

Beim Start einer Forms-Web-Applikation gibt es verschiedene Möglichkeiten, diese wieder wie eine Desktop-Applikation aussehen zu lassen: SeparateFrame, JavaScript, WebStart, mshta und Prism.

Forms als Desktop-Applikation

Michael Tietz, imining gmbh

Beim Umstieg auf Forms im Web gibt es einige Gründe, die Forms-Applikation nicht mit einem Browser zu starten, sondern wieder die gewohnte Desktop-Applikation anzubieten. Das war zum einen der Wunsch der Anwender nach ihrem Programm, vor allem aber bei FullScreen-Applikationen im industriellen Umfeld eine Platzfrage, denn Menü, URL etc. des Browsers schieben die Applikation Zeile für Zeile in den scrollbaren Bereich. Immer wieder zwei Forderungen: Die Windows-Taskleiste ist beim Maximieren zu berücksichtigen und die Forms-Konsole muss wegen der Fehlerausgabe immer sichtbar sein.

Der in der Konfiguration (formsweb.cfg) angebotene Parameter „separateFrame=true“ löst das Problem nur zum Teil. Er schafft zudem ein neues Problem: Der Frame ist ohne sein Fenster nicht lauffähig und der Anwender ist auf Dauer nicht zu motivieren, das leere Browser-Fenster offen zu lassen.

Als Lösung wurde ein kleines Java-Skript zum Start benutzt (Test: Win2000, IE 6, JRE 1.3, Forms 9i): Browser öffnen, Window öffnen, etwas warten (hier fünf Sekunden), Browser schließen (siehe Listing 1, Seite 35).

Mit Aufkommen der Registerkarten im Browser und der Microsoft-Frage „Was wollen Sie wirklich schließen?“ mussten andere Wege gesucht werden, um nicht in die Abhängigkeit von lokalen Browser-Einstellungen zu gelangen.

Mit JRE 1.4.2 (Test: WinXP, IE 7, JRE 1.6, Forms 10gR2) kam die Web-Start-Technologie ins Gespräch: „To manage applications right off the Web“, dazu von Oracle der Hinweis: „WebStart is not supported with Forms“.

Für manche Applikationen kann der Start über die *.jnlp-Datei eine Alternative darstellen (siehe Listing 2). Es treten aber einige Unannehmlichkeiten auf: So wird zum Beispiel

bei modalen Fenstern automatisch das Menü des Main-Windows vererbt. Dabei entsteht eine Scrollbar und der Inhalt des Fensters wird nach unten geschoben – schlecht, wenn dort der OK- und Cancel-Button liegen. Unter Forms11.3 bricht der Start leider mit FRM-92052 ab. Gelegentlich hilft aber auch ein Blick über den Forms-Tellerand.

Hinter dem Windows-Exe verbirgt sich Microsofts Skript-Interpreter und auch der kann JavaScript. Die zugehörige Datei-Extension heißt „*.hta“ als

Abkürzung für „HTML Application“ (siehe Listing 3, Test: WinXP, IE 8.0.6, JRE 1.6, Forms 11.3).

Die Parameter sind selbsterklärend, einzig der Parameter „border“ fällt aus dem Rahmen. Hinter „thick“ verbirgt sich die Resize-Möglichkeit des Windows im Gegensatz zu „thin“. Der zugehörige Windows-Prozess taucht auch als mshta.exe auf und nicht als iexplorer.exe. Erwähnt sei noch die Möglichkeit, den IE 8 mit der Option „-k“, dem Kiosk-Mode, zu starten. Damit erscheint ein maximiertes Fenster ohne

```
<jnlp spec="1.0+" codebase="http://myserver:myport/forms/java/"
href="start_forms.jnlp">

<information>
<title>Myform</title>
<vendor>Mycompany/vendor>
<homepage href="http://www.Myweb.com"/>
<description>Mydesc</description>
</information>

<resources>
<j2se version="1.6+" />
<jar href="http://myserver:myport/forms/java/frmall.jar"/>
</resources>

<applet-desc main-class="oracle.forms.engine.Main" name="Form"
width="1024" height="768">
<param name="width" value="1024" />
<param name="height" value="768" />
<param name="separateFrame" value="false" />
<param name="lookAndFeel" value="oracle" />
<param name="splashScreen" value="no" />
<param name="colorScheme" value="red" />
<param name="serverURL" value="http://myserver:myport/
frmservlet?ifcfs=http://myserver:myport/forms/
frmservlet?config=webutil" />
<param name="logo" value="no" />
<param name="serverArgs" value="module=myform.fmx usesdi=yes use-
rid= sso_userid=
debug= buffer_records= debug_messages= array=
query_only= quiet= render=
host= port= record= tracegroup= log= " />
</applet-desc>

</jnlp>
```

Listing 2: start_forms.jnlp

```

<HTML>
<HEAD><TITLE>WebUtil</TITLE></HEAD>

<BODY %HTMLbodyAttrs%>
<script language="JavaScript">

window.opener = top;
window.document.write("Dieses Browser Fenster wird automa-
tisch geschlossen");

wPar = "toolbar=0,menubar=no,location=0,personalbar=0,statu
s=0,scrollbars=0,resizable=1,top=0,left=0,width="+screen.
width)+",height="+screen.height)+"";
w = window.open("http://myserver:myport/forms/frmservlet?con
fig=webutil","Forms",wPar);
w.focus();

window.setTimeout("self.close()",5000);

</script>
<body>
</body>
</HTML>

```

Listing 1: start_forms.htm

```

<html>
<head>
<hta:application
id="htaapp"
applicationname="MyApplication"
border="thick"
borderstyle="static"
caption="yes"
contextmenu="yes"
icon="C:\My.ico"
innerborder="no"
maximizebutton="yes"
minimizebutton="yes"
navigable="yes"
scroll="auto"
scrollflat="yes"
selection="yes"
showintaskbar="yes"
singleinstance="yes"
systemu="yes"
version="2.3"
windowstate="maximize"
>
<title> My Application </title>

<script language="JavaScript">
window.location = "http://
myserver:myport/forms/frmservlet?config=webutil","Forms","to
olbar=0,menubar=no,location=0,personalbar=0,status=0,scrollb
ars=0,resizable=1,width=100%,height=100%";
</script>

</head>
<body>
</body>
</html>

```

Listing 3: start_forms.hta

URL, Buttons etc., aber der Desktop ist nach dem Start nicht mehr erreichbar – was für industrielle Anwendungen durchaus ein lang gehegter Wunsch ist. Es sei darauf hingewiesen, dass der Skript-Interpreter einige Sicherheitsstandards des Browsers umgeht. Für den Firefox stellt sich die Lösung noch einfacher dar.

Prism

Prism wurde als Mozilla-WebRunner entwickelt und basiert jetzt auf Firefox. Die Standalone-Installation war aber lange Zeit auf die Installation mit Opera ausgelegt und erforderte einige Handarbeit. Prism als Firefox-Extension ist ein Add-on (Test: Firefox 3.6.13, Prism 1.0), das nach der Installation unter „Extras“ einen Menüpunkt „Convert Website to Application“ enthält. Der folgende Dialog erzeugt eine webapp.ini-Datei mit den Konfigurationsdaten unter „C:\Doku...\User\Anwend...\WebApps\OracleFusion“. Dies ist die einfachste Lösung, wenn man die Suche nach Parametern und den entsprechenden Wertevorräten der einzelnen Skripts umgehen will. Aber auch hier lassen sich Hinweise darauf finden, dass einige Sicherheitsstandards des Browsers nicht berücksichtigt sind. Abschließend noch ein Hinweis von Oracle Support zur weißen Zeile im Kopf der Forms-Applikation im Firefox: Ersetze in webutiljpi.htm in der Applet-Definition die Zeile „VSPACE="0"“ durch:

```

VSPACE="0"
STYLE="position:absolute;top:0p
x;left:0px;border-style:none">

```

Michael Tietz
imining gmbh
michael.tietz@imining.de

