

Export/Import mit Oracle Data Pump

(Stand BPS 2.24.4)

Vorbemerkung

Die alten BPS Skripte `export.js` und `import.js` sind praktisch für einige spezielle Fälle. Vorteil dieser Skripts ist die Unabhängigkeit vom Datenbanktyp, es ist also damit auch möglich ein BPS Schema von Oracle nach PostgreSQL oder umgekehrt umzuziehen. Grosser Nachteil ist jedoch, dass diese Skripte viel langsamer ausgeführt werden als native Datenbank-Tools wie Oracle Data Pump. Deshalb bevorzugen viele Administratoren die direkte Verwendung von Oracle Data Pump, was besonders für 1:1-Backups der ganzen Datenbank absolut in Ordnung ist.

Ab BPS 2.24 stehen nun die nachfolgend beschriebenen Skripte zur Verfügung welche den Export und Import eines Schemas per DATA PUMP massiv vereinfachen.

Traditionelle Methode

Das Exportieren eines BPS-Schemas zu Sicherungszwecken kann mit Oracle Data Pump problemlos durchgeführt werden, indem entweder die vollständige Datenbank oder nur das Schema exportiert wird. Das Wiederherstellen in einer vollständigen Datenbank, in der noch keine der Schema- oder BPS-Rollen oder Benutzer vorhanden ist, ist unkompliziert. Wenn jedoch nur ein einzelnes Schema zu übertragen ist sind einige weitere Aktionen erforderlich: Erstellen der Tablespace und der Rollen `schema_usr` und `schema_gst` vor dem Import und rekreieren der Datenbankbenutzer nach dem Import in der Benutzerverwaltung des BPS Arbeitsplatzes.

Es gibt jedoch Fälle, in denen mehr Aktionen erforderlich sind, z.B. das Kopieren auf einen anderen Schemanamen oder das Reorganisieren, weil etwas beschädigt ist (Indizes, Berechtigungen, Trigger, Funktionen ...). Weitere Informationen finden sie unter [Export/Import Übersicht](#).

Hier kommen die Skripte `bpsoraexp.cmd` und `bpsoraimp.cmd` ins Spiel, da sie die meisten Schritte automatisieren die für einen sauberen Import oder eine Reorganisation erforderlich sind.

BPS Skripte für Oracle Data Pump

bpsoraexp exportiert einerseits die wichtigen Tabellendaten per `expdp` in eine DMP-Datei, und andererseits die Anweisungen zum Erstellen der Sequenzen in eine SQL-Datei. Der perfekte Zeitpunkt für den Export eines Produktionsschemas ist während niemand anderes daran arbeitet. Nichtsdestotrotz verwendet der Export das FLASHBACK Feature, sodass es immer eine konsistente Kopie des Zeitpunkts zu dem der Export gestartet wurde gibt.

Vor dem Import mit **bpsoraimp** muss auf dem Ziel-Oracle-Server ein sauberes BPS-Schema der gleichen Version bereits vorhanden sein. Das kann ein schon vorhandenes Schema sein oder eines das frisch mit dem BPS Assistenten für neue Datenbanken erstellt wird. Die allfällig dort bereits vorhandenen Tabellendaten werden erst gelöscht, und danach die Tabellendaten der Exportdatei

frisch importiert. Die Sequenzen werden ebenfalls zu Beginn gelöscht und dann aus der Exportdatei neu erstellt. Auf diese Weise bleiben alle Objekte, Berechtigungen usw. in einem sauberen Zustand und Sie haben nachher eine einwandfreie BPS Datenbank.

Nach dem Import mit bpsoraimp müssen sie nur noch die BPS-Datenbankbenutzer in der Benutzerverwaltung des BPS Arbeitsplatzes neu generieren.

Die Skripte müssen mit den Anmeldeinformationen des Benutzers SYSTEM oder eines anderen Benutzers mit Oracle DBA-Berechtigungen ausgeführt werden.

Instruktionen

Erzeugen des Data Pump Verzeichnisses

Die Skripte benötigen ein Oracle Verzeichnis mit Namen BPS_DATA_PUMP.

Dateiverzeichnis auf dem Oracle Server erzeugen

In Windows

```
mkdir C:\oracle\bps_data_pump
```

In Linux als Benutzer oracle

```
mkdir -p /home/oracle/bps_data_pump
```



Die nachfolgenden Schritte sind auf einem Windows PC durchzuführen.

Die DATA PUMP UTILITIES sind auch für den Oracle Instant Client verfügbar als Tools Package.

Im Gegensatz zu anderen Oracle Client/Server Komponenten welche diesbezüglich flexibler sind, **müssen** die Versionen von Oracle Server und Client bei den DATA PUMP UTILITIES übereinstimmen.

Verzeichnis in der Datenbank definieren

Mittels sqlplus beim PDB Service anmelden. Auf einem Windows PC z.B. via TNSNAMES:

```
sqlplus system/*****@vmora18
```

Falls sie sqlplus direkt auf dem Server verwenden, vergewissern sie sich dass sie mit dem PDB verbunden sind, nicht mit dem CDB, z.B:

```
sqlplus / as sysdba
```

```
ALTER SESSION SET CONTAINER = ORCLPDB;
```

Verzeichnis bei einem Windows Server definieren:

```
CREATE OR REPLACE DIRECTORY BPS_DATA_PUMP AS 'C:\oracle\bps_data_pump';
```

oder bei einem Linux Server:

```
CREATE OR REPLACE DIRECTORY BPS_DATA_PUMP AS '/home/oracle/bps_data_pump';
```

Prüfen

```
SELECT DIRECTORY_NAME, DIRECTORY_PATH FROM DBA_DIRECTORIES WHERE  
DIRECTORY_NAME='BPS_DATA_PUMP';
```

Export durchführen

Öffnen sie eine BPS Kommandozeile, erzeugen sie ein Arbeitsverzeichnis und wechseln sie dorthin:

```
mkdir C:\oracle\bpsimpexp  
cd /d C:\oracle\bpsimpexp
```

Prüfen ob sqlplus, expdp and bpsoraexp im PATH gefunden werden:

```
sqlplus -v  
expdp help=y  
bpsoraexp
```

Skript ausführen

```
bpsoraexp system/*****@vmora12 lu_buv
```

Beispielausgabe

```
C:\oracle\bpsimpexp>bpsoraexp system/*****@vmora12 lu_buv  
Export table data with expdp to lu_buv-0-2024001-20200723.dmp  
  
Export: Release 12.2.0.1.0 - Production on Thu Jul 23 16:22:53 2020  
  
Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights  
reserved.  
  
Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 -  
64bit Production  
Starting "SYSTEM"."SYS_EXPORT_TABLE_01": system/*****@vmora12  
version=12.2 parfile=lu_buv_exp.par  
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA  
. . exported "LU_BUV"."TL_STOCK" 70.82 KB 3293
```

```

rows
. . exported "LU_BUV"."T_ACTORS"                5.242 KB          1
rows
. . . . .
. . . . .
. . exported "LU_BUV"."T_ZONEPARTNERS"          6.742 KB          322
rows
. . exported "LU_BUV"."T_ZONES"                  5.671 KB           4
rows
Master table "SYSTEM"."SYS_EXPORT_TABLE_01" successfully loaded/unloaded
*****
**
Dump file set for SYSTEM.SYS_EXPORT_TABLE_01 is:
  C:\ORACLE\BPS_DATA_PUMP\lu_buv-0-2024001-20200723.DMP
Job "SYSTEM"."SYS_EXPORT_TABLE_01" successfully completed at Thu Jul 23
16:23:06 2020 elapsed 0 00:00:12

Export sequences to lu_buv-0-2024001-20200723.sql
Script bpsoraexp completed.

C:\oracle\bpsimpexp>

```

- Im aktuellen Arbeitsverzeichnis finden sie die Datei lu_buv-<custom>-<version>-<date>.sql mit den SQL Anweisungen um die Sequenzen zu erzeugen.
- Im DATA PUMP Verzeichnis finden sie die Datei lu_buv-<custom>-<version>-<date>.dmp sowie die Logdatei lu_buv-<custom>-<version>-<date>-exp.log.

Für den späteren Import benötigen sie die Dumpdatei mit den Tabellendaten, sowie die SQL-Datei zur Erzeugung der Sequenzen.

Import Durchführen

In unserem Beispiel haben wir die Dateien lu_buv-0-2024001-20200723.dmp und lu_buv-0-2024001-20200723.sql mit dem Export des Schemas LU_BUV von einem anderen Datenbankserver. Das Schema soll auf den Zielservers vmora12 und hier ebenfalls in ein Schema names LU_BUV importiert werden.

- Kopieren sie die Datei lu_buv-0-2024001-20200723.dmp in das BPS Data Pump Verzeichnis auf dem Zielservers.
- Kreieren sie ein Arbeitsverzeichnis von wo aus bpsoraimp ausgeführt werden soll.
- Kopieren sie die Datei lu_buv-0-2024001-20200723.sql in das Arbeitsverzeichnis.
- Prüfen/vorbereiten des Schema LU_BUV auf dem Server vmora12:
 - Falls das Schema noch nicht existiert, erzeugen sie es mit Hilfe des [BPS Assistenten für neue Datenbanken](#).
 - Falls das Schema schon existiert aber eine alte Version ist, aktualisieren sie es mit dem BPS Assistenten für die Datenbankaktualisierung.
 - Falls das Schema schon existiert aber möglicherweise beschädigt ist, löschen sie es mit dem BPS Assistenten für Datenbank Löschung, und kreieren sie es neu mit Hilfe des BPS Assistenten für neue Datenbanken.

- In allen Fällen stellen sie sicher dass das Schema auf dem Zielsystem die gleiche Version hat wie jenes des Quellsystems, sonst wird der Import möglicherweise fehlschlagen. (Zu sehen in den BPS Einstellungen unter Central System Settings/Install/DbLevel)
- Öffnen sie eine BPS Kommandozeile und führen sie nachfolgendes aus

In das Arbeitsverzeichnis wechseln

```
cd /d c:\oracle\bpsimpexp
```

Kontrollieren dass sqlplus, impdp und bpsoraimp gefunden werden

```
sqlplus -v
impdp help=y
bpsoraimp
```

Skript starten

```
bpsoraimp system/*****@vmora12 lu_buv-0-2024001-20200723
```

Beispielausgabe

```
C:\oracle\bpsimpexp>bpsoraimp system/*****@vmora12
lu_buv-0-2024001-20200723

Disable constraints and triggers
Drop sequences and truncate tables
Import tables with impdp

Import: Release 12.2.0.1.0 - Production on Thu Jul 23 16:18:05 2020

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights
reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 -
64bit Production
Master table "SYSTEM"."SYS_IMPORT_TABLE_02" successfully loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_TABLE_02": system/*****@vmora12
parfile=lu_buv_imp.par
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
. . imported "LU_BUV"."T_REPORTZONES"          4.906 KB          19
rows
. . imported "LU_BUV"."T_NEWLOGISTICUNITS"      5.484 KB           1
rows
. . . . .
...
. . imported "LU_BUV"."T_ORDERPICKS"           1.087 MB        62246
rows
. . imported "LU_BUV"."T_ORDERITEMS"           661.4 KB        62215
rows
Job "SYSTEM"."SYS_IMPORT_TABLE_02" successfully completed at Thu Jul 23
```

16:19:48 2020 elapsed 0 00:01:42

```
Import sequences from lu_buv-0-2024001-20200723.sql
Enable constraints and triggers
Disable auditing
Restore install settings
Rebuild shadows
Enable auditing
Grant permissions on sequences
Script bpsoraimp completed.

C:\oracle\bpsimpexp>
```

- Im DATA PUMP Verzeichnis auf dem Oracle Server finden sie die Logdatei
lu_buv-0-2024001-20200723_imp.log mit der Protokollierung des Datenimports per impdp.

Kundenspezifische Tabellen und Sequenzen

Wenn sie eigene Datenbank-Objekte innerhalb der BPS Datenbank angelegt haben, so *sollte* die Benennung den BPS Konventionen folgen und jeweils ein c (für „Custom“) vorangestellt haben:

Objektyp	BPS Eigene Objekte	Kundenspezifische Objekte
TABLE	t_* ta_* tl_* tn_* ...	ct_* ...
SEQUENCE	s_* sl_* ...	cs_* ...
VIEW	v_* va_* ...	cv_* ...
INDEX	i_* ...	ci_* ...
FUNCTION	f_* ...	cf_* ...
PROCEDURE	p_* ...	cp_* ...
TRIGGER	tr_* tra_* trd#_* trl_* trp#_* ...	ctr_* ...
TYPE	tp_* ...	ctp_* ...
PACKAGE	ptr_* ptrp_* ...	cptr_* ...

Wenn sie bei der Benennung ihrer kundenspezifischen Objekte generell ein c voranstellen, so ist einerseits garantiert dass sie jetzt und in Zukunft nie in Konflikt mit den BPS eigenen Objekten kommen.

Ausserdem profitieren sie dann davon dass die hier beschriebenen Export- und Import-Programme auch die Inhalte ihrer mit ct_* benannten Tabellen, sowie den aktuellen Stand der mit cs_* benannten Sequenzen sichern und zurückspielen können.



Alle anderen Objekte ausser Tabellen und Sequenzen sind für den Export/Import dieser Skripts irrelevant. Natürlich benötigen sie die auch für ihre Applikationen, aber jene Objekte werden nicht durch diese Skripts erzeugt sondern durch ihre eigenen Setup-Skripte.

Der Exportskript schaut standardmässig ob es irgendwelche Tabellen ct_* oder Sequenzen cs_* gibt,

und exportiert diese automatisch mit.

Dateien welche nicht nur BPS eigene Daten sondern auch kundenspezifische enthalten, erkennen sie am Namen. Sie haben nach dem Schema eine 1 statt eine 0:

lu_buv-**0**-2024004-20210428.sql -> Enthält keine kundenspezifischen Daten

lu_buv-**1**-2024004-20210428.sql -> Enthält kundenspezifische Daten

Möglicherweise haben sie zwar solche kundenspezifischen Objekte, möchten diese aber doch nicht exportieren (wir sind ihnen z.B. dankbar wenn sie uns ggf. einen Export ohne diese senden). Sie können das beim Export erreichen indem sie am Schluss des Exportbefehls nocustom anhängen, z.B:

```
C:\dev\bpsoradmp>bpsoraexp system/*****@vmora12 lu_buv nocustom
```

Der Importskript erkennt am Dateinamen ob er kundenspezifische Daten mitimportieren muss, dort müssen sie also nichts spezielles angeben. Beim Import müssen jedoch die kundenspezifischen Objekte im Zielschema bereits vorhanden und auf dem gleichen Versionsstand sein wie beim Quellschema. Das machen Sie analog den BPS Objekten. Wenn sie z.B. ein neues Schema mit Hilfe des [BPS Assistenten für neue Datenbanken](#) initialisieren, führen sie danach auch die Skripts aus welche ihre kundenspezifischen Objekte generieren. Danach klappt dann auch der Import inklusive den Daten in ihren kundenspezifischen Tabellen und den Zählerständen ihrer kundenspezifischen Sequenzen.

In ein Schema mit anderem Namen importieren

Die Schritte sind grundsätzlich die gleichen wie oben wo wir in das Schema LU_BUV importiert haben.

Diesesmal wollen wir von lu_buv-0-2024001-20200723.dmp und lu_buv-0-2024001-20200723.sql in das Schema LU_FGB importieren.

Statt das Schema LU_BUV auf dem Zielserver vorzubereiten, tun wir das diesmal einfach für das Schema LU_FGB .

Ein Schema LU_BUV muss in diesem Fall nicht auf dem Zielserver existieren, allerdings schadet es auch nicht.



Diese Methode erlaubt es also auch in ein anderes Schema auf dem Quellserver zurück zu importieren, ohne das Original-Schema dabei zu verändern.

Das können sie also gut verwenden um ihr Produktions-Schema auf ein Test- oder Qualitäts-Schema zu kopieren.

Skript starten, aber dieses mal hängen wir den Namen des Ziel-Schemas noch an

```
bpsoraimp system/*****@vmora12 lu_buv-0-2024001-20200723 lu_fgb
```

Einfach, oder?

From:

<https://bps.ibk-software.com/> - **BPS WIKI**

Permanent link:

<https://bps.ibk-software.com/dok:orapump>

Last update: **22.06.2021 09:20**

