

ADO ConnectionString

An vielen Stellen wird in meinen Code-Schnipseln über einen ConnectionString die Verbindung zur Datenbank aufgebaut, aber wie kommt man zu diesem ConnectionString? Natürlich kann man im Internet suchen oder einfach die von mir generierten nutzen oder man schreibt sich ein kleines Programmchen!

Dafür braucht wir die Importe der Type Libraries "Microsoft ActiveX Data Objects 2.x-Objektbibliothek" und "OLE DB Service Component 1.0-Typbibliothek" und das Package LazActiveX

Import ADODB : Microsoft ActiveX Data Objects 2.x-Objektbibliothek

C:\Program Files\Common Files\System\ado\msado15.dll  [adodb_6_1_tlb.pas \(/downloads/adodb_6_1_tlb.pas\)](#) (189,84 kByte) 22.03.2013 10:20 (189,84 KB)

Import MSDASC : OLE DB Service Component 1.0-Typbibliothek

C:\Program Files\Common Files\System\Ole DB\oledb32.dll  [msdasc_1_0_tlb.pas \(/downloads/msdasc_1_0_tlb.pas\)](#) (18,68 kByte) 01.12.2013 13:50 (18,68 KB)

 [ConnectionStr.lpr \(/code/ConnectionStr.lpr?mode=download\)](#) Pascal (666 bytes) 21.12.2013 17:25

```
program ConnectionStr;
{$APPTYPE CONSOLE}
{$mode objfpc}{$H+}
uses
  Interfaces, // used packages: LazActiveX
  adodb_6_1_tlb, // Microsoft ActiveX Data Objects 2.x-Objektbibliothek
  msdasc_1_0_tlb; // OLE DB Service Component 1.0-Typbibliothek

function getConnectionString: Wdestring;
var
  cn: _Connection;
  dl: DataLinks;
begin
  result := "";
  cn := CoConnection.Create;
  dl := CoDataLinks.Create;
  dl.hwnd:= 0;
  try
    cn := Connection(dl.PromptNew);
    if Assigned(cn) then
      result := cn.ConnectionString;
  finally
    dl := nil;
    cn := nil;
  end;
end;

begin
  WriteLn(getConnectionString());
  ReadLn;
end.
```

Autor: [Udo Schmal \(https://plus.google.com/107378996518617284564?rel=author\)](https://plus.google.com/107378996518617284564?rel=author), veröffentlicht: 19.12.2013, letzte Änderung: 03.02.2016

© Copyright 2018 Udo Schmal (/)