

Einzelkurzbeschreibungen der Events des IE (teilweise mit Beispielen, Auswahl)

Einzelkurzbeschreibungen der Events des IE (teilweise mit Beispielen, Auswahl).....1

- onabort*2
- onactivate*2
- onafterprint*2
- onafterupdate*2
- onbeforeactivate*3
- onbeforecopy*3
- onbeforecut*3
- onbeforedeactivate*4
- onbeforeeditfocus*4
- onbeforepaste*4
- onbeforeprint*4
- onbeforeunload*4
- onbeforeupdate*5
- onbegin*5
- onblur*5
- onbounce*5
- oncellchange*6
- onchange*6
- onclick*6
- oncontentready*6
- oncontentsave*6
- oncontextmenu*7
- oncontrolselect*7
- oncopy*7
- oncut*7
- ondataavailable*8
- ondatasetchanged*8
- ondatasetcomplete*8
- ondblclick*8
- ondeactivate*8
- ondetach*8
- ondocumentready*9
- ondrag*9
- ondragend*9
- ondragenter*9
- ondragleave*9
- ondragover*9
- ondragstart*9
- ondrop*9
- onend*9
- onerror*10
- onerrorupdate*11
- onfilterchange*11
- onfinish*11
- onfocus*11
- onfocusin*12
- onfocusout*12
- onhelp*12
- onhide*12
- onkeydown*12
- onkeypress*12
- onkeyup*12
- onlayoutcomplete*13
- onload*13
- onlosecapture*13
- onmediacomplete*13
- onmediaerror*14
- onmousedown*14
- onmouseenter*14
- onmouseleave*15
- onmousemove*15
- onmouseout*15
- onmouseover*15
- onmouseup*15
- onmove*15
- onmoveend*16
- onmovestart*17
- onopenstatechange*18
- onoutofsync*18



onpaste.....18
onpause.....19
onplaystatechange19
onpropertychange20
onreadystatechange20
onrepeat.....20
onreset21
onresize.....22
onresizeend.....22
onresizestart22
onresume22
onreverse23
onrowenter23
onrowexit.....23
onrowsdelete.....23
onrowsinserted23
onsave.....23
onscroll.....24
onseek.....24
onselect.....25
onselectionchange25
onselectstart.....25
onshow.....25
onstart.....26
onstop26
onsubmit.....26
onsyncrestored26
ontimeerror26
ontrackchange.....26
onunload.....27
onURLFlip.....27

Events sind grundsätzlich im Kontext des Objektes zu sehen, das das Event wegen objektspezifischem Vorgang auslöst. Nachfolgend werden Events exemplarisch erklärt. Events werden anhand von Objekt window.event verwaltet: Das Objekt reicht sein ausgelöstes Event entlang der DOM-Hierarchie nach oben (bubble) zum window- und damit zum event-Objekt. Nicht alle Events dürfen bubble. Die Einbindung von Eventhandlern ist der Beschreibung zum Objekt window.event bzw. nachfolgenden Beispielen zu entnehmen

onabort

Abbruch des Download z.B. vom Image

onactivate

erzeugt wenn Objekt als aktives gesetzt wird (event.fromElement to the event.srcElement)
aktiv setzen bedingt nicht den Erhalt des Focus:
aber Aktivierung per Fokus-Methode möglich
wird immer direkt vor onload erzeugt
ab IE 5.5

Beispiel

```
<HTML>
<HEAD>
<SCRIPT>
function EventOnActivate()
{ID_Div.innerHTML += "onactivate erkannt";}

function EventOnLoad()
{ID_Div.innerHTML += "onload erkannt";}
</SCRIPT>
</HEAD>
<BODY onactivate="EventOnActivate();" onload="EventOnLoad();">
<DIV ID="ID_Div"></DIV>
</BODY>
</HTML>
```

onafterprint

erzeugt mit Ende des Druck bzw. Druckvorschau

onafterupdate

erzeugt wenn Daten in einem Datasource-Objekt erfolgreich geupdatet wurden



onbeforeactivate

erzeugt direkt vor der Aktivierung eines Objektes

onbeforecopy

erzeugt direkt vor dem Kopieren einer Selektion im Objekt in das Clipboard

Beispiel

```
<HEAD>
<SCRIPT>
    var Daten= "Das sind Text-Daten";

    function Quelle_Beforecopy()
    {event.returnValue = false; }

    function Quelle_Copy()
    {window.clipboardData.setData("Text", Daten); }

    function Ziel_BeforePaste()
    {event.returnValue = false; }

    function Ziel_Paste()
    {
        ID_Input.value = window.clipboardData.getData("Text");
        event.returnValue = false;
    }
</SCRIPT>
</HEAD>
<BODY>
    <SPAN ID="ID_Span" onbeforecopy="Quelle_Beforecopy()"
        oncopy="Quelle_Copy()"
    >
        diesen Text kopieren
    </SPAN>
    <INPUT ID="ID_Input" onbeforepaste="Ziel_BeforePaste()"
        onpaste="Ziel_Paste()"
    >
</BODY>
```

onbeforecut

erzeugt direkt vor dem Ausschneiden einer Selektion im Objekt

Beispiel 1

```
<HEAD>
<SCRIPT>
    var Kette = "";

    function EventBeforeCut()
    {event.returnValue = false; }

    function EventCut()
    {
        Kette= ID_Span.innerText;
        ID_Span.innerText = "";
        event.returnValue = false;
    }

    function EventBeforePaste()
    {event.returnValue = false; }

    function EventPaste()
    {
        ID_Div.innerText = Kette;
        event.returnValue = false;
    }
</SCRIPT>
</HEAD>
<BODY>
    <SPAN ID="ID_Span" onbeforecut="EventBeforeCut()" oncut="EventCut()">
        diesen Text selektieren und ausschneiden
    </SPAN>
    <BR>
    <DIV ID="ID_Div" onbeforepaste="EventBeforePaste()" onpaste="EventPaste()">
        hierher den ausgeschnittenen Text einfügen
    </DIV>
```



```

</BODY>
Beispiel 2
<HEAD>
<SCRIPT>
    var TextDaten = "new content associated with this object";
    var Kette = "";

    function Init()
    {
        var TextBereich = document.body.createTextRange(); // Bereich aller Texte
        TextBereich.findText(ID_Span1.innerText);           // aber den Text im ID_Span1 suchen
        TextBereich.select();                               // und dann selektieren
    }

    function EventBeforeCut()
    {
        Kette = ID_Span1.innerText;
        event.returnValue = false;
    }

    function EventCut()
    {window.clipboardData.setData("Text", TextDaten); }      // setData liefert return-Wert
                                                                // also event.returnValue = ...;
                                                                // noch nötig zu kodieren

    function EventBeforePaste()
    {event.returnValue = false; }

    function EventPaste()
    {
        ID_Span2.value = window.clipboardData.getData("Text", TextDaten);
        event.returnValue = false;
    }
</SCRIPT>
</HEAD>
<BODY onload="Init()">
    <SPAN ID="ID_Span1" onbeforecut="EventBeforeCut()" oncut="EventCut()">
        Diesen Text ausschneiden
    </SPAN>
    <INPUT ID=" ID_Span2" TYPE="text" VALUE="Den ausgeschrittenen Text hier einfuegen"
        onbeforepaste="EventBeforePaste()"
        onpaste="EventPaste()"
    >
</BODY>

```

onbeforedeactivate

erzeugt direkt bevor ein Objekt als deaktiv gesetzt wird

onbeforeeditfocus

erzeugt direkt vor der User-Ineraktivität auf ein editierbares Objekt

immer erzeugt wenn INPUT-Objekt oder TEXTAREA-Objekt den focus erhält

onbeforepaste

erzeugt direkt vor dem Einfügen aus dem Clipboard in ein Objekt

onbeforeprint

erzeugt direkt vor dem Druck bzw. Druckvorschau

onbeforeunload

erzeugt direkt vor dem Unload eines Dokumentes

unload = verlassen des Dokumentes durch Laden eines neuen Dokumentes in das selbe Fenster

Beispiel

```

<HTML>
<HEAD>
<SCRIPT>
    function DokumentSchliessen()
    {event.returnValue = "Irgendeinen Stringwert liefern. Vor dem Schliessen wird Dialogbox angezeigt."}
</SCRIPT>
</HEAD>

```



```
<BODY onbeforeunload="DokumentSchliessen ()">
  <A HREF="http://www.test.de">zu www.test.de</A>
</BODY>
</HTML>
```

onbeforeupdate

erzeugt mit dem Beginn des Update von Daten eines Datasource-Objektes

onbegin

erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird

nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Element
 erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Eltern-Element
 siehe onend und onrepeat
 siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:EXCL ID="ID_Excl"
  BEGIN="indefinite"
  DUR="5"
  REPEATCOUNT="5"
  onbegin="alert('Start der Animation');"
>
  <t:ANIMATEMOTION ID="ID_Animatemotion"
    TARGETELEMENT="ID_Div"
    TO="200,0"
    BEGIN="0"
    DUR="2"
    AUTOREVERSE="true"
  >
  </t:ANIMATEMOTION>
</t:EXCL>
<DIV ID="ID_Div"
  CLASS="time_line_klasse"
  STYLE="position:relative;top:15px;left:25px;height:100px;width:100px;
  border:solid black 1px;
  "
>
  sich bewegender DIV
</DIV>
<BUTTON onclick=" ID_Animatemotion.beginElement();">Start</BUTTON>
<BUTTON onclick=" ID_Animatemotion.endElement();">Stop</BUTTON>
</BODY>
</HTML>
```

onblur

erzeugt wenn Objekt den Focus verliert

Beispiel

```
<HTML>
<BODY>
  <INPUT TYPE=text NAME="Test" VALUE="onblur Test" onblur="alert(event.srcElement.name)">
</BODY>
</HTML>
```

onbounce

erzeugt wenn Behavior-Eigenschaft des Marquee-Objektes auf "alternate" gesetzt ist **und** der Marquee-Text eine der beiden Seiten des Elternfensters erreicht hat

Beispiel 1:

```
<BODY>
  <MARQUEE BEHAVIOR="alternate" WIDTH=200 LOOP=3
    onbounce="alert('onbounce-Ereignis erkannt')">
```



```

    >
        Marquee Text
    </MARQUEE>
</BODY>

```

Beispiel 2:

```

<HTML>
<BODY bgcolor="0000ff"
onBlur="document.bgColor='ff0000'"
onFocus="document.bgColor='0000ff'"
>
    Dieses Fenster &auml;ndert die Hintergrundfarbe, wenn es nicht mehr aktiv ist.
</BODY>
</HTML>

```

oncellchange

erzeugt wenn Daten verändert wurden in der Datenquelle z.B. Objekt

onchange

erzeugt, wenn Kontext des Objektes sich ändert

Beispiel:

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function aenderung(wert)
    {
        alert("Die Eingabe hat sich geaendert!\n" + "Der neue Wert ist " + wert);
    }
// -->
</SCRIPT>
</HEAD>
<BODY>
<FORM>
    <INPUT TYPE="text" onchange="aenderung(this.value)">
</FORM>
</BODY>
</HTML>

```

onclick

erzeugt mit Druck der linken Maustaste auf ein Objekt

benötigt vorher die Eventfolge erzeugt onmousedown and onmouseup
wobei die Maus auch dabei über dem Objekt sein muss
bei Radio-Buttons-Gruppe: onclick event immer nach
onbeforeupdate and onafterupdate events for the control group.
bei focusfähigem Objekt: onfocus event erzeugt vor dem onclick event
bei Doppelklick per linker Maustaste:
zuerst onclick ausgelöst, dann ondblclick

oncontentready

erzeugt, wenn auf die HTC-Komponente zugegriffen wird und dieses komplett geparkt ist und das Event

window.onload erzeugt wurde

Hinweis: Das Parsen einer HTC-Komponente kann durch die Komponente selbst verhindert werden.
siehe Objekt element und HTC-Datei

Beispiel:

```

<PUBLIC:ATTACH EVENT="oncontentready" ONEVENT="Anzeige()" />
<SCRIPT LANGUAGE="JScript">
    function Anzeige()
    {window.alert ('Objekt element mit innerHTML = ' + element.innerHTML); }
</SCRIPT>

```

oncontentsave

erzeugt, wenn die Umgebung der HTC-Komponente gespeichert oder kopiert wird

Hinweis: Änderung einer Eigenschaft der HTC-Komponente wird per Event

onpropertychange angezeigt

siehe Objekt element und HTC-Datei



oncontextmenu

erzeugt wenn rechte Maustaste gedrückt wurde

Unterdrückung, wenn der Eventhandler event.returnValue=false; enthält
(mit return false; erfolgt keine Unterdrückung)

Beispiel

```
<SPAN STYLE="width:300; background-color:blue; color:white;" oncontextmenu="return false">
  <P>Das Kontextmenue ist innerhalb dieses SPAN abgeschaltet</P>
</SPAN>
```

oncontrolselect

erzeugt wenn User eine Control-Selektion im Objekt tätigt (z.B. CTRL+ Maus)

oncopy

erzeugt wenn Quellelement oder Selektion dem Clipboard hinzugefügt wird

per rechte Maus-Click und Anwahl des Menüpunktes Copy
Hinweis: Festlegung der Datenart per Methode .setData()
oder CTRL-C

oncut

erzeugt durch Quell-Objekt wenn Daten aus Quellobjekt in das Clipboard verschoben werden

Beispiel 1

```
<HEAD>
<SCRIPT>
  var Kette = "";

  function EventBeforeCut()
  {event.returnValue = false; }

  function EventCut()
  {
    Kette= ID_Span.innerText;
    ID_Span.innerText = "";
    event.returnValue = false;
  }

  function EventBeforePaste()
  {event.returnValue = false; }

  function EventPaste()
  {
    ID_Div.innerText = Kette;
    event.returnValue = false;
  }
</SCRIPT>
</HEAD>
<BODY>
  <SPAN ID="ID_Span" onbeforecut="EventBeforeCut()"oncut="EventCut()">
    diesen Text selektieren aund ausschneiden
  </SPAN>
  <BR>
  <DIV ID="ID_Div" onbeforepaste="EventBeforePaste()"onpaste="EventPaste()">
    hierher den ausgeschnittenen Text einfügen
  </DIV>
</BODY>
```

Beispiel 2

```
<HEAD>
<SCRIPT>
  var TextDaten = "new content associated with this object";
  var Kette = "";

  function Init()
  {
    var TextBereich = document.body.createTextRange(); // Bereich aller Texte
    TextBereich.findText(ID_Span1.innerText); // aber den Text im ID_Span1 suchen
    TextBereich.select(); // und dann selektieren
  }

  function EventBeforeCut()
  {
    Kette = ID_Span1.innerText;
```



```

        event.returnValue = false;
    }

    function EventCut()
    {window.clipboardData.setData("Text", TextDaten); }           // setData liefert return-Wert
                                                                    // also event.returnValue = ...;
                                                                    // noch nötig zu kodieren

    function EventBeforePaste()
    {event.returnValue = false; }

    function EventPaste()
    {
        ID_Span2.value = window.clipboardData.getData("Text", TextDaten);
        event.returnValue = false;
    }
</SCRIPT>
</HEAD>
<BODY onload="Init()">
    <SPAN ID="ID_Span1" onbeforecut="EventBeforeCut()" oncut="EventCut()">
        Diesen Text ausschneiden
    </SPAN>
    <INPUT ID=" ID_Span2" TYPE="text" VALUE="Den ausgeschrittenen Text hier einfuegen"
    onbeforepaste="EventBeforePaste()"
    onpaste="EventPaste()"
    >
</BODY>

```

ondataavailable

erzeugt von der Datenquelle bei jedem Senden von Daten (asynchrones Senden von der Datenquelle (Objekt))

ondatachanged

erzeugt von der Datenquelle wenn Daten geändert wurden in der Datenquelle verwendet bei Filter-Operationen

ondatacomplete

erzeugt von Datenquelle wenn alle Daten der Quelle verfügbar also sendbar sind

ondblclick

erzeugt wenn Doppelklick mit Maus mit folgender Eventfolge
onmousedown, onmouseup, onclick, onmouseup, and then ondblclick.

ondeactivate

erzeugt mit Deaktivierung eines Objektes

ondetach

erzeugt, wenn die Event-Überwachung in einer HTC-Komponente abgeschaltet wird siehe Objekt element und HTC-Datei

Beispiel:

```

<PUBLIC:ATTACH EVENT="ondetach" ONEVENT="EventEntfernen()"/>

<SCRIPT LANGUAGE="JScript">
    function CursorNeu()
    {
        if (event.srcElement == element)
        {
            normalColor = style.color;
            runtimeStyle.color = "red";
            runtimeStyle.cursor = "hand";
        }
    }

    function CursorNormal()
    {
        if (event.srcElement == element)
        {
            runtimeStyle.color = normalColor;

```




```

        runtimeStyle.cursor = "";
    }
}

function EventEntfernen()
{
    detachEvent ('onmouseover', CursorNeu); // nicht () kodieren !!!
    detachEvent ('onmouseout', CursorNormal); // nicht () kodieren !!
}

attachEvent ('onmouseover', CursorNeu);
attachEvent ('onmouseout', CursorNormal);
</SCRIPT>

```

onreadystatechange

erzeugt, wenn das HTML-Dokument, das die HTC-Komponente implementiert hat, komplett geparkt ist und das Event window.onload erzeugt wurde
 Parsen des HTML-Dokumentes schliesst Implementierung der HTC-Datei und Parsen der HTC-Komponenten ein
 Hinweis: Das Parsen einer HTC-Komponente kann durch die Komponente selbst verhindert werden. siehe Objekt element und HTC-Datei

ondrag

erzeugt kontinuierlich während allen Drag-Operationen, aber erst nach ondragstart Event
 Drag = Maus gedrückt halten

ondragend

erzeugt am Ende der Dragoperation also wenn Maus losgelassen wird
 erzeugt immer vom Quellobjekt
 Hinweis: Zielobjekt erzeugt anschliessend ondragleave Event

ondragenter

erzeugt wenn Maus über Objekt gedrückt wurde UND der User das Quell-Objekt bei gedrückter Maustaste zieht
 erzeugt immer vom Ziel-Objekt wenn Ziel erreicht wurde
 ist das ERSTE Event des Ziels, also das als Erstes ausgelöste Event
 erzeugt immer vom Quell-Objekt solange Ziel nicht erreicht wurde
 wird immer direkt vor dem ondragover Event erzeugt

ondragleave

erzeugt vom Ziel-Objekt wenn User die Maus aus dem Ziel bewegt während Drag-Operation

ondragover

erzeugt vom Ziel-Objekt wenn User die Maus über das Ziel bewegt während Drag-Operation
 UND immer direkt nach dem ondragenter Event

ondragstart

erzeugt vom Quell-Objekt wenn User die Selektion für Drag-Operationen ausführt
 auch bei Textelement
 ist ERSTES Event für Drag-und Drop
 Hinweis: Quell-Objekt nutzt Methode .setData() für Übergabe der Daten

ondrop

erzeugt vom Ziel-Objekt wenn Maustaste losgelassen wird über dem Ziel
 Drop = Ablegen
 direkt erzeugt vor ondragleave and ondragend Events.

onend

erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut .repeatCount
 bzw. wenn das aktive Element gestoppt wird
 erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit des Kindelement ebenfalls endet
 nicht erzeugt wenn .fill mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht



siehe Methode .endElement() und Eigenschaft .end
 siehe onbegin und onrepeat
 siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:EXCL ID="ID_Excl"
BEGIN="indefinite"
DUR="5"
REPEATCOUNT="5"
onend="alert('Ende der Animation');"
>
<t:ANIMATEMOTION ID="ID_Animatemotion"
TARGETELEMENT="ID_Div"
TO="200,0"
BEGIN="0"
DUR="2"
AUTOREVERSE="true"
>
</t:ANIMATEMOTION>
</t:EXCL>
<DIV ID="ID_Div"
CLASS="time_line_klasse"
STYLE="position:relative;top:15px;left:25px;height:100px;width:100px;
border:solid black 1px;
"
>
sich bewegender DIV
</DIV>
<BUTTON onclick=" ID_Animatemotion.beginElement();">Start</BUTTON>
<BUTTON onclick=" ID_Animatemotion.endElement();">Stop</BUTTON>
</BODY>
</HTML>
```

onerror

erzeugt wenn Laufzeit-Fehler beim Laden eines Objektes

Beispiel: Fehlerverfolgung

```
<SCRIPT>
function FehlerAufspueren (MeldungString, UrlString, ZielenNummerString)
{
ID_Div.innerHTML += "<B>Fehler erkannt</B>";
ID_Div.innerHTML += "Error: " + MeldungString + "<BR>";
ID_Div.innerHTML += "Line: " + ZielenNummerString + "<BR>";
ID_Div.innerHTML += "URL: " + UrlString + "<BR>";

return false;
}

window.onerror=FehlerAufspueren; // ohne () kodieren
</SCRIPT>
<DIV ID="ID_Div"></DIV>
```

Beispiel für Bild

```
<SCRIPT>
var Kette = '<IMG STYLE="display: none;" ID="ID_IMG" ALT="Das ist ein Bild ">';

function Laden()
{
ID_Div.innerHTML=Kette; // wird sofort geparkt, also IMG-Tag ausgeführt
// ( anstelle von eval()
// bzw. document.write() )

ID_IMG.src="";
ID_IMG.style.display="block";

ID_IMG.onerror= IMGAltTextAufFehlerMeldung; // ohne ()
}
}
```



```

function IMGAltTextAufFehlerMeldung ()
{
    ID_IMG.alt="Das Bild konnte nicht geladen werden.";
    return true;
}
</SCRIPT>
<INPUT TYPE=button VALUE="Klicke zum Bildladen" onclick="Laden()">
<DIV ID="ID_Div"></DIV>

```

onerrorupdate

erzeugt wenn Fehler während Datenupdate für ein Datasource-Objekt auftrat

onfilterchange

erzeugt wenn ein visueller Filter komplett abgelaufen ist

onfinish

erzeugt wenn alle Durchläufe des Marquee-Objektes komplett beendet sind

Anzahl der Durchläufe laut LOOP-Attribut

LOOP-Attribut muss Wert > 1 haben, wenn onfinish erzeugt werden soll

Beispiel

```

<BODY>
<MARQUEE LOOP=2 onfinish="alert(event.srcElement.id + ' onfinish-Ereignis erkannt')">
    Marquee Text
</MARQUEE>
</BODY>

```

onfocus

erzeugt, wenn Objekt den Focus erhält

Fokus nur erhaltbar nach dem kompletten Laden des Dokumentes

Beispiel für Label

```

<STYLE>
.normal {background-color:#FFFFFF; color:#000000; font-weight:normal; font-size:8pt; font-family:Arial;}
.accessible { background-color:"beige"; font-weight:bold; font-size:10pt;}
</STYLE>
<SCRIPT>
function StyleWechsel()
{
    // Input-Objekt
    event.srcElement.className="accessible"; // Input-Objekt

    // Label-Objekt
    var ZeigerAuf Label =eval(event.srcElement.id + "_Label ");
    // ID ist "ID_Input"
    // also "ID_Input" + "_Label" = "ID_Input_Label"
    // Zeichenkettenoperation leider nötig, wenn mehrere
    // Objekte mit Eventerzeugung vorhanden wären
    // und ebenfalls nötig wegen Bezug im Label auf das
    // ID des Objektes, das Label haben soll
    ZeigerAuf Label.className="accessible";
}
</SCRIPT>
<LABEL FOR="ID_Input" oInput" CLASS="normal" ID="ID_Input_Label">Text eingeben</LABEL>
<INPUT TYPE="text" CLASS="normal" onfocus="StyleWechsel()" ID="ID_Input">

```

Beispiel für INPUT:

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
function setzeFokus()
{
    document.meinFormular.eingabe.focus();
    document.meinFormular.eingabe.select();
}
// -->
</SCRIPT>
</HEAD>
<BODY>

```



```

<FORM NAME="meinFormular">
  <INPUT TYPE="text"
        NAME="eingabe"
        VALUE="Eingabe..."
  >
  <INPUT TYPE="button"
        VALUE="Bitte drücken"
        onclick="setzeFokus();">
</FORM>
</BODY>
</HTML>

```

onfocusin

erzeugt bevor das Element den Focus erhält

onfocusout

erzeugt nachdem das Element den Focus verloren hat

onhelp

erzeugt wenn F1-Taste gedrückt im aktuellen Fenster

onhide

erzeugt wenn der Media Bar Player gerade unsichtbar gemacht wird (versteckt wird)

siehe Eigenschaft .enabled
entspricht dem Schliessen des Media Bar Player durch den User
Media Bar Player ist der Windows Media Player
siehe Behavior .style.mediaBar

onkeydown

erzeugt wenn irgend eine Tastatur-Taste gedrückt wurde

Beispiel

```

<SCRIPT>
  function TastenCodeHolen()
  {
    if(ID_Input.checked)
    {
      ID_Textarea.innerText+="[Keycode = " + event.keyCode + "];
      event.returnValue=false;
    }
    else
    { ID_Textarea.innerText+=String.fromCharCode(event.keyCode); }
  }
</SCRIPT>
<INPUT TYPE="checkbox" ID="ID_Input">
<INPUT TYPE="text" onkeydown="TastCodeHolen()">
<TEXTAREA ID="ID_Textarea" ROWS="10" COLS="50"></TEXTAREA>

```

onkeypress

erzeugt bei Druck einer alphanumerischen Tastatur-Taste

Beispiel

```

<HEAD>
<SCRIPT>
  function ShiftPruefen()
  {
    if (window.event.shiftKey)
    { ID_Input.value = "Shift erkannt"; }
  }
</SCRIPT>
</HEAD>
<BODY>
  drücke SHIFT mit anderer Taste
  <INPUT TYPE="text" onkeypress="ShiftPruefen()">
  <INPUT TYPE="text" ID="ID_Input">
</BODY>

```

onkeyup

erzeugt wenn gedrückte Tastatur-Taste losgelassen wird



onlayoutcomplete

erzeugt wenn das Druckobjekt bzw. Druckvorschau-Objekt komplett gefüllt ist
setzt Eigenschaft .contentOverflow auf true

onload

erzeugt mit dem Laden eines Objektes

Beispiel 1

```
<BODY>
  <SCRIPT FOR=window EVENT=onload LANGUAGE="JScript">
    window.status = "Seite ist geladen!";
  </SCRIPT>
</BODY>
```

Beispiel 2

```
<SCRIPT>
  function Meldung()
  {window.status = "Image " + event.srcElement.src + " ist geladen";}
</SCRIPT>
<BODY>
  <IMG SRC="test.gif" onload="Meldung()">
</BODY>
```

Beispiel 3 für Ermittlung der Verweilzeit des Users auf der Webseite

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
<!--
  var start_zeit;      // muss global sein

  function start()
  {start_zeit = new Date();}

  function ende()
  {
    var ende_zeit = new Date();
    var wert=ende_zeit.getTime() - start_zeit.getTime();
    wert = wert /1000;   // in Sekunden umrechnen
    alert("Verweilzeit in Sekunden: " + Math.floor(wert).toString());
  }
-->
</SCRIPT>
</HEAD>

<BODY ... onLoad="start();" onUnload="ende();">
</BODY>
<HTML>
```

onlosecapture

erzeugt, wenn Objekt die Mausüberwachung verliert

Beispiel

```
<BODY onload="ID_Div.setCapture()"
  onclick="ID_Div.releaseCapture();"
>
  <DIV ID="ID_Div" divOwnCapture
    onmousemove="ID_Textarea.value=event.clientX + event.clientY"; // kein alert() !!!!!
    onlosecapture="alert(event.srcElement.id + ' ohne Maus-Ueberwachung)";
  >
    Ueber diesen Text mit der Maus fahren
    <BR>
    <TEXTAREA ID="ID_Textarea" COLS=2></TEXTAREA>
  </DIV>
  <DIV>
    Klick hier fuer Event onlosecapture per Methode .releaseCapture()
  </DIV>
</BODY>
```

onmediacomplete

erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde
für Animation per Timeline
Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer



Datei (streaming media file) z.B. Gif-Bild, Video etc.
 sinnvoll für Start des Elementes auf der Timeline
 siehe onmediaerror
 siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:VIDEO ID="ID_Video"
SRC="test.wmv"
onmediacomplete="ID_Span.innerText +=
'Datei test.wmv wurde komplett geladen !'
"
>
</t:VIDEO>
<BR>
<SPAN ID="ID_Span"></SPAN>
</BODY>
</HTML>
```

onmediaerror

erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde
 oder das Medium einen Fehler enthält
 für Animation per Timeline
 Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer
 Datei (streaming media file) z.B. Gif-Bild, Video etc.
 sinnvoll für Start des Elementes auf der Timeline
 ersetzt das Ereignis onmedialoadfailed, das deprecated ist und nicht mehr verwendet
 werden darf !
 siehe onmediacomplete
 siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:IMG ID="ID_Img"
SRC="test.gif"
STYLE="position:relative;top:25px;left:50px;height:100px;width:100px;"
onmediaerror="alert('Datei test.gif nicht ladbar !')">
>
</t:IMG>
<t:ANIMATION ID="ID_Animation"
TARGETELEMENT="ID_Img"
TO="0,400"
DUR="2"
BEGIN="0"
ACCELERATE="1"
AUTOREVERSE="true"
REPEATCOUNT="5"
>
</t:ANIMATION>
</BODY>
</HTML>
```

onmousedown

erzeugt wenn irgendeine Maustaste gedrückt wird

onmouseenter

erzeugt wenn Maus ein Objektbereich betritt



onmouseleave

erzeugt wenn Maus gerade den Objektbereich verlässt

onmousemove

erzeugt wenn Maus im Bereich eines Objekt bewegt wird

Beispiel

```
<SCRIPT>
    function Anzeige()
    {ID_Span.innerHTML="Coords: (" + event.clientX + ", " + event.clientY + ")";} // kein alert() !!!!
</SCRIPT>
<DIV onmousemove="Anzeige()">
    <SPAN ID="ID_Span"></SPAN>
</DIV>
```

onmouseout

erzeugt wenn Maus den Bereich eines Objektes gerade verlässt
genau 1x erzeugt pro Verlassen

onmouseover

erzeugt wenn Maus den Bereich eines Objektes gerade betritt
genau 1x erzeugt pro Betreten

onmouseup

erzeugt wenn Maustaste losgelassen wird

Hinweis: Eigenschaft .button liefert die losgelassene Taste

onmousewheel

erzeugt wenn Mause rad gedreht wird
mit dem Drehen wird Eigenschaft .wheelDelta gefüllt
mit Integer-Vielfachen von 120 Grad Umdrehung
wenn > 0 so Drehung vom User weg
wenn < 0 so Drehung zum User hin

ab IE 6.0

Beispiel

```
<HTML>
<HEAD>
<SCRIPT>
    var ZoomFaktorStartWert = 10; // > 0, ganzzahlig, in Prozent
    var ZoomSchrittWeite = 1; // > 0, ganzzahlig
    var DrehSchrittWeite = 1; // 1 oder 2 oder 3 da Faktor von 120 Grad

    var ZoomFaktor = ZoomFaktorStartWert; // > 0, ganzzahlig, in Prozent
    function VergroessernVerkleinern()
    {
        if (event.wheelDelta >= (DrehSchrittWeite * 120))
        { ZoomFaktor += ZoomSchrittWeite; }
        else
        {
            if (event.wheelDelta <= (-1 * (DrehSchrittWeite * 120)))
            { ZoomFaktor -= ZoomSchrittWeite; }
        }

        if (ZoomFaktor <= 0)
        { ZooFaktor = ZoomFaktorStartWert; }

        ID_Image.style.zoom = ZoomFaktor + '%';
    }
</SCRIPT>
</HEAD>
<BODY>
    <IMG ID="ID_Image" SRC="test.jpg" onmousewheel="VergroessernVerkleinern()">
</BODY>
</HTML>
```

onmove

erzeugt, wenn Objekt bewegt wird:

Objekt muss **absolute positioniert** sein (im STYLE per "**position:absolute;**")

nicht erzeugt, wenn ein Unterobjekt bewegt wird:

Unterobjekt muss **eigenen** Handler haben



Beispiel

```

<HTML>
<HEAD>
<SCRIPT>
    function HandleFuerOnMoveStart()
    {
        var ZeigerAufObjektMitEvent = event.srcElement; // anstelle des ID vom DIV
                                                         // falls mehrere bewegbare
                                                         // Objekte vorhanden sind
        ZeigerAufObjektMitEvent.style.backgroundColor = "green";
        ZeigerAufObjektMitEvent.innerText = "DIV wird bewegt ";
    }

    function HandlerFuerOnMove ()
    {
        ID_Span1.innerHTML = event.srcElement.offsetLeft;
        ID_Span2.innerHTML = event.srcElement.offsetTop;
    }

    function HandleFuerOnMoveEnd()
    {
        var ZeigerAufObjektMitEvent = event.srcElement; // anstelle des ID vom DIV
                                                         // falls mehrere bewegbare
                                                         // Objekte vorhanden sind
        ZeigerAufObjektMitEvent.style.backgroundColor = "red";
        ZeigerAufObjektMitEvent.innerText = "DIV wird nicht mehr bewegt";
    }

    document.execCommand("2D-position",false,true); // 2-D Positionierung einschalten
</SCRIPT>
</HEAD>
<BODY onmovestart="HandleFuerOnMoveStart();"
onmove="HandlerFuerOnMove();"
onmoveend="HandleFuerOnMoveEnd();"
>
    offsetLeft = <SPAN ID="ID_Span1"></SPAN>
<BR>
    offserTop = <SPAN ID="ID_Span2"></SPAN>
<BR>
    <DIV CONTENTEDITABLE="true">
        <DIV STYLE="position:absolute;width:300px;height:100px; background-color:red;">
            bewegbarer DIV
        </DIV>
    </DIV>
</BODY>
</HTML>

```

onmoveend

erzeugt, wenn das Bewegen eines Objektes endet:

Objekt muss **absolute positioniert** sein (im STYLE per "**position:absolute;**")

nicht erzeugt, wenn ein Unterobjektbewegung endet:

Unterobjekt muss **eigenen** Handler haben

Beispiel

```

<HTML>
<HEAD>
<SCRIPT>
    function HandleFuerOnMoveStart()
    {
        var ZeigerAufObjektMitEvent = event.srcElement; // anstelle des ID vom DIV
                                                         // falls mehrere bewegbare
                                                         // Objekte vorhanden sind
        ZeigerAufObjektMitEvent.style.backgroundColor = "green";
        ZeigerAufObjektMitEvent.innerText = "DIV wird bewegt ";
    }

    function HandlerFuerOnMove ()
    {
        ID_Span1.innerHTML = event.srcElement.offsetLeft;
        ID_Span2.innerHTML = event.srcElement.offsetTop;
    }

    function HandleFuerOnMoveEnd()
    {

```




```

        var ZeigerAufObjektMitEvent = event.srcElement; // anstelle des ID vom DIV
                                                    // falls mehrere bewegbare
                                                    // Objekte vorhanden sind
        ZeigerAufObjektMitEvent.style.backgroundColor = "red";
        ZeigerAufObjektMitEvent.innerText = "DIV wird nicht mehr bewegt";
    }

    document.execCommand("2D-position",false,true); // 2-D Positionierung einschalten
</SCRIPT>
</HEAD>
<BODY onmovestart="HandleFuerOnMoveStart();"
onmove="HandlerFuerOnMove();"
onmoveend="HandleFuerOnMoveEnd();"
>
    offsetLeft = <SPAN ID="ID_Span1"></SPAN>
    <BR>
    offserTop = <SPAN ID="ID_Span2"></SPAN>
    <BR>
    <DIV CONTENTEDITABLE="true">
        <DIV STYLE="position:absolute;width:300px;height:100px; background-color:red;">
            bewegbarer DIV
        </DIV>
    </DIV>
</BODY>
</HTML>

```

onmovestart

erzeugt, wenn das Bewegen eines Objektes beginnt:

Objekt muss **absolute positioniert** sein (im STYLE per "**position:absolute;**")

nicht erzeugt, wenn ein Unterobjektbewegung beginnt:

Unterobjekt muss **eigenen** Handler haben

Beispiel

```

<HTML>
<HEAD>
<SCRIPT>
    function HandleFuerOnMoveStart()
    {
        var ZeigerAufObjektMitEvent = event.srcElement; // anstelle des ID vom DIV
                                                    // falls mehrere bewegbare
                                                    // Objekte vorhanden sind

        ZeigerAufObjektMitEvent.style.backgroundColor = "green";
        ZeigerAufObjektMitEvent.innerText = "DIV wird bewegt ";
    }

    function HandlerFuerOnMove ()
    {
        ID_Span1.innerHTML = event.srcElement.offsetLeft;
        ID_Span2.innerHTML = event.srcElement.offsetTop;
    }

    function HandleFuerOnMoveEnd()
    {
        var ZeigerAufObjektMitEvent = event.srcElement; // anstelle des ID vom DIV
                                                    // falls mehrere bewegbare
                                                    // Objekte vorhanden sind

        ZeigerAufObjektMitEvent.style.backgroundColor = "red";
        ZeigerAufObjektMitEvent.innerText = "DIV wird nicht mehr bewegt";
    }

    document.execCommand("2D-position",false,true); // 2-D Positionierung einschalten
</SCRIPT>
</HEAD>
<BODY onmovestart="HandleFuerOnMoveStart();"
onmove="HandlerFuerOnMove();"
onmoveend="HandleFuerOnMoveEnd();"
>
    offsetLeft = <SPAN ID="ID_Span1"></SPAN>
    <BR>
    offserTop = <SPAN ID="ID_Span2"></SPAN>
    <BR>
    <DIV CONTENTEDITABLE="true">
        <DIV STYLE="position:absolute;width:300px;height:100px; background-color:red;">
            bewegbarer DIV

```



```

    </DIV>
  </BODY>
</HTML>

```

onopenstatechange

erzeugt, wenn Media Bar Player den Status bezüglich Playliste, Codec, Lizenz und Individualisierung ändert
 siehe Eigenschaft .openState
 Media Bar Player ist der Windows Media Player
 siehe Behavior .style.mediaBar

Beispiel:

```

<DIV ID="ID_Div"
  STYLE="behavior:url(#default#mediaBar)"
  onopenstatechange="alert(ID_Div.openState);">
  >
</DIV>
<INPUT TYPE=button
  VALUE='abspielen von test.asx'
  onclick=" ID_Div.playURL('http://www.test.de/test.asx','video/x-ms-asf');
  ID_Div.disabledUI = true;
  ">

```

onoutofsync

erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes)
 sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked":
 Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch .resetElement()
 siehe onsyncstored
 siehe Objekt currTimeState und Behavior .style.time2

onpaste

erzeugt vom Zielobjekt wenn Daten aus Clipboard in das Objekt eingefügt werden

Beispiel 1

```

<HEAD>
<SCRIPT>
  var Kette = "";

  function EventBeforeCut()
  {event.returnValue = false; }

  function EventCut()
  {
    Kette= ID_Span.innerText;
    ID_Span.innerText = "";
    event.returnValue = false;
  }

  function EventBeforePaste()
  {event.returnValue = false; }

  function EventPaste()
  {
    ID_Div.innerText = Kette;
    event.returnValue = false;
  }
</SCRIPT>
</HEAD>
<BODY>
  <SPAN ID="ID_Span" onbeforecut="EventBeforeCut()"oncut="EventCut()">
    diesen Text selektieren aund ausschneiden
  </SPAN>
  <BR>
  <DIV ID="ID_Div" onbeforepaste="EventBeforePaste()"onpaste="EventPaste()">
    hierher den ausgeschnittenen Text einfügen
  </DIV>
</BODY>

```

Beispiel 2



```

<HEAD>
<SCRIPT>
    var TextDaten = "new content associated with this object";
    var Kette = "";

    function Init()
    {
        var TextBereich = document.body.createTextRange(); // Bereich aller Texte
        TextBereich.findText(ID_Span1.innerText);           // aber den Text im ID_Span1 suchen
        TextBereich.select();                               // und dann selektieren
    }

    function EventBeforeCut()
    {
        Kette = ID_Span1.innerText;
        event.returnValue = false;
    }

    function EventCut()
    {window.clipboardData.setData("Text", TextDaten); }           // setData liefert return-Wert
                                                                    // also event.returnValue = ...;
                                                                    // noch nötig zu kodieren

    function EventBeforePaste()
    {event.returnValue = false; }

    function EventPaste()
    {
        ID_Span2.value = window.clipboardData.getData("Text", TextDaten);
        event.returnValue = false;
    }
</SCRIPT>
</HEAD>
<BODY onload="Init()">
    <SPAN ID="ID_Span1" onbeforecut="EventBeforeCut()" oncut="EventCut()">
        Diesen Text ausschneiden
    </SPAN>
    <INPUT ID=" ID_Span2" TYPE="text" VALUE="Den ausgeschrittenen Text hier einfuegen"
        onbeforepaste="EventBeforePaste()"
        onpaste="EventPaste()"
    >
</BODY>

```

onpause

erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren
auch bei body Objekt
siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <t:VIDEO ID="ID_Video"
        SRC="test.wmv"
        onpause="ID_Span.innerText = "Pause !"
        onmediacomplete="ID_Span.innerText =
            'Datei test.wmv wurde komplett geladen !'
        "
    >
    </t:VIDEO>
    <BR>
    <SPAN ID="ID_Span"></SPAN>
</BODY>
</HTML>

```

onplaystatechange

erzeugt, wenn Media Bar Player den Status bezüglich Wiedergabe ändert
siehe Eigenschaft .playState



Media Bar Player ist der Windows Media Player
siehe Behavior .style.mediaBar

Beispiel:

```
<DIV ID="ID_Div"
STYLE="behavior:url(#default#mediaBar)"
onplaystatechange="alert(ID_Div.playState);"
>
</DIV>
<INPUT TYPE=button
VALUE='abspielen von test.asx'
onclick=" ID_Div.playURL('http://www.test.de/test.asx',video/x-ms-asf);
ID_Div.disabledUI = true;
"
>
```

onpropertychange

erzeugt wenn eine Objekteigenschaft geändert wird

außer Änderung von .innerText und .innerHTML

Beispiel

```
<HEAD>
<SCRIPT>
function ValueAendern()
{ ID_Input1.value = "Neuer Wert von VALUE";}

function FarbeAendern()
{ ID_Input2.style.backgroundColor = "aqua";}

function Anzeige()
{alert(event.propertyName + " wurde verändert");}

</SCRIPT>
</HEAD>
<BODY>
<INPUT TYPE=button ID="ID_Input1"
VALUE="Click um VALUE zu ändern"
onclick="ValueAendern()"
onpropertychange="Anzeige()"
>
<INPUT TYPE=button ID="ID_Input2"
VALUE="Click um Farbe zu ändern"
onclick="FarbeAendern()"
onpropertychange="Anzeige()"
>
</BODY>
```

onpropertychange

erzeugt, wenn die Methode **ID_HTC_Komponente.fireChange()** in der Funktion laut Attribut PUT

aufgerufen wird (Aufruf muss programmiert werden), wobei die Funktion eine Eigenschaft der

HTC-Komponente mit neuem Wert belegt

siehe Objekt element und HTC-Datei

onreadystatechange

erzeugt wenn Status eines Objektes sich ändert

immer erzeugt von datenladenden folgender Objekte:

applet, document, frame, frameSet, iframe, img, link, object, script und xml elements.

Beispiel

```
function Preufen ()
{
if (document.readyState=="complete")
{alert('Dokument komplett geladen und mit den Daten initialisiert.);}
}

document.onreadystatechange=Preufen; // ohne () kodieren
```

onrepeat

erzeugt mit Start **jeder** Wiederholung der Animation des aktiven Elementes

nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist

nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Heraufreichen des



Ereignisses onrepeat zum Elternelement)
 .repeatCount bzw. .repeat muss > 1 sein:
 onrepeat wird also .repeatCount - 1 mal erzeugt
 Kind eines Elementes
 siehe onend, onbegin, .repeatCount und .repeat
 siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:IMG ID="ID_Img"
SRC="test.gif"
STYLE="position:relative;top:25px;left:50px;height:100px;width:100px;"
onmediaerror="alert('Datei test.gif nicht ladbar !')"
onrepeat="alert('Aktuelle Wiederholung: ' + ID_Animation.currTimeState.repeatCount);"
>
</t:IMG>
<t:ANIMATION ID="ID_Animation"
TARGETELEMENT="ID_Img"
TO="0,400"
DUR="2"
BEGIN="0"
ACCELERATE="1"
AUTOREVERSE="true"
REPEATCOUNT="5"
>
</t:ANIMATION>
</BODY>
</HTML>
```

onreset

erzeugt wenn Formular zurückgesetzt wird

Resetaktion erzeugbar per
 INPUT TYPE="reset"
 BUTTON TYPE="reset"

Beispiel

```
<HTML>
<HEAD>
<SCRIPT>
function Anzeige()
{ ID_Span.innerText += "Anzeige zum Formular Reset";}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="Formular" ID="ID_Formular" onreset="Anzeige();"
<INPUT TYPE="text" NAME="InputText" VALUE="">
<BR>
<INPUT TYPE="reset" VALUE="Formular Reset">
<BUTTON onclick="ID_Formular.reset();">Formular Reset</BUTTON>
</FORM>
<SPAN ID="ID_Span"></SPAN>
<BR>
<BUTTON onclick="location.reload(true);">Refresh der Seite</BUTTON>
</BODY>
</HTML>
```

onreset

erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(),

also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht
 und damit zurückgesetzt wurde

also nicht beim Initialisieren des Elementes und seiner Timeline
 durch Instanziierung des Elementes

siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
```



```

<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  <t:EXCL ID="ID_Excl"
        BEGIN="indefinite"
        DUR="5"
        REPEATCOUNT="5"
        onreset="alert('Rücksetzen der Animation und Timeline');">
    <t:ANIMATEMOTION ID="ID_Animatemotion"
                    TARGETELEMENT="ID_Div"
                    TO="200,0"
                    BEGIN="0"
                    DUR="2"
                    AUTOREVERSE="true">
    </t:ANIMATEMOTION>
  </t:EXCL>
  <DIV ID="ID_Div"
        CLASS="time_line_klasse"
        STYLE="position:relative;top:15px;left:25px;height:100px;width:100px;
        border:solid black 1px;
        ">
    <!--
        sich bewegender DIV
    -->
  </DIV>
  <BUTTON onclick=" ID_Animatemotion.beginElement();">Start</BUTTON>
  <BUTTON onclick=" ID_Animatemotion.endElement();">Stop</BUTTON>
  <BUTTON onclick=" ID_Animatemotion.resetElement();">Reset</BUTTON>
</BODY>
</HTML>

```

onresize

erzeugt, wenn Objektgröße verändert wird
nicht für embedded-Objekt

Beispiel: Reload des Dokumentes nach Resize des Browserfensters

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function neuLaden()
    {
        if (document.all)
        {history.go(0);}
    }
// -->
</SCRIPT>
</HEAD>
<BODY onResize=neuLaden();">
</BODY>

```

onresizeend

erzeugt, wenn Änderung der Objektgröße endet

onresizestart

erzeugt wenn Veränderung Objektgröße endet

onresume

erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird
siehe Objekt currTimeState und Behavior .style.time2
auch für body Objekt

Beispiel:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">

```



```

<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  <t:VIDEO ID="ID_Video"
           SRC="test.wmv"
           onpause="ID_Span.innerText = "Pause !"
           onresume=" ID_Span.innerText = "Pause beendet ""
           onmediacomplete="ID_Span.innerText =
                               "
                               'Datei test.wmv wurde komplett geladen !'
  >
  </t:VIDEO>
  <BR>
  <SPAN ID="ID_Span"></SPAN>
</BODY>
</HTML>

```

onreverse

erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird
 (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist)
 .autoReverse muss auf "true" stehen
 siehe Objekt currTimeState und Behavior .style.time2

Beispiel:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  <t:EXCL AUTOREVERSE="true"
         DUR="6"
         REPEATCOUNT="indefinite"
         onreverse="alert('Rückwärts auf der Timeline);"
  >
  <DIV CLASS="time_line_klasse" BEGIN="0" DUR="2">Zeile 1</DIV>
  <DIV CLASS="time_line_klasse" BEGIN="2" DUR="2">Zeile 2</DIV>
  <DIV CLASS="time_line_klasse" BEGIN="4" DUR="2">Zeile 3</DIV>
  </t:EXCL>
</BODY>
</HTML>

```

onrowenter

erzeugt wenn neue Daten verfügbar sind in der aktuellen Spalte
 aufgrund Änderung der Daten in der Datenquelle zu Spalte
 nur für databound-Objekte, die selbst Daten halten (Quelle und Ziel identisch)

onrowexit

erzeugt direkt Spaltenwechsel, also vor dem Verlassen der aktuellen Spalte
 nur für databound-Objekte, die selbst Daten halten (Quelle und Ziel identisch)

onrowsdelete

erzeugt direkt vor dem Löschen von Spalten

onrowsinserted

erzeugt direkt nach dem Einfügen einer Spalte per Methode .AddNew()

onsave

erzeugt, wenn Webseite als Datei gespeichert wird
 oder Webseite als Bookmark in die Favoritenliste eingetragen wird
 oder Webseite verlassen wird

Beispiel

```

<HTML>
<HEAD>
<META NAME="save" CONTENT="favorite">

```



```

<STYLE>
    .saveFavorite { behavior:url(#default#savefavorite);}
</STYLE>
</HEAD>
<BODY>
    <INPUT TYPE=text ID="ID_Input" CLASS=saveFavorite
        onsave="javascript:alert('Event onsave erkannt');"
    >
</BODY>
</HTML>

```

onscroll

erzeugt wenn User die Scrollpfeile benutzt (nicht den Scrollbalken)

onseek

erzeugt wenn eine Seek-Methode zum Element aktiviert wurde:

```

    .seekActiveTime()
    .seekSegmentTime()
    .seekTo()
    .seekToFrame()

```

siehe Objekt currTimeState und Behavior .style.time2

Beispiel :

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
    var FrameAnzahl=600;

    function Suchen()
    {
        // prüfen ob Element nicht aktiv ist
        if (!ID_Video.currTimeState.isActive)
        {
            // nicht aktiv, also starten
            ID_Video.beginElement();
        }
        else
        {
            // prüfen des Eingabewertes
            if( ( isFinite(ID_Input.value) )
                && (ID_Input.value < FrameAnzahl )
                && (ID_Input.value > 0)
            )
            {
                // ist okay, also ab Zeitpunkt animieren
                ID_Video.seekToFrame(ID_Input.value);
            }
            else
            {
                // fehlerhaft
                alert( "Fehler ! Frame-Wert muss > 0 und < "
                    + FrameAnzahl
                    + " sein"
                );
                ID_Input.focus();
            }
        }
    }
</SCRIPT>
</HEAD>
<BODY>
    <SPAN CLASS="time_line_klasse"
        DUR=".01"
        REPEATCOUNT="indefinite"
        FILL="hold"
        onrepeat="innerText=parseInt(ID_Video.currentFrame);"
    >
        0
    </SPAN>

```




```

<BR>
<t:VIDEO          ID="ID_Video"
                  SRC"test.avi"
                  STYLE="width:175px; height:150px;"
                  onmediacomplete="ID_Span.innerText= ID_Video.mediaDur;"
                  onseek="alert('Der Frame ' + ID_Input.value + ' wird eingestellt !)"
>
</t:VIDEO>
<BR>
Dauer des AVI:
<SPAN ID="ID_Span"></SPAN>
&nbsp;  Sekunden
<BR>
setze seekToFrame:
<INPUT  TYPE="text"
        VALUE=""
        NAME="ID_Input"
        SIZE="3"
>&nbsp;  Sekunden
<BR>
<BUTTON onclick="Suchen();">
        Klick fuer Seek
</BUTTON>
<BUTTON onclick=" ID_Video.beginElement()">
        Restart
</BUTTON>
</BODY>
</HTML>

```

onselect

erzeugt wenn etwas selektiert wird
 Beispiel für INPUT:

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
  <!--
        function setzeFokus()
        {
                document.meinFormular.eingabe.focus();
                document.meinFormular.eingabe.select();
        }
  // -->
</SCRIPT>
</HEAD>
<BODY>
  <FORM NAME="meinFormular">
    <INPUT TYPE="text"
          NAME="eingabe"
          VALUE="Eingabe..."
    >
    <INPUT TYPE="button"
          VALUE="Bitte dr&uuml;cken"
          onclick="setzeFokus();">
  </FORM>
</BODY>
</HTML>

```

onselectionchange

erzeugt wenn Selektionsstatus im Dokument verändert wird

onselectstart

erzeugt wenn Objekt selektiert wird

onshow

erzeugt wenn der Media Bar Player gerade sichtbar gemacht wird (nicht versteckt wird)
 siehe Eigenschaft .enabled
 entspricht dem Öffnen des Media Bar Player durch den User
 Media Bar Player ist der Windows Media Player
 siehe Behavior .style.mediaBar



onstart

erzeugt mit Beginn eines jedem Loop-Durchlaufes des des Marquee-Objektes
 Anzahl der Durchläufe laut LOOP-Attribut
 LOOP-Attribut muss Wert > 0 haben, wenn onstart erzeugt werden soll

Beispiel

```
<BODY>
  <MARQUEE BEHAVIOR="alternate" LOOP=2 onstart="alert('onstart-Ereignis erkannt')">
    Marquee Text
  </MARQUEE>
</BODY>
```

onstop

erzeugt wenn STOP-Button im Browser-Menü gedrückt wurde
 wird direkt nach den onbeforeunload Event erzeugt
 wird vor dem onunload Event erzeugt, da die Webseite verlassen wird

Beispiel

```
var TimerID;

function Rekursion()
{
    // nach belieben etwas tun
}

function TimerLoeschen()
{ window.clearInterval(TimerID); }

function Init()
{ TimerID = window.setInterval("Rekursion()",1000); }

document.onstop= TimerLoeschen; // ohne ()
window.onload=Init; // ohne ()
```

onsubmit

erzeugt wenn das Formular gesendet wird
 Eventhandler **muss** einen Rückkehrcode liefern (true oder false) z.B. per return-Anweisung
 Sendeaktion erzeugbar per
 INPUT TYPE="submit"
 BUTTON TYPE="submit"
 Art der Formularaktion laut ACTION-Attribut des Formular-Tag
 Senden unterbindbar, wenn Eventhandler false liefert als Rückkercode (muss programmiert werden)
 z.B. wenn Daten im Formular fehlerhaft vom User eingegeben wurden

Beispiel

```
<BODY>
  <FORM NAME="Formular" onsubmit="return(SubmitEventHandler());"></FORM>
</BODY>
```

onsyncrestored

erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird
 nach vorausgegangenem Abbruch der Synchronisation
 siehe onoutofsync
 sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked"
 siehe Objekt currTimeState und Behavior .style.time2

ontimeerror

erzeugt bei Zeitfehler nur für BODY-Objekt bei benutzung der Timeline

Beispiel

```
<SCRIPT FOR="BODY" event="ontimeerror">
  alert("HTML+TIME error erkannt");
</SCRIPT>
```

ontrackchange

erzeugt wenn der Track einer Advanced Stream Redirector (ASX) Datei (*.asx)
 in der Playliste gewechselt wurde

Beispiel für Aufbau einer ASX-Datei:

```
<ASX Version="1.0" PreviewMode="No" >
  <ENTRY>
    <TITLE>First title</TITLE>
    <AUTHOR>Test</AUTHOR>
    <COPYRIGHT>2002</COPYRIGHT>
```



```

<ABSTRACT>WAV File</ABSTRACT>
<REF href=""> </REF>
<banner href = "first_title.gif" >
  <moreinfo href = "first_title.doc"><moreinfo>
  <abstract>Visit the first abstract Web site</abstract>
</banner>
</ENTRY>
</ASX>

```

siehe Objekt currTimeState und Behavior .style.time2

onunload

erzeugt mit dem direkt vor dem Unload eines Dokumentes

unload = verlassen des Dokumentes durch Laden eines neuen Dokumentes in das selbe Fenster

Beispiel

```

<HEAD>
<SCRIPT FOR=window EVENT=onunload>
  alert("Event onunload erkannt");
</SCRIPT>

<SCRIPT>
  function Umlenken()
  {location.href="test.htm";}
</SCRIPT>
</HEAD>
<BODY>
  <INPUT TYPE=button VALUE="Klicken ... weiter zu TEST.HTM" onclick="Umlenken()">
  <IMG SRC="test.gif">
</BODY>

```

Beispiel für Ermittlung der Verweilzeit des Users auf der Webseite

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
<!--
  var start_zeit;      // muss global sein

  function start()
  {start_zeit = new Date();}

  function ende()
  {
    var ende_zeit = new Date();
    var wert=ende_zeit.getTime() - start_zeit.getTime();
    wert = wert /1000;  // in Sekunden umrechnen
    alert("Verweilzeit in Sekunden: " + Math.floor(wert).toString());
  }
  //-->
</SCRIPT>
</HEAD>

<BODY ... onLoad="start();" onUnload="ende();">
</BODY>
<HTML>

```

onURLFlip

erzeugt wenn ein Scriptkommando, das in einer Advanced Streaming Format (ASF)-Datei (*.asf) liegen, ausgeführt wird

siehe Objekt currTimeState und Behavior .style.time2

Hinweis: window.event.URL Url laut einem Kommando aus einer Advanced Streaming Format (ASF)-Datei von Element auf der Timeline
es muss Ereignis onURLFlip aufgetreten sein
siehe Objekt currTimeState und Behavior .style.time2



*.asf 27
 *.asx 26
 .autoReverse 23
 .begin 21
 .enabled 12, 25
 .end 10
 .endElement() 10
 .fill 9
 .openState 18
 .repeat 21
 .repeatCount 5, 9, 21, 23
 .resetElement() 18, 21
 .seekActiveTime() 24
 .seekSegmentTime() 24
 .seekTo() 24
 .seekToFrame() 24
 .setData() 7
 .style.mediaBar Behavior 12, 18, 20, 25
 .style.time2 Behavior 5, 10, 14, 18, 19, 21, 22, 23, 24, 26, 27
 .syncBehavior 26
 .URL 27
 Advanced Stream Redirector Datei 26
 Advanced Streaming Format (ASF)-Datei 27
 Advanced Streaming Format Datei 27
 Aktivierung eines Objektes 3
 ASF 27
 ASF-Datei 27
 ASX 26
 ASX-Datei 26
 asynchrones Senden von Daten 8
 Behavior .style.mediaBar 12, 18, 20, 25
 Behavior .style.time2 5, 10, 14, 18, 19, 21, 22, 23, 24, 26, 27
 body Objekt 19, 22
 Bookmark 23
 Clipboard 3, 4, 7
 Clipboard hinzufügen 7
 Copy 7
 CTRL-C 7
 currTimeState Objekt 5, 10, 14, 18, 19, 21, 22, 23, 24, 26, 27
 Daten asynchron senden 8
 Deaktivierung Objekt 4, 8
 Dokument als Datei speichern 23
 Dokument entladen 4
 Dokument Unload 4, 27
 Dokument verlassen 4, 23
 Download vom Image 2
 Drag & Drop 9
 Druck 4
 Druck Ende 2
 Druckvorschau 4, 13
 Druckvorschau Ende 2
 Eigenschaft Objekt ändern 20
 element Objekt 6, 8, 9, 20
 Ende des Druck 2
 Ende Druckvorschau 2
 entladen eines Dokumentes 4
 event Objekt Url laut einem Kommando aus einer Advanced
 Streaming Format (ASF) Datei 27
 F1 Hilfe 12
 Favoritenliste 23
 Focus 12
 Formular reset 21
 Formular senden 26
 Formular submit 26
 Formular zurücksetzen 21
 freeze 9
 getTime() 13, 27
 Gif-Bild 14
 Größe Objekt 22
 Hilfe F1 12
 hinzufügen zum Clipboard 7
 history.go(0) 22
 hold 9

HTC-Datei 6, 8, 9, 20
 HTML-Dokument als Datei speichern 23
 HTML-Dokument verlassen 23
 IE Reload des Dokumentes nach Resize des Browserfenster 22
 Image Download 2
 Kontextänderung eines Objektes 6
 Kontextmenü 7
 locked 26
 LOOP 5, 11, 26
 Math.floor() 13, 27
 Mausrad 15
 Mausüberwachung 13
 Media Bar Media-Datei Wiedergabe Statusänderung 19
 Media Bar Player anzeigen 12, 25
 Media Bar Player Media-Datei Wiedergabe Statusänderung 19
 Media Bar Player Statusänderung bezüglich Codec 18
 Media Bar Player Statusänderung bezüglich Individualisierung 18
 Media Bar Player Statusänderung bezüglich Lizenz 18
 Media Bar Player Statusänderung bezüglich Playliste 18
 Media Bar Player Statusänderung bezüglich Wiedergabe 19
 Media Bar Player Wiedergabe Statusänderung 19
 Media Bar Player Windows Media Player 12, 18, 20, 25
 Media-Element 13, 14
 Objekt Aktivierung 3
 Objekt bewegen 15
 Objekt body 19, 22
 Objekt currTimeState 5, 10, 14, 18, 19, 21, 22, 23, 24, 26, 27
 Objekt Deaktivierung 4, 8
 Objekt Eigenschaft ändern 20
 Objekt element 6, 8, 9, 20
 Objekt event Url laut einem Kommando aus einer Advanced
 Streaming Format (ASF) Datei 27
 Objekt Kontextänderung 6
 Objekt Mausüberwachung 13
 Objekt Status 20
 Objektgröße 22
 onabort 2
 onactivate 2
 onafterprint 2
 onafterupdate 2
 onbeforeactivate 3
 onbeforecopy 3
 onbeforecut 3
 onbeforedeactivate 4
 onbeforeeditfocus 4
 onbeforepaste 4
 onbeforeprint 4
 onbeforeunload 4
 onbeforeupdate 5
 onbegin 5, 10, 20
 onblur 5
 onbounce 5
 oncellchange 6
 onchange 6
 onclick 6
 oncontentready 6
 oncontentsave 6
 oncontextmenu 7
 oncontrolselect 7
 oncopy 7
 oncut 7
 ondataavailable 8
 ondatasetchanged 8
 ondatasetcomplete 8
 ondblclick 8
 ondeactivate 8
 ondetach 8
 ondocumentready 9
 ondrag 9
 ondragend 9
 ondragenter 9
 ondragleave 9
 ondragover 9



- ondragstart 9
- ondrop 9
- onend 5, 9, 21
- onerror 10
- onerrorupdate 11
- onfilterchange 11
- onfinish 11
- onfocus 11
- onfocusin 12
- onfocusout 12
- onhelp 12
- onhide 12
- onkeydown 12
- onkeypress 12
- onkeyup 12
- onlayoutcomplete 13
- onload 13
- onlosecapture 13
- onmediacomplete 13, 14
- onmediaerror 14
- onmousedown 14
- onmouseenter 14
- onmouseleave 15
- onmousemove 15
- onmouseout 15
- onmouseover 15
- onmouseup 15
- onmousewheel 15
- onmove 15
- onmoveend 16
- onmovestart 17
- onopenstatechange 18
- onoutofsync 18
- onpaste 18
- onpause 19
- onplaystatechange 19
- onpropertychange 20
- onreadystatechange 20
- onrepeat 5, 10, 20
- onreset 21
- onresize 22
- onResize 22
- onresizeend 22
- onresizestart 22
- onresume 22
- onreverse 23
- onrowenter 23
- onrowexit 23
- onrowsdelete 23
- onrowsinserted 23
- onsave 23
- onscroll 24
- onseek 24
- onselect 25
- onselectionchange 25
- onselectstart 25
- onshow 25
- onstart 26
- onstop 26
- onsubmit 26
- onsyncrestored 18, 26
- ontimeerror 26

- ontrackchange 26
- onunload 27
- onURLFlip 27
- Reload des Dokumentes nach Resize des Browserfenster IE 22
- Scrollbar 24
- Scrollpfeile 24
- Selektion 3, 25
- speichern Webseite 23
- Status Objekt 20
- STOP-Button 26
- streaming media file 14
- Tastatur 12
- Timeline 26
- Timeline Abbruch der Synchronisation 26
- Timeline aktives Element auf der aktiven Timeline beginnt gerade zu pausieren 19
- Timeline aktives Element pausiert 19
- Timeline aktivieren 5
- Timeline beenden 9
- Timeline Element auf der Timeline aktivieren 5
- Timeline Element beenden auf der Timeline 9
- Timeline Element initialisieren 21
- Timeline Element mit seiner Timeline synchronisieren 26
- Timeline Element nicht synchron zur seiner Timeline 18
- Timeline Element Pause aufheben 22
- Timeline Element rückwärts auf der Timeline animieren 23
- Timeline Element stoppen auf der Timeline 9
- Timeline Element verliert seine Timeline 18
- Timeline Element Wiederholung der Animation 20
- Timeline Element zurücksetzen 21
- Timeline Ende der aktiven Timeline erreicht 9
- Timeline Event Seek-Methode 24
- Timeline initialisieren 21
- Timeline Media-Element 13, 14
- Timeline Medium zum Element ist komplett geladen 13
- Timeline Medium zum Element ist wegen Fehler nicht geladen 14
- Timeline Pause aufheben 22
- Timeline Pause eines aktiven Elementes aufheben 22
- Timeline rückwärts 23
- Timeline Scriptkommando 27
- Timeline Seek-Methode zum Element 24
- Timeline stoppen 9
- Timeline streaming media file 14
- Timeline Synchronisation Abbruch 26
- Timeline synchronisieren 26
- Timeline Trackwechsel 26
- Timeline Url laut einem Kommando aus einer Advanced Streaming Format (ASF) Datei 27
- Timeline Wiederholung der Animation des aktiven Elementes 20
- Timeline zurücksetzen 21
- Timeline Zwangssynchronisierung 18, 26
- Timeline pausieren 19
- Unload eines Dokumentes 4, 27
- verlassen Dokument 23
- verlassen eines Dokumentes 4
- Video 14
- Webseite als Datei speichern 23
- Webseite verlassen 23
- window.onload 6
- Windows Media Player 12, 18, 20, 25
- Zeitfehler 26
- Zwangssynchronisierung 18, 26

