USING SQL WITH FIA

Andrew Colin

Flametree Technologies

Release version 1.0.0

CONTENTS

INTRODUCTION

This document describes how to interface FIA directly to an SQL database, without the need to generate or read intermediate CSV files for data transfer.

ODBC AND DSN

To allow FIA to connect to as many database types as possible, we use an *ODBC connection*.

This requires the user to set up an ODBC link on their host machine. Please see your systems administrator for details of how to do this.

Once the link has been set up, a DSN (Data Source Name) will be available for you to refer to the database. Note that a username and password may also be required to read from, and write to, the database.

DATA TABLES

FIA requires the following data to run:

- Weights and returns for portfolio
- Weights and returns for benchmark (if provided)
- Security master data
- Yield curve data
- Index data (optional)
- FX data (optional)
- Security map data (optional)
- Stress scenario data (optional)

The format of all these tables is identical to the CSV file formats used by FIA. Please refer to our wiki at www.flametreetechnologies.com for more information.

PREREQUISITES

FIA has no way of knowing how data is stored in the target SQL database. The FIA interface layer therefore requires the presence of queries (or views) in the database that provide required information in the required format. The names and fields returned by these queries are described in the following sections.

1. The following queries must be present in the target database:

Query name	Description
`FIA_security_data`	Security master data
`FIA_portfolio_data`	Weights and returns for portfolio and benchmark
`FIA_curve_data`	Yield curve data
`FIA_index_data`	Index data
`FIA_fx_data`	FX data
`FIA_map_data`	Security mapping data
`FIA_stress_data`	Stress scenario data

2. In addition, a table called `FIA_results` must be present in the target database to store FIA's results.

DATA QUERIES

The simplest way to define FIA's queries is as *pass-through* queries.

For instance, if your yield curve data is stored in a table called `tbl_curve_data` using the fields `[CurveName, CurveDate, CurveTenor, CurveYield]` then a simple query of the form

```
SELECT * FROM tbl_curve_data
```

can be used to define `FIA_curve_data`.

However, this query passes all data in the table to FIA, which may be more than is required. For instance, if a portfolio only uses one yield curve from a collection of several hundred, it will usually be worthwhile filtering all but the required curve when populating `FIA_curve_data`, so that only the required data is passed to FIA.

Consider carefully whether it is worth writing such filtered queries. FIA will ignore unused records, and is fast enough that you are unlikely to see a performance hit from passing unused data. However, it may be worthwhile restricting the data passed to FIA if you have a slow database connection.

All FIA-specific views or queries have the prefix 'FIA_'.

SECURITY MASTER DATA

FIA requires a query called `FIA_security_data` in your target database. This query must return the following fields:

Field	Data type
SecurityID	varchar[255]
SecurityName	varchar[255]
Classification	Text
EffectiveDate	Date
PricingFunction	varchar[255]
RiskFunctions	Text
CreditRating	varchar[255]
EffectiveExposure	Int
SecurityCurrency	varchar[255]
ResidualSector	varchar[255]
Curves	Text
StartDate	Date
Maturity	Date

Coupon	Real
Frequency	Int
Strike	Real
Term	Real
PSA	Real
Paydown	Real

WEIGHTS AND RETURNS

FIA requires a query called `FIA_portfolio_data` in your target database. This query must return the following fields:

Field	Data type
PortfolioDate	Date/time
PortfolioName	varchar[255]
SecurityID	varchar[255]
Weight	real
BaseReturn	real
LocalReturn	real
YTM	real
ModifiedDuration	real
Convexity	real
OAS	real
SpreadDuration	real
ZSpread	real
Price	real
Volatility	real
Delta	real
Gamma	real
Theta	real
Rho	real
Vega	real

YIELD CURVES

FIA requires a query called `FIA_curve_data` in your target database. This query must return the following fields:

Field	Data type
CurveName	varchar[255]
CurveDate	Date
CurveTenor	Real
CurveYield	Real

INDEX DATA

FIA requires a query called `FIA_index_data` in your target database. This query must return the following fields:

Field	Data type
IndexDate	Date
IndexName	varchar[255]
IndexValue	Real

Note that this query can return an empty set, if no index data is required by the selected portfolio and benchmark.

FX DATA

FIA requires a query called `FIA_FX_data` in your target database. This query must return the following fields:

Field	Data type
FXDate	Date
FXName	varchar[255]
FXValue	Real

Note that this query can return an empty set, if no exchange rate data is required by the selected portfolio and benchmark.

MAP DATA

FIA requires a query called `FIA_map_data` in your target database. This query must return the following fields:

Field	Data type
FromID	varchar[255]
ToID	varchar[255]

Note that this query can return an empty set, if no security mapping data is used.

STRESS DATA

FIA requires a query called `FIA_stress_data` in your target database. This query must return the following fields:

Field	Data type
StressLabel	varchar[255]
StressTime	Real
StressShift	Real
StressTwist	Real
StressCurvature	Real

Note that this query can return an empty set, if no stress scenario data is required.

CONFIGURATION FILE SETTINGS

To use the SQL link in FIA, several new configuration file settings have been defined.

Setting	Type	Description	Notes
DSN	String	Data source name. If this field is left blank, data is read from CSV files instead of the database, and all other database settings are ignored.	Data structure that contains the information about a specific database that an Open Database Connectivity (ODBC) driver needs in order to connect
DSN_user	String	User name to log into database	
DSN_password	String	Password to log into database	
PortfolioName	String	Name of portfolio	Passed to database so queries can return appropriate data
BenchmarkName	String	Name of benchmark (if required)	Passed to database so queries can return appropriate data

RUNNING FIA WITH A DATABASE

To read and write results from FIA into a database requires the following steps:

- Ensure a DSN has been set up
- Set up the configuration file to include the DSN name, user ID, password for the database, names of portfolio and benchmark
- Run FIA as normal

FIA will still generate Excel and CSV reports if these have been requested. However, data will be read from the database and results will be stored in the table `FIA_results`.

RESULTS

FIA stores the following quantities in table `FIA_results`:

User ID	User	
Timestamp	datetime	Date and time at which this record was created
Username	varchar[255]	Identifier for user who created this record
ResultsDate	Date	Date of the current record
Portfolio	varchar[255]	Portfolio identifier
Benchmark	varchar[255]	Benchmark identifier
SecurityID	varchar[255]	Security identifier
Risk	varchar[255]	Source of risk. In the absence of smoothing, the sum over all sources of risk for a given security at a given date will equal the base currency return.
PortfolioWeight	Real	Weight of security in portfolio at given date
PortfolioReturn	Real	Return of security due to named risk at given date
BenchmarkWeight	Real	Weight of security in benchmark at given date
BenchmarkReturn	Real	Return of security in benchmark at given date

This table contains all data needed to calculate any attribution report. For instance, to calculate the (additive) aggregated returns due to risks in the portfolio and benchmark over the entire interval, run the query

```
SELECT FIA_Results.Risk, Sum([PortfolioWeight]*[PortfolioReturn]) AS
P, Sum([BenchmarkWeight]*[BenchmarkReturn]) AS B

FROM FIA_Results

GROUP BY FIA_Results.Risk;
```

This assumes that additive smoothing has been applied.

SMOOTHING OF RESULTS

All attribution results should be smoothed to ensure path-dependence over time and over attribution effects. However, the smoothing algorithm used affects how data should be stored.

For geometric smoothing, attribution results can be combined without reference to the interval in question. The user can therefore store and use geometrically smoothed returns without further complication.

For arithmetic smoothing, the smoothing algorithm requires a measure of the total return over the current interval. Therefore smoothed returns can differ depending on the interval and whether historical results have been linked.

To put this another way, one cannot combine results calculated for two successive months, and assume that the results are the same as if the calculation had been performed over the single two-month interval. Although the totals will be consistent, the sector and security-level returns may be different.

How one copes with this is up to the user. FIA allows the user to store both smoothed and unsmoothed returns, allowing the user to calculate their own smoothing on arbitrary intervals. To assist with this, our sample database provides a stored procedure that performs Carino smoothing on arbitrary data sets.