

## Re: typo error...

**Source:**

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2005-02/1096.html>

---

**From:** Ami Levin (XXX\_NO\_SPAM\_\_XXX\_Levin\_Ami\_at\_Yahoo.com)

**Date:** 02/01/05

Date: Tue, 1 Feb 2005 19:35:28 +0200

Hi Paula,

Yes, you are correct. I said I didn't test it :-)

This trick works with some errors but not with linked server errors apparantly.

I see now what you ment with the creation of the so called temp SP.

I remember I encountered the same problem a few years ago but I can't remember how I solved it eventually and digging through my old scripts would take forever.

In the mean time, till I find my old scripts or till someone has a better idea,

here is a very creative, bad and "ugly" solution that works (this time I tested it) using OSQL with xp\_cmdshell to seperate the batches completely.

Since xp\_cmdshell does not return error codes, we need to create an object and then test for it's existance. Just change "TempDB" to your DB name

I recommend leaving the WHERE 1=0 to make this trick a little less costly than it already is.

You can use dynamic execution to build the query like we did before.

```
EXECUTE master..xp_cmdshell
'osql -E -d "tempdb" -Q "SELECT NULL AS N INTO foo FROM
my_srv.my_db.dbo.my_table WHERE 1=0", no_output
IF OBJECT_ID('foo') IS NOT NULL
BEGIN
SELECT 'success'
DROP TABLE foo
END
ELSE SELECT 'Fail'
```

SQL Server 2005 will solve all these issues (hopefully :-).

HTH

Ami

microsoft.public.sqlserver.programming: Re: typo error...

"PaulaPompey" <PaulaPompey@discussions.microsoft.com> wrote in message  
news:A67038AD-FBE0-4751-BA89-0B962BD1763E@microsoft.com...

> *Hi Ami*

>

> *It's just not working if I do it this way – halts after connection to*

> *linked*

> *server fails.*

>

> *CREATE PROCEDURE DayOldDataCheck\_02\_TestConnection @server SYSNAME,@DB*

> *SYSNAME,@owner SYSNAME,@table SYSNAME AS*

> *DECLARE @TSQL varchar(1000)*

> *SET @TSQL = 'SELECT TOP 1 \* FROM [' + @Server + '].' + @DB + '.' + @Owner*

> *+*

> *'.' + @table*

> *EXEC (@TSQL)*

> *RETURN @@error*

> *GO*

>

> *--Repeat code*

> *DECLARE @RETURN\_CODE int*

> *EXEC @RETURN\_CODE = DayOldDataCheck\_02\_TestConnection*

> *'I','TravelDB','dbo','Status'*

> *SELECT @RETURN\_CODE*

>

> *Any ideas?*

>

> *Regards*

>

> *Paula*

>

> *"Ami Levin" wrote:*

>

>> *Hi Paula,*

>>

>> *You don't need the "GO" which is a batch seperator used for query*

>> *analyzer /*

>> *osql etc and is not part of T-SQL.*

>> *I don't see why you need to create a temp procedure for that.*

>> *Why don't you just take the names as parameters and use dynamic execution*

>> *inside a generic procedure?*

>> *(code not tested)*

>>

>> *CREATE PROC test\_availability*

>> *@server SYSNAME,*

>> *@DB SYSNAME,*

>> *@owner SYSNAME,*

>> *@table SYSNAME*

>> *AS*

>> *DECLARE @sql NVARCHAR(1000)*

>> *SET @sql = 'SELECT NULL FROM ' + @Server + '.' + @DB + '.' + @Owner + '.'*

>> *+*

Re: typo error...

```
>> @table + ' WHERE 1=0'
>> EXEC (@sql)
>> RETURN @@error
>> GO
>>
>> -- inside your caller SP:
>> DECLARE @Error INT
>> EXEC @Error = test_availability 'server1','db2','dbo','my_table'
>> IF @Error <> 0 blahblahblah.....
>>
>> I've used WHERE 1=0 so that it won't actually scan the whole table in
>> case
>> it is available...
>>
>> HTH
>>
>> Ami
>>
>> "PaulaPompey" <PaulaPompey@discussions.microsoft.com> wrote in message
>> news:FEC9816F-104E-4490-8256-11BBF277E0EB@microsoft.com...
>> > Hi Ami
>> >
>> > Thanks very much for the advice.
>> >
>> > I can now do what I want to do through Query Analyser using the
>> > following
>> > code, however, I'd like to save this code in a stored procedure but
>> > can't
>> > because of the need to us "GO" after dropping the temp procedure and
>> > creating
>> > procedure – any tips on how to get around this?
>> > Code I have working is as follows:
>> > --This procedure is not re-created each time
>> > CREATE PROCEDURE DayOldDataCheck_CheckConnection @ERR int OUTPUT AS
>> > exec Temp1
>> > SELECT @ERR = @@error
>> > RETURN
>> > GO
>> >
>> > --This is the code I need to run every night within another process
>> > --I'd like to be able to save this in a stored procedure of it's own
>> > DROP PROCEDURE Temp1
>> > GO
>> > CREATE PROCEDURE Temp1 AS SELECT * FROM [LINKEDSVR].TravelDB.dbo.Status
>> > GO
>> >
>> > DECLARE @ERR_CODE int
>> >
>> > exec DayOldDataCheck_CheckConnection @ERR = @ERR_CODE OUTPUT
>> >
>> > exec DayOldDataCheck_LiveData 'TravelDB', @ERR_CODE
```

```
>>>
>>> Thanks again for your help
>>>
>>> Paula
>>>
>>> "Ami Levin" wrote:
>>>
>>>> First Proc should be p1 and not p3
>>>>
>>>>> create proc p1 as
>>>>> select * from nonexistent
>>>>> select 'passed after error', @@error
>>>>> go
>>>>>
>>>>> exec p1
>>>>> -- batch was terminated without returning the message
>>>>>
>>>>> "Ami Levin" <XXX_NOSPAM__XXX__amlevin@mercury.com> wrote in message
>>>>> news:uqAPs2ECFHA.3824@TK2MSFTNGP10.phx.gbl...
>>>>> Hi Paula,
>>>>>
>>>>> Error handling in SQL Server 2000 is "somewhat" problematic as you
>> have
>>>>> seen.
>>>>> For these cases, I use the following trick of nesting the execution
>>>>> scopes:
>>>>>
>>>>> USE tempdb
>>>>>
>>>>> select * from nonexistent
>>>>> select 'passed after error', @@error
>>>>> go
>>>>> -- batch was terminated without returning the message
>>>>>
>>>>> exec ('select * from nonexistent')
>>>>> select 'passed after error', @@error
>>>>> go
>>>>> -- inner scope was aborted, outer scope continued
>>>>>
>>>>> create proc p3 as
>>>>> select * from nonexistent
>>>>> select 'passed after error', @@error
>>>>> go
>>>>>
>>>>> exec p1
>>>>> -- batch was terminated without returning the message
>>>>>
>>>>> create proc p2 as
>>>>> select * from nonexistent
>>>>> go
```

```
>>>>>
>>>>> create proc p3 as
>>>>> exec p2
>>>>> select 'passed after error', @@error
>>>>> go
>>>>>
>>>>> exec p3
>>>>> -- inner procedure was aborted, outer procedure continued
>>>>>
>>>>> This should work for most cases although some errors will stop and
>>>>> rollback
>>>>> the whole batch including outer scopes.
>>>>> I have tested it with inaccessible linked servers and it worked
>>>>> fine
>> for
>>>> me.
>>>>>
>>>>> See the following thread for more details:
>>>>>
>>>>>
>>>>>
>>
<a href="http://groups-beta.google.com/group/microsoft.public.sqlserver.programming/browse_thread/thread/df3390d2b3475...">http://groups-beta.google.com/group/microsoft.public.sqlserver.programming/browse_thread/thread/df3390d2b3475...
>>>>>
>>>>> HTH
>>>>>
>>>>> Ami
>>>>>
>>>>> "PaulaPompey" <PaulaPompey@discussions.microsoft.com> wrote in
>>>>> message
>>>>> news:752B8EAA-BC40-4B0D-B413-EFC8F94189A7@microsoft.com...
>>>>>> Over night we take a copy of various live SQL databases onto
>>>>>> another
>>> SQL
>>>>>> server for reporting purposes.
>>>>>> I have a stored procedure that compares the latest live data
>>>>>> against
>>> the
>>>>> I
>>>>>> day old copies to ensure that they are up to date.
>>>>>> I connect to the live databases using linked servers.
>>>>>> Here's where the problem is - when one of the external links is
>>>>>> down
>>> or
>>>>>> one
>>>>>> of the live databases is offline the stored procedure has an
>>>>>> error
>>> and
>>>>>> stops.
>>>>>> How can I test within the stored procedure that the database on
>>>>>> the
```

