

# Zeitversetzte Applikationsspiegelung und Oracle Flashback

Franz Diegruber  
Libelle AG  
Stuttgart

## Schlüsselworte

Oracle, Flashback, Applikationsspiegelung, Logische Fehler, Disaster Recovery.

## Einleitung

Schnell ist es passiert, dass geschäftskritische Daten z. B. durch Anwender oder auch fehlerhafte Software gelöscht oder überschrieben werden. Allerdings können gelöschte, beschädigte oder gar komplett verlorene Daten eine äußerst kostspielige Angelegenheit sein.

Die Wiederherstellung einer Applikation nach einem schwerwiegenden Ausfall oder Stillstand ist oft mit einem massiven stunden- oder gar tagelangen Aufwand verbunden, und nicht immer ist gewährleistet, dass überhaupt wieder ein sinnvoll konsistenter Datenbestand der jüngsten Vergangenheit erzeugt wird.

Ideal ist es, wenn eine innovative Datenspiegelungslösung im Einsatz ist, die automatisiert oder mit wenigen Mausklicks einfach und schnell vom Produktions- auf das Spiegelsystem umstellt, entweder auf den letzten bekannten Datenbestand vor dem Ausfall, oder auf einen früheren Zustand, z.B. vor Auftreten eines logischen Fehlers.

Die hohe Kunst der Datenwiederherstellung ist es hierbei den richtigen Zeitpunkt zu erwischen. Nicht zu früh, um keine unnötigen Datenverluste zu generieren, aber definitiv auch nicht zu spät, um den Datenbestand logisch korrekt zu halten.

Im Vortrag erfahren Sie, wie sie jetzt noch mehr Freiraum und Flexibilität zur Aktivierung eines Standby-Systems gewinnen, und so noch präziser an den idealen Zeitpunkt einer Umschaltung herankommen. Statt des bisherigen einmaligen Vorgangs der Umschaltung auf einen bestimmten Zeitpunkt (Point-In-Time) durch die Applikationsspiegelung, gibt es durch die Kombination mit Oracle Flashback nun deutlich mehr und flexiblere Möglichkeiten für den DBA die Standby-Umgebung mit der korrekten Behandlung eines logischen Fehlers zu aktivieren. Ohne Dateninkonsistenz oder unnötige Datenverluste zu riskieren und dazu sehr sparsam mit Infrastrukturressourcen umzugehen.

## Datenbank und Applikation

Bei unseren Untersuchungen zur Absicherung der Spiegelung gehen wir von zwei zu betrachtenden Bereichen aus:

- Datenbank
- Applikationsfiles

Betrachten wir zunächst die Datenbank.

## Absicherung der Datenbank

### Beschreibung Flashback und Restore Points

Hierbei ist es erst einmal elementar dass Sie sich eine Standby Datenbank aufbauen. Dies kann über Skripting, DataGuard oder Tools von Drittanbietern geschehen.

Für die Terminologie sprechen wir hier davon dass die Datenbank vom produktiven Echtserver auf den Spiegelserver aufgebaut wird.

Wir gehen von einem „Physical Standby“ aus, da ein „Logical Standby“ von vielen Applikationsanbietern und im speziellen von der SAP nicht supported wird. Ob auf der Standbyseite

die Transaktionen dabei mit oder ohne Zeitversatz appliziert werden ist davon abhängig, wie die Definitionen für RTO (Recover time objective), d.h. wie schnell das Standbysystem nach einem Ausfall wieder produktiv zur Verfügung stehen muss.

Auf diese Standbydatenbank wird nun noch zusätzlich eine Flashback Logging bzw. ein/mehrere Restore Points definiert, um kein reines PITR (Point in time recovery) zu fahren. Beim **reinen PITR** gäbe es **nicht** die Möglichkeit sich dem **Fehler anzunähern** bzw. bereits applizierte **Transaktionen** wieder **rückgängig zu machen**.

Flashback ermöglicht das zeitliche Zurücksetzen der Datenbank ohne ein zeitaufwendiges Zurückladen aller Datenbankdateien. Die Dauer für das "Zurückspulen" der Datenbank mit Flashback Database ist unabhängig von der Größe der Datenbank. Die Dauer zur Durchführung eines **PITR** hängt hauptsächlich davon ab, wie lange es dauert, bis alle Dateien der Datenbank von der Sicherung zurückgeladen sind, und wie lange das anschließende Recovery mit den Archive Logs dauert.

Beim **Flashback Database** hingegen ist die Dauer in etwa proportional zur Zeit, die die Datenbank zurückversetzt wird. Der Aufwand ist in etwa proportional zu der Menge an Änderungen, die während dieser Zeit gemacht worden sind und die nun rückgängig gemacht werden.

Oracle stellt verschiedene Level von Flashback zur Verfügung. So ist es mit „Flashback Table“ möglich einzelne Tabellen zurückzusetzen. Aufgrund der Komplexität des Datenmodells von SAP ist dies aber nicht nutzbar. So konzentrieren wir uns auf die Funktionalität von „Flashback Database“.

Mit Oracle 11g wurde Flashback dahingehend weiter optimiert, dass Flashback aktiviert werden kann während die DB geöffnet ist.

Neben dem fortlaufenden **Flashback Logging** zum Schutz gegen Datenkorruptionen aufgrund von User oder Applikationsfehlern, existiert auch noch die Absicherung mit **Restore Points**. Diese sind vor allem für geplante Aktionen zu empfehlen wie:

- SAP-Upgrade
- Datenbank-Upgrade
- Real-Applikation-Testing

Dabei existieren **normale Restore Points** und **garantierte Restore Points**. Bei normalen Restore Points besteht die Gefahr, dass notwendige Informationen aufgrund von Platzproblemen in der FRA überschrieben werden. Bei **garantierten Restore Points** besteht die Gefahr, dass die Datenbank stehen bleibt aufgrund von Platzproblemen in der FRA.

### **Architektur, Funktionsweise, Merkmale von Flashback Database**

Flashback Database basiert auf speziellen Logdateien, den sog. **Flashback Logs**. Wenn **Flashback Logging** eingeschaltet ist, werden Flashback Logs vom Oracle Hintergrundprozeß **RVWR** im laufenden Betrieb ausschließlich in den Flash Recovery-Bereich geschrieben. Oracle schreibt in regelmäßigen Abständen die veränderten Blöcke aller Datenbankdateien in Flashback Logs. Bei einem **Flashback Database** werden die seit dem Flashback-Zeitpunkt veränderten Datenblöcke aus den **Flashback Logs** gelesen und in die entsprechende Datenbankdatei zurückgeschrieben. Danach werden mit Hilfe der **Archive Logs** noch die Änderungen in der Datenbank nachgefahren, die zwischen dem Zeitpunkt, zu dem der Block in die Flashback Logs kopiert worden ist, und dem gewünschten Recovery-Zielzeitpunkt stattgefunden haben. Für Flashback Database werden also nicht nur Flashback Logs benötigt, sondern zusätzlich für den gewünschten Zeitraum (flashback retention target) auch die entsprechenden Archive Logs.

Flashback Logs werden vom Oracle Server automatisch aus dem Flash-Recovery-Bereich gelöscht, sobald sie nicht mehr benötigt werden oder der Platz im Flash Recovery Bereich für andere, wichtigere Dateien benötigt wird.

## Einschränkungen

- Voraussetzung für Flashback Database ist, daß die Datenbank physisch intakt ist! Wenn Datenbankdateien defekt sind oder fehlen, kann Flashback Database die Datenbank nicht in einen intakten Zustand zurückversetzen.
- Flashback Database kann nur Änderungen an Oracle Datenbankdateien zurücknehmen. Profildateien (spfile, init<SID>.ora), Controlfiles, Paßwortdateien, Oracle Wallet Datei, Net Services Konfigurationsdateien (sqlnet.ora etc.) werden durch Flashback Database nicht zurückgesetzt.
- DDL-Commands verhindern Flashback, d.h. wenn „drop table“ oder „Spalten hinzu/weg“ kann auf diese Tabelle nicht mehr zurück geblickt werden.
- Bestimmte Datenbankoperationen können nicht mit Flashback Database rückgängig gemacht werden. Zu diesen Operationen gehören
  - Datafile Shrink
  - Drop Tablespace
  - Drop Datafile
- Nach einer solchen Operation beginnt das Flashback Database Zeitfenster von vorne.
- Wenn die Datenbank mit Flashback Database auf einen Zeitpunkt zurückgesetzt werden soll, zu dem gerade eine NOLOGGING-Operation lief, sind die betroffenen Datenblöcke mit hoher Wahrscheinlichkeit inkonsistent/defekt.

## Flashback und SAP:

- Der Einsatz von **Flashback Table** zum Wiederherstellen von Daten im Notfall ist erlaubt, wenn die anwendungsseitige Konsistenz sichergestellt werden kann. Hierzu SAP-Hinweis 937492 beachten und im Zweifelsfall SAP hinzuziehen.
- Der Einsatz von **Flashback Database** ist für Oracle >= 10g erlaubt. Auch hierzu gibt es einen SAP-Hinweis 966117.
- **Flashback Archive / Total Recall** kann für Oracle >= 11g eingesetzt werden, jedoch gibt es dafür keinen SAP-Support.

Hier nochmals eine kurze Übersicht über die relevanten **SAP-Hinweise**:

- 105047: Welche Oracle-Tools sind SAP supported
- 966073: Beschreibungen zu und zum Einsatz von Oracle Flash Recovery Area/Fast Recovery Area
- 966117: Oracle Flashback Database Technologie

## Einrichtung und Überwachung des Flashback

Prüfen ob **Flashback** aufgrund der Lizenz **möglich**:

```
SQL> select * from v$option where parameter like '%Flashback%';
```

PARAMETER	VALUE
Flashback Table	TRUE
Flashback Database	TRUE

Parameter zur Definition des Flashback:

```
db_recovery_file_dest= <RECOVER_ROLLBACK_PATH>  
db_recovery_file_dest_size= <RECOVER_ROLLBACK_SIZE>
```

Alternativ zu RECOVER\_ROLLBACK\_SIZE könnte der Umfang der Rollback-Informationen nicht über die Größe in Bytes sondern über einen Zeitangabe in Minuten gemacht werden:

```
RECOVER_ROLLBACK_TIME
```

Der Parameter Recover-Rollback-Time gibt die Zeitspanne an, die zurückgerollt werden soll. Bei Oracle entspricht dies db\_flashback\_retention\_target. Diese Angabe kann nur dann garantiert werden, wenn in flash recovery area noch alle Rollback-Informationen vorhanden sind.

```
db_flashback_retention_target = <RECOVER_ROLLBACK_TIME>
```

Normales Flashback Logging einschalten

Die Flashback-Parameter im init<SID>.ora für die Standby DB setzen

```
>SQL alter system set db_recovery_file_dest='/oracle/AS11/rollback'  
>SQL alter system set db_recovery_file_dest_size=2048M
```

Für den Flash Recovery Bereich sollte ein schnelles Filesystem eingesetzt werden, möglichst ohne betriebssystemseitiges Caching

Flashback Logging aktivieren:

```
>SQL startup nomount;  
>SQL alter database mount standby database;  
>SQL alter database flashback on;
```

Abfrage ob flashback aktiviert ist:

```
>SQL select flashback_on from v$database;
```

YES|NO

Flashback Logging deaktivieren:

```
>SQL ALTER DATABASE FLASHBACK OFF;
```

Beim Recovern werden **Flashback-Logs (\*.flb)** in flash recovery area **geschrieben**:

```
ll /oracle/AS11/rollback/AS11/flashback  
total 754688  
-rw-r----- 1 oraas11 dba          15941632 Sep 12 14:04 o1_mf_850ym8jp_.flb  
-rw-r----- 1 oraas11 dba          15941632 Sep 12 14:04 o1_mf_850ymc1g_.flb
```

Beim Recovern ist darauf zu achten, daß Recover-Files nicht gelöscht werden.

### Flashback Retention festlegen

Der Parameter DB\_FLASHBACK\_RETENTION\_TARGET (in min) gibt an, wie lange Flashback Logs im Flash Recovery Bereich vorgehalten werden sollen.

Oracle Default: 1440 (=24h)

Empfehlung: je nach Anforderung, jedoch nicht unter 60 min

Beispiel: Flashback Retention = 5 Stunden

```
>SQL alter system set db_flashback_retention_target=300;
```

Einen Überblick über den aktuellen Status des Flashback Logging gibt die View V\$FLASHBACK\_DATABASE\_LOG.

```
>SQL select * from v$flashback_database_log;
```

OLDEST_FLASHBACK_SCN	OLDEST_FL	RETENTION_TARGET	FLASHBACK_SIZE
ESTIMATED_FLASHBACK_SIZE			
-----	-----	-----	-----
-----			
0	23594651	12-SEP-12	1440
			31866880

Flashback Database kann maximal soweit in die Vergangenheit gehen, wie die älteste SCN in den Flashback Logs. Da Flashback Logs nicht außerhalb der Flash Recovery-Bereich gesichert werden können, ist die einzige Möglichkeit, dafür zu sorgen, daß die Flashback Retention auch eingehalten wird, genügend Platz im Flash Recovery-Bereich bereitzustellen.

Abfrage der **ältesten SCN/Zeit**, die mit flashback **erreichbar** ist:

```
>SQL SELECT OLDEST_FLASHBACK_SCN, TO_CHAR(OLDEST_FLASHBACK_TIME, 'YYYY-MON-DD HH24:MI:SS') FROM V$FLASHBACK_DATABASE_LOG;
```

OLDEST_FLASHBACK_SCN	TO_CHAR(OLDEST_FLASH
-----	-----
23594651	2012-SEP-12 13:53:06

**ACHTUNG:** Durch bestimmte Datenbankoperationen wird das Flashback-Fenster neu gestartet (z.B. Drop Tablespace). Diese Query berücksichtigt diese Beschränkung leider nicht!!

### Max. Zeit Rollforward

Dieser Zeitpunkt wird vom zuletzt kopierten Archivefile (ArchiveNum -1) anhand des File-Timestamps ermittelt.

Alternativ kann die last\_scn aus dem Archivefile ermittelt und durch scn\_to\_timestamp der zugehörige Timestamp.

Stand "Datenbank / State of mirror"

Der Wert Stand Datenbank kann durch folgende Abfrage für Oracle 11 genauer als bisher ermittelt werden

```
>SQL select current_scn from v$database;
```

Statistik von Flashback abfragen

Zur Überwachung der Systemperformance dienen die beiden Views V\$FLASHBACK\_DATABASE\_STAT und V\$SYSSTAT

```
>SQL      select      TO_CHAR(BEGIN_TIME,      'YYYY-MON-DD      HH24:MI:SS'),
TO_CHAR(END_TIME,      'YYYY-MON-DD      HH24:MI:SS'),      FLASHBACK_DATA,      DB_DATA,
REDO_DATA, ESTIMATED_FLASHBACK_SIZE from V$FLASHBACK_DATABASE_STAT;
```

```
TO_CHAR(BEGIN_TIME, ' TO_CHAR(END_TIME, 'YY      FLASHBACK_DATA      DB_DATA
REDO_DATA ESTIMATED_FLASHBACK_SIZE
```

```
-----
-----
2012-SEP-12 14:04:57 2012-SEP-12 14:21:56      16384      0
0      0
```

Platzüberwachung der Fast Recovery Area

```
>SQL SELECT * FROM V$RECOVERY_FILE_DEST;
>SQL SELECT * FROM V$FLASH_RECOVERY_AREA_USAGE;
```

Überblick über die Flashback Logs

```
>SQL select * from v$flashback_database_logfile;
```

Abschätzung des Speicherbedarfs für Flashback Logs

Um den Speicherbedarf für die Flashback Logs abzuschätzen, dient folgende Abfrage:

```
>SQL SELECT ESTIMATED_FLASHBACK_SIZE FROM V$FLASHBACK_DATABASE_LOG;
```

Damit das Ergebnis dieser Abfrage auch annähernd richtig ist, muß zuvor Flashback Logging eingeschaltet sein, die Flashback Retention muß festgelegt sein und die Datenbank sollte eine Weile mit normaler, typischer Last gelaufen sein.

## Einrichtung und Überwachung Restore Points

### Normalen Restore Point anlegen (NRP)

Bevor man einen normalen Restore Point erstellen kann, muss Flashback Logging bereits aktiviert sein.

```
>SQL CREATE RESTORE POINT <NAME>;
```

Anmerkung: V\$RESTORE\_POINT.STORAGE\_SIZE=0 für normale Restore Points

### Garantierten Restore Point anlegen (GRP)

```
>SQL CREATE RESTORE POINT <NAME> GUARANTEE FLASHBACK DATABASE;
```

Vor dem Anlegen eines garantierten Restore Points ist es nicht notwendig, vorher den normalen Flashback Modus zu aktivieren. Im Gegenteil, dies ist aus Platz- und Performancegründen nachteilig und daher nicht zu empfehlen.

### Alle Restore Points anzeigen

```
>SQL SELECT * from V$RESTORE_POINT;
```

### Garantierte Restore Points anzeigen

```
>SQL SELECT * from V$RESTORE_POINT WHERE guarantee_flashback_database = 'YES';
```

### Wieviel Platz wird pro NRP / GRP an Flashback Logs belegt

```
>SQL SELECT NAME, SCN, STORAGE_SIZE FROM V$RESTORE_POINT;
```

```
>SQL SELECT NAME, SCN, STORAGE_SIZE FROM V$RESTORE_POINT WHERE GUARANTEE_FLASHBACK_DATABASE = 'YES';
```

### Wieviel Platz belegen alle GRP Flashback Logs

```
>SQL SELECT SUM(STORAGE_SIZE) FROM V$RESTORE_POINT WHERE GUARANTEE_FLASHBACK_DATABASE = 'YES';
```

### Löschen von Restore Points

Kommando zum expliziten Löschen eines Restore Points:

```
>SQL DROP RESTORE POINT <name>;
```

Ein normaler Restore Point wird automatisch aus dem Controlfile gelöscht, wenn er älter ist als im Parameter CONTROL\_FILE\_RECORD\_KEEP\_TIME spezifiziert ist. Ein GRP bleibt solange aktiv und im Controlfile bestehen, bis er explizit gelöscht worden ist.

### Übersicht über incarnation

```
>SQL select * from v$database_incarnation;
```

### **Absicherung der Applikation:**

Neben der Datenbankebene gilt es jedoch auch die Applikationsebene zu betrachten. Zahlreiche Applikationen schreiben im laufenden Betrieb umfangreiche Daten außerhalb der Datenbank. Diese Applikationen sind nur dann mit einem aktuellen Stand auf dem Standby-System lauffähig, wenn diese Daten in das Ausfallrechenzentrum übertragen werden. Hier ist z.B. SAP mit dem Java-Stack, Profilen, Transportverzeichnissen etc., sowie den eigentlichen Programm-Binaries zu nennen. Aus diesem Grund ist es in vielen Fällen nicht ausreichend auf eine reine Datenbankspiegelung zurückzugreifen. Es bedarf eines integrierten Ansatzes zur ganzheitlichen Applikationsspiegelung. Hierfür bietet sich neben der individuellen Entwicklung von Skripten auch die Möglichkeit auf den Einsatz entsprechender Produkte von Drittanbietern, wie beispielsweise der Firma Libelle.

Für die Spiegelung der Applikationsfiles kann auf

- Betriebssystemmittel
- Stagemittel
- Produkte von Drittanbietern zurückgegriffen werden

Die entsprechenden, zur Applikation gehörenden Flat Files können z.B. mittels rsync (Unix) bzw. robocopy (Windows) kopiert werden. Möglichst identische Filesystemstrukturen auf Produktiv- und Standby-System erleichtern die Kopie und später auch den möglichen Backup und Restore.

Im laufenden Betrieb muss auf Filesystemebene nach editierten und gelöschten Files und Directories gesucht werden, genauso wie nach Permission- und owner:group-Änderungen. Volumen und Peaks können in der Datenbank über die Information der Logswitches gewonnen werden. Die Ermittlung des Volumens der Änderungen pro Tag und Peaks auf Flat File-Ebene kann nur auf historischen Erfahrungswerten basieren. Auch gilt es zu ermitteln, inwieweit die Daten komprimiert werden können und welche CPU-Auslastung hierfür notwendig ist.

Wichtig ist auch, dass die Prüfung nach Veränderungen in möglichst kleinen Zyklen erfolgt, um im Falle einer Aktivierung auf dem Spiegelsystem die DB und Applikation möglichst konsistent zueinander zur Verfügung stellen zu können.

Für eine SAP-Umgebung empfiehlt es sich folgende Verzeichnisse und Files zu überwachen:

- /usr/sap/SID/SYS/profile **besser:** /sapmnt/SID/profile: spiegeln
- /usr/sap/SID/SYS/global **besser:** /sapmnt/SID/global: spiegeln
- /usr/sap/SID/SYS/global/001JOBLOG: **nicht** spiegeln, da häufige Änderungen)
- /usr/sap/SID/SYS/exe **besser:** /sapmnt/SID/exe: spiegeln
- /usr/sap/trans eigentlich auf Profile-Parameter „DIR\_TRANS“ gehen: wenn lokal, dann spiegeln
- /usr/sap/SID/DVEBMGS00/sec: spiegeln
- /usr/sap/SID/DVEBMGS00\SDM: spiegeln
- /usr/sap/SID/DVEBMGS00\j2ee: spiegeln
- /usr/sap/SID/DVEBMGS00\j2ee\cluster\server0\log: Protokollinfos! Häufige Änderungen
- /usr/sap/SID/DVEBMGS00/exe **nicht** spiegeln, da dies über das ../SYS/exe gespiegelt werden sollte. Somit wird die open-file-Problematik unter Windows vermieden.



## Mögliche Szenarien und Behandlung

Weiter oben wurde bereits erwähnt dass die Standby-Datenbank mit oder ohne Zeitversatz beim Applizieren der Transaktionen gefahren werden kann. Hierzu sollte der RTO betrachtet werden, aber auch dass ein Grossteil der Umschaltungen in logischen Fehlern begründet liegen.

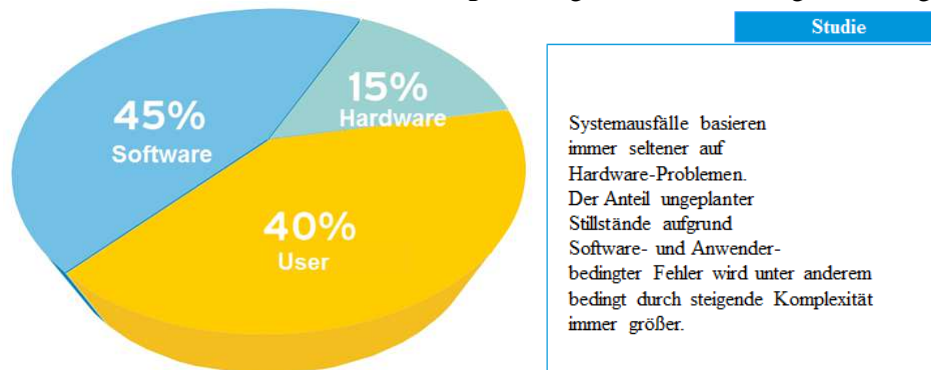


Abb. 1: Ursachen für Systemausfälle

Laut Studien lassen sich etwa 70% aller Ausfälle auf logische Fehler zurückführen. Unter logischen Fehlern sind Datenkorruptionen oder -löschungen aufgrund von Benutzer- oder Softwarefehlern zu verstehen. Nur durch ein zeitversetztes Recovery kann auf diese reagiert werden.

## Übernahmearten

Im Fehlerfall sollten Datenbank und Applikation möglichst schnell und weitgehend automatisiert auf dem Standby-System aktiviert werden können. Dabei muss aber vorab schon definiert sein wie auf welche Szenarien reagiert werden soll.

Grundsätzlich gibt es zwei Arten der Umschaltung.

**Switchover:** geplante Übernahme des Betriebs

**Failover:** bei einem Ausfall der Primärdatenbank den Betrieb übernehmen.

Den Switchover werden wir nicht weiter betrachten. Hier kann ein PITR gefahren werden und die DB und Applikation auf dem Standbyserver aktiviert werden.

Der Failover lässt sich noch weiter aufteilen in Umschaltung aufgrund von:

- Bei **Hardwareausfällen** empfiehlt es sich auf einen möglichst aktuellen Zeitpunkt zu recovern. Auf welchen Zeitpunkt maximal reconvert werden kann ist hier abhängig, wann die letzten Änderungsdaten vor dem Fehler übertragen wurden. Hierbei wird mit DB, App, VIP produktiv auf das Spiegelsystem gegangen.
- Bei **logischen Fehlern** soll lediglich bis kurz vor den Zeitpunkt des logischen Fehlers reconvert werden. Für die Analyse des Zeitpunkts des logischen Fehlers soll hier das Flashback genutzt werden.
  - Hier gäbe es die Möglichkeit weitestgehend nah an den logischen Fehler heranzufahren und anschließend über einen **Datenexport auf dem Standbysystem** und einen **Datenimport auf dem Echtsystem** weiterhin produktiv auf Echtsystem zu bleiben. Aufgrund des komplexen Datenbankmodells wird dies jedoch in einer SAP-Umgebung selten der Fall sein.
  - Deshalb gehen wir davon aus, dass auch hier mit DB, App, VIP **produktiv auf das Spiegelsystem** gegangen wird.

Ist die Möglichkeit gegeben, dass im Notbetrieb über eine Virtuelle IP-Adresse (**VIP**) und **Aliasname** die bisherige Konnektivität wieder verfügbar gemacht werden kann, sind keine Anpassungen für den connect notwendig. Für eine SAP-Umgebung mit **Java-Stack** ist eine Aktivierung mit dem

identischen Aliasname **unbedingt erforderlich**, da ansonsten tiefere Anpassungen im Java-Stack notwendig sind.

Dennoch kann es für die Umschaltung erforderlich sein, Prozesse auf Applikationsservern im Netzwerk durchzustarten. Dies kann über Agents oder Skripte auf den Applikationsservern umgesetzt werden, welche bei der Umschaltung angetriggert werden.

### **Rantasten und Überprüfung**

Tritt ein gravierendes Problem mit der DB und/oder der Applikation auf, sollte die Überlegung sein das **produktive System zu Stoppen**. Ansonsten werden weiterhin Transaktionen durchgeführt die ggfs. verloren gehen, wenn auf einen Zeitpunkt in der Vergangenheit reconvert wird.

#### Open Read Only

Die Datenbank kann auf einen Zeitpunkt in der Zukunft reconvert und Open Read Only geöffnet werden. Durch RECOVER\_ROLLBACK=TRUE besteht zusätzlich die Möglichkeit, die Standby Datenbank auf einen Zeitpunkt in der Vergangenheit zurückzurollen und Open Read Only zu öffnen.

Wird als Zeitpunkt 2012-09-12 13:25:00 gewählt, lauten die Befehlsfolgen dazu:

```
>SQL flashback database to time to_timestamp( '2012-09-12 13:25:00', 'YYYY-MM-DD HH24:MI:SS');
>SQL alter database open read only;
```

Ebenso möglich sind flashback auf:

#### TO SCN

```
>SQL FLASHBACK DATABASE TO SCN <number>;
```

#### TO TIME

```
>SQL FLASHBACK DATABASE TO TIME "TO_DATE('09/20/00','MM/DD/YY')";
```

#### TO RESTORE POINT

```
>SQL FLASHBACK DATABASE TO RESTORE POINT <name of restore point>;
```

Wird der Recover wieder gestartet, befindet sich die Standby Datenbank auf Stand '2012-09-12 13:25:00'.

Beginnend ab dem OpenReadOnly Zeitpunkt kann das Rollforward Recovery wieder gestartet werden.

```
>SQL shutdown;
>SQL startup nomount;
>SQL alter database mount standby database;
>SQL recover standby database until cancel;
```

Recover database und flashback database lassen sich beliebig oft wiederholen, solange die zugehörigen Recover-Files noch existieren.

Datenbank im **READ ONLY** Modus öffnen, um zu prüfen, ob der Zeitpunkt richtig gewählt war:

```
>SQL ALTER DATABASE OPEN READ ONLY;
```

Für das weitere Vorgehen gibt es verschiedene Möglichkeiten:

Option 1: Datenbank wieder bis zum aktuellen Zeitpunkt recovern

```
>SQL RECOVER DATABASE;
```

Dadurch wird das Flashback Database Kommando wieder rückgängig gemacht und die Datenbank wieder in den aktuellen Zustand gebracht.

Option 2: Nochmaliges Flashback Database zu einer früheren SCN

```
>SQL FLASHBACK DATABASE TO SCN <scn number>;
```

Wenn der Zeitpunkt falsch (zu spät) gewählt war, kann man durch ein weiteres Flashback-Kommando noch weiter in die Vergangenheit gehen.

Option 3: Datenbank recovern zu einer späteren SCN

```
>SQL RECOVER DATABASE UNTIL SCN <scn number>;
```

Wenn der Zeitpunkt falsch (zu früh) gewählt war, kann man wieder in Richtung des aktuellen Zeitpunkts recovern.

Option 4: Datenbank für produktiven Gebrauch öffnen

```
>SQL ALTER DATABASE OPEN RESETLOGS;
```

Diese Möglichkeit wird i.d.R. dann gewählt, wenn der Flashback Database Zeitpunkt richtig gewählt worden ist und das System mit diesem Zeitpunkt wieder aufsetzen soll.

Hierbei ist zu beachten, daß es **nicht möglich** ist, das **SAP-System zu starten**, wenn die Datenbank nur im **READ ONLY** Modus geöffnet ist!

Muss für die Überprüfung des sauberen Stands das SAP auf dem Standbyserver gestartet werden, ist die DB im „**READ WRITE**“ Mode zur Verfügung zu stellen.

Bevor das SAP zur Überprüfung gestartet wird, sollten allerdings folgende Dinge beachtet werden:

- Es sollten keine Batch-Prozesse loslaufen
- Schnittstellen sollten nicht aktiviert sein

```
>SQL select open_mode, database_role, flashback_on from v$database;
```

OPEN_MODE	DATABASE_ROLE	FLASHBACK_ON
MOUNTED	PHYSICAL STANDBY	YES

```
>SQL alter database convert to snapshot standby;
```

Database altered.

```
>SQL select open_mode, database_role, flashback_on from v$database;
```

OPEN_MODE	DATABASE_ROLE	FLASHBACK_ON
MOUNTED	SNAPSHOT STANDBY	YES

```
>SQL alter database open;
```

Database altered.

```
>SQL select open_mode, database_role, flashback_on from v$database;
```

OPEN_MODE	DATABASE_ROLE	FLASHBACK_ON
READ WRITE	SNAPSHOT STANDBY	YES

Hier kann nun das SAP auf dem Standbyserver gestartet werden und überprüft werden, ob auf den richtigen Zeitpunkt hinrecovert wurde.

Option A:

Wenn **ja**, kann mit dieser Stand produktiv gegangen werden. Da beim Umwandeln in die „database\_role“ „snapshot standby“ ein GRP erstellt wurde muss dieser manuell gelöscht werden.

Auszug Oracle alert-File:

...

```
alter database convert to snapshot standby
```

```
Created guaranteed restore point SNAPSHOT_STANDBY_REQUIRED_09/12/2012 16:08:42
```

...

Option B:

wenn **nein**, muss die DB wieder in den Zustand „physical standby“ versetzt werden und mittels flashback ein weiterer möglicher Zeitpunkt der DB hergestellt werden.

```
>SQL shutdown immediate;
```

Database closed.

Database dismounted.

ORACLE instance shut down.

```
>SQL startup mount;
```

...

Database mounted.

```
>SQL select open_mode, database_role, flashback_on from v$database;
```

OPEN_MODE	DATABASE_ROLE	FLASHBACK_ON
MOUNTED	SNAPSHOT STANDBY	YES

```
>SQL alter database convert to physical standby;
```

Database altered.

```
>SQL select open_mode, database_role, flashback_on from v$database;
```

```
select Open_mode, database_role, flashback_on from v$database
*
```

ERROR at line 1:

ORA-01507: database not mounted

```
>SQL shutdown immediate;
```

```
ORA-01507: database not mounted  
ORACLE instance shut down.
```

```
>SQL startup mount;
```

```
...  
Database mounted.
```

```
SQL> select open_mode, database_role, flashback_on from v$database;
```

OPEN_MODE	DATABASE_ROLE	FLASHBACK_ON
-----	-----	-----
MOUNTED	PHYSICAL STANDBY	YES

Wurde erfolgreich auf das Standby-System umgeschaltet, die Applikation aktiviert und die Benutzer mit diesem verbunden, ist die Grundlage für die **Weiterführung des Geschäftsbetriebes** hergestellt. Jedoch sollte auch bedacht werden wie der Umzug **zurück** auf das Produktiv-System realisiert werden kann. Meist hat das Standby-System eine geringere Performance und/oder die Konnektivität ist eingeschränkt. Durch entsprechende Vorarbeiten im normalen Betrieb kann vermieden werden, dass die Spiegelung eine reine **Einbahnlösung** ist.

### Umsetzung des Flashback in Libelle

Im DBShadow von Libelle ist das Handling von Flashback umgesetzt.

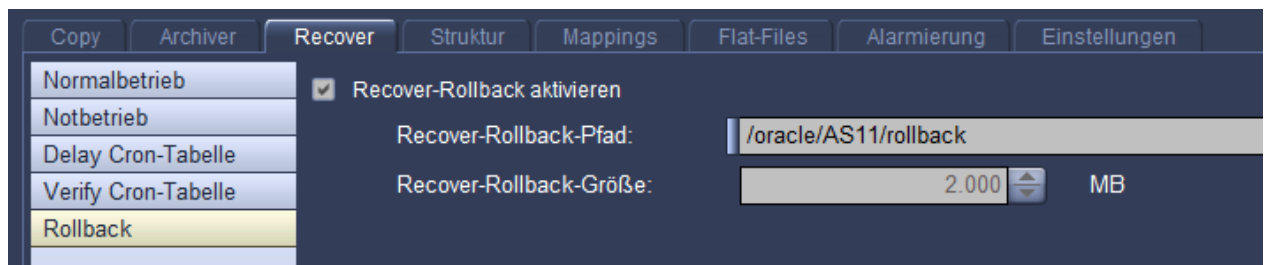


Abb. 2: Definition von FRA und Grösse

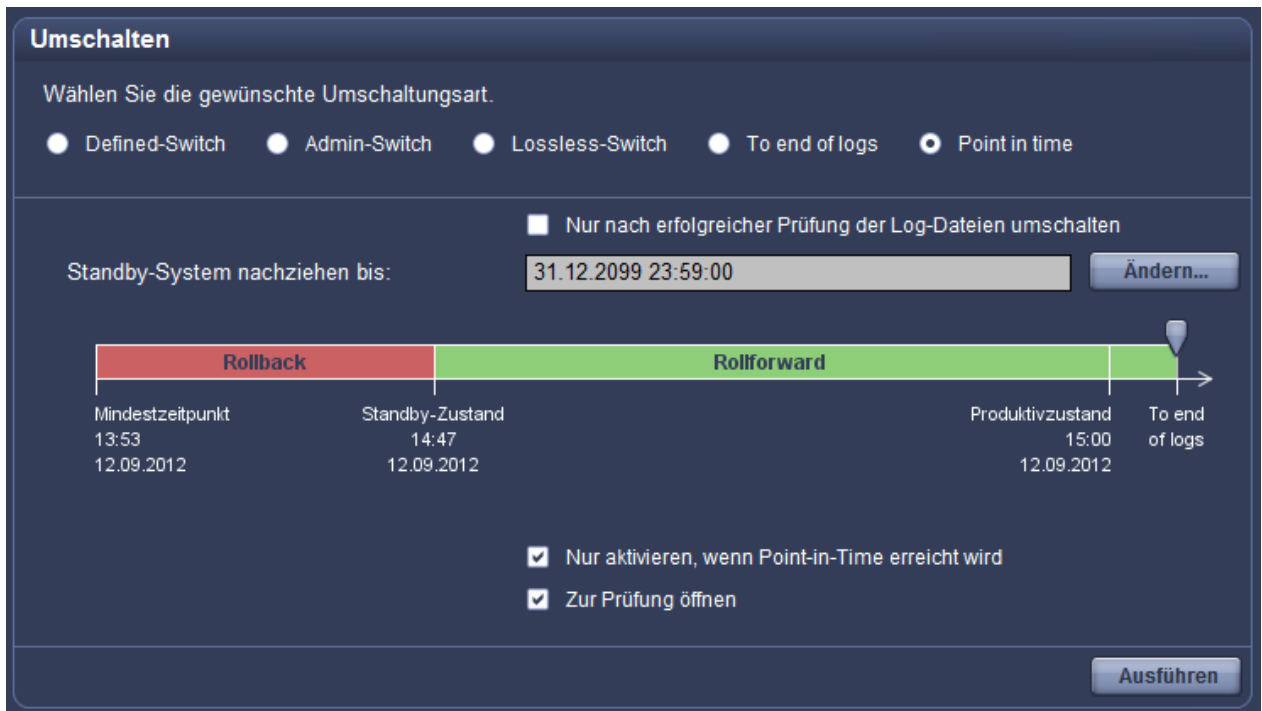


Abb. 3: Definition mit welchem Zeitpunkt soll Standby-DB READ ONLY geöffnet werden

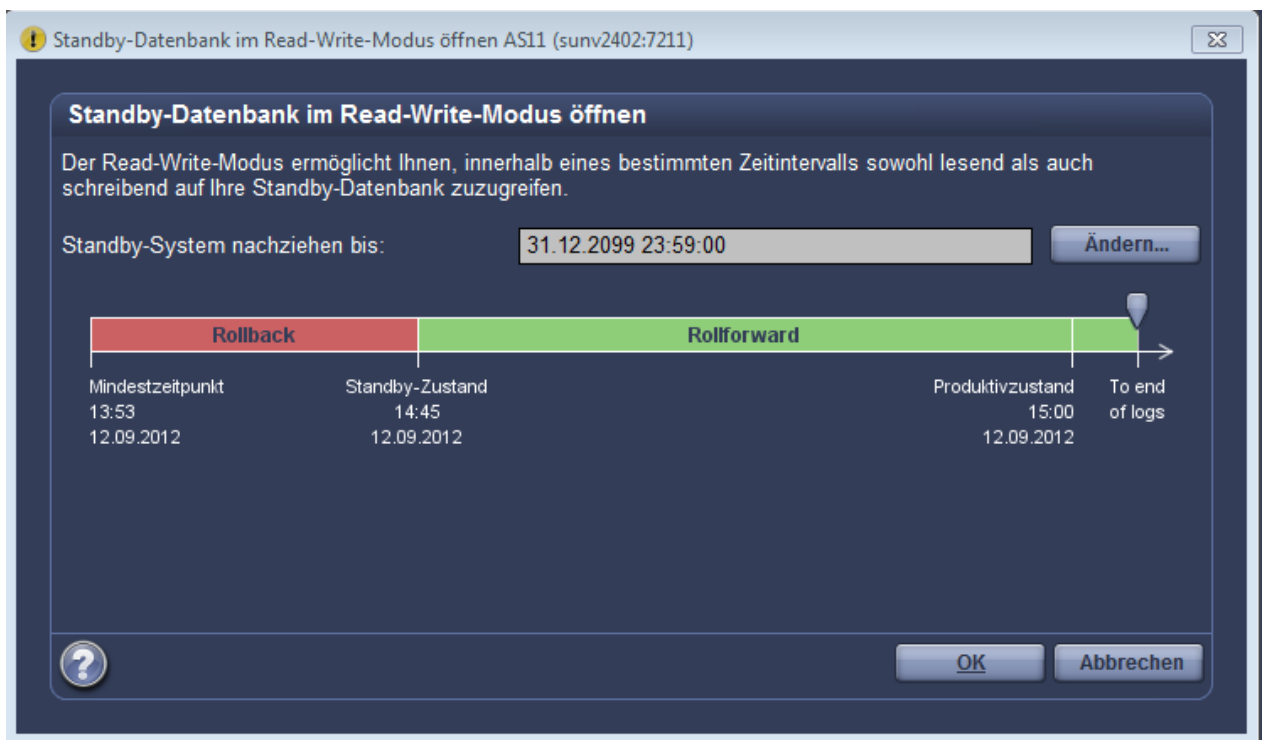


Abb. 4: Definition mit welchem Zeitpunkt soll Standby-DB READ WRITE geöffnet werden

### **Zusammenfassung**

Flashback bietet gute Möglichkeiten sich an logische Fehler heranzutasten und schnell die DB mit einem sauberen Stand zu aktivieren. Damit lassen sich die heute geforderten strengen RTOs erreichen. Jedoch gilt es auch die Applikationsebene zu betrachten und sicher zu stellen dass DB und Applikationsfiles einen konsistenten Stand darstellen.

In einer SAP-Landschaft gilt es weiterhin zu beachten dass es erforderlich nicht nur ein System isoliert zu betrachten, sondern dass über die gesamte Landschaft eine Konsistenz gewährleistet werden muss (RCO = recovery consistency objective). Hier kommt es zu einer weiteren Komplexität.

Deshalb ist es unbedingt erforderlich die Prozesse und Prüfungen genau zu definieren und schriftlich festzuhalten, damit in einer Stresssituation alles eindeutig durchlaufen werden kann. Hierbei ist es eine Frage des Geschmacks wie weit automatisiert wird (möglicherweise schneller) oder ob manuell jeder Schritt durchlaufen werden soll (flexibler um auf spezielle Situationen und Fehler reagieren zu können).

### **Kontaktadresse:**

Franz Diegruber  
Libelle AG  
Gewerbestr. 42  
D-70565 Stuttgart

Telefon: +49 (0) 711-78335 312  
Fax: +49 (0) 711-78335 340  
E-Mail [fdiegruber@libelle.com](mailto:fdiegruber@libelle.com)  
Internet: [www.libelle.com](http://www.libelle.com)