

ADO Connection Failure

Source: <http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.database.ado/2004-01/0344.html>

From: Keith Comley (*klcomley_at_ntlworld.com*)

Date: 01/20/04

Date: Tue, 20 Jan 2004 21:32:27 -0000

Hello,

The following post is by Todd Neuman on the elists support site, he is asking the same question as I am and as he puts it so eloquently I felt it inappropriate to re-invent the wheel. Unfortunately he didn't get a satisfactory response, his post dates from 2001

Thanks,

Keith Comley:

"Hi Everyone,

I'm curious what other people have done about broken connections using TADOConnection in D5 Enterprise. We are having an issue in our application where we use ADO 2.6 over TCP/IP to communicate with MS SQL 2000 data servers. The issue is if anything hiccups in your physical connection to the database ADOExpress does not reflect this change but rather generates a Connection Failure error message. Even after this error message the ADOConnection Status is still "Connected" erroneously.

I have been able to trap the error at the ADOCommand.Execute level and check the ADOConnection.Errors.Count and grab the message. I have even been able to then close the ADOConnection and reopen it successfully. The problem is that any queries that were previously open are no longer open once you re-establish the connection.

First Question – Does anyone know how to move this check to the ADOConnection Level? I have one ADOConnection object and 70 – 80 Queries/Commands, I'd like to solve it at the single ADOConnection level if possible. I tried the OnWillExecute, OnInformation, and OnError events in ADOConnection but they all pass successfully because they all execute prior to the actual execution of the ADOCommand.Execute attempting to complete and then failing.

Second Question – Does anyone know of a way to reestablish the ADOConnection without Closing all of the active Queries using that connection?

borland.public.delphi.database.ado: ADO Connection Failure

Thanks for all your help,
Todd"