

شش مرحله برای ایجاد WCFTransactions در WCF مقدمه و هدف:

هدف از مطلب فوق اجرا نمودن عملیات Insert ، Update و غیره... بوسیله چندین Connection در یک Transaction در زمان اجرای سرویسهای WCF میباشد. برای پیاده سازی و شرح Transaction ، سه پروژه ایجاد می‌نماییم. دو پروژه WCF سرویس و یک پروژه Client ، هر سه پروژه را در یک Solution به نام WCFTransaction اضافه می‌نماییم. در هر دو پروژه WCF بطور جداگانه Connection روی Database ایجاد می‌نماییم. سپس سعی می‌کنیم بوسیله Transaction عملیات Insert هر دو Service را کنترل نماییم. بطوریکه اگر یکی از Service ها در زمان عملیات Insert دچار مشکل شود. دیگری نیز Commit نگردد. به عبارتی در قدیم نمی‌توانستیم بیش از یک Connection در یک Transaction ایجاد نماییم. اما بوسیله Transactionscope ، انجام عملیات ، Insert ، Update و غیره... بوسیله چندین Connection به یک Database بطور همزمان در یک Transaction فراهم شده است. برای نمایش دادن عملیات Rollback نیز، به عمد خطایی ایجاد می‌کنیم، تا نحوه Rollback شدن در Transaction را مشاهده نماییم.

سعی شده است پیاده سازی و استفاده از Transaction در شش مرحله انجام شود.

مرحله اول: ایجاد دو پروژه WCFService و یک پروژه Client جهت فراخوانی (Call) کردن سرویسها

در این مرحله همانطور که از قبل نیز توضیح داده شده است، دو پروژه WCF به نامهای WCFService1 و WCFService2 ایجاد شده است و یک پروژه Client به نام WCFTransactions نیز ایجاد می‌کنیم.



مرحله دوم: افزودن Attribute ی به نام TransactionFlow به Interface سرویسها.

در این مرحله در Interface هریک از سرویس‌ها متد جدیدی به نام UpdateData اضافه می‌نماییم. که عملیات Insert into Database را انجام می‌دهد. حال بالای متد UpdateData از صفت TransactionFlow استفاده می‌نماییم. تا قابلیت Transaction برای متد فوق فعال گردد و متد فوق اجازه می‌یابد از Transaction استفاده نماید.

```
<ServiceContract(>> _
Public Interface IService1

    <OperationContract(>> _
    Function GetData(ByVal value As Integer) As String
```

```

<OperationContract(>> _
Function GetDataUsingDataContract(ByVal composite As CompositeType) As CompositeType

<OperationContract(>> _
<TransactionFlow(TransactionFlowOption.Allowed)> _
Sub UpdateData()

End Interface

```

مرحله سوم:

در این مرحله متد **UpdateData** را پیاده سازی می‌نماییم. بطوریکه یک **Insert Into** ساده در **Database** انجام می‌دهیم. و بالای متد فوق نیز کد زیر را می‌افزاییم.

```

<OperationBehavior(TransactionScopeRequired:=True)>

```

کد متد **UpdateData**

```

<OperationBehavior(TransactionScopeRequired:=True)>
Public Sub UpdateData() Implements IService1.UpdateData
Dim objConnection As SqlConnection = New SqlConnection(strConnection)
objConnection.Open()
Dim objCommand As SqlCommand = New SqlCommand("insert into T(ID, Age) values(10,10)",
objConnection)
objCommand.ExecuteNonQuery()
objConnection.Close()
End Sub

```

مرحله دوم و سوم را برای **Service** دوم نیز تکرار می‌نماییم.

مرحله چهارم:

در این مرحله **TransactionFlow** را در **Web.Config** دو سرویس فعال می‌نماییم. تا قابلیت استفاده از **TransactionFlow** برای سرویسها نیز فعال گردد. نحوه فعال نمودن بصورت زیر میباشد:

برای **WcfService1** خواهیم داشت:

```

<bindings>
    <wsHttpBinding>
        <binding name="TransactionalBind" transactionFlow="true"/>
    </wsHttpBinding>
</bindings>

```

و در ادامه داریم:

```

<endpoint address="" binding="wsHttpBinding"
bindingConfiguration="TransactionalBind"
contract="WcfService1.IService1">

```

برای **WcfService2** نیز خواهیم داشت:

```

<bindings>
    <wsHttpBinding>
        <binding name="TransactionalBind" transactionFlow="true"/>
    </wsHttpBinding>
</bindings>

```

و در ادامه داریم:

```
<endpoint address="" binding="wsHttpBinding"
bindingConfiguration="TransactionalBind"
contract="WcfService2.IService1">
```

مرحله پنجم:

در این مرحله دو سرویس فوق را به پروژۀ **WCFTransactions** اضافه نموده و قطعه کد زیر را درون فرم **Load** می‌نویسیم.

```
Private Sub frmmain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

    Using ts As New TransactionScope(TransactionScopeOption.Required)
        Try
            Dim obj As ServiceReference1.Service1Client = New ServiceReference1.Service1Client()
            obj.UpdateData()
            Dim obj1 As ServiceReference2.Service1Client = New ServiceReference2.Service1Client()
            obj1.UpdateData()
            ts.Complete()

            Catch ex As Exception
                ts.Dispose()
            End Try
        End Using
    End Sub
```

پس از اجرای برنامه دو رکورد در جدول درج خواهد شد.

مرحله ششم:

حال برای **RollBack** کردن کل عملیات و مشاهده آنها کافیه در یکی از متدهای **UpdateData** یک **Throw Exception** ایجاد نماییم.

سعی می‌کنیم با کمی تغییر در متد **UpdateData** در **WCFService2**، خطایی ایجاد شود، تا نحوه **RollBack** را مشاهده نماییم.

```
Public Sub UpdateData() Implements IService1.UpdateData
```

()Throw New Exception

```
Dim objConnection As SqlConnection = New SqlConnection(strConnection)
objConnection.Open()
Dim objCommand As SqlCommand = New SqlCommand("insert into T(ID,Age) values(101,101)",
objConnection)
objCommand.ExecuteNonQuery()
objConnection.Close()
End Sub
```

فقط کد زیر به متد **UpdateData** اضافه شده است:

```
Throw New Exception()
```

و در رویداد **Load** فرم نیز پیاده سازی آن بشکل زیر خواهد بود:

```
Using ts As New TransactionScope(TransactionScopeOption.Required)
    Try
        Dim obj As ServiceReference1.Service1Client = New ServiceReference1.Service1Client()
        obj.UpdateData()
        Throw New Exception("There was Error")
        Dim obj1 As ServiceReference2.Service1Client = New ServiceReference2.Service1Client()
        obj1.UpdateData()
        ts.Complete()

    Catch ex As Exception
        ts.Dispose()
    End Try
End Using
```

وقتی برنامه را اجرا نمایید، مشاهده می‌کنید که هیچ رکوردی درون دیتابیس درج نشده است.

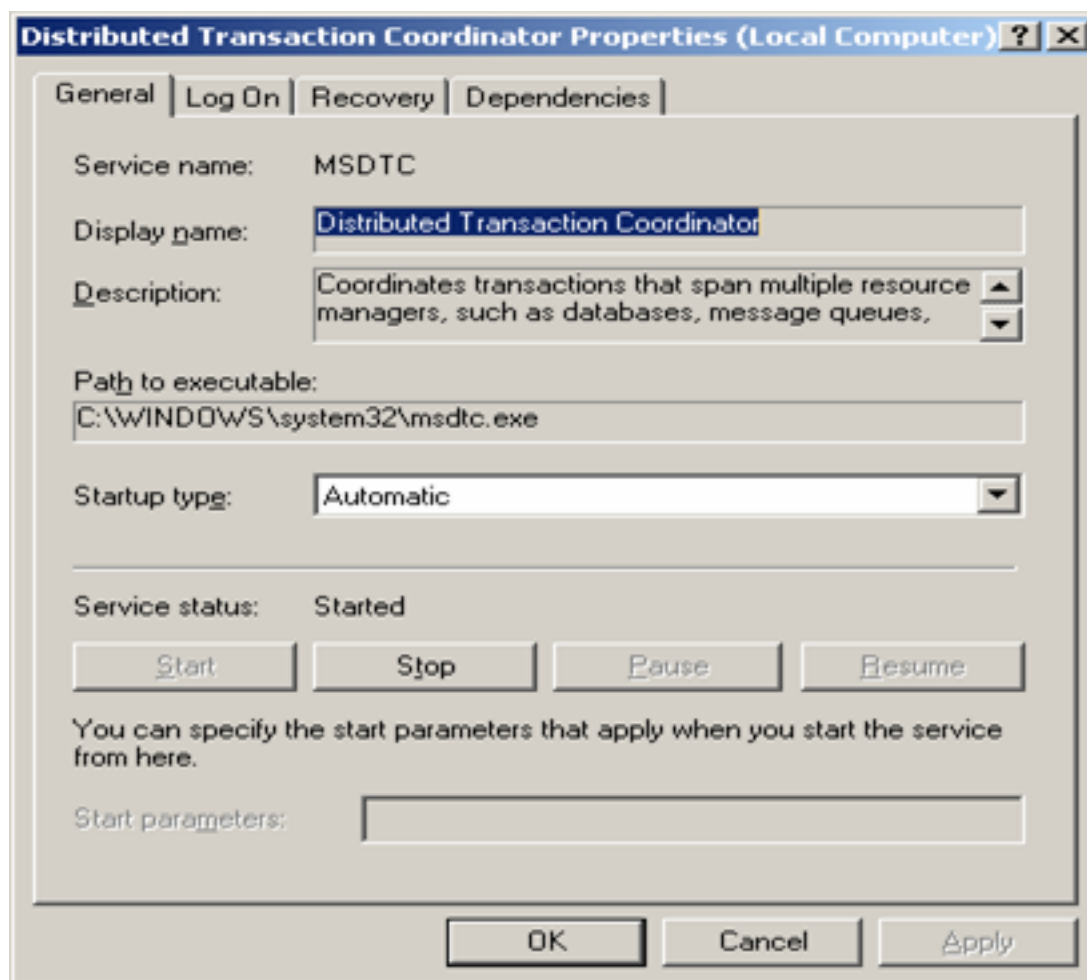
بسیار مهم: برای اینکه بتوانید بصورت **Distributed** عملیات **Transaction** را انجام دهید می‌بایست تنظیماتی را روی سرور که دیتابیس و سرویسها و کامپیوتر کلاینت انجام دهید که بصورت زیر می‌باشد:

نحوه تنظیم:

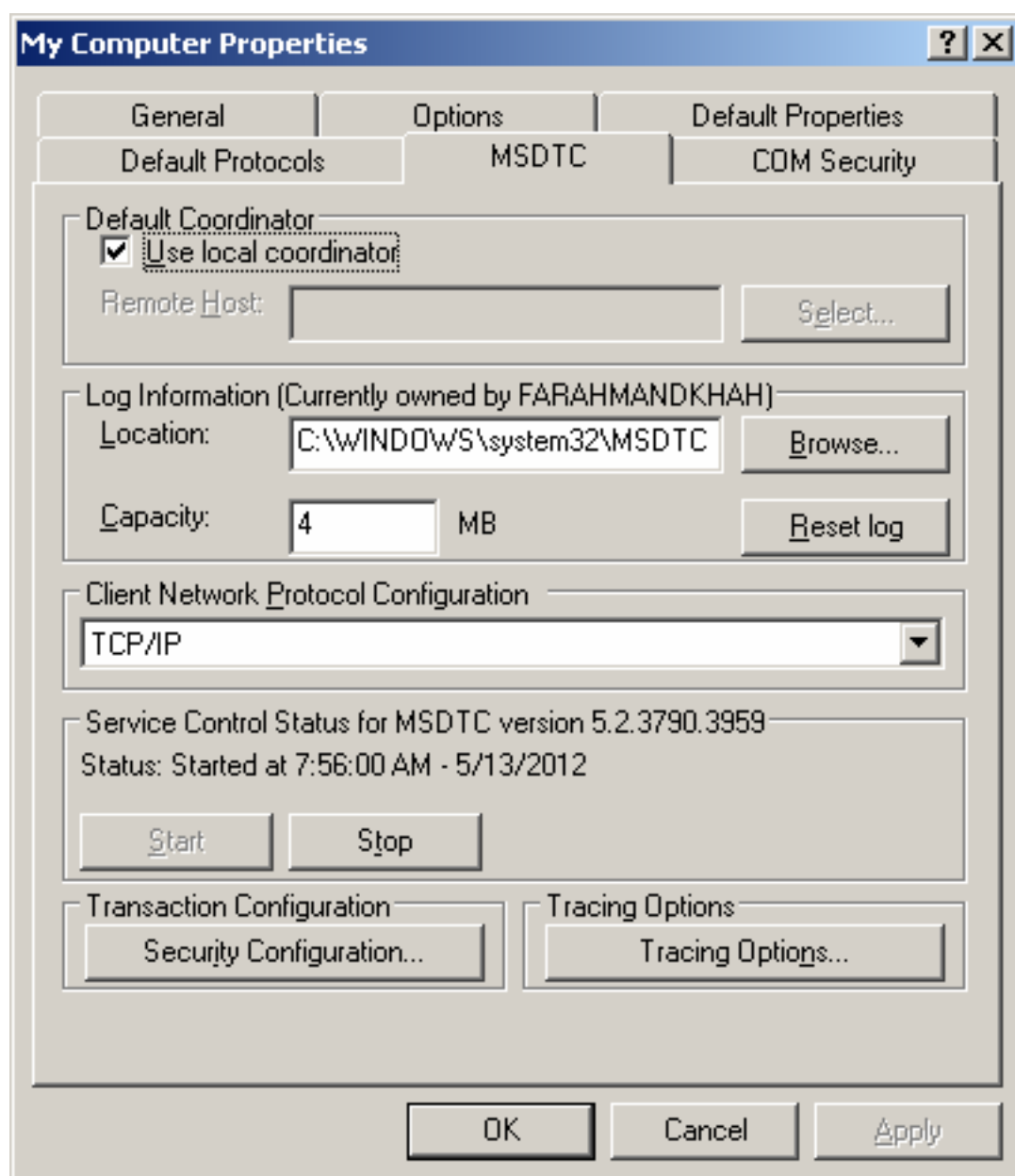
1- سرویس **Distribute Transaction Coordinator** را روی هر دو **Server**های **Database** ، **WCFService** و کامپیوتر کلاینت، **Start** می‌نماییم.

البته در شرایطی که **Service**های **WCF** و برنامه **Client** و **Database** روی یک سیستم باشد، تنظیمات فوق فقط روی همان سیستم انجام می‌شود.

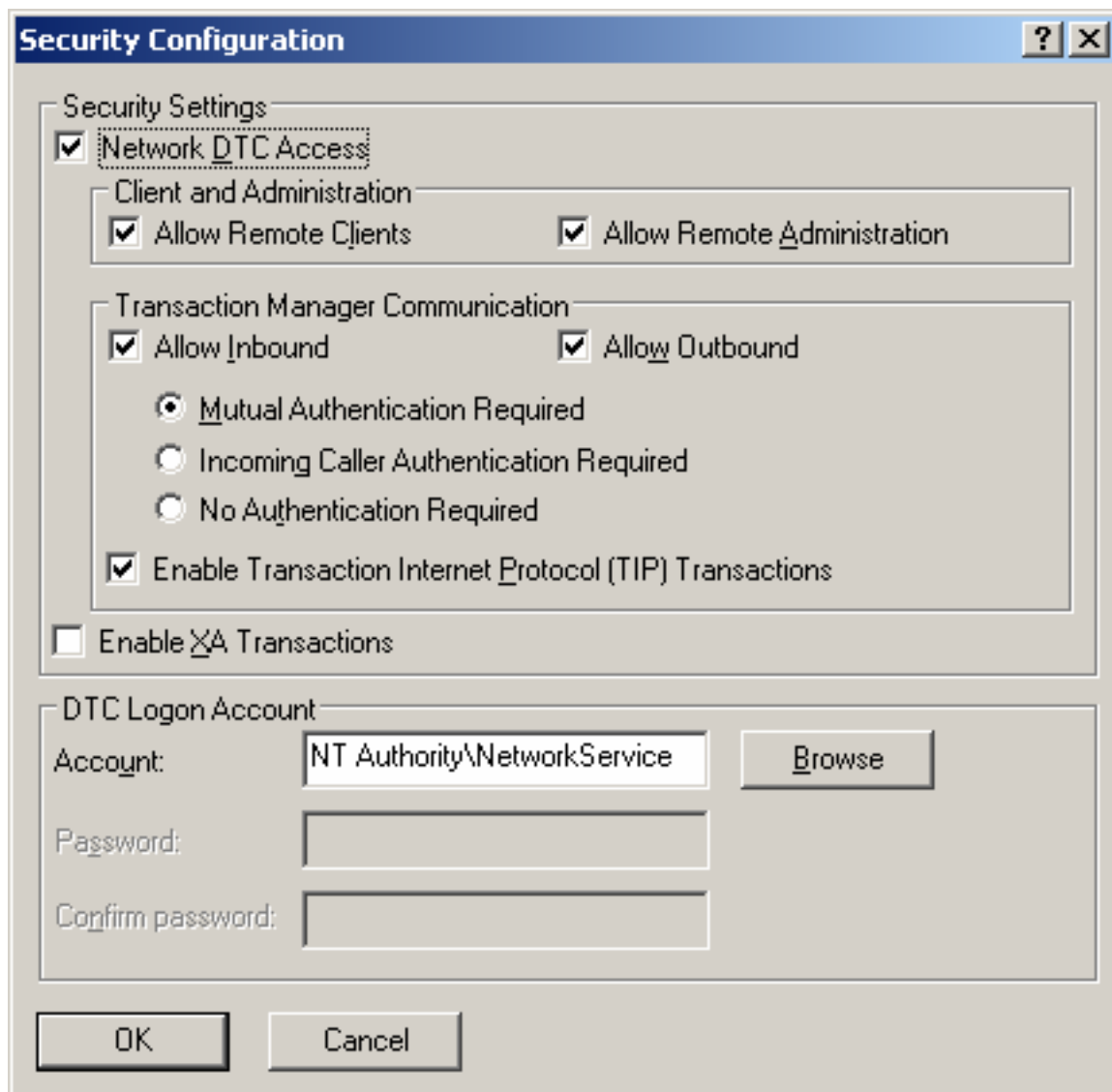
برای دسترسی به قسمت **Service** های **Windows** ابتدا **Administrative Tools** و سپس **Service** را باز نمایید و روی **Start** کلیک کنید.



2- در ادامه روی MY Computer کلیک راست نموده و تب MSDTC را انتخاب نمایید:



در ادامه روی [Security Configuration](#) کلیک نمایید. تا فرم زیر نمایش داده شود.



مطمئن شوید که آیتمهای زیر انتخاب شده باشند:

• [Network DTC Access](#)

• [Allow Remote Clients](#)

• [Allow Inbound](#)

• [Allow Outbound](#)

• [Enable Transaction Internet Protocol\(TIP\) Transactions](#)

سپس با OK کردن Service، سرویس بطور خودکار Restart می‌شود.

در ضمن اگر از SQL Server 2000 استفاده می‌نمایید، لازم است تنظیم زیر را انجام دهید.

روی SQL Server Service Manager کلیک نموده و کامبوی Service را Dropdown نمایید و [Distribute Transaction Coordinator](#) را انتخاب کنید. اما برای ورژن‌های بالاتر از SQL Server 2000 نیاز به انتخاب [Distribute Transaction](#)

Coordinator نمی‌باشد.

امیدوارم مطلب فوق مفید واقع شود، چنانچه کم و کاستی مشاهده نمودید، اینجانب را از نظرات خود بهره مند سازید.

منبع:

<http://www.codeproject.com/Articles/38793/6-Steps-to-Enable-Transactions-in-WCF>