

## Re: Transaction Log ??

**Source:** <http://www.derkeiler.com/Newsgroups/microsoft.public.sqlserver.security/2002-07/1445.html>

---

**From:** Peter Lin ([lxya@yahoo.com](mailto:lxya@yahoo.com))

**Date:** 07/23/02

From: "Peter Lin" <[lxya@yahoo.com](mailto:lxya@yahoo.com)>  
Date: Tue, 23 Jul 2002 08:26:23 +0800

Hi,

Here is the SP, we used to simplify our database restore process: 1. it will restore the database you specified if it is not there 2. it will check which log file(s) should be applied and restore them one by one.  
assumptions: every database got their respective backup folder.

If you found that it cannot cannot, please let me know. I will improve it.

```
-----  
CREATE PROC lsp_RestoreDB  
@DbName nvarchar(128)  
,@DbBackupPath nvarchar(128)  
,@LogBackupPath nvarchar(128)  
,@DbBackupFileName nvarchar(128) = "  
,@LogFileExt char(4) = '.TRN'  
,@DBFileExt char(4) = '.BAK'  
-----  
-- Proc Name: lsp_RestoreDB  
-- Date: May/16/2002  
-- Last modified: May/28/2002  
-- Author: Peter Lin  
-- Function: Auto restore full database backup and trans log  
-- Usage: EXEC lsp_RestoreDB 'DatabaseName','database backup pathP\  
-- , 'transaction log backup path\  
-- , 'full backup file name without path',  
-- , 'Log Backup file Extension'  
-- , 'Database backup file extension'  
-----  
as  
set nocount on  
/*  
** Check to see that the @DbName is valid.  
*/  
declare @returncode int  
exec @returncode = sp_validname @DbName  
if @returncode <> 0
```

```
begin
raiserror(15224,-1,15,@DbName)
return(1)
end
/*
** Restore database, if it is there then skip
*/
declare @UndoFileName nvarchar(128)
set @UndoFileName =@LogBackupPath+'undofile.udf'
declare @SqlDataPath nvarchar(128)
select @SqlDataPath =
substring([filename],0,patindex('%master%',[filename])) from
master..sysfiles where [name]='master'
declare @DbBackupFullName nvarchar(128) ,@LogicalName
nvarchar(128),@PhysicalName nvarchar(260)
declare @iFileExists int,@cmd nvarchar(1000)
declare @source nvarchar(255),@description nvarchar(255)

-- DDL SECTION

-- store full database backup filename
create table #dbk (fName nvarchar(500))

-- accept results from RESTORE FILELISTONLY
create table #FileList (
LogicalName nvarchar(128)
,PhysicalName nvarchar(260)
,Type char(1)
,FileGroupName nvarchar(128)
,[Size] numeric(20,0)
,[MaxSize] numeric(20,0)
)

-- accept results from RESTORE HEADERONLY
-- openquery(linked_server,'SET FMTONLY OFF; exec("...")') can save a lot
columns
-- however it will need a linked_server to be configured
create table #RH (BackupName nvarchar(128),BackupDescription nvarchar(255)
,BackupType smallint,ExpirationDate datetime,Compressed tinyint
,Position smallint,DeviceType tinyint,UserName nvarchar(128)
,ServerName nvarchar(128),DatabaseName nvarchar(128),DatabaseVersion int
,DatabaseCreationDate datetime,BackupSize numeric(20,0),FirstLSN
numeric(25,0)
,LastLSN numeric(25,0),CheckpointLSN numeric(25,0),DatabaseBackupLSN
numeric(25,0)
,BackupStartDate datetime,BackupFinishDate datetime,SortOrder smallint
,CodePage smallint,UnicodeLocaleId int,UnicodeComparisonStyle int
,CompatibilityLevel tinyint,SoftwareVendorId int,SoftwareVersionMajor
int
,SoftwareVersionMinor int,SoftwareVersionBuild int,MachineName
nvarchar(128)
```

microsoft.public.sqlserver.security: Re: Transaction Log ??

```
,Flags int,BindingID uniqueidentifier ,RecoveryForkID uniqueidentifier  
,Collation nvarchar(128) )
```

```
create table #dir (fName nvarchar(500))
```

```
-- Log file list
```

```
create table #LogFileLsn (LogFileName nvarchar(128),FirstLSN  
numeric(25,0),LastLSN numeric(25,0))
```

```
create table #DL (Error nvarchar(500))
```

```
-----  
if (db_id(@DbName) IS NULL) -- if database not exists
```

```
begin
```

```
-- auto select latest database backup file
```

```
if len(@DbBackupFileName) = 0
```

```
begin
```

```
set @cmd = 'dir "' + @DbBackupPath + '*' + @DBFileExt + "' /a:-d /t:w  
/o:-d'
```

```
insert #dbk EXEC master..xp_cmdshell @cmd
```

```
exec('delete #dbk where NOT (fName LIKE '%' + @DBFileExt + ''  
OR fName IS NULL')
```

```
select top 1 @DbBackupFileName =  
reverse(substring(reverse(fName),0,patindex('% %',reverse(fName)))) from  
#dbk
```

```
end
```

```
print '@DbBackupFileName=' + @DbBackupFileName
```

```
set @DbBackupFullName = @DbBackupPath+@DbBackupFileName
```

```
-- // detect existence of the db full backup file
```

```
exec master..xp_fileexist @DbBackupFullName, @iFileExists out
```

```
if @@ERROR<>0 OR @iFileExists = 0
```

```
begin
```

```
set @source = 'Restore Database Procedure'
```

```
set @description='File "' + @DbBackupFullName + "' doesn't exist'
```

```
GOTO Error_Handler
```

```
end
```

```
select @cmd = 'RESTORE FILELISTONLY from DISK = ''
```

```
+@DbBackupFullName + ''
```

```
insert #FileList EXEC sp_executesql @cmd
```

```
-- build the move argument of the RESTORE DATABASE
```

```
declare @MoveArgu nvarchar(500)
```

```
set @MoveArgu = ''
```

```
declare dcr cursor local read_only
```

```
for select LogicalName,PhysicalName from #FileList
```

```
open dcr
```

```
while 'OK' = 'OK'
```

```
begin
```

```
fetch next from dcr into @LogicalName,@PhysicalName
```

```
if @@fetch_STATUS < 0 BREAK
```

```
set @MoveArgu = @MoveArgu
```

microsoft.public.sqlserver.security: Re: Transaction Log ??

```
+ ',MOVE "' + @LogicalName + "' TO "'
+ @SqlDataPath + @DbName + '__'
+
reverse(substring(reverse(@PhysicalName),0,patindex('%\%',reverse(@PhysicalName)))) + ""
end
deallocate dcr

-- restore the database
select @cmd = 'RESTORE DATABASE ' + @DbName + ' from DISK="'
+ @DbBackupFullName
+ "' WITH STANDBY =" + @UndoFileName + ""
+ @MoveArgu
+ ',REPLACE'

EXECUTE(@cmd)
end

-- /*
-- ** Restore the log file
-- */
declare @LogFileName nvarchar(128),@LogFileFullName nvarchar(128)
set @cmd = 'dir "' + @LogBackupPath + '*' + @LogFileExt + "' /a:-d /t:w
/o:d'

insert #dir EXEC master..xp_cmdshell @cmd
EXECUTE('delete #dir where NOT (fName LIKE "' + @LogFileExt +
"' ) OR fName IS NULL')

insert into #LogFileLsn
select reverse(substring(reverse(fName),0,patindex('%
%',reverse(fName))))),0,0 from #dir
-- get logFileName, FirstLSN, LastLSN list

declare cr CURSOR LOCAL READ_ONLY
for select LogFileName from #LogFileLsn
open cr
while 'OK' = 'OK'
begin
fetch next from cr into @LogFileName
if @@fetch_STATUS < 0 BREAK
set @LogFileFullName = @LogBackupPath + @LogFileName
delete from #RH
insert #RH EXEC ('RESTORE HEADERONLY from DISK="' + @LogFileFullName + "'")
update #LogFileLsn set FirstLSN = r.FirstLSN, LastLSN = r.LastLSN
from #LogFileLsn, #RH r
where LogFileName = @LogFileName
end
deallocate cr
```

microsoft.public.sqlserver.security: Re: Transaction Log ??

```
declare @MaxLSN numeric(25,0), @MinLSN numeric(25,0)
select @MinLSN = min(FirstLSN), @MaxLSN=max(LastLSN) from #LogFileLsn
-- detect log file breakage
if @@ERROR<>0 OR ( exists(select * from #LogFileLsn t1
  where t1.FirstLSN <> @MinLSN
  -- missing: previous tranaction log
  AND NOT exists(select * from #LogFileLsn t2
    where t2.LastLSN = t1.FirstLSN)
  )
  OR exists ( select * from #LogFileLsn t3
  where t3.LastLSN <> @MaxLSN
  -- missing: subsequent tranaction log
  AND NOT exists(select * from #LogFileLsn t4
    where t4.FirstLSN = t3.LastLSN)
  ))
begin
set @source = 'Restore Database Procedure'
set @description='LOG FILES BREAKAGE DETECTED!!!'
print ' Source: ' + @source
print ' Description: ' + @description
print ' Warning: '
-- GOTO Error_Handler

print 'Missing: subsequent tranaction log'
select 'File:'+LogFileName + ' FirstLSN:' + convert(varchar(30),FirstLSN)
+ ' LastLSN:' +convert(varchar(30),LastLSN)
from #LogFileLsn t3
where t3.LastLSN <> @MaxLSN
AND NOT exists(select * from #LogFileLsn t4
  where t4.FirstLSN = t3.LastLSN)

print 'Missing: previous tranaction log'
select 'File:'+LogFileName + ' FirstLSN:' + convert(varchar(30),FirstLSN)
+ ' LastLSN:' +convert(varchar(30),LastLSN)
from #LogFileLsn t1
  where t1.FirstLSN <> @MinLSN
  AND NOT exists(select * from #LogFileLsn t2
    where t2.LastLSN = t1.FirstLSN)
end

-- select LogFileName,FirstLSN,LastLSN from #LogFileLsn order by FirstLSN
-- get the database lsn
select top 1 @LogFileName = LogFileName from #LogFileLsn order by FirstLSN
desc
set @LogFileFullName = @LogBackupPath + @LogFileName
print @LogFileFullName

set @cmd = 'osql.exe -E -Q"RESTORE LOG ' + @DbName + ' from DISK="' +
@LogFileFullName + '" WITH STANDby="' + @UndoFileName + ""'
```

microsoft.public.sqlserver.security: Re: Transaction Log ??

```
-- create table #DL (Error nvarchar(500))
insert #DL EXEC master..xp_cmdshell @cmd

if @@ERROR<>0 OR exists( select * from #DL where Error LIKE '%4306%')
begin
    set @source = 'Restore Database Procedure'
    set @description=(select [description] from master..sysmessages where
error=4306)
    GOTO Error_Handler
end

declare @DbLSN numeric(25,0)
if exists( select * from #DL where Error LIKE '%4326%' OR Error LIKE
'%4305%')
select @DbLSN = CONVERT(numeric(25,0),substring(Error, 0,patindex('% can
be restored%',Error))) from #DL where Error LIKE '% can be restored%'
else
begin
print 'Last log file has been restored!'

GOTO EOF
end

if @DbLSN >= @MinLSN AND @DbLSN <= @MaxLSN
begin
if NOT exists(select * from #LogFileLsn where FirstLSN <= @DbLSN AND
LastLSN >= @DbLSN )
begin
set @source = 'Restore Database Procedure'
set @description= 'cannot find the log file containing the LSN: ' +
CONVERT(VARCHAR(30),@DbLSN)
GOTO Error_Handler
end

-- build restore file list
-- get rid of outdated log file, eliminate the "too early" message
delete #LogFileLsn where LastLSN < @DbLSN

declare @A numeric(25,0)
-- Find the first log backup file which doesn't have subsequent backup
file
if exists(select * from #LogFileLsn t3
where t3.LastLSN <> @MaxLSN
-- missing: subsequent transaction log
AND NOT exists(select * from #LogFileLsn t4
where t4.FirstLSN = t3.LastLSN))
begin
select @A=min(LastLSN) from #LogFileLsn t3
where t3.LastLSN <> @MaxLSN
-- missing: subsequent transaction log
AND NOT exists(select * from #LogFileLsn t4
```

Re: Transaction Log ??

microsoft.public.sqlserver.security: Re: Transaction Log ??

```
    where t4.FirstLSN = t3.LastLSN)
-- print '@A = ' + convert(varchar(30),@A)
delete #LogFileLsn where FirstLSN > @A
select @LogFileName = LogFileName from #LogFileLsn where LastLSN=@A
print 'The last log file to be restored : ' + @LogFileName
end

-- verify the first log file in the sequence, contain the required LSN
-- eliminate the "too late" message

declare @SeqMinLSN numeric(25,0)
select @SeqMinLSN = min(FirstLSN) from #LogFileLsn
-- print ' @DbLSN = ' + CONVERT(varchar(30),@DbLSN)
-- print '@SeqMinLSN = ' + CONVERT(varchar(30),@SeqMinLSN)
-- select * from #LogFileLsn
if @SeqMinLSN > @DbLSN
begin
    set @source = 'Restore Database Procedure'
    set @description= 'Exclamation!!! Exceptions: outdate, too
late,breakage. in a word, corrupted!!!'
    GOTO Error_Handler
end

declare rcr CURSOR READ_ONLY for
select LogFileName from #LogFileLsn order by FirstLSN
open rcr
while 'OK' = 'OK'
begin
    fetch next from rcr into @LogFileName
    if @@fetch_STATUS <> 0 BREAK
    set @LogFileFullName = @LogBackupPath + @LogFileName
    print 'Restoring ... ' + @LogFileFullName
    EXEC('RESTORE LOG ' + @DbName + ' from DISK="' + @LogFileFullName + ""
WITH STANDby="" + @UndoFileName + ""')
    if @@error <> 0
    begin
        set @source = 'Restore Database Procedure'
        set @description= 'Exclamation! Corrupted Log Files!!!'
        GOTO Error_Handler
    end

end
deallocate rcr
end
else if @DbLSN < @MinLSN
begin
    set @source = 'Restore Database Procedure'
    set @description='Database is outdated, try to restore the latest full
backup!'
    GOTO Error_Handler
end
```

Re: Transaction Log ??

microsoft.public.sqlserver.security: Re: Transaction Log ??

```
else if @DbLSN > @MaxLSN
begin
    set @source = 'Restore Database Procedure'
    set @description='Database is latest, no log file can be applied!'
    GOTO Error_Handler
end
GOTO EOF
```

```
Error_Handler:
declare @output varchar(255)
select @output = ' Source: ' + @source
print @output
select @output = ' Description: ' + @description
print @output
-- RAISERROR (@description, 16, 1)
EOF:
print 'Restore proc completed!'
-----
```

Any correction is welcome!

Peter Lin

"Shamim" <[shamim.abdul@railamerica.com](mailto:shamim.abdul@railamerica.com)> wrote in message news:#5Hi9\$ZMCHA.2456@tkmsftngp09...  
> *HI All,*  
> *SQL SERVER 7.0*  
>  
> *Iam in the process of restoring a database from last night complete backup*  
> *and then applying transaction log till the recent (we backup transaction*  
> *log*  
> *every 30 minutes).*  
>  
> *Transaction Log backups are stored with the name DB\_tlog\_200207221030 (for*  
> *today's 10.30 backup).*  
>  
> *In Enterprise Manager, when trying to restore it shows only the complete*  
> *backup. How to look at all the transaction log??*  
>  
> *In database Transaction Log is pointed to :d\...\dbLog.LDF.*  
> *If I want to use T-SQL*  
> *Example says*  
>  
> *RESTORE LOG MyDB FROM MyDB\_Log1 WITH NORECOVERY*  
>  
> *What is MyDB\_Log1 ??? is it transaction log filename ?? If so, should i*  
> *have*  
> *to give the full filename ??*  
>  
> *Please help/Advice.*  
>

Re: Transaction Log ??

microsoft.public.sqlserver.security: Re: Transaction Log ??

- > *Thanks*
- > *Shamim*
- >
- >