

# Microsoft SQL-Server 2005 - Teil 2

Christian Zahler

## 10 Datenbank-Snapshots

Seit SQL Server 2005 werden Datenbank-Snapshots unterstützt. Datenbankssnapshots sind nur in der Enterprise Edition von Microsoft SQL Server 2005 verfügbar. Alle Wiederherstellungsmodelle unterstützen Datenbankssnapshots.

Sie sind schreibgeschützt und bieten eine statische Datenbanksicht (der Quelldatenbank). Für jede Quelldatenbank können mehrere Snapshots vorhanden sein. Diese befinden sich immer auf derselben Serverinstanz wie die Datenbank. Die einzelnen Datenbankssnapshots sind hinsichtlich der Transaktionen mit der Quelldatenbank zum Zeitpunkt der Snapshoterstellung konsistent. Ein Snapshot besteht immer nur bis zu dem Zeitpunkt, zu dem er vom Besitzer der Datenbank explizit gelöscht wird.

Datenbankssnapshots sind von der Quelldatenbank abhängig. Die Snapshots einer Datenbank müssen sich auf der gleichen Serverinstanz wie die Datenbank selbst befinden. Ist diese Datenbank außerdem aus irgendeinem Grund nicht verfügbar, stehen die zugehörigen Datenbankssnapshots ebenfalls nicht zur Verfügung.

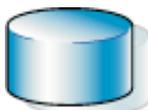
### Funktionsweise

Datenbankssnapshots arbeiten auf der Ebene der Datenseiten. Bevor eine Seite der Quelldatenbank zum ersten Mal geändert wird, wird die Originalseite der Quelldatenbank auf den Snapshot kopiert. Dieser Vorgang wird als Kopie bei Schreibvorgang bezeichnet. Im Snapshot wird die Originalseite gespeichert, wodurch die Datensätze in dem Zustand erhalten werden, wie sie zum Zeitpunkt der Snapshoterstellung vorhanden waren. Nachfolgende Aktualisierungen von Datensätzen in einer geänderten Seite wirken sich nicht auf den Inhalt des Snapshots aus. Der gleiche Vorgang wird für jede Seite wiederholt, die zum ersten Mal geändert wird. Auf diese Weise bleiben im Snapshot die Originalseiten für alle Datensätze erhalten, die seit dem Erstellen des Snapshots geändert worden sind.

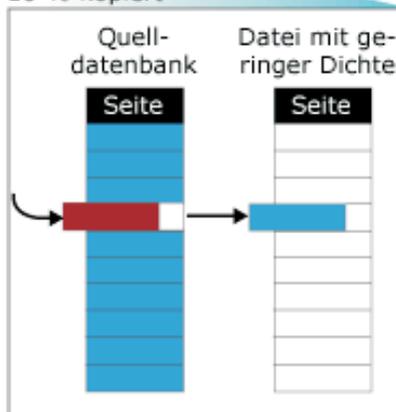
Um die kopierten Originalseiten zu speichern, wird vom Snapshot mindestens eine Datei mit geringer Dichte verwendet. Ursprünglich ist eine Datei mit geringer Dichte im Wesentlichen eine leere Datei, die keine Benutzerdaten enthält und für die noch kein Speicherplatz für Benutzerdaten auf einem Speichermedium zugeordnet worden ist. Je mehr Seiten in der Quelldatenbank aktualisiert werden, desto größer wird die Datei. Wenn ein Snapshot erstellt wird, verbraucht die Datei mit geringer Dichte nur wenig Speicherplatz. Bei nachfolgenden Aktualisierungen der Datenbank kann eine Datei mit geringer Dichte allerdings sehr groß werden. Weitere Informationen zu Dateien mit geringer Dichte finden Sie unter *Grundlegendes zur Größe von Dateien mit geringer Dichte* in *Datenbankssnapshots*.

Die folgende Abbildung veranschaulicht einen Kopie-bei-Schreibvorgang. Die hellgrauen Rechtecke im Snapshotdiagramm repräsentieren potenziellen Platz in einer Datei mit gerin-

ger Dichte, der bis jetzt noch nicht zugeordnet wurde. Beim Empfang der ersten Aktualisierung einer Seite in der Quelldatenbank wird von Database Engine (Datenbankmodul) ein Schreibvorgang auf der Datei ausgeführt und vom Betriebssystem Platz in den Dateien mit geringer Dichte des Snapshots zugeordnet und die Originalseiten dorthin kopiert. Von Database Engine (Datenbankmodul) wird die Seite dann in der Quelldatenbank aktualisiert. Die folgende Abbildung veranschaulicht einen solchen Kopie-bei-Schreibvorgang.



10 % kopiert



Schlüssel

- Nicht zugeordnet
- Ursprüngliche Seite
- Aktualisierte Seite
- Kopie bei Schreibvorgang

### Lesevorgänge auf einem Datenbankssnapshot

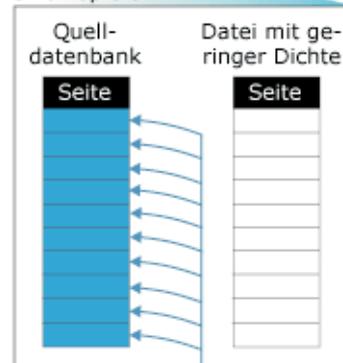
Für den Benutzer scheint sich ein Datenbankssnapshot niemals zu ändern, weil von Lesevorgängen auf einem Datenbankssnapshot immer auf die Originaldatenseiten zugegriffen wird, unabhängig von deren Speicherort.

Wenn die Seite noch nicht auf der Quelldatenbank aktualisiert wurde, wird von einem Lesevorgang auf dem Snapshot die Originalseite von der Quelldatenbank gelesen. Die folgende Abbildung zeigt einen Lesevorgang auf einem neu erstellten Snapshot, dessen Datei mit geringer Dichte dementsprechend keine Seiten enthält. Von diesem Lesevorgang wird nur von der Quelldatenbank gelesen.

Nachdem eine Seite aktualisiert worden ist, wird von einem Lesevorgang weiterhin auf die Originalseite zugegriffen, die sich dann in einer Datei mit geringer Dichte befindet. Die folgende Abbildung veranschaulicht einen Lesevorgang auf dem Snapshot, von dem auf eine Seite zugegriffen wird, nachdem sie in der Quelldatenbank aktualisiert worden ist. Vom Le-



0 % kopiert



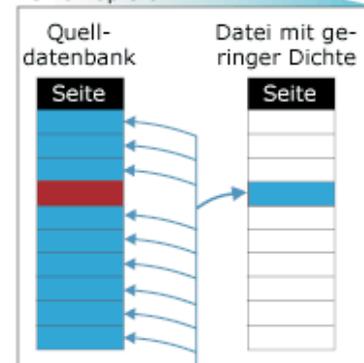
Schlüssel

- Nicht zugeordnet
- Ursprüngliche Seite
- Aktualisierte Seite

sevorgang wird die Originalseite von der Datei mit geringer Dichte vom Snapshot gelesen.



10 % kopiert



Schlüssel

- Nicht zugeordnet
- Ursprüngliche Seite
- Aktualisierte Seite

```

/**/ Objekt: Database
[AdventureWorks_Snapshot] Skriptdatum:
11/20/2006 13:59:51 ***/
CREATE DATABASE [AdventureWorks_Snapshot] ON
( NAME = N'AdventureWorks_Data', FILENAME =
N'C:\Programme\Microsoft SQL
Server\MSSQL.1\MSSQL\Data\AW_1200.ss' ) AS
SNAPSHOT OF [AdventureWorks]
GO
RESTORE DATABASE [AdventureWorks] FROM
DATABASE_SNAPSHOT = 'AdventureWorks_Snapshot'
    
```

11 Überwachung

11.1 Ausführungspläne

Öffnen Sie ein Transact-SQL-Skript, das die Abfragen enthält, die analysiert werden sollen, oder geben Sie es im Management Studio-Abfrage-Editor ein. Nachdem das Skript in den Management Studio-Abfrage-Editor geladen wurde, können Sie über die Schaltflächen "Geschätzten Ausführungsplan anzeigen" oder "Tatsächlichen Ausführungsplan einschließen" auf der Symbolleiste des Abfrage-Editors entweder einen geschätzten oder den tatsächlichen Ausführungsplan aufrufen. Wenn Sie auf "Geschätzten Ausführungsplan anzeigen" klicken, wird das Skript vom Parser analysiert und anschließend ein geschätzter Ausführungsplan generiert. Wenn Sie auf "Tatsächlichen Ausführungsplan einschließen" klicken, müssen Sie das Skript erst ausführen, bevor der Ausführungsplan generiert wird. Nachdem das Skript analysiert oder ausgeführt wurde, können Sie auf die Registerkarte "Ausführungsplan" klicken, um sich eine grafische Darstellung der Ausführungsplanausgabe anzusehen

Ausführungspläne können helfen, Flaschenhalse bei der Abarbeitung von SQL-Anweisungen festzustellen. So ist es möglich, fehlende Indizes oder problematische Indizes zu erkennen.

Beispiel

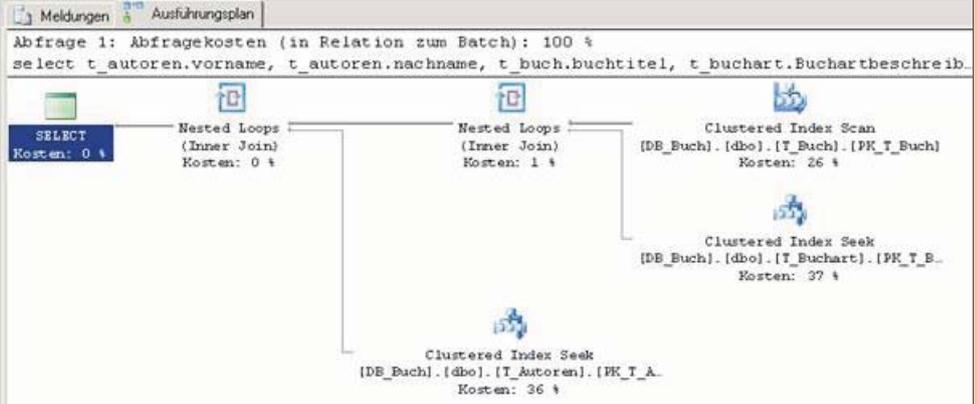
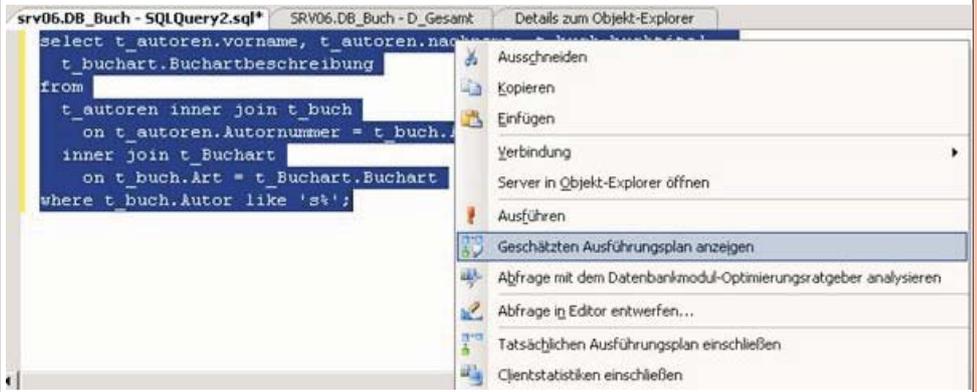
```
select t_autoren.vorname, t_autoren.nachname,
t_buch.buchtitel,
t_buchart.Buchartbeschreibung
from
t_autoren inner join t_buch
on t_autoren.Autornummer = t_buch.Autor
inner join t_Buchart
on t_buch.Art = t_Buchart.Buchart
where t_buch.Autor like 's%';
```

Erklärungen der Symbole (Auswahl)

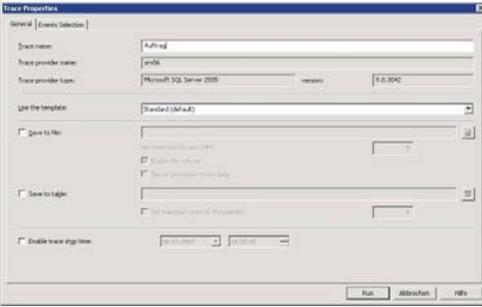
-  Table Scan: Hier wird die gesamte Tabelle durchsucht, ein Vorgang, der geringe Performance bietet. Es ist zu prüfen, ob nicht durch Erzeugen von Indizes die Leistung der Abfrage massiv verbessert werden kann.
-  Clustered Index Scan: Sehr schneller Vorgang
-  Clustered Index Seek: Sehr schneller Vorgang
-  Nonclustered Index Scan: schneller Vorgang
-  Nonclustered Index Seek: schneller Vorgang

Der Nested Loops-Operator führt die logischen Operationen Inner Join, Left Outer Join und andere Joins aus. Nested Loops-Verknüpfungen führen für jede Zeile der äußeren Tabelle eine Suche in der inneren Tabelle aus, in der Regel mithilfe eines Indexes. Microsoft SQL Server legt auf der Grundlage der geschätzten Kosten fest, ob die äußere Eingabe sortiert werden soll, um die Treffsicherheit der Suchvorgänge auf dem Index über die innere Eingabe zu verbessern.

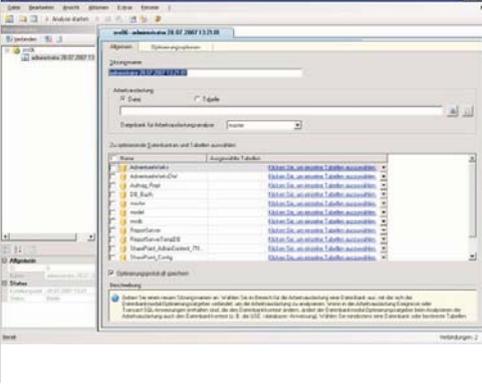
SELECT-Anweisung: generiert Resultset, üblicherweise das Stammelement jedes Ausführungsplans



11.2 SQL Server Profiler



11.3 Datenbankmodul-Optimierungsratgeber



11.4 Erkennung von Deadlocks

Deadlocks werden von SQL Server automatisch ausgelöst, "normale" Locks (etwa wenn ein User eine Transaction beginnt, aber nicht beendet, und dann ein 2. User denselben Datensatz bearbeiten will) müssen allerdings vom Administrator aufgelöst werden.

Die Erkennung von Deadlocks ist mit sp\_lock oder dem SQL Server Profiler möglich.

Ergebnisse von sp\_lock:

spid	dbid	objid	indid	Type	Resource	Mode	Status
52	4	0	0	DB		S	GRANT
53	4	0	0	DB		S	GRANT
54	4	0	0	DB		S	GRANT
55	11	0	0	DB		S	GRANT
55	11	85575343	0	TAB		IS	GRANT
57	11	2041058307	0	PAG	1:105	IX	GRANT
57	11	2041058307	0	TAB		IX	GRANT
57	11	0	0	DB		S	GRANT
57	11	2041058307	0	RID	1:105:0	X	GRANT
58	11	2041058307	0	RID	1:105:0	U	WAIT
58	11	0	0	DB		S	GRANT
58	11	2041058307	0	PAG	1:105	IU	GRANT
58	11	2041058307	0	TAB		IX	GRANT

- Lock Type
- DB = Database
- FILE = File
- IDX = Index
- PAG = PAGE
- KEY = Key
- TAB = Table
- EXT = Extent
- RID = Row identifier

SQL Server uses these resource lock modes.

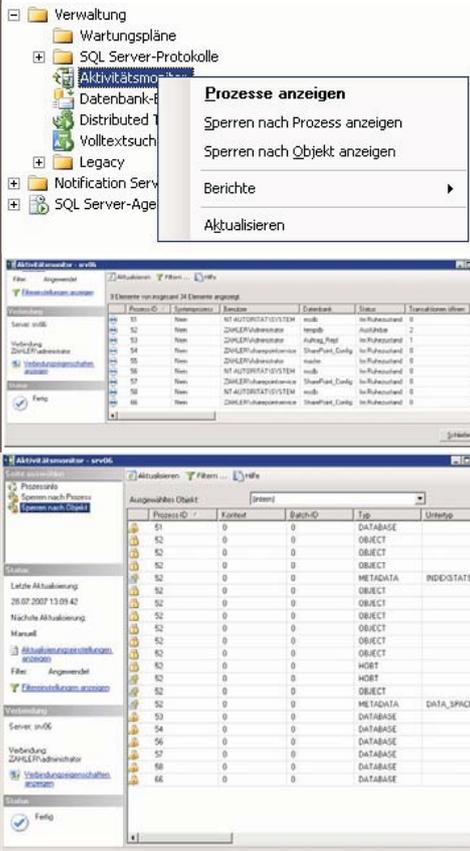
Lock mode	Description
Shared (S)	Used for operations that do not change or update data (read-only operations), such as a SELECT statement.
Update (U)	Used on resources that can be updated. Prevents a common form of deadlock that occurs when multiple sessions are reading, locking, and potentially updating resources later.
Exclusive (X)	Used for data-modification operations, such as INSERT, UPDATE, or DELETE. Ensures that multiple updates cannot be made to the same resource at the same time.

**Intent** Used to establish a lock hierarchy. The types of intent locks are: intent shared (IS), intent exclusive (IX), and shared with intent exclusive (SIX).

**Schema** Used when an operation dependent on the schema of a table is executing. The types of schema locks are: schema modification (Sch-M) and schema stability (Sch-S).

**Bulk Update (BU)** Used when bulk-copying data into a table and the TABLOCK hint is specified.

**11.5 Aktivitätsmonitor**



**13 Datenbankspiegelung**

Auf beiden Servern: Spiegelung ist standardmäßig deaktiviert; daher muss der SQL-Server-Dienst mit Ablaufverfolgungsflag 1400 neu gestartet werden.

```
net stop sqlserveragent
net stop mssqlserver
net start mssqlserver /T1400
net start sqlserveragent
```

Fehlermeldung, wenn Flag 1400 nicht gesetzt wurde:

Meldung 1498, Ebene 16, Status 2, Zeile 5 Datenbankspiegelung ist standardmäßig deaktiviert. Datenbankspiegelung ist ausschließlich zu Evaluierungszwecken erhältlich und darf nicht in Produktionsumgebungen verwendet werden. Um Datenbankspiegelung zu Evaluierungszwecken zu aktivieren, verwenden Sie Ablaufverfolgungsflag 1400 beim Start. Weitere Informationen über Ablaufverfolgungsflags und Startoptionen finden Sie in der SQL Server-Onlinedokumentation.

**Voraussetzung**

- Service-Konten beider SQL-Server müssen Mitglied beider lokalen Administrator-Gruppen sein!
- Wiederherstellungsmodell der zu spiegelnden Datenbanken muss "Full" sein!

```
ALTER DATABASE AdventureWorks
SET RECOVERY FULL;
```

**Auf primärem SQL Server**

Erstellen eines Endpunkts auf der Prinzipalinstanz und Festlegen des Wiederherstellungsmodells

```
CREATE ENDPOINT endpoint_mirroring
STATE = STARTED
AS TCP ( LISTENER_PORT = 5022 )
FOR DATABASE_MIRRORING (ROLE=PARTNER);
GO
```

**Auf Spiegel-SQL Server**

Erstellen eines Endpunkts auf der Spiegelinstanz

```
CREATE ENDPOINT endpoint_mirroring
STATE = STARTED
AS TCP ( LISTENER_PORT = 5023 )
FOR DATABASE_MIRRORING (ROLE=PARTNER);
GO
```

**Auf Zeugen-SQL-Server**

Erstellen eines Endpunkts auf der Zeugeninstanz

```
CREATE ENDPOINT endpoint_mirroring
STATE = STARTED
AS TCP ( LISTENER_PORT = 5024 )
FOR DATABASE_MIRRORING (ROLE=WITNESS);
GO
```

**Auf primärem SQL Server**

Sichern der Prinzipaldatenbank

```
BACKUP DATABASE AdventureWorks
TO DISK='C:\MirrorBackup\AW_Backup.BAK'
WITH INIT;
GO
```

21008 Seiten wurden für die 'AdventureWorks'-Datenbank, Datei 'AdventureWorks\_Data' für Datei 1, verarbeitet. 1 Seiten wurden für die 'AdventureWorks'-Datenbank, Datei 'AdventureWorks\_Log' für Datei 1, verarbeitet. BACKUP DATABASE hat erfolgreich 21009 Seiten in 14.379 Sekunden verarbeitet (11.969 MB/s).

Kopieren Sie den Ordner MirrorBackup auf den Spiegelserver.

**Auf Spiegel-SQL Server**

Wiederherstellen der Spiegeldatenbank

```
RESTORE DATABASE AdventureWorks
FROM DISK='C:\MirrorBackup\AW_Backup.BAK'
WITH NORECOVERY;
```

```
MOVE 'AdventureWorks_Data' TO
'C:\Programme\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\AdventureWorks_Data.m
df',
MOVE 'AdventureWorks_Log' TO
'C:\Programme\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\AdventureWorks_Log.1d
f';
GO
```



21008 Seiten wurden für die 'AdventureWorks'-Datenbank, Datei 'AdventureWorks\_Data' für Datei 1, verarbeitet. 2 Seiten wurden für die 'AdventureWorks'-Datenbank, Datei 'AdventureWorks\_Log' für Datei 1, verarbeitet. RESTORE DATABASE hat erfolgreich 21010 Seiten in 10.426 Sekunden verarbeitet (16.507 MB/s). Festlegen des Prinzipalserver

```
ALTER DATABASE AdventureWorks
SET PARTNER =
'TCP://MIAMI:5022'
GO
```

Festlegen des Spiegelserver auf dem Prinzipalserver

```
ALTER DATABASE AdventureWorks
SET PARTNER = 'TCP://sq104:5023'
GO
```

Festlegen des Zeugen auf dem Prinzipalserver

```
ALTER DATABASE AdventureWorks
SET WITNESS =
'TCP://MIAMI:5024'
GO
```

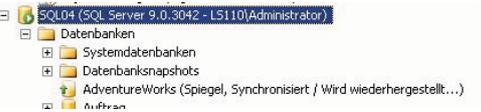
**Ergebnis**



Ausführen eines manuellen Failovers

```
ALTER DATABASE AdventureWorks
SET PARTNER FAILOVER
GO
```

Beenden des Spiegelns



```
ALTER DATABASE AdventureWorks
SET PARTNER OFF
```

Löschen des Endpunkts auf dem Spiegelserver

```
DROP ENDPOINT endpoint_mirroring
```

Löschen der Datenbank auf dem Spiegelserver

```
DROP DATABASE AdventureWorks
```

Löschen des Endpunkts auf dem Zeugenserver

```
DROP ENDPOINT endpoint_mirroring
```

Löschen des Endpunkts am Prinzipalserver

```
DROP ENDPOINT endpoint_mirroring
```

**13.1 Datenbankspiegelungsmonitor**



**12 Hochverfügbarkeitstechnologien im Überblick**

Feature	Hot Standby		Warm Standby			Cold Standby	
	Database Mirroring	Failover Clustering	Peer-to-Peer Replication	Transaktionale Replikation	Log Shipping	Backup / Restore	Detach / Copy / Attach
Datenverlust	Keiner (optional)	Keiner	Möglich	Möglich	Möglich	Möglich	Wahrscheinlich
Failover automatisch	Ja	Ja	Optional	Nein	Nein	Nein	Nein
Transparent für Client	Ja, Auto-Redirect	Ja, Recovery auf selbe IP	Optional	Nein, NLR nicht	Nein, NLR nicht	Nein	Nein
Ausfallzeit	Sekunden	20 Sec + DB Recovery	keine	Sekunden	Sekunden + DB Recovery	Erkennen, Reattach, Manueller Failover	Erkennen, Attach, Manueller Failover
Standby Leszugriff	Über Snapshot	nein	Deverhaft verfügbar	Deverhaft verfügbar	Manchmal verfügbar	Nein	Nein
Ebene minimal	Datenbank	Instanz	Tabelle oder Sicht	Tabelle oder Sicht	Datenbank	Datenbank	Datenbank
Behandelt Plattenfehler	Ja	Nein, RAID erforderlich	Ja	Ja	Ja	Ja	Ja
Spezielle Hardware	Nein, zweites System	Spezielle Hardware aus Cluster HCL	Nein, zweites System	Nein, zweites System	Nein, zweites System	Nein, zweites System	Nein, zweites System
Komplexität	Mittel	Hoch	Hoch	Hoch	Hoch	Mittel	Mittel

**Auf primärem SQL Server**

Wiederherstellen der Spiegeldatenbank

```
RESTORE DATABASE AdventureWorks
FROM DISK='C:\MirrorBackup\AW_Backup.BAK'
WITH NORECOVERY;
```

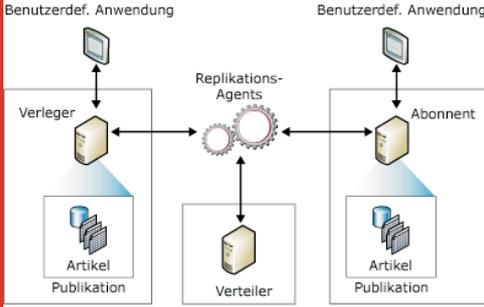
14 Replikation

Bei der Replikation werden eigenständige Programme verwendet, die als Agents bezeichnet werden. Standardmäßig werden Replikations-Agents als unter SQL Server-Agent geplante Aufträge durchgeführt.

14.1 Konzept

Bei der Replikation unterscheidet man:

- **Verleger (publisher):** Dieser veröffentlicht Daten (die man in diesem Fall „Artikel“ nennt) in einer sogenannten Publikation.
- **Verteiler (distributor):** Dieser kümmert sich um die richtige Verteilung der Daten.
- **Abonnent (subscriber):** Dieser SQL-Server dient als Replikationsziel und empfängt die abonnierten „Artikel“.



Der Einfachheit halber können Verleger und Verteiler auf demselben SQL Server eingerichtet werden.

Man unterscheidet:

- **Transaktionsreplikation:** Nach einem Erst-Snapshot werden regelmäßig Transaktionsprotokollabschnitte zum Abonnenten-Server gesendet. Mit geringer Zeitverzögerung steht daher ein (nicht änderbares) Replikat zur Verfügung, da aber sehr wohl als Redundanz-Server verwendet werden kann.
- **Mergereplikation:** Nach erfolgreicher Replikation bestehen zwei schreibbare Kopien der Datenbank. Änderungen können an beiden Replikaten durchgeführt werden und werden automatisch synchronisiert.
- **Snapshotreplikation:** In regelmäßigen Abständen wird ein Snapshot zum Zielsystem gesendet. Die Zeitverzögerung ist größer als bei der Transaktionsreplikation, das Replikat ist standardmäßig schreibgeschützt.

14.2 Transaktionsreplikation

Beteiligte Agents:

- Snapshot-Agent
- Protokolllese-Agent

**Schritt 1: Neue Transaktionspublikation erstellen**

„Transaktionspublikation mit aktualisierbaren Abonnements“ muss genommen werden, wenn eine IDENTITY-Spalte vorhanden ist, sonst kommt am Ende folgende Fehlermeldung:

**Assistent für neue Publikation**

**Publikation wird erstellt**  
Klicken Sie auf 'Beenden', um den Vorgang zu unterbrechen.

<b>Erfolg</b>	3 Gesamt	0 Fehler
	3 Erfolg	0 Warnung

Details:

Aktion	Status	Meldung
Publikation 'Auftrag_tKunden_Transakti...	Erfolg	
1-Artikel von 1 wird hinzugefügt	Erfolg	
Snapshot-Agent wird gestartet	Erfolg	

Beenden Bejcht

Schließen

**Ergebnis:**

- Replikation
  - Lokale Publikationen
    - [Auftrag]: Auftrag\_tKunden\_Transaktion

Überprüfen Sie, ob in den Eigenschaften der Publikation als Speicherort des Snapshot ein UNC-Pfad (und kein lokaler Pfad) angegeben ist (sonst lassen sich keine Pull-Abonnements erstellen!).

**Publikationseigenschaften - Artikel-Trans**

Snapshotformat

Systemeigener Format von SQL Server - alle Abonnenten müssen Server mit SQL Server sein

Speicherort der Snapshotdaten

Daten in Standardort speichern

Daten in folgendem Ordner speichern: (Vorgl.)

Weitere Skripts ausführen

Dieses Skript vor Anwenden des Snapshots ausführen

Dieses Skript nach Anwenden des Snapshots ausführen

OK Abbrechen

**Schritt 2: Replikationsdatenbank erstellen**

**Schritt 3: Abonnent konfigurieren**

Je nachdem, wo die Agents laufen sollen, ist die Konfiguration von Push- oder Pull-Abonnements möglich.

srv06 (SQL Server 9.0.3042 - ZAHLER\administrator)

- Datenbanken
- Sicherheit
- Serverobjekte
- Replikation
  - Lokale Publikationen
  - Lokale Abonnements
- Verwaltung
- Multiplizität Services

Neue Abonnements...  
Skripts generieren...  
Verteilung konfigurieren...

**Assistent für neue Abonnements**

**Assistent für neue Abonnements**

Dieser Assistent hilft Ihnen bei folgenden Aufgaben:

- Erstellen mindestens eines Abonnements für eine Publikation
- Angeben, wo und wann die Agents ausgeführt werden sollen, die das Abonnement synchronisieren

Diese Anfangsseite nicht mehr anzeigen

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

Es stehen Ihnen keine Publikationen zum Abonnieren zur Verfügung, weil dieser Server über Publikationen verfügt, oder weil Sie über keine ausreichenden Privilegien verfügen, um auf die Publikationen zuzugreifen.

**Assistent für neue Abonnements**

**Publikation**  
Wählen Sie die Publikation aus, für die Sie mindestens ein Abonnement erstellen möchten.

Verleger: srv06

<SQL Server-Verleger suchen...>  
<Oracle-Verleger suchen...>  
srv06

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Publikation**  
Wählen Sie die Publikation aus, für die Sie mindestens ein Abonnement erstellen möchten.

Verleger: dc02

Datenbanken und Publikationen:

- Auftrag
  - Auftrag\_Artikel\_Meige
  - Auftrag\_tKunden\_Transaktion

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Speicherort des Verteilungs-Agents**  
Wählen Sie aus, wo die Verteilungs-Agents ausgeführt werden sollen.

Mit diesem Assistenten erstelle Abonnements wie folgt behandelt:

Alle Agents auf dem Verteiler 'DC02' ausführen (Pushabonnements)

Mit dieser Option wird die zentrale Verwaltung der Synchronisierung von Abonnements vereinfacht.

Jeden Agent auf seinem Abonnenten ausführen (Pullabonnements)

Mit dieser Option wird der Verarbeitungsaufwand auf dem Verteiler reduziert. Die Synchronisierung des jeweiligen Abonnements wird auf dem Abonnenten verwaltet.

Führen Sie den Assistenten mehrmals aus, wenn einige Agents auf dem Verteiler und andere Agents auf Abonnenten ausgeführt werden sollen.

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Abonnenten**  
Wählen Sie mindestens einen Abonnenten aus, und geben Sie die jeweilige Abonnementdatenbank an.

Abonnenten und Abonnementdatenbanken:

Abonnement	Abonnementdatenbank
srv06	Auftrag_Repl

Abonnent hinzufügen

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Sicherheit für den Verteilungs-Agent**  
Geben Sie das Domänen- oder Computerkonto an, unter dem der Verteilungs-Agent-Prozess beim Synchronisieren dieses Abonnements ausgeführt wird.

Unter dem folgenden Windows-Konto ausführen:

Prozesskonto: zahler\administrator  
Beispiel: Domäne\Konto

Kenntwort:

Kenntwort bestätigen:

Unter dem SQL Server-Agent-Dienstkonto ausführen (das ist keine empfohlene bewährte Sicherheitsmethode)

Verbindung mit dem Verteiler herstellen

Identität des Prozesskontos annehmen

SQL Server-Anmeldung verwenden

Die Verbindung mit dem Server, auf dem der Agent ausgeführt wird, muss die Identität des Prozesskontos annehmen. Das Prozesskonto muss ein Element der Publikationszugriffsliste sein.

Verbindung mit dem Abonnenten herstellen

Identität des Prozesskontos annehmen

Folgende SQL Server-Anmeldung verwenden:

Anmeldename:   
Kenntwort:   
Kenntwort bestätigen:

Der Anmeldename zum Verbinden mit dem Abonnenten muss ein Datenbankbesitzer der Abonnementdatenbank sein.

OK Abbrechen Hilfe

**Assistent für neue Abonnements**

**Sicherheit für den Verteilungs-Agent**  
Geben Sie das Prozesskonto und die Verbindungsoptionen für jeden Verteilungs-Agent an.

Abonnenteigenschaften:

Agent für Abonnent	Verbindung mit Verteiler	Verbindung mit Abonnent
srv06	Identität von 'zahler\ad...	Identität von 'zahler\ad...

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Synchronisierungszeitplan**  
Geben Sie den Synchronisierungszeitplan für jeden Agent an.

Agentzeitplan:

Abonnement	Agentspeicherort	Agentzeitplan
srv06	Verteiler	Fortlaufend ausführen

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Abonnements initialisieren**  
Geben Sie an, ob jedes Abonnement mit einem Snapshot der Publikationsdaten und des Publikationsschemas initialisiert werden soll.

Abonnenteigenschaften:

Abonnement	Initialisieren	Initialisierungszeitpunkt
srv06	<input checked="" type="checkbox"/>	Sofort

Eine Abonnementdatenbank muss mit einem Snapshot der Publikationsdaten und des Publikationsschemas initialisiert werden, es sei denn, sie wurde bereits speziell für das Abonnement vorbereitet.

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Aktionen des Assistenten**  
Wählen Sie die Aktion aus, die nach dem Klicken auf 'Fertig stellen' ausgeführt werden soll.

Am Ende des Assistenten:

Abonnements erstellen

Skriptdatei mit Schritten zur Abonnentenerstellung generieren

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

**Assistenten abschließen**  
Überprüfen Sie die im Assistenten vorgenommene Auswahl, und klicken Sie dann auf 'Fertig stellen'.

Klicken Sie auf 'Fertig stellen', um die folgenden Aktionen auszuführen:

- Erstellen Sie Abonnements
- Erstellen ein Abonnement für die Auftrag\_tKunden\_Transaktion-Publikation von Verleger 'DC02'.
- Erstelle Abonnements auf den folgenden Abonnenten:
  - srv06
    - Abonnementdatenbank: Auftrag\_Repl
    - Agentspeicherort: Verteiler
    - Agentzeitplan: Fortlaufend ausführen
    - Agentspeicherort: zahler\administrator
    - Verbindung mit Verteiler: Identität von 'zahler\administrator' annehmen
    - Verbindung mit Abonnent: Identität von 'zahler\administrator' annehmen
    - Initialisieren: Sofort

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Abonnements**

Abonnements werden erstellt...  
Klicken Sie auf 'Beenden', um den Vorgang zu unterbrechen.

<b>Erfolg</b>	1 Gesamt	0 Fehler
	1 Erfolg	0 Warnung

Details:

Aktion	Status	Meldung
Abonnement für 'sv06' wird erstellt	Erfolg	

Beenden Beicht

Schließen

**Assistent für neue Publikation**

**Verteiler**  
Verwenden Sie den Server als seinen eigenen Verteiler, oder wählen Sie einen anderen Server als Verteiler aus.

Der Verteiler ist der Server, auf dem die beim Synchronisieren verwendeten Replikationsinformationen gespeichert werden.

'DC02' als seinen eigenen Verteiler verwenden: SQL Server erstellt eine Verteilungsdatenbank und ein Protokoll

Folgenden Server als Verteiler verwenden (Hinweis: Der ausgewählte Server muss bereits als Verteiler konfiguriert sein):

Hinzufügen

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Publikation**

**Artikel**  
Wählen Sie Tabellen und andere Objekte aus, die als Artikel veröffentlicht werden sollen. Wählen Sie Spalten aus, um Tabellen zu filtern.

Zu veröffentliche Objekte:

- Tabellen
- Artikel (dbo)

Artikeleigenschaften:

- Die durch Hervorhebung markierte Tabelle ist nur downloadbar
- Nur in der Liste aktivierte Objekte anzeigen

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Ergebnis:**

- Replikation
  - Lokale Publikationen
    - Lokale Abonnements
      - [Auftrag\_Repl] - [DC02].[Auftrag]: Auftrag\_tKunden\_Transaktion

### 14.3 Mergereplikation

Beteiligte Agents:

- Snapshot-Agent
- Merge-Agent

### Schritt 1: Freigegebenen Ordner für Snapshot einrichten

Adresse: C:\Programme\Microsoft SQL Server\MSSQL.1\MSSQL\repldata

Ordner:

- Internet Explorer
- Microsoft Analysis Services
- Microsoft Office
- Microsoft SQL Server
  - 80
  - 90
  - MSSQL.1
    - MSSQL
      - Backup
      - Binn
      - Data
      - FTData
      - Install
      - JOBS
      - LOG
      - repl
        - repldata
  - MSSQL.2
  - Microsoft Visual S...
  - Microsoft Visual S...

**Assistent für neue Publikation**

**Snapshotordner**  
Geben Sie das Stammverzeichnis an, in dem Snapshots gespeichert werden sollen.

Um Verteilungs- und Merge-Agents auf Abonnenten den Zugriff auf die Snapshots ihrer Publikationen zu ermöglichen, müssen Sie mithilfe eines Netzwerkpfades auf den Snapshotordner verweisen.

Snapshotordner:  
C:\Programme\Microsoft SQL Server\MSSQL.1\MSSQL\ReplData

Publ abonnements, die auf dem Abonnenten erstellt wurden, werden von diesem Snapshotordner zugedrehter Laufwerkbuchstabe. Verweisen Sie mit einem

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Publikation**

**Artikelprobleme**  
Folgende Probleme erfordern möglicherweise Änderungen an der Anwendung, damit diese weiterhin funktionsfähig ist.

Probleme:  
Uniqueidentifizier-Spalten werden zu Tabellen hinzugefügt.

Beschreibung:  
Jeder Mergeartikel muss eine uniqueidentifizier-Spalte mit einem eindeutigen Index und der ROWGUIDCOL-Eigenschaft enthalten. Wenn der erste Snapshot generiert wird, fügt SQL Server eine uniqueidentifizier-Spalte jeder veröffentlichten Tabelle hinzu, die keine solche Spalte hat.  
Das Hinzufügen einer neuen Spalte hat folgende Auswirkungen:  
> Fehler bei INSERT-Anweisungen ohne Spaltenlisten  
> Die Tabelle wird vergrößert  
> Die Zeit zum Generieren des ersten Snapshots wird länger  
Microsoft SQL Server fügt eine uniqueidentifizier-Spalte mit einem eindeutigen Index und der ROWGUIDCOL-Eigenschaft jeder der folgenden Tabellen hinzu.

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Publikation**

**Publikationsdatenbank**  
Wählen Sie die Datenbank aus, die die Daten oder Objekte enthält, die Sie veröffentlichen möchten.

Datenbanken:  
Auftrag  
SMS\_NDE

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Publikation**

**Tabellenzeilen filtern**  
Fügen Sie Filter zu veröffentlichten Tabellen hinzu. Erweitern Sie die Filter durch Hinzufügen von Verknüpfungen auf andere Tabellen.

Gefilterte Tabellen:  
Klicken Sie auf 'Weiter', wenn die Daten in der Publikation nicht gefiltert werden müssen.  
Klicken Sie auf 'Hinzufügen', um mit dem Filtern der Publikation zu beginnen.

Hinzufügen Bearbeiten... Löschen...

Filter:

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

### Schritt 2: Konfiguration einer Merge-Publikation

Replikation

- Lokale Publikationen
  - Lokale Abonnements:
    - Neue Publikation...
    - Neue Oracle-Publikation...
    - Replikationsmonitor starten
    - Skripts generieren...
    - Verteilung konfigurieren...
    - Aktualisieren
- Verwaltung
- Notification Services
- SQL Server-Agent

**Assistent für neue Publikation**

**Publikationstyp**  
Wählen Sie den Publikationstyp aus, der die Anforderungen Ihrer Anwendung am besten erfüllt.

Publikationstyp:  
 Snapshotpublikation  
 Transaktionspublikation  
 Transaktionspublikation mit aktualisierbaren Abonnements  
 Mergereplikation

Beschreibungen von Publikationstypen:  
einen Anfangssnapshot der veröffentlichten Daten empfangen haben. Vom Abonnenten stammende Transaktionen werden auf dem Verteiler angewendet.

**Mergereplikation:**  
Der Verteiler und die Abonnenten können die veröffentlichten Daten unabhängig voneinander aktualisieren, nachdem die Abonnenten einen Anfangssnapshot der veröffentlichten Daten empfangen haben. Änderungen werden regelmäßig zusammengeführt. Microsoft SQL Server Mobile Edition kann nur Mergereplikationen abonnieren.

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Publikation**

**Snapshot-Agent**  
Geben Sie an, wann der Snapshot-Agent ausgeführt werden soll.

Abonnements werden mit einem von Snapshot-Agent erstellen Snapshot des Publikationsschemas und der Publikationsdaten initialisiert.

- Snapshot sofort erstellen
- Ausführung des Snapshot-Agents zu folgenden Zeitpunkten planen:  
Alle 14 Tag(e) um 00:05:00. Zeitplan wird ab 27.07.2007 verwendet.

Ändern...

Wenn Sie die Snapshot-Eigenschaften ändern möchten, starten Sie den Snapshot-Agent erst nach dem Ändern der Eigenschaften im Eigenschaftendialogfeld für Publikationen.

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

Achtung: Der Pfad muss auf einen freigegebenen Netzwerkpfad in UNC-Syntax geändert werden.

**Assistent für neue Publikation**

Dieser Assistent hilft Ihnen bei folgenden Aufgaben:

- Auswählen der zu replizierenden Daten und Datenbankobjekte
- Filtern der veröffentlichten Daten, sodass Abonnenten nur die benötigten Daten empfangen

Nachdem Sie den Assistenten fertig gestellt haben, können Sie Abonnements für die neue Publikation auf anderen Servern erstellen.

Diese Anfangsseite nicht mehr anzeigen

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

**Assistent für neue Publikation**

**Abonnementstypen**  
Geben Sie die SQL Server Versionen an, die von Abonnenten dieser Publikation verwendet werden.

Die Assistent konfiguriert die Publikation ausschließlich mit Funktionalität, die von allen angegebenen Abonnementstypen unterstützt wird.

- SQL Server 2005
- SQL Server 2005 Mobile Edition  
Snapshotdateien müssen im Zeichenformat vorliegen.
- SQL Server 2000  
Logische Datenätze, Replikation von DDL-Änderungen und bestimmte Optimierungen für gefilterte Publikationen werden nicht unterstützt.
- SQL Server für Windows CE  
Die Einschränkungen für SQL Server 2000 gelten, außerdem müssen Snapshotdateien im Zeichenformat vorliegen.

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen

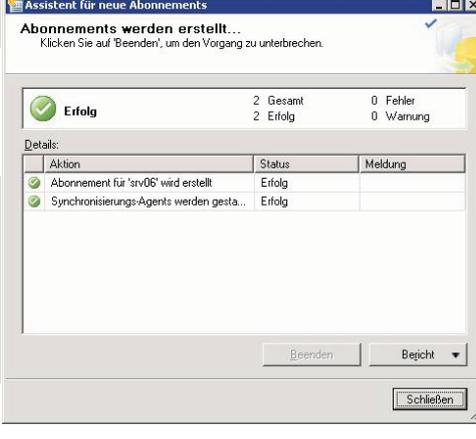
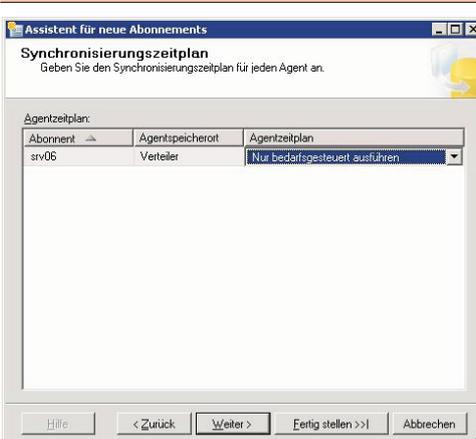
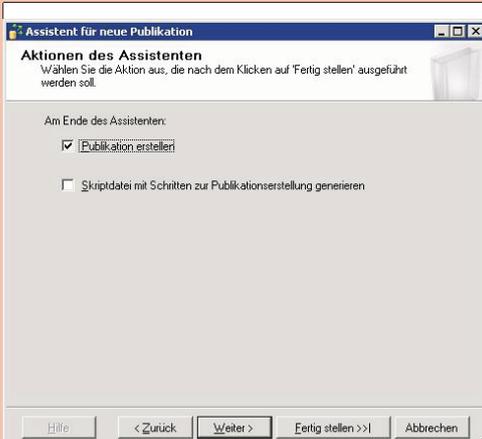
**Assistent für neue Publikation**

**Agentensicherheit**  
Geben Sie für jeden Agent das Konto, unter dem er ausgeführt wird, und seine Verbindungsinstellungen an.

Snapshot-Agent  
publiz/administrator

Sicherheitseinstellungen

Hilfe < Zurück Weiter > Fertig stellen >> Abbrechen



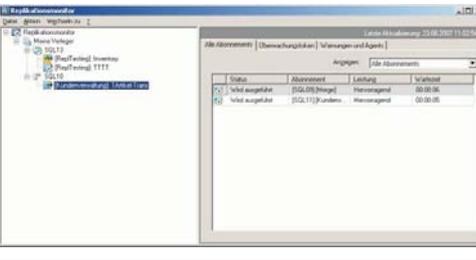
**Schritt 3: Replikationsdatenbank anlegen**

**Schritt 4: Abonnement erzeugen**

Dieser Vorgang läuft genauso ab wie bei der Transaktionsreplikation. Natürlich müssen hier statt den Sicherheitseinstellungen für den Verteilungs-Agent jene für den Merge-Agent konfiguriert werden.



**14.4 Replikationsmonitor**



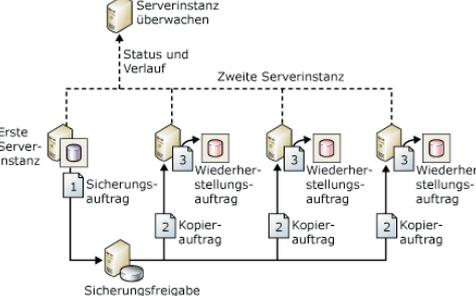
**15 Transaction Log Shipping (Protokollversand)**

Auch diese Methode stellt mit Hilfe der Backup- und Restore-Technologien von SQL Server 2005 eine Möglichkeit dar, eine betriebsbereite Standbylösung zu erstellen. Dabei werden regelmäßige Transaktionsprotokolle vom primären Server auf einen sekundären Standbyserver gesendet. Dadurch bleibt die sekundäre Datenbank nahezu synchron mit der primären Datenbank. Ein optionaler dritter Server, der als Überwachungsserver (witness server) bezeichnet wird, zeichnet den Verlauf und Status von Sicherungs- und Wiederherstellungsvorgängen sowie die Aktivitäten der überwachten Server auf.

Die Technologie des Protokollversands funktioniert unter SQL Server 2000 nur mit Hilfe von SQL-Skripts; sie wurde nun gründlich überarbeitet und bietet nun „grafischen Komfort“.

**Kernpunkte**

- Es kann nur die gesamte Datenbank gesichert werden.
- Voraussetzung: Wiederherstellungsmodell „Full“ oder „Bulk-Logged“
- Kein automatisches Failover auf den sekundären Server
- Erstellen Sie keine zusätzlichen Transaktionsprotokollsicherungen für eine Datenbank, die Protokollversand verwendet (Konflikte!)

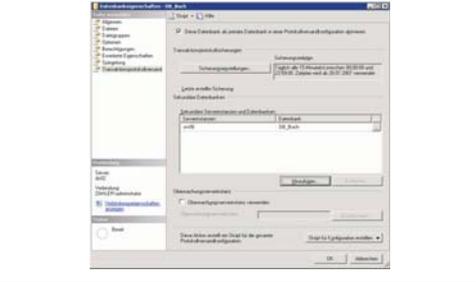
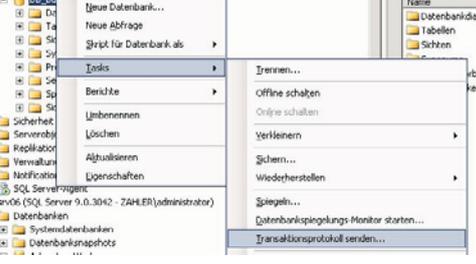


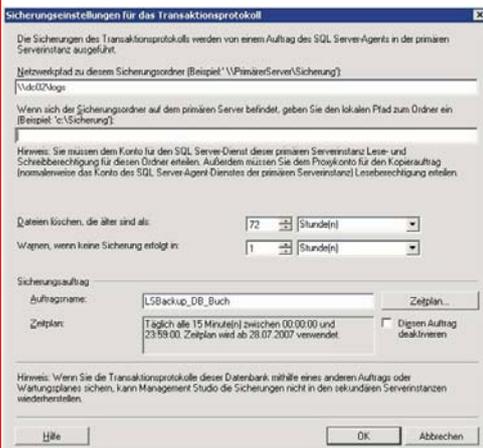
**Schritt 1: Legen Sie auf dem Master-SQL-Server einen Ordner an und geben ihn frei. Freigabeberechtigung: Jeder – Vollzugriff**

- Inetpub
- Logs
- MSOCache

**Schritt 2: Legen Sie auf dem Ziel-SQL-Server ebenfalls einen Ordner mit dem gleichen Namen an und geben auch ihn mit Jeder – Vollzugriff frei.**

**Schritt 3: Wählen Sie am primären Server aus dem Kontextmenü der Datenbank den Befehl [Tasks] – [Transaktionsprotokoll senden]**





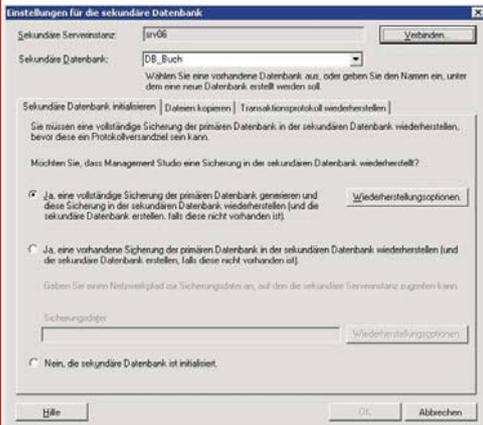
Ergebnis: Am Masterserver wurde ein Log-Backup erstellt:



Am Zielserver wurde die Datenbank erzeugt und im Status „Wird wiederhergestellt...“ hinterlassen:



**Ändern von Serverrollen:** Wenn es nun zu einem Ausfall des primären Servers kommt, kann der sekundäre Server die Rolle des primären übernehmen. Für die Vorbereitung folgender „Rollentäusche“ müssen ein Mal folgende Schritte durchgeführt werden:



**Schritt 4: Deaktivieren des ursprünglichen Protokollversand-Sicherungsauftrags auf dem primären Server**



**Schritt 5: Deaktivieren der Kopier- und Wiederherstellungsaufträge auf dem sekundären Server**



**Schritt 6: Durchführen eines manuellen Failover**

(a) Am primären Server: Sichern des letzten Transaktionslogs mit Option **NO\_RECOVERY**

```
USE master
BACKUP LOG DB_Buch TO
DISK='C:\Logs\logtail.bak'
WITH NORECOVERY
```

3 Seiten wurden für die 'DB\_Buch'-Datenbank, Datei 'DB\_Buch\_Log' für Datei 1, verarbeitet. BACKUP LOG hat erfolgreich 3 Seiten in 0.049 Sekunden verarbeitet (0.376 MB/s).

(b) Kopieren aller Transaktionsprotokollsicherungen aus der Sicherungsfreigabe des primären Servers in den Zielordner des sekundären Servers

(c) Am sekundären Server: Wiederherstellen aller nicht verarbeiteten Transaktionslogs mit Option **NORECOVERY**

```
RESTORE LOG DB_Buch FROM
DISK='C:\Logs\DB_Buch_20070728074503.trn' WITH
NORECOVERY
```

0 Seiten wurden für die 'DB\_Buch'-Datenbank, Datei 'DB\_Buch\_Data' für Datei 1, verarbeitet. 3 Seiten wurden für die 'DB\_Buch'-Datenbank, Datei 'DB\_Buch\_Log' für Datei 1, verarbeitet. RESTORE LOG hat erfolgreich 3 Seiten in 0.023 Sekunden verarbeitet (0.823 MB/s).

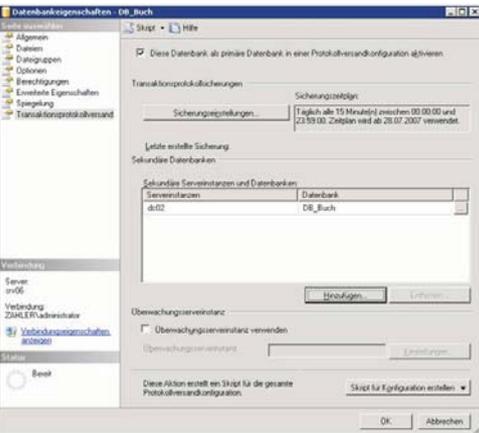
(d) Am sekundären Server: Wiederherstellen des letzten Transaktionslogs mit Option **RECOVERY**

```
RESTORE LOG DB_Buch FROM
DISK='C:\Logs\logtail.bak' WITH RECOVERY
```

0 Seiten wurden für die 'DB\_Buch'-Datenbank, Datei 'DB\_Buch\_Data' für Datei 1, verarbeitet. 3 Seiten wurden für die 'DB\_Buch'-Datenbank, Datei 'DB\_Buch\_Log' für Datei 1, verarbeitet. RESTORE LOG hat erfolgreich 3 Seiten in 0.024 Sekunden verarbeitet (0.768 MB/s).

Ergebnis: DB\_Buch ist auf dem ehemaligen sekundären – nun primären – Server online, am neuen sekundären – ehemals primären – Server ist sie im Status „Wird wiederhergestellt...“

**Schritt 7: Konfigurieren Sie nun auf dem neuen primären Server die Einstellungen für den Protokollversand wie oben, mit folgenden Unterschieden**



- Wählen Sie die ursprüngliche Datenbank als sekundäre Datenbank aus
- Im Dialogfeld „Einstellungen für die sekundäre Datenbank“ wählen Sie: „Nein, die sekundäre Datenbank ist initialisiert aus.“



Nun kann mit wenigen Schritten ein immer wiederkehrender Wechsel der Serverrollen durchgeführt werden.

**16 Tipps und Tricks**

**16.1 Verbindungsserver (Linked Server)**

Ein Verbindungsserver ermöglicht den Zugriff auf verteilte, heterogene Abfragen für OLE DB-Datenquellen. Nachdem ein Verbindungsserver mithilfe von `sp_addlinkedserver` erstellt wurde, können verteilte Abfragen für diesen Server ausgeführt werden. Wenn der Verbindungsserver als Instanz von SQL Server definiert wird, können remote gespeicherte Prozeduren ausgeführt werden.

```
use master
exec sp_addlinkedserver 'sq107','N'SQL Server'
select * from sq107.Verkauf.dbo.tKunden
```

**16.2 Umbenennen eines Servers, auf dem SQL Server 2005 als eigenständige Instanz ausgeführt wird**

**Achtung:** Es kann sein, dass sich die SQL-Instanz nicht mit umbenennen. Daher müssen folgende Schritte durchgeführt werden:

```
sp_dropserver srv2003
```

```
sp_addserver srv06,local
```

### Überprüfen

```
select @@servername
select * from sys.servers
```

### 16.3 Registrieren des SPN (Service Principal Name)

Es ist von großer Wichtigkeit, dass der MSSQL-Server-Dienst im Active Directory korrekt registriert ist.

#### Probleme

- Ein SPN ist nicht registriert: Wenn ein SPN nicht registriert ist, funktioniert die Kerberos-Authentifizierung von dem lokalen Computer aus, auf dem die Instanz von SQL Server ausgeführt wird, schlägt auf Remoteclientcomputern jedoch fehl. Das bedeutet, dass auf den SQL Server von einem anderen PC aus nicht zugegriffen werden kann.
- Ein SPN wird mehrmals registriert: Es sind verschiedene Szenarien denkbar, in denen ein Administrator die Dienstprinzipalnamen (*Service Principal Names, SPNs*) im Domänenverzeichnis mit der Auswirkung doppelt vergibt, dass die Kerberos-Authentifizierung einen Fehler erzeugt. Dabei handelt es sich z. B. um die folgenden Aktionen:
  - Vornehmen von Änderungen am Domänenkonto, unter dem die Instanz von SQL Server ausgeführt wird: Wenn `setSpn.exe` ausgeführt wird, während eine Instanz von SQL Server als ein Domänenkonto (z. B. `DOMAIN\User1`) ausgeführt wird, und anschließend das Domänenkonto, das zum Ausführen von SQL Server verwendet wird (z. B. `DOMAIN\User2`), geändert wird, führt eine erneute Ausführung von `setSPN.exe` dazu, dass der gleiche SPN in das Verzeichnis unter beiden Konten eingefügt wird.
  - Installieren mehrerer Instanzen von SQL Server, die unter verschiedenen Konten ausgeführt werden: Wenn Sie mehrere Instanzen von SQL Server installieren und dann jede dieser Instanzen unter einem anderen Konto ausführen, werden doppelte Konten im Verzeichnis unter jedem SQL Server-Dienstkonto erstellt, wenn `setSpn.exe` für die einzelnen Instanzen ausgeführt wird. Dies gilt für Instanzen, die unter einem Domänenbenutzerkonto oder dem lokalen Systemkonto ausgeführt werden.
  - Entfernen und Neuinstallieren einer Instanz von SQL Server unter einem anderen Konto: Wenn Sie SQL Server unter einem Konto installieren, die SPNs registrieren, SQL Server entfernen und unter einem anderen Konto neu installieren und dann die SPNs erneut registrieren, verfügen die einzelnen Domänenkonten über die gleichen SPNs. Dies bedeutet, dass die SPNs doppelt vorhanden sind.

Anzeigen der Registrierungen mit `setspn` (Bestandteil der Windows 2003-Support Tools):

```
C:\>setspn -L dc02
Registered ServicePrincipalNames for
CN=DC02,OU=Domain Controllers,DC=zahler,DC=at:
exchangeAB/DC02
exchangeAB/dc02.zahler.at
MSSqlSvc/dc02.zahler.at:1433
MSSqlSvc/dc02:1433

ldap/dc02.zahler.at/ForestDnsZones.zahler.at

ldap/dc02.zahler.at/DomainDnsZones.zahler.at
DNS/dc02.zahler.at
GC/dc02.zahler.at/zahler.at
HOST/dc02.zahler.at/zahler.at
HOST/dc02.zahler.at/ZAHLER

ldap/892db81a-be8b-442d-a6c7-6e481e4e4e53._msdc
s.zahler.at
ldap/dc02.zahler.at/ZAHLER
ldap/DC02
```

```
ldap/dc02.zahler.at
ldap/dc02.zahler.at/zahler.at
```

```
E3514235-4B06-11D1-AB04-00C04FC2DC02/892db81a-b
e8b-442d-a6c7-6e481e4e4e53/zahler.at
```

```
NtFrs-88f5d2bd-b646-11d2-a6d3-00c04fc9b232 /dc02
.zahler.at
HOST/DC02
HOST/dc02.zahler.at
```

#### Erstellen eines neuen SPN

```
setspn -A <spn> <Serviceaccount>
setspn -A MSSqlSvc/sq103.domain.intern:1433
domain\sqliService
```