My sql:

EXEC sp\_MSforeachdb 'USE [?] IF ''?'' NOT IN(''master'', ''model'', ''msdb'', ''tempdb'', "ReportServer", "ReportServerTempDB")

SELECT ''?''as DbName, name FROM sysobjects WHERE xtype = ''U'' and lower(name) like ''%log%'' ORDER BY name'

<http://www.mssqltips.com/tip.asp?tip=1414>

Run The Same SQL Command Against All SQL Server Databases

**Read these related tips:**

* [Click here for related tips](http://www.mssqltips.com/category.asp?catid=26)
* [Still need your problem solved?](http://www.mssqltips.com/tipcomments.asp?tip=1414)
* [Let us know what you think about the site](http://www.mssqltips.com/survey20110419.asp)

**Problem**There are times when I find myself needing to run a SQL command against each database on one of my SQL Server instances. There is a handy undocumented stored procedure that allows you to do this without needing to set up a cursor against your sysdatabases table in the master database: sp\_MSforeachdb.

**Solution**The syntax for this undocumented procedure is:

**EXEC sp\_MSforeachdb @command**

(Where @command is a variable-length string.)

**Example 1: Query Information From All Databases On A SQL Instance**

|  |
| --- |
| --This query will return a listing of all tables in all databases on a SQL instance:  DECLARE @command varchar(1000)  SELECT @command = 'USE ? SELECT name FROM sysobjects WHERE xtype = ''U'' ORDER BY name'  EXEC sp\_MSforeachdb @command |

You can alternately omit the process of declaring and setting the @command variable. The T-SQL command below behaves identically to the one above and is condensed to a single line of code:

|  |
| --- |
| --This query will return a listing of all tables in all databases on a SQL instance:  EXEC sp\_MSforeachdb 'USE ? SELECT name FROM sysobjects WHERE xtype = ''U'' ORDER BY name' |

**Example 2: Execute A DDL Query Against All User Databases On A SQL Instance**

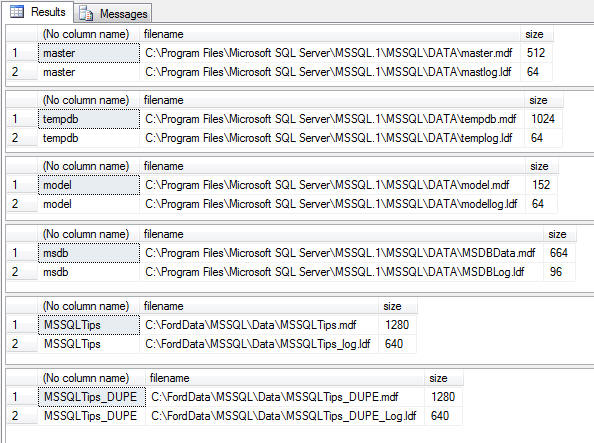
|  |
| --- |
| --This statement creates a stored procedure in each user database that will return a listing of all users in a database, sorted by their modification date  DECLARE @command varchar(1000)  SELECT @command = 'IF ''?'' NOT IN(''master'', ''model'', ''msdb'', ''tempdb'') BEGIN USE ? EXEC(''CREATE PROCEDURE pNewProcedure1 AS SELECT name, createdate, updatedate FROM sys.sysusers ORDER BY updatedate DESC'') END'  EXEC sp\_MSforeachdb @command |

As you may notice, there are additional items to take into consideration when limiting the scope of the sp\_MSforeachdb stored procedure, particularly when creating or modifying objects. You must also set the code to execute if the IF statement is true by using the T-SQL keywords BEGIN and END. You should take note that the USE ? statement is contained within the BEGIN...END block. It is important to remember key T-SQL rules and account for them. In this case the rule that when creating a procedure, the CREATE PROCEDURE phrase must be the first line of code to be executed. To accomplish this you can encapsulate the CREATE PROCEDURE code within an explicit EXEC() function.

**What about the "?" Placeholder**Throughout the examples provided above you'll see the use of the question mark as a placeholder for the database/database name. To reference the database name as a string to be returned in a query, embed it between a double set of single quotation marks. To treat it as a reference to the database object simply use it by itself (as presented in Example 3b.) It is necessary to set the database for the query to run against, by using the USE ? statement, otherwise the code will execute in the context of the current database, for each database in your SQL instance. If you have 5 databases hosted in the current instance and you were to run the stored procedure code above while in the context of DBx it would execute the T-SQL text of the @command 5 times in DBx. This behavior is evident in the output of Example 3 below.

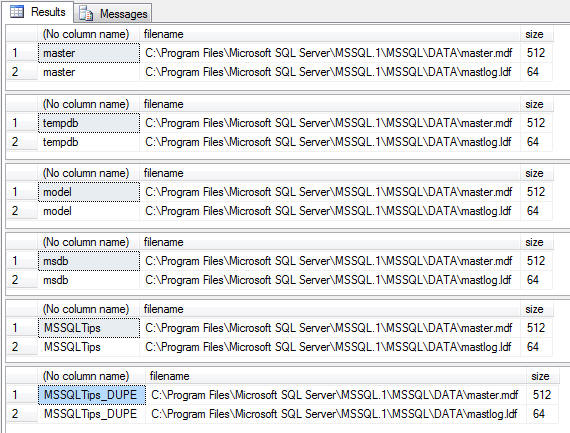
**Example 3: Query File Information From All Databases On A SQL Instance**

|  |
| --- |
| --This query will return a listing of all files in all databases on a SQL instance:  EXEC sp\_MSforeachdb 'USE ? SELECT ''?'', SF.filename, SF.size FROM sys.sysfiles SF' |



What happens though if we omit the USE ? clause, which sets the scope of the query? As you can see below, though it is apparent the code executed for each database, it never changed context. Pay particular interest to the filename column and you will see that the query executed from within the context of the master database (where I ran the query from) for each database in the SQL instance (as noted by the database name being returned via the use of the "?" placeholder).

|  |
| --- |
| --Remove the USE [?] clause and you end up executing the query repetitively within the context of the current database:  EXEC sp\_MSforeachdb 'USE [?] SELECT ''?'', SF.filename, SF.size FROM sys.sysfiles SF' |



**Why Not Just Use a Cursor?**Sure, a cursor can accomplish all that I've presented above, but let's look at the code required to set up a cursor to execute the command used in Example 3:

|  |
| --- |
| DECLARE @DB\_Name varchar(100)  DECLARE @Command nvarchar(200)  DECLARE database\_cursor CURSOR FOR  SELECT name  FROM MASTER.sys.sysdatabases  OPEN database\_cursor  FETCH NEXT FROM database\_cursor INTO @DB\_Name  WHILE @@FETCH\_STATUS = 0  BEGIN       SELECT @Command = 'SELECT ' + '''' + @DB\_Name + '''' + ', SF.filename, SF.size FROM sys.sysfiles SF'      EXEC sp\_executesql @Command       FETCH NEXT FROM database\_cursor INTO @DB\_Name  END  CLOSE database\_cursor  DEALLOCATE database\_cursor |

Considering the behavior is similar I'd rather type and execute a single line of T-SQL code versus sixteen.

**Next Steps**

* sp\_MSforeachdb is extremely useful for pulling together metadata about your various SQL databases. I use it quite frequently for reporting on such important metrics as database file sizes, amount of free space, and backup status.
* In future tips I will present how to collect those metrics and many more, and report on them via SQL Server Reporting Services. Stay Tuned!

|  |  |
| --- | --- |
| **Wednesday, December 10, 2008 - 3:23:23 PM - robvon** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=2387) |
| You may need some "[" and "]" around the '?' to deal with dodgy database names.......   as in:  EXEC sp\_MSforeachdb 'USE [?] SELECT ''?''as DbName, SF.filename, SF.size FROM sys.sysfiles SF'  R | |

|  |  |
| --- | --- |
| **Sunday, December 14, 2008 - 9:04:32 PM - timmer26** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=2404) |
| You are absolutely correct.  This is particularly important when dealing with spaces in database names. | |

|  |  |
| --- | --- |
| **Friday, November 20, 2009 - 3:49:19 AM - bobmihada** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=4474) |
| --// Just another flavor to offer IF OBJECT\_ID('tempdb..##TempDBInfo') IS NOT NULL BEGIN PRINT 'Previous ##TmpDBInfo found and Dropped.' DROP TABLE ##TempDBInfo END CREATE TABLE ##TempDBInfo(DBName nVarChar(255), FileName nvarchar(255), fileSize INT) EXEC sp\_MSforeachdb 'USE [?] INSERT INTO ##TempDBInfo SELECT ''?'', SF.filename, SF.size FROM sys.sysfiles SF' SELECT \* FROM ##TempDBInfo --// DROP TABLE ##TempDBInfo | |

|  |  |
| --- | --- |
| **Wednesday, January 27, 2010 - 11:19:23 AM - Akbar** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=4799) |
| Hi All, Is there are way to grant Execute permission to a particular Stored procedure or Function in all database of SQL Server 2008 Instance? The below one use to Grant Execute permission to any procedure.   EXEC sp\_msforeachdb 'USE ?; GRANT EXECUTE TO UserOrRoleName'  But when trying the below one I'm getting error  EXEC sp\_msforeachdb 'USE ?; GRANT EXECUTE ON [?].[dbo].[sp\_StoredProc1] TO UserOrRoleName'  Error Message: Cannot find the object 'sp\_StoredProc1', because it does not exist or you do not have permission.  I have the 'sp\_StroedProc1' exist in all my database in the sql server 2008 instance. Any help will be highly appreciated. | |

|  |  |
| --- | --- |
| **Friday, January 29, 2010 - 12:02:55 PM - aprato** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=4820) |
| EXEC sp\_msforeachdb 'USE ?; if object\_id(''sp\_StoredProc1'') is not null begin print ''?'' GRANT EXECUTE ON [dbo].[sp\_StoredProc1] TO public end' | |

|  |  |
| --- | --- |
| **Monday, April 26, 2010 - 3:02:40 PM - thomashmaine** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=5287) |
| is there a way to have a variabled declared  withing the statement?    select @command1 = 'use [?]; exec sp\_grantlogin [@user2]'  print @user2  exec sp\_MSforeachdb @command1 | |

|  |  |
| --- | --- |
| **Wednesday, November 03, 2010 - 10:11:28 AM - Pradip Patel** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=10329) |
| You need to replace       SELECT @Command = 'SELECT ' + '''' + @DB\_Name + '''' + ', SF.filename, SF.size FROM sys.sysfiles SF' with       SELECT @Command = 'USE '+ @DB\_NAME +  ' SELECT ' + '''' + @DB\_Name + '''' + ', SF.filename, SF.size FROM sys.sysfiles SF'  when you want to execute by cursor otherwise it gives you all file name as master.mdf | |

|  |  |
| --- | --- |
| **Tuesday, February 01, 2011 - 11:49:50 AM - Larrybud** | [read the tip](http://www.mssqltips.com/tip.asp?tip=1414)[flag as SPAM](http://www.mssqltips.com/tipcommentscomplaint.asp?id=12774) |
| There are times when you might need a cursor:  1) running a script against certain databases, instead of all  2) wanting your script results in one table (for a select, for example) | |