Security im Forms/Reports Umfeld ohne SSO

Jan-Peter Timmermann
PITSS GmbH Stuttgart
Hamburg

Schlüsselworte
Forms, Reports, SSO, Security, Weblogic , WLS_FORMS, WLS_REPORTS.

Einleitung
Immer wieder werden auch in der heutigen Zeit noch Umgebungen mit Oracle Forms oder Oracle Reports installiert und Produktiv genutzt. Für viele Anwender war es in der Oracle Forms/Reports Client Server Umgebung normal, das sie sich gar nicht angemeldet haben sondern automatisch an der Datenbank zum arbeiten angemeldet wurden. In der Oracle Fusion Middleware auf Basis der Web-Technologie ist das leider so nicht mehr möglich. Wie kann man nun die automatische Anmeldung durchreichen ohne auf die Oracle Komponenten Single-Sign-On (SSO) zurückgreifen zu müssen. Ebenso will dieser Vortrag versuchen aufzuzeigen wie man den Report –Server im Zusammenspiel mit Oracle Forms und Reports absichern kann.

Wie sieht das bisher aus S

Für alle Forms/Reports Umgebungen in der Version 10g gab es immer die Möglichkeit, seine Umgebung durch das OID von Oracle zu schützen. Der Benutzer hat über die URL seine Anwendung aufgerufen und das Forms Servlet hat die Anfrage direkt an den SSO-Server von Oracle weiter geleitet. Über die Eingabe von Benutzer Namen und Password in der SSO - Login - Maske hat der Benutzer sich gegen den LDAP Server ausgewiesen und der LDAP - Server lieferte dann die Datenbank Verbindung.

Es wird ein SSO-Cookie gesetzt der für die Dauer der Browser Verbindung aktiv bleibt.

SSO im OAS 10g

In der Anfangsphase des Applikation Servers 10g war dies auch nur die einzige Möglichkeit Forms/Reports zu installieren. Die Installation wurde von Oracle vorgegeben, in dem man zuerst die Infrastruktur installierte, diese konfigurierte und danach die Middletier nachschob. Danach standen einem die SSO-Seiten von Oracle in der Version 10g zur Verfügung, so dass hier eine Absicherung der Forms und Reports Umgebung per default implementiert war. Es mussten nur noch die Benutzer mit ihren RAD's im OID hinterlegt werden und schon konnten alle Anwender per SSO mit dieser Umgebung arbeiten. Die Installation dieser Komponenten war in der Version 10g bereits von Oracle vorgegeben. Bei der Installation des Oracle Applikation Server's musste erst die Infrastruktur mit dem notwendigem Repository installiert werden und anschließend der Applikation Server. Oracle hat einen hier sehr stark geführt.

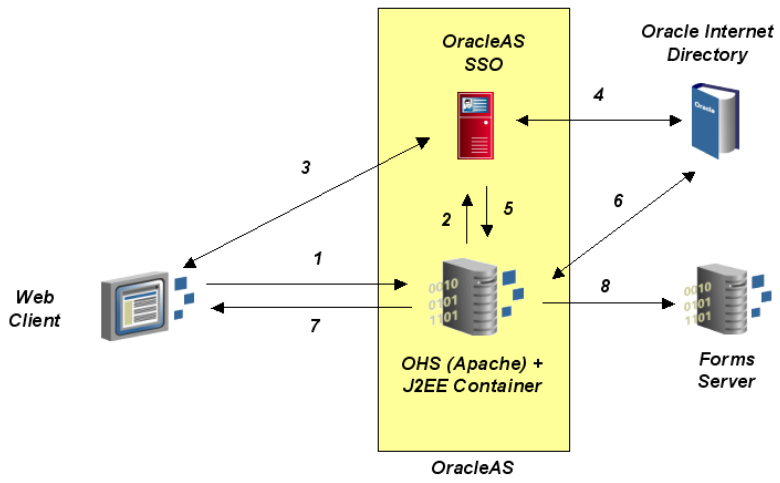


Abb. 1: Auswahl der Installationen unter 10gR2 Installation Guide Oracle

Nachteilig war die Situation, das in der Version 10g es nicht möglich war eine direkte Benutzung des Oracle Access Managers zu tätigen.

Diese Art der Installation gibt es unter 11g nicht mehr. Hier ist der Administrator aufgefordert alle notwendigen Komponenten individuell selber zu installieren.

Ich habe hier einmal alle Komponenten aufgeführt, die wir bei einer Installation benötigt haben.

Database	**Für das Repository**
RCU	Zum erstellen des Repository's
Weblogic	Für die Laufzeit Umgebung der Applikation Server
Patch for OID:	Notwendig für 11gR2 / Linux
SOA (for OIM):	
OIM	
WebTier	
Patch for WebTier	
Forms 11.1.2.0.0:	Unsere eigentliche Forms Umgebung
Patch for OHS_Forms	

In der Version 11g ist es recht Komplex geworden, eine OFM (Forms/Reports) in einer Single-Sign-On Umgebung laufen zu lassen, zumal es auch noch Abhängigkeiten gibt was die Versionen betrifft. Das hat viele Anwender, die einmal die Enterprise Edition des Applikation Servers genutzt haben, davon abgehalten hier die „Große“ Installation zu tätigen. Aus diesem Grunde sind viele dazu übergegangen erst einmal gar nichts zu machen.

Aber gar nichts zu machen ist nun mal nicht immer richtig.

Was gilt es nun alles abzusichern

Bei der Grundinstallation einer Oracle Forms Reports Umgebung werden folgende Komponenten automatisch installiert und halb automatisch konfiguriert.

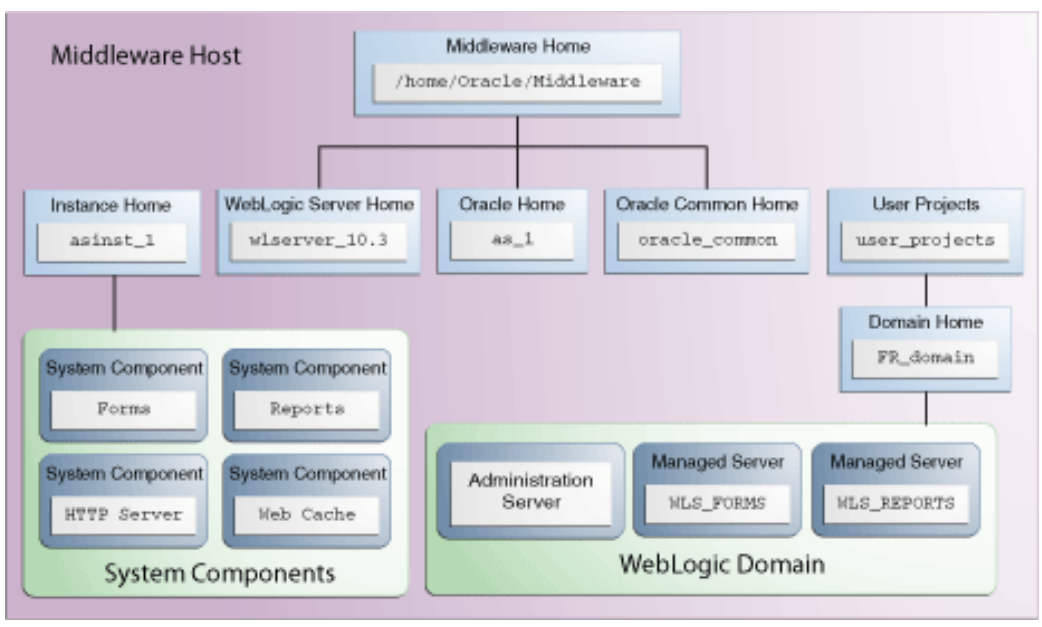


Abb.2 Quick Installation Guide for Oracle Forms and Reports

Es sind hier Komponenten installiert, wie Webcache, http-Server und diverse managed Server innerhalb einer Domaine.

Wenn man sich nun einmal im ersten Schritt anschaut wie man auf die einzelnen Komponenten zugreift, hat man gleich den ersten Ansatz.

Im Umfeld von einer Weblogic Domaine ist es normal, das der Administrations- Server unter dem Rechner Namen und dem Port 7001 über http erreichbar ist. Dies sollte man als erstes ändern. Dazu hat man die Möglichkeit, über die Konsole für den Administrations-Server den Port auf SSL zu ändern. Dies kann mit einem eigenem Zertifikat erfolgen.

Die beiden Server WLS_FORMS sowie WLS_REPORTS sind Standardmäßig über den Rechner Namen sowie die Ports 9001 und 9002 zu erreichen. Auch hier hätte man die Möglichkeit diese auf SSL zu ändern. Ich persönlich würde diese aber auf Lokalhost einstellen um sicher zu stellen, das die Kommunikation mit diesen Servern, nur über den http-Server erfolgen kann.

Den http-Server wiederum sollte man dann über den SSL-Port laufen lassen.

Den Administrations-Server sollte ich dann auch nur auf der lokalen Maschine laufen lassen um zu gewährleisten, das von außen kein unberechtigter Zugriff erfolgen kann. Was ebenso möglich wäre ist den Einstieg auf die „Konsole“ zu ändern. Dies ist recht einfach über die „Konsole“ möglich. Somit besteht schon mal hier eine höhere Schwierigkeit auf den Administrations-Server zuzugreifen.

Starten der Forms Anwendung

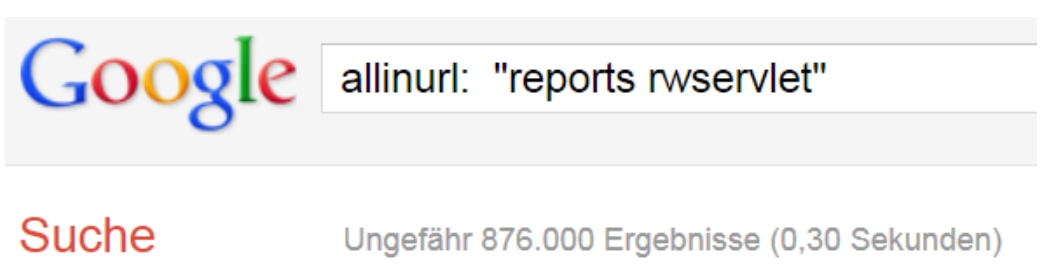
Was für Möglichkeiten gibt es nun sich mit einer Forms-Anwendung zu verbinden. Die von Oracle empfohlene ist über Single-Sign-on. Hier wird der Benutzer mittels eines LDAP Accounts berechtigt auf die Anwendung zuzugreifen. Über die Kombination <config=? > und RAD (Datenbank Beschreibung) wird für die Anwendung der Datenbank connect erstellt und ausgeführt. Was für Möglichkeiten bestehen nun wenn kein Single-Sign-on vorhanden ist.

Da ist zum einen natürlich der normale Aufbau der Verbindung über die Logon Maske. Aber gibt es nicht auch die Möglichkeit den Betriebssystem Benutzer auszulesen? Mit der Komponente „Webutil“ besteht die Möglichkeit, den Betriebssystem User auszulesen. Mit dieser Information könnte man dann einen automatischen Login in Forms realisieren. Über den Standard User verbindet man sich gegen die Datenbank mit der ausgelesen Information des Betriebssystem – User. Anhand dieser Information kann dann ein neuer Connect aus einer Datenbank Tabelle ausgelesen werden um sich mit diesen Informationen dann gegen die Datenbank zu verbinden.

Reports und Security

Wie schaut es nun mit Reports aus. Es werden heute noch sehr viele Reports Server aus Forms heraus genutzt. Die Probleme die sich bei Reports ergeben kann man sehr schön über eine Google Suche ansehen.

Als Beispiel einmal eine Google – Suche „**allinurl: "reports/rwservlet“** hier bekommt man eine Trefferliste von



Wenn ich jetzt auf die Ergebnis – Liste gehe, habe ich durchaus die Möglichkeit wichtige Informationen zu erfahren. Fangen wir einfach mal mit „/**showenv**“ an. Hier ist für mich der wichtigste Bereich einmal der „**Server_Name** : www.XXXXX.ch**“** sowie der Port auf diese Server hört. „**Server_Port : 80**“.

Wenn ich mir jetzt die anderen Möglichkeiten des „Reports - Servlets“ anschaue, kann ich auf „alte“ sowie laufende Reports zugreifen.

Job-ID	Job-Typ	Job-Name	Job-Status	Job-Eigentümer	Ausgabetyp	Ausgabename	Server-Name	In Warteschlange gestellt
2155215	report	C:\oracle_re\j2ee\OC4J_BI_Forms\applications\reports\web\examples//1254/1a10_Import_Entscheide_V9.jsp		RWUser	Cache	Undefiniert	rep_mip-apps_oraclere	04.10.2012 08:25:06
2155223	report	C:\oracle_re\j2ee\OC4J_BI_Forms\applications\reports\web\examples//1254/1a10_Import_Entscheide_V9.jsp		RWUser	Cache	Undefiniert	rep_mip-apps_oraclere	04.10.2012 08:31:19

Ebenso kann ich mir natürlich Informationen von bereits gedruckten Dokumenten anschauen, mit allen Informationen.

In der Oracle Reports Version 9/10 gab es schon immer die Möglichkeit diesen Bereich zu schützen. Dazu wurde in der Servlet Konfiguration der Parameter „**DIAGNOSTIC=No**“ gesetzt. Dies hat zur Folge, da kein Web-Befehl mehr ausgeführt werden konnte und war ein erheblicher Beitrag zur Sicherheit im Reports Umfeld. Leider führte dieses dann auch dazu, das aus Forms heraus das Dokument nicht mehr in der Vorschau mit „**web.showdocument**“ angezeigt werden konnte. Daraus folgte, das man entweder eine eigene Security einbauen musste, dies geschah am besten durch eigenen Konfiguration des http-Servers, oder als alternative alle Dokumente immer nur als „Datei“ erstellt worden sind und diese dann per „Download“ auf den Client übertragen um dort mittels „Host-Befehl“ angezeigt zu werden.

Als Beispiel für eine http-Server Konfiguration könnten man nehmen:

```
<Location /reports/rwservlet/[sS][hH][oO][wW][eE][nN][vV]*>
    Order deny,allow
    Deny from all
    Allow from localhost fmwr2.pitss.com
</Location>
```

Hiermit könnte ich den Zugriff auf „showenv“ verbieten.

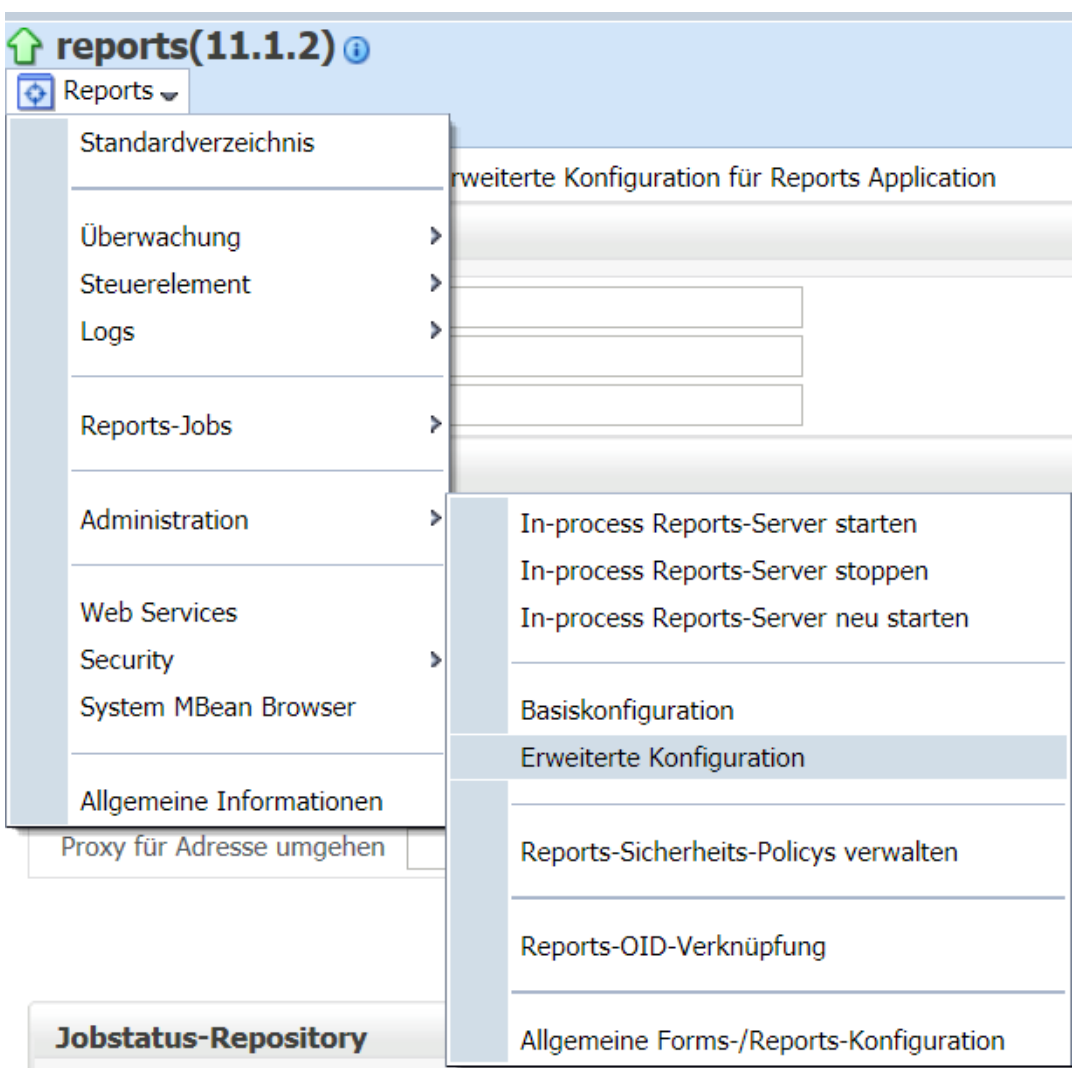
Hier hat sich mit der Einführung der Version 11g einiges an Veränderungen mit sich gebracht. Oracle hat gerade im Bereich der Reports - Security etliche Neuerungen eingebaut. Was ist neu an dieser Stelle.

Es wurde ein auf „Java EE Standard“ basiertes Sicherheitsmodell implementiert. (Oracle Plattform Security Services) dies ermöglicht es einem eine einfache Administration. In Oracle Applikation Server 10g gab es nur die Möglichkeit das „Oracle Internet Directory“ zu nutzen.

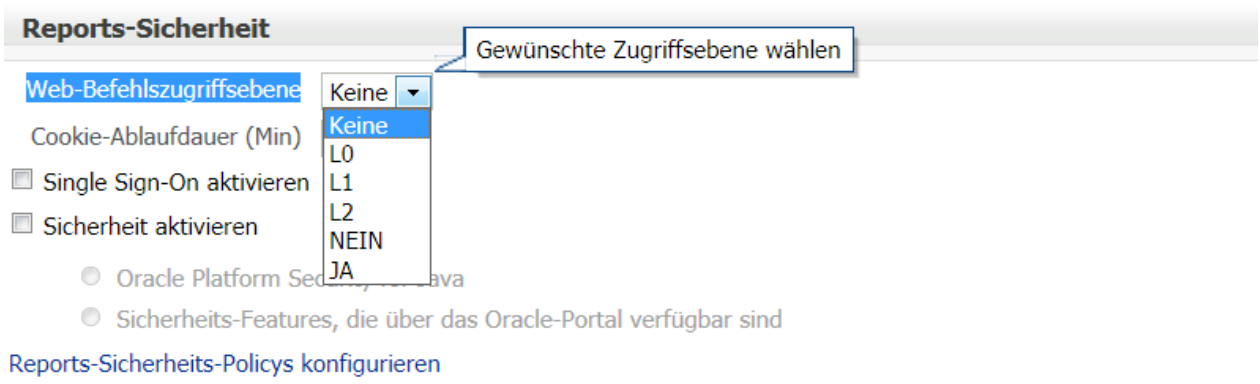
Schreib/ Lese Rechte auf einzelne Verzeichnisse sind zu vergeben. Bisher war das in 10g nicht möglich. Als vorübergehenden Weg hat Oracle den Parameter „REPORT_RESTRICT_DIRECTORIES“ eingeführt.

„Database proxy authentication“ ist mit 11g neu eingeführt. Dieses war in 10g nicht verfügbar.

Über die Administration Konsole (EM) besteht unter anderm die Möglichkeit über die erweiterte Konfiguration hier die neuen Reports – Sicherheits Einstellungen zu tätigen



An dieser Stelle besteht die Möglichkeit, die „Web-Befehlszugriffsebene“ einzustellen. Folgende Auswahl Möglichkeiten werden einem angeboten.



Die Auswahl „NEIN“ oder „JA“ entspricht den Einstellungen im Applikation Server 10g „**DIAGNOSTIC=No“** und hat auch das selbe verhalten.

Folgende Befehle gelten als „End User“

```
GETJOBID, KILLJOBID, SHOWAUTH, SHOWJOBID
```

Für die Administration werden folgende Befehle herangezogen

```
DELAUTH, GETSERVERINFO, KILLENGINE, PARSEQUERY, SHOWENV,
SHOWJOBS, SHOWMAP, SHOWMYJOBS. AUTHID
```

Über die Einstellungen L0 bis L2 und Nein (No) und Ja (Yes) besteht nun die Möglichkeit hier den Zugriff zu steuern.

Bei der Einstellung „L0“ sind gar keine Webkommandos erlaubt. Bei der Einstellung „L1“ nur Kommandos für den End Benutzer. Alles andere wird immer durch eine Abfrage Benutzer/Password unterbunden.

JOBID

Eine Möglichkeit Daten aus dem Reportserver auszulesen besteht durch den Aufruf „**/getjobid<ID>“**. Durch die Einstellung L1 kann man zwar keine Job-Queue sich mehr anschauen, es ist allerdings immer noch möglich einen direkten Job auszulesen. Da die Job’s per default mit einer aufsteigenden Nummer versehen werden, ist es relative einfach hier durch „testen“ eine bestehende Nummer zu finden. Dies kann damit unterbunden werden, in dem man eine „Non-Sequential Job ID“ generiert. Dies erfolgt durch einen start Parameter entweder in der Oracle Weblogic Server Umgebung oder über die „REPORTS_JVM_OPTIONS“ Variable. Der Parameter der gesetzt werden muss ist:

"-Djobid=random"

Nachdem neuem starten der Maschinen werden keine aufsteigenden ID’s mehr generiert.

Ausführen von Reports aus Forms heraus

Wenn man einen Report aus Forms heraus aufrufen möchten, ist die Berechtigung für die Datenbank durch den „login“ in Forms bereits erledigt.

```
SET_REPORT_OBJECT_PROPERTY(Rep_id,Report_server,V_server.Report_server);
SET_REPORT_OBJECT_PROPERTY(Rep_id,Report_destype,CACHE);
SET_REPORT_OBJECT_PROPERTY(Rep_id,Report_execution_mode,Runtime);
V_rep         := RUN_REPORT_OBJECT(Rep_id);
Rep_status    := REPORT_OBJECT_STATUS(V_rep);
WHILE Rep_status IN('RUNNING', 'OPENING_REPORT', 'ENQUEUED')
LOOP
            Rep_status    := REPORT_OBJECT_STATUS(V_rep);
END LOOP;
```

Über diesen Weg ist es sehr schnell möglich einen Report zu erstellen und ihn in der Vorschau angezeigt zu bekommen.

Aber was passiert wenn dem Report eine Parameter Maske vorgeschaltet ist.

Reports mit Parameter Form

Es besteht die Möglichkeit aus Form heraus einen Report zu starten, der eine Parameter Form anbietet. Bei kleineren Parameter Form Masken bietet es sich an diese in dem Forms – Modul abzubilden. Bei Umfangreicheren ist das aber nicht immer Möglich. Daher bietet Forms/ Report die Möglichkeit eine vorgelagerte Parameter Form aufzurufen.

Es wird zur Ergänzung in der Übergabe die Parameter Form erlaubt.

```
pl_id:=Get_Parameter_List('parm');
if not id_null(pl_id) then
              destroy_parameter_list(pl_id);
end if;
pl_id:= Create_Parameter_List('parm');
 Add_Parameter(pl_id, 'PARAMFORM',TEXT_PARAMETER,'YES');
```

Nach dem die Parameter eingegeben worden sind und die Abfrage abgesendet ist, wird das Ergebnis ab Bildschirm da gestellt.

adressen

Firma	Plz	Ort	Strasse	Hausnummer	Id
Logan D. Forbes		Ap #944-2532 Mauris St.	Rycroft		
Logan D. Forbes	26814	Ap #696-1782 Duis St.	Tuscaloosa		387065
Logan D. Forbes	25300	Ap #733-745 Ipsum. Rd.	Talgarth	RvxTh	284643
Logan D. Forbes	14170	1887 Eleifend. Av.	Tulsa	]	810211
Logan D. Forbes	29339	503-8244 Amet, Rd.	Reading	[Eyda_^DRN	408697
Logan D. Forbes	87646	Ap #765-5650 Ut Rd.	Lloydminster	FVsF\d[	647935
Logan D. Forbes		901-1151 Duis St.	Saive	hHU	379933
Logan D. Forbes	85683	3290 Elit, St.	B?cancour		403712
Logan D. Forbes	6973	2976 Risus. Avenue	Daly	pFRR^wEX	744068
Logan D. Forbes	70200	224-3572 Ac Street	Rochester	klerJDmiaC	803011

Leider greift an dieser Stelle die automatische Datenbank Verbindung nicht mehr. Diese muss als Parameter mit übergeben werden. Entweder über die URL (dann ist sie sichtbar) oder als sogenannter „Hidden Parameter“

```
-- hidden Parameter aufbauen
hidden_action := hidden_action
||'&report='||GET_REPORT_OBJECT_PROPERTY(Rep_id,REPORT_FILENAME);
hidden_action :=
hidden_action||'&destype='||GET_REPORT_OBJECT_PROPERTY(Rep_id,REPORT_DESTYPE);
hidden_action := hidden_action||'&desformat='||GET_REPORT_OBJECT_PROPERTY
(Rep_id,REPORT_DESFORMAT);
hidden_action := hidden_action
||'&userid='||get_application_property(username)||'/'||

get_application_property(password)||'@'||get_application_property(connect_stri
ng);
hidden_action := reports_servlet||'?_hidden_server='||V_server.Report_server||
encode(hidden_action);
--message(hidden_action);
SET_REPORT_OBJECT_PROPERTY(Rep_id,REPORT_OTHER,'pfaction='||hidden_action);
-- ende hidden parameter
```

In der URL ist dann nur der Aufruf sichtbar aber keine LOGIN Informationen. Allerdings kann der Anwender über „Anzeige Quellcode“ diese Information einsehen.

```
<form method=post action="http://windemo:7001/reports/rwservlet?">
<input name="hidden_run_parameters" type=hidden
value="server=rep_adminserver_windemo_asinst&report=pitss_daten.rdf&destype=CA
CHE&desformat=PDF&userid=SUMMIT_ADF%2Fadmin1%40orcl">
<font color=red></font>
```

Kontaktadresse:
Jan-Peter Timmermann
PITSS GmbH
Zettachring, 2
D-00000 Stuttgart

Telefon:	+49 172-215 1043
Fax:	+49 4103-180 371
E-Mail	Jtimmermann@pitss.com
Internet:	www.pitss.de