

Stata Users Group Meeting

September 18th, 2015

Lisbon School of Economics & Management



Using ODBC with Stata

Rita Sousa (rcsousa@bportugal.pt)

Bank of Portugal



BANCO DE
PORTUGAL
EUROSISTEMA

Center for Mathematics and Applications



Summary



- ❖ Introduction
- ❖ ODBC Support
- ❖ Configuring ODBC
- ❖ ODBC in Stata
- ❖ Loading data
- ❖ Practical Example
- ❖ Useful Tips
- ❖ Final Remarks

Introduction



- Open DataBase Connectivity (ODBC) is a standardized set of function calls that can be used to access data stored in database management systems.
- Stata's `odbc` command allows us to load, write, and view data from ODBC sources.
- In my presentation I want to show you how to deal with large databases using the ODBC resources.

ODBC Support



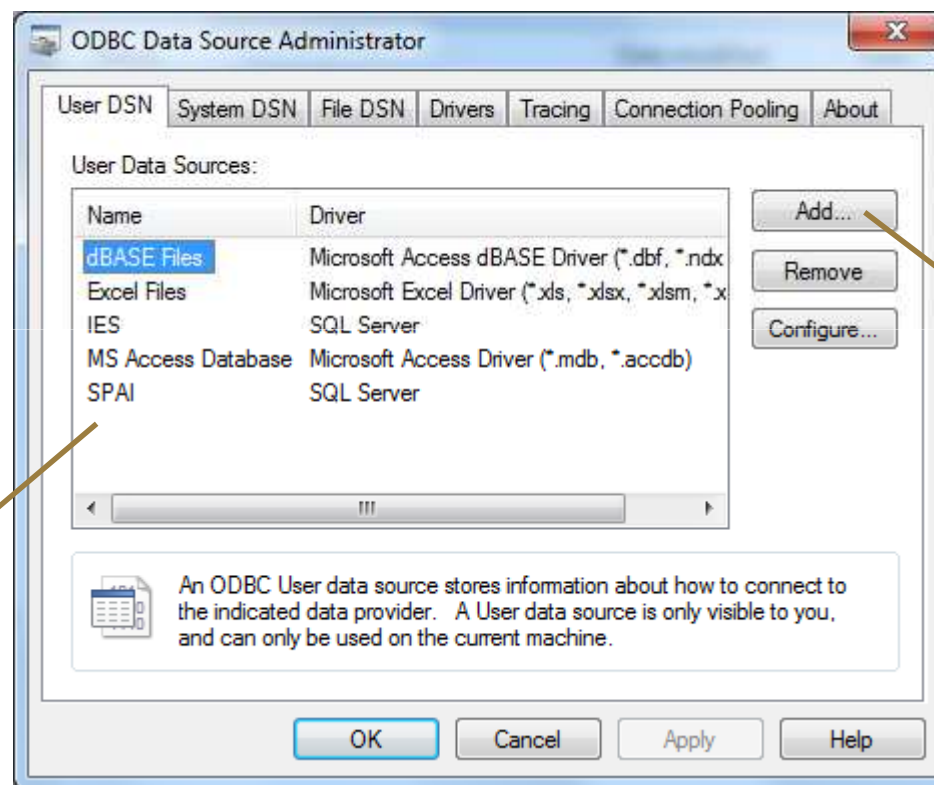
- Import data from any ODBC data source, such as Oracle, SQL Server, Access, Excel, MySQL, and DB2.
- Export data to new or existing ODBC tables
- Execute custom SQL commands



Configuring ODBC for Windows

- Before you start using the `odbc` command in Stata, you must first set up a data source name (DSN) in the ODBC Data Source Administrator.
 1. Select **Control Panel**
 2. Select **System and Security** in the Control Panel
 3. Next select **Administrative Tools**
 4. Double-click on **Data Sources (ODBC)** to open the ODBC Data Source Administrator

Configuring ODBC for Windows

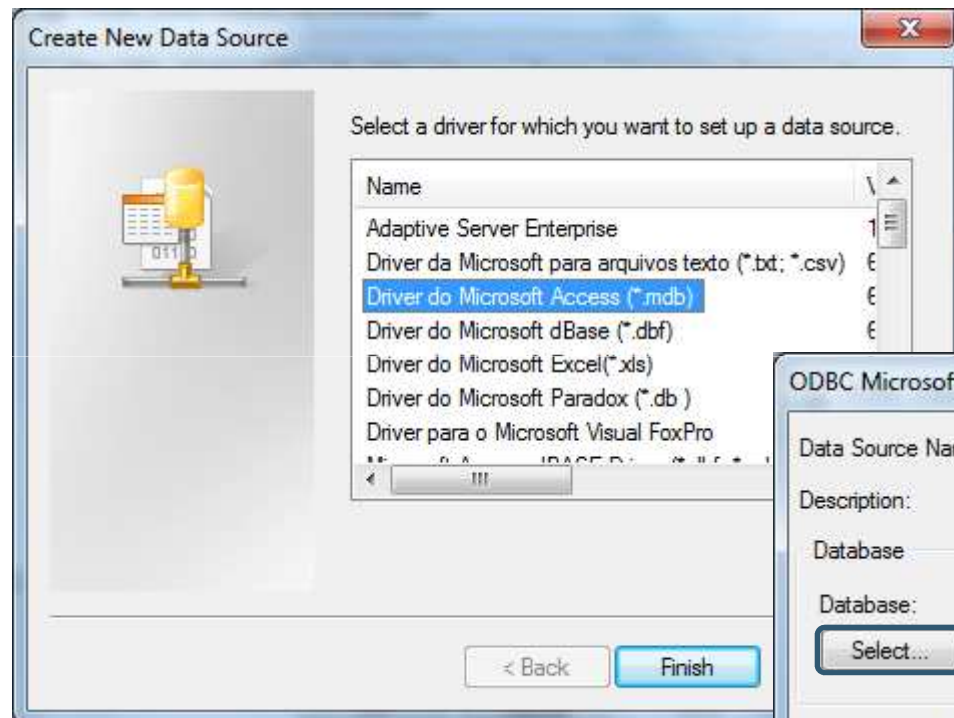


List of User Data Sources Names (DSN)

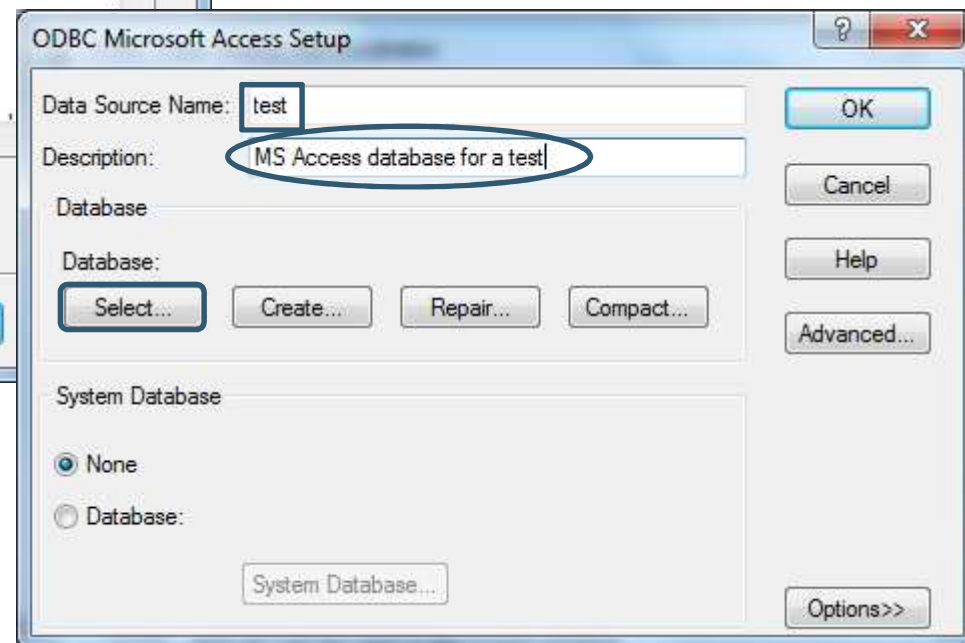
Create a New DSN



Configuring ODBC for Windows

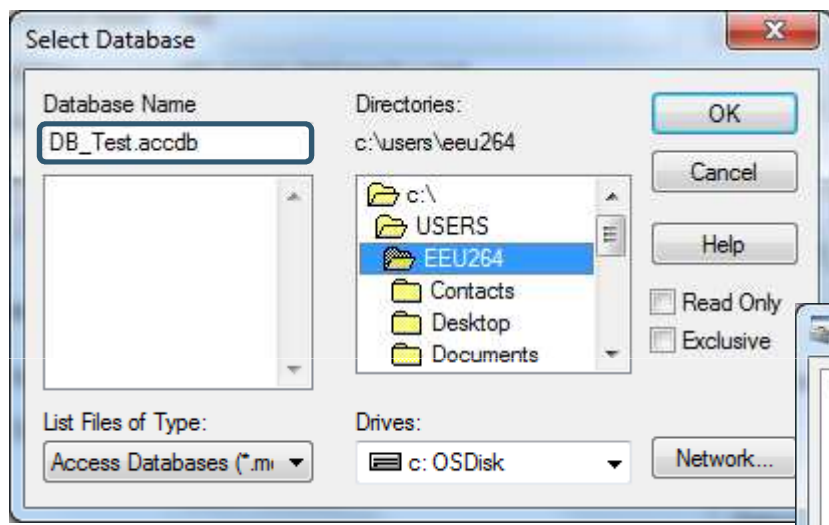


- ❖ Select the appropriate driver from the list
- ❖ For example, choose **Microsoft Access Driver (*.mdb)**

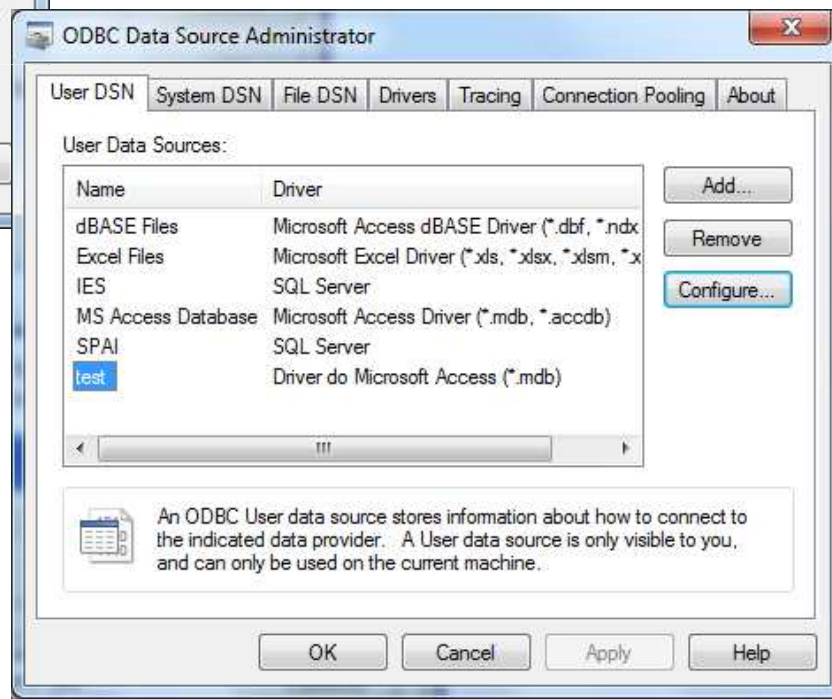




Configuring ODBC for Windows



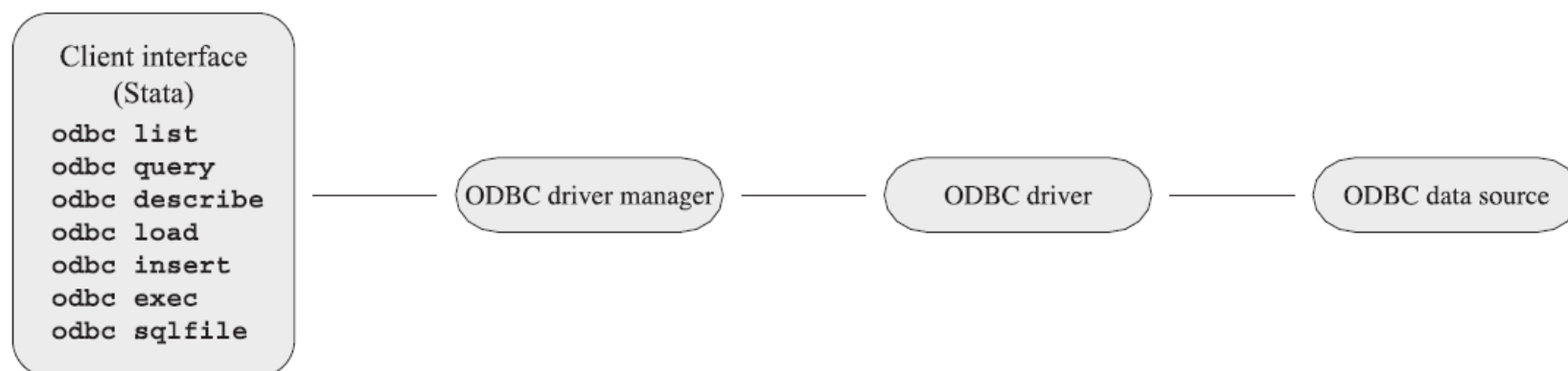
- ❖ Select your MS Access file to be in the Data Source test that we are creating
- ❖ After click OK





Configuring ODBC for Windows

- 64-bit Windows ships with two different ODBC Data Source Administrators, 64-bit and 32-bit
- Make sure you are using the correct version according to your Stata version
- Stata provides `odbc` as the Client interface





ODBC in Stata

- `odbc list` produces a list of ODBC data source names to which Stata can connect

```
. odbc list
```

Data Source Name	Driver
IES	SQL Server
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.ndx)
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xl)
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)
SPAI	SQL Server
test	Driver do Microsoft Access (*.mdb)

- `odbc query` show a list of table names available from a specified data source
- `odbc describe` lists column names and types for each table available



Loading Data

- In Stata we can use some commands to import data stored in formats different from `.dta` such as `insheet`, `infix`, `import excel`, etc.
- `odbc load` reads an ODBC table into memory.
- You can load an ODBC table specified in the `table()` option or load an ODBC table generated by an SQL Command specified in the `exec()` option.

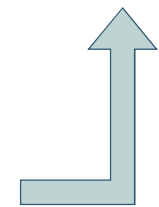
Loading Data



nuemp	pemp	vn	dtemp	caemp	ldemp	nut3_emp
580071	1	21667	1	52410	101	161
581033	8	2171229	1	52310	101	161
581915	1	68694	1	28401	101	161
581952	1	32515	1	36141	101	161
582290	4		1	28742	101	161
582594	1	39500	1	45211	101	161
582596	1		1	5020	101	161
582816	18		1	35420	101	161
583335	3		1	93021	101	161
583671	2		1	74120	101	161
583861	1	26404	1	52112	101	161
583865	5		1	25240	101	161
585590	11		1	05291	101	161

nuemp	nemp	turnover
580071	1	21667
581033	8	2171229
581915	1	68694
581952	1	32515
582590	4	

- `odbc load id=nuemp nemp=pemp turnover=vn in 1/5, table("Table_Test") dsn("test")`
- `odbc load, exec(`Select top 5 nuemp, pemp as pemp, vn as turnover From Table_Test`) dsn("test")`





Practical Example

- Suppose now you want to access data stored in a Sql Server database.
- You have to register your ODBC database with the ODBC Data Source Administrator:
 1. Create a **new Data Source**
 2. Choose **Sql Server Driver**
 3. Specify the **name of the server** you want to connect



Practical Example

Microsoft SQL Server DSN Configuration

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

Name: SPAI

How do you want to describe the data source?

Description: Database from a system to chare information

Which SQL Server do you want to connect to?

Server: srv-itr-sqld01\srdrn2

```
. odbc list
```

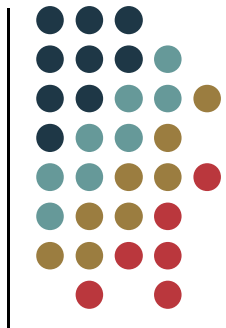
Data Source Name	Driver
IES	SQL Server
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.ndx)
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xl)
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)
SPAI	SQL Server
test	Driver do Microsoft Access (*.mdb)

```
. odbc query "SPAI"
```

```
DataSource: SPAI  
Path       : dipai060
```

```
AFS_Sem_Grupo  
AQ_IBAN  
AUX_Carreg_GFC  
AUX_PaisOriCap  
AUX_PaisOriCap_UltAFV  
Aux_TABAS_CambiosComplementares  
Aux_TABAS_CambiosComplementares_Hist  
Aux_TABAS_CambiosOficiais  
Aux_TABAS_CambiosOficiais_Hist  
AUX_TABAS_Divisa  
AUX_TABAS_Divisa_Hist  
Aux_TABAS_SabDomFer  
AUX_TABAS_Territorio  
AUX_TABAS_Territorio_Hist  
AUX_TAFTAE  
AUX_TAFTAE_NAO_EXISTE  
AUX_TPAITABAS_Divisa  
AUX_TPAITABAS_Territ  
—more—
```

Practical Example



```
Do-file Editor - Practical_Example.do
File Edit View Project Tools
Practical_Example.do Untitled1.do*
1 clear
2 set more off
3 odbc list
4 odbc query "SPAI"
5 *****
6 local year=2015
7 global path "C:\Users\eeu264\Documents\"
8
9 local query1 "select PAIDST_IDBP, PAIDST_CodDstr, PAIDST_DsgDistrito from Distrito_BPLIM"
10 local query2 "SELECT * from EntEstrut_BPLIM WHERE left(PAIENF_DataRefIni,4)='`year`'"
11 *****
12 *Loading the descriptions of district variable
13 odbc load, exec("`query1`") clear dsn("SPAI")
14 save "${path}/DST", replace
```

Creating the connection to the Data Source "SPAI"

Defining the Query statement

Loading and Saving Data

Practical Example



`. describe`

Contains data from C:\Users\eeu264\Documents\DST.dta

```
obs:      38
vars:     3
size:    1,064
```

variable name	storage type	display format	value label
PAIDST_IDBP	long	%12.0g	
PAIDST_CodDstr	str4	%9s	
PAIDST_DsgDis~o	str20	%20s	

15 Sep 2015 16:16

The screenshot shows the Stata Data Editor interface for the file DST.dta. The main window displays a list of 38 observations with columns for PAIDST_IDBP, PAIDST_CodDstr, and PAIDST_DsgDistrito. The PAIDST_DsgDistrito column contains names of Portuguese districts. The right-hand side of the window shows the 'Variables' and 'Properties' panels. The 'Variables' panel lists the three variables with checkboxes for each. The 'Properties' panel shows the details for the selected variable, PAIDST_IDBP, including its name, label, type (long), format (%12.0g), and value label.

Practical Example



Loading data from a master table with information of enterprises

```
Do-file Editor - Practical_Example.do*
File Edit View Project Tools
Practical_Example.do* Untitled1.do*
1 clear
2 set more off
3 odbc list
4 odbc query "SPAI"
5 *****
6 local year=2015
7 global path "C:\Users\eeu264\Documents\"
8
9 local query1 "select PAIDST_IDBP, PAIDST_CodDstr, PAIDST_DsgDistrito from Distrito_BPLIM"
10 local query2 "SELECT * from EntEstrut_BPLIM WHERE left(PAIENF_DataRefIni,4)='`year`'"
11 *****
12 *Loading the descriptions of district variable
13 odbc load, exec("`query1'") clear dsn("SPAI")
14 save "${path}/DST", replace
15 describe
16
17 *Importing the data from the table EntEstrut_BPLIM for the year 2015
18 odbc load, exec("`query2'") clear dsn("SPAI")
19
20 merge m:1 PAIDST_IDBP using "${path}/DST"
21 keep if _merge==1 | _merge==3
22 drop _merge
23
```

Merging information from districts dataset

Practical Example



```
. save "${path}/Data`year'", replace  
file C:\Users\eeu264\Documents\Data2015.dta saved
```

	PAIEMP_IDBP	PAIDST_IDBP	PAIDST_Cod~r	PAIDST_DsgDistrito
1	100000002	309000013	13	Porto
2	100000014	309000006	06	Coimbra
3	100000053	309000013	13	Porto
4	100000057	309000013	13	Porto
5	100000097	309000013	13	Porto
6	100000108	309000011	11	Lisboa
7	100000111	309000006	06	Coimbra
8	100000134	309000011	11	Lisboa
9	100000165	309000003	03	Braga
10	100000165	309000003	03	Braga
11	100000179	309000011	11	Lisboa
12	100000208	309000006	06	Coimbra
13	100000246	309000013	13	Porto
14	100000252	309000011	11	Lisboa
15	100000259	309000011	11	Lisboa
16	100000282	309000011	11	Lisboa
17	100000363	309000005	05	Castelo Branco
18	100000372	309000003	03	Braga
19	100000377	309000011	11	Lisboa
20	100000388	309000011	11	Lisboa
21	100000388	309000011	11	Lisboa



Writing Data

- `odbc insert` writes data from memory to an ODBC table.
- The data can change an existing table or create a new ODBC table.

```
odbc insert, table("Table_Name") dsn("DSN") insert_options
```

- ❖ `create` – create a simple ODBC table
- ❖ `overwrite` – clear data and write the data in memory to the ODBC table
- ❖ `insert` - default mode to append data in memory to the ODBC table



Some Useful Tips

- Stata keeps its entire dataset in memory. Usually it is faster but it can be a disadvantage when your dataset is very large.
 - ❖ `drop` any variables you don't need for your analysis
 - ❖ use the same variable names when you are combining datasets
 - ❖ Use `compress` to optimize variables format
 - ❖ `encode` strings
 - ❖ Use the cycles `foreach` or `forvalues` to do repeated commands
 - ❖ Use `preserve` and `restore` to make temporary changes to datasets



Final Remarks

- Stata command `odbc` allows Stata to load, write, or view data from ODBC sources.
- This command offers a useful set of function calls that can be used to access data stored in many different types of database management systems.
- Oracle, SQL Server, Access, Excel, MySQL and DB2 are some examples of available ODBC data sources.
- Stata's `odbc` is a resourceful solution that allow us to query external databases and insert or update records in those databases.

Stata Users Group Meeting

September 18th, 2015

Lisbon School of Economics & Management



Thank you for your attention

Using ODBC with Stata

Rita Sousa (rcsousa@bportugal.pt)

Bank of Portugal



Center for Mathematics and Applications

