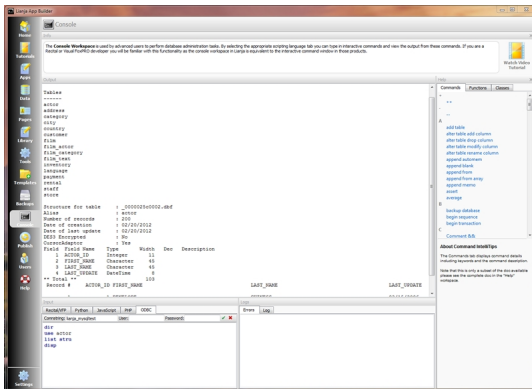


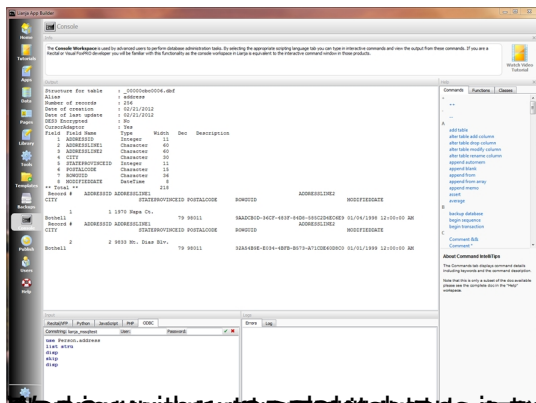
Lianja App Builder Beta6 provides connectivity to third party data sources using ODBC in conjunction with the Visual FoxPro style "CursorAdapter" class.

You can connect to ODBC data sources and work with data from MSSQL, MySQL, PostgreSQL and any other database that has an ODBC driver available.

The "Console" workspace in the Lianja App Builder now has an ODBC tab. This allows you to work interactively with external databases using familiar Visual FoxPro commands such as DIR, LIST STRUCTURE, LIST STATUS, REPLACE, DELETE, APPEND FROM etc as well as execute SQL commands against the external data source such as CREATE TABLE, ALTER TABLE, SELECT, DELETE, INSERT etc. The image below shows Lianja working with the MySQL sample sakila database.



The image below shows Lianja working with the MSSQL AdventureWorks database. Note that AdventureWorks uses schemas extensively so you have to qualify the table names with the schema that they belong to. You can see what tables belong to what schemas by querying sys.schemas and sys.tables.



Import an external data source to use for development

- Select the "Data" workspace
- create a new database e.g. sakila for MySQL
- Click the "gear" icon of the "Databases" panel
- Select "Import..."
- Choose an ODBC data source and all tables and data will be imported into your new database.

You can then build apps against the imported MySQL data.

Using the "CursorAdapter" framework class

Here is a simple example using the "CursorAdapter" class in your scripting code.

```
ca = CreateObject("CursorAdapter")
ca.DataSourceType="ODBC"
ca.DataSource="DRIVER=SQL
Server;Network=DBMSSOCN;SERVER=w2kas;database=(local);uid=sa;pwd="
ca.alias = "foo"
ca.SelectCommand="select * from foo"
ca.CursorFill()
ca.DataSource = "lianja_mysqltest"
ca.alias = "actor"
ca.SelectCommand = "select * from actor"
ca.cursorFill()
```

Notice how you can specify a complete connection string or alternatively an ODBC data source in the **datasource** property.

You can find full details of the methods and properties of CursorAdapter class [here](#) .

Using the Visual FoxPro SQLXXX() functions

The following Visual FoxPro compatible functions work against ODBC data sources in Lianja.

- SQLCONNECT()
- SQLSTRINGCONNECT()
- SQLDISCONNECT()
- SQLEXEC()
- SQLTRANSACTION()
- SQLCOMMIT()
- SQLCANCEL()
- SQLROLLBACK()
- SQLTABLES()
- SQLCOLUMNS()

Simplified data access using an explicit connection string

You can gain easy access to external data sources by specifying the "CONNSTR" clause on the "USE" command which will create a local cursor against the ODBC data source.

```
use customers connstr "odbc_data_source_name"
```

You can restrict the data that is selected into the local cursor by specifying a "WHERE" condition.

```
use customers connstr "odbc_data_source_name" where amountdue > 1000
```

You can also order the data in the local cursor by specifying the "ORDER BY" clause.

```
use customers connstr "odbc_data_source_name" where amountdue > 1000 order by  
customername
```

On completion of the USE command, the data from the external data source is now available in a local cursor. You can update records in the external data source using the REPLACE command. e.g.

```
use customers connstr "odbc_data_source_name" where amountdue > 1000 order by  
customername  
goto reccount()  
replace amountdue with amountdue - 100
```

You can also create temporary indexes against the data which now resides in the local cursor and use the SEEK command to lookup keys.

```
use customers connstr "odbc_data_source_name" where amountdue > 1000 order by  
customername  
index on custid tag custid  
seek "HA567"
```

The CURSORADAPTER() function returns a reference to the internal **CursorAdapter** for the ODBC connection. You can then call methods and get/set properties on this.

```
use customers connstr "odbc_data_source_name" where amountdue > 1000 order by  
customername  
ca = cursoradaptor("customers")  
ca.cursorRefresh()  
ca.selectcmd = "select * from customers where amountdue > 2000"  
ca.cursorRefresh()
```

to close the ODBC connection, simply execute a USE command on its own against the local cursor.

```
use customers connstr "odbc_data_source_name" where amountdue > 1000 order by  
customername  
use
```