

Re: About MSDTC,The same ContextUtil.TransactionId,but the state of Database A,B is different.

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.adonet/2004-12/1414.html>

From: Sahil Malik (contactmethrumblog_at_nospam.com)

Date: 12/28/04

Date: Mon, 27 Dec 2004 22:52:49 -0500

Try this – change [Transaction(TransactionOption.RequiresNew)] to [Transaction(TransactionOption.Required)]

– SM

"Huang Lin" <lin-huang@vip.sina.com> wrote in message
news:OuPKqHI7EHA.1452@TK2MSFTNGP11.phx.gbl...

> *Thanks for your reply,Sahil Malik.*

> *I think the only one MSDTC is used(which is in application server),MSDTC*

> *communicate with RMs of different databases.*

>

> *Here is my code below:*

>

> *public interface IDistributeTransaction*

> {

> *int StartTransaction();*

>

> *int CommitTransaction();*

>

> *int RollBackTranstion();*

>

> *int ExecuteSql(string connectionstring,string SqlStr,out string ErrStr);*

>

> }

>

> *public class AppDistributeTrans:MarshalByRefObject,IDistributeTransaction*

> {

>

> *//It stores DistributeTransaction Instance;*

> *private static ArrayList m_DistributeTransList;*

>

> *//The exact Instance which created when the first time remote machine*

> *called*

```
> .
> private DistributeTransaction m_DistributeTrans
>
> public AppDistributeTrans()
> {
> //Create or find the proper Instance of DistributeTransaction,each
> Instance have a different ID,so I can find it.
> .....
>
>
> }
>
> #region IDistributeTransaction ³ÉÔ±
>
> int ShangHaiERP.Corp.IDistributeTransaction.StartTransaction()
> {
> return m_DistributeTrans.StartTransaction();
>
> }
>
> int ShangHaiERP.Corp.IDistributeTransaction.RollBackTranstion()
> {
> int rt = m_DistributeTrans.RollBackTranstion();
> m_DistributeTrans = null;
> return rt;
> }
>
>
> int ShangHaiERP.Corp.IDistributeTransaction.CommitTransaction()
> {
> int rt = 0;
> string err = "";
> try
> {
> rt = m_DistributeTrans.CommitTransaction();
>
> }
> catch(Exception ex)
> {
> err = ex.Message + ex.StackTrace + ex.Source;
> }
>
> m_DistributeTrans = null;
> return rt;
> }
>
> int ShangHaiERP.Corp.IDistributeTransaction.ExecuteSql(string
> connectionstring,string SqlStr,out string ErrStr)
> {
> return m_DistributeTrans.ExecuteSql(connectionstr,SqlStr,out
> ErrStr);
```

```
> }
>
>
> #endregion
>
> }
>
> [ Transaction( TransactionOption.RequiresNew) ]
> [ ObjectPooling(true, 5, 10) ]
> public class DistributeTransaction:ServicedComponent
> {
> .....
>
> public DistributeTransaction()
> {
>
> }
>
> #region IDistributeTransaction
>
> //Just do nothing,I don't want to autocomplete,programmer can use it just
> like ADO.Net's transaction.
> public int StartTransaction()
> {
> AddLogEx(1,m_LogFileName,"StartTrans" +
> ContextUtil.TransactionId.ToString());
> return 0;
> }
>
> public int RollBackTranstion()
> {
> AddLogEx(1,m_LogFileName,"RollBack" +
> ContextUtil.TransactionId.ToString());
> ContextUtil.SetAbort();
> return 0;
> }
>
>
> public int CommitTransaction()
> {
>
> AddLogEx(1,m_LogFileName,ContextUtil.TransactionId.ToString());
> ContextUtil.SetComplete();
> return 0;
> }
>
>
>
> public int ExecuteSql(string connectionstr, string SqlStr, out string
> ErrStr)
> {
```

```
> ErrStr = "";
> try
> {
> AddLogEx(1,m_LogFileName,"Exe1" +
> ContextUtil.TransactionId.ToString());
> OleDbConnection cn = new OleDbConnection(connectionstr);
> OleDbCommand cm = new OleDbCommand(SqlStr);
> cm.Connection = cn;
> cn.Open();
> cm.ExecuteNonQuery();
> cn.Close();
>
> }
> catch(OleDbException ex)
> {
> AddLogEx(1,m_LogFileName,ex.Message);
> return DistributeTransaction.ERROR_SQL;
>
> }
> catch(Exception ex)
> {
> AddLogEx(1,m_LogFileName,ex.Message);
> return DistributeTransaction.ERROR_OTHER;
>
> }
> return 0;
> }
>
> private bool AddLogEx(int LogLevel,string UserName,string LogStr)
> {
> string FileName;
> string TmpStr,LogFileDir;
> StreamWriter Sw;
>
> LogFileDir = m_LogPath;
> Directory.CreateDirectory(LogFileDir);
>
>
> switch (LogLevel)
> {
> case 0:
> FileName = LogFileDir + UserName + "_DBLOG";
> break;
> case 1:
> FileName = LogFileDir + UserName + "_LOG";
> break;
> default:
> goto case 1;
> }
>
>
```

```
> try
> {
>
> FileInfo Fi = new FileInfo(FileName);
> if (Fi.Exists == false)
> {
> FileStream Fs = new FileStream(FileName, FileMode.Create);
> Fs.Close();
> }
> if (Fi.Length > this.m_FileMaxSize)
> {
> Fi.CopyTo(FileName + "_" +
> DateTime.Now.ToString("yyyyMMddHHmm"),true);
> Fi.Delete();
> }
> }
> catch
> {
> }
>
> Sw = new StreamWriter(FileName,true);
> try
> {
> TmpStr = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss # ") + LogStr;
> Sw.WriteLine(TmpStr);
> Sw.Close();
> }
> catch (Exception e)
> {
> TmpStr = e.Message;
>
> return false;
> }
> return true;
> }
>
> #endregion
> }
>
> "Sahil Malik" <contactmethrumyblog@nospam.com> Ð`ÈëÓÊ¼þ
> news:uSx4xfC7EHA.1112@TK2MSFTNGP14.phx.gbl...
>> Oh my, so we have remoting, msdtc on the two appdomains possibly two
>> machines involving remoting, and we have two databases involving a
>> distributed transaction between them.
>>
>> My first kneejerk reaction is to try and find a simpler solution to what
> you
>> are trying to acheive. MSDTC transactions by default use
>> IsolationLevel.Serializable between databases, in Sql Server 2005 and
>> .NET
>> 2.0 combination, this is slightly better in the sense that the isolation
```

>> level is promoted to serializable once the transaction enlists a second
>> database. But still it ends up becoming Serializable which is a complete
>> pig.
>>
>> The second reason for my recommendation to change your architecture is
>> the
>> heavy duty traffic you will generate due to two MSDTC instances on two
>> machines.
>>
>> The third reason is the pain you will have to go through as far as
>> deployment of your application is concerned.
>>
>> And the final reason is System.Transactions in .NET 2.0 will subsume
>> transactional capabilities of System.EnterpriseServices, but maybe you
>> are
>> writing the below code for .NET 2.0 as nothing in your post seems to
>> contradict that. Anyway, the above 3 reasons still hold good.
>>
>> If you decide to change the architecture a bit, please let us know what
>> exactly are you trying to acheive and we could suggest something. Can the
>> two databases talk to each other? viz. can you setup a remote login
> between
>> them and do a BEGIN DISTRIBUTED TRANSACTION ... or ... use linked tables
> to
>> acheive the same?
>>
>> But if for some reason you have compelling reasons to use MSDTC, I would
>> need to see how exactly is your class that implements
> IDistributeTransaction
>> and m_DistributeTrans is an instance of, ... how exactly is that
>> enlisting
>> itself the transactions.
>>
>> – Sahil Malik
>> <http://dotnetjunkies.com/weblog/sahilmalik>
>> [http://blogs.apress.com/authors.php?author=Sahil Malik](http://blogs.apress.com/authors.php?author=Sahil%20Malik)
>>
>>
>>
>>
>>
>>
>>
>> "Huang Lin" <lin-huang@vip.sina.com> wrote in message
>> news:#1tWii\$6EHA.3828@TK2MSFTNGP09.phx.gbl...
>> > I want to update MultiDatabase with Microsoft's MSDTC.Just like ADO.Net
>> > provided,
>> > I designed an interface below:
>> > public interface IDistributeTransaction
>> > {
>> > int StartTransaction();

```
>>>
>>> int CommitTransaction();
>>>
>>> int RollBackTranstion();
>>>
>>> int ExecuteSql(string connectionstring,string SqlStr,out string
> ErrStr);
>>> }
>>>
>>> I called the Interface by Remoting.
>>>
>>> int rt = m_DistributeTrans.StartTransaction();
>>> if(rt == 0)//sucess
>>> {
>>> rt = ExecuteSql(...);//Update A database;
>>> if(rt == 0)//sucess
>>> {
>>> rt = m_DistributeTrans.ExecuteSql(...);//Update B database
>>> }
>>>
>>> if(rt == 0)//sucess
>>> m_DistributeTrans.CommitTransaction();//Commit;
>>> else //failed
>>> DistributeTrans.RollBackTranstion();//Rollback;
>>> }
>>> But when Rollback,the A database is updated sucessful. The
>>> ContextUtil.TransactionId is the same during the whole Transaction. Why
>>> this
>>> happened?
>>>
>>>
>>
>
>
```