

Migration der Daten eines CRS-Mandanten von Kunde zu Kunde

Dr. Kurt Franke
targens GmbH

Kurt.Franke@targens.de, Kurt-Franke@web.de

Agenda

- targens Kurzvorstellung
- Situation
- Grobe Vorgehensweise
- Datapump-Export des CRS-System von Bank1
- Vorbereitung CRS-3.1-System bei Bank2
- Import Dumpfile in CRS-3.1-System und Nacharbeiten
- Anhebung intermediäres CRS-3.1 auf Version 4.3
- Vorbereitung Datenübernahme ins Ziel-CRS
- Datenübernahme ins Ziel-CRS
- Abschließende Aktionen
- Zusammenfassung

Zahlen und Fakten auf einen Blick

Firmensitz:	Stuttgart
Geschäftsstellen:	Frankfurt am Main, München
Branchenerfahrung:	seit 1988
Umsatz:	€ 30 Mio.
Anzahl Mitarbeiter:	210
Rechtsform:	Gesellschaft mit beschränkter Haftung
Geschäftsführer:	Thomas Wild
Gesellschafter:	100 % Landesbank Baden-Württemberg (LBBW)
Aufsichtsratsvorsitz:	Dr. Martin Setzer



SMARAGD Compliance Suite



- Mehr als 1.600 Institute in über 50 Ländern vertrauen auf SMARAGD Produkte
- Unter den 50 Top Banken weltweit nutzen 9 Banken SMARAGD Lösungen
- Die ersten 7 der Top 10 Banken in Deutschland setzen auf die SMARAGD Compliance Suite
- Die deutsche und die österreichische Zentral-Bank nutzen SMARAGD Lösungen
- Seit 1999 stehen wir mit unseren Lösungen der Finanzwirtschaft und seit 2011 der Industrie zur Verfügung

Referenzen SMARAGD (Auszug)



Migration CRS-Mandant von Kunde zu Kunde

Situation

- Bank 1 hat CRS-System
 - als separater Mandant bei externem Betreiber
 - CRS Version 3.1
- Bank 2 betreibt ein CRS-System haus-intern
 - CRS Version 4.3
- Bank 2 übernimmt Bank 1
 - Anforderung:
 - Integration des historischen CRS-System von Bank 1
 - in das vorhandene CRS-System von Bank 2

Grobe Vorgehensweise

- Datapump-Export des CRS-System von Bank1
 - durch externen Betreiber
 - mit bereitgestelltem Script von targens
 - Einschränkung auf Mandant von Bank 1 durch Script
 - andere Mandanten dürfen keinesfalls im Export sein !
 - nachdem der letzte CRS-Jahresabschluß von Bank 1 erfolgt ist
- Bereitstellung des Exports an Bank 2
- Bank 2 benötigt übergangsweise ein CRS-3.1 System
 - damit der bereitgestellte Datapump-Export importiert werden kann
- Das Übergangs-CRS-3.1 wird auf CRS-4.3 angehoben
 - nach erfolgreichem Export und notwendigen Anpassungen
 - mit Standard-Update-Verfahren von CRS

Grobe Vorgehensweise II

- Export des Risikomodells jeder Version
 - mit CRS-Utility (aus Übergangs-CRS)
- Import des Risikomodells jeder Version
 - mit CRS-Utility (ins haus-interne CRS)
 - Die ID´s des Risikomodells werden automatisch verschoben
 - in einen freien Bereich der jeweiligen Sequence
 - mit Sequence-Anpassung hinter das Ende des ID-Bereiches
- Übernahme der Massendaten (Kunde etc.)
 - partitionsweise über Database Link aus Übergangs-CRS
 - die GAE-ID wird dabei umgesetzt auf die neue ID
 - über das beim Risikomodell-Import erstellte ID-Mapping

Datapump-Export des CRS-System von Bank1

- Script zur Generierung Export-Parameter-File
 - build_expdp_paramlist.sql (SQL*Plus Script)
 - Bereitstellung durch targens passend für CRS-3.1
 - als Aufruf-Parameter wird Mandanten-Name erwartet
 - ➔ ergibt Mandant-ID
 - Liste der beteiligten GAE´s für den Mandanten bestimmt
 - Liste der Risiko-Modell-Tabellen wird unter TABLES ausgegeben
 - hier relevante Massendaten-Tabellen listpartitioniert nach GAE
 - Ausgabe der betroffenen Tabellen-Partitionen unter TABLES
 - in der Form TABLE_NAME:PARTITION_NAME
 - für Risiko-Modell-Tabellen werden WHERE-Klauseln definiert
 - basierend auf dem CRS-Datenmodell
 - zur Rückführung für jede Tabelle auf die Mandant-ID
 - mit Mandant-ID als Selections-Kriterium
 - Parameterfile Expdp_CRS_Mandant.par

Datapump-Export des CRS-System von Bank1

- Auszug aus generiertem Export-Parameter-File

```

TABLES= (\
  MANDANT\
, VERSION\
, LAND\
, GAE\
, ... \
, KUNDE_KONTO:KKGAE_29610799\
, KUNDE_KONTO:KKGAE_60965319\
, ... \
, KUNDE:KGAE_29610799\
, KUNDE:KGAE_60965319\
)

```

Datapump-Export des CRS-System von Bank1

- Auszug aus generiertem Export-Parameter-File

```
QUERY='MANDANT:"WHERE id IN (29386210)''
```

```
QUERY='VERSION:"WHERE id IN (SELECT id FROM version v WHERE  
v.mandant_id IN (29386210))''
```

```
QUERY='LAND:"WHERE version_id IN (SELECT id FROM version v  
WHERE v.mandant_id IN (29386210))''
```

```
QUERY='GAE:"WHERE version_id IN (SELECT id FROM version v  
WHERE v.mandant_id IN (29386210))",
```

...

- Aufruf-Script für Export : expdp-call.sh
 - expdp ?/?@? DUMPFILERE=? LOGFILE=? DIRECTORY=?
PARFILE=Expdp_CRS_Mandant.par

Vorbereitung CRS-3.1-System bei Bank2

- für Import des Datapump Exports von Bank1
 - gleiche CRS-Version für übereinstimmende Tabellen-Definitionen
 - Bank2 erhält eine CRS-3.1- Auslieferung von targens
 - Installation mit CRS-Standard-Mechanismus
 - `dbadminutil.sh -product crs -user smaragd_dba -schema crs_31_imp -action install`
- Löschen von Teilen der Standard-System-Initialisierung
 - mit Utility zum Leeren des CRS-Systems
 - zuvor Sichern Tabellen-Inhalte von RECHTE und ROLLE
 - zur Steuerung CRS GUI Zugriffe
 - `pkg_admin_util.deleteBeforeImport()`
 - `pkg_admin_util.enOrDisableAll(FALSE)` (Foreign Keys)
 - `pkg_admin_util.truncateTablesBeforeImport()`
 - Wiederherstellen Tabellen-Inhalte von RECHTE und ROLLE

Import Dumpfile in CRS-3.1-System bei Bank2

- impdp : Anlegen von Partitionen für importierte Daten
 - wird nur zusammen mit Anlegen der Tabelle durchgeführt
 - dann werden alle Partitionen angelegt
 - auch die, für die keine Daten im Dumpfile sind
- ➔ Es ist notwendig, partitionierte Tabellen zu dropen
 - wenn dafür Daten importiert werden
 - zur Vermeidung Zusammenbau der kompletten Statements
- alle vorbereitenden Aktionen sind in Script enthalten
 - bereitgestellt von targens
- Aufruf Vorbereitungsscript für Import
- Aufruf Import : impdp-call.sh
 - impdp system/?@? DUMPFILERE=? REMAP_SCHEMA=??:?
LOGFILE=? TABLE_EXISTS_ACTION=APPEND DIRECTORY=?

Nacharbeiten nach Dumpfile Import (via Script)

- Einschalten Foreign Keys
 - `pkg_admin_util.enOrDisableAll(TRUE)`
- Sequences an importierte Daten anpassen
 - `pkg_admin_util.adjustSequence()`
- Statistiken der Risikomodell-Tabellen erzeugen
 - `pkg_admin_util.gen_crs_stats()`
- Statistiken für partitionierte Massendaten-Tabellen
 - explizite Aufrufe von `dbms_stats.gather_table_stats()`
- Alle GAE's schließen – keine Änderung mehr zulässig
- GUI-User mit Benutzer-Admin-Rechten anlegen
- Dropen aller GAE-Partitionen ohne GAE in den Daten
 - ca. 95 % der Partitionen im vorliegenden Fall

Anhebung intermediäres CRS-3.1 auf Version 4.3

- importierte CRS-Daten liegen in der Version 3.1 vor
- Ziel-CRS-System hat Version 4.3
- ein direktes Kopieren der Daten ist nicht möglich
 - wegen teilweise geänderten Tabellen-Definitionen
 - und auch neuen oder nicht mehr vorhandenen Tabellen in 4.3
- → Daten müssen migriert werden
- CRS-Standard-Update-Verfahren
 - beinhaltet alle notwendigen Migrationen
 - für jeden einzelnen Update-Schritt
 - hier muss die CRS-4.3 File-System Installation eingesetzt werden
 - beinhaltet auch eine neuere DBAU-Version
 - `dbadminutil.sh -product crs -user smaragd_dba -schema crs_31_imp -action update`
- Daten jetzt passend für CRS-System von Bank2

Vorbereitung Datenübernahme ins Ziel-CRS

- Ausführung von Generator-SQL-Scripten
 - im importierten CRS
 - zur Erstellung eines SQL-Scripts zum Anlegen eines Mandanten
 - im CRS-System von Bank2
 - mit rudimentären Versionen für die vorhandenen Jahre
 - zur Aufnahme der importierten Daten von Bank1
 - notwendig wegen verschiedenen Risikomodellen zu Bank2-CRS
 - zur Erstellung eines Shellscripts mit DBAU-Aufrufen
 - zum Export der Risikomodelle der vorhandenen Versionen
 - in Reihenfolge der Jahre der Versionen
 - Unterjährige Versionen (UV, eine Art Snapshot) gibt es keine
 - zur Erstellung eines Shellscripts mit DBAU-Aufrufen
 - zum Import der Risikomodelle der vorhandenen Versionen
 - ins CRS-System von Bank2
 - in Reihenfolge der Jahre der Versionen

Datenübernahme ins Ziel-CRS I

- Ausführung generiertes Shellsript mit DBAU-Aufrufen
 - im importierten CRS
 - zum Export der Risikomodelle der vorhandenen Versionen
- Ausführung generiertes Mandant-Create-Script
 - im CRS-System von Bank2
- Aufruf Script zum Setzen eines DEBUG-Parameters
 - im CRS-System von Bank2
 - zur Unterdrückung des Dropping der GAE-ID-Mappingtabellen
 - die beim Risikomodel-Import erstellt werden
 - GAE-ID-Mappingtabellen werden noch benötigt
 - für nachfolgendes Kopieren der partitionierten Massendaten

Datenübernahme ins Ziel-CRS II

- bei Bedarf Transport der Risikomodell-XML-Dateien
 - falls das CRS für den Import in einer Datenbank ist
- Ausführung generiertes Shellscript mit DBAU-Aufrufen
 - im CRS-System von Bank2
 - nach Abschluß des Exports aus Import-CRS
 - zum Import der Risikomodelle der vorhandenen Versionen
 - ID-Sets werden dabei in leere Bereiche verschoben
 - die betroffenen Sequences werden angepasst
 - zur Vermeidung Zuteilung Werte aus neu benutztem Bereichs
 - versionsübergreifende ID-Reihenfolge bleibt erhalten
 - durch jahresweise Reihenfolge der DBAU-Aufrufe
 - für die enthaltenen Versionen

Datenübernahme ins Ziel-CRS III

- Erstellung eines Datenbank Links
 - aus CRS-System von Bank2 ins Import-CRS
- Aufruf Kopier-Script für die partitionierten Massendaten
 - im CRS-System von Bank2
 - LOOP über alle Versionen des neuen Mandanten
 - inner-LOOP über alle Massendaten-GAE's (Typ G, nicht L oder Z)
 - inner-2-LOOP über alle partitionierten Massendaten-Tabellen
 - erzeugt zuerst Partition für GAE-ID im Ziel-CRS
 - Erzeugung Exchange-Table
 - Kopiert die zugehörigen Daten aus Import-CRS in Exchange-Table
 - mit den aus der Mapping-Tabelle erhaltenen GAE-ID's
 - die im Import-CRS in Verwendung waren
 - Erzeugung Indexe auf Exchange-Table
 - Integration in Zieltabelle durch Partition-Exchange mit Indexes

Abschließende Aktionen

- Aufruf Script zur Erteilung initiale Rechte neuer Mandant
 - Erteilung Rechte für Benutzeradministratoren von Bank2-CRS
 - für GUI Login
 - zur Sichtbarmachung der GAE´s des neuen Mandanten
 - zum Login in diese GAE´s des neuen Mandanten
- Aufruf Script zur Bereinigung von Arbeitstabellen
 - aus den Risikomodell-Importen
 - die GAE-ID-Mapping-Tabellen bleiben aber erhalten
 - aus Gründen der Nachvollziehbarkeit
- Erteilung der notwendigen Rechte für Anwender
 - mit CRS GUI Bordmitteln
 - durch einen oder mehrere Benutzeradministratoren

Zusammenfassung

- eine derartige Migration ist meist einmaliges Ereignis
 - zumindest was die beteiligten CRS-Versionen betrifft
- die Entwicklung eine Reihe von Scripten ist notwendig
 - viel Einmal-Aufwand
- Einsatz möglichst vieler Standard-Verfahren anstreben
 - Datapump zur Mandanten-Extraktion einsetzen
 - Parameterfile-Generierungsscript relativ aufwändig
 - Anpassung an CRS-Version notwendig
 - immer Standard-Update zur Version-Angleichung verwenden
 - solange ein Kunde sich nicht explicit dagegen ausspricht
 - Verfahren wie Risikomodell-Export / -Import einsetzen
 - gibt es erst ab Version CRS-4.3
 - ➔ immer: Ziel-CRS-System-Version ≥ 4.3
 - ggf. vorher update durchführen
 - zur Vermeidung Entwicklung eines speziellen Verfahrens

Fragen & Antworten

Vielen Dank für Ihre Aufmerksamkeit

Kontakt: **Dr. Kurt Franke**

targens GmbH, Calwer Str. 33, D-70173 Stuttgart

Email: Kurt.Franke@targens.de , Kurt-Franke@web.de

Phone: +49-(0)711-222992676 , Mobil: +49-(0)171-7963089