

## Tipps & Tricks: Oktober 2005

Bereich:	DBA, B&R	Erstellung:	10/2005 MM
Versionsinfo:	8.1.1.7, 9.2.0.8, 10.2.0.4, 11.1.0.6	Letzte Überarbeitung:	07/2009 MM

 [Als PDF Downloaden!](#)

## Recoverymöglichkeiten im NoArchivelog-Modus

Sie können es sich erlauben Ihren Datenbank-Server regelmäßig herunterzufahren, um ihn offline zu sichern? Darüber hinaus halten Sie nichts von Archivierung oder wollen Ihrem Datenbanksystem zusätzliche Arbeitslast ersparen?

Auf die wesentlichen Nachteile (u.a. SGA-Verlust, Nicht-Verfügbarkeit und im Fehlerfall in der Regel Datenverlust) dieser Backup-Methode soll an dieser Stelle nicht näher eingegangen werden. Vielmehr sollen Ihnen Möglichkeiten aufgezeigt werden, wie Sie unter gewissen Umständen (und mit etwas Glück) auch beim Offline Backup ohne Archivierung verloren geglaubte Daten wiederherstellen bzw. mit dem aktuellen Datenbestand weiterarbeiten können.

### Durchführung eines Offline Backups

Beim Offline Backup werden nach dem Stoppen der Instanz alle Control, Redo-Log- und Daten-Dateien in ein geeignetes Backup-Verzeichnis kopiert. Danach wird die Instanz wieder gestartet. Ein Skript zur automatisierten Durchführung finden Sie [hier](#).

Für die Durchführung des Backups selbst, ist es dabei unerheblich, ob die Archivierung eingeschaltet worden ist oder nicht.

### Möglichkeiten der Wiederherstellung (Recovery) im Fehlerfall

Für die nachfolgende Fallunterscheidung gehen wir davon aus, dass jeden Freitag Abend um 20:00 Uhr ein Offline Backup angestoßen wird. Am nachfolgenden Montag wird die Instanz wieder gestartet. Es findet keine Archivierung der Online Redo-Log-Dateien statt.

Kommt es bei dieser Konstellation nun zu einem Fehlerfall, z.B. wird eine der Daten-Dateien beschädigt, dann muss das letzte Backup komplett eingespielt und somit die Datenbank auf den Stand der letzten Sicherung gebracht werden. Die Konsequenz ist immer der Verlust der letzten Änderungen.

Das muss aber nicht zwangsläufig so sein!

Für die Mehrzahl an Fällen trifft dies sicherlich zu, unter bestimmten (glücklichen) Umständen kann aber auf das Zurücksetzen der gesamten Datenbank verzichtet werden. Und diese Umstände sollen nun vorgestellt und die entsprechenden Lösungsmöglichkeiten aufgezeigt werden.

#### Fall 1

Alle Änderungen, die seit dem letzten Backup durchgeführt worden sind, befinden sich in der aktuellen Redo-Log-Datei. Durch folgenden SELECT können Sie die aktuelle herausfinden:

```
SELECT t1.group#, t1.sequence#, t1.status, t2.member
FROM v$log t1, v$logfile t2
WHERE t1.t1.group# = t2.group#;
```

```
GROUP# SEQUENCE# STATUS    MEMBER
-----
1      1013 INACTIVE C:\ORACLE\ORADATA\O11G\REDO_01.LOG
2      1011 INACTIVE C:\ORACLE\ORADATA\O11G\REDO_02.LOG
3      1014 CURRENT C:\ORACLE\ORADATA\O11G\REDO_03.LOG
```

Die Datei REDO\_03.LOG ist demnach die aktuell beschriebene Log-Datei.

Für das Testen von Fall 1 erzeugen wir uns im Tablespace USERS eine Tabelle TEST und fügen eine Zeile hinzu. Nach einem COMMIT erhalten wir die Bestätigung, dass die Änderungen dieser Transaktion festgeschrieben sind und sich - spätestens nach einem Checkpoint - auch in der Daten-Datei USERS01.DBF befinden.

```
CREATE TABLE test (nr NUMBER);
INSERT INTO test VALUES (1);
COMMIT;
```

Anschließend fällt genau diese Datei aus. Spielen wir nun das Backup der Datei ein, ist diese natürlich auf einem alten (Checkpoint-)Stand. Die Tabelle TEST fehlt. Um die Datei auf den aktuellen Stand zu bringen werden sämtliche Informationen aus der aktuellen Redo-Log-Datei benötigt.

```
SHUTDOWN ABORT
HOST COPY <backup_dir>\users01.dbf c:\oracle\oradata\o11g
STARTUP
```

In der Mount-Phase bricht der Startvorgang mit folgender Fehlermeldung ab:

```
ORA-01113: Für Datei '4' ist Datenträger-Recovery notwendig
ORA-01110: Datendatei 4: 'c:\oracle\oradata\o11g\users01.dbf'
```

Die Datenbank lässt sich nicht öffnen, da nicht alle Daten-Dateien denselben Checkpointstand haben. Durch ein Recovery muss die alte Datei aktualisiert werden, erst dann ist ein Öffnen möglich. Es wird ein sog. Rollforward durchgeführt.

```
RECOVER TABLESPACE users
-- ebenfalls möglich
RECOVER DATAFILE 4
ALTER DATABASE OPEN;
```

Der Wiederherstellungsvorgang war erfolgreich und die Tabelle TEST steht uneingeschränkt zur Verfügung.

## Fall 2

Seit dem letzten Backup ist mindestens ein Logswitch durchgeführt worden, es ist aber noch keine Online Redo-Log-Datei zum zweiten Mal beschrieben worden. Das bedeutet, dass sich sämtliche Änderungen nicht mehr allein in der aktuellen Redo-Log-Datei befinden, sondern auf mehrere erstrecken.

Wie in Fall 1 wird eine Tabelle TEST im Tablespace USERS angelegt. Die vielen Änderungen werden durch zwei Logswitches simuliert.

```
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM SWITCH LOGFILE;
```

Jetzt tritt in der Datei USERS01.DBF ein Fehlerfall ein.

Die Lösung dieses Problems entspricht genau der Vorgehensweise im Fall 1. Auch hier lässt sich durch ein Recovery in der Mount-Phase die veraltete Daten-Datei auf den aktuellen Stand bringen. Die Wiederherstellung ist erfolgreich abgeschlossen und die Datenbank kann geöffnet werden.

Hinweis:

Der Recovery-Prozess muss nicht zwingend in der MOUNT-Phase angestoßen werden. Falls es sich "nur" um eine Datei eines Daten-Tablespace handelt (also nicht SYSTEM oder UNDO) und die Instanz nicht bereits heruntergefahren wurde, kann die Wiederherstellung auch in der OPEN-Phase vorgenommen werden. Dazu muss lediglich die Daten-Datei OFFLINE gesetzt, durch die Backup-Datei ersetzt, recovert und abschließend wieder ONLINE gesetzt werden.

### Fall 3

Im dritten Fall wird - ebenfalls nach dem letzten Backup - die Tabelle TEST im Tablespace USERS erzeugt. Die Vielzahl an Änderungen, die im Anschluss durchgeführt werden, ziehen eine Vielzahl an Logswitches (mindestens drei bei drei Online Redo-Log-Dateien) nach sich. Jetzt kommt es zu einem Fehlerfall, bei dem die Daten-Datei INDX01.DBF (enthält lediglich Indizes) beschädigt wird.

Die Datei kann nicht recovert werden, da sich nicht mehr alle Transaktionen in den Redo-Log-Dateien befinden. Würde aber die gesamte Datenbank zurückgesetzt werden, wäre die "wichtige" Tabelle TEST verloren.

An dieser Stelle bleibt nur die Alternative, auf die defekte Datei zu verzichten, die Datenbank ohne sie zu öffnen und sie anschließend wieder, samt Inhalte, neu anzulegen.

```
SHUTDOWN ABORT
STARTUP MOUNT
ALTER DATABASE DATAFILE 'c:\oracle\oradata\o11g\indx01.dbf' OFFLINE DROP;
ALTER DATABASE OPEN;
DROP TABLESPACE indx INCLUDING CONTENTS AND DATAFILES;
CREATE TABLESPACE indx DATAFILE 'c:\oracle\oradata\o11g\indx01.dbf'
    SIZE 20M REUSE;
CREATE INDEX ...
```

Für den Fall, dass auf die beschädigte Datei nicht verzichtet werden kann, bleibt nur noch das Einspielen des gesamten letzten Backups und damit das Zurücksetzen der Datenbank. Eine Wiederherstellung auf den aktuellen Stand ist hier nicht möglich.

### Fazit

Die Aussage "Ohne Archivierung ist im Fehlerfall keine Wiederherstellung möglich" ist nicht allgemeingültig. Es bedarf allerdings besonderer Umstände - und die treten im Ernstfall leider eher selten ein - die ein Recovery im NoArchivelog-Modus möglich machen.

Also gilt: Wenn schon Offline Sicherung, dann wenigstens mit eingeschalteter Archivierung!! Achten Sie bis einschließlich Version 9.2 auf die Automatisierung der Archivierung

```
ALTER SYSTEM SET log_archive_start = true SCOPE=SPFILE;
```