

Maxum ODBC 3.11

Table of Contents

Installing the Polyhedra ODBC driver	2
Using ODBC with the Maxum Database.....	2
Microsoft Access 2000 Example.....	2
Access Example (Prior to 2000):.....	5
Simple Microsoft Excel Example.....	10
Microsoft Excel Example with VBA functions.....	14
Microsoft Visual Basic 6.0 Example	20
A More Complicated Microsoft Visual Basic 6.0 Example	29
Editing Maxum tables with Visual Basic 6.0	32
VB6 Code Sample for building connection string.....	33

The purpose of this document is to give examples of different ways to use ODBC with the Maxum. Many different vendors offer ODBC connectivity from within their products. We only give a small number of examples for products that most customers will have available. It is assumed that the ODBC driver will work with other products. To use ODBC, a knowledge of relational database structure and the Maxum's database is required.

ODBC means Open Database Connectivity. It allows Windows applications to access a variety of different databases using a common interface. In order to use ODBC, the database vendor, Polyhedra in our case, must provide an ODBC driver that is then installed and configured on the local workstation.

Installing the Polyhedra ODBC driver

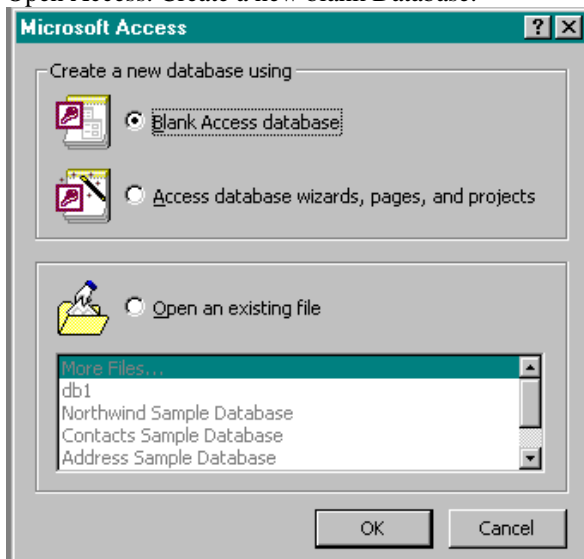
For Windows 98 SE, Windows 2000, Windows Millennium, and Windows NT, ODBC should be available under the Control Panel. If it is not, Run the DataAcc.exe from the Advance System Manager\Odbc\samples\ODBCsamples. Go to Advance System Manager\Odbc\odbc directory and execute the Setup. Once the driver is installed, Data Sources must be defined. The method for doing this differs based on the type of Windows application to be used. Follow the instructions under the following topics to proceed. Go to Control Panel>ODBC Data Sources or 32 ODBC to view and add ODBC drivers and data sources.

Using ODBC with the Maxum Database

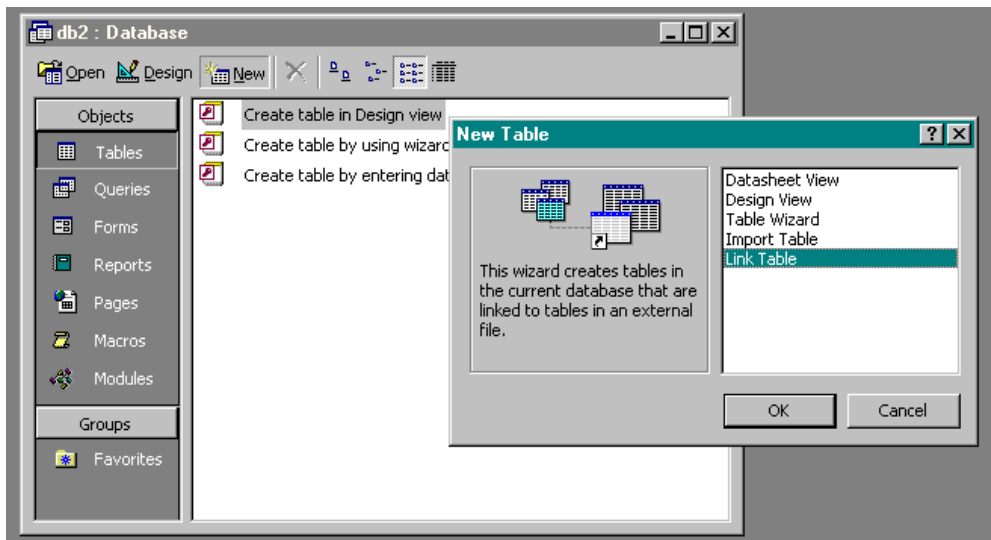
To use the ODBC driver to access data from the Maxum Database, it is necessary to have a working knowledge of the Maxum tables. Consult the Maxum Tables document for a complete listing of these tables and a short description of the data contained in each table. Pay particular attention to the fact that the Maxum database is object-oriented – each table contains a script, portions of which may be executed when values change in the database. Care must be taken when modifying values in the database. Depending on which tool uses ODBC, a knowledge of Structured Query Language(SQL) may be required. Try this site for an SQL tutorial: <http://www.sqlcourse.com/>. Note that the Maxum's SQL is not a full version, i.e., not all commands work.

Microsoft Access 2000 Example

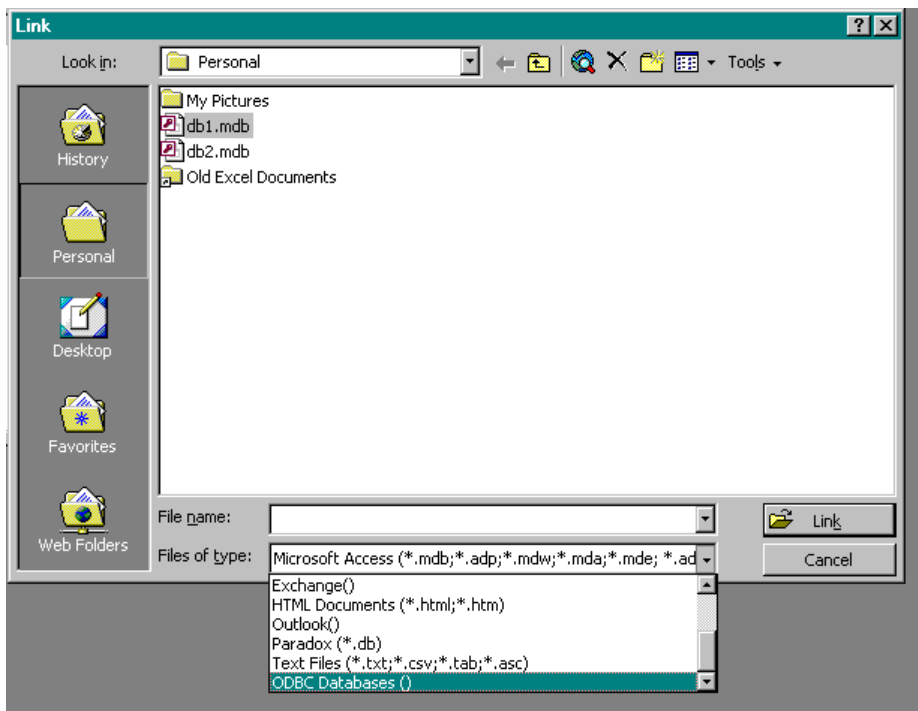
Open Access. Create a new blank Database.



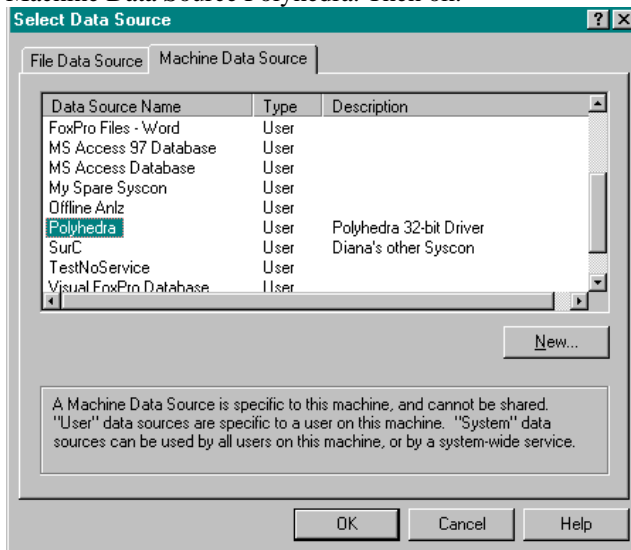
Select New and import or link table:



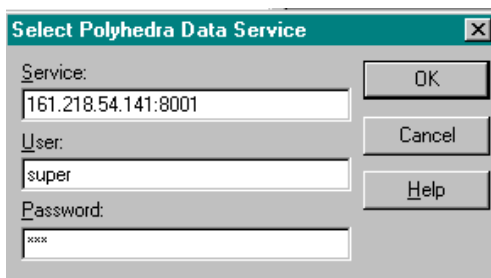
Select ODBC:



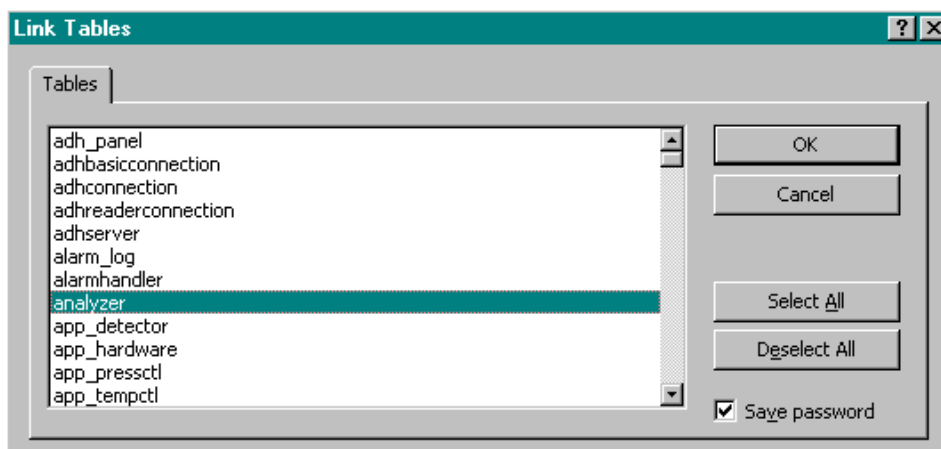
Machine Data Source Polyhedra. Then ok.



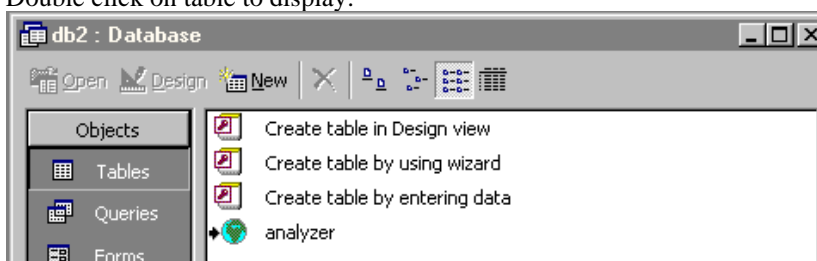
Enter ip address, user id, and password:



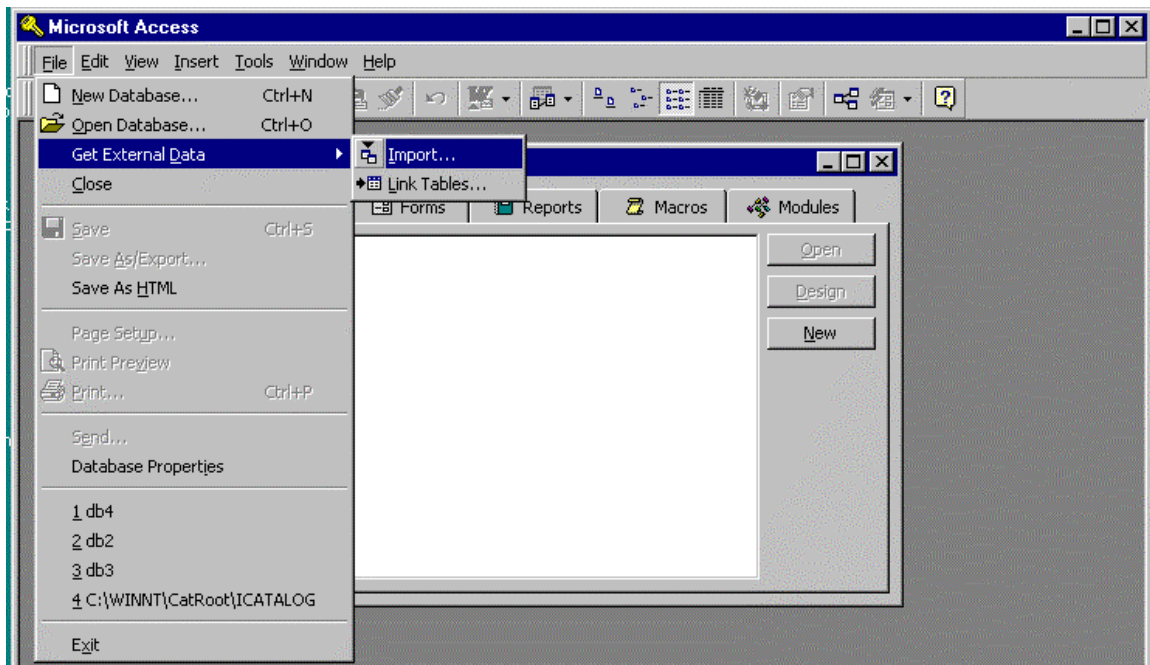
Select tables and be sure to save password:



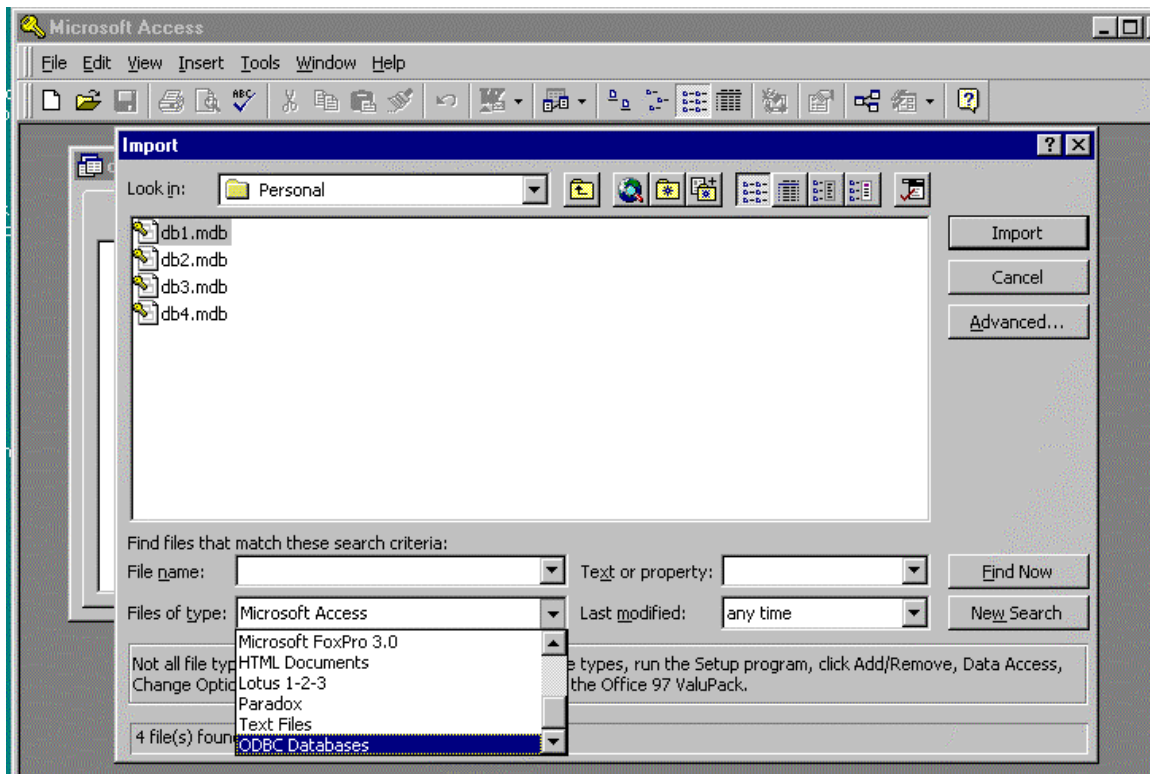
Double click on table to display:



Access Example (Prior to 2000):

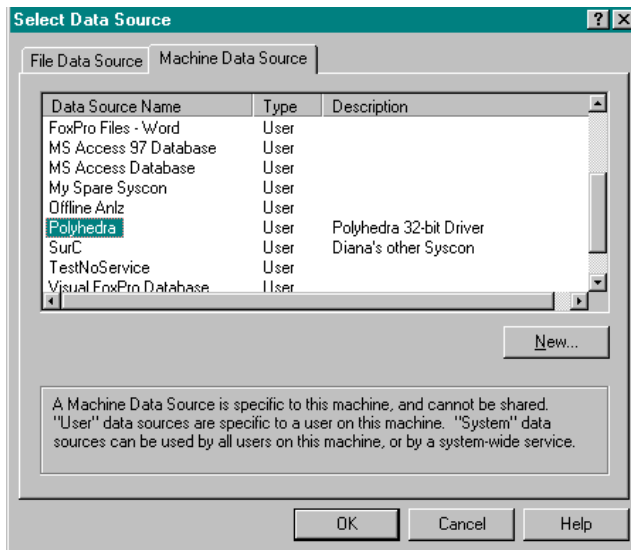


To read from the Maxum tables, select Import. To read and write, select Link Tables.

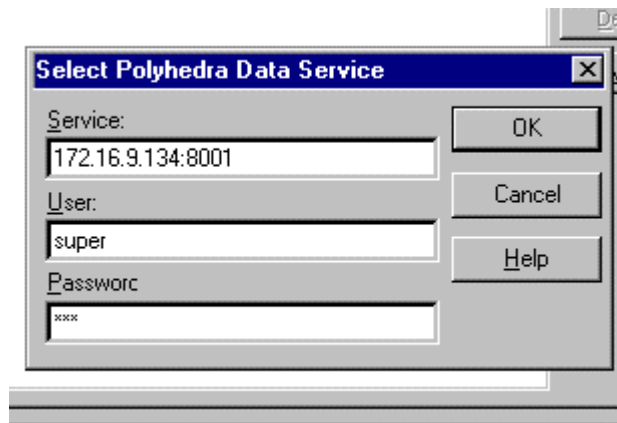


Select ODBC Databases for Files of type.

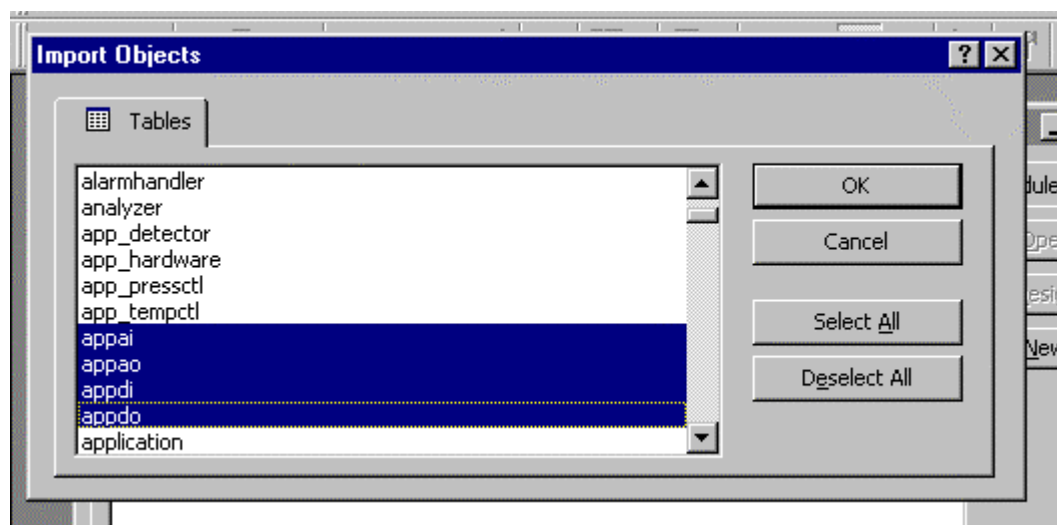
Select the Polyhedra datasource. Click OK.



Type in the ip address, user, and password. Click OK.



Select table(s) that are to be read. Click OK.

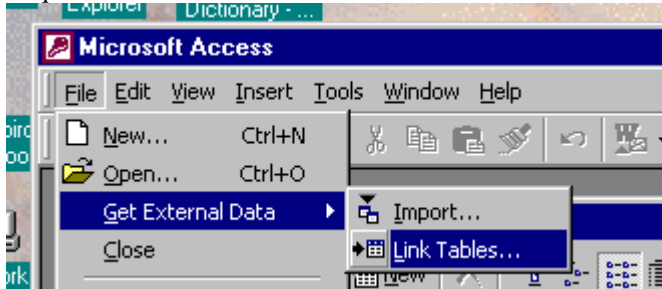


By opening each table, you will be able to read the attributes:

appai : Table							
application_id	id	name	io_status	enable	hrdwr_id	uni	
100	100	TCD L1 Meas	-1	0	11:4-6.1-2.1.1	Volts	
100	101	TCD L1 BalSig	-1	0	11:4-6.1-2.1.145	%	
100	110	TCD U1 Meas	-1	0	11:4-6.1-2.1.17	Volts	
100	111	TCD U1 BalSig	-1	0	11:4-6.1-2.1.161	%	
100	120	TCD L2 Meas	-1	0	11:4-6.1-3.1.1	Volts	
100	121	TCD L2 BalSig	-1	0	11:4-6.1-3.1.145	%	
100	130	TCD U2 Meas	-1	0	11:4-6.1-3.1.17	Volts	
100	131	TCD U2 BalSig	-1	0	11:4-6.1-3.1.161	%	
100	140	TCD L3 Meas	-1	0	11:4-6.1-4.1.1	Volts	
100	141	TCD L3 BalSig	-1	0	11:4-6.1-4.1.145	%	
100	150	TCD U3 Meas	-1	0	11:4-6.1-4.1.17	Volts	
100	151	TCD U3 BalSig	-1	0	11:4-6.1-4.1.161	%	
101	100	FID Meas	-1	0	11:4-5.2-2.1.1	Volts	
101	101	FID BalSig	-1	0	11:4-5.2-2.1.145	%	
101	110	FIL Meas	-1	0	11:4-5.2-2.1.17	Volts	
101	111	FIL BalSig	-1	0	11:4-5.2-2.1.161	%	
102	100	TCD L1 Meas	-1	0	11:4-6.3-2.1.1	Volts	
102	101	TCD L1 BalSig	-1	0	11:4-6.3-2.1.145	%	
102	110	TCD U1 Meas	-1	0	11:4-6.3-2.1.17	Volts	
102	111	TCD U1 BalSig	-1	0	11:4-6.3-2.1.161	%	

Although you can change data value here, this will NOT change the database. A linked table will allow editing.

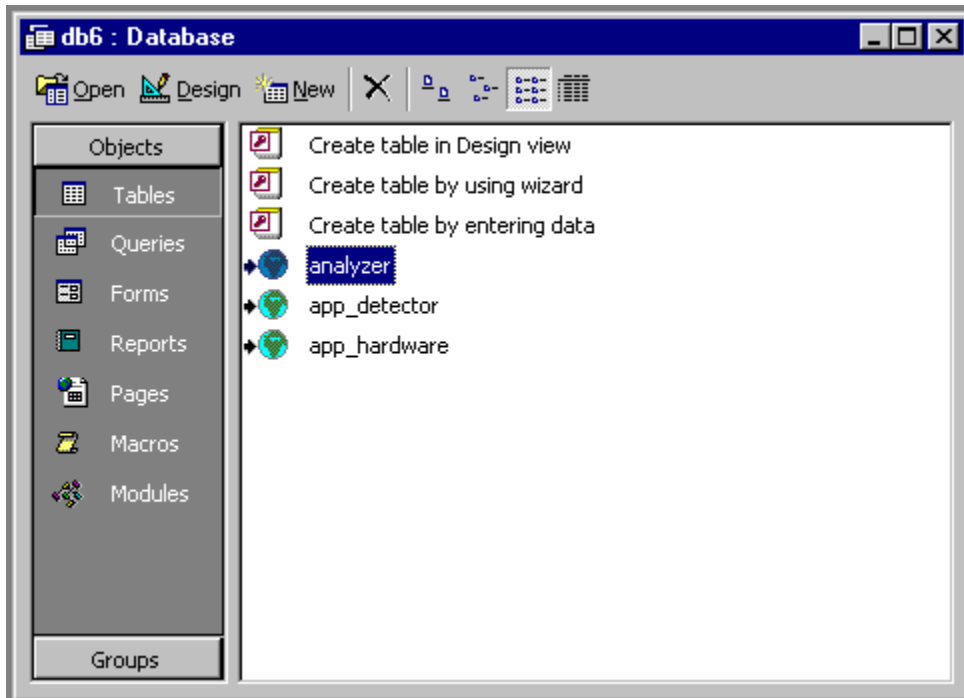
Request that external data be linked:



Select Tables(be sure to save the password):



Tables are now linked to the database. Double click on the table to view the contents:

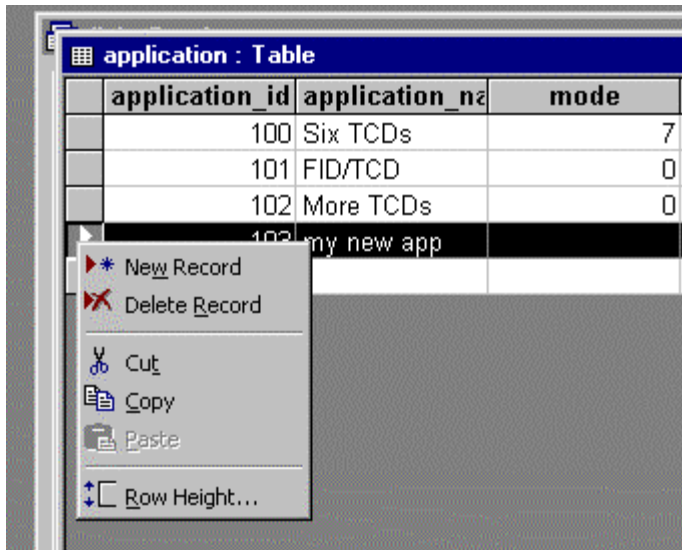


By selecting the application table, I can both read and write into that table.

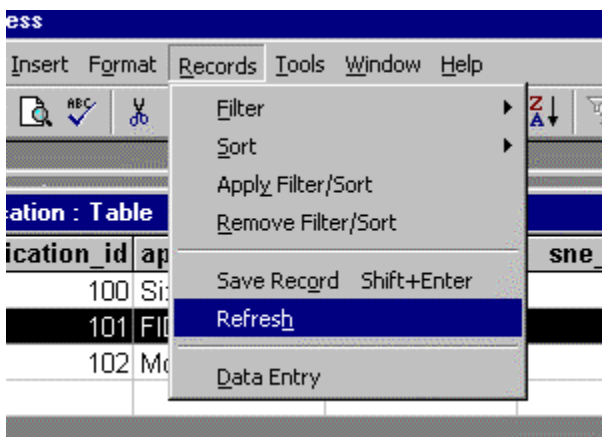
Write values to the database by entering values and then exiting the record. To add a new record, click on the * at the bottom and enter values.

	application_id	application_name	mode	sne_mode	active_app	uen	paused
	100	Six TCDs	7	0			
	101	FID/TCD	0	5			
	102	More TCDs	0	5			
	103	my new app					
*							

To delete a record, right click on the entry that is to be deleted.

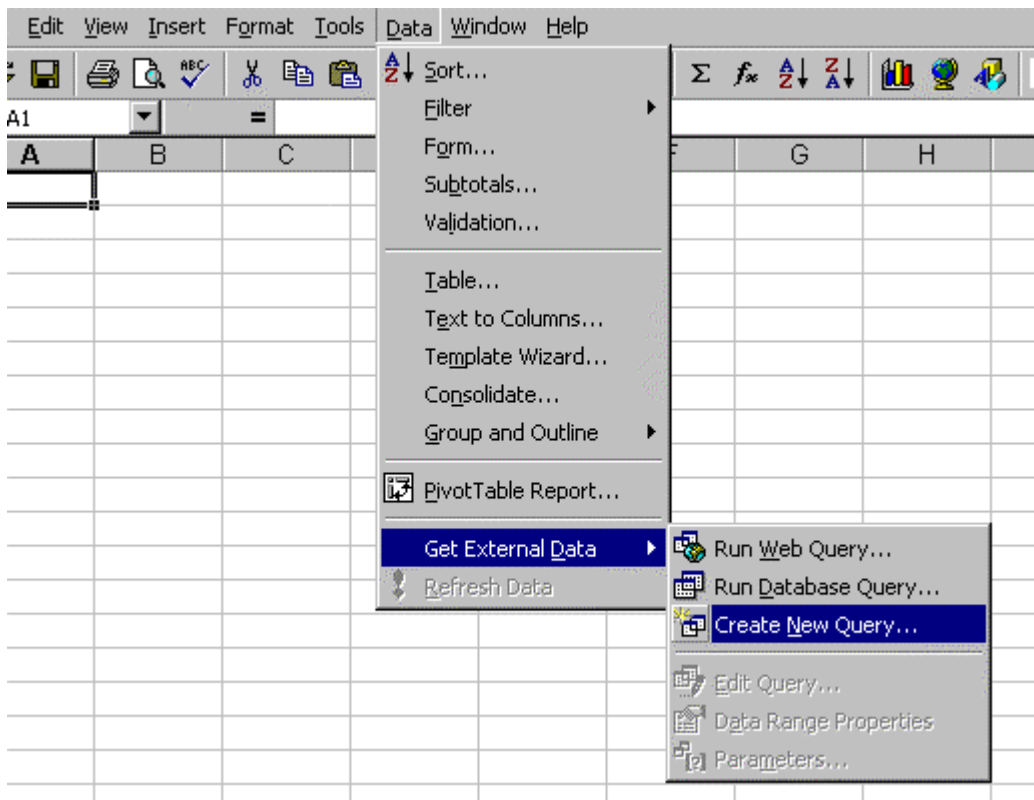


Note that as you modify values, your view of the table is not dynamic, i.e, indirect changes will not be apparent. To refresh the table view:

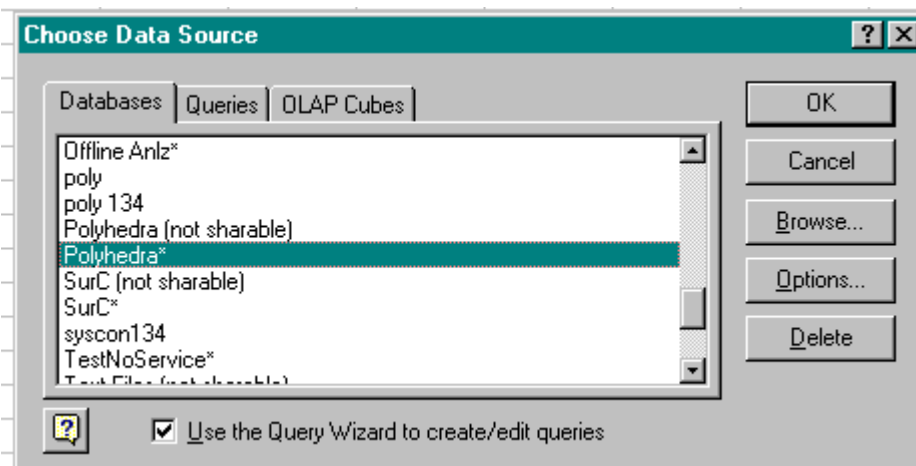


Simple Microsoft Excel Example

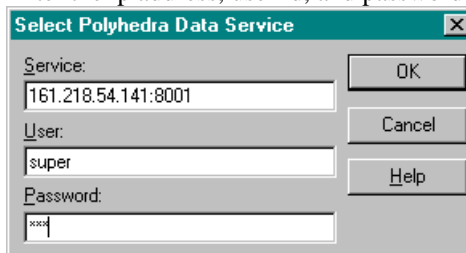
Excel 97/2000 uses MS Query to extract data using ODBC.
Make an Excel Spreadsheet and select :



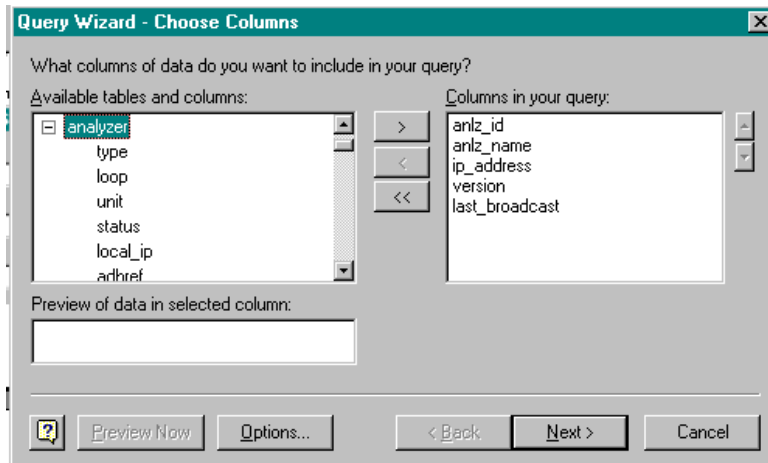
Select Data Source:



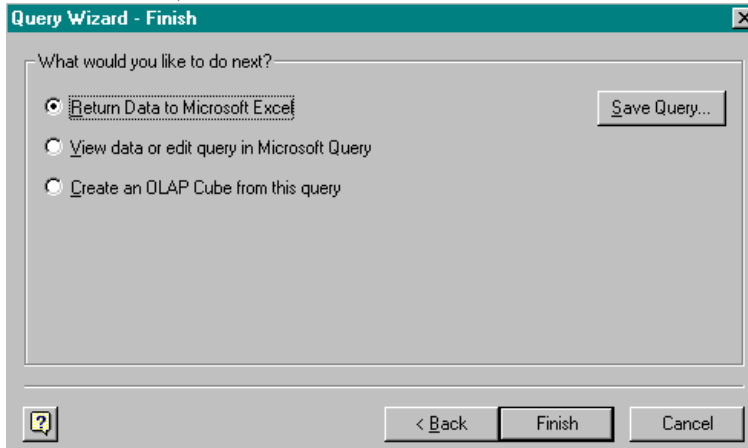
Enter the ip address, user id, and password:



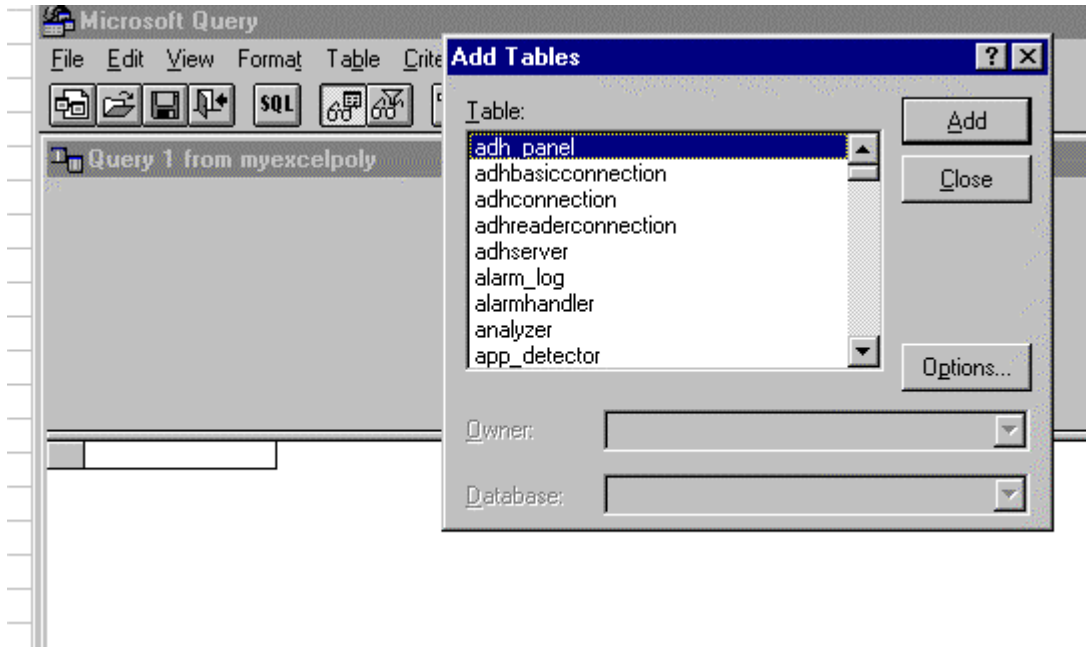
Select the table and attributes:



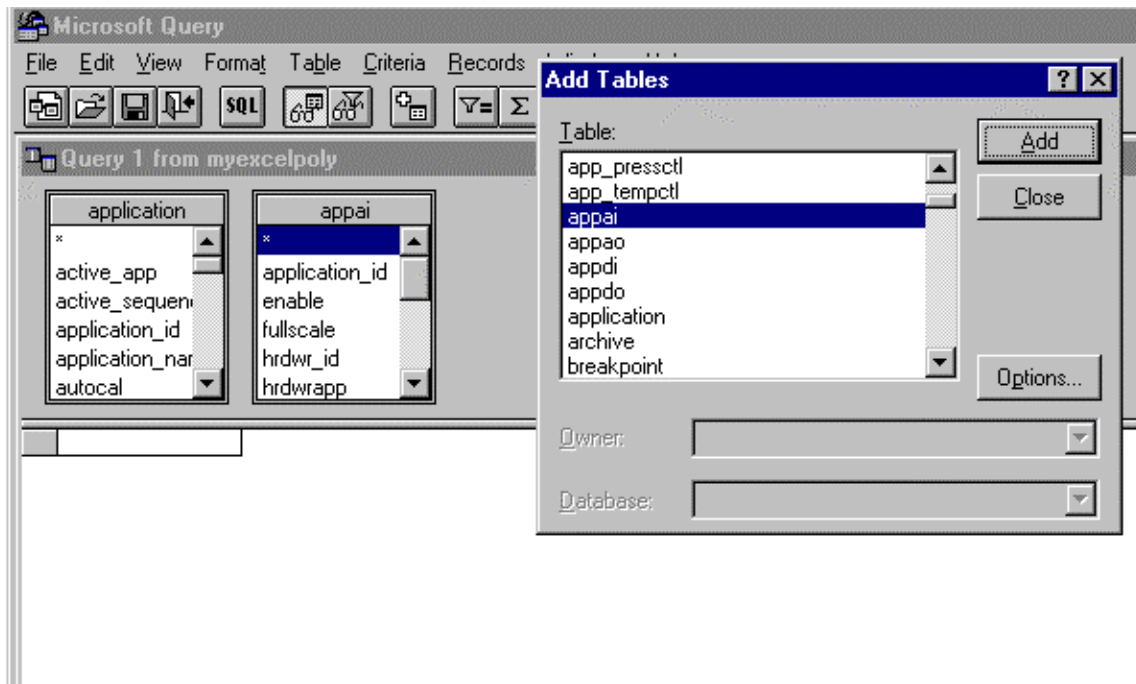
fill out the filter, sort. Return data to Excel:



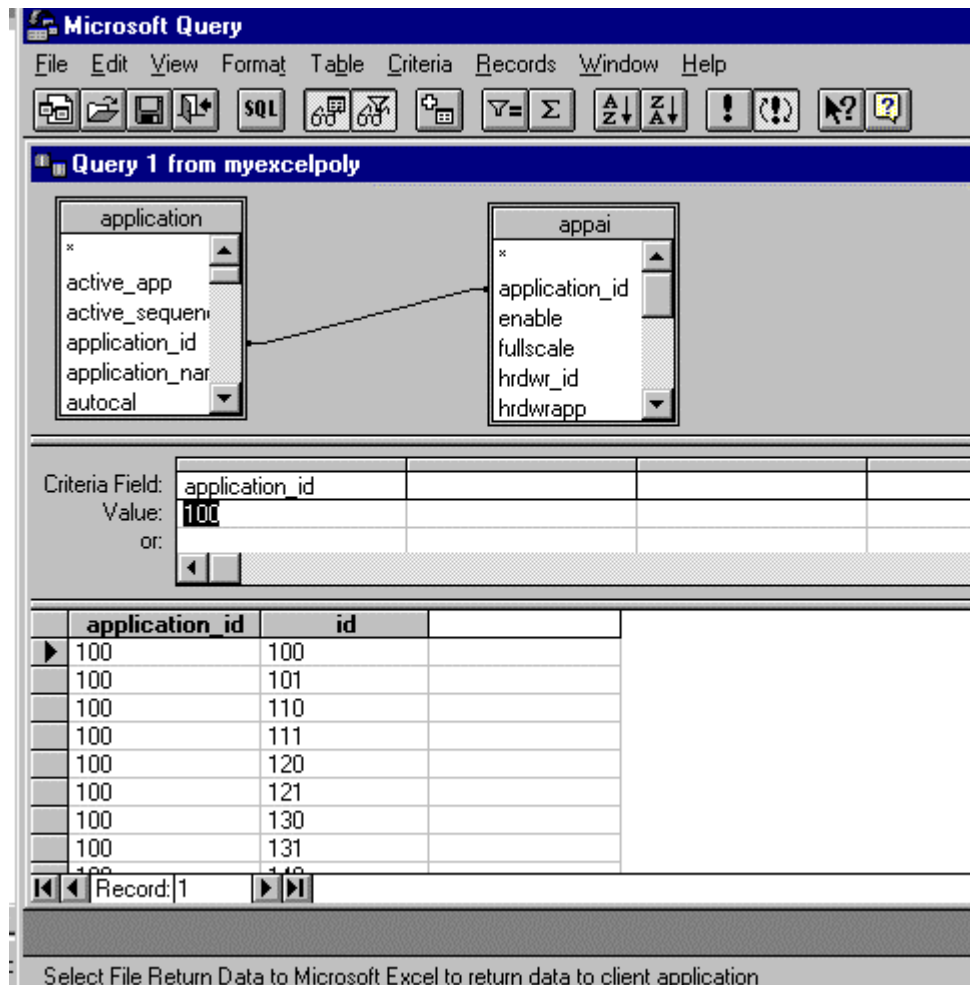
If you don't use the query wizard, you will see the Microsoft Query screens:



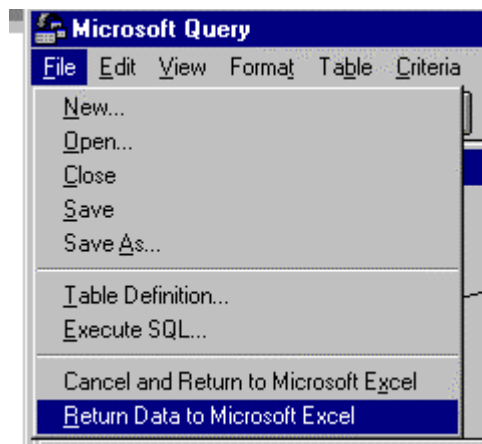
As you select tables they will appear in your query:



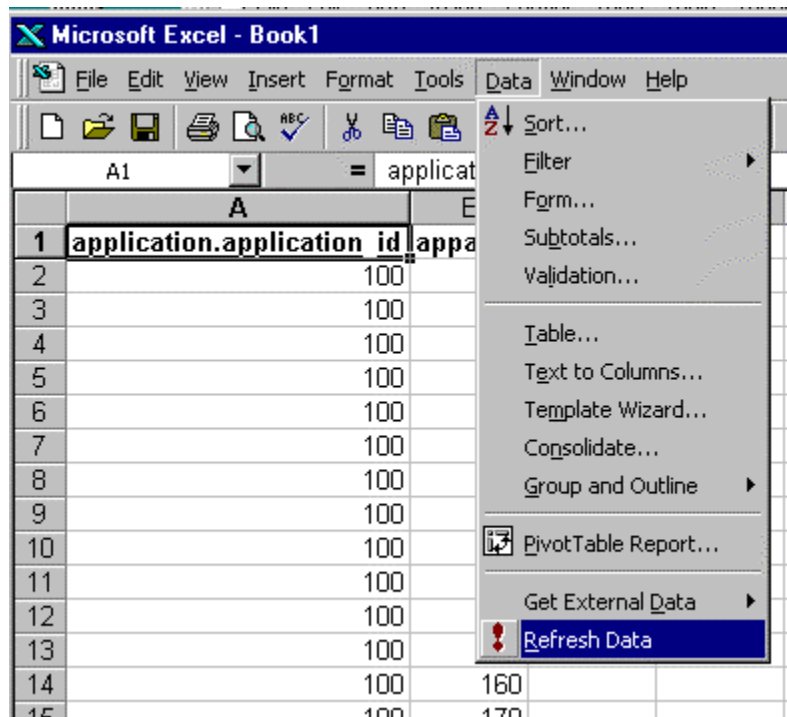
You can now select the attributes of interest. Note that if you select attributes from two different tables, you must perform a join to make sense of the relationship between the tables. Criteria can also be added to qualify your search. It is also possible to use simple SQL. MS Query is an excellent tool for extraction of read-only information from the Maxum. A more complicated example:



Once the proper query has been defined, export to Excel:



To refresh the query from Excel:



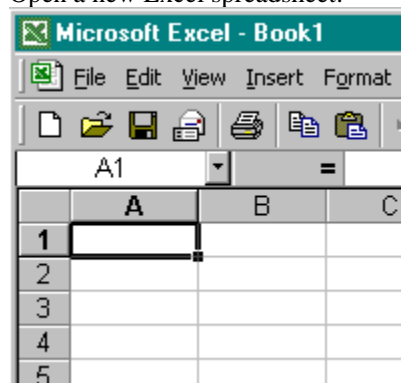
The external data toolbar can be used as a shortcut:



Microsoft Excel Example with VBA functions

Visual Basic functions that translate the database coded data into user-readable form have been delivered in the VBAfunctions.bas file. This file can be included in a worksheet to provide more readable output. This example also shows how to do a timed requery of the database.

Open a new Excel spreadsheet:



Connect to the database, save password, as in the previous example. Select the appdo table.

Query Wizard - Choose Columns

What columns of data do you want to include in your query?

Available tables and columns:

- appdo
 - application_id
 - id
 - ezid
 - name
 - io_status
 - enable

Columns in your query:

Preview of data in selected column:

[?] Preview Now Options... < Back Next >

Query Wizard - Choose Columns

What columns of data do you want to include in your query?

Available tables and columns:

- limitapp
- limitref
- hrdwrapp
- hrdwref
- auto_offtime
- timerref
- + application

Columns in your query:

- application_id
- name
- io_status
- hrdwr_id
- value

Preview of data in selected column:

[?] Preview Now Options... < Back Next > Cancel

Return data to Excel:

Query Wizard - Finish

What would you like to do next?

☒ Return Data to Microsoft Excel

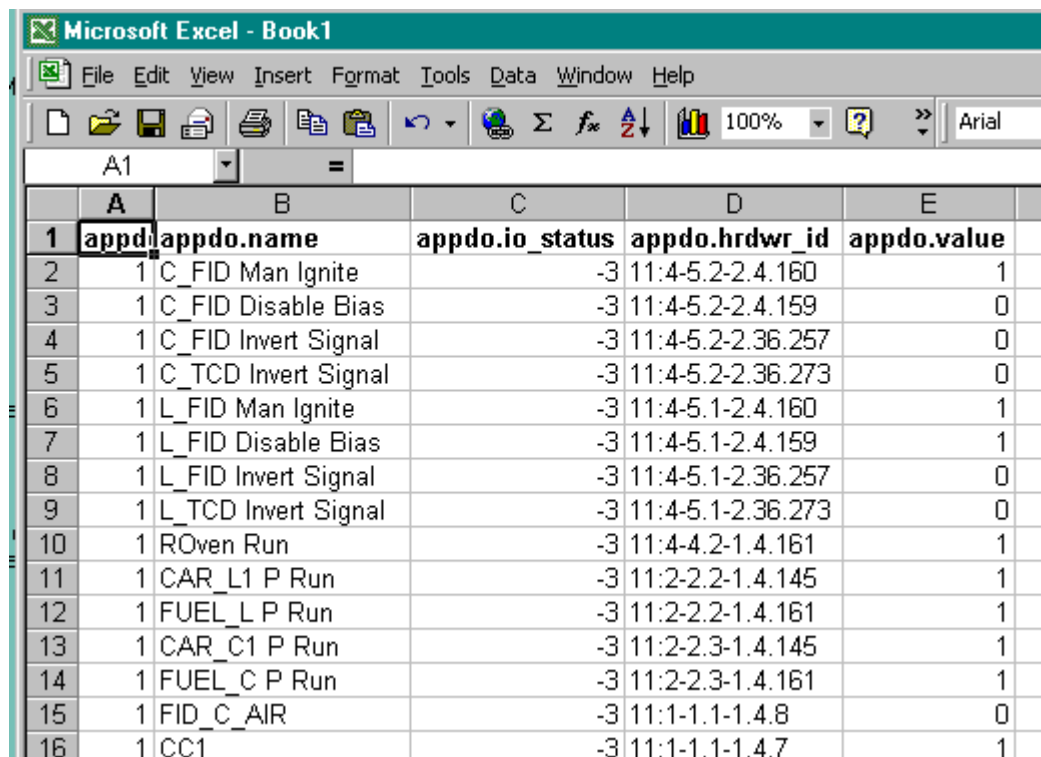
☐ View data or edit query in Microsoft Query

☐ Create an OLAP Cube from this query

Save Query...

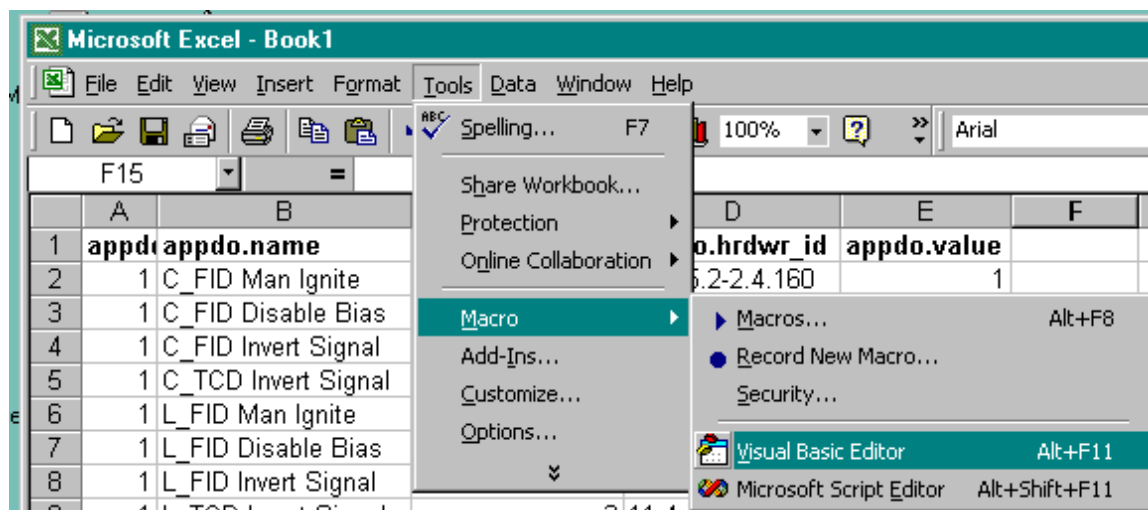
[?] < Back Finish Cancel

Data in the io_status and hrdwr_id columns are not useful in this form:

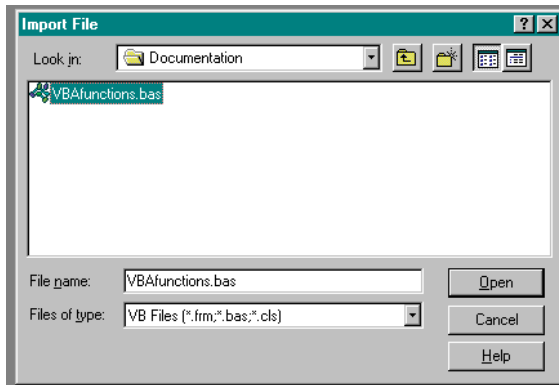


	A	B	C	D	E
1	appdo	appdo.name	appdo.io_status	appdo.hrdwr_id	appdo.value
2	1	C_FID Man Ignite	-3	11:4-5.2-2.4.160	1
3	1	C_FID Disable Bias	-3	11:4-5.2-2.4.159	0
4	1	C_FID Invert Signal	-3	11:4-5.2-2.36.257	0
5	1	C_TCD Invert Signal	-3	11:4-5.2-2.36.273	0
6	1	L_FID Man Ignite	-3	11:4-5.1-2.4.160	1
7	1	L_FID Disable Bias	-3	11:4-5.1-2.4.159	1
8	1	L_FID Invert Signal	-3	11:4-5.1-2.36.257	0
9	1	L_TCD Invert Signal	-3	11:4-5.1-2.36.273	0
10	1	ROven Run	-3	11:4-4.2-1.4.161	1
11	1	CAR_L1 P Run	-3	11:2-2.2-1.4.145	1
12	1	FUEL_L P Run	-3	11:2-2.2-1.4.161	1
13	1	CAR_C1 P Run	-3	11:2-2.3-1.4.145	1
14	1	FUEL_C P Run	-3	11:2-2.3-1.4.161	1
15	1	FID_C AIR	-3	11:1-1.1-1.4.8	0
16	1	CC1	-3	11:1-1.1-1.4.7	1

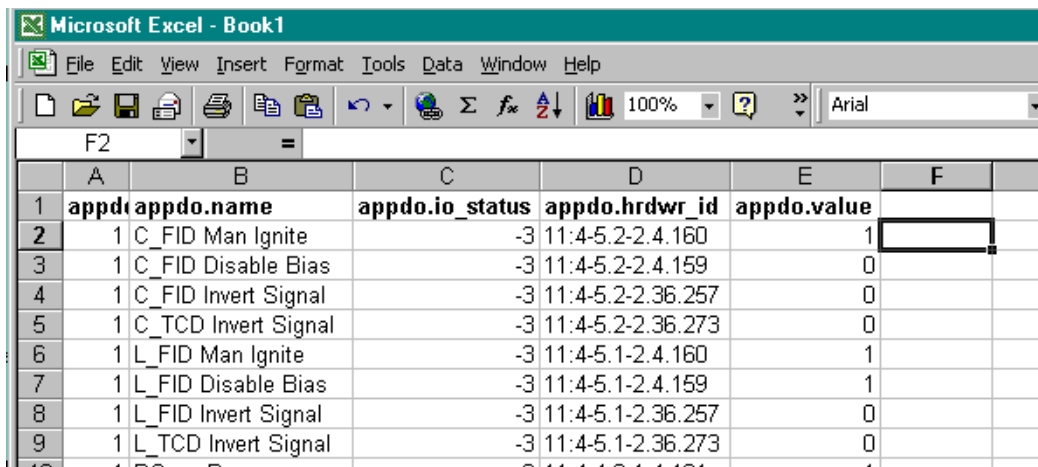
Start the Visual Basic Editor:



Insert the VBAfunctions module by selecting Import File under the File Menu:

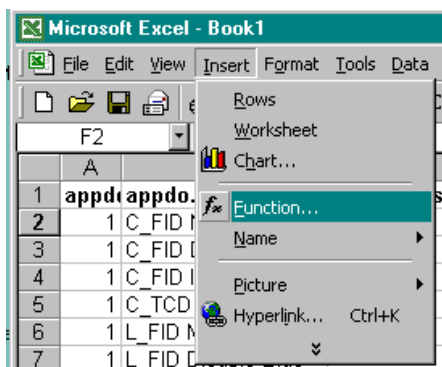


Close the VB editor and return to the spreadsheet. Select An empty cell outside the data area:

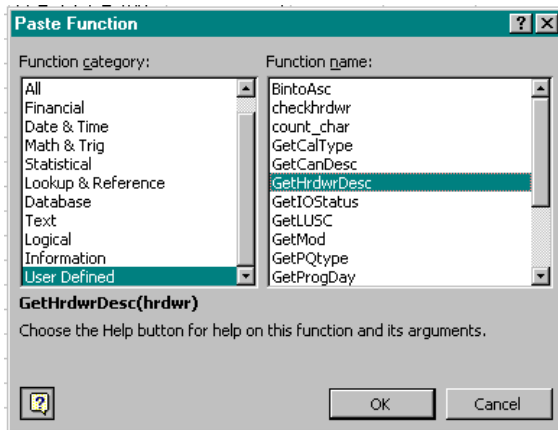


	A	B	C	D	E	F
1	appdo	appdo.name	appdo.io_status	appdo.hrdwr_id	appdo.value	
2	1	C_FID Man Ignite	-3	11:4-5.2-2.4.160	1	
3	1	C_FID Disable Bias	-3	11:4-5.2-2.4.159	0	
4	1	C_FID Invert Signal	-3	11:4-5.2-2.36.257	0	
5	1	C_TCD Invert Signal	-3	11:4-5.2-2.36.273	0	
6	1	L_FID Man Ignite	-3	11:4-5.1-2.4.160	1	
7	1	L_FID Disable Bias	-3	11:4-5.1-2.4.159	1	
8	1	L_FID Invert Signal	-3	11:4-5.1-2.36.257	0	
9	1	L_TCD Invert Signal	-3	11:4-5.1-2.36.273	0	

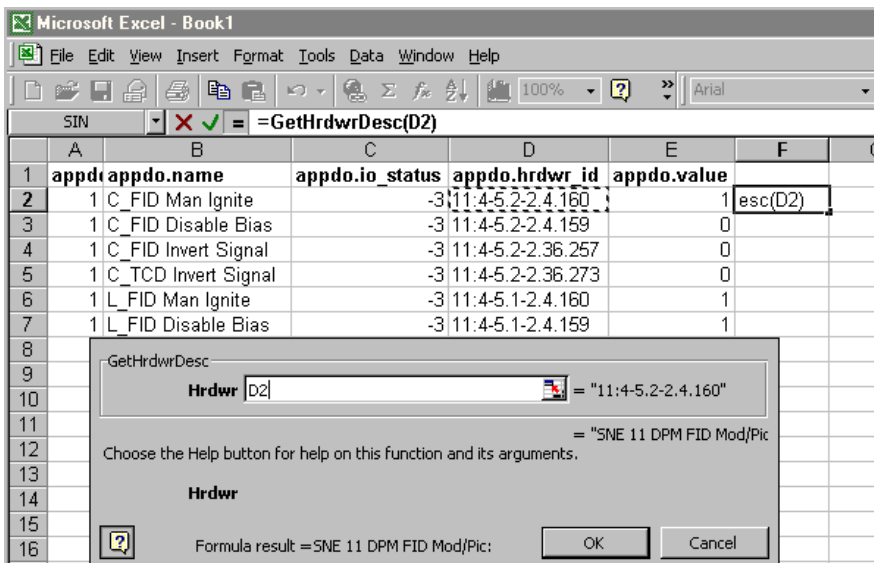
Insert a function:



Select User Defined Functions/GetHrdwrDesc:



Select the first hrdwr_id cell:

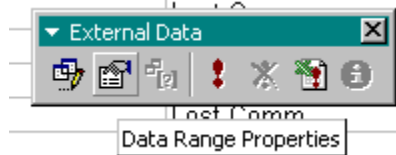


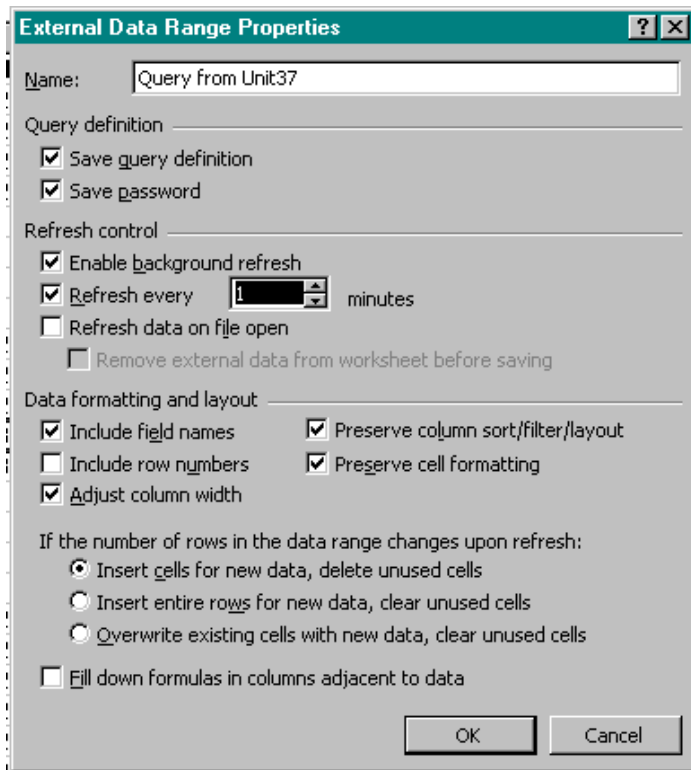
Select OK. Drag box down to include all rows and widen column:

D	E	F
appdo.hrdwr_id	appdo.value	
11:4-5.2-2.4.160	1	SNE 11 DPM FID Mod/Pic: 2/2 DO Chan 160
11:4-5.2-2.4.159	0	SNE 11 DPM FID Mod/Pic: 2/2 DO Chan 159
11:4-5.2-2.36.257	0	SNE 11 DPM FID Mod/Pic: 2/2 DO Chan 257
11:4-5.2-2.36.273	0	SNE 11 DPM FID Mod/Pic: 2/2 DO Chan 273
11:4-5.1-2.4.160	1	SNE 11 DPM FID Mod/Pic: 1/2 DO Chan 160
11:4-5.1-2.4.159	1	SNE 11 DPM FID Mod/Pic: 1/2 DO Chan 159
11:4-5.1-2.36.257	0	SNE 11 DPM FID Mod/Pic: 1/2 DO Chan 257
11:4-5.1-2.36.273	0	SNE 11 DPM FID Mod/Pic: 1/2 DO Chan 273
11:4-4.2-1.4.161	1	SNE 11 DPM Temperature Mod/Pic: 2/1 DO Chan 161
11:2-2.2-1.4.145	1	SNE 11 EPC Pressure Mod/Pic: 2/1 DO Chan 145
11:2-2.2-1.4.161	1	SNE 11 EPC Pressure Mod/Pic: 2/1 DO Chan 161
11:2-2.3-1.4.145	1	SNE 11 EPC Pressure Mod/Pic: 3/1 DO Chan 145
11:2-2.3-1.4.161	1	SNE 11 EPC Pressure Mod/Pic: 3/1 DO Chan 161
11:1-1.1-1.4.8	0	SNE 11 SVCM 8 Valve Mod/Pic: 1/1 EEPROM Chan 8
11:1-1.1-1.4.7	1	SNE 11 SVCM 8 Valve Mod/Pic: 1/1 EEPROM Chan 7
11:1-1.1-1.4.2	0	SNE 11 SVCM 8 Valve Mod/Pic: 1/1 DO Chan 2
11:1-1.1-1.4.1	0	SNE 11 SVCM 8 Valve Mod/Pic: 1/1 DO Chan 1
11:1-1.2-1.4.8	0	SNE 11 SVCM 8 Valve Mod/Pic: 2/1 EEPROM Chan 8
11:1-1.2-1.4.7	1	SNE 11 SVCM 8 Valve Mod/Pic: 2/1 EEPROM Chan 7
11:1-1.2-1.4.6	1	SNE 11 SVCM 8 Valve Mod/Pic: 2/1 EEPROM Chan 6
11:1-1.2-1.4.2	0	SNE 11 SVCM 8 Valve Mod/Pic: 2/1 DO Chan 2
11:1-1.2-1.4.1	1	SNE 11 SVCM 8 Valve Mod/Pic: 2/1 DO Chan 1
11:4-5.2-2.4.147	0	SNE 11 DPM FID Mod/Pic: 2/2 DO Chan 147
11:4-5.2-2.4.163	0	SNE 11 DPM FID Mod/Pic: 2/2 DO Chan 163

The same is done with the io_status column and function GetIOStatus:

To set up an automatic refresh, select Data Range Properties on the External Data toolbar:





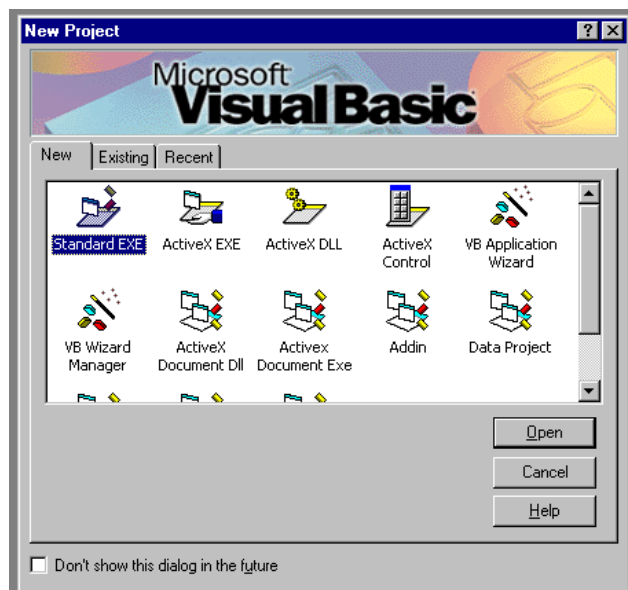
Set up a one minute refresh. Excel will requery the database every minute.

Microsoft Visual Basic 6.0 Example

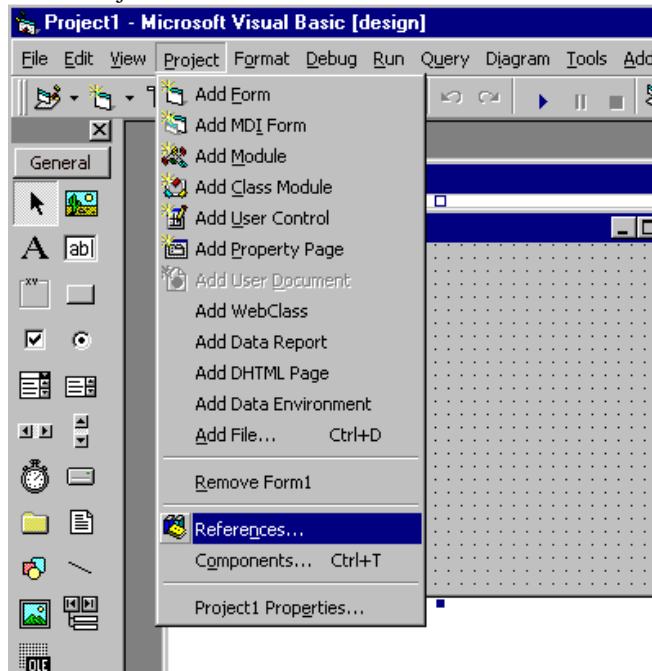
Visual Basic provides a simple, but powerful interface to the Maxum analyzer. Static reads and writes can be accomplished with very little effort. The simple example requires no Visual Basic code to be written. The more complicated examples require a knowledge of Visual Basic 6 programming. Recommended reading: Gary Cornell's *Visual Basic 6 From the Ground Up* and John Connell's *Visual Basic 6 Database Programming*. Note that the Maxum's ODBC driver is not fully functional. For a VB6 tutorial, try this site: <http://www.vbinformation.com/tutorial.htm>.

Follow this example to set up a simple application:

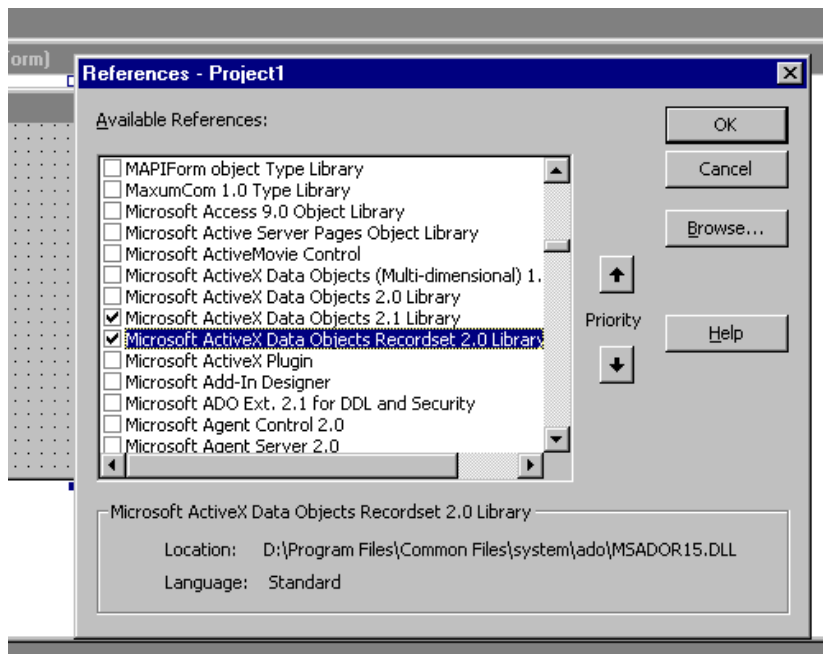
1. Setup a new standard.exe VB project.



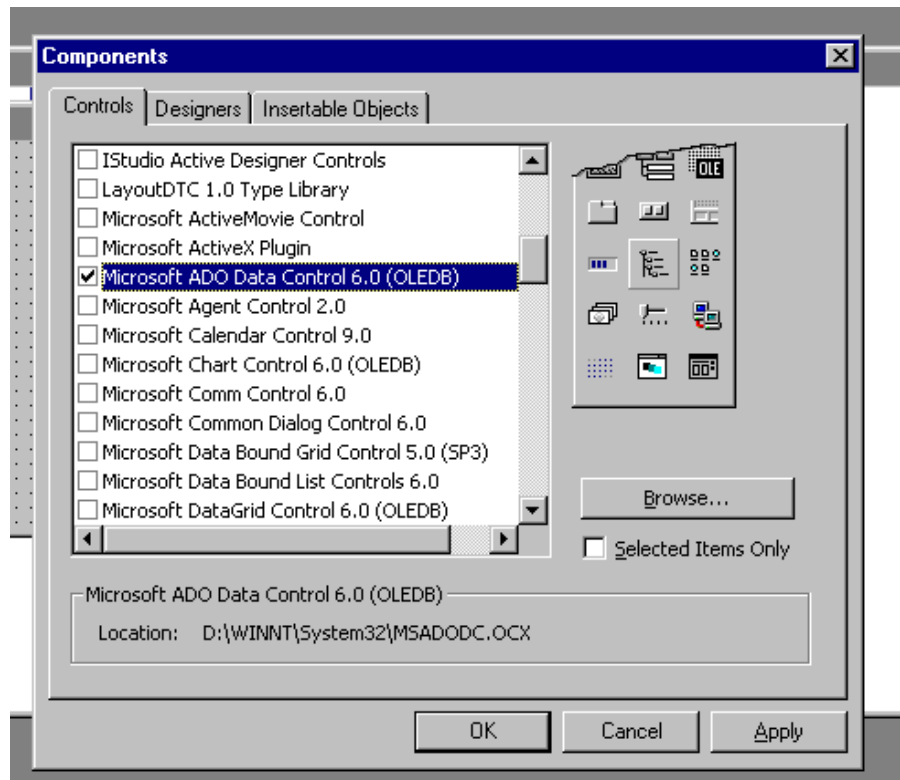
2. Select Project References



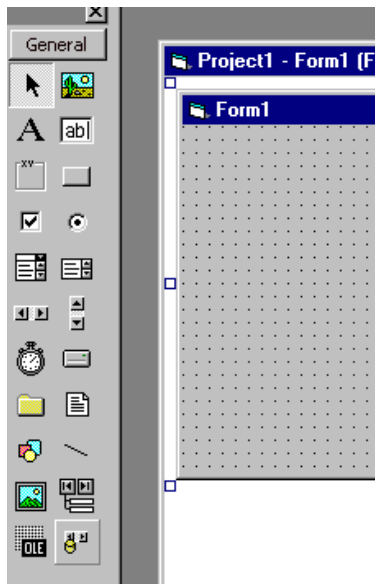
3. Select MS ActiveX Data Objects 2.1 Library and MS ActiveX Data Objects Recordset 2.0 Library



4. Under Project/Components, select MS ADO Data Control 6.0 (OLEDB)

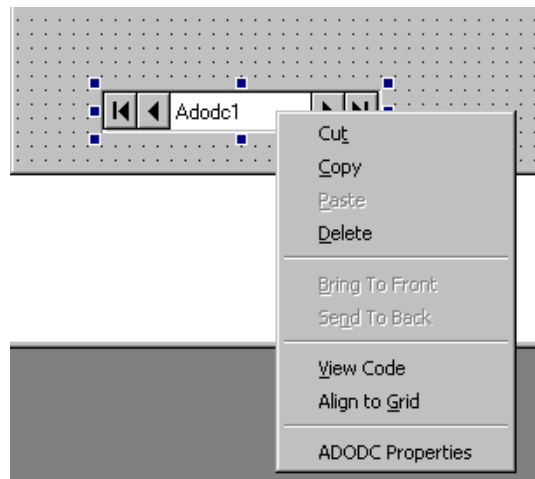


5. Select the Adodc from the Control toolbar

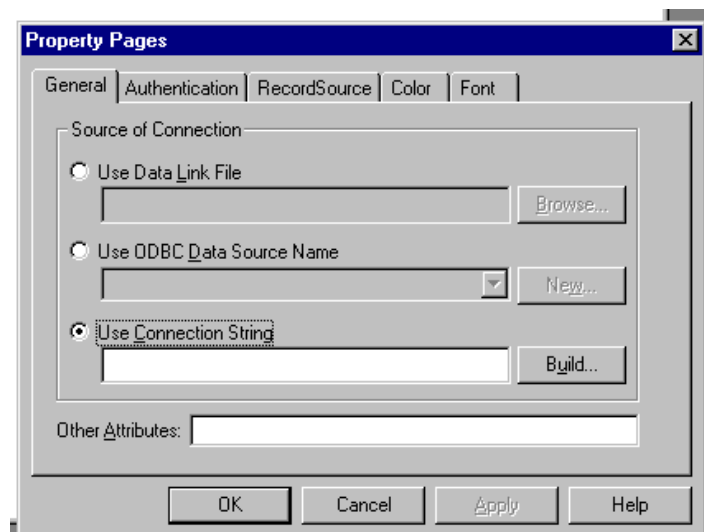


It should be the bottom control.

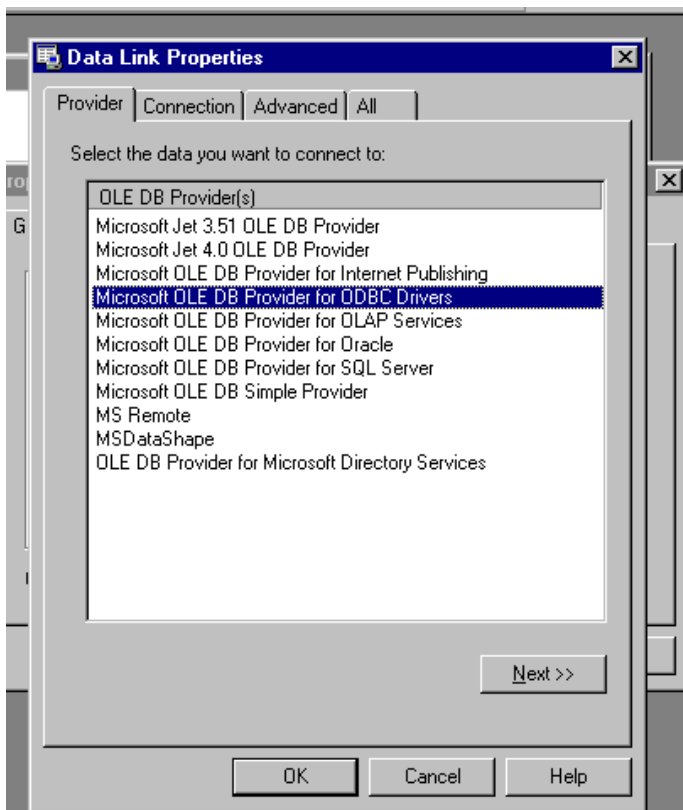
6. Place the control on the form and select ADODC Properties (right click on the control)



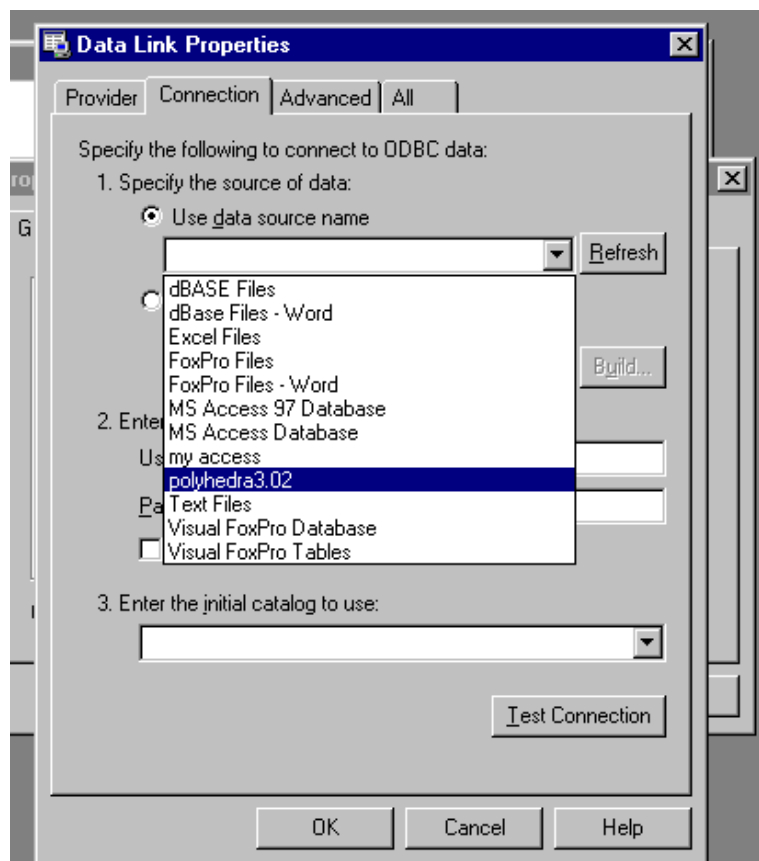
7. Select Build for Connection String



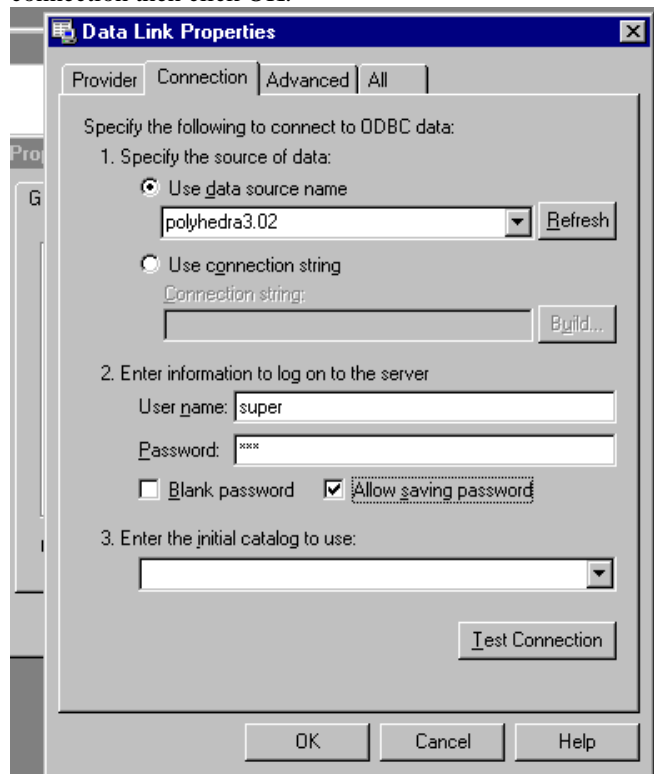
8. Select Microsoft OLE DB Provider for ODBC Drivers, then click Next.



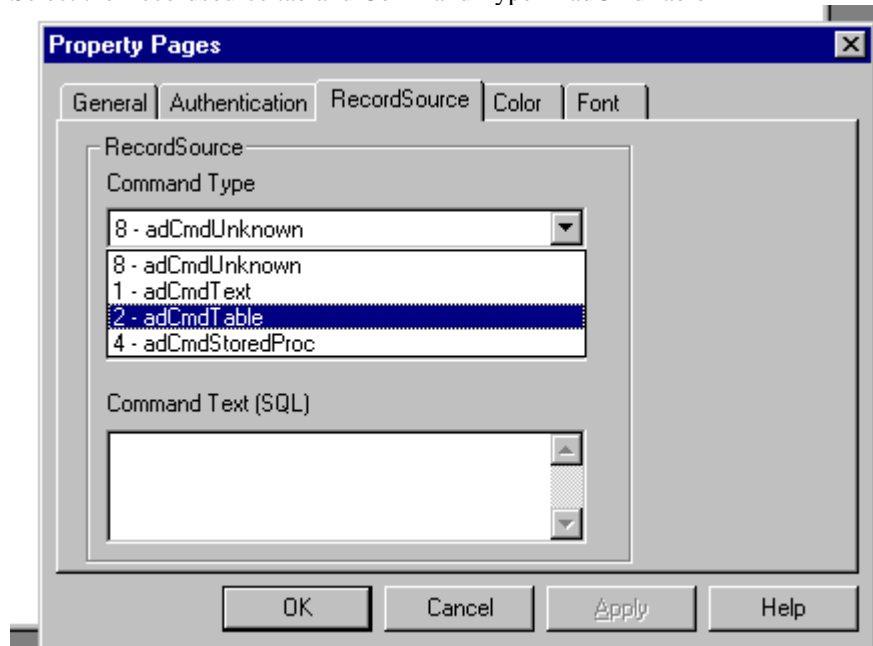
9. Select the Polyhedra datasource:



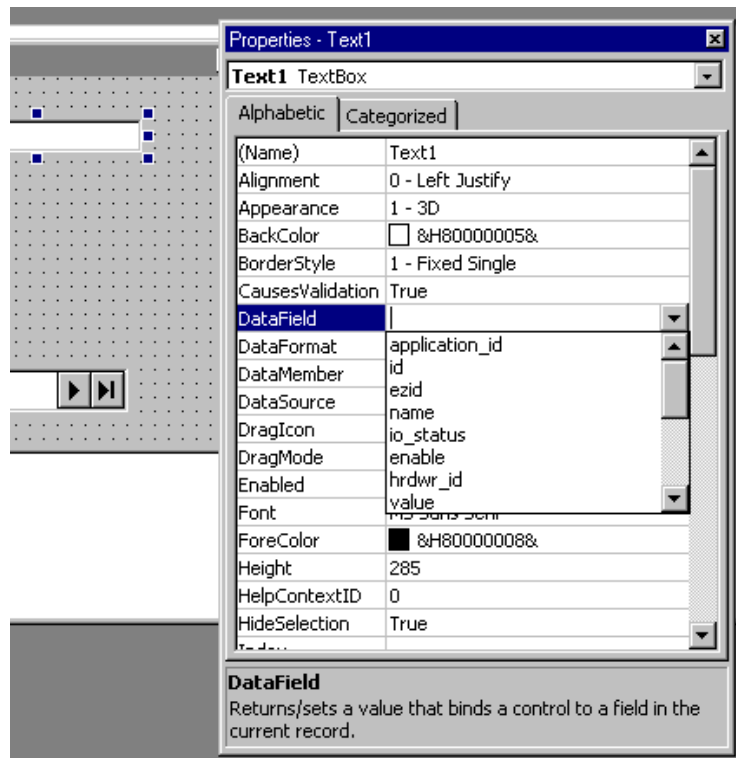
10. Enter a user and password and allow saving of the password. It's a good idea to test the connection then click OK.



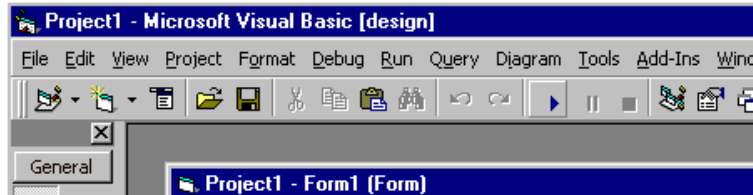
11. Select the Recordsource tab and Command Type 2- adCmdTable



14. Under DataField, select a column from the table(this is the table that was selected in the ADO data control's recordsource).



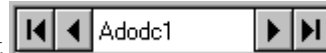
15. Test the form with Run, or the  button







16. It should look like this (with name displayed in the text box)



17. Use the arrow controls to navigate through the recordset



- Buttons indicate:
-  goes to first record of recordset
 -  goes back one record
 -  goes forward one record
 -  goes to the last record of the recordset

18. Change the name in the appdo table by typing over a name and then going to a different record.
19. Add more fields to the form. Note that Boolean values should be used with a check box. Binary fields should be avoided. Datetimes are stored in GMT and there is no automatic conversion.
- Here is a finished form to display the appdo table. No Visual Basic code is used, but could now be added to customize this form.

A screenshot of a Visual Basic form titled 'Form1' in design view. The form has a dotted grid background. It contains several controls: a label 'id' with a corresponding 'Label1' text box; a label 'name' with a 'Text1' text box; two checkboxes labeled 'enable' and 'value'; a label 'Hardware id' with a 'Text4' text box; and a recordset navigation control at the bottom labeled 'Adodc1'.

Which results in this:

A screenshot of the same 'Form1' in runtime view. The data is populated: 'id' is 752, 'name' is 'BAL_CL3_TCD', 'enable' is checked, 'value' is unchecked, and 'Hardware id' is '11:4-6.2-4.4.147'. The 'Adodc1' navigation control is at the bottom.

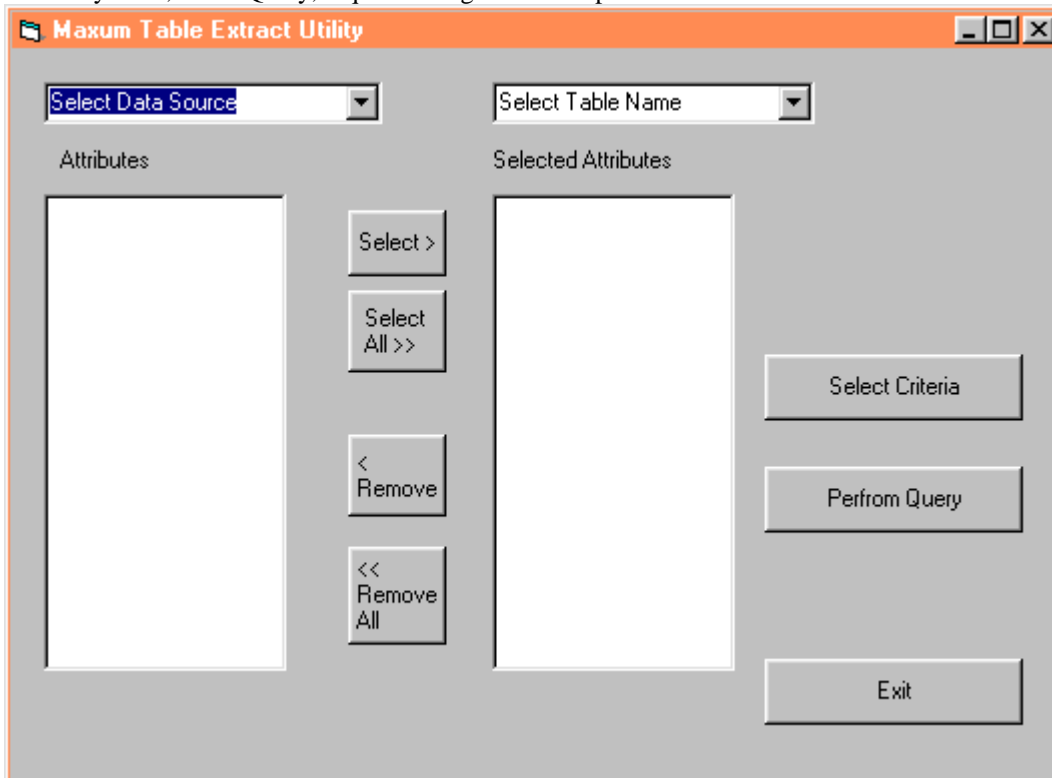
A More Complicated Microsoft Visual Basic 6.0 Example

In the ODBC directory, a sample application called MaxumExtract is available. This program can be used as is, or the project source can be used as a template to develop another application. The files are:



The project includes four forms. Its purpose is to select data from a Maxum database and allow it to be viewed, printed, and extracted to a comma-separated file (suitable for MS Excel). It does not allow editing of the data. The MaxumExtract.exe is only usable on systems that have the Visual Basic files installed, so it is not suitable for distribution. These required files may be installed by other applications, like Internet Explorer.

The entry form, SelectQuery, requires designation of a predefined datasource and table of interest:



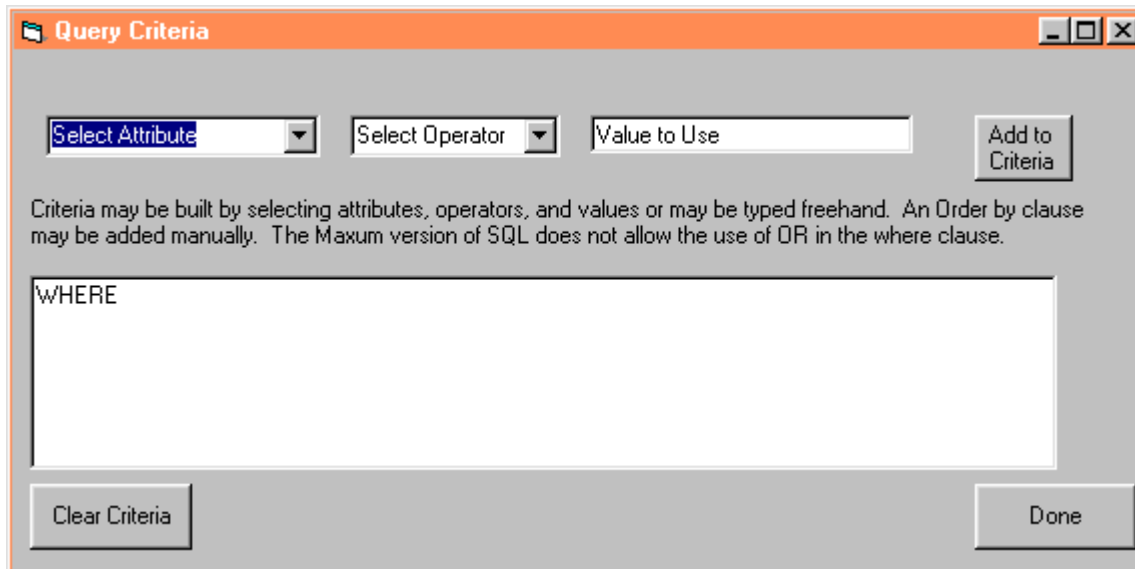
Once the table is selected, select the attributes(data fields) of interest:

The screenshot shows the 'Maxum Table Extract Utility' window. At the top, there are two dropdown menus: the first contains 'DianaH' and the second contains 'appdc'. Below these are two lists of attributes. The 'Attributes' list on the left contains: ezid, name, io_status, enable, hrdwr_id, value, readback, text0, text1, limitapp, limitref, hrdwrapp, hrdwrref, auto_offtime, and timerref. The 'Selected Attributes' list on the right contains: *application_id and *id. Between the lists are four buttons: 'Select >', 'Select All >>', '< Remove', and '<< Remove All'. To the right of the 'Selected Attributes' list are three buttons: 'Select Criteria', 'Perfrom Query', and 'Exit'.

The primary key fields, in this case, application_id and id are always required.

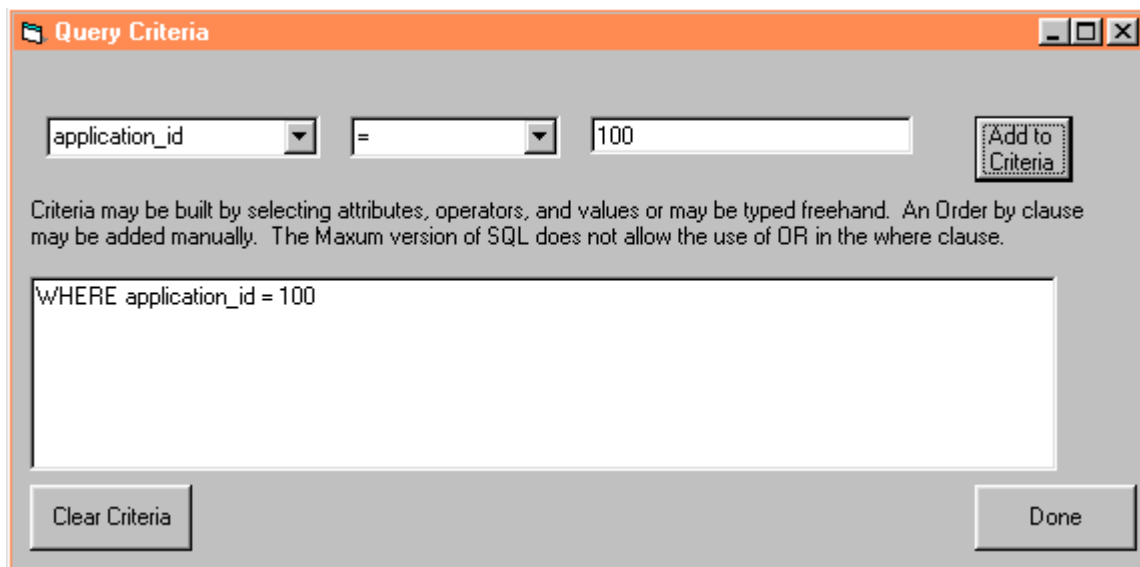
This screenshot shows the same 'Maxum Table Extract Utility' window, but with more attributes selected. The 'Attributes' list on the left now contains: ezid, io_status, readback, text0, text1, limitapp, limitref, hrdwrapp, hrdwrref, and timerref. The 'Selected Attributes' list on the right now contains: *application_id, *id, name, enable, hrdwr_id, value, and auto_offtime. The 'Select >' button between the lists is highlighted with a dashed border. The other buttons and dropdown menus remain the same as in the previous screenshot.

The Select Criteria button activates the fmCriteria form, which allows selection of a “where” clause.



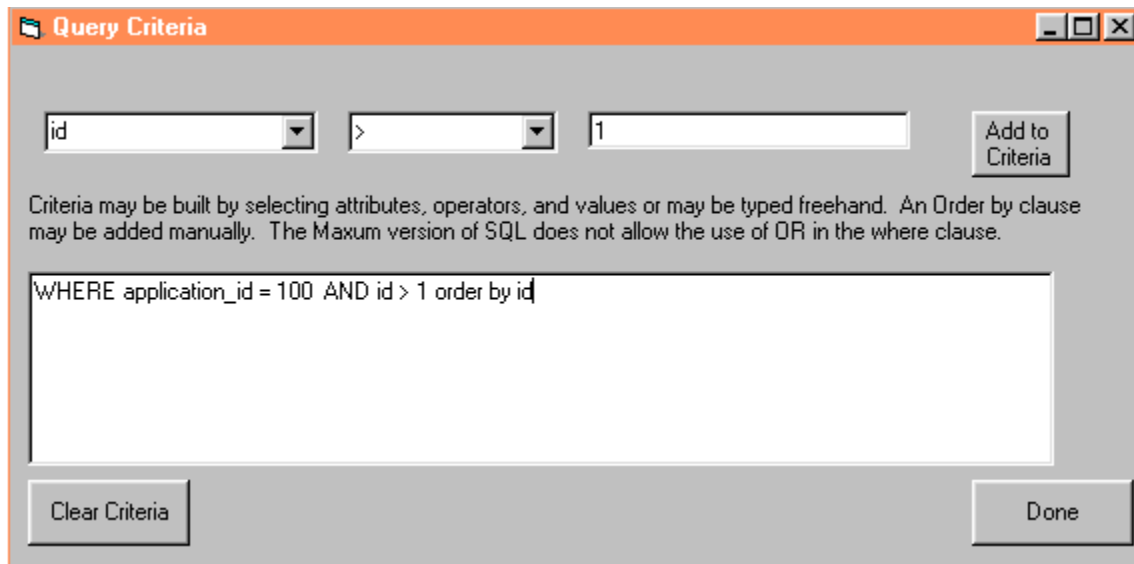
The image shows a dialog box titled "Query Criteria" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the dialog, there are three input fields at the top: "Select Attribute" (a dropdown menu), "Select Operator" (a dropdown menu), and "Value to Use" (a text box). To the right of these fields is a button labeled "Add to Criteria". Below these fields is a paragraph of text: "Criteria may be built by selecting attributes, operators, and values or may be typed freehand. An Order by clause may be added manually. The Maxum version of SQL does not allow the use of OR in the where clause." Below this text is a large text area containing the word "WHERE". At the bottom of the dialog are two buttons: "Clear Criteria" on the left and "Done" on the right.

A where clause can be keyed in , or built with the three upper boxes:



The image shows the same "Query Criteria" dialog box, but with the fields populated. The "Select Attribute" dropdown now shows "application_id", the "Select Operator" dropdown shows "=", and the "Value to Use" text box contains "100". The "Add to Criteria" button is now outlined with a dashed border, indicating it is active. The text area below now contains the full clause: "WHERE application_id = 100". The "Clear Criteria" and "Done" buttons remain at the bottom.

Multiple items can be selected, and an “order by” clause can be typed in to sort the items.



The Query Criteria dialog box has an orange title bar. It contains three input fields: a dropdown menu with 'id', a dropdown menu with '>', and a text box with '1'. To the right is an 'Add to Criteria' button. Below these is a text area containing the SQL query: 'WHERE application_id = 100 AND id > 1 order by id'. At the bottom are 'Clear Criteria' and 'Done' buttons.

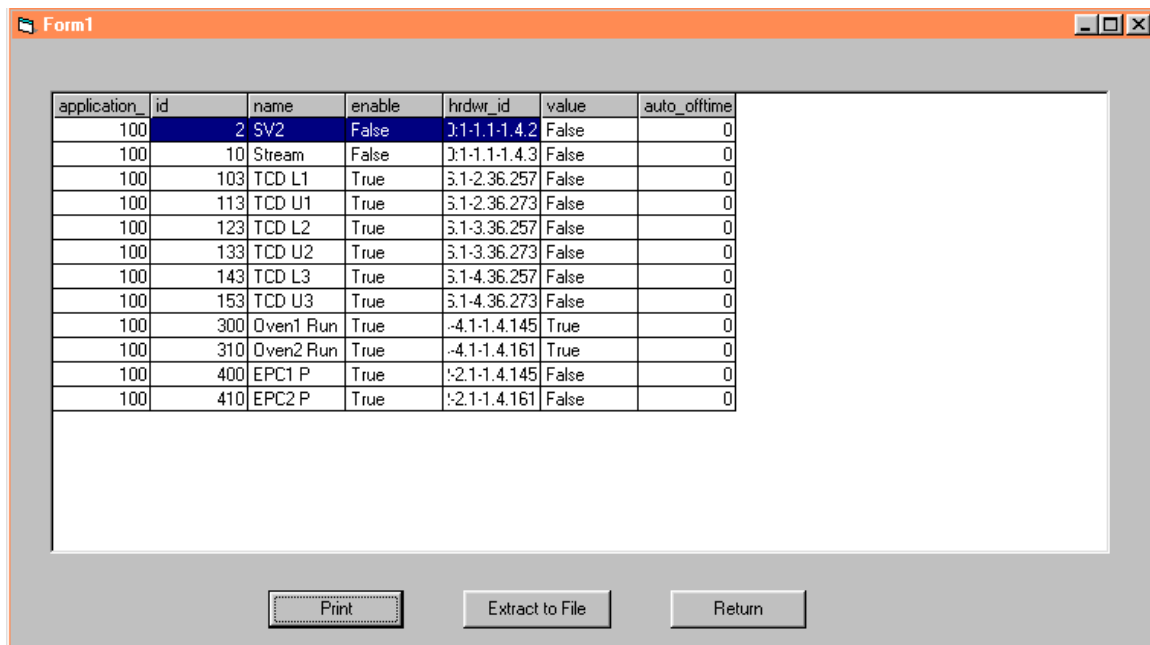
id > 1 Add to Criteria

Criteria may be built by selecting attributes, operators, and values or may be typed freehand. An Order by clause may be added manually. The Maxum version of SQL does not allow the use of OR in the where clause.

WHERE application_id = 100 AND id > 1 order by id

Clear Criteria Done

Select Done, then Perform Query on the SelectQuery form. The QueryGrid form is displayed.



The Form1 window has an orange title bar. It displays a data grid with 7 columns: application_id, id, name, enable, hrdwr_id, value, and auto_offtime. The grid contains 15 rows of data. Below the grid are three buttons: Print, Extract to File, and Return.

application_id	id	name	enable	hrdwr_id	value	auto_offtime
100	2	SV2	False	3:1-1.1-1.4.2	False	0
100	10	Stream	False	3:1-1.1-1.4.3	False	0
100	103	TCD L1	True	3:1-2.36.257	False	0
100	113	TCD U1	True	3:1-2.36.273	False	0
100	123	TCD L2	True	3:1-3.36.257	False	0
100	133	TCD U2	True	3:1-3.36.273	False	0
100	143	TCD L3	True	3:1-4.36.257	False	0
100	153	TCD U3	True	3:1-4.36.273	False	0
100	300	Oven1 Run	True	-4:1-1.4.145	True	0
100	310	Oven2 Run	True	-4:1-1.4.161	True	0
100	400	EPC1 P	True	3:2.1-1.4.145	False	0
100	410	EPC2 P	True	3:2.1-1.4.161	False	0

Print Extract to File Return

Print will print the grid to the default printer. Printing format is limited. Extract to File will print the grid to a comma-separated file that can be viewed in Excel or Notepad or loaded into Access. Return closes the form.

The MaxumExtract application does a snapshot query of the Maxum, with not ability to edit the values. The .vbp file will allow you to load the project and view the source code.

Editing Maxum tables with Visual Basic 6.0

In the ODBC directory, a sample application called AppMonitor is available. This program is an incomplete application that demonstrates the ability to poll the database with timers and update tables with the ADOConnection.Execute method. This method only is available in the Maxum 3.1 release. The files are:



It is left to the user to view the code and use it as a template for developing applications. Note that the forms use timers to requery the database periodically. Requerying the database frequently can seriously degrade the performance of the analyzer. The ODBC driver does not have the ability to make dynamic connections to the database.

VB6 Code Sample for building connection string

This code comes from a form that has a text1 text box for entering the IP address. Count_char is a function in the VB function library(VBAFunctions.bas) that is delivered with the 3.11 release.

```
Dim connectString As String
Dim i As Integer

If count_char(Text1.Text, ".") <> 3 Then
    MsgBox "Invalid ip address"
    Text1.SetFocus
    Exit Sub
End If
If Right(Text1.Text, 5) <> ":8001" Then Text1.Text = Text1.Text + ":8001"
Set Records = New ADODB.Recordset
'a data source was selected
connectString = "Provider=MSDASQL.1;Password=555;Persist Security Info=True;User ID=super;"
connectString = connectString & "Extended Properties=""DSN=Polyhedra;SERVICE=" + Text1.Text + ";UID= super;""
If adoConnection.State = adStateOpen Then adoConnection.Close
adoConnection.Open connectString
```