

sqlstd_sp.txt

```
USE [master]
GO
/***** Object: StoredProcedure [dbo].[sp_BackupDatabases] *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author: Microsoft
-- Create date: 2010-02-06
-- Description: Backup Databases for SQLExpress
-- Parameter1: databaseName
-- Parameter2: backupType F=full, D=differential, L=log
-- Parameter3: backup file location
-- =====

CREATE PROCEDURE [dbo].[sp_BackupDatabases]
    @databaseName sysname = null,
    @backupType CHAR(1),
    @backupLocation nvarchar(200)
AS

    SET NOCOUNT ON;

    DECLARE @DBs TABLE
    (
        ID int IDENTITY PRIMARY KEY,
        DBNAME nvarchar(500)
    )

    -- Pick out only databases which are online in case ALL databases are chosen to
    be backed up
    -- If specific database is chosen to be backed up only pick that out from @DBs
    INSERT INTO @DBs (DBNAME)
    SELECT Name FROM master.sys.databases
    where state=0
    AND name=@DatabaseName
    OR @DatabaseName IS NULL
    ORDER BY Name

    -- Filter out databases which do not need to backed up
    IF @backupType='F'
    BEGIN
        DELETE @DBs where DBNAME IN ('tempdb','Northwind','pubs','AdventureWorks')
    END
    ELSE IF @backupType='D'
    BEGIN
        DELETE @DBs where DBNAME IN
        ('tempdb','Northwind','pubs','master','AdventureWorks')
    END
    ELSE IF @backupType='L'
```

sqlstd_sp.txt

```
BEGIN
DELETE @DBs where DBNAME IN
('tempdb','Northwind','pubs','master','AdventureWorks')
END
ELSE
BEGIN
RETURN
END

-- Declare variables
DECLARE @BackupName varchar(100)
DECLARE @BackupFile varchar(100)
DECLARE @DBNAME varchar(300)
DECLARE @sqlCommand NVARCHAR(1000)
DECLARE @dateTime NVARCHAR(20)
DECLARE @Loop int

-- Loop through the databases one by one
SELECT @Loop = min(ID) FROM @DBs

WHILE @Loop IS NOT NULL
BEGIN

-- Database Names have to be in [dbname] format since some have - or _ in their
name
SET @DBNAME = '['+(SELECT DBNAME FROM @DBs WHERE ID = @Loop)+'']'

-- Set the current date and time n yyyyhhmmss format
SET @dateTime = REPLACE(CONVERT(VARCHAR, GETDATE(),101),'/',',') + '_' +
REPLACE(CONVERT(VARCHAR, GETDATE(),108),':',',')

-- Create backup filename in path\filename.extension format for full,diff and
log backups
IF @backupType = 'F'
SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME, '[' , '' ), ']' , '' )+
'_FULL_' + @dateTime+ '.BAK'
ELSE IF @backupType = 'D'
SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME, '[' , '' ), ']' , '' )+
'_DIFF_' + @dateTime+ '.BAK'
ELSE IF @backupType = 'L'
SET @BackupFile = @backupLocation+REPLACE(REPLACE(@DBNAME, '[' , '' ), ']' , '' )+
'_LOG_' + @dateTime+ '.TRN'

-- Provide the backup a name for storing in the media
IF @backupType = 'F'
SET @BackupName = REPLACE(REPLACE(@DBNAME, '[' , '' ), ']' , '' ) + ' full backup for '+
@dateTime
IF @backupType = 'D'
SET @BackupName = REPLACE(REPLACE(@DBNAME, '[' , '' ), ']' , '' ) + ' differential
backup for '+ @dateTime
IF @backupType = 'L'
SET @BackupName = REPLACE(REPLACE(@DBNAME, '[' , '' ), ']' , '' ) + ' log backup for '+
```

sqlstd_sp.txt

@dateTime

-- Generate the dynamic SQL command to be executed

```
IF @backupType = 'F'
BEGIN
SET @sqlCommand = 'BACKUP DATABASE ' +@DBNAME+ ' TO DISK = '''+@BackupFile+ '''
WITH INIT, NAME= '''+@BackupName+''', NOSKIP, NOFORMAT'
END
IF @backupType = 'D'
BEGIN
SET @sqlCommand = 'BACKUP DATABASE ' +@DBNAME+ ' TO DISK = '''+@BackupFile+ '''
WITH DIFFERENTIAL, INIT, NAME= '''+@BackupName+''', NOSKIP, NOFORMAT'
END
IF @backupType = 'L'
BEGIN
SET @sqlCommand = 'BACKUP LOG ' +@DBNAME+ ' TO DISK = '''+@BackupFile+ ''' WITH
INIT, NAME= '''+@BackupName+''', NOSKIP, NOFORMAT'
END
```

-- Execute the generated SQL command

```
EXEC(@sqlCommand)
```

-- Goto the next database

```
SELECT @Loop = min(ID) FROM @DBs where ID>@Loop
```

END