

Re: Identity error?? Please help

Source: <http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.database.ado/2004-05/0119.html>

From: Fabio Dell'Aria (*thesun_at_inwind.it*)

Date: 05/11/04

Date: 11 May 2004 01:24:16 -0700

mc-nospam-google@durbin.net (Marc Durbin) wrote in message
news:<73b873d2.0405101649.475df1f3@posting.google.com>...

> "Andre Greyling" <agreyling@optusnet.com.au> wrote in message
news:<4085148d\$1@newsgroups.borland.com>...

>> Thanks for the help – yes the problem is @@IDENTITY related. It is still
>> strange how Delphi behaves though. The code is simply:

>> (1) qry1.append;

>> (2) qry1.fieldbyname('ID').asinteger

>>

>> qry1 does a simple SELECT on table Master. Master has a trigger (on INSERT)

>> which creates detail tables

>> each table has an identity column.

>> line (2) then returns the identity value of the last detail table – odd !

>>

>> However, what is the best way to create a master record and retrieve it's

>> identity value ? Should the code be in a stored procedure or is there a

>> simpler way ?

>> Any ideas ?

>>

>> Bye

>

> It sounds like you've run up against the ADO bug of the year... ADO

> uses @@IDENTITY to retrieve the last inserted identity value after a

> Post, when it should be using SCOPE_IDENTITY(). There has been much

> discussion on whether this issue is a 'bug' as it works as

> documented... But if it is not a bug, then it is certainly an

> extremely serious design flaw, IMHO. I have seen various workarounds

> listed online, none of which really solve the problem:

>

> 1. Don't use triggers.

>

> 2. Always use stored procedures for INSERTs and UPDATEs.

>

> 3. Don't use IDENTITY columns

>

> 4. Microsoft recommends: You could implement a business object that

> dispenses "identity" column information directly to your application.

> This dispenser should be free-threaded, in case multiple applications

> are using it, and it probably needs to work with clients across a
> network. In this case, a Microsoft Transaction Server implementation
> would work the best. (KB article 195910)
>
> 5. Use GUIDs in place of IDENTITY
>
> The scary and ugly solution that we are using (because it just
> *works*) is to save the @@IDENTITY value on entry to a trigger and
> restore it at the end. This is not a new idea – similar solutions
> have been used with SQL Server 7.0 in the past to work around the lack
> of a SCOPE_IDENTITY() function. For example:
>
> DECLARE @example TABLE (ii INT IDENTITY (5,23))
>
> INSERT @example DEFAULT VALUES
> PRINT 'Initial Identity = '+CAST(@@IDENTITY AS VARCHAR)
>
> -- START OF IDENTITY BACKUP
>
> DECLARE @BackupIdentitySeederFunc VARCHAR(1000)
> SET @BackupIdentitySeederFunc =
> 'DECLARE @BackupIdentity TABLE
> (IdentityID INT IDENTITY('+CAST(@@IDENTITY AS VARCHAR)+' , 1))
> INSERT @BackupIdentity DEFAULT VALUES'
>
> -- END OF IDENTITY BACKUP
>
> INSERT @example DEFAULT VALUES
> PRINT 'New Identity = '+CAST(@@IDENTITY AS VARCHAR)
>
> INSERT @example DEFAULT VALUES
> PRINT 'Another New Identity = '+CAST(@@IDENTITY AS VARCHAR)
>
> -- RETRIEVE ORIGINAL IDENTITY
>
> EXEC (@BackupIdentitySeederFunc)
> PRINT 'Back to Original Identity = '+CAST(@@IDENTITY AS VARCHAR)
>
> We have ending up with this solution because each of the other options
> suggested cause no end of problems for us... and it is a simple patch
> to our triggers that should work even if ADO behaviour changes in the
> future.
>
> GUIDs would probably be the most flexible solution – but they do have
> a substantial performance and size cost – for performance, both in
> INSERTs and in SELECTs, as their randomness makes the indexes
> inefficient. I found one document that had a workaround for this
> performance cost – but I don't know how safe the method described is:
> <http://www.informit.com/articles/prINTERfriendly.asp?p=25862>.
>
> You would probably also need to use GUIDs from the beginning of a

borland.public.delphi.database.ado: Re: Identity error?? Please help

- > *project, as the changeover cost would be very high.*
- >
- > *I hope that one of these ideas might help resolve the issue for you.*

You can try EurekaLog (www.eurekalog.com).

EurekaLog is an add-in tool that gives to your application (GUI, Console, Web, etc.) the ability to catch every exception (even those raised by memory leaks) and every infinite-loops/deadlock bugs, generating a detailed log of call stack (with unit, class, method and line #), showing and sending it back to you via email.

--
Best regards...
Fabio Dell'Aria.