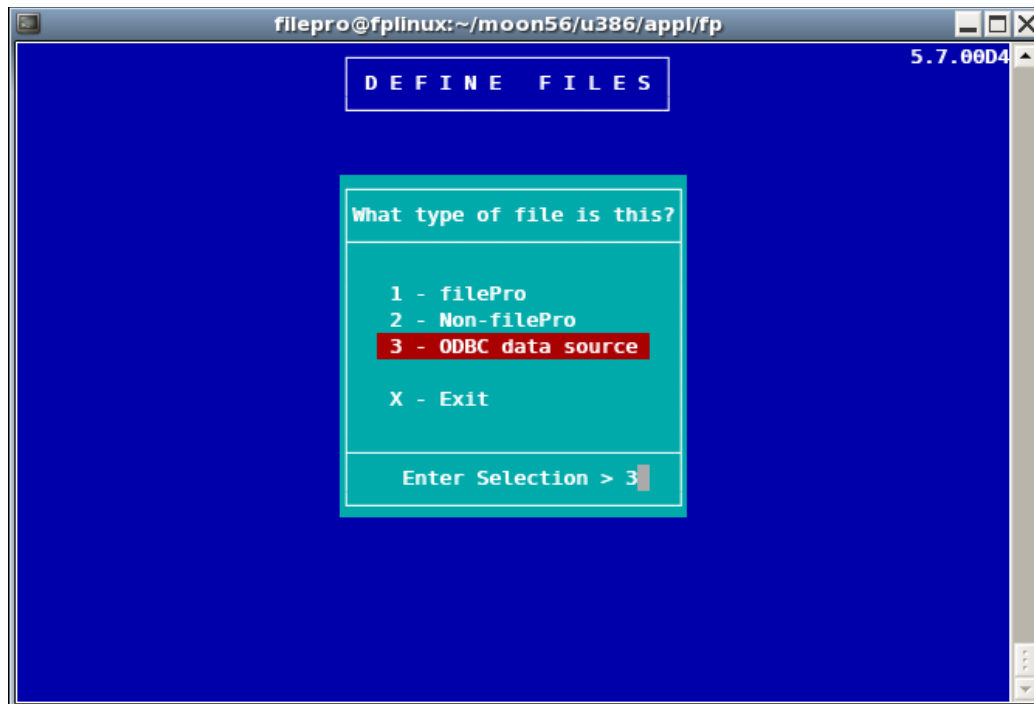


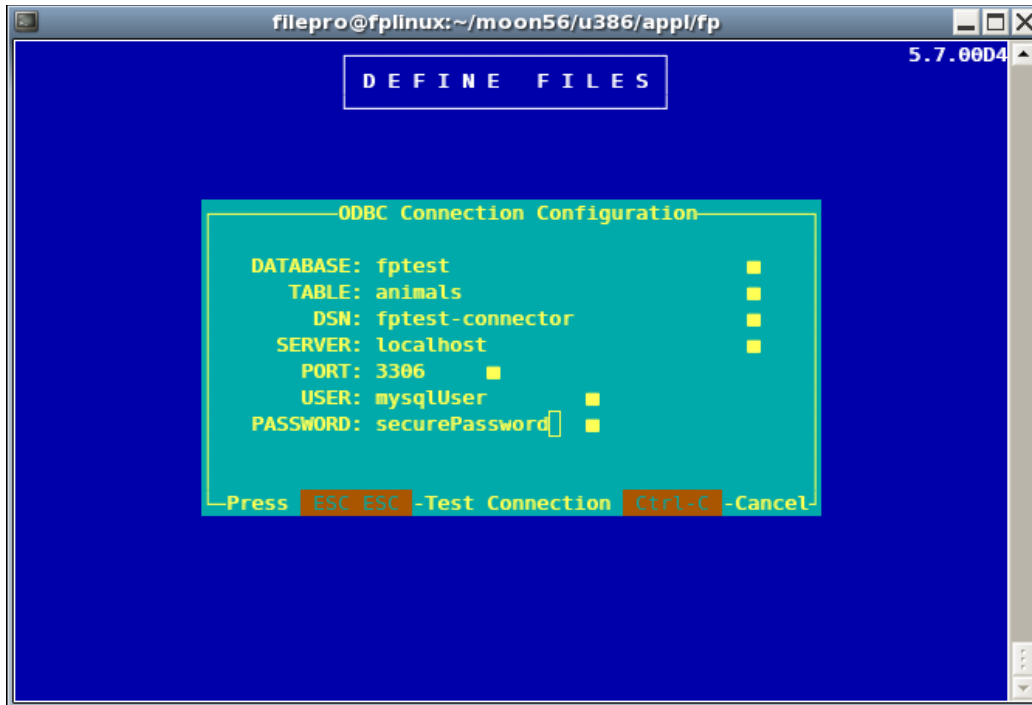
MySQL Interface

Define Files

When using "Define Files" with the MySQL version for UNIX/LINUX, you will have an additional option to Link to ODBC data sources as follows.



When selecting this option, the following screen will be presented to enter the ODBC Connection Configuration parameters to connect to a MySQL database table.



1. Information and any custom changes are set into the map file as a connection string and the connection is tested. If the connection is successful, you will see the table columns presented as fields like any other filePro file along with the field lengths and edit types.

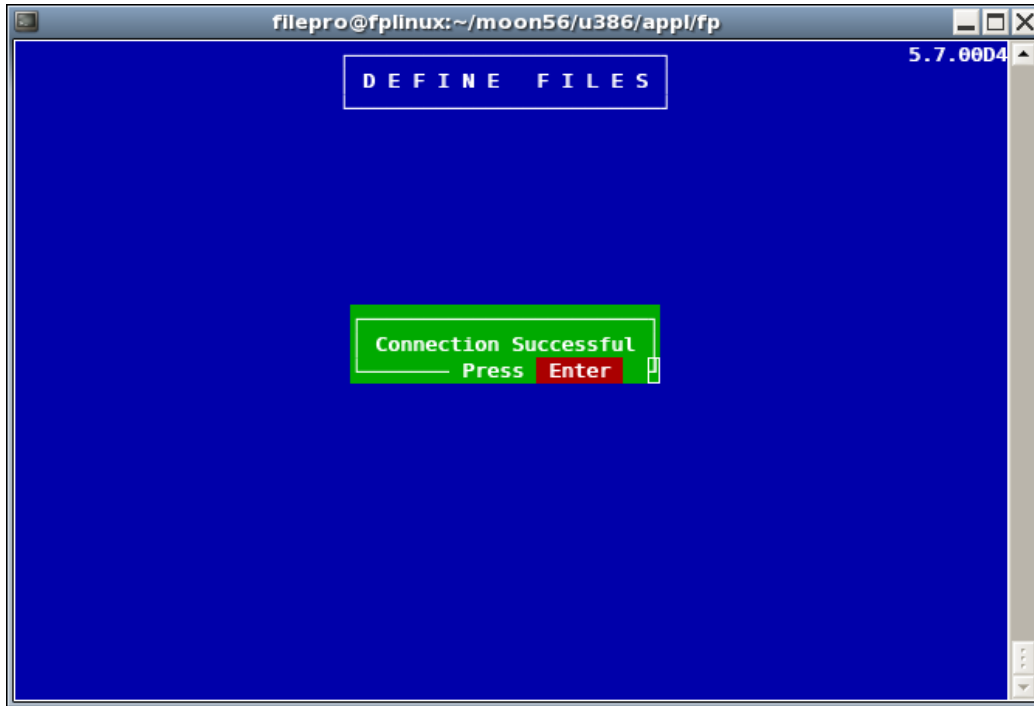
2. filePro reads DSN information from the ODBC initialization files. In most cases, these files are maintained in either the /usr/etc or /etc directory as file names "odbc.ini" and "odbcinst.ini". Refer to the unixODBC documentation for your platform for details. Your DSN entry and other parameters must match entries in these files to connect to a data source. Refer to procedures for your LINUX installation for details related to installation and use of unixODBC, MySQL ODBC Connector and MySQL and to the SCO and Skunkware archives for MySQL and unixODBC packages for OS/2 and Unixware. The MySQL ODBC Connector may have to be compiled from source code available at mysql.com. UnixODBC source code is available at unixODBC.org along with precompiled archives for various platforms.

A sample odbc.ini entry for the DSN referenced in the wizard above would be

```
[fptest-connector]
driver = MySQL
Database = fptest
Server = localhost
Socket = /var/lib/mysql/mysql.sock
User = mysql_user_name
Password = secure_password
```

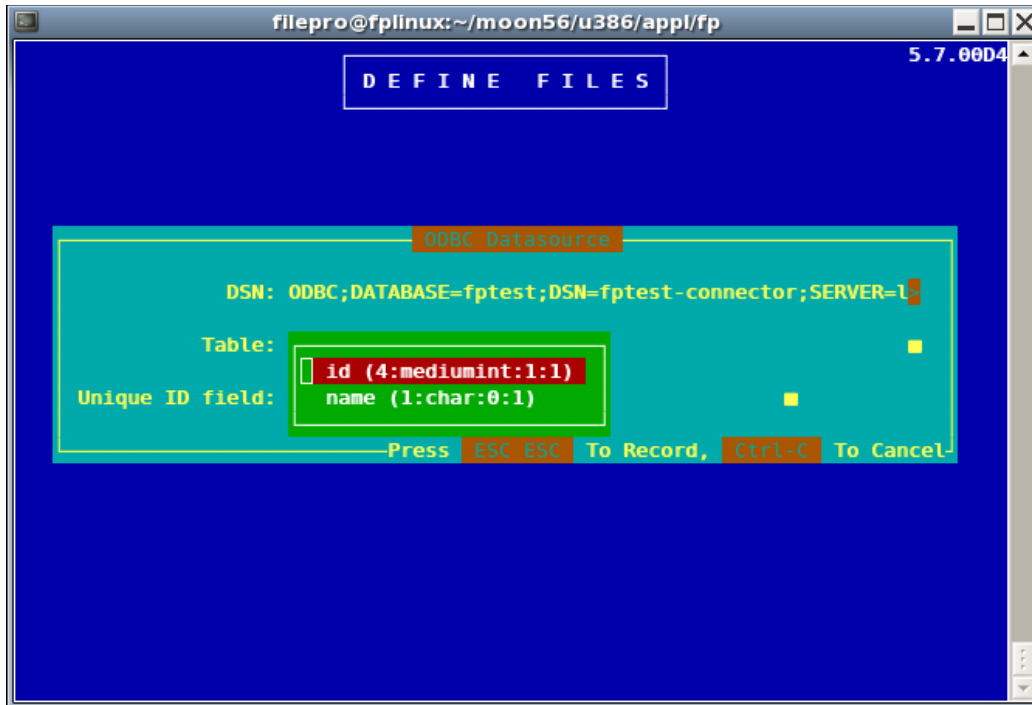
The User and Password must match those of a MySQL user. The access granted to that user for that Database must be sufficient for the functionality desired. See either the MySQL documentation, available at dev.mysql.com, or from inside MySQL issue the 'help grant' command.

3. If a successful connection to the MySQL database is made the following screen will appear



A failure to connect will display an error box showing the reason the connection failed and return you to the wizard. A failure may be caused by incorrect information in the wizard, incorrect information in the DSN entry in the `odbc.ini` file or `odbcinst.ini` file, failure to properly install any of the required components. Examples of incorrect information could be the wrong table or database name, the wrong port, specifying a user who has not been added as a MySQL user, specifying an incorrect password, or specifying a user who does not have read/write permission for the specified MySQL database.

4. Pressing enter at the Connection Successful message will bring up the ODBC Datasource Wizard where you are shown the DSN as it will be written to the filePro map and are asked to enter the Table from the previously specified MySQL database as well as the field name of the required unique id field. Pressing F6 will show a list of the fields in the MySQL table with the likely unique id field highlighted. filePro requires that every mapped MySQL table have a unique id field with the MySQL `auto_increment` attribute. Should you neglect to specify this field in this wizard or specify a field that is not `auto_increment` the records you add from filePro will be saved into the MySQL table but will not show up in filePro. Make sure that the data type you choose for this field will hold values large enough for the expected number of records. The Integer Types table in the MySQL documentation provides the information you need to make this choice. An unsigned `MEDIUMINT` type will, for example hold, over 16 million records. Signed `TINYINT` will only allow for the entry of 128 records so its use is not normally recommended unless you are certain that a particular table will never hold more than 128 records (including deleted records).



5. From this point on Define Files will proceed as it would for any filePro file. Note that the unique id field specified in step 4 will not appear in the map since it can not be assigned a value from filePro.

6. "Define Files" (ddefine) uses edit type information from the ODBC data source including Datetime and timestamp SQL types and these can be set up as a filePro date and time edits.

Inquire Update and Add

Subject to the permissions granted to the MySQL user the filePro user can

1. Browse by record number.
2. Add new records
3. Update records
4. Delete records. Deleted records do not appear on the free chain – they are not removed but can not be moved to. New records are always added to the end of the file.
5. Browse by index. Leaving the index prompt empty goes to first record in index(lowest), hitting F7 in the index prompt goes to last record in the index(highest)
6. List of odbc indexes pulled from odbc.
7. Index search prompt almost identical to regular index search prompt
First, last, next, and previous records based on an index are retrieved using a stored procedure on the mysql database. This allows use to achieve faster results and not have to open a cursor and move through a result set ordered by an index. We generate dynamic queries to limit the results we have to examine.

8. If a table has the columns: created_by (varchar(16)), created_date(timestamp), updated_by(vchar(16)), updated_date(timestamp), filePro will be able to use @cb, @c4, @ub, @u4 system fields to retrieve date from those columns. Dates are updated by the MySQL NOW() function.

9. created_by, updated_by, created_date, updated_date mysql columns are hidden from the filePro map then same way the unique identifier column is. These fields will not show up in define files. They can be accessed using @cb, @c4, @ub, @u4.

Indexes

1. MySQL index keys must be unique in order for filePro to properly traverse the index. You may have indexes with unique keys without including the unique id field but it is recommended to have the record number field (unique id field) be a column in all indexes, and that it be the last sort column in the index. Doing this allows the table to have a covering index. The record number is part of the index stored in cache and in most cases can be retrieved from memory without doing a disk read (on mysql's side). A covering index thus speeds up index searches by allowing a query to be executed without having to read the record.

2. The record number column in the table should always be unique and auto incrementing. The record number column is best supported as a SQL_INTEGER type.

Technical Notes

There are differences in the way filePro and MySQL handle numeric edits.

filePro does not enforce the decimal point and decimal values in numeric edits. For example, with the 8.2 edit, you have 8 characters available of which normally 5 characters are used for the integer portion (left of the decimal point), 1 for the decimal point and 2 characters to the right of the decimal point. However, the 8.2 edit will also accept an integer value "12345678", and values with one character to the right "123456.1" as legal values for filePro 8.2 edit.

In a MySQL decimal column, the decimal digits are reserved as part of the length. In a decimal(8,2) MySQL column, 123456.12 and 123456.1 are legal values. 1234567 will overflow. If a MySQL column is defined as decimal(M, N), there can only be M-N digits to the left of the decimal. To further complicate matters, the negative sign and decimal do not count as part of the length. Entering a number that is too large for a MySQL decimal column will set the value of that column to the highest possible number and will not create an error.

The trouble with just using the minimal MySQL decimal column to represent the same range is that values that are valid in the MySQL column would not be valid in the filePro edit. Take the filePro 8.2 edit for example. We need a decimal(10,2) to be able to represent the 8 digits to the left of the decimal as the entire number in filePro 99999999.99, 99999999.9, and 99999999 would then be valid in MySQL but only 99999999 would be valid in filePro

The following table shows the minimal MySQL decimal column to represent *at least* the same range of numbers as its filePro counterpart edit.

filePro Edit	Smallest	Largest	MySQL Decimal	Smallest	Largest
8.2	-9999999	99999999	Decimal(10,2)	-99999999.99	99999999.99
5.4	-9999	99999	Decimal(9,4)	-99999.9999	99999.9999
6.0	-99999	9999999	Decimal(6,0)	-999999	999999

Suggest changing the length of the filePro field to the total number of digits required to handle the largest number in MySQL. Note that MySQL does not consider a minus sign "-" in the total defined length.

Keep in mind that filePro will show a "/OV" in numeric fields in cases where the value length in MySQL exceeds the filePro edit length.

Special Requirements

The latest available stable unixODBC for your platform. Most Linux distribution make this available through their standard installation mechanisms. Installation archives are available for some distributions at unixODBC.org. Source code is also available at unixODBC.org. Skunkware packages are available for OSR6 and Unixware.

The latest stable MySQL ODBC Connector for your platform. Most Linux distribution make this available through their standard installation mechanisms. Installation archives are available for some distributions at mysql.org. Some platforms automatically install this connector when MySQL is installed. Source code is also available at mysql.org.

Binary copies of the the required MySQL ODBC Connector libraries for Unixware and OSR6 are also available through the fPTech developers web site.

The all.sql MySQL stored procedures supplied with the filePro installation must be inserted into every Database with a Table that filePro will map. Failure to do so will cause indexes searches to fail. This can be done by at a command prompt with the following command
`mysql --user=userName --password=userPassword databaseName < all.sql`

Current Limitations

General:

Currently only MySQL is supported as an ODBC Datasource in Unix/Linux
filePro MySQL integration is not currently available for SCO OSR5
Low level ODBC commands and functions are not currently supported

Define Files:

Fields that are deleted from MySQL tables in MySQL will not be deleted from the filePro map.
Fields that are added through MySQL will appear in the filePro map only after the file has been opened and re-saved in Define Files.

Indexes:

There currently is no support for demand indexes
There currently is no support for descending indexes
Indexes must be created in MySQL
Index keys in MySQL must be unique for filePro to properly traverse them.
Partial indexes are not supported since MySQL can not use partial indexes in ORDER BY clauses.