

Q-108.144 - Installation und Betrieb eines JavaScript TLDs

Q-108.144 - Installation und Betrieb eines JavaScript TLDs

aus Tobit KB importiert - vom 04.05.2006 - [Beitrag bei Tobit](#)

Problem

Ein JavaScript TLD soll installiert und eingerichtet werden.

Antwort

Nachfolgend wird die Einrichtung eines Extended TLDs für den Einsatz von JavaScript erläutert. Als Beispiel dient hierbei die Ansteuerung eines Panasonic Video Projektors. Führen Sie die beschriebenen Schritte in der hier angegebenen Reihenfolge durch.

Einrichten eines Extended Ports

Für den Betrieb eines Extended TLDs sollte ein separater Port verwendet werden. Gehen Sie dazu folgendermaßen vor:

- **Starten des Port Setups**

Öffnen Sie im David Administrator das Menü »Konfiguration -> Ports« und klicken Sie dann in der Symbolleiste auf das Icon »Neu«. Daraufhin startet das Setup-Programm. Klicken Sie auf »Hinzufügen«, um einen neuen Port einzurichten.

- **Bestimmen des Port-Typs**

Wählen Sie aus der dann angezeigten Liste den Typ »Extended (z.B. Automation)« und bestätigen Sie Ihre Auswahl mit »Weiter«.

- **Auswahl des Plugins**

Im nächsten Dialogfenster ist das für den neuen Port zu nutzende PlugIn anzugeben. Da ein JavaScript Port erstellt werden soll, wählen Sie an dieser Stelle den Eintrag »JavaScript« aus. Klicken Sie anschließend auf »Weiter«, um Ihre Auswahl zu bestätigen und mit dem Setup fortzufahren.

- **Benennen des Ports und Abschluss des Setups**

Vergeben Sie schließlich eine eindeutige Bezeichnung für den neuen Port (z.B. »Panasonic Beamer«) und klicken Sie auf »Fertig stellen«, um das Setup abzuschließen. Falls daraufhin eine Meldung erscheint, die besagt, dass Ergänzungen der Windows-Registry-Datenbank erforderlich sind, klicken Sie auf »Ja«, damit diese Anpassungen vom Port-Setup automatisch durchgeführt werden. Damit ist die Port-Installation beendet und das Setup-Programm schließt sich eigenständig.

Anpassen der Port-Konfiguration

Nach der Installation des neuen Ports müssen einige Anpassungen in dessen Konfiguration vorgenommen werden. Wählen Sie dazu im David Administrator den Menüpunkt »Konfiguration -> Ports« und öffnen Sie den Konfigurationsdialog des soeben erstellten Ports durch einen Doppelklick auf den zugehörigen Listeneintrag. Nehmen Sie auf der Registerkarte »Allgemein« folgende Einstellungen vor:

- **TLDSubType**

Tragen Sie hier eine Bezeichnung ein, unter der dieser Port direkt adressierbar sein soll. Die Verwendung dieses TLDSubType ist sowohl im Adressfeld des Nachrichten Editors als auch in Verbindung mit DVISE Befehlen (z.B. @@NUMMER@@) möglich. Geben Sie als Empfängeradresse zunächst diese Bezeichnung an, direkt gefolgt von einem Doppelpunkt und der

eigentlichen Empfängeradresse.

- **Erweiterung der Sendeservices**

Aktivieren Sie die Funktion »Diesen Port mit der folgenden Bezeichnung im Nachrichten Editor des InfoCenters unter »Sendeservices« zur Auswahl anbieten«, wenn Sie wünschen, dass dieser Port über einen eigenen Eintrag in der Liste »Sendeservices« im Nachrichten Editor des David InfoCenters direkt ansprechbar sein soll. Vergeben Sie dazu in dem darunter stehenden Feld »Bezeichnung« einen Namen, der den verfügbaren Sendeservices hinzugefügt werden soll. Bei der Erstellung von Aufträgen, die über diesen Port abgewickelt werden sollen, ist dann als Sendeservice dieser Eintrag zu wählen.

Einbinden des JavaScript Codes

Erstellen Sie den auszuführenden JavaScript Code und speichern Sie ihn unter dem Dateinamen TLDX.JS auf dem David Server in dem Port Verzeichnis des zuvor erstellten Extended Port ab (z.B. DAVID\TLD\PORT\002, für den Fall dass es sich hierbei um den Port mit der Nummer 002 handelt). Starten Sie anschließend den zugehörigen TLD neu, um den Code zu laden. Öffnen Sie dazu im David Administrator das Dialogfenster »DvISE Services« durch einen Klick auf die zugehörige Schaltfläche in den Symbolleisten. Markieren Sie dann in der Liste der DvISE Services den TLD des betreffenden Ports (also beispielsweise »TLD 002« für den Port mit der Nummer 002) und klicken Sie auf »Neu starten«.

Betrieb des JavaScript Ports

Nach dem Speichern der gewünschten JavaScript Datei und dem anschließenden Neustart des TLDs ist der Port betriebsbereit. Aufträge für diesen Port können per Text-Datei im ASCII-Format über die API- oder über die FileScan-Services erzeugt werden, wobei in der Adressierung der in der Port-Konfiguration definierte TLDSubType zu verwenden ist.

Die Erstellung von Aufträgen ist natürlich auch über den Nachrichten Editor des David InfoCenters möglich. Hier geschieht die Adressierung komfortabel über die Auswahl des Sendeservices, der ebenfalls zuvor in der Port-Konfiguration definiert wurde.

Port Monitor

Sobald der Port Aufträge bearbeitet, kann die System-Kommunikation im Port Monitor überwacht werden. Wählen Sie dazu im David Administrator den Menüpunkt »Monitor -> Ports« und öffnen Sie den Monitor des gewünschten Ports per Doppelklick auf den zugehörigen Listeneintrag. Wechseln Sie hier auf das Register »Communication« um den Verlauf der Kommunikation zu verfolgen.

Beispiele

Hier finden Sie einige Beispiel-Listings für den Einsatz von JavaScript. Speichern Sie den Code unter dem Namen TLDX.JS im Port Verzeichnis des Port ab, um ihn zu nutzen. Zur Aktivierung muss anschließend der zugehörige TLD neu gestartet werden.

Standard COM-Port-Objekt

Die unten stehenden Listings sind Beispiele für eine synchrone COM-Port-Übertragung. Beachten Sie hierbei folgende allgemeine Informationen:

- Um eine synchrone COM-Port-Übertragung einzurichten, sollte ein entsprechendes Objekt mit der Funktion »TLD.CreateObject("TLDX.ScriptComm")« erzeugt werden.
- Die Funktion "OnSendMessage()" wird bei jeder über diesen Port ausgehenden Nachricht ausgeführt.
- Die Funktion »OnReceiveMessage()« wird sekundlich ausgeführt.
- Die Funktion »OnExit()« wird beim Entladen des TLD aufgerufen.
- Die Funktion »TLD.ODS(Text)« gibt den als Parameter übergebenen Text direkt im Port Monitor aus.
- Über das Objekt »TLD.SR« kann z.B. auf die »From«, »To« und »Subject« Daten der zu versendenden Nachricht zugegriffen werden.

Beispiel 1

In diesem Beispiel wird beim Starten des Ports der COM-Port initialisiert. Der Rückgabewert wird auf dem

Portmonitor ausgegeben. Wird eine Nachricht versendet, erfolgt im Port Monitor die Ausgabe der Von- und An-Informationen sowie des Betreffs der Nachricht. Im Anschluss an den Versand wird überprüft, ob Daten über den COM-Port empfangen wurden, und diese gegebenenfalls auch im Port Monitor angezeigt. Über die Funktion OnExit() wird in diesem Beispiel der COM-Port wiederfreigegeben.

```
var oCom = TLD.CreateObject("TLDX.ScriptComm");
var Port = 1;
var Baud = 9600;
var DataBits = 8;
var StopBits = 0; // 0 = 1 StopBit, 1 = 2 StopBits
var Parity = 0; // 0 = none Parity
var RetVal = 0;

RetVal = oCom.OpenPort(Port, Baud, DataBits, Parity, StopBits);
TLD.ODS("OpenPort returned "+ RetVal + "\n");

function OnSendMessage()
{
  TLD.ODS("-----Send Message-----" + "\n");
  MsgOut = "From: "+TLD.SR.From+"\n To: "+
  TLD.SR.To+"\n Subject: "+ TLD.SR.Subject+"\n";
  TLD.ODS(MsgOut);
  RetVal = oCom.WriteString(TLD.SR.Subject + "\r\n");
  TLD.ODS("Result: " + RetVal + "\n");
  TLD.ODS("-----Receive Message-----\n");
  RetVal = oCom.ReadString();
  TLD.ODS("Receive String: " + RetVal + "\n");
}

function OnExit()
{
  • oCom.ClosePort();
    TLD.ODS("Ende...\n");
}
```

Beispiel 2

Dieses Beispiel greift, wie das erste Beispiel, auf die serielle Schnittstelle zu. Es wurde speziell für die Steuerung von Panasonic Video Beamern geschrieben (Modelle PT-D7600/D7500, PT-D7700/DW7000, PT-D5500). Der Betreff einer über diesen Port gesendeten Nachricht wird ausgewertet und als spezielle Befehlskette an den Beamer gesendet. Der Betreff muss also den gewünschten Befehl und gegebenenfalls erforderliche Parameter enthalten (z.B. PON für Power on, IIS:DVI für DVI Quellenwahl oder VBR:63 für volle Helligkeit).

```
var oCom = TLD.CreateObject("TLDX.ScriptComm");
var Port = 1;
var Baud = 9600;
var DataBits = 8;
var StopBits = 0; // 0 = 1 StopBit, 1 = 2 StopBits
var Parity = 0; // 0 = none Parity
var RetVal = 0;

RetVal = oCom.OpenPort(Port, Baud, DataBits, Parity, StopBits);
TLD.ODS("OpenPort returned "+ RetVal + "\n");

function OnSendMessage()
{
```

```

TLD.ODS("-----Send Message-----" + "\n");
MsgOut = TLD.SR.Subject + "\n"
TLD.ODS(MsgOut);
RetVal = oCom.WriteString(unescape("%02") + "ADZZ;" + TLD.SR.Subject
+ unescape("%03"));
TLD.ODS("Send-Result: " + RetVal + "\n");
TLD.ODS("-----" + "\n");}

function OnExit()
{
    • oCom.ClosePort();
      TLD.ODS("Ende...\n");
}

```

Asynchrones COM-Port Objekt

Dieses Listing dient als Beispiel für eine asynchrone COM-Port-Übertragung. Hier gelten folgende allgemeinen Hinweise:

- Um eine asynchrone COM-Port-Übertragung einzurichten, sollte ein entsprechendes Objekt mit der Funktion »TLD.CreateObject("TLDX.ScriptAsyncComm")« erzeugt werden.
- Um das Überwachen der Schnittstelle zu starten, wird die Funktion »oCom.StartCommReadEvent()« aufgerufen.
- Die Funktion »OnCommData (data)« wird automatisch aufgerufen, sobald Daten an der seriellen Schnittstelle anliegen; der ankommende Daten-String wird als Parameter übergeben.
- Die Funktion »OnSendMessage()« wird bei jeder über diesen Port ausgehenden Nachricht ausgeführt.
- Die Funktion »OnExit()« wird beim Entladen des TLD aufgerufen.
- Die Funktion »TLD.ODS()« gibt den als Parameter übergebenen Text im Port Monitor aus.
- Über das Objekt »TLD.SR« kann z.B. auf die »From«, »To« und »Subject« Daten der zu versendenden Nachricht zugegriffen werden.

Beispiel 1

```

var oCom = TLD.CreateObject("TLDX.ScriptAsyncComm");
//Asynchrones COMPort Objekt erzeugen

var PortState = oCom.OpenPort(Port, Baud, DataBits, Parity, StopBits);
//COM Port öffnen

oCom.StartCommReadEvent();
// Event Funktion starten

function OnCommData (Text)
{TLD.ODS ("OnCommData: Daten empfangen:\n");
TLD.ODS (Text);
// hier kann jetzt reagiert werden.
// z.B.:
oCom.WriteString ("OK\n");}
function OnExit()
{
    • oCom.ClosePort();
      TLD.ODS("Ende...\n");
}

```

Beispiel 2

Dieses Beispiel zeigt, wie durch Nutzung der Funktion »AddRXMessage« Daten, die über das JavaScript-TLD empfangen wurden, direkt ins State Archive geschrieben werden können. Hierbei gilt folgende Syntax:

```
TLD.AddRXMessage ("subject", "from", "value", icon);
```

Parameter:

- subject - Text für Betreff
- from - Absenderangabe (Adresse des Gerätes)
- value - Wert, der gesetzt werden soll
- icon - Nummer des zu verwendenden Symbols (nicht in "")

```
function Test ()
```

```
{
```

- TLD.ODS ("Start\n");
if (TLD.AddRXMessage ("Dies ist das Subject", "hier das From", "100", 217) == 0)
{
 - TLD.ODS ("Senden erfolgreich\n");
}
else
{
 - TLD.ODS ("Senden nicht erfolgreich\n");
}

```
if (TLD.AddRXMessage ("Dies ist Subject Nr. 2", "noch ein From",  
"200", 218) == 0)
```

```
{
```

- TLD.ODS ("Senden erfolgreich\n");

```
}
```

```
else
```

```
{
```

- TLD.ODS ("Senden nicht erfolgreich\n");

```
}}
```

```
Test ();
```

Hinweis:

Beachten Sie auch das Anwendungsbeispiel im [KnowledgeBase Artikel Q-108.288](https://www.david-forum.de/wiki/entry/4348-q-108-144-installation-und-betrieb-eines-javascript-tlds/).