtbarkeit anstelle STYLE-visible "invisible" für unsichtbar
Volume	Laustärke als Basis dient die aktuelle Windows-Lautstärke integer 0 für maximimale, also aktuelle Windows-Lautstärke (höher geht nicht !!!) -10000 für total still

Den ganzen MediaPlayer unsichtbar machen: per STYLE-Attribut im OBJECT-Tag



document.ID-Bezeichner.readyState (MediaPlayer 6.4, funktioniert aber auch mit höheren Versionen)

.readyState hat folgende Werte

0	The FileName property has not been initialized.
1	Windows Media Player control is asynchronously loading a file.
3	Windows Media Player control loaded a file, and downloaded enough data to play the file, but has not yet received all data.
4	All data has been downloaded.

Methoden:

.IsSoundCardEnabled()

liefert boolean
true Soundkarte erkannt
false keine Soundkarte erkannt

.Run(); Erzeugt und startet Instanz des Mediaplayers und ermöglicht danach weitere Aktionen wie .Play etc.

.Mute ohne ()

lesen, schreiben
true Player volume is muted.
false (Default) Player volume is not muted.

.Play ohne ()

Wiedergabe starten

.AutoStart ohne ()

true (Default) Clip automatically starts.
false Disables automatic start.

.Stop ohne ()

.Pause ohne ()

Audio wiedergeben

```
<HEAD>
<SCRIPT>
var SoundStarten_TimeoutID=0;

function SoundStarten()
{
    // auf Soundkarte prüfen
    if (ID_PlayerObjekt.IsSoundCardEnabled())
    {
        // Soundkarte vorhanden
        // Musik geladen und abspielbar ?
        if (ID_PlayerObjekt.readyState == 4)
        {
            // geladen, also Playerinstanz starten
            ID_PlayerObjekt.Run();
            // Musik wiedergeben
            ID_PlayerObjekt.Play;
            alert('Musik ist gestartet !');
        }
        else { SoundStarten_TimeoutID=window.setTimeout('SoundStarten()', 100); }
    }
    else { alert('Keine Soundkarte gefunden !'); }
}
</SCRIPT>
</HEAD>
<BODY>
<OBJECT ID="ID_PlayerObjekt" CLASSID="CLSID:05589FA1-C356-11CE-BF01-00AA0055595A">
    <PARAM NAME="FileName" VALUE="a0.mp3">
    <PARAM NAME="AutoStart" VALUE="0">    <!-- kein Autoplay -->
</OBJECT>
<SCRIPT>
if (ID_PlayerObjekt != null) { SoundStarten(); }
</SCRIPT>
</BODY>
```



</HTML>

5.2.4.3. Instanziierung des Windows Media Player im HTML-Dokument

Zunächst erst ein Beispiel, das zur **Verdeutlichung** wichtiger Kriterien der Instanziierung dient:

```
<HTML>
<HEAD>
<SCRIPT>
<!--
    function WiedergabeStarten()
    { ID_Player.controls.play();}

    function WiedergabeStoppen()
    { ID_Player.controls.stop();}

-->
</SCRIPT>
</HEAD>
<BODY>
    <OBJECT          ID="ID_Player"
                    CLASSID="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"
    >
        <PARAM NAME="URL" VALUE="test.wma">
    </OBJECT>
    <INPUT TYPE="BUTTON" NAME="StartButton" VALUE="Play" onclick="WiedergabeStarten()">
    <INPUT TYPE="BUTTON" NAME="StopButton" VALUE="Stop" onclick="WiedergabeStoppen()">
</BODY>
</HTML>
```

5.2.4.3.1 Belegung CLASSID-Attribut im OBJECT-Tag

Variante	Wert des CLASSID-Attributes im OBJECT-Tag
Windows Media Player in minimaler Version der Control-Elemente	05589FA1-C356-11CE-BF01-00AA0055595A
Windows Media Player in erweiterter Version der Control-Elemente	22d6f312-b0f6-11d0-94ab-0080c74c7e95
Media Bar Player, also Windows Media Player	52ca3bcf-3b9b-419e-a3d6-5d28c0b0b50c
Windows Media Player 7.1	6BF52A52-394A-11D3-B153-00C04F79FAA6

Hinweis: Groß-Kleinschreibung des Wertes vom CLASSID-Attributes ist egal.

5.2.4.3.2. ID-Attribut für die Referenz auf die Instanz des Windows Media Players

Der Wert des ID-Attributes im OBJECT-Tag wird als Referenz auf das Objekt des Windows Media Player benutzt. Über dieses ID kann auf sämtliche Objekte und Collectionen zum Windows Media Player zugegriffen werden.

In der nachfolgenden Beschreibung zum Windows Media Player wird "ID_Player" synonym zum Begriff "Objekt des Windows Media Players" verwendet.

5.2.4.3.3. HTML-Vorbelegung von Eigenschaften der Instanz des Windows Media Players

Nicht alle Eigenschaften des Windows Media Player sind schreibbar, also mit Wert belegbar. HTML bietet nur einen Ausschnitt der implementierten Eigenschaften des Windows Media Players.

Das PARAM-Tag innerhalb des OBJECT-Tags lässt die Vorbelegung einer **schreibbaren** Eigenschaft des Windows Media Player per HTML zu. Die Vorbelegung ist optional und kann auch komplett in JScript erfolgen (dort mit allen implementierten und schreibbaren Eigenschaften).

Pro Eigenschaft, die in HTML belegbar ist, existiert ein PARAM-Tag:

Name der Eigenschaft laut NAME-Attribut
Wert der Eigenschaft laut VALUE-Attribut

Objekt	Eigenschaft als String-Wert in NAME
ID_Player	enabled enableContextMenu fullScreen stretchToFit uiMode URL
ID_Player.closedCaption	captioningID SAMIFilename SAMILang SAMISyle
ID_Player.controls	currentMarker currentPosition
ID_Player.currentMedia	duration name



ID_Player.settings	autoStart
	balance
	baseURL
	defaultFrame
	enableErrorDialogs
	invokeURLs
	mute
	playCount
	rate
	volume

Hinweis: Groß-Kleinschreibung des Wertes im NAME-Attribut ist egal.

Der Wert des VALUE-Attributes im PARAM-Tag ist der Wert der Eigenschaft, mit dem sie vorbelegt wird.

Mögliche Werte des VALUE-Attributes: siehe entsprechende Objektbeschreibung

Beispiele zum Sinn der Vorbelegung von Eigenschaften:

Control-Elemente des Players:

Wenn Control-Elemente des Players sichtbar sind, dann kann der User die Wiedergabe steuern. Die Art der Control-Elemente wird automatisch zum Typ der Media-Datei erzeugt (Tasten zur Steuerung der

Wiedergabe

durch den User, wobei diese zum Typ der Media-Datei passen).

Sollen die Control-Elemente generell sichtbar oder nicht sichtbar sein, so ist das auch per Script über die Eigenschaft ID_Player.enabled einstellbar.

Soll der Umfang der Control-Elemente eingestellt werden, so ist das auch per Script über die Eigenschaft ID_Player.uiMode möglich.

Videogröße:

Es kann sein, dass die widerzugebende Media-Datei einen größeren Platz braucht, als das aktuelle Player-Fenster. Dann muss der User das Fenster manuell vergrößern. Das lässt sich aber vermeiden.

Alternativ kann auch per Script die Eigenschaft ID_Player.stretchToFit verwendet werden.

5.2.4.3.4. HTML-Attribute WIDTH und HEIGHT im OBJECT-Tag

Diese Attribute bestimmen die Dimension des Playerfenster und die Möglichkeit der Darstellung von Control-Elementen (Tasten zur Steuerung der Wiedergabe durch den User, wobei diese zum Typ der Media-Datei passen).

Eine Höhenangabe muss mindestens 41 Pixel sein, damit überhaupt Control-Elemente des Players sichtbar werden können. 41 Pixel benötigt die Leiste der Control-Elemente mindestens.

Alternativ zu Angaben per PARAM-Attribute im OBJECT-Tag kann die Anzeige von Control-Elementen auch per Script gesteuert werden:

Sollen die Control-Elemente generell sichtbar oder nicht sichtbar sein, so ist das per Script über die Eigenschaft

ID_Player.enabled

einstellbar.

Soll der Umfang der Control-Elemente eingestellt werden, so ist das per Script über die Eigenschaft

ID_Player.uiMode

einstellbar.

Es kann sein, dass die widerzugebende Media-Datei einen größeren Platz braucht, als das aktuelle Player-Fenster. Dann muss der User das Fenster manuell vergrößern. Alternativ kann auch per Script die Eigenschaft ID_Player.stretchToFit verwendet werden.

5.2.4.3.5. automatische Fehleranzeige

Die automatische Fehleranzeige ist ein- bzw. abschaltbar per Eigenschaft ID_Player.settings.enableErrorDialogs

5.2.4.3.6. Beispiel

Hinweis: In diesem Beispiel wurden Kommentare innerhalb von HTML ungeachtet der Syntaxvorschriften eingearbeitet.

Im Beispiel werden GIF-Dateien verwendet:

Wenn im Datei-Namen die Zeichenkette "_hi" auftaucht, dann ist es die **aufgehellte** Grafik der Normalversion

Wenn im Datei-Namen die Zeichenkette "_low" auftaucht, dann ist es die **abgedunkelte** Grafik der Normalversion

Bsp.:	stop.gif	Normalversion
	stop_hi.gif	aufgehellte Normalversion
	stop_low.gif	abgedunkelte Normalversion

<HTML>

<HEAD>

<SCRIPT>

```
var ActiveXAktiv; // ist true wenn ActiveX benutzt wird
                  // ist sonst false
```

```
var timerID;
```



```

function Init() // mit Start des Dokumentes aktiviert im BODY
{
    ActiveXAktiv=false; // Annahme: kein ActiveX

    // prüfen auf ActiveX
    if ( (navigator.userAgent.indexOf('IE') > -1)
        && (navigator.platform == "Win32")
        )
    {
        // ActiveX erkannt
        ActiveXAktiv = true;
    }

    // Player starten anstelle Autostart im OBJECT-Tag
    Starten();
}

// Eventhandler
function EventHandler(NewState)
{
    // alle Bilder in abgedunkelter Version anzeigen
    timerID = setTimeout("document.all.Name_Image1.src = 'play_low.gif';",0);
    timerID = setTimeout("document.all.Name_Image2.src = 'pause_low.gif';",0);
    timerID = setTimeout("document.all.Name_Image3.src = 'stop_low.gif';",0);

    // den neuen Status abklappen und Bilder in dementsprechend anzeigen
    switch(NewState)
    {
        case 1:
            timerID = setTimeout("document.all.Name_Image3.src = 'stop_hi.gif';",0);
            alert('gestoppt');
            break;
        case 2:
            timerID = setTimeout("document.all.Name_Image2.src = 'pause_hi.gif';",0);
            alert('pausiert');
            break;
        case 3:
            timerID = setTimeout("document.all.Name_Image1.src = 'play_hi.gif';",0);
            alert('gestartet');
            break;
    }
}

function BildWechsel(ID_BildAlsKette,whichImage, URL_BildAlsKette)
{
    // Wechsel nur, wenn es kein aufgehelltes Bild ist
    if ( ID_BildAlsKette.src.indexOf('_hi') == -1 )
    { ID_BildAlsKette.src = URL_BildAlsKette; }
}

function Starten()
{
    if (ActiveXAktiv == true)
    {
        // ActiveX
        document.all.ID_Player.controls.play();
    }
}

function Stoppen()
{
    if (ActiveXAktiv == true)
    {
        // ActiveX

        // Wiedergabe stoppen
        document.all.ID_Player.controls.stop();

        // Player auf Anfang setzen, also aktuelle Position in Media-Datei ist 0
        document.all.ID_Player.controls.currentPosition=0;
    }
}

function Pausieren() // Status muss 3 sein also pausieren

```



```

    {
        if (ActiveXAktiv == true)
        {
            // ActiveX
            if (document.all.ID_Player.playState == 3)
            {document.all.ID_Player.controls.pause();}
        }
    }
</SCRIPT>
</HEAD>

<BODY onload="Init()">
    <SCRIPT FOR="ID_Player" EVENT="playStateChange(NewState)" LANGUAGE="JScript">
        // IE-Event-Capturing starten
        EventHandler(NewState);
    </SCRIPT>

    <OBJECT ID="ID_Player"
        CLASSID="CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6"
        WITH=176
        HEIGHT=144
    >
        <PARAM NAME="AutoStart" VALUE="False">
        <PARAM NAME="URL" VALUE="test.wma">
    </OBJECT>

    <TABLE>
    <TR>
    <TD>
        <A HREF="#" onmouseover="BildWechsel(
        Name_Image1, 'play.gif')
        onmouseout="BildWechsel(
        Name_Image1, 'play_low.gif')
        onclick="Starten()"
        >
            <IMG NAME="Name_Image1" SRC="play_low.gif" BORDER=0>
        </A>
    </TD>
    <TD>
        <A HREF="#" onmouseover="BildWechsel(
        Name_Image2, 'pause.gif')
        onmouseout="BildWechsel(
        Name_Image2, 'pause_low.gif')
        onclick="Pausieren()"
        >
            <IMG NAME="Name_Image2" SRC="pause_low.gif" BORDER=0>
        </A>
    </TD>
    <TD>
        <A HREF="#" onmouseover="BildWechsel(
        Name_Image3, 'stop.gif')
        onmouseout="BildWechsel(
        Name_Image3, 'stop_low.gif')
        onclick="Stoppen()"
        >
            <IMG NAME="Name_image3" SRC="stop_low.gif" BORDER=0>
        </A>
    </TD>
    </TR>
    </TABLE>
</BODY>
</HTML>

```

5.2.4.4. Eventbehandlung

Events laufen beim Windows Media Player nur auf der obersten Ebene, also auf der Ebene der Instanz laut ID-Attribut im OBJECT-Tag, ein.

Zunächst erst ein Beispiel, das zur Verdeutlichung wichtiger Kriterien der Eventbehandlung dient:

```

<HEAD>
<SCRIPT>
    function EventHandler(NewState)
    {
        // hier die Massnahmen kodieren
        // es kann der Parameter NewState weiterverarbeitet werden, muss aber nicht
    }
</SCRIPT>
</HEAD>
<BODY>

```



```

<SCRIPT FOR="ID_Player" EVENT="playStateChange(NewState)" LANGUAGE="JScript">
    EventHandler(NewState);
</SCRIPT>
<BODY>

```

Die im HEAD kodierte EventHandler-Routine

hat einen freien Funktionsbezeichner (im Beispiel den Bezeichner EventHandler)
muss Parameter **exakt** in der Schreibweise, die zum genau Event definiert ist, besitzen
(im Beispiel **NewState** (und nicht newstate etc.))

Das <SCRIPT FOR> </SCRIPT> aktiviert das Abfangen des Ereignisses und bindet den Eventhandler ein, der das Ereignis verwaltet.

Es ist übersichtlicher, den Code des Eventhandlers in eine Funktion zu legen. Eine direkte Script-Kodierung zwischen
<SCRIPT FOR ...> </SCRIPT> ist jedoch auch möglich.

Scriptsprache kann nur "JScript" sein !

Der gesamte Scriptcode des HTML-Dokumentes darf die vordefinierten Eventbezeichner und Event-Paramter nicht neu definieren, da sonst die Eventbehandlung scheitert.

Bsp.: Der Bezeichner es Eventhandlers darf nicht identisch sein mit dem Bezeichner eines Events.
Der Bezeichner des Events darf nicht re-definiert werden.

Eventbezeichner und Event-Parameter haben **globale Gültigkeit** im gesamten Dokument, das den Windows Media Player instanziiert hat.

5.2.4.5. Objekte, Collectionen und Events des Windows Media Player

Der Wert des ID-Attributes im OBJECT-Tag wird als Referenz auf das Objekt des Windows Media Player benutzt. Über dieses ID kann auf sämtliche Objekte und Collectionen zum Windows Media Player zugegriffen werden.

In der nachfolgenden Beschreibung zum Windows Media Player wird "ID_Player" synonym zum Begriff "Objekt des Windows Media Players" verwendet.

| | | |
|------------|------------------------|--|
| Beispiele: | aktuelles media Objekt | ID_Player.currentMedia |
| | aktuelles Media Item | ID_Player.controls.currentItem |
| | aktuelle Playliste | ID_Player.currentPlaylist |
| | beliebige Playliste | ID_Player.playlistCollection.item() |

5.2.4.5.1. Objekt des Windows Media Players (ID_Player)

Über ID_Player, also den Zeiger auf die Instanz, also auf das Objekt des Windows Media Players im HTML-Dokument, lassen sich grundlegende Eigenschaften und Methoden verwalten
Events auslösen und behandeln: Die Events aller Objekte und Collectionen landen auf der Ebenen der Instanz. laut ID_Player.

Die Programmierung von Skins (Layouts zum Player) wird nicht beschrieben.

Syntax:

ID_Player.eigenschaft
ID_Player.methode()

Eigenschaften:

Nicht für Skins verwendbar sind folgende Eigenschaften:

ID_Player.enableContextMenu
ID_Player.enabled
ID_Player.fullScreen
ID_Player.uiMode

| | |
|--------------------|---|
| .cdromCollection | Zeiger auf Collection cdromCollection |
| .closedCaption | Zeiger auf Objekt closedCaption |
| .controls | Zeiger auf Objekt controls |
| | Hinweis: aktuelles media Objekt ID_Player.currentMedia |
| | aktuelles Media Item ID_Player.controls.currentItem |
| | aktuelle Playliste ID_Player.currentPlaylist |
| | Playliste ID_Player.playlistCollection.item() |
| .currentMarker | Nummer des aktuellen Marker in der Media-Datei |
| | Marker als aktuellen Marker setzen |
| | Anzahl der Marker in der Media-Datei laut ID_Player.currentMedia.markerCount |
| | Marker dient der Wiedergabe der Media-Datei ab diesem Punkt in der Media-Datei |
| .currentPlaylist | Zeiger auf Objekt currentPlaylist |
| | Hinweis: aktuelles media Objekt ID_Player.currentMedia |
| | aktuelles Media Item ID_Player.controls.currentItem |
| | aktuelle Playliste ID_Player.currentPlaylist |
| | Playliste ID_Player.playlistCollection.item() |
| .enableContextMenu | Kontextmenü ein bzw. aus bei rechte Maus-Click |
| .enabled | Windows Media Player-Control-Elemente ein bzw. aus (true bzw. false) |
| .error | Zeiger auf Objekt error |
| .fullScreen | Vollbildmodus bei Wiedergabe eines Video ein bzw. aus (true bzw. false) |
| .isOnline | prüfen ob Client on- bzw. offline zum Netzwerk ist (z.B Internet) (true bzw. false) |



| | |
|---------------------|--|
| .mediaCollection | <p>Zeiger auf Collection mediaCollection</p> <p>Hinweis: aktuelles media Objekt ID_Player.currentMedia
 aktuelles Media Item ID_Player.controls.currentItem
 aktuelle Playliste ID_Player.currentPlaylist
 Playliste ID_Player.playlistCollection.item()</p> |
| .network | Zeiger auf Objekt network |
| .openState | aktueller Status des Players bezüglich Playliste, Media-Datei, Codec, Lizenz, Individualisierung |
| .playlistCollection | <p>Zeiger auf Collection playlistCollection</p> <p>Hinweis: aktuelles media Objekt ID_Player.currentMedia
 aktuelles Media Item ID_Player.controls.currentItem
 aktuelle Playliste ID_Player.currentPlaylist
 Playliste ID_Player.playlistCollection.item()</p> |
| .playState | <p>Status der Wiedergabe der aktuellen Media-Datei (Status des Players bei Wiedergabe)</p> <p>Media-Datei muss bestimmte Wiedergabe-Möglichkeiten unterstützen, damit deren Status angezeigt wird</p> <p>Download und Wiedergabe parallel möglich bei Dateien mit Suffix</p> <p>*.ASF
 *.AVI
 *.MP3
 *.MPEG
 *.WAV
 *.WM
 *.WMA
 *.WMV</p> |
| | <p>Integer</p> <p>0 undefined
 1 Wiedergabe ist gestoppt
 2 Wiedergabe pausiert
 3 Wiedergabe erfolgt fortlaufend
 4 Wiedergabe mit schnellem Vorlauf
 5 Wiedergabe mit schnellem Rücklauf
 8 Wiedergabe ist komplett und geendet
 10 Wiedergabe kann beginnen, da Player bereit dafür ist</p> |
| .settings | Zeiger auf Objekt settings |
| .status | interner und laufender Status des Players |
| .stretchToFit | <p>Playerfenster bei Media-Datei als Video in der Ansicht automatisch auf Videogröße ausdehnen ein bzw. aus</p> <p>sinnvoll, wenn Video größere Dimension haben könnte als das Layout des Players (Player-Fenster)</p> <p>true, so Playerfenster automatisch anpassen auf Videogröße
 false Default
 Playerfenster nicht automatisch anpassen auf Videogröße</p> |
| .uiMode | <p>Umfang der anzuzeigenden Control-Elemente des Players aus der Menge aller Control-Elemente zum aktuellen Media-Datei-Typ</p> <p>Control-Elemente kann der User zur Steuerung der Wiedergabe benutzen</p> <p>Control-Elemente nur sichtbar, wenn der Player sichtbar ist</p> <p>falls im OBJECT-Tag das HEIGHT-Attribut kodiert wurde, dann mit Wert > 40 Pixel</p> <p>String</p> <p>"none" keine Control-Elemente anzeigen
 User kann nichts steuern
 Achtung: Im OBJECT-Tag stets HEIGHT=0 und WIDTH=0 setzen</p> <p>"mini" angezeigt werden nur die Elemente
 Pause-Taste
 Stop-Taste
 Mute-Taste
 Lautstärkeregelung</p> <p>"mini" angezeigt werden alle Elemente</p> |
| .URL | <p>Url der Media-Datei, die wiedergegeben werden soll</p> <p>Media-Datei wird geladen und zum aktuellen Medium (aktuelles media Objekt)</p> <p>Autostart der Wiedergabe: laut ID_Player.settings.autoStart</p> <p>wenn Autostart erlaubt:</p> <p>Wiedergabe sofort gestartet bei Zuweisung von Werten zu
 ID_Player.URL
 ID_Player.currentMedia</p> <p>wenn Autostart nicht erlaubt ist:</p> <p>Wiedergabe nach Zuweisung von Werten zu
 ID_Player.URL
 ID_Player.currentMedia</p> <p>durch Aufruf von ID_Player.controls.play()</p> <p>Hinweis: Zuweisung zu
 ID_Player.URL
 ID_Player.currentMedia</p> |



| | |
|--|--|
| | setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht |
| .versionInfo | siehe ID_Player.currentMedia
Version der instanziierten Windows Media Player |
| Methoden: | |
| .close() | schliesst den Player im HTML-Dokument |
| .launchURL() | eine Media-Datei an den Standardbrowser im System schicken |
| Events: | |
| automatische Fehleranzeige ein- bzw. abschaltbar per | ID_Player.settings.enableErrorDialogs |
| buffering | erzeugt, wenn das Puffern von Media-Daten startet bzw. endet
Syntax:
buffering(Start)

Start true, so Puffern starten
false, so Puffern stoppen |
| | Hinweis: Download und Wiedergabe parallel möglich bei Dateien mit Suffix
*.ASF
*.AVI
*.MP3
*.MPEG
*.WAV
*.WM
*.WMA
*.WMV |
| currentItemChange | erzeugt, wenn ID_Player.controls.currentItem sich ändert
Hinweis: aktuelles media Objekt ID_Player.currentMedia
aktuelles Media Item ID_Player.controls.currentItem
aktuelle Playliste ID_Player.currentPlaylist
Playliste ID_Player.playlistCollection.item()
Syntax:
currentItemChange() |
| currentPlaylistChange | erzeugt, wenn ID_Player.currentPlaylist sich ändert
Hinweis: aktuelles media Objekt ID_Player.currentMedia
aktuelles Media Item ID_Player.controls.currentItem
aktuelle Playliste ID_Player.currentPlaylist
Playliste ID_Player.playlistCollection.item()
Syntax:
currentPlaylistChange(change)

change Integer
0 undefiniert
1 Playliste ist leer
2 Playlisten-Info wurde geändert
3 Playlisten-Eintrag wurde verschoben
4 Playlisten-Eintrag wurde gelöscht
5 Playliste wurde erweitert durch einfügen
eines Playlisten-Eintrages
6 Playliste wurde erweitert durch anhängen
eines Playlisten-Eintrages
8 Playliste-Name wurde geändert |
| error | erzeugt, wenn irgendein Fehler auftritt
Syntax:
error() |
| markerHit | erzeugt, wenn ein Marker erreicht wurde
Syntax:
markerHit(MarkerNum)

MarkerNum Integer
Nummer des Markers
ab 1 |
| mediaChange | erzeugt, wenn ein media Objekt bzw. Media Item sich ändert
Hinweis: aktuelles media Objekt ID_Player.currentMedia
aktuelles Media Item ID_Player.controls.currentItem
aktuelle Playliste ID_Player.currentPlaylist
Playliste ID_Player.playlistCollection.item()
Syntax:
mediaChange(Item) |



| | | | |
|-----------------------|---|---|---|
| | Item | Zeiger auf media Objekt bzw. Media Item | |
| mediaCollectionChange | erzeugt, wenn ein Collection mediaCollection sich ändert | | |
| Hinweis: | aktuelles media Objekt | ID_Player.currentMedia | |
| | aktuelles Media Item | ID_Player.controls.currentItem | |
| | aktuelle Playliste | ID_Player.currentPlaylist | |
| | Playliste | ID_Player.playlistCollection.item() | |
| Syntax: | mediaCollectionChange() | | |
| modeChange | erzeugt, wenn sich der Modus der Wiedergabe von Tracks ändert (Modus des Players) | | |
| | aktuellen Modus ermitteln per ID_Player.settings.getMode() | | |
| Syntax: | modeChange(ModeName, NewValue) | | |
| | ModeName | String | |
| | | "shuffle" | Tracks ab nächsten Track in Zufallsreihenfolge wiedergeben |
| | | "loop" | aktueller Track wird endlos wiedergegeben |
| | NewValue | true, false, | so Modus laut ModeName ist aktiv
so Modus laut ModeName ist nicht aktiv |
| openStateChange | erzeugt, wenn sich ID_Player.openState ändert | | |
| Syntax: | openStateChange(NewState) | | |
| | NewState | Integer | |
| | | 0 | undefined |
| | | 1 | eine neue Playliste wird gerade geladen (auch bei Netzwerk) |
| | | 2 | Player sucht Playliste
Playliste kann sein
lokal oder im Netzwerk
Datenbank oder Textdatei |
| | | 3 | Player verbindet sich mit der Playliste (auch bei Netzwerk) |
| | | 4 | Playliste wurde gefunden und es wird nun auf sie zugegriffen (auch bei Netzwerk) |
| | | 5 | Playliste wurde gefunden und wird nun ausgelesen (auch bei Netzwerk) |
| | | 6 | Playliste ist offen (auch bei Netzwerk) |
| | | 7 | neue Playliste wurde zur aktuellen Playliste |
| | | 8 | eine neue Media-Datei wird gerade geladen (auch bei Netzwerk) |
| | | 9 | Player sucht Media-Datei
Media-Datei kann sein
lokal oder im Netzwerk |
| | | 10 | Player verbindet sich mit Server, der die Media-Datei hat (nur bei Netzwerk) |
| | | 11 | Media-Datei wurde gefunden und es wird nun auf sie zugegriffen (auch bei Netzwerk) |
| | | 12 | Media-Datei wurde gefunden und sie wird nun geöffnet (auch bei Netzwerk) |
| | | 13 | Media-Datei ist offen (auch bei Netzwerk) |
| | | 14 | Codec-Ermittlung wird gestartet |
| | | 15 | Codec-Ermittlung ist komplett und beendet
Codec ist gültig |
| | | 16 | Lizenz-Ermittlung wird gestartet
Lizenz bei DRM protected Medium |
| | | 17 | Lizenz-Ermittlung ist komplett und beendet
Lizenz ist gültig |
| | | 18 | Individualisierung wird gestartet
DRM-Individualisierung |
| | | 19 | Individualisierung ist komplett und beendet |
| | | 20 | Player wartet auf ein Medium |
| | | 21 | Medium-Datei hat unbekannten Typ |
| playlistChange | erzeugt, wenn sich eine Playliste ändert | | |
| Syntax: | | | |



playListChange(Playlist, change)

Playlist Zeiger auf Playliste (kann die aktuelle Playliste sein, muss nicht)

change Integer

- 0 undefiniert
- 1 Playliste ist leer
- 2 Playlisten-Info wurde geändert
- 3 Playlisten-Eintrag wurde verschoben
- 4 Playlisten-Eintrag wurde gelöscht
- 5 Playliste wurde erweitert durch Einfügen eines Playlisten-Eintrages
- 6 Playliste wurde erweitert durch Anhängen eines Playlisten-Eintrages
- 8 Playliste-Name wurde geändert

playStateChange
(Wiedergabe)

erzeugt, wenn sich der Status der Wiedergabe der aktuellen Media-Datei (Status des Players bei

ändert

Media-Datei muss bestimmte Wiedergabe-Möglichkeiten unterstützen, damit deren Status angezeigt wird

Download und Wiedergabe parallel möglich bei Dateien mit Suffix

*.ASF

*.AVI

*.MP3

*.MPEG

*.WAV

*.WM

*.WMA

*.WMV

Syntax:

playStateChange(NewState)

NewState Integer

- 0 undefined
- 1 Wiedergabe ist gestoppt
- 2 Wiedergabe pausiert
- 3 Wiedergabe erfolgt fortlaufend
- 4 Wiedergabe mit schnellem Vorlauf
- 5 Wiedergabe mit schnellem Rücklauf
- 6 Media-Datei wird vom Server gelesen und gepuffert
Download und Wiedergabe parallel möglich bei
Dateien mit Suffix
 - *.ASF
 - *.AVI
 - *.MP3
 - *.MPEG
 - *.WAV
 - *.WM
 - *.WMA
 - *.WMV
- 7 Player wartet auf Antwort vom Server für
Beginn einer Sitzung
- 8 Wiedergabe ist komplett und beendet
- 9 es wird gerade eine neue Media-Datei vorbereitet
für Wiedergabe
- 10 Wiedergabe kann beginnen, da Player bereit dafür ist

positionChange

erzeugt, wenn die aktuelle Position im Medium eine andere ist als eine Vergleichsposition

Syntax:

positionChange(oldPosition, newPosition)

oldPosition

Integer

newPosition

Integer

scriptCommand

erzeugt, wenn ein synchronisiertes Kommando oder eine URL aus einer Media-Datei erkannt wurde
z.B. aus einer *.asf-Datei

Die Scriptkommandos der Datei sind Teil des Media-Datenstromes und steuern während der
Medium-Wiedergabe den Windows Media Player.

Desweiteren kann zusätzlich Javascript oder VBScript und HTML eingebunden werden für eine
individuelle Animation zum Player.

ASF-Datei ist Text-Datei, die je 2 Zeilen für Scriptkommandos besitzt:

z.B.



```
scType="URL"
Param=http://www.test.de/start.htm&&frame_name
```

mit frame_name als Name eines Frames in start.htm

Die erste Zeile ist der Kommandotyp also **scType**="kette_in_grossbuchstaben"

Kodierung von " " ist **zwingend**

Die zweite Zeile ist der Kommandowert also **Param** = kette_mit_freiem_inhalt

keine Kodierung von " "

Syntax:

```
scriptCommand(scType, Param)
```

scType String

Typ des Script-Kommandos

anhand Typ weiss der Player, wie der Wert laut Param zu

verarbeiten ist

z.B.

"URL"

Wert laut Param ist Url eines HTML-Dokumentes
bewirkt sofortigen Laden des Dokumentes in den Standard-Webbrowser, wenn player.settings.invokeURLs auf true steht (sonst kein laden)

ID_Player.settings.baseUrl wird ignoriert

Standardname des Frame laut

ID_Player.settings.defaultFrame

"FILENAME"

Wert laut Param ist Url einer Media-Datei

"TEXT"

Wert laut Param ist Caption (Text), der angezeigt wird

Param Wert des Scriptkommandos

z.B. Wert ist eine Url, wenn scType="URL"

Beispiel:

```
<SCRIPT FOR="ID_Player"
  EVENT="ScriptCommand(scType, Param)"
  LANGUAGE="JScript"
>
  if (scType.toLowerCase() == "url")
  {document.all.zeiger_auf_objekt.innerHTML = Param; }
</SCRIPT>
```

Hinweis: zeiger_auf_objekt laut ID-Attribut des HTML-Elementes

Die Zuweisung auf .innerHTML löst sofortiges Parsen aus.

scType und Param sind Schlüsselwörter zum Event ScriptCommand

statusChange erzeugt, wenn ID_Player.status sich verändert

Syntax:

```
statusChange()
```

5.2.4.5.2. ID_Player.cdromCollection Collection

Collection zur Verwaltung von CD-Laufwerken.

Ein Element der Collection ist genau ein Objekt, das **genau** 1 CD-Laufwerk verwaltet.

Syntax:

Collection:

ID_Player.cdromCollection.eigenschaft

ID_Player.cdromCollection.methode()

Element (CD-Laufwerk):

ohne CD-Laufwerksbuchstabe:

```
[ var Zeiger = ] ID_Player.cdromCollection.item(index)
```

index

Index in der Collection cdromCollection

Integer

ab 0



mit CD-Laufwerksbuchstabe:

```
[ var Zeiger = ] ID_Player.cdromCollection.getByDriveSpecifler(Kette)
```

Kette String
Laufwerksbuchstabe

Eigenschaften:

.count Anzahl der im System verfügbaren CD-Laufwerke
Anzahl der Elemente in der Collection
Achtung: **nicht** .length
.driveSpecifler liefert Laufwerksbuchstabe des CD-Laufwerkes als Element der Collection
.playlist Zeiger auf Playliste auf einer CD im CD-Laufwerk
aktuelle Playliste: siehe Objekt ID_Player.currentPlaylist

Methoden:

.eject() CD-Laufwerk als Element der Collection öffnen
ohne Laufwerksbuchstabe
Achtung: CD darf nicht gerade wiedergegeben werden, also vorher ID_Player.playState prüfen
.item() Zeiger auf ein CD-Laufwerk als Element der Collection liefern:
ohne Laufwerksbuchstabe
.getByDriveSpecifler() Zeiger auf ein CD-Laufwerk als Element der Collection liefern:
mit Laufwerksbuchstabe

5.2.4.5.3. ID_Player.closedCaption Objekt

dient zum Einschliessen von Captions (Text) in ein media Objekt
siehe auch ID_Player und dort Event scriptCommand

Syntax:

ID_Player.closedCaption.eigenschaft

Eigenschaften:

.captioningID Name des Frame oder des Control, in denen das Caption angezeigt wird
.SAMIFileName Name der Synchronized Accessible Media Interchange (SAMI)-Datei, die Informationen zum Caption enthält
Suffix der Datei *.smi oder *.sami
.SAMILang Sprache der der Synchronized Accessible Media Interchange (SAMI)-Datei, die Informationen zum Caption enthält
Suffix der Datei *.smi oder *.sami
Sprache liegt in der Datei innerhalb <STYLE> .. </STYLE>
z.B. .ENUSCC für US-Englisch
Sprache ist alphanumerisch und muss mit Punkt beginnen
mehrere Sprachen möglich
.SAMISyle Style des Caption
Sprache liegt in der Datei innerhalb <STYLE> .. </STYLE>
ist Kette mit erstem Zeichen stets das Nummernkreuz #
Bsp. #BigFont

Methoden:

keine

5.2.4.5.4. ID_Player.controls Objekt

dient zur Steuerung der Wiedergabe der aktuellen Media-Datei / aktuellen Media-Item

Playliste: mindestens 2 Einträgen aus media Objekten
Einträge werden Media Item benannt

Hinweise: ID_Player.currentMedia referenziert das aktuell **wiedergegebene** media Objekt, das aus eine Playliste stammen kann (Media Item).

Autostart der Wiedergabe: laut ID_Player.settings.autoStart
wenn Autostart erlaubt:

Wiedergabe sofort gestartet bei Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia

wenn Autostart nicht erlaubt ist:

Wiedergabe nach Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia
durch Aufruf von ID_Player.controls.play()

Hinweis: Zuweisung zu ID_Player.URL
ID_Player.currentMedia
setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht

ID_Player.currentMedia kann automatisch auf null-Zeiger gesetzt werden, wenn per ID_Player.currentPlaylist.removeItem()
ein Playlist-Element aus der aktuellen Playliste entfernt wird:

Wird das zu entfernende Media Item gerade wiedergegeben, dann wird Wdergabe gestoppt und das nächste



Playlisten-Eement aktiviert.

Wenn nächstes Item nicht vorhanden, so Vorgänger vom entfernten Item aktiviert.

Wenn auch das nicht vorhanden ist, also aktuelle Playliste leer ist, dann wird ID_Player.currentMedia auf null-Zeiger gesetzt !

Media-Item ist ein spezielles media Objekt, das aus einer Playliste stammt und per ID_Player.currentPlaylist Objekt referenziert wird.

In der Playliste wird anstelle media Objekt der Begriff Media Item verwendet.

ID_Player.currentMedia.media Objekt umfasst also nicht die Eigenschaften und Methoden zu Media Item als aktuellem Playlisten-Eintrag.

Dafür ähneln sich die Objekte ID_Player.currentMedia und ID_Player.currentPlaylist:

Sie haben z.B. gemeinsame Methoden, die sich jedoch auf verschiedene Objekte beziehen.

Hinweise: media Objekt erzeugen aus geladenener Media-/Meta-Datei
media Objekt erzeugen aus geladenener Media-/Meta-Datei
Media-Datei laden und als media Objekt der Bibliothek hinzufügen
media Objekt aktuell setzen
aktuelles media Objekt
prüfen ob aktuelles media Objekt eine Media Item ist,
also einer Playliste angehört

ID_Player.URL
ID_Player.launchURL
ID_Player.mediaCollection.add()
ID_Player.currentMedia = zeiger_auf_media_objekt;
ID_Player.currentMedia
ID_Player.currentMedia.isMemberOf()

Playliste
Playliste mit Namen erzeugen
Playliste in die Media-Bibliothek importieren
Playliste aus der Media-Bibliothek entfernen
Playliste aus allen Elemente der Media-Bibliotheks erzeugen
Playliste auf einer CD im CD-Laufwerk
Playliste als aktuell setzen
aktuelle Playliste
media Objekt an das Ende der aktuellen Playliste anhängen und
damit dort einen Media-Item erzeugen

ID_Player.playlistCollection.item()
ID_Player.playlistCollection.newPlaylist()
ID_Player.playlistCollection.importPlaylist()
ID_Player.playlistCollection.remove()
ID_Player.mediaCollection.getAll()
ID_Player.cdromCollection.item().playlist
ID_Player.currentPlaylist = zeiger_auf_playliste;
ID_Player.currentPlaylist
ID_Player.currentPlaylist.appendItem()

Objekt zum CD-Laufwerk ohne CD-Laufwerksbuchstabe:
Objekt zum CD-Laufwerk mit CD-Laufwerksbuchstabe:

ID_Player.cdromCollection.item()
ID_Player.cdromCollection.getByDriveSpecifier()

aktuell per Player-Control manipulierbares media Objekt bzw. Media Item
Verfügbarkeit von Controls zu einer Media-Datei / Media-Item
nächsten Playlist-Eintrag (Media Item) vorwärts in der Playliste anwählen
Laden eines Playlist-Eintrages (Media Item) und Autostart der Wiedergabe
vorhergehenden Playlist-Eintrag (Media Item) rückwärts in der Playliste
anwählen

ID_Player.controls.currentItem
ID_Player.controls.isAvailable()
ID_Player.controls.next()
ID_Player.controls.playItem()

Anzahl der Wiedergaben (nicht der Wiederholungen)
Wiedergabe-Modus (zufall, endlos)
Lautstärke für Ton (überschreibt Windows-Lautstärke-Regelung
zum Mediumtyp)

ID_Player.controls.previous()
ID_Player.settings.playCount
ID_Player.settings.setMode()
ID_Player.settings.volume

Syntax:

ID_Player.controls.eigenschaft
ID_Player.controls.methode()

Eigenschaften:

.currentItem Zeiger auf aktuelles media Objekt in einer Playliste (Media Item)
media Objekt als aktuelles Element der Playliste setzen
.currentMarker Nummer des aktuellen Marker in der Media-Datei
Marker als aktuellen Marker setzen
Anzahl der Marker in der Media-Datei laut ID_Player.currentMedia.markerCount
Marker dient der Wiedergabe der Media-Datei ab diesem Punkt in der Media-Datei
.currentPosition aktuelle Wiedergabe-Position als Wert in Sekunden in der Media-Datei ab Beginn der Datei
aktuelle Wiedergabe-Position setzen
.currentPositionString aktuelle Wiedergabe-Position als String in der Media-Datei ab Beginn der Datei

Methoden:

.fastForward() schnelles Vorspulen während der Wiedergabe einer Media-Datei
Standardgeschwindigkeit des schnellen Vorspulens während der Wiedergabe ist
5 faches der normalen Wiedergabegeschwindigkeit
ansonsten per ID_Player.setting.rate einstellbar
ignoriert ID_Player.settings.rate
wird aufgehoben durch ID_Player.controls.play
ID_Player.controls.stop
hebt auf ID_Player.controls.fastReverse()
nicht möglich z.B. für Audio
Ermittlung, ob Medium schnelles Vorspulen bei Wiedergabe unterstützt, per
ID_Player.controls.isAvailable()
.fastReverse() schnelles Rückspulen während der Wiedergabe einer Media-Datei



| | | |
|----------------|--|--|
| | Standardgeschwindigkeit des schnellen Rückspulens während der Wiedergabe ist
5 faches der normalen Wiedergabegeschwindigkeit
ansonsten per ID_Player.setting.rate einstellbar
nur innerhalb eines Track
ignoriert | ID_Player.settings.playCount
ID_Player.settings.rate |
| | wird aufgehoben durch | ID_Player.controls.fastForward()
ID_Player.controls.play
ID_Player.controls.stop |
| | nicht möglich z.B. für Audio
Ermittlung, ob Medium schnelles Rückspulen bei Wiedergabe unterstützt, per | ID_Player.controls.isAvailable() |
| .isAvailable() | Verfügbarkeit von Controls zu einer Media-Datei | |
| .next() | nächsten Eintrag (Media Item) vorwärts in der Playliste anwählen:
Wenn bereits LETZTER Eintrag der Playliste erreicht wurde und dann .next() aktiviert wird,
dann wird der ERSTE Eintrag in der Playliste eingestellt. | |
| .pause() | Medium muss Playliste unterstützen
pausieren der Wiedergabe des aktuellen media Objektes bzw. Media Items
siehe auch | ID_Player.controls.play()
ID_Player.controls.playItem()
ID_Player.controls.stop() |
| .play() | Wiedergabe des aktuellen media Objektes bzw. Media Item starten
oder wenn aktuelles Media bzw. Media Item bereits pausiert, so wird Pause aufgehoben
siehe auch | ID_Player.controls.playItem()
ID_Player.controls.pause()
ID_Player.controls.stop() |
| .playItem() | Laden eines Playlist-Eintrages (Media Item) und Autostart der Wiedergabe
ID_Player.setting.autoStart wird ignoriert
Hinweis: beliebige Media Datei laden und wiedergeben per
per ID_Player.url
und danach ID_Player.controls.play()
siehe auch | ID_Player.controls.play()
ID_Player.controls.pause()
ID_Player.controls.stop() |
| .previous() | vorhergehenden Eintrag (Media Item) rückwärts in der Playliste anwählen:
Wenn bereits ERSTER Eintrag der Playliste erreicht wurde und dann .previous() aktiviert wird,
dann wird der LETZTE Eintrag in der Playliste eingestellt. | |
| .stop() | Medium muss Playliste unterstützen
Wiedergabe des aktuellen media Objektes bzw. Media Item (aktuelles media Objekt der Playliste) beenden
oder wenn aktuelles Media bzw. Media Item bereits pausiert, so wird Pause aufgehoben
Achtung: Die Systemressourcen zur Media-Datei werden freigegeben !
bei Track: automatisch immer auf Trackanfang gesetzt
siehe auch | ID_Player.controls.play()
ID_Player.controls.playItem()
ID_Player.controls.pause() |

5.2.4.5.5. ID_Player.currentMedia Objekt

ID_Player.currentMedia referenziert das aktuell **wiedergegebene** media Objekt, das aus eine Playliste stammen kann (Media Item).

Autostart der Wiedergabe: laut ID_Player.settings.autoStart

wenn Autostart erlaubt:

Wiedergabe sofort gestartet bei Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia

wenn Autostart nicht erlaubt ist:

Wiedergabe nach Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia
durch Aufruf von ID_Player.controls.play()

Hinweis: Zuweisung zu

ID_Player.URL
ID_Player.currentMedia

setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht

ID_Player.currentMedia kann automatisch auf null-Zeiger gesetzt werden, wenn per ID_Player.currentPlaylist.removeItem()
ein Playlist-Element aus der aktuellen Playliste entfernt wird:

Wird das zu entfernende Media Item gerade wiedergegeben, dann wird Wdergabe gestoppt und das
nächste Playlisten-Eement aktiviert.

Wenn nächstes Item nicht vorhanden, so Vorgänger vom entfernten Item aktiviert.

Wenn auch das nicht vorhanden ist, also aktuelle Playliste leer ist, dann wird ID_Player.currentMedia auf null-Zeiger
gesetzt !

Hinweise: Media-Item ist ein spezielles media Objekt, das aus einer Playliste stammt und per ID_Player.currentPlaylist Objekt referenziert



wird.

In der Playliste wird anstelle media Objekt der Begriff Media Item verwendet.

ID_Player.currentMedia.media Objekt umfasst also nicht die Eigenschaften und Methoden zu Media Item als aktuellem Playlisten-Eintrag.

Dafür ähneln sich die Objekte ID_Player.currentMedia und ID_Player.currentPlaylist: Sie haben z.B. gemeinsame Methoden, die sich jedoch auf verschiedene Objekte beziehen.

Hinweise: media Objekt erzeugen aus geladenener Media-/Meta-Datei
media Objekt erzeugen aus geladenener Media-/Meta-Datei
Media-Datei laden und als media Objekt der Bibliothek hinzufügen
media Objekt aktuell setzen
aktuelles media Objekt
prüfen ob aktuelles media Objekt eine Media Item ist,
also einer Playliste angehört

Playliste
Playliste mit Namen erzeugen
Playliste in die Media-Bibliothek importieren
Playliste aus der Media-Bibliothek entfernen
Playliste aus allen Elemente der Media-Bibliotheks erzeugen
Playliste auf einer CD im CD-Laufwerk
Playliste als aktuell setzen
aktuelle Playliste
media Objekt an das Ende der aktuellen Playliste anhängen und
damit dort einen Media-Item erzeugen

Objekt zum CD-Laufwerk ohne CD-Laufwerksbuchstabe:
Objekt zum CD-Laufwerk mit CD-Laufwerksbuchstabe:

aktuell per Player-Control manipulierbares media Objekt bzw. Media Item
Verfügbarkeit von Controls zu einer Media-Datei / Media-Item
nächsten Playlist-Eintrag (Media Item) vorwärts in der Playliste anwählen
Laden eines Playlist-Eintrages (Media Item) und Autostart der Wiedergabe
vorhergehenden Playlist-Eintrag (Media Item) rückwärts in der Playliste
anwählen
Anzahl der Wiedergaben (nicht der Wiederholungen)
Wiedergabe-Modus (zufall, endlos)
Lautstärke für Ton (überschreibt Windows-Lautstärke-Regelung
zum Mediumtyp)

ID_Player.URL
ID_Player.launchURL
ID_Player.mediaCollection.add()
ID_Player.currentMedia = zeiger_auf_media_objekt;
ID_Player.currentMedia

ID_Player.currentMedia.isMemberOf()

ID_Player.playlistCollection.item()
ID_Player.playlistCollection.newPlaylist()
ID_Player.playlistCollection.importPlaylist()
ID_Player.playlistCollection.remove()
ID_Player.mediaCollection.getAll()
ID_Player.cdromCollection.item().playlist
ID_Player.currentPlaylist = zeiger_auf_playlist;
ID_Player.currentPlaylist

ID_Player.currentPlaylist.appendItem()

ID_Player.cdromCollection.item()
ID_Player.cdromCollection.getByDriveSpecifier()

ID_Player.controls.currentItem
ID_Player.controls.isAvailable()
ID_Player.controls.next()
ID_Player.controls.playItem()

ID_Player.controls.previous()
ID_Player.settings.playCount
ID_Player.settings.setMode()

ID_Player.settings.volume

Syntax:

ID_Player.currentMedia.eigenschaft
ID_Player.currentMedia.methode()

Beispiel:

```
function Ermitteln()
{
    var Kette;
    var AktuellesAttribut_Name;
    var AktuellesAttribut_Wert;

    // Media-Bibliothek
    var MediaBibliothek = ID_Player.mediaCollection;

    // Medium hinzufügen
    var Medium = MediaBibliothek.add("test.mid");

    // Medium als aktuelles Medium setzen durch Zuweisung
    ID_Player.currentMedia = Medium;

    // Zeiger auf aktuelles Medium
    var AktuellesMedium = Medium;

    // Attribute erzeugen
    AktuellesMedium.setItemInfo("Attribut1", "wert1");
    AktuellesMedium.setItemInfo("Attribut2", "wert2");

    var AktuellesMediumAnzahlAttribute = AktuellesMedium.attributeCount;

    Kette = "Anzahl der Attribute akt. Medium = " + AktuellesMediumAnzahlAttribute + "\r\r"
    + " Attribute sind :\r\r";
```



```

// alle Attribute des aktuellen Medium abklappern
for (var AttributeZahler = 0; AttributeZahler < AktuellesMediumAnzahlAttribute; ++ AttributeZahler)
{
    // Attributname
    AktuellesAttribut_Name = AktuellesMedium.getAttributeName(AttributeZahler);

    // Attributwert
    AktuellesAttribut_Wert = AktuellesMedium.getItemInfo(AktuellesAttribut_Name);

    Kette += AktuellesAttribut_Name + ": " + AktuellesAttribut_Wert + "\r";
}

ID_Div.innerHTML= Kette;
}

<DIV ID="ID_Div">
</DIV>

```

"ID_Player" ist der Wert des ID-Attributes im OBJECT-Tag

Eigenschaften:

| | |
|--------------------|--|
| .attributeCount | Anzahl der Attribute des aktuellen media Objektes |
| .duration | zeitliche Dauer in Sekunden (Integer) des aktuellen media Objektes |
| .durationString | zeitliche Dauer in Sekunden (String) des aktuellen media Objektes |
| .imageSourceHeight | Höhe in Pixel des aktuellen media Objektes |
| .imageSourceWidth | Breite in Pixel des aktuellen media Objektes |
| .markerCount | Anzahl der Marker in der Media-Datei des aktuellen media Objektes |
| | Media-Datei muss Marker unterstützen |
| | Marker dient der Wiedergabe der Media-Datei ab diesem Punkt in der Media-Datei |
| | Nummer des aktuellen Marker laut ID_Player.controls.currentMarker |
| .name | Name des aktuellen media Objektes |
| .sourceURL | Url des aktuellen media Objektes |

Methoden:

| | |
|----------------------|--|
| .getAttributeName() | Name eines Attributes des aktuellen media Objektes liefern |
| .getItemInfo() | Wert eines Attributes des aktuellen media Objektes liefern |
| .getItemInfoByAtom() | Wert eines Attributes des aktuellen media Objektes liefern |
| .getMarkerName() | Name eines Markers in der Media-Datei des aktuellen media Objektes liefern |
| | Media-Datei muss Marker unterstützen |
| | Marker dient der Wiedergabe der Media-Datei ab diesem Punkt in der Media-Datei |
| | Anzahl der Marker in der Media-Datei laut ID_Player.currentMedia.markerCount |
| | Nummer des aktuellen Marker laut ID_Player.controls.currentMarker |
| .getMarkerTime() | Zeitpunkt eines Markers in der Media-Datei des aktuellen media Objektes liefern |
| | Media-Datei muss Marker unterstützen |
| | Marker dient der Wiedergabe der Media-Datei ab diesem Punkt in der Media-Datei |
| | Anzahl der Marker in der Media-Datei laut ID_Player.currentMedia.markerCount |
| | Nummer des aktuellen Marker laut ID_Player.controls.currentMarker |
| .isIdentical() | auf Identität des aktuellen media Objektes mit einem anderen (nicht aktuellen) media Objektes prüfen |
| .isMemberOf() | prüfen ob aktuelles media Objekt eine Media Item ist, also einer Playliste angehört |
| .isReadOnlyItem() | Veränderbarkeit eines Attributes des aktuellen media Objektes |
| .setItemInfo() | Wert eines Attributes des aktuellen media Objektes setzen |
| | wenn Attribut nicht vorhanden, so erzeugt |
| | Veränderbarkeit eines Attributes laut ID_Player.currentMedia.isReadOnlyItem() |

5.2.4.5.6. ID_Player.currentPlaylist Objekt

referenziert die aktuelle Playliste der Media-Bibliothek

Hinweise: ID_Player.currentMedia referenziert das aktuell **wiedergegebene** media Objekt, das aus eine Playliste stammen kann (Media Item).

Autostart der Wiedergabe: laut ID_Player.settings.autoStart

wenn Autostart erlaubt:

Wiedergabe sofort gestartet bei Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia

wenn Autostart nicht erlaubt ist:

Wiedergabe nach Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia
durch Aufruf von ID_Player.controls.play()

Hinweis: Zuweisung zu

ID_Player.URL
ID_Player.currentMedia

setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht



ID_Player.currentMedia kann automatisch auf null-Zeiger gesetzt werden, wenn per ID_Player.currentPlaylist.removeItem() ein Playlist-Element aus der aktuellen Playliste entfernt wird:

Wird das zu entfernende Media Item gerade wiedergegeben, dann wird Wdergabe gestoppt und das nächste Playlisten-Element aktiviert.

Wenn nächstes Item nicht vorhanden, so Vorgänger vom entfernten Item aktiviert.

Wenn auch das nicht vorhanden ist, also aktuelle Playliste leer ist, dann wird ID_Player.currentMedia auf null-Zeiger gesetzt !

Media-Item ist ein spezielles media Objekt, das aus einer Playliste stammt und per ID_Player.currentPlaylist Objekt referenziert wird.

In der Playliste wird anstelle media Objekt der Begriff Media Item verwendet.

ID_Player.currentMedia.media Objekt umfasst also nicht die Eigenschaften und Methoden zu Media Item als aktuellem Playlisten-Eintrag.

Dafür ähneln sich die Objekte ID_Player.currentMedia und ID_Player.currentPlaylist: Sie haben z.B. gemeinsame Methoden, die sich jedoch auf verschiedene Objekte beziehen.

Hinweise: media Objekt erzeugen aus geladenener Media-/Meta-Datei
media Objekt erzeugen aus geladenener Media-/Meta-Datei
Media-Datei laden und als media Objekt der Bibliothek hinzufügen
media Objekt aktuell setzen
aktuelles media Objekt
prüfen ob aktuelles media Objekt eine Media Item ist,
also einer Playliste angehört

Playliste
Playliste mit Namen erzeugen
Playliste in die Media-Bibliothek importieren
Playliste aus der Media-Bibliothek entfernen
Playliste aus allen Elemente der Media-Bibliotheks erzeugen
Playliste auf einer CD im CD-Laufwerk
Playliste als aktuell setzen
aktuelle Playliste
media Objekt an das Ende der aktuellen Playliste anhängen und
damit dort einen Media-Item erzeugen

Objekt zum CD-Laufwerk ohne CD-Laufwerksbuchstabe:
Objekt zum CD-Laufwerk mit CD-Laufwerksbuchstabe:

aktuell per Player-Control manipulierbares media Objekt bzw. Media Item
Verfügbarkeit von Controls zu einer Media-Datei / Media-Item
nächsten Playlist-Eintrag (Media Item) vorwärts in der Playliste auswählen
Laden eines Playlist-Eintrages (Media Item) und Autostart der Wiedergabe
vorhergehenden Playlist-Eintrag (Media Item) rückwärts in der Playliste
auswählen
Anzahl der Wiedergaben (nicht der Wiederholungen)
Wiedergabe-Modus (zufall, endlos)
Lautstärke für Ton (überschreibt Windows-Lautstärke-Regelung
zum Mediumtyp)

ID_Player.URL
ID_Player.launchURL
ID_Player.mediaCollection.add()
ID_Player.currentMedia = zeiger_auf_media_objekt;
ID_Player.currentMedia

ID_Player.currentMedia.isMemberOf()

ID_Player.playlistCollection.item()
ID_Player.playlistCollection.newPlaylist()
ID_Player.playlistCollection.importPlaylist()
ID_Player.playlistCollection.remove()
ID_Player.mediaCollection.getAll()
ID_Player.cdromCollection.item().playlist
ID_Player.currentPlaylist = zeiger_auf_playlist;
ID_Player.currentPlaylist

ID_Player.currentPlaylist.appendItem()

ID_Player.cdromCollection.item()
ID_Player.cdromCollection.getByDriveSpecifier()

ID_Player.controls.currentItem
ID_Player.controls.isAvailable()
ID_Player.controls.next()
ID_Player.controls.playItem()

ID_Player.controls.previous()
ID_Player.settings.playCount
ID_Player.settings.setMode()

ID_Player.settings.volume

Syntax:

ID_Player.currentPlaylist.eigenschaft
ID_Player.currentPlaylist.methode()

Beispiel:

```
function Ermitteln()
{
    var Kette;
    var AktuellesAttribut_Name;
    var AktuellesAttribut_Wert;

    // aktuelle Playliste
    var AktuellePlayListe = ID_Player.currentPlaylist;

    // Playlist-Attribut erzeugen
    AktuellePlayListe.setItemInfo("PlaylistAttribut1", "changed");

    // Playlisteintrag mit Index 0 erweitern mit neuem Attribut
    AktuellePlayListe.item(0).setItemInfo("PlaylistAttribut2", "5");

    var AnzahlPlayListAttribute = AktuellePlayListe.attributeCount;
```



```

Kette = "Anzahl der Attribute der Playliste = " + AnzahlPlayListAttribute + "\r\r"
      + " Playlist-Attributes sind :\r\r";

// alle Attribute der Playliste abklappen
for (var AttributeZahler = 0; AttributeZahler < AnzahlPlayListAttribute; ++ AttributeZahler)
{
    // Attributname
    AktuellesAttribut_Name = AktuellePlayListe.attributeName(AttributeZahler);

    // Attributwert
    AktuellesAttribut_Wert = AktuellePlayListe.getItemInfo(AktuellesAttribut_Name);

    Kette += AktuellesAttribut_Name + ": " + AktuellesAttribut_Wert + "\r\r";
}

// alle Playlisten-Einträge abklappen
var AnzahlPlayListEintraege = AktuellePlayListe.count;

for ( var PlayListEintragZahler = 0;
      PlayListEintragZahler < AnzahlPlayListEintraege;
      ++ PlayListEintragZahler
    )
{
    var PlayListEintrag = AktuellePlayListe.item(PlayListEintragZahler);
    var PlayListEintrag_AnzahlAttribute = PlayListEintrag.attributeCount;

    Kette += "\rPlaylisteneintrag " + PlayListEintragZahler + "\r"
          + " mit " + PlayListEintrag_AnzahlAttribute + " Attributen:\r\r";

    // alle Attribute zum Playlisteintrag abklappen
    for ( var AttributeZahler = 0;
          AttributeZahler < PlayListEintrag_AnzahlAttribute;
          ++ AttributeZahler
        )
    {
        AktuellesAttribut_Name = PlayListEintrag.getAttributeName(AttributeZahler);
        AktuellesAttribut_Wert = PlayListEintrag.getItemInfo(AktuellesAttribut_Name);

        Kette += "Attribut " + AktuellesAttribut_Name
              + " mit Wert = " + AktuellesAttribut_Wert + "\r";
    }
}

ID_Div.innerHTML= Kette;
}

<DIV ID="ID_Div">
</DIV>

```

"ID_Player" ist der Wert des ID-Attributes im OBJECT-Tag

Eigenschaften:

| | |
|-----------------|---|
| .attributeCount | Anzahl der Attribute der aktuellen Playliste |
| .count | Anzahl der Media Item's in der aktuellen Playliste |
| .name | Name der aktuellen Playliste |
| | Jede Playliste hat in der Media-Bibliothek einen eindeutigen Namen. |

Methoden:

| | |
|------------------|--|
| .appendItem() | media Objekt an das Ende der aktuellen Playliste anhängen, also dort Media-Item erzeugen
aktuelle Playliste erweitern |
| .attributeName() | Name eines Attributes der aktuellen Playliste liefern |
| .getItemInfo() | Wert eines Attributes eines Media Item's in der aktuellen Playliste liefern |
| .insertItem() | media Objekt in die aktuelle Playliste einfügen, also dort Media Item erzeugen
aktuelle Playliste erweitern
vor dem Einfügen: Media Item am Einfügeplatz und seine Nachfolger rutschen automatisch 1 Platz
weiter runter in der Playliste |
| .isIdentical() | es wird ID_Player.currentPlaylist.count erhöht um Wert 1
auf Identität der aktuellen Playliste mit einer anderen (nicht aktuellen) Playliste prüfen |
| .item() | Zeiger auf ein Media Item in der aktuellen Playliste liefern
per Zeiger kann Media Item wie ein media Objekt behandelt werden |
| .moveItem() | Media Item in der aktuellen Playliste verschieben durch Indextausch
aktuelle Playliste verändern |
| .removeItem() | Media Item aus der aktuellen Playliste entfernen
aktuelle Playliste bereinigen
Wird das zu entfernende Media Item gerade wiedergegeben, dann wird Wdergabe gestoppt und das
nächste Playlisten-Eement aktiviert. |



Wenn nächstes Item nicht vorhanden, so Vorgänger vom entfernten Item aktiviert.

Wenn auch das nicht vorhanden ist, also aktuelle Playliste leer ist, dann wird

ID_Player.currentMedia auf null-Zeiger gesetzt !

.setItemInfo()

Wert eines Attributes eines Media-Item's in der aktuellen Playliste setzen

wenn Attribute nicht vorhanden, so erzeugt

es besteht kein Schreibschutz für Attribute

5.2.4.5.7. ID_Player.error Objekt

Fehlerinformationen zu media Objekten bzw. Media Item's

Fehlerinformationen werden in ihrer Reihenfolge des Auftretens **per Zeiger** in einer Queue als Zeigerfeld gesammelt.

Syntax:

ID_Player.error.eigenschaft

ID_Player.error.methode()

ID_Player.error.item().eigenschaft

ID_Player.error ist ein error Objekt und enthält genau 1 Fehlerinformation

ID_Player.error.item() ist ein Eintrag aus dem Feld (Queue) der Zeiger aller error Objekte

ist ein Zeiger auf ein error Objekt

Fehlerinformationen werden in ihrer Reihenfolge des Auftretens in einer Queue als Zeigerfeld gesammelt.

Eigenschaften:

.errorCount Anzahl der Fehlerinformationen in der Fehler-Queue

.errorDescription Fehlertext (String)

Methoden:

.clearErrorQueue() Fehler-Queue löschen, also alle Fehlerinformationen zu allen media Objekten / Media Item's löschen

.item() Zeiger auf Fehler (errorItem Objekt) in der Fehler-Queue liefern

.webHelp() Hilfe-Seite von Microsoft im Internet zum Windows Media Player aktivieren

Auf der Hilfe-Seite stehen Informationen zu Fehlern.

5.2.4.5.8. ID_Player.mediaCollection Collection

eingeschränkte Verwaltung der Media-Bibliothek bezüglich ihrer media Objekte (nicht Media Item's) und vorallem Playlisten

Media-Bibliothek wird ansonsten vom Player verwaltet und gespeichert

für Feld der Zeiger aller Playlisten der Media-Bibliothek: siehe playlistCollection Collection

Hinweise: ID_Player.currentMedia referenziert das aktuell **wiedergegebene** media Objekt, das aus eine Playliste stammen kann (Media Item).

Autostart der Wiedergabe: laut ID_Player.settings.autoStart

wenn Autostart erlaubt:

Wiedergabe sofort gestartet bei Zuweisung von Werten zu

ID_Player.URL

ID_Player.currentMedia

wenn Autostart nicht erlaubt ist:

Wiedergabe nach Zuweisung von Werten zu

ID_Player.URL

ID_Player.currentMedia

durch Aufruf von ID_Player.controls.play()

Hinweis: Zuweisung zu ID_Player.URL

ID_Player.currentMedia

setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht

ID_Player.currentMedia kann automatisch auf null-Zeiger gesetzt werden, wenn per ID_Player.currentPlaylist.removeItem()

ein Playlist-Element aus der aktuellen Playliste entfernt wird:

Wird das zu entfernende Media Item gerade wiedergegeben, dann wird Wdergabe gestoppt und das

nächste Playlisten-Element aktiviert.

Wenn nächstes Item nicht vorhanden, so Vorgänger vom entfernten Item aktiviert.

Wenn auch das nicht vorhanden ist, also aktuelle Playliste leer ist, dann wird ID_Player.currentMedia auf null-Zeiger gesetzt !

Media-Item ist ein spezielles media Objekt, das aus einer Playliste stammt und per ID_Player.currentPlaylist Objekt referenziert wird.

In der Playliste wird anstelle media Objekt der Begriff Media Item verwendet.

ID_Player.currentMedia.media Objekt umfasst also nicht die Eigenschaften und Methoden zu Media Item als aktuellem Playlisten-Eintrag.

Dafür ähneln sich die Objekte ID_Player.currentMedia und ID_Player.currentPlaylist: Sie haben z.B. gemeinsame Methoden, die sich jedoch auf verschiedene Objekte beziehen.

Hinweise: media Objekt erzeugen aus geladenener Media-/Meta-Datei

ID_Player.URL



media Objekt erzeugen aus geladenener Media-/Meta-Datei
 Media-Datei laden und als media Objekt der Bibliothek hinzufügen
 media Objekt aktuell setzen
 aktuelles media Objekt
 prüfen ob aktuelles media Objekt eine Media Item ist,
 also einer Playliste angehört

Playliste
 Playliste mit Namen erzeugen
 Playliste in die Media-Bibliothek importieren
 Playliste aus der Media-Bibliothek entfernen
 Playliste aus allen Elemente der Media-Bibliotheks erzeugen
 Playliste auf einer CD im CD-Laufwerk
 Playliste als aktuell setzen
 aktuelle Playliste
 media Objekt an das Ende der aktuellen Playliste anhängen und
 damit dort einen Media-Item erzeugen

Objekt zum CD-Laufwerk ohne CD-Laufwerksbuchstabe:
 Objekt zum CD-Laufwerk mit CD-Laufwerksbuchstabe:

aktuell per Player-Control manipulierbares media Objekt bzw. Media Item
 Verfügbarkeit von Controls zu einer Media-Datei / Media-Item
 nächsten Playlist-Eintrag (Media Item) vorwärts in der Playliste anwählen
 Laden eines Playlist-Eintrages (Media Item) und Autostart der Wiedergabe
 vorhergehenden Playlist-Eintrag (Media Item) rückwärts in der Playliste
 anwählen
 Anzahl der Wiedergaben (nicht der Wiederholungen)
 Wiedergabe-Modus (zufall, endlos)
 Lautstärke für Ton (überschreibt Windows-Lautstärke-Regelung
 zum Mediumtyp)

```
ID_Player.launchURL
ID_Player.mediaCollection.add()
ID_Player.currentMedia = zeiger_auf_media_objekt;
ID_Player.currentMedia

ID_Player.currentMedia.isMemberOf()

ID_Player.playlistCollection.item()
ID_Player.playlistCollection.newPlaylist()
ID_Player.playlistCollection.importPlaylist()
ID_Player.playlistCollection.remove()
ID_Player.mediaCollection.getAll()
ID_Player.cdromCollection.item().playlist
ID_Player.currentPlaylist = zeiger_auf_playlist;
ID_Player.currentPlaylist

ID_Player.currentPlaylist.appendItem()

ID_Player.cdromCollection.item()
ID_Player.cdromCollection.getByDriveSpecifier()

ID_Player.controls.currentItem
ID_Player.controls.isAvailable()
ID_Player.controls.next()
ID_Player.controls.playItem()

ID_Player.controls.previous()
ID_Player.settings.playCount
ID_Player.settings.setMode()

ID_Player.settings.volume
```

Syntax:

```
ID_Player.mediaCollection.methode()
ID_Player.mediaCollection.getAttributeStringCollection().eigenschaft
ID_Player.mediaCollection.getAttributeStringCollection().methode()
```

Beispiel:

```
function Ermitteln()
{
    var Kette;
    var AktuellesAttribut_Name;
    var AktuellesAttribut_Wert;

    // Media-Bibliothek
    var MediaBibliothek = ID_Player.mediaCollection;

    // Medium hinzufügen
    var Medium = MediaBibliothek.add("test.mid");

    // Medium als aktuelles Medium setzen durch Zuweisung
    ID_Player.currentMedia = Medium;

    // Zeiger auf aktuelles Medium
    var AktuellesMedium = Medium;

    // Attribute erzeugen
    AktuellesMedium.setItemInfo("Attribut1", "wert1");
    AktuellesMedium.setItemInfo("Attribut2", "wert2");

    var AktuellesMediumAnzahlAttribute = AktuellesMedium.attributeCount;

    Kette = "Anzahl der Attribute akt. Medium = " + AktuellesMediumAnzahlAttribute + "\r\r"
    + " Attribute sind :\r\r";

    // alle Attribute des aktuellen Medium abklappern
    for (var AttributeZahler = 0; AttributeZahler < AktuellesMediumAnzahlAttribute; ++ AttributeZahler)
    {
        // Attributname
        AktuellesAttribut_Name = AktuellesMedium.getAttributeName(AttributeZahler);

        // Attributwert
        AktuellesAttribut_Wert = AktuellesMedium.getItemInfo(AktuellesAttribut_Name);

        Kette += AktuellesAttribut_Name + ": " + AktuellesAttribut_Wert + "\r";
    }
}
```



```

    }

    ID_Div.innerHTML= Kette;
}

<DIV ID="ID_Div">
</DIV>

```

"ID_Player" ist der Wert des ID-Attributes im OBJECT-Tag

Eigenschaften:

.count Anzahl der Elemente im Zeichenkettenfeld laut ID_Player.mediaCollection.getAttributeStringCollection()

Methoden:

.add() Media-Datei laden und als media Objekt der Bibliothek hinzufügen und Zeiger liefern

.getAll() Playliste aus allen Elemente der Media-Bibliothek erzeugen und Zeiger liefern

.getAttributeStringCollection() je mehr Elemente in der Bibliothek, um so länger dauert die Abarbeitung der Methode
Zeichenkettenfeld aus allen Werte **eines** Attributes zu einem Media-Typ erzeugen
pro media Objekt vom Media-Typ wird ein Element in der Collection erzeugt
media Objekt nur dann verwendet, wenn auch Attribut mit ihm verbunden ist

Bsp.: für Attribut:
"Album"
"Author"

Bsp.: für Media-Typ
"Audio"
"Video"

.getByAlbum() Playliste aus all denjenigen Elementen der Media-Bibliothek erzeugen, die das Attribut "Album"
und darin einen **bestimmten** Wert besitzen, und Zeiger liefern

.getByAttribute() je mehr Elemente in der Bibliothek, um so länger dauert die Abarbeitung der Methode
Playliste aus all denjenigen Elementen der Media-Bibliothek erzeugen, die **ein** Attribut mit **bestimmten**
Wert besitzen, und Zeiger liefern

.getByAuthor() je mehr Elemente in der Bibliothek, um so länger dauert die Abarbeitung der Methode
Playliste aus all denjenigen Elementen der Media-Bibliothek erzeugen, die das Attribut "Author" und darin
einen **bestimmten** Wert besitzen, und Zeiger liefern

.getByGenre() je mehr Elemente in der Bibliothek, um so länger dauert die Abarbeitung der Methode
Playliste aus all denjenigen Elementen der Media-Bibliothek erzeugen, die das Attribut "Genre" und darin
einen **bestimmten** Wert besitzen, und Zeiger liefern

.getByName() je mehr Elemente in der Bibliothek, um so länger dauert die Abarbeitung der Methode
Playliste aus all denjenigen Elementen der Media-Bibliothek erzeugen, die das Attribut "Name" und darin
einen **bestimmten** Wert besitzen, und Zeiger liefern

.getMediaAtom() je mehr Elemente in der Bibliothek, um so länger dauert die Abarbeitung der Methode
Index eines Attributes liefern, also Art des Attributes

.isDeleted() prüfen ob Media Datei aus dem Datei-Pool der Bibliothek entfernt wurde
bei lokaler Media Datei: den Papierkorb des Systems prüfen
bei nicht-lokaler Datei: auf logische Verknüpfung prüfen
Media-Datei aus dem Datei-Pool der Media-Bibliothek entfernen:
ID_Player.mediaCollection.setDeleted()
Media-Datei aus ID_Player.mediaCollection. entfernen und optional aus der Datenbank der Namen:
ID_Player.mediaCollection.remove()

.item() Wert eines Elementes im Zeichenkettenfeld laut ID_Player.mediaCollection.getAttributeStringCollection()
ermitteln

.remove() media Objekt aus Media-Bibliothek, also aus ID_Player.mediaCollection, entfernen:
aus Datenbank der Namen der Objekte und **optional** aus dem Datei-Pool
Media-Datei aus dem Datei-Pool der Media-Bibliothek entfernen: ID_Player.mediaCollection.setDeleted()
Media-Datei aus dem Datei-Pool der Media-Bibliothek entfernen:
bei lokaler Media Datei: **immer** in den Papierkorb des Systems verschieben
bei nicht-lokaler Datei: logische Verknüpfung entfernen

5.2.4.5.9. ID_Player.network Objekt

Objekt für Netzwerkverbindung/Internetverbindung des Windows Media Player z.B. per Proxy

Syntax:

ID_Player.network.eigenschaft

ID_Player.network.methode()

Eigenschaften:

.bandWidth aktuelle Bandbreite (Bits pro Sekunde) der Media-Datei mit Datenstrom also Video

.bitRate aktuelle Bitrate der Media-Datei

.bufferingCount Anzahl der Zeitpuffer während der Wiedergabe des aktuellen media Objektes mit Datenstrom,
also Video

.bufferingProgress aktueller Prozentanteil der Pufferung während der Wiedergabe des aktuellen media Objektes
mit Datenstrom, also Video
Fortgeschrittenheit der Pufferung
100% Pufferung entspricht komplett gepuffert

.bufferingTime Anzahl der Millisekunden als Zeit für Pufferung vor der Wiedergabe der Media-Datei

.downloadProgress aktueller Prozentanteil des Downloads des aktuellen media Objektes von einem Webserver
Fortgeschrittenheit des Download
100% Download entspricht komplett geladen



| | |
|----------------------------|--|
| | Download und Wiedergabe parallel möglich bei Dateien mit Suffix |
| | *.ASF |
| | *.AVI |
| | *.MP3 |
| | *.MPEG |
| | *.WAV |
| | *.WM |
| | *.WMA |
| | *.WMV |
| .encodedFrameRate | Framerate (Frames pro Sekunden) der Media-Datei als Video
Framerate laut Hersteller des Video
nicht aktuelle Framerate des Video (siehe ID_Player.network.FrameRate) |
| .FrameRate | aktuelle Framerate (Frames pro Sekunden) der Media-Datei als Video
nicht Framerate laut Hersteller des Video (siehe ID_Player.network.encodedFrameRate) |
| .framesSkipped
Objektes | aktuelle Anzahl der bisher wiedergegebenen Frames ab Beginn der Wiedergabe des aktuellen media
als Video |
| .lostPackets | aktuelle Anzahl der bisher verlorengegangenen Datenpakete ab Beginn der Wiedergabe des aktuellen media
Objektes mit Datenstrom, also Video
Verlust an Datenpaketen nicht bei HTTP möglich
hängt von der Netzwerkverbindung ab |
| .maxBandwidth | maximale Bandbreite (Bits pro Sekunde) für die Wiedergaben der Media-Datei mit Datenstrom also Video
besonders bei einem Datenfluss verenden, der mit verschiedenen Bitraten ausgestattet ist
(MBS Multiple streams width different bit rates)
z.B. Herabsetzung der maximalen Bitrate wegen der schlechteren Netzwerkqualität, weil
Media-Datei mindestens eine höhere Bandbreite, als die Verbindung Server zum
Client (Player) physisch leisten kann |
| .receivedPackets | aktuelle Anzahl der bisher nicht verlorengegangenen Datenpakete ab Beginn der Wiedergabe des aktuellen
media Objektes mit Datenstrom, also Video
Verlust an Datenpaketen nicht bei HTTP möglich
hängt von der Netzwerkverbindung ab |
| .receptionQuality | aktuelle Prozentzahl der in den letzten 30 Sekunden nicht verlorengegangenen Datenpakete während der
Wiedergabe des aktuellen media Objektes mit Datenstrom, also Video,
gegenüber der Gesamtanzahl aller in den letzten 30 Sekunden von der Media-Datei abgesendeten
Datenpakete, also inklusive der verlorenen Datenpakete.
Verlust an Datenpaketen nicht bei HTTP möglich
hängt von der Netzwerkverbindung ab |
| .recoveredPackets | aktuelle Anzahl der bisher verlorengegangenen aber wiederhergestellten Datenpakete ab Beginn der
Wiedergabe des aktuellen media Objektes mit Datenstrom, also Video
Verlust an Datenpaketen nicht bei HTTP möglich
hängt von der Netzwerkverbindung ab |
| .sourceProtocol | Datentransport-Protokoll während der Wiedergabe des media Objektes |
| Methoden: | |
| .getProxyBypassForLocal() | prüfen, ob Proxyserver übergangen wird, wenn der Server als Media-Datei-Quelle im lokalen Netzwerk
liegt |
| .getProxyExceptionList() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Ausnahmeliste des Proxyservers liefern bezüglich
Computer
Domains
IP's
die den Proxyserver übergehen |
| .getProxyName() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Name des benutzten Proxyservers liefern |
| .getProxyPort() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Nummer des benutzten Ports des Proxyservers liefern |
| .getProxySettings() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Einstellungen des Proxyservers liefern |
| .setProxyBypassForLocal() | einstellen, ob Proxyserver übergangen wird, wenn der Server als Media-Datei-Quelle im lokalen Netzwerk
liegt |
| .setProxyExceptionList() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Ausnahmeliste des Proxyservers setzen bezüglich
Computer
Domains
IP's
die den Proxyserver übergehen |
| .setProxyName() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Name des benutzten Proxyservers setzen |
| .setProxyPort() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Nummer des benutzten Ports des Proxyservers setzen |
| .setProxySettings() | nur möglich, wenn ID_Player.network.getProxySettings() den Wert 2 liefert
Einstellungen des Proxyservers setzen |

5.2.4.5.10. ID_Player.playlistCollection Collection

Feld der Zeiger aller Playlisten der Media-Bibliothek

für Verwaltung der Media-Bibliothek bezüglich ihrer media Objekte (nicht Media Item's) und Playlisten: siehe mediaCollection Collection



Hinweise: ID_Player.currentMedia referenziert das aktuell **wiedergegebene** media Objekt, das aus eine Playliste stammen kann (Media Item).

Autostart der Wiedergabe: laut ID_Player.settings.autoStart

wenn Autostart erlaubt:

Wiedergabe sofort gestartet bei Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia

wenn Autostart nicht erlaubt ist:

Wiedergabe nach Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia
durch Aufruf von ID_Player.controls.play()

Hinweis: Zuweisung zu

ID_Player.URL
ID_Player.currentMedia
setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht

ID_Player.currentMedia kann automatisch auf null-Zeiger gesetzt werden, wenn per ID_Player.currentPlaylist.removeItem() ein Playlist-Element aus der aktuellen Playliste entfernt wird:

Wird das zu entfernende Media Item gerade wiedergegeben, dann wird Wdergabe gestoppt und das nächste Playlisten-Element aktiviert.

Wenn nächstes Item nicht vorhanden, so Vorgänger vom entfernten Item aktiviert.

Wenn auch das nicht vorhanden ist, also aktuelle Playliste leer ist, dann wird ID_Player.currentMedia auf null-Zeiger gesetzt !

Media-Item ist ein spezielles media Objekt, das aus einer Playliste stammt und per ID_Player.currentPlaylist Objekt referenziert wird.

In der Playliste wird anstelle media Objekt der Begriff Media Item verwendet.

ID_Player.currentMedia.media Objekt umfasst also nicht die Eigenschaften und Methoden zu Media Item als aktuellem Playlisten-Eintrag.

Dafür ähneln sich die Objekte ID_Player.currentMedia und ID_Player.currentPlaylist: Sie haben z.B. gemeinsame Methoden, die sich jedoch auf verschiedene Objekte beziehen.

Hinweise: media Objekt erzeugen aus geladenener Media-/Meta-Datei
media Objekt erzeugen aus geladenener Media-/Meta-Datei
Media-Datei laden und als media Objekt der Bibliothek hinzufügen
media Objekt aktuell setzen
aktuelles media Objekt
prüfen ob aktuelles media Objekt eine Media Item ist,
also einer Playliste angehört

ID_Player.URL
ID_Player.launchURL
ID_Player.mediaCollection.add()
ID_Player.currentMedia = zeiger_auf_media_objekt;
ID_Player.currentMedia

ID_Player.currentMedia.isMemberOf()

Playliste

Playliste mit Namen erzeugen

Playliste in die Media-Bibliothek importieren

Playliste aus der Media-Bibliothek entfernen

Playliste aus allen Elemente der Media-Bibliotheks erzeugen

Playliste auf einer CD im CD-Laufwerk

Playliste als aktuell setzen

aktuelle Playliste

media Objekt an das Ende der aktuellen Playliste anhängen und
damit dort einen Media-Item erzeugen

ID_Player.playlistCollection.item()
ID_Player.playlistCollection.newPlaylist()
ID_Player.playlistCollection.importPlaylist()
ID_Player.playlistCollection.remove()
ID_Player.mediaCollection.getAll()
ID_Player.cdromCollection.item().playlist
ID_Player.currentPlaylist = zeiger_auf_playlist;
ID_Player.currentPlaylist

ID_Player.currentPlaylist.appendItem()

Objekt zum CD-Laufwerk ohne CD-Laufwerksbuchstabe:

Objekt zum CD-Laufwerk mit CD-Laufwerksbuchstabe:

ID_Player.cdromCollection.item()
ID_Player.cdromCollection.getByDriveSpecifier()

aktuell per Player-Control manipulierbares media Objekt bzw. Media Item
Verfügbarkeit von Controls zu einer Media-Datei / Media-Item
nächsten Playlist-Eintrag (Media Item) vorwärts in der Playliste auswählen
Laden eines Playlist-Eintrages (Media Item) und Autostart der Wiedergabe
vorhergehenden Playlist-Eintrag (Media Item) rückwärts in der Playliste
auswählen

ID_Player.controls.currentItem
ID_Player.controls.isAvailable()
ID_Player.controls.next()
ID_Player.controls.playItem()

Anzahl der Wiedergaben (nicht der Wiederholungen)

Wiedergabe-Modus (zufall, endlos)

Lautstärke für Ton (überschreibt Windows-Lautstärke-Regelung
zum Mediumtyp)

ID_Player.controls.previous()
ID_Player.settings.playCount
ID_Player.settings.setMode()

ID_Player.settings.volume

Syntax:

ID_Player.playlistCollection.eigenschaft
ID_Player.playlistCollection.methode()



```
ID_Player.playlistCollection.getAll().eigenschaft
ID_Player.playlistCollection.getAll().methode()
```

Hinweis: ID_Player.playlistCollection.getAll() liefert Zeiger auf das Feld der Zeiger **aller** Playlisten in der Media-Bibliothek

Eigenschaften:

.count Anzahl der Playlisten in der Media-Bibliothek

Methoden:

```
.getAll() Zeiger auf Feld aller Playlisten in der Media-Bibliothek liefern
.getByName() Zeiger auf Feld aus maximal 1 Playliste aus der Media-Bibliothek liefern
jede Playliste hat einen eindeutigen Namen: Feld kann also maximal 1 Element besitzen
siehe auch .getAll()
.importPlaylist() Playliste in die Media-Bibliothek importieren
.isDeleted() prüfen ob Playliste im Papierkorb des Systems liegt
.item() Zeiger auf eine Playliste in der Media-Bibliothek
.newPlaylist() Playliste mit Name als Objekt instanzieren und Zeiger liefern, aber nicht in die Media-Bibliothek
importieren
Jede Playliste muss einen eindeutigen Namen haben:
Es wird Fehler erzeugt, wenn Name bereits in der Media-Bibliothek vorhanden ist.
.remove() Playliste aus der Media-Bibliothek entfernen
.setDeleted() Playliste aus der Media-Bibliothek in den Papierkorb des System verschieben
```

5.2.4.5.11. ID_Player.settings Objekt

Einstellungen des Players für die Wiedergabe des jeweiligen aktuellen media Objektes bzw. Media Item's

Einstellungen gelten für alle Wiedergaben

Änderung der Einstellungen der Wiedergabe **während aktiver** Wiedergabe theoretisch möglich

Viele Eigenschaften vom ID_Player.settings Objekt entsprechen den PARAM's im OBJECT-Tag.

Hinweise: ID_Player.currentMedia referenziert das aktuell **wiedergegebene** media Objekt, das aus eine Playliste stammen kann (Media Item).

Autostart der Wiedergabe: laut ID_Player.settings.autoStart

wenn Autostart erlaubt:

Wiedergabe sofort gestartet bei Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia

wenn Autostart nicht erlaubt ist:

Wiedergabe nach Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia
durch Aufruf von ID_Player.controls.play()

Hinweis: Zuweisung zu

ID_Player.URL
ID_Player.currentMedia

setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht

ID_Player.currentMedia kann automatisch auf null-Zeiger gesetzt werden, wenn per ID_Player.currentPlaylist.removeItem() ein Playlist-Element aus der aktuellen Playliste entfernt wird:

Wird das zu entfernende Media Item gerade wiedergegeben, dann wird Wdergabe gestoppt und das nächste Playlisten-Element aktiviert.

Wenn nächstes Item nicht vorhanden, so Vorgänger vom entfernten Item aktiviert.

Wenn auch das nicht vorhanden ist, also aktuelle Playliste leer ist, dann wird ID_Player.currentMedia auf null-Zeiger gesetzt !

Media-Item ist ein spezielles media Objekt, das aus einer Playliste stammt und per ID_Player.currentPlaylist Objekt referenziert wird.

In der Playliste wird anstelle media Objekt der Begriff Media Item verwendet.

ID_Player.currentMedia.media Objekt umfasst also nicht die Eigenschaften und Methoden zu Media Item als aktuellem Playlisten-Eintrag.

Dafür ähneln sich die Objekte ID_Player.currentMedia und ID_Player.currentPlaylist: Sie haben z.B. gemeinsame Methoden, die sich jedoch auf verschiedene Objekte beziehen.

Hinweise: media Objekt erzeugen aus geladenener Media-/Meta-Datei
media Objekt erzeugen aus geladenener Media-/Meta-Datei
Media-Datei laden und als media Objekt der Bibliothek hinzufügen
media Objekt aktuell setzen
aktuelles media Objekt
prüfen ob aktuelles media Objekt eine Media Item ist,
also einer Playliste angehört

```
ID_Player.URL
ID_Player.launchURL
ID_Player.mediaCollection.add()
ID_Player.currentMedia = zeiger_auf_media_objekt;
ID_Player.currentMedia
ID_Player.currentMedia.isMemberOf()
```



| | |
|--|---|
| Playliste | ID_Player.playlistCollection.item() |
| Playliste mit Namen erzeugen | ID_Player.playlistCollection.newPlaylist() |
| Playliste in die Media-Bibliothek importieren | ID_Player.playlistCollection.importPlaylist() |
| Playliste aus der Media-Bibliothek entfernen | ID_Player.playlistCollection.remove() |
| Playliste aus allen Elemente der Media-Bibliotheks erzeugen | ID_Player.mediaCollection.getAll() |
| Playliste auf einer CD im CD-Laufwerk | ID_Player.cdromCollection.item().playlist |
| Playliste als aktuell setzen | ID_Player.currentPlaylist = zeiger_auf_playliste; |
| aktuelle Playliste | ID_Player.currentPlaylist |
| media Objekt an das Ende der aktuellen Playliste anhängen und damit dort einen Media-Item erzeugen | ID_Player.currentPlaylist.appendItem() |
| Objekt zum CD-Laufwerk ohne CD-Laufwerksbuchstabe: | ID_Player.cdromCollection.item() |
| Objekt zum CD-Laufwerk mit CD-Laufwerksbuchstabe: | ID_Player.cdromCollection.getByDriveSpecifier() |
| aktuell per Player-Control manipulierbares media Objekt bzw. Media Item | ID_Player.controls.currentItem |
| Verfügbarkeit von Controls zu einer Media-Datei / Media-Item | ID_Player.controls.isAvailable() |
| nächsten Playlist-Eintrag (Media Item) vorwärts in der Playliste anwählen | ID_Player.controls.next() |
| Laden eines Playlist-Eintrages (Media Item) und Autostart der Wiedergabe | ID_Player.controls.playItem() |
| vorhergehenden Playlist-Eintrag (Media Item) rückwärts in der Playliste anwählen | ID_Player.controls.previous() |
| Anzahl der Wiedergaben (nicht der Wiederholungen) | ID_Player.settings.playCount |
| Wiedergabe-Modus (zufall, endlos) | ID_Player.settings.setMode() |
| Lautstärke für Ton (überschreibt Windows-Lautstärke-Regelung zum Mediumtyp) | ID_Player.settings.volume |

Syntax:

ID_Player.settings.eigenschaft
ID_Player.settings.methode()

Eigenschaften:

| | |
|---------------------|---|
| .autoStart | Autostart der Wiedergabe
wenn Autostart erlaubt:
Wiedergabe sofort gestartet bei Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia
wenn Autostart nicht erlaubt ist:
Wiedergabe nach Zuweisung von Werten zu
ID_Player.URL
ID_Player.currentMedia
durch Aufruf von ID_Player.controls.play()
Hinweis: Zuweisung zu ID_Player.URL
ID_Player.currentMedia
setzt Medium immer als aktuelles Medium, egal ob Autostart erlaubt oder nicht |
| .balance | Stereo-Balance (Kanalausrichtung)
-100 bis 100
Standard ist 0
-100 ganz links
100 ganz rechts
0 mittig |
| .baseURL | Basis-Pfad einer HTTP-Url
wird ignoriert von der Url-Angabe im Event ID_Player.scriptCommand (scType mit Wert "URL")
relativer Pfad:
"/" kodierbar, wenn nicht erstes Zeichen
erstes Zeichen darf nicht sein
"."
"
"
"
wenn ja, so wird es automatisch gelöscht
".." etc. nicht kodierbar |
| .defaultFrame | Standardname des Frame für Event ID_Player.scriptCommand, wenn scType mit Wert "URL", falls im Kommando-Wert kein anderer Frame kodiert wurde |
| .enableErrorDialogs | automatische Fehleranzeige ein/ aus (true/false) |
| .invokeURLs | Wirksamkeit der Url-Zuweisung an den Standard-Webbrowser aufgrund eines eintreffenden Events
ID_Player.scriptCommand mit scType auf "URL" und Param mit Wert als Kette mit der zuzuweisenden Url
Scriptkommando liegt z.B. in einer ASF-Datei
Achtung: Wenn Script zum Event kodiert wurde, so wird dieses Script vor der Url-Zuweisung abgearbeitet. Das Script ist frei programmierbar und kann z.B. eine Wertänderung von Param und oder ID_Player.settings.invokeURLs durchführen.
Danach wird erst die Url laut Wert von Param an den Webbrowser weitergeleitet, falls das ID_Player.settings.invokeURLs zulässt.
Es muss kein Script zum Event ID_Player.scriptCommand zu scType mit Wert "URL" kodiert sein.
true für weiterreichen
false für nicht weiterreichen |



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

5.3. Webspeech von Logox im Internet Explorer und Netscape

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

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

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

