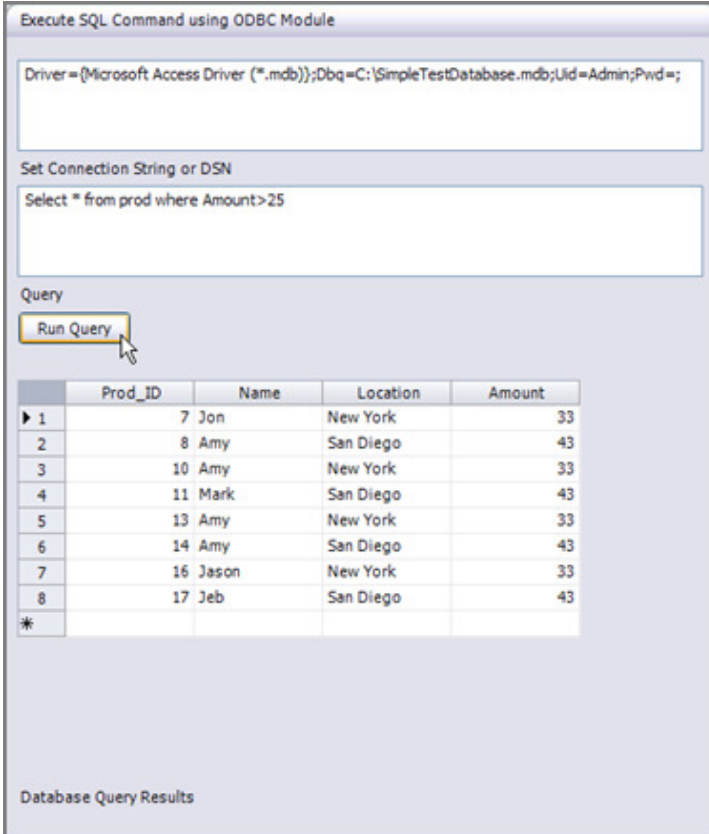


Connect to Databases

MiniTutorial



Execute SQL Command using ODBC Module

Driver={Microsoft Access Driver (*.mdb)};Dbq=C:\SimpleTestDatabase.mdb;Uid=Admin;Pwd=;

Set Connection String or DSN

Select * from prod where Amount>25

Query

Run Query

	Prod_ID	Name	Location	Amount
▶ 1	7	Jon	New York	33
2	8	Amy	San Diego	43
3	10	Amy	New York	33
4	11	Mark	San Diego	43
5	13	Amy	New York	33
6	14	Amy	San Diego	43
7	16	Jason	New York	33
8	17	Jeb	San Diego	43
*				

Database Query Results

In this MiniTutorial, you will learn how to connect to a database using the ODBC primitive module as well as VBA.

This tutorial should take 15-30 minutes, to complete.

MiniTutorial Overview

In this MiniTutorial, you will learn how to connect to databases using the ODBC primitive module and VBA. You will:

- Learn about the ODBC module to read data from ODBC data sources.
- Learn about to basic Components that use VBA to read and write to databases.
- Examine the VBA scripts to learn how to apply them in new ways on your own.

Before you start this tutorial, you will need to have installed Proto Financial Version 1.8 and downloaded the tutorial files from the Proto website. Tutorial files can be downloaded in a zipped format from Proto's website at

<http://www.protosw.com/devcenter>

This MiniTutorial also uses the **Northwind Traders Sample Database**, which is available for download from Microsoft at:

<http://www.microsoft.com/downloads/thankyou.aspx?familyId=c6661372-8dbe-422b-8676-c632d66c529c&displayLang=en#>

Run the executable from Microsoft and save the Northwind Traders Sample Database to: C:\Nwind.mdb

Before you begin this MiniTutorial, save a copy of the associated Proto file to work with.

- 1 Locate the folder named "MiniTutorial – Connect to Databases" in the folder titled "Proto Getting Started Lessons."
- 2 Open the file named "Database Connections.proto" by double-clicking on the file.
- 3 Save this Proto dashboard with a new name in the "MiniTutorial – Connect to Databases" folder. To do so, select File > Save As from the application menu bar. Then browse to the appropriate location and save the file as "Database Connections - Copy.proto."

A Note about Database Connection Strings

There are a number of different ways to connect to a database through VBA. However, we will focus on the ODBC module which uses ODBC and the VBA samples which use ADODB. Database connection strings and queries can be tricky, and this MiniTutorial assumes that the reader has familiarity with databases, connection strings, SQL commands and the syntactical issues that can come up.

For a good connection string reference, try this website:

<http://www.connectionstrings.com/>

Using the ODBC Primitive Module

The ODBC primitive module imports data from ODBC data sources into Proto as Entrysets. It provides read access to any ODBC compatible data source using standard connection string and query commands. Given a connection string and query, the ODBC Module performs the query on the target data source and outputs the results in an Entryset on the "Get Data" output connector.

In this section, you will use a Component with simple controls connected to use the ODBC module, and you will learn how to configure the ODBC module on your own.

The ODBC Component in the MiniTutorial Proto File

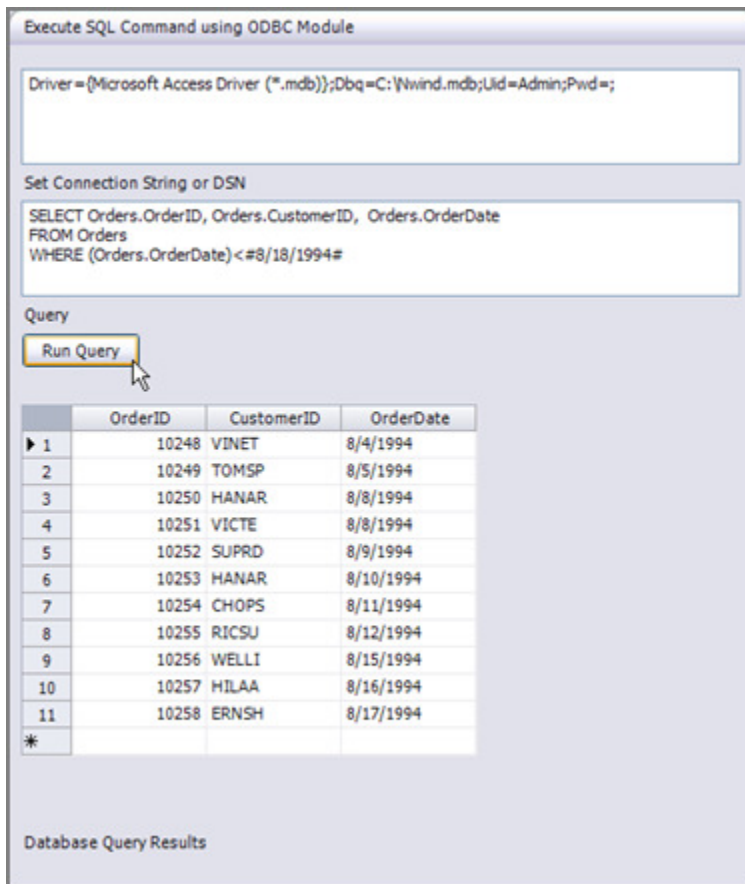
In the Proto file included with this MiniTutorial there is a Component with a simple preconfigured interface to use the ODBC primitive module. This Component can also be found in the Getting Started Components folder under the directory:

“..\Get Data\Database Connections”

This Component allows you to enter a connection string and query string, and then execute the command to generate an Entryset. If you open up and explore this Component as learned in Lesson 1 and Lesson 2, you will see that it is simply a number of interface controls used to configure and trigger the ODBC module.

You can use this Component as is, or you can edit it to alter how it works. For example, you can hide certain controls to only leave the button to refresh the query. Or make new Connections to the “Do Query” input from the same Textbox that sets the query string, to automatically retrigger the request whenever the query is changed.

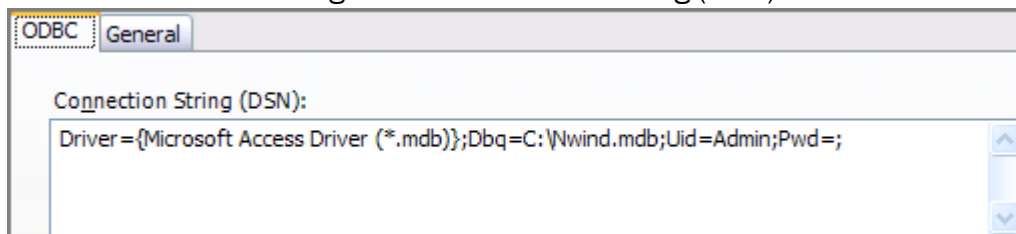
The “Execute SQL Command using ODBC Module” Component is shown below.



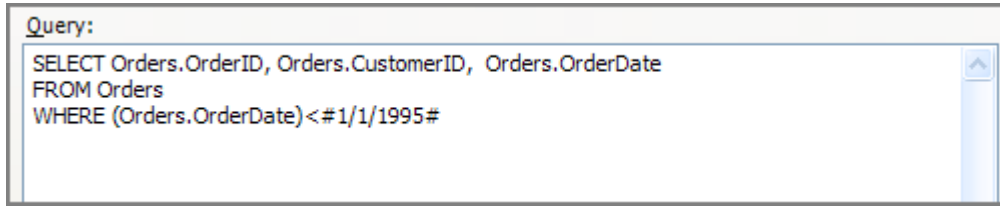
Configuring the ODBC Primitive Module Using the Property Panel

The ODBC module's Property Panel looks very similar to the Component shown above. To configure the ODBC primitive module, go to its Property Panel and enter a connection string and a query.

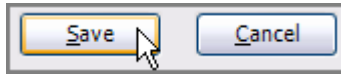
- 1 Set the connection string in the "Connection String (DSN):" section.



- 2 Set the query in the “Query:” section.



- 3 And then click “Save” to close the Property Panel and run the query.

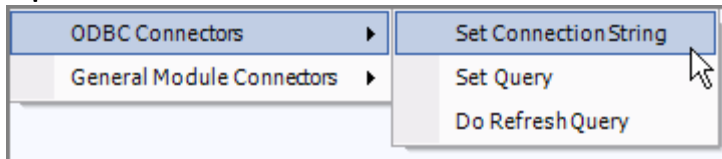


It is somewhat more common to use the ODBC module through it's inputs though, as discussed in the next section.

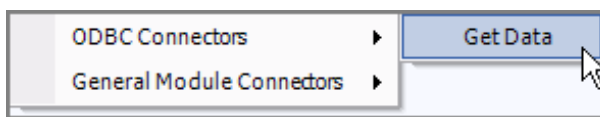
Configuring the ODBC Primitive Module Using Connections

The ODBC primitive module has very few inputs and outputs. To function, you must set the connection string and the query string, and then send an input to trigger the module to query the database. The only output is the resulting Entryset.

Inputs



Outputs



You can make Connections from other modules or Components on the Builder to set the ODBC module's inputs. The Component explored above illustrates one such configuration. But you could use other modules to assemble a query string based on what a user selects in a Listbox, and then use that query string to set both the “Query” input and the “Do Refresh Query” input of the ODBC module. This would have the effect of requerying a database whenever the end user makes a new selection in a Listbox.

Using VBA to Connect to Databases

Using VBA, you can not only read from a database, but also insert an Entryset, update rows, and generally perform any other combination of task in a controlled procedural manner. To use VBA to interface with a database beyond the functionality provided in the

Getting Started Components, you will need to have some experience working with databases through VBA. For an introduction to using VBA within Proto, please reference the tutorial called “MiniTutorial - Introduction to Using VBA in Proto.”

There are two Components included in the “Database Connections.proto” file that you can use and edit to fit your particular needs.

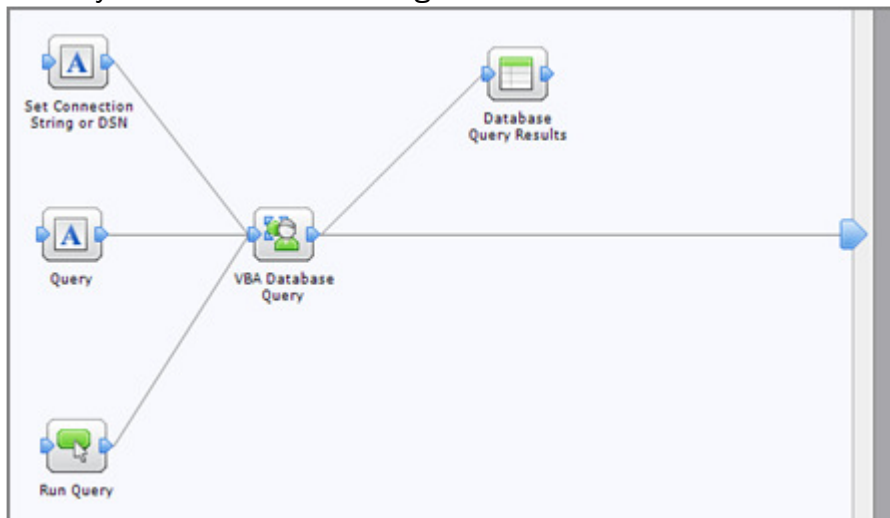
VBA - Query Database using ADODB

The first Component illustrates how to query a database through an ADODB.Connection object and then build an Entryset with the Recordset returned.

The interface in the Component is identical to the interface explored in the “Execute SQL Command using ODBC Module” Component covered in the first section of this tutorial.

To explore the VBA code running the query and building the Entryset, follow these steps.

- 1 Go to the Builder and open up the Component called “VBA - Insert Entryset into Database using ADODB” by right-clicking on it and selecting “Open” from the menu.
- 2 Inside you will see the following modules connected to another Component.



- 3 Open the Component called “VBA Database Query” and inside you will see a single VBA primitive module. This is yet another example of encapsulating a VBA script within a Component and using it via inputs and outputs on the Builder.
- 4 Open the VBA module Property Panel and then click the “Visual Basic Editor...” button to launch the VBA IDE.
- 5 Inspect the VBA script, noting that the script will only be executed when the “Do Query” input is triggered.

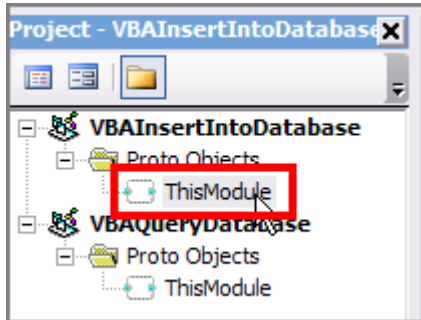
```
' Exit immediately if no input was received on the trigger
If Not (HasBeenTriggered("Do Query", TriggeredInputs)) Then
    GoTo ExitSub
End If
```

VBA - Insert Entryset into Database using ADODB

The second Component illustrates how to insert rows from an Entryset into a database through an ADODB.Connection object.

To explore this Component, follow the same sequence of steps to locate and launch the VBA IDE as you did for “VBA - Query Database using ADODB” Component.

Or, if you already have the VBA IDE open, you can explore the code directly by clicking on the appropriate object in the project browser as shown below.



To use this Component, you will have to create an Entryset with Attribute names that exactly match a subset of the column names in the database table (order does not matter). You do not have to include every column if the database table schema permits empty values for the columns in question.

- To learn how to define an Entryset structure in the Table module, please refer to Lesson 2. As a brief reminder, go to the Table’s Property Panel, click the button titled “Define Entryset...” and create an Entryset structure.
- Alternatively, you may use any Entryset so long as you rename the columns prior to sending it to the “VBA - Insert Entryset into Database using ADODB” Component. For example, you may have a user enrich data from a CSV file, perform calculation and filter results to create a final list of rows to insert. Then use an Attribute Editor module (covered in detail during Lesson 2) to prepare the Entryset structure for insertion.

Thank You!

Thank you for your interest in Proto. If you have feedback regarding this tutorial or would like to see more training material covering specific topics or questions, please email us with suggestions at: devcenter@protosw.com