

# **Siebel CRM 7.7 SEA zu 8.1 SIA Upgrade**

**Alexander Doubek  
Riverland Reply  
München**

## **Schlüsselworte**

SIEBEL, Upgrade, SEA, SIA

## **Einleitung**

Im folgendem werden wichtige Aspekte eines Siebel Upgrade anhand eines der größten deutschen Siebel CRM Systems beleuchtet. Es werden folgende Punkte näher Betrachtet:

- Windows zu Linux-Server Migration,
- Siebel Open UI,
- Konvertierung von LONG in CLOB Spalten,
- Two-Step Migration von Siebel CRM 7.7 SEA zu 8.1 SIA
- kurzes Downtime Fenster
- Vergleich des Adresse Modelles von 8.1 SIA und 7.7 SEA
- Tabellenpartitionierung,
- eine alternative Dokument-Server Lösung.

## **Das System des Kunden**

Das System besteht aus folgenden teilen:

- Oracle Weblogic – Als Backend System für die Webseite und enthält zusätzlich den Hauptteil der Processlogic
- Webmethods – Für System für alle Batch Import / Export Prozesse.
- Oracle Datenbank – mit 2 Instanzen für Siebel und das Kern Loyalty Management System
- Oracle Siebel – welches alle personen- und partnerbezogenen Informationen speichert und für alle Kommunikationsprozessen und die Call Center Oberfläche zuständig ist.

## **Der Siebel-Teil des Systems**

Das Oracle Siebel System ist das führende System, für alle Personen - und Partnerbezogenen Informationen und basiert auf Siebel 7.7 SEA Callcenter es werden ~ 30 Mio. Contact, ~ 200 Mio. Service Request und ~ 200 Mio. Action verwaltet. Dies führt zu ~ 1,7 TB Daten. Das System ist für die Abarbeitung folgender Prozess zuständig:

- MDM
- Claim Management
- Fulfillment

Das System hat folgende Geschichte:

- 2000 - Erstes Setup mit 7.0 SEA als ein Teil des Kern- Loyalty Systems und Ersatz für einen c++ Call Center Client
- 2005 - Upgrade auf 7.7 SEA
- 2013 - Hardware-Upgrade und Virtualisierung des Servers
- 2014 - Upgrade auf 8.1 SIA

### **Warum upgraden**

Gründe für den Upgrade? Das System unterstützen nur IE6, Windows Server 2003, im Development konnten nur noch XP Rechner für Siebel Tools verwendet werden obwohl der Kunde schon Windows 7 als Standard verwendet und Gesamtsystem war Out of Support

All dies führte zu einem technischen Upgrade.

Aber es gibt trotzdem Mehrwert für das Business wie z.B. das System wieder von Oracle supported wird. Neue Version mehr Möglichkeiten (neue Features) bittet. Das Gesamtsystem sicher wird und der TCO sinkt.

### **Die Ziele für den Upgrade:**

#### **Unsere Kernziele**

Ziel für den Upgrade lassen sich somit wie folgt zusammenfassen: wieder Support, weg mit dem IE6, Supportes Server OS, Weg mit XP.

Aus Businesssicht kommen folgende Ziele hinzu:

- Kurze Downtime (< 24h)
- Schneller go-live
- Keine Unterbrechung des Developments somit sollten weiterhin monatliche Release geliefert werden.

#### **nice to have Ziele**

Da der Upgrade aus technischer Sicht eine einmalige Gelegenheit ist gewisse Themen anzugehen ergaben sich einige nice to have Ziele

- Konsolidierung der Table spaces
- Einführung von Open UI
- Aktivieren von TAF für Datenbank
- Das auf Case In sensitiv umzustellen
- Migration auf Linux als Server Betriebssystem.

- Konvertieren der LONG Spalten zu CLOB
- Partitionierung der großer Tabellen

### Auswirkungen der Ziel

All diese ist uns klar werden dass wir gewisse Themen direkt angehen müssen somit bedeutet der Umstieg auf Linux Server das eine neue Correspondence Lösung benötigt wird. Auch die Datenbankgröße und das vorgegebene Downtime Zeitfenster passt nicht zu Standard Upgrade-Prozess

werden wir einige Themen aus den Bereichen

- Infrastruktur
- Organisation
- Entwicklung
- Datenbank Upgrade

Näher eingegangen um Denkanstöße für eigene Projekte zu gehen

### Infrastruktur

#### CORRESPONDENCE

Der Linux Server OS benötigen einen Ersatz für den DocServer welcher Nach der sehr alten Technologie MS Word auf dem Windows Server im Hintergrund öffnet und ein Dokument erstellt.

Als Teil des Upgrade Projektes wurden folgende Lösung analysiert:

Einen Windows. Server behalten	Wechsel zu BI Publisher	C4	Eigenentwicklung
(+) Kein Mehraufwand für das upgrade Project  (-) Erhöht die Anzahl der Server für DEV, TST, INT, PRD	(+) Oracle Lösung als Ersatz für den Doc Server  (+) Fügt sich perfekt in die vorhanden Weblogic Architektur ein  (-) kleines Projekt für sich.  (-) Seht große Lösung um ein paar Felder zu ersetzen  (- Keine Lizenz Migration möglich	(+) Mächtige 3rd Party Lösung  (-) kleines Projekt für sich.  (-) Seht große Lösung um ein paar Felder zu ersetzen  (-) Weitere Lizenzkosten	(+) Einfach zu entwickeln  (+) Nur Entwicklungskosten  (+) Keine GUI Änderungen notwendig

Basierend auf diesen Ergebnissen haben wir uns für die Eigenentwicklung entschieden. Diese Lösung bestehend aus einem JAVA Business Service basierend auf der Klasse CSSJavaBusinessService und etwas eScript in der Ursprünglich Correspondence BC. All dies konnte vollständig im Hintergrund implementiert werden so dass für den Käufer versenden keine Änderungen an der Oberfläche zu sehen sind

Bei der Implementierung von auf JAVA basierend Business Services sind einige Punkte zu beachten. Ganz wichtig ist hier das Session-Management. Bei Siebel muss immer davon ausgegangen werden dass ein Prozesse mehrmals gleichzeitig ausgeführt wird und auch auf verschiedene Server laufen kann. Darf ich wenigstens Systeme nur aus einem Server bestehen. Weiterhin ist strikt darauf zu achten das keine Interaktion mit den Siebel-Daten (Datenbank) selbst geschehen. Um zum Beispiel keine 16 Session Management in Richtung Datenbank etablieren zu müssen. Der Isarprozess soll als klassischer Eingabe Verarbeitung Ausgabe Prozess designend werden. Wobei die Verarbeitung vollständig in Java Teil passiert.

### **Siebel File System Migration**

Durch den Umstieg auf das Linux Betriebssystem ist eine SMB zu NFS Konvertierung des Siebel File System notwendig da Windows nur SMB Nativ unterstützt und Linux nur NFS. Datenkonvertierung von mehreren 100.000 Dateien sehr lange dauern haben wir diesen Task einer temporären Änderung am Livesystem aus dem Migration Wochenende genommen!

Waren folgende Schritte notwendig:

- ① Klonen der virtuell Festplatte der Windows VM
- ② Verschieben dieses Klons zur Linux vm und dort mounten.
- ③ Lokale Kopie der Dateien von NTFS zur NFS Festplatte
- ④ Löschen der SMB Klone Disk
- ⑤ Aktivieren eines SMB Shares für die NFS Festplatte
- ⑥ Finaler abgleich mit rsync über das Netzwerk und Anpassung des Win Prod. System zur Verwendung des "neuen" Siebel File System
- ⑦ Das neue Linux basierende Siebel System kann das migrierte FS parallel verwenden

### **ARCHITEKTUR**

#### **High Available Setup**

Das High Availability Setup des Systems hat sich über die Jahre weiter entwickelt

Das Erst Setup mit Version 7.7 bestand aus einem klassischem Windows Cluster und weiteren Load balanced Servern. Über die Jahre hat sich gezeigt das diese Setup zu komplex zu betreiben ist. In der Realität sinkt die Verfügbarkeit erstattet zu steigen.

Somit wurde während der Data Center Migration wurde das System nicht nur vollständig virtualisiert. Sondern es wurde auch der Siebel Server Cluster (Gateway) wurde durch einen einfachen Server mit automatischen restart durch VMware ersetzt.

Im Zuge des Upgrade auf Siebel 8.1 Und der Umstellung auf Linux Wurde entschieden die SWSE direkt auf den Servern welche den Objekt Manager Beherbergen zu installieren. Dies reduziert die ehemalige Gateway Webserver Maschine auch einen sehr kleinen simplen Server welcher nur noch das Gateway bereit stellt. Diese reduziert die Last und Komplexität und somit das Risiko von der VM des Crashes. Weiterhin vereinfacht es die Architektur Da das System jetzt ohne Probleme Horizontal durch Kopieren der Objekt Manger/Webserver erweitert werden kann

### New UI

Obwohl die Einführung der Open UI ein der Ziele des Upgrade war haben wir uns am Ende für die Weiterverwendung der HI Version für die core Benutzer (Callcenter)entschieden. Der Hauptgrund war das alle automatischen Test für die HI Oberfläche erstellt sind Und wir es als keine gute Idee empfunden haben zeitgleich das System und die Tests zu verändern. Somit kommt jetzt die Klassische HI Oberfläche im Callcenter zum Einsatz. Open UI wird aber aktive von Administratoren und BackOffice Benutzer verwendet. Das bisherige Feedback ist sehr gut. Aktuell bauen wir jetzt die Testcases um danach vollständig auf Open UI zu wechseln.

Open UI konfrontiert uns auch mit Den Browser Version Problem. Wie ein folgender Tabelle zu sehen ist gibt es keinen Browser der HI und Open UI zeitgleich unterstützt

Browser	HI	Open UI
IE 7	OK	Funktioniert aber sehr langsam
IE 8	OK aber Anpassungen notwendig	Funktioniert aber sehr langsam
IE 9	Nein	Funktioniert aber sehr langsam
>= IE 10	Nein	OK <input type="checkbox"/>
Firefox	Nein	OK
Chrome	Nein	OL (politisch schwierig)

### SEA zu SIA

Warum ist ein SEA zu SIA Migration Notwendig? Die einfache Antwort ist das SEA nicht mehr verfügbar ist. Die Oracle Dokumentation zeigt uns das ein 2 Schritt Upgrade notwendig ist.

Aber welche Variante wählen?

- SEA 7.7 → SIA 7.7 → SIA 8.1 oder

- SEA 7.7 → SEA 8.1 → SIA 8.1

Die Also ist so schnell wie möglich auf die neueste Codebasis zu wechseln. Da hier die Upgrade-Routinen und Scripts 10 Jahre jünger und in einem besserem Zustand sind.

somit SEA 7.7 > SEA 8.1 > SIA 8.1

## **2 Stepp Migration**

Der Upgrade in Zwei Schritten ist kein Problem Für unsere kleinen development und Test Systeme: benötigt aber viel zu viel Zeit für die Produktion Umgebung.

Somit mussten wir eine angepasste Migration für Produktions-System entwickeln.

- ① Kopie der Produktion zur Integration (Code freeze auf Produktion)
- ② Migration des Integration Environment mit Standard Tools
- ③ Testen (seed data Probleme lösen)
- ④ Exportieren aller Repository und Konfigtabellen mit DB Tools aus der Integrationsumgebung
- ⑤ Import des Repository und Konfigdumps während der Produktion Downtime
- ⑥ Ausführen der gemergten zwei Schritt Migration Skripte in einem Schritt (ohne Repository und Seed data Import)

## **Adresse Model SEA 7.7 Adresse Model**

Siebel 7.7.x verwendet eine 1:M-Beziehung um Adressen zu speichern. Und verwendet zwei unterschiedliche Adresstabellen eine für jede/n Account, Partner, Organisation und eine Für jeden Member, Kontakt, User.

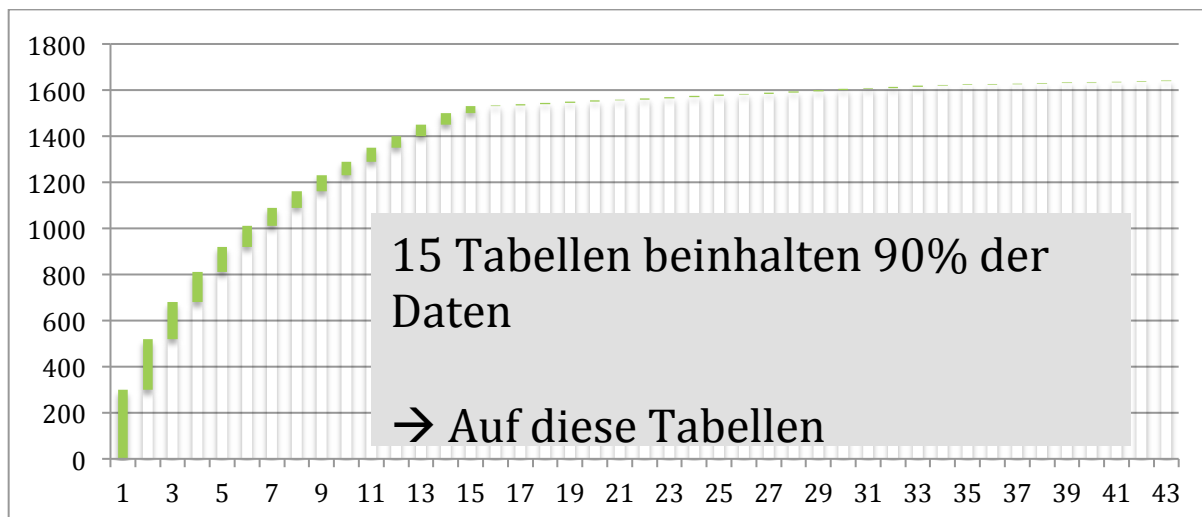
Das SIA 8.1 Adresse Model verwendet eine M:M Beziehung um Adressen zu speichern und benötigt hierfür auch nur noch eine core Adresstabelle welche die Adressen für jeden Account, Partner, Organization, Member, Contact, User, etc. enthält

Einer Analyse hat gezeigt das für die Umstellung des Adressdatenmodell in unserem Fall ein Intersektion Tabelle mit mehr als fünfzig Millionen Datensätzen erzeugt werden muss. Was uns im Produktion Upgrade Stunde kostet und dem Business keinen Mehrwert liefern.

Somit wurde Entscheidung beim alte 1:1 Adresse Modell für CONTACT zu bleiben Und die während des Repository mergers erzeugt n:m Beziehung wieder zurückzubauen.

## **Database Migration**

Eine Analyse zeigte das 15. Tabellen 90 % unserer Daten enthalten



Somit haben wir uns beim Datenbank Upgrade auf diese Tabellen konzentrieren und die Standard-Methode für alle anderen angewendet.

Unsere Spezielle Datenbank bezogene Ziele wahren:

#### **Partitionierung von großen Tabellen**

Warum Sich die Mühe machen die Tabellen zu partitionieren? Große Tabellen sind partitionierter Form einfacher von den DBA's zu verwalten. Es lassen sie alte Daten (Partitionen) Komprimieren. Es kann zu Performance Vorteilen durch Partition pruning kommen. Es erleichtert ein House keeping Seit Release 7.7 supported Siebel Partitionierte Tabellen sie lassen sich aber auch mit Release 8.1 nicht über den DDLSYNC erzeugen. Somit müssen diese manuell auch die lokale Indizes Erzeugt werden. Für große Tabelle bietet sich hier das package DBMS\_REDEF an.

#### **Umwandlung des Datentyps LONG in CLOB**

Warum Long in CLOB umwandeln? LONG Spalten sind seit Jahren von Oracle abgekündigt und generell schwierig mit SQL zu bearbeiten. Des weiteren haben Sie eine limitierte Größe.

CLOB können seit Siebel 8.0 im Siebel Repository verwaltet werden. Es wird aber keine Automatische Umwandlung angeboten. Somit muss man sich um diesen Task wieder manuell kümmern.

```
alter table mytable modify (long_column clob);
```

funktioniert wenn genügend Tempespace zur Verfügung steht Für große Tabelle bietet sich hier das package DBMS\_REDEF an. Es bleibt zu beachten dass Siebel Trotzdem nur eine maximale Datensatzgröße von 128 KB im BC Layer Unterstützt.

#### **Konsolidieren der tablespaces von 6 verschiedenen TSD/TSI zu 2 zentralen tablespaces TS\_DATA, TS\_INDEX**

Die Konsolidierung der table spaces Ist eine Maßnahme um den Administrationsaufwand Der Datenbank Administratoren zu verringern. Sie benötigt aber eine menge extra IO Leistung während der Migration und somit Zeit. aber der Upgrade ist eine der wenigen Möglichkeiten eine solche Konsolidierung umzusetzen.

Alter Table move table space für kleine und mittlere Tabellen funktioniert problemlos. Ergänzt werden kann dies durch ein Löschen von leeren Tabellen. Diese Züge der Migration von Siebel selbstständig im neuen Tablespace angelegt werden und nicht mit add Column angepasst werden müssen. Für große Tabelle bietet sich auch hier das package DBMS\_REDEF an

## **DBMS\_REDEFINITION**

Alle unsere ToDos für die 20 großen Tabellen konnten mit Die Datenbank Paket DBMS\_REDEFINITION in einem Schritt gelöst werden. Das Paket ist in der Lage die Daten Und eher ein neues Zielformat zu kopiert. Es erledigt somit in einen Schritt Das verschiebe der Daten/Tabellen in einen neuen Partitionierte Tabellen welche direkt im richtigem table space liegt und auch schon local and global Indexe enthält. Weiterhin kommen als Datenbereinigung auch einige “case when” Bedingungen angewendet und Default Werte erzeugt werden. Alles online in Hintergrund während das Siebel 7.7 Produktion System läuft. In der Downtime wird nur ein finaler Abgleich der neue Kabel durchgeführt und das DBMS\_REDEF beendet. Bei beenden tauscht Oracle die aktuellen SEA 7.7 Tabellen mit den im Hintergrund erzeugte Schattentabellen (SIA8.1) aus. Das Aufbau der Schattentabellen und Indizes Benötigte unserem Fall zwar Tage aber Das Produktionssystem konnte in dieser Zeit ohne Probleme weiter laufen. Der Austausch der Tabelle in der down Time konnte dann In Minuten durchgeführt werden. Ein Verhältnis hat der Abtretung dieser dann schon fertig mit Hilde wählen vorgefunden und Nicht weiter angefasst.

## **WIE WIR GETESTET HABEN**

Für unsere Test war im ersten Schritt das Hauptproblem – Woher eine Datenbank mit genügend Plattenplatz zu bekommen? Glücklicherweise hat der Kunde hat eine eingelagerte Exadata V1 welche Genügend Speicher für Eine Kopie alle Siebel Tabellen bereitstellen konnte.

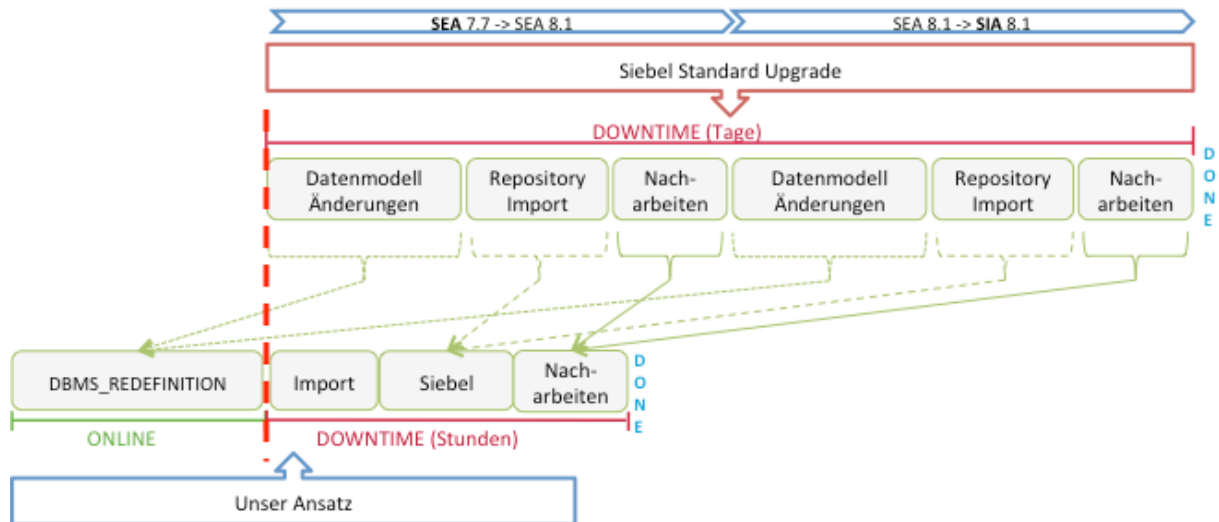
- ① Produktionsdaten wurden mit datapump auf die EXA kopiert
- ② Rudimentäre Siebel Installation für Testupgrades
- ③ Restore auf der Datenbank setzen.
- ④ Ausführen von eigenen und Standardskripten für den Upgrade
- ⑤ Prüfung des Ergebnisses
- ⑥ Verbessern der Skripte
- ⑦ Flashback der Datenbank zu Restorepunkt (3)

Schritte (4) bis (7) wurden mehrmals durchlaufen bis wir einen stabilen und ausreichend schnellen Prozess hatten

Der komplette Upgrade Prozess und der funktionale Test wurden mehrmals durchgeführt. So wurden 2 der 3 Developmentssysteme alle 3 Testsysteme und das Integration System Upgrade Prozess unterzogen



## Vergleich: Standardvorgehensweise – Unser Ansatz



Alle diese Einzelmaßnahmen könnten unsere Datenbank Ziel erreicht und die down time drastisch verkürzt werden.

## Organisation

Das Upgrade Projekt wurde in der Scrum Methode Scrum durchgeführt. Der Initiale Scope wurde in 150 User Stories gruppiert nach

- Testing
  - Test Interface A
  - Test Screen B
  - Test Smart Script C
- Infrastruktur
  - Setup Development
  - Setup Produktion
- Datenbank
  - Migrate S\_CONTACT
  - Migrate S\_EVT\_ACT

Aufgeteilt und in zweiwöchigen Sprints abgearbeitet. Dies erlaubte uns den Fortschritt des Projektes anhand der Abarbeitung der User Storys zu steuern und zu überwachen

**Kontaktadresse:**

**Riverland Reply**

**Alexander Doubek**

Arnulfstr. 27

D-80335 München

T: +49 (89) 411142-600

F: +49 (89) 411142-649

M: +49 (176) 103 582 96

E: [a.doubek@reply.de](mailto:a.doubek@reply.de)

W: [www.reply.de](http://www.reply.de)