

Re: Indexing Service, Openquery and sp_executesql

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.fulltext/2004-08/0118.html>

From: Hilary Cotter (*hilaryk_at_att.net*)

Date: 08/10/04

Date: Tue, 10 Aug 2004 13:26:32 -0400

I am completely bewildered by your post.

I modified your code and it works on my machine.

are you saying that my code does not work on your machine?

What os, SQL Server version and sp are you running?

```
--
Hilary Cotter
Looking for a book on SQL Server replication?
http://www.nwsu.com/0974973602.html
"Fritz" <fritzwells@fdwells.cnc.net> wrote in message
news:35d101c47efd$815762f0$a301280a@phx.gbl...
> Hilary
>
> Thanks for the reply.
>
> I tested your code against a valid linked server and it
> produced an unparameterized select statement that returned
> data from the indexing service catalog when pasted into
> query analyzer, but failed when put against sp_executesql
> in a sp. ie
>
> Exec sp_executesql @strString
>
> I modified the string concatenation as:
> @strString = @strString + N'additional text'
>
> I practiced with sp_executesql in a sp with sql database
> tables and I think that I have the syntax down for using
> parameters with sp_executesql.
>
> However, I'm wondering if there is something about
> sp_executesql that doesn't work with Openquery. I choose
> sp_executesql because of the parameters and I can't get it
> to work without them.
>
> I would be interested to know if anyone else has been
> successful in using sp_executesql with Openquery, or has
> an alternate approach that would allow multiple queries
> against an indexing service catalog in the shortest
```

microsoft.public.sqlserver.fulltext: Re: Indexing Service, Openquery and sp_executesql

```
> possible time.
>
> Fritz
>
>
> >-----Original Message-----
> >I think your problem is with your query as opposed to any
> >ansi settings
> >
> >Create PROCEDURE sp_IndexQuery
> >(@strword nvarchar(100),
> >@strDrive nvarchar(100)
> >)
> > AS
> >
> >Declare @strString as nvarchar(500)
> >
> >set @strString = 'Select R.FileName,R.Directory From
> >Openquery(system,'Select Path,FileName,Directory From
> >Scope(''')deep
> >traversal of '
> >select @strString=@strString+@strDrive+'''') Where
> >Contains(''')
> >select @strString=@strString+ @strWord +''''')' ) as R'
> >
> >Exec sp_executesql @strString,N'@Word nvarchar(100),@Drive
> >nvarchar(100)',@Word=@strword,@Drive=@strDrive
> >GO
> >
> >--
> >Hilary Cotter
> >Looking for a book on SQL Server replication?
> >http://www.nwsu.com/0974973602.html
> >
> >
> >"Fritz" <fritzwells@fdwells.cnc.net> wrote in message
> >news:335301c47edf$9e5cbd0$a301280a@phx.gbl...
> >> The following is from an application that uses Microsoft
> >> Access to survey text to discover the occurrence of
> > words
> >> of interest to the user.
> >>
> >> A string search where tblDoc.DocumentText is the text of
> >> the document to be searched and LevelOne.word is a
> > column
> >> in a table containing 12,000 words of interest. If
> > there
> >> are hits from the LevelOne list, additional queries
> >> against the document table are made using more lists of
> >> words (LevelTwo and LevelThree Tables). Each document
> > is
> >> searched in turn and the word hits are recorded in a
> > Hits
> >> table
> >>
> >> strSQL = "select DISTINCT LevelOne.Word " & _
> >>           "FROM LevelOne, tblDoc " & _
> >>           "WHERE tblDoc.DocumentText " & _
> >>           "LIKE '%" & ' ' & LevelOne!Word & ' '
> >> & '%" "
> >> & _
> >>           "AND tblDoc.DocumentID='" & Current_Doc
```

microsoft.public.sqlserver.fulltext: Re: Indexing Service, Openquery and sp_executesql

```
> & "';"
> >>
> >> Since this application is not using a full-text index on
> >> TblDoc.DocumentText, the queries for each document can
> >> take a minute or two, depending on the size of the text
> >> and the number of hits from the first query using the
> >> LevelOne words. With the number of documents to be
> >> searched approaching 600,000 documents, the whole
> process
> >> can take hours or days to complete.
> >>
> >>
> >> My idea to speed up this application is to migrate this
> >> application into an environment where the text from the
> >> documents could first be indexed and then searched;
> hoping
> >> that the time to perform the indexing would be offset by
> >> the increased speed in running the queries to find
> words.
> >>
> >> The application would like to "look into" words and find
> >> the target word as both a prefix or a suffix, using
> >> wildcards. Following John Kane's comment in an earlier
> >> email that Full-Text Search is not designed for string
> >> pattern matching, I would choose Microsoft Indexing
> >> Service over Microsoft SQL Full Text Search. His
> >> suggestion to use PATINDEX would not give me the benefit
> >> of an index on the text data.
> >>
> >> Using Microsoft Indexing Service, I would first index
> all
> >> the documents as files. At this point the documents are
> >> text files. I would then search all the documents for
> >> words in LevelOne, LevelTwo and LevelThree tables in
> turn,
> >> rather than search each document in turn for all levels
> of
> >> words.
> >>
> >> Using a SQL Server and Microsoft Index Server 2.0 or
> >> Microsoft Indexing Service 3.0, my stored procedure
> query
> >> might be:
> >>
> >> Select R.FileName,R.Path
> >> FROM OPENQUERY(MyLinkedServer,
> >> 'Select FileName,Path
> >> FROM SCOPE('' "D:\Documents" '')
> >> WHERE CONTAINS('' "Canada" '')') >0 as R
> >>
> >> What is the best way to substitute for the word
> >> value "Canada" from a table, LevelOne, for a series
> >> (~12,000) of queries.
> >>
> >>
> >> I have attempted to create a stored procedure on SQL
> 2000
> >> (sp3) using sp_executesql, hoping to reuse the
> optimizer's
> >> execution plan, but have failed to get it to work. Am I
> >> attempting the impossible with this approach?
> >>
```

microsoft.public.sqlserver.fulltext: Re: Indexing Service, Openquery and sp_executesql

```
> >> The sp:
> >>
> >> CREATE PROCEDURE sp_IndexQuery
> >> (@strword nvarchar(100),
> >> @strDrive nvarchar(100)
> >> )
> >> AS
> >>
> >> Declare @strString as nvarchar(500)
> >>
> >> set @strString ='Select R.FileName,R.Directory From
> >> Openquery(LinkedServer,' 'Select Path,FileName,Directory
> >> From Scope(@Drive) Where Contains(@Word )' ' ) as R'
> >>
> >>
> >> Exec sp_executesql @strString,N'@Word nvarchar
> (100),@Drive
> >> nvarchar(100)',@Word=@strword,@Drive=@strDrive
> >> GO
> >> -----
> >> The call to the sp:
> >> exec sp_indexquery 'Canada','C:\'
> >>
> >> -----
> >> The error message:
> >>
> >> Server: Msg 7405, Level 16, State 1, Line 1
> >> Heterogeneous queries require the ANSI_NULLS and
> >> ANSI_WARNINGS options to be set for the connection. This
> >> ensures consistent query semantics. Enable these options
> >> and then reissue your query.
> >> -----
> >> I have set ansi_nulls on and ansi_warnings on for the
> >> database that contains this procedure.
> >>
> >> I do get results when I run the query in the query
> >> analyzer without the parameters. Both the documents, the
> >> indexing service catalog and the instance of SQL Server
> >> 2000 are running on my Win2k workstation.
> >>
> >> Thanks for any suggestions.
> >>
> >> Fritz
> >
> >
> >.
> >
```