Olsen Risk Data Extraction System (RIDE) Technical Overview

PAUL BRESLAW, THOMAS DOMENIG, MICHEL DACOROGNA

PBR.1999-08-01

Olsen Data AG
Seefeldstrasse 233, CH-8008, Zürich, Switzerland

01 September 1999
Current Version: 29 March 2001

Abstract

This paper provides a technical overview of Olsen Data's Value at Risk data delivery system. It covers how VaR data is collected, named, computed, and assembled for delivery, and outlines the various stages of the computer software for doing this.

Contents

1 Introduction **3**

2 Raw and Filtered Data **3**

3 Contributed and Computed Instruments **3**

4 Types of VaR Data **3**

4.1 Snapshot data . . . 3

4.1.1 Regular data . . . 3

4.2 High and Low data . . . 4

4.3 Tick data . . . 4

4.4 Extraction time points . . . 4

5 Snapshot Data **4**

5.1 Tick Before. . . . 4

5.2 Tick After. . . . 4

5.3 Previous Tick Interpolation. . . . 5

5.4 Linear Interpolation. . . . 5

5.4.1 Staleness and Hole Filling . . . 5

5.4.2 Non-interpolated Instruments . . . 6

6 Instrument Types in RIDE **6**

6.1 Naming . . . 6

6.2 Futures Instruments . . . 6

7 Computed Instruments **7**

7.1 Inverted FX Rates `fx-spot-inv` . . . 7

7.2 FX Cross Rates `fx-spot-cross` . . . 7

7.3 Computed FX Forward Rates `fx-fwd-comp` . . . 9

7.3.1 Cross FX Forward Rates `fx-fwd-comp-cross` . . . 9

7.4 Yield Curves . . . 9

7.5 Historical Volatilities and Correlations . . . 11

7.5.1 FX Spot Rate Input Data . . . 11

7.5.2 Interest Rate Price Input Data . . . 12

7.5.3 Interest Rate Yield Input Data . . . 12

7.5.4 Equity Index Input Data . . . 13

7.5.5 Implied Volatility Input Data . . . 13

7.5.6 RiskMetrics Volatility Model . . . 13

7.5.7 BIS volatility Model 14
7.5.8 RiskMetrics Correlation Model 14
7.5.9 BIS Correlation Model 14
7.6 Equity Betas 15

8 Data Formats **15**
8.1 Snapshot Data Format 15
8.2 High Low Data Format 16
8.3 Tick Data Format 16
8.4 Volatility And Correlation Data Format 17
8.5 Regular Data Format 17
8.6 Alternative Data Formats 17

9 RIDE Jobs **18**
9.1 Daily Jobs 18
9.2 Historical Jobs 18
9.3 Special Extractions 18

10 Client Data Collection **18**

11 Software Overview **19**
11.1 Year 2000 Compliancy 19

12 Monitoring Daily Extractions **19**

13 Reliability And Security **20**
13.1 Reliability 20
13.1.1 Data Errors 21
13.1.2 Software Errors 21
13.2 Data Archiving 21
13.3 Data Security 22

References **23**

A Variability of volatilities and correlations **24**

B Supported instrument types **25**
B.1 Contributed Instruments 25
B.2 Computed Instruments And Curves 26
B.3 Historical Volatilities And Correlations 27

1 Introduction

RIDE is a high quality delivery system of Value At Risk (VaR) data for a large range of financial instruments. Its prime purpose is to provide daily ‘best’ real and synthetic prices computed from optimally filtered high frequency data. At the same time it does not sacrifice speed and timeliness of availability. In addition to current daily data, RIDE is also used to produce historical daily prices.

2 Raw and Filtered Data

Olsen Data collect raw high frequency financial data from a number of sources, filter it in real-time, and store it. Details of the complex filtering system are beyond the scope of this document, but can be found in [Müller, 1999]. Suffice it to say here that every received tick is stored, but marked with a degree of credibility, which can subsequently be used as a selection criterion when extracting data from the databases.

Each financial instrument is stored in an individual time-series database. The same nominal instrument collected from different data suppliers is stored in separate databases, one per source. When requesting the extraction of a given instrument, time series can be merged, and ticks selected according to a large range of criteria. All the same, a naive view of an instrument is a single time-series irrespective of its source.

3 Contributed and Computed Instruments

If the price or level of a requested instrument can be drawn directly from one of the ticks stored in the time series, then this is refered to as a *contributed instrument.* However if a price has to be calculated by manipulation of one or more ticks from the same or different time series, then this is called a *computed instrument.*

4 Types of VaR Data

At the time of writing RIDE offers three types of VaR data:-

4.1 Snapshot data

This is the price of a requested instrument at or around a given *snapshot time*. For example, the bid and ask quote of a foreign exchange spot rate of US dollar against Japanese yen at 16:30 EST on a given date. The time here refers to prices quoted at that time, not necessarily the time when the observation of the database was made. Snapshot prices are always extracted from filtered data.

4.1.1 Regular data

A special form of snapshot data that is not particularly suited to daily extraction is called *regular data*. These are snapshot prices taken at regular intervals over a period of time between a few days and a few years. The interval can range from a few minutes up to one day, thus providing a sampled time series, each point of which has the same properties as a daily snapshot price.

At the time of writing regular data is only available as one–off special extractions (see *Special Extractions* below). However in the near future it is planned to implement regular data as a normal RIDE service providing the time period is limited to less than one day.

4.2 High and Low data

These are the high and low values of an instrument between two given times. The maximum time interval is 24 hours. Where the instrument is normally quoted (has both a bid and an ask price), separate highs and lows for each are provided. High and low prices are always extracted from filtered data.

4.3 Tick data

These are the high frequency contributed ticks selected according to user supplied criteria. For example all 3 month US dollar money market rates contributed by one or more named brokers. Tick data can be extracted from either filtered or unfiltered data.

4.4 Extraction time points

Apart from the snapshot time, or the time ranges for high and low or tick data, we distinguish two further time points.

- **Extraction time**. This is the time at which the observations are made and the data extracted. It is always equal to or later than the snapshot time, or the end of a timerange.
- **Collection time**. This is when the extracted data is first available for collection or delivery.

5 Snapshot Data

Because ticks almost never arrive exactly on a snapshot time, some method must be adopted for calculating the value of a time series at a specific time. RIDE offers a variety of methods, some of which involve changing the timestamp of a tick to the snapshot time. It should be noted that not all contributed instruments can be subjected to this process. For example, in the case of settlements and fixing instruments, it does not make sense to change the time. Appendix [B] lists the instrument types, and indicates whether manipulation of the time is normally performed. The presently available snapshot methods are:–

5.1 Tick Before.

Deliver the best tick immediately before the snapshot time. If there is none, return a no-data condition.

5.2 Tick After.

Deliver the best tick immediately after the snapshot time. If there is none, return a no-data condition.

5.3 Previous Tick Interpolation.

Take the best tick immediately before the snapshot time, convert its timestamp to the snapshot time, and deliver that. If there is no tick, return a no-data condition.

5.4 Linear Interpolation.

The best ticks just before and just after the snapshot time are noted. From these two, a synthetic tick is constructed whose values (where appropriate) are computed by linearly interpolating their respective times to the snapshot time. In the case of there being no tick yet after the snapshot time, then revert to either Tick Before or Previous Tick Interpolation.

A side effect of this latter rule, is that it is sometimes hard to exactly reproduce delivered data, since for a re-extraction at a later time, there could by then be a tick after the snapshot time, thus altering the results.

Where linear interpolation is required, a compromise must be made between timeliness of delivery and maximising the chance of there being a tick after the snapshot time. Clients are free to choose the gap between snapshot and extraction times according to how they view this compromise.

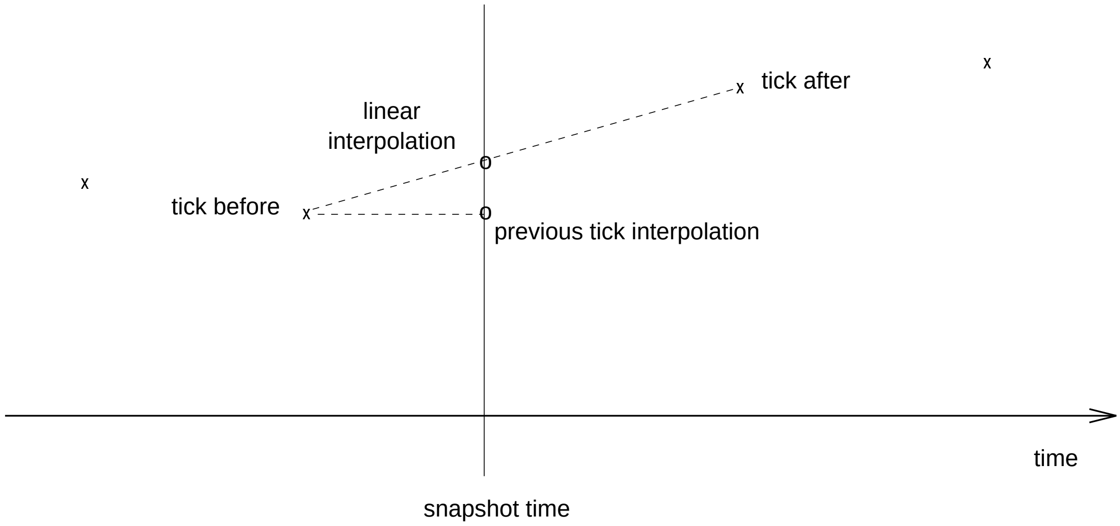

Figure 1: Interpolation styles for snapshot data

5.4.1 Staleness and Hole Filling

Linear interpolation is only meaningful if the two participating ticks are close to the snapshot time. If *both* ticks are greater than some threshold time interval from the snapshot, then they are deemed to be *stale*. The staleness interval is a user settable parameter, as is the action to be taken in this condition.

RIDE adopts the policy that hole filling is outside of its brief, since the requirements and conditions of hole filling vary from application to application. To this end, stale ticks are still reported, leaving it up to the client to decide what to do with them.

If the user sets a threshold staleness interval, and a tick is stale, then it will be returned with price or level values set to the special symbol `NaN`.

5.4.2 Non-interpolated Instruments

Whether prices are interpolated is normally decided by the user. For certain classes of instrument, however, interpolation is never applied.

- Fixings, eg. interest rate fixings (`ir-fixing`).
- Closing prices, eg. stock index closing prices (`equity-index-close`).
- Settlements, eg. of options on bond futures (`opt-bond-future-settle`).
- Benchmark bonds (`bmk-bond`) where the underlying bond maturity can change from tick to tick.

6 Instrument Types in RIDE

Daily extraction with RIDE starts from a list of instruments that a client requires. At present RIDE supports a repertoire of over 90 different kinds of financial instrument, each of which has to be specified in a manner that is comprehensible to both computer software and the people that use the resultant data.

6.1 Naming

Requests to Olsen Data's time series databases are made in a complex language called SQDADL [Beck et al., 1998]. While suited to computer programs, it is not the kind of syntax that lends itself easily to communication amongst non-IT people. To help bridge the gap between an everyday description of an instrument (eg. the spot exchange rate between US dollar and Swiss franc) and the software request to a time-series database, RIDE introduces the notion of *instrument nicknames*, which are simple enough to be understood by non-IT market professionals, yet precise enough to be translated into unambiguous SQDADL requests.

Some examples. The above spot exchange rate would be called `fx-spot_USD_CHF` in RIDE, while its SQDADL form is

```
(Time(),FX(USD,CHF),Quote(,,),Collected(RE,,),Filter(,,)).
```

Similarly, a 3 month money market rate on the Danish krona would be `ir-deposit_DKK_3m` in RIDE, and

```
(Time(),Deposit(DKK,3M),Quote(,,),Collected(RE,,),Filter(,,))
```

in SQDADL. The names `fx-spot` and `ir-deposit` are called the *instrument type*; the currencies, maturities, etc (`CHF USD 3m`) are called *instrument parameters*; while the full name eg. `ir-deposit_DKK_3m` is the *instrument identifier* or simply the *instrument.* A full list of instrument types is provided in Appendix [B].

6.2 Futures Instruments

Futures instruments are a special case. These are characterized by having a pre-determined end–point (the expiry date) after which the time–series will never be updated. A particular Futures instrument, can be described in two ways:–

- By contract expiry date. Here the instrument is defined precisely by its expiry date, eg. `bond-future-qt_USD_cbot_30y_20.12.1999`. It is then only meaningful to extract prices from this time–series after the first quotes have arrived, and before the expiry date.

- By contract position. Here the instrument is defined by the position of its contract expiry relative to some point in time – typically today. The first contract to expire relative to this point is called position 1, the second 2, etc. As contract 1 reaches its expiry date, it goes out of existence, and position 2 becomes position 1, 3 becomes 2, and so on.

  Using this scheme, instruments are named, for example:–
  `bond-future-tx_USD_cbot_30y_2` or `equity-index-future-qt_SP500_cme_1`.

  The advantage of this method is that the generalized contract can be specified without identifying which particular contracts are meant; for example 4 positions of 30y USD bond futures traded on CBOT.

  In this way, the specification of the instrument holds good for present, past and future dates, allowing historical extractions to be performed on consistent sets of positions, rolling over seamlessly from contract to contract as each expiry date is passed.

  In practice, for a given snapshot time, each requested position is translated into the corresponding expiry date to access the appropriate time–series database.

7 Computed Instruments

In this section we shall describe the various kinds of computed instruments and their methods of calculation.

7.1 Inverted FX Rates `fx-spot-inv`

Foreign exchange instruments are represented with the *per* currency first and the *expressed* currency second, ie `fx-spot_USD_JPY` means the number of yen *per* dollar. This is the way the markets normally quote these two currencies.

If there is a need for the inverted rate (ie dollars per yen), then the same time-series database must be consulted, and the prices reversed as follows. If P_{bid} is the bid price of the normally quoted instrument, P_{ask} is the corresponding ask price, and P^{inv}_{bid} is the inverted bid and P^{inv}_{ask} is the inverted ask, then

$$P^{inv}_{bid} = 1/P_{ask} \tag{7.1}$$

and

$$P^{inv}_{ask} = 1/P_{bid} \tag{7.2}$$

7.2 FX Cross Rates `fx-spot-cross`

Foreign exchange cross rates are are computed with the following formulæ. If the requested rate is equivalent to the instrument `fx-spot_XXX_YYY` and there are underlying quotes of XXX and YYY against a common third–party currency (normally USD), ie. `fx-spot_USD_XXX` and `fx-spot_USD_YYY`, then

$$c_{bid} = y_{bid}/x_{ask} \tag{7.3}$$

and

$$c_{ask} = y_{ask}/x_{bid} \tag{7.4}$$

If XXX is quoted against the third–party currency as fx-spot_XXX_USD as is the case for example with GBP, then

$$c_{bid} = y_{bid} * x_{bid} \tag{7.5}$$

and

$$c_{ask} = y_{ask} * x_{ask} \tag{7.6}$$

If YYY is quoted against the third–party currency as fx-spot_YYY_USD

$$c_{bid} = 1/(y_{bid} * x_{bid}) \tag{7.7}$$

and

$$c_{ask} = 1/(y_{ask} * x_{ask}) \tag{7.8}$$

Finally if XXX and YYY are quoted as fx-spot_XXX_USD and fx-spot_YYY_USD respectively

$$c_{bid} = x_{bid}/y_{ask} \tag{7.9}$$

and

$$c_{ask} = x_{ask}/y_{bid} \tag{7.10}$$

For a given snapshot time, the ticks for the two underlying time–series will in general have different timestamps. In historical extractions this is not a problem, since it is easy to interpolate both underlying instruments to the snapshot time, and then apply the appropriate formula above.

However if the extraction time is present time, and this is close to the snapshot time, then there is some chance that there are not yet any ticks after the snapshot time, thus eliminating the opportunity for interpolation. In these cases the underlying ticks will have different timestamps. RIDE offers four methods for handling these conditions:-

- Synchronization. Take the timestamp of the earlier of the two underlying ticks, and interpolate the later of the two to this time. Then apply one of the above formulæ, and produce a cross rate with a timestamp of the earlier tick. This method will also work correctly in circumstances when there are ticks after the snapshot time, since by virtue of interpolation, both underlying instruments will now have the same timestamp, ie. already be synchronized.

- Exactness. This method demands that both underlying instruments have the same timestamp, ie. it only produces a cross rate if there are ticks after the snapshot time for both instruments. Otherwise it produces a no-data condition.
- Slack time. This method accepts the two underlying ticks providing that both of them are within a certain period from the snapshot time. Otherwise it produces a no-data condition.
- Open. This accepts the two underlying ticks regardless of their timestamps.

7.3 Computed FX Forward Rates `fx-fwd-comp`

Synthetic FX forward rates are computed from the corresponding spot rates and the discount factors derived from money market rates or swap (interbank) yield curves depending on the required forward period. For periods up to and including one year, the corresponding money market yield converted to an annualized discount is used. For periods greater than one year, the corresponding intercept off the interest rate swap discount curve is used.

The forward rate bid and ask over a period p of currency c_2 per currency c_1 are given by $F_{bid}(c_1, c_2, p)$ and $F_{ask}(c_1, c_2, p)$. If P_{bid} and P_{ask} are the FX spot bid and ask of c_1/c_2, and $D^{c_1}_{bid}(p)$, $D^{c_1}_{ask}(p)$, $D^{c_2}_{bid}(p)$, and $D^{c_2}_{ask}(p)$ are the discount factors of their corresponding interest rates at period p, then

$$\tilde{F}_{bid}(c_1, c_2, p) = P_{bid}\left(\frac{D_{ask}(c_1, p)}{D_{bid}(c_2, p)} - 1\right) \tag{7.11}$$

and

$$\tilde{F}_{ask}(c_1, c_2, p) = P_{ask}\left(\frac{D_{bid}(c_1, p)}{D_{ask}(c_2, p)} - 1\right) \tag{7.12}$$

Because this formula can sometimes lead to rather extreme spreads, we reduce it by a user configurable percentage A (typically 50%) thus

$$F_{bid}(c_1, c_2, p) = \tilde{F}_{bid}(c_1, c_2, p) + \frac{A}{2}\left(\frac{F_{ask}(c_1, c_2, p) - F_{bid}(c_1, c_2, p)}{100}\right) \tag{7.13}$$

$$F_{ask}(c_1, c_2, p) = \tilde{F}_{ask}(c_1, c_2, p) - \frac{A}{2}\left(\frac{F_{ask}(c_1, c_2, p) - F_{bid}(c_1, c_2, p)}{100}\right) \tag{7.14}$$

7.3.1 Cross FX Forward Rates `fx-fwd-comp-cross`

Where no spot rate exists corresponding to the required forward rate, a computed cross rate is obtained using the method described above in *FX Cross Rates*.

7.4 Yield Curves

RIDE can snapshot the discount and zero coupon yield curves of interest rate swaps (`interbank-curve`), and government treasury bonds (`treasury-curve`).

Each curve is produced by first specifying the constituent underlying instruments from which it should be constructed. Normally, interbank curves are built from money market rates at the short end and interest rate swaps and the long end. Treasury curves also use money market at the short end, and treasury benchmark bonds at the long end.

The exact make up of the underlying instruments for a given curve is configurable, but the default sets provided by Olsen Data are carefully chosen to select the most liquid instruments of their class. For a requested date and time, the configured underlying instruments are snapshot using the same method for Snapshot Data above. Here again staleness criteria can be applied, and an extra level of filtering is applied to reject underliers older than some threshold.

To the set of quoted yields or bond prices are added the maturity dates and coupon rates where appropriate. All the input prices and yields are first converted to discount factors to which one or more curves are fitted using a quadratic spline method.

A discount factor D for a zero bond maturing at time t is written using some basis function $\phi(t)$ such that

$$D(t) = \sum_{j=0}^{k} \beta_j \cdot \phi_j(t) \tag{7.15}$$

where k is the number of knot points. The spline factors β_j are estimated using an ordinary least square method.

For the algorithm to work correctly, each spline segment between two knots must be populated with a sufficient number of bonds. In addition the longer term segments should have more stiffness than the shorter. To achieve this, the spline schedule is optimised with reference to the maturity date of the input bond data, by determining n, s, and x where

n is the number of spline segments with $3 \leq n \leq 10$.
s is the number of days corresponding to the first segment such that $s \geq 365$.
x is the multiplier, $x > 1$, which determines the mth knot point as sx^m.

The curve is only extended to the longest actual maturity of the underlying data, irrespective of the nominal maturity of the instruments. This applies also to the short end of the curve.

The raw spline data is represented as a series of maturity/discount–factor pairs, which is then augmented by the corresponding annualized zeros for those maturities represented by the underlying swaps or bonds respectively. If the longest actual maturity is less than its nominal maturity, a new point is added at the nominal maturity by copying the actual maturity zero, and recomputing the discount factor at this point. This guarantees that a given curve will have the same set of maturity intercepts despite daily fluctuations in long end maturities. For those short maturities whose underlying data is represented by money market rates, the spline points are replaced by the underlying data additionally converted to discount factors.

Normally separate bid and ask curves are generated from their respective underlying time series. These can be combined on request to provide a mid yield curve. A side effect of separate bid and ask curves, is that the fitted splines may occasionally cause bids to be slightly greater than asks. When this happens, the intercepts are re–calculated. First a spread is interpolated from the spreads of the two surrounding curve intercepts. This is then applied to the mid yield of the bad bid and ask, to produce a new bid and ask. If there is only one neighbouring intercept, then its spread is used directly.

Another rare occurrance is the appearance of discount factors greater than unity. Again this is due to the nature of spline fitting being unconstrained by the meaning of the values fitted. When yields are very small, as is often the case with say JPY, the curve can cross the boundary. In these cases, the bad values are discarded, and replaced by values linearly interpolated from the two closest surrounding good values. Should the bad value occur at an extreme maturity where there is only one neighbour, then its value is copied directly.

7.5 Historical Volatilities and Correlations

RIDE can compute a snapshot of the daily historical volatility of a range of underlying instrument types. At present, these consist of:–

- `fx-spot` – FX spot rates.
- `ir-deposit` – money market rates.
- `equity-index` – equity indices.
- `cap-imp-vol` – implied volatility of caps.
- `swaption-imp-vol` – implied volatility of swaptions.
- `treasury-zero` – treasury curve zero coupon yields.
- `interbank-zero` – interest rate swap curve zero coupon yields.
- `pfandbrief-zero` – German pfandbrief curve zero coupon yields.

These can be produced according to the following volatility models:–

- JP Morgan RiskMetrics model
- Basel regulatory BIS model
- GARCH11 model [1].

The same underlying data can also be used to calculate correlations within and cross–correlations between instrument classes.

All models have input data that is common for a given underlying instrument type. The formulæused for computing this input are described below.

7.5.1 FX Spot Rate Input Data

If $P_{bid}(t)$ and $P_{ask}(t)$ are the bid and ask prices of an `fx-spot` time series at a certain time t, and $P_{bid}(t-1)$ and $P_{ask}(t-1)$ are the prices at the same time one business day earlier, then the logarithmic mid price is given by

$$x(t) = 1/2(\ln P_{bid}(t) + \ln P_{ask}(t)) \tag{7.16}$$

and

$$x(t-1) = 1/2(\ln P_{bid}(t-1) + \ln P_{ask}(t-1)) \tag{7.17}$$

---

[1] At the time of writing, the GARCH model will shortly be available. However it will be limited to those currencies and instruments for which we already have parameters. Alternative parameters, and additional parameters for other currencies can be provided by RIDE customers where appropriate.

and the return $r(t)$ used as the input datum is given by

$$r(t) = x(t) - x(t-1) \tag{7.18}$$

7.5.2 Interest Rate Price Input Data

The following is true for all interest rate underlying time series represented as a percentage yield, ie `ir-deposit`, `interbank-curve` and `treasury-curve`. $R_{bid}(t)$ and $R_{ask}(t)$ are the annualised percentage bid and ask yields of an interest rate time series at a certain time t, and $R_{bid}(t-1)$ and $R_{ask}(t-1)$ are the yields at the same time one business day earlier. The maturity of the asset is broken into a (whole) *years* part and a $fraction$ (of a year) part. The logarithmic mid price is calculated from the yield as follows

$$P_{bid}(t) = \frac{1}{1 + \left(fraction * \frac{R_{ask}}{100}\right)} * \left(\frac{1}{1 + \frac{R_{ask}(t)}{100}}\right)^{years} \tag{7.19}$$

$$P_{ask}(t) = \frac{1}{1 + \left(fraction * \frac{R_{bid}}{100}\right)} * \left(\frac{1}{1 + \frac{R_{bid}(t)}{100}}\right)^{years} \tag{7.20}$$

and

$$x(t) = 1/2(\ln P_{bid}(t) + \ln P_{ask}(t)) \tag{7.21}$$

Similarly for $t-1$, and the return $r(t)$ used as the input datum is again given by equation 7.18.

7.5.3 Interest Rate Yield Input Data

To compute the yield volatility of the above interest rate assets, the mid yield R_{mid} can be calculated from the mid price $P_{mid} = (P_{ask}P_{bid})^{1/2}$, where P_{ask} and P_{bid} are given by (7.20) and (7.19), respectively. On the other hand straightforward calculation leads to the following.

If the maturity is less than 1 year

$$R_{mid} = (100/fraction + R_{ask})^{1/2}(100/fraction + R_{bid})^{1/2} - 100/fraction \tag{7.22}$$

If the maturity is integral and equal to or greater than 1 year

$$R_{mid} = (100 + R_{ask})^{1/2}(100 + R_{bid})^{1/2} - 100 \tag{7.23}$$

And so

$$x(t) = \ln R_{mid} \tag{7.24}$$

Similarly for $t-1$, and the return $r(t)$ used as the input datum is again given by equation 7.18.

7.5.4 Equity Index Input Data

If $L(t)$ is the level of an index at a certain time t, and $L(t-1)$ is the level at the same time on the previous business day, then

$$x(t) = \ln L(t) \tag{7.25}$$

Similarly for $t-1$, and the return $r(t)$ used as the input datum is again given by equation 7.18.

7.5.5 Implied Volatility Input Data

All implied volatility time series, ie. `cap-imp-vol` and `swaption-imp-vol` are computed as follows. If $P_{bid}(t)$ and $P_{ask}(t)$ are the bid and ask percentages of an implied volatility time series at a certain time t, and $P_{bid}(t-1)$ and $P_{ask}(t-1)$ are the percentages at the same time one business day earlier, then the logarithmic mid percentage is given by

$$x(t) = 1/2(\ln P_{bid}(t) + \ln P_{ask}(t)) \tag{7.26}$$

Similarly for $t-1$, and the return $r(t)$ used as the input datum is again given by equation 7.18.

7.5.6 RiskMetrics Volatility Model

The time range is defined as

$$timerange = \frac{1}{\ln(1/\lambda)} \quad \text{where } \lambda = 0.94$$

and if we accept a decay cutoff of e^{-8}, ie. less than .05%, the build–up size N is

$$N = 8 * timerange$$

This formula gives a $timerange \approx 16.2$ business days, so the build–up period is

$$N = 8 * 16.2 = 130 \text{ business days}$$

If $r(t)$ is the input datum and $\sigma(t+1)^2$ is the forecasted variance for time $t+1$ given the dataset up to time t, then

$$\sigma^2(t+1) = \lambda * \sigma^2(t) + (1-\lambda) * r^2(t) \tag{7.27}$$

and for a starting date of t_0, the initial value $var(t_0 - N)$ is given by

$$\sigma^2(t_0 - N) = r^2(t_0 - N)$$

The volatility over 1 day for a confidence level of 95% is defined as

$$V_{\text{RiskMetrics}}(t+1) = 1.65\sigma(t+1).$$

When delivered in the RiskMetrics data format, this value is multiplied by 100 to provide a percentage.

7.5.7 BIS volatility Model

If N is the build–up size, $r(t)$ is the input datum and $\sigma^2(t+1)$ is the forecasted variance at time $t+1$, then

$$\sigma^2(t+1) = \frac{1}{N-1} \sum_{i=0}^{N-1} r^2(t-i) \tag{7.28}$$

We shall use a build–up size N of 250 business days.

The volatility over 1 day for a confidence level of 95% is defined as

$$V_{\mathrm{BIS}}(t+1) = 1.65\sigma(t+1).$$

When delivered in the RiskMetrics data format, this value is multiplied by 100.

7.5.8 RiskMetrics Correlation Model

For input series $r_1(t)$ and $r_2(t)$ the covariance $c_{12}(t+1)$ is defined as

$$c_{12}(t+1) = \lambda c_{12}(t) + (1-\lambda) r_1(t) r_2(t)$$

and for a starting date of t_0, the initial value $c_{12}(t_0 - N)$ is given by

$$c_{12}(t_0 - N) = r_1(t_0 - N) r_2(t_0 - N).$$

The parameters are chosen as $N = 130$ and $\lambda = 0.94$. The correlation is defined as

$$\rho_{\mathrm{RiskMetrics},12}(t+1) = \frac{c_{12}(t+1)}{\sigma_1(t+1)\sigma_2(t+1)}$$

where σ_1 and σ_2 are defined as in (7.27) for $r_1(t)$ and $r_2(t)$, respectively.

Remark As stated earlier, the choice $\lambda = 0.94$ corresponds to a time range of 16 business days. We emphasize that this is very short for a quantity like correlation, and results in a strong variability. In the Appendix we have included a short study of the distribution of RiskMetrics volatility and correlation changes in comparison with corresponding BIS quantities. This study suggests that in order to obtain resonably stable correlation, it would be preferable to choose a time range of eg. three months for the RiskMetrics model. This corresponds to $\lambda = 0.984$. For a deeper study of properties of correlations we refer to [Lundin et al., 1998].

7.5.9 BIS Correlation Model

For input series $r_1(t)$ and $r_2(t)$ the covariance $c_{12}(t+1)$ is defined as

$$c_{12}(t+1) = \frac{1}{N-1} \sum_{i=0}^{N-1} r_1(t-i) r_2(t-i)$$

with $N = 250$. The correlation is defined as

$$\rho_{\mathrm{BIS},12}(t+1) = \frac{c_{12}(t+1)}{\sigma_1(t+1)\sigma_2(t+1)}$$

where σ_1 and σ_2 are defined as in (7.28) for $r_1(t)$ and $r_2(t)$, respectively.

7.6 Equity Betas

The Beta is a measure of the expected change in equity return $r(t)$, given a change in the local market index return $r_I(t)$. As a general reference see [Elton and Gruber, 1994].

Equity betas are computed using *annualised returns*, which are defined as

$$r_{ann}(t) = (\ln P(t) - \ln P(t-1)) \cdot \sqrt{365.25} \tag{7.29}$$

The covariance between an equity i and the index I is computed as an exponential moving average

$$Cov_{iI}(t) = (1-\mu) \cdot r_{i,ann}(t) \cdot r_{I,ann}(t) + \mu \cdot Cov_{iI}(t-1) \tag{7.30}$$

Using the analogous formula for the variance of the index,

$$Var_I(t) = (1-\mu) \cdot r_{I,ann}(t)^2 + +\mu \cdot Var_I(t-1)$$

we finally define the Beta as

$$\beta_{iI} = \frac{Cov_{iI}(t)}{\max[\varepsilon,\ Var_I(t)]}$$

where $\varepsilon = 10^{-10}$ is a small constant. Note that β_{iI} is not the same thing as the correlation coefficient, unless the variance of the equity happens to coincide with that of the index.

For the current implementation we use a decay factor $\mu = 0.94$. To initialize the system, we set

$$\begin{gathered} Cov_{iI}(0) = r_{i,ann}(0)^2 \\ Var_I(0) = r_I(0)^2 \end{gathered}$$

8 Data Formats

When data is first extracted in the RIDE system it is held in a format called RIDE internal format, and subsequently transformed into the final client format. The internal format is used to archive customer data, and as input for a number of Olsen Data's internal maintenance tools.

RIDE supports a variety of client data formats depending on the type of data and customers' specific requirements. All data types have a corresponding *Olsen standard format* which is the recommended final format. Standard formats are in ASCII, line oriented to assist legibility for human use, yet machine readable by simple text processing software.

Dates are always written as `DD.MM.YYYY` and times as `hh:mm:ss`. In high frequency data, times can additionally be written to micro–second resolution as `hh:mm:ss.uuuuuu`.

8.1 Snapshot Data Format

Snapshot data typically consists of a single price or level for each of a large number of instruments. The standard format for this kind of data consists of newline terminated records (lines), one for each instrument. Where the instrument has more than one price (eg. bids and asks), these are presented as separate records.

Each record is self–describing, consisting of a number of white–space separated fields made up of

```
<type> <params> ...  <value type> <date> <time> <value>
```

for example

```
ir-deposit CAD 3m bid 30.08.1999 02:29:27 4.7538
ir-deposit CAD 3m ask 30.08.1999 02:29:27 4.8788
```

The `type` field is the RIDE instrument type. The *params* fields are the same as the RIDE instrument parameters in the same order, but broken out to be space–separated. The *value type* can be one of

```
ask  bid  close  coupon  fixing  level  settlement  transaction
```

Records beginning with the # character should be treated as comments. Future versions of RIDE may include a commented header giving information about the circumstances of the extraction. Blank lines should be ignored.

The *date* and *time* fields refer to the observed time of the price, not the snapshot time which anyway is constant for all records in a dataset. The time zone is GMT by default, but can be set to any recognized timezone. Clients frequently ask for times to be given in the same timezone as the requested snapshot time,

Normally the snapshot time is recorded in the name of the dataset, which consists of one file per extraction, called something like `RiskData_YYYYMMDDhhmmss`.

8.2 High Low Data Format

The standard format for high low data follows the same principes as snapshot data with the following pattern

```
<type> <params> ...  <value type> <date> <time> <high> <date> <time> <low>
```

for example

```
fx-spot GBP CHF bid 27.08.1999 11:38:40 2.4319 27.08.1999 10:23:35 2.4285
fx-spot GBP CHF ask 27.08.1999 11:38:40 2.4349 27.08.1999 10:23:36 2.4315
```

8.3 Tick Data Format

Because tick data contains many data points per instrument, a different style of data format is used. Each instrument is stored in a separate file consisting of a #–commented header followed by a number of price records each of a line of ASCII text. The header has the following style

```
# Ride Tick Format: <version>
# Format: plain
# Instrument: <instrument name>
# Fields: <price/value field names>
```

for example

```
# Ride Tick Format: 1.0
# Format: plain
# Instrument: fx-spot_EUR_USD
# Fields: date time bid ask
```

Data records then follow using the format specified in the `Fields` header, eg.

```
27.08.1999 07:25:42.402497 1.0455 1.0457
27.08.1999 07:25:44.419861 1.0452 1.0458
27.08.1999 07:25:46.429671 1.0456 1.0459
27.08.1999 07:25:49.424214 1.0456 1.0466
```

The recommended name for a tick data file is `<instrument>_YYYYMMDD`

8.4 Volatility And Correlation Data Format

While historical volatilities and correlations are really just another kind of snapshot, their data require additional information and hence another format. Again the principle follows that of snapshot data with the following pattern

```
<type> <params> ...  <model> <xform> <conf> <mkt> <tz> <date> <time> <value>
```

for example

```
fx-hist-vol EUR USD BIS LogMidPrice 95 European GMT 30.08.1999 14:58:01 0.978078
```

`Model` is the volatility model used in the calculation, modified by `xform` describing the kind of transformation applied to the input data. `Conf` is the confidence level in volatility calculations – with correlations it should be ignored. The `mkt` field is the market with reference to which historical daily data was used to prime the model; and similarly `tz` describes the time zone to lock daily samples to the same time of day.

File naming follows the convention for snapshot data.

8.5 Regular Data Format

Regular snapshot data uses the same format as tick data except that times are represented without micro–seconds.

8.6 Alternative Data Formats

Although standard formats are the preferred way to represent data, Olsen Data can offer a bespoke data formatting service to meet any particular client's needs. As a result of such special requests in the past, RIDE can support the following additional formats for snapshot data:–

- Infinity Panorama format.
- Algorithmics comma separated values (csv) format.
- FX Protocol (FXP) format.
- JP Morgan RiskMetrics format (volatilities and correlations only).

9 RIDE Jobs

Data extraction and computation perfomed by RIDE is divided into *jobs*, each one representing a single client's requested data. A job is defined as a set of requested instruments, an extraction time, which days it should run, information about the kind of data to produce (snapshot, etc), how the data is to be formatted and possibly post-processed (for example compressed or encrypted), and finally how it is to be made available to the client.

9.1 Daily Jobs

Where clients are interested in more than one market, there may be multiple jobs per day, possibly with different instruments per job. It is also possible to run a job only on specific days of the week.

The end result of a job is one or more data files in some agreed–upon format. Where a client has multiple jobs, data can be post-processed so that there is only one daily file or package of files.

Setting up the job, scheduling and supervising its execution, are all handled by the *RIDE scheduler*. This constructs a series of calls to lower level RIDE tools at the appropriate extraction time specified in each client's job description. It also makes its activities available to continuous monitoring by tools such as the *RIDE monitor*. Should any stage in an extraction fail, the scheduler notifies Olsen Data Operations staff via e–mail.

9.2 Historical Jobs

Historical data is viewed as a series of daily jobs. The only difference between today's job and one in the past is the date. The RIDE software is able to run clients' jobs between any two dates in the past. The resulting data is simply a series of daily files identified by date. These can be packaged together for loading into clients' Value at Risk engines with exactly the same software as is used for each day's data.

9.3 Special Extractions

Sometimes it is useful to extract data for a subset of a client's instruments, or even just a single instrument, perhaps one that the client does not presently collect. This might be to replace some lost data, or to compare the difference between historical and 'at the moment' extractions. It can also be that Olsen Data have improved the filtering of some instrument class, and the client wishes to compare the 'before' and 'after' effects of the new filter.

The RIDE software makes it particularly easy to perform this kind of special extraction, for a given date or dates, or over some period of time. Formatting and delivery follow the usual convention for the client in question, though the file names are created by mutual agreement, so as not to conflict the the names of daily data.

10 Client Data Collection

Collection from Olsen Data's FTP server, rather than delivery by e–mail, is the preferred way of getting daily RIDE data to customers. There are a number of reasons for this:–

- The volume of daily data often exceeds what is manageable in a single e–mail message.

- Delivery of e–mail is difficult to verify. With a collection system, the customer can confirm that the data has been collected, and Olsen Data can trace logins onto their FTP server.
- In general a collection system has greater security. If e–mail were undeliverable, it would remain on an intermediate IP host where it could be tampered with.
- Since the data remains in Olsen Data's control, failure by the customer to collect at one point in time, can easily be remedied by collecting the data later, with no administrative overhead. Re-sending e–mail, on the other hand, poses considerable administrative difficulties.
- There is some indeterminacy with delivery times of e–mail. With a collection system, Olsen Data undertakes to have the data available at a certain time each day, so the client knows exactly when it can be collected.

However, in special cases where the volume of daily data is very small, and the client is prepared to accept the potential unreliability and insecurity of e–mail, RIDE can deliver data through this medium.

11 Software Overview

RIDE is written as a family of layered software tools. At the lowest level are *extractors* for the primary contribued data types – snapshot, high–low, and tick data; these address the Olsen Data data repository directly, and typically produce a value for just one instrument. The next level tools are *high–level extractors*, which use the output of the lower level to produce computed data, such as volatilities, yield curves, computed FX, etc. Except for the volatility engine, these too act on a single instrument. Above these are driver tools which map instrument request lists to invocations of the appropriate extractors for each instrument, collating results and handling errors.

All these relatively low level tools operate in the timezone of the primary databases (GMT). Requests for snashot times in other time zones are translated into GMT by supervisory tools such as the RIDE scheduler.

Once the data has been extracted into RIDE internal format, it is simultaneously archived and processed by the appropriate formatter. Here, as well as direct formatting, the raw GMT time stamps can be mapped back into the client's requested timezone. The final post–processing stage involves collating, compressing, and encrypting according to individual requirements.

11.1 Year 2000 Compliancy

RIDE software is written in C^{++} and Perl. All of the software is Y2K compliant. It runs on Sun Microsystems' Solaris (Unix) operating system which is also certified Y2K compliant.

12 Monitoring Daily Extractions

Olsen Data runs a number of monitoring tools related to the data repository and RIDE extractions. Some of these relate specifically to individual customers, while others are more general:–

- The *Database Monitor* watches datafeeds for all the time series being collected and sends warnings if statistically too long passes between sucessive raw ticks.

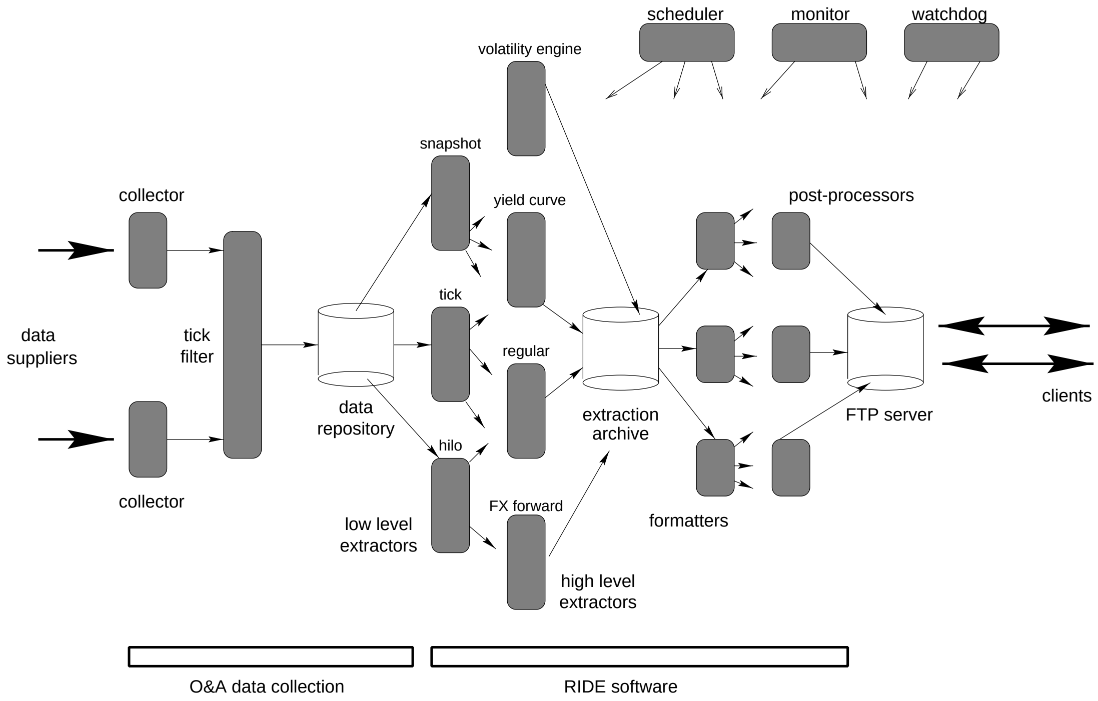

Figure 2: Data flow in collection and extraction

- The *RIDE Monitor* provides a real–time update on the configuration and progress of all registered RIDE jobs, and alerts an operator if a job fails or runs late.
- The *RIDE Watchdog* analyses RIDE jobs after they have run and produces a report indicating those instruments for which no data was produced, or whose prices were stale. The watchdog can be configured to send reports via e–mail to individual customers. However at the time of writing its output requires careful knowledge of the markets to be of much use, and hence is not generally available. This situation will improve over time as the program is given more intelligence about particular market conditions.
- The *RIDE Instrument Warning System* runs price comparisons on successive days of all collected time–series, and tries to spot unusually large changes. These are passed on to market experts who can distinguish normal market conditions from major market changes (eg. re–valuations), or possible filtering problems.

13 Reliability And Security

This section deals with RIDE’s features for ensuring a reliable and secure daily VaR service.

13.1 Reliability

Daily jobs can fail, or partially fail, for a number of reasons. These have to be considered individually, distinguishing errors caused by data conditions from those caused by environmental software and hardware conditions.

13.1.1 Data Errors

Because of the atomic nature of Snapshot, Tick and HighLow data, failure to extract individual instruments does not cause complete job failure. Rather, the single instrument will report a no data condition. This will show up in the daily administrative watchdog run to be handled by Olsen Data support staff. If the instrument in question has gone dead for market reasons, then this is not really an error at all, and nothing more can be done than look for alternative sources for this information, or simply remove the instrument from the customer's requirements.

On the other hand, if the error is due to a collection problem, for example Reuters might have changed the name of a RIC, then this is easily fixed by re-configuring the database collector, with no serious interruption in service.

Volatility and correlation data is different because of the all–or–nothing nature of correlations – one cannot program a matrix with invalid entries. In these cases, the offending instrument will be temporarily eliminated from the client's list, and the whole job will be run again. This is a rare occurrance, but past experience has shown that it results in data being delivered a few hours late.

A more general point about mathematical modelling, is the fact that computers by their nature cannot represent all possible numbers. Since a model can conceivably produce a number that is too small or too large to be represented, these cases become `Infinity` or `NaN`. Whether they are delivered or not depends on the ability of the data format to accept these exceptional values. Olsen Data standard format can handle them, but RiskMetrics format, for example, cannot.

Yield curves are sensitive to staleness of the underlying instruments. If the 25y and 30y GBP benchmark bonds become illiquid for a period of time, they will no longer participate in the corresponding treasury curve, which will appear shorter than expected. This can lead to a certain irregularity in the curve history, of which clients should be aware.

13.1.2 Software Errors

All software breaks sometime. This can be due to internal bugs or unexpected environmental conditions. Experience with RIDE has shown the latter to be the more common cause of errors. The atomic nature of extraction coupled with the separation of the stages of processing into independent activities, makes recovery relatively simple. Again we can learn much from past experience of problems. The most important tools are those which monitor the progress of daily extractions.

If, for example, copying to the FTP server is disrupted because of a local network failure, other parts of the job proceed successfully up to that point. The RIDE scheduler notifies Olsen Data Operations staff giving exact details of what went wrong, and instructions how to proceed with the job once the problem has been fixed. It is then not necessary to restart the job from scratch, rather it can be continued from the stage where the error occurred.

13.2 Data Archiving

Data on the FTP server remains for a maximum of 7 days before being deleted. This allows a customer to regain lost or damaged data within a week of the extraction date without intervention by Olsen Data staff.

Raw daily extractions for every RIDE client are archived on–line on the RIDE machine. At present there is no limit to the length of archived history. Archives are maintained in RIDE internal format, rather than the customer's final output format. This makes it easy to run general purpose analysis and monitoring software on all customer data.

Should a client require a copy of some old daily data, rather than a historical re–run of the extraction, then the archived raw data can be formatted and post–processed accordingly.

13.3 Data Security

All Olsen Data's internal computers are shielded from the Internet by established firewall technology. In addition, the computers used to store time–series databases and to run RIDE software are insulated from other Olsen Data internal machines by attachment to a separate IP sub–net. Access to the RIDE machines from within the Olsen Data network is password protected and limited to a small number of authorised account holders. Client data files are produced on the same machine as the database repository and then pushed to the FTP server. While the RIDE machine can access the FTP server, the reverse is not the case.

Access to the FTP server from within Olsen Data is limited in the same way as the RIDE machine. Access to the FTP server from outside Olsen Data is via password protected individual FTP login accounts. Each account holder has a personal directory from which data files can be collected. The directory is read–only for the account holder. Once an FTP account holder is logged in, only the collection directory belonging to the account holder is visible. It is not possible to change to another directory.

References

[Beck et al., 1998] **Beck D., Bowen D., and Meissner C.**, 1998, *A high-frequency data repository for financial time series*, Internal document DAB.1998-03-27, Olsen & Associates, Seefeldstrasse 233, 8008 Zürich, Switzerland.

[Elton and Gruber, 1994] **Elton E. J. and Gruber M. J.**, 1994, *Modern Portfolio Theory and Investment Analysis*, John Wiley & Sons, Singapore, 4th edition.

[Lundin et al., 1998] **Lundin M. C., Dacorogna M. M., and Müller U. A.**, 1998, *Correlation of high frequency financial time series*, Internal document MCL.1998-01-26, Olsen & Associates, Seefeldstrasse 233, 8008 Zürich, Switzerland.

[Müller, 1999] **Müller U. A.**, 1999, *The O&A filter for data in finance*, Internal document UAM.1999-04-27, Olsen & Associates, Seefeldstrasse 233, 8008 Zürich, Switzerland.

A Variability of volatilities and correlations

For this small study we chose the three FX rates USD_CHF, USD_JPY and GBP_USD, sampled daily at 12:00 MET from 8.1.1995 to 30.7.1999. Reserving the first 250 business days for buildup, the testing period starts on 2.1.1996, leaving a sample size of 926 business days.

We computed the frequency of daily correlation changes $\rho_t - \rho_{t-1}$ over threshholds of 5, 10, 15 and 20 percent. For volatilities relative daily changes $(\sigma_t - \sigma_{t-1})/\sigma_{t-1}$ were recorded; taking log-changes $\log\sigma_t - \log\sigma_{t-1}$ led to similar results. The tables below show eg. that correlation changes exceeding 5 percent occur 16 to 19 percent of the time for RiskMetrics with $\lambda = 0.94$, while for $\lambda = 0.984$ it only happens between 0.75 and 2.7 percent of the time. Note that the results for relative changes in volatilitites are quite similar in order of magnitude to the results for changes in correlations.

daily change in %	5	10	15	20
RiskMetrics $\lambda = 0.94$				
CHF-JPY	16.00	3.6	1.8	0.64
CHF-GBP	16.00	6.3	2.3	0.86
GBP-JPY	19.00	7.5	3.8	1.80
RiskMetrics $\lambda = 0.984$				
GBP-JPY	0.75	0.11	0.00	0.00
GBP-JPY	1.40	0.32	0.00	0.00
GBP-JPY	2.70	0.32	0.11	0.00
BIS 250 days				
GBP-JPY	0.21	0.0	0.0	0.00
GBP-JPY	0.00	0.0	0.0	0.00
GBP-JPY	0.21	0.0	0.0	0.00

Table 1: Absolute correlation changes

relative change in %	5	10	15	20
RiskMetrics $\lambda = 0.94$				
CHF-JPY	11.0	5.10	2.40	1.50
CHF-GBP	10.0	5.40	3.40	1.70
GBP-JPY	11.0	5.40	3.20	1.50
RiskMetrics $\lambda = 0.984$				
CHF-JPY	1.5	0.43	0.21	0.21
CHF-GBP	2.1	0.96	0.43	0.21
GBP-JPY	2.1	0.43	0.11	0.00
BIS 250 days				
CHF-JPY	0.43	0.00	0.0	0.0
CHF-GBP	0.64	0.11	0.0	0.0
GBP-JPY	0.21	0.00	0.0	0.0

Table 2: Relative volatility changes

B Supported instrument types

B.1 Contributed Instruments

`bmk-bond`	Quotes on benchmark bonds
`bond-future-qt`	Quotes on bond futures
`bond-future-settle`	Settlement prices on bond futures
`bond-future-tx`	Transaction prices on bond futures
`brady-bond`	Quotes on Brady bonds
`cap-imp-vol`	Quotes on implied volatility of caps
`commodity-future-qt`	Quotes on commodity futures
`commodity-future-settle`	Settlement prices on commodity futures
`commodity-future-tx`	Transaction prices on commodity futures
`equity-index`	Levels of equity indices
`equity-index-close`	Closing levels of equity indices
`equity-index-future-qt`	Quotes on equity index futures
`equity-index-future-settle`	Settlement prices on equity index futures
`equity-index-future-tx`	Transaction prices on commodity futures
`equity-qt`	Stock/equity quotes
`equity-tx`	Stock/equity transaction prices
`floor-imp-vol`	Quotes on implied volatility of floors
`fx-fixingqt`	Quotes of FX fixings
`fx-fwd`	Quotes of FX forward rates
`fx-imp-vol`	Quotes of implied volatility of FX
`fx-spot`	Quotes of FX spot rates
`ir-deposit`	Quotes of cash interest rates
`ir-fixing`	Levels of interest rate fixings
`ir-future-qt`	Quotes on interest rate futures
`ir-future-settle`	Settlement prices on interest rate futures
`ir-future-tx`	Transaction prices on interest rate futures
`ir-swap`	Quotes on interest rate swaps
`opt-bond-future-qt`	Quotes of options on bond futures
`opt-bond-future-settle`	Settlement prices of options on bond futures
`opt-ir-future-qt`	Quotes of options on interest rate futures
`opt-ir-future-settle`	Settlement prices of options on interest rate futures
`pfandbrief-curve`	Levels of contributed Pfandbrief zero and yield curves
`pfandbrief-qt`	Quotes of Pfandbriefe
`pfandbrief-yield`	Pfandbrief yield curve intercept
`pfandbrief-zero`	Pfandbrief zero curve intercept
`swaption-imp-vol`	Quotes of implied volatility of swaptions
`term-index`	Levels of term indices

B.2 Computed Instruments And Curves

`fx-fixingqt-cross`	Computed cross quotes of FX fixings
`fx-fwd-comp`	Computed quotes of FX forward rates
`fx-fwd-comp-cross`	Computed quotes of FX forward cross rates
`fx-fwd-comp-inv`	Computed quotes of inverted FX forward rates
`fx-fwd-curve-comp`	Curves of computed quotes of FX forward rates
`fx-fwd-curve-comp-cross`	Curves of computed quotes of FX forward cross rates
`fx-fwd-curve-comp-inv`	Curves of computed quotes of inverse FX forward rates
`fx-spot-cross`	Computed quotes of FX cross rates
`fx-spot-inv`	Quotes of inverse FX spot rates
`interbank-curve`	Curves of computed quotes of interbank interest rates
`interbank-discount`	Intercepts of computed interbank discounts
`interbank-zero`	Intercepts of computed interbank zero yields
`pex-curve`	Curves of computed PEX quotes
`rex-curve`	Quotes of computed REX zero, yield and discount curves
`treasury-curve`	Computed quotes of zero and discount treasury curves
`treasury-discount`	Intercepts of computed treasury discounts
`treasury-zero`	Intercepts of computed treasury zero yields

B.3 Historical Volatilities And Correlations

`cap-corr`	Quotes on implied volatility of caps
`cap-hist-vol`	Quotes on implied volatility of caps
`cap-interbank-corr`	Caps with swap zero yield curves
`cap-pfandbrief-corr`	Caps with pfandbrief zeros
`cap-swaption-corr`	Caps and swaption quotes
`cap-treasury-corr`	Caps with treasury zero yield curves
`equity-beta`	Levels of equity beta
`equity-index-corr`	Equity indices
`equity-index-hist-vol`	Equity indices
`equity-vola`	Equities
`fx-corr`	FX spot rates
`fx-hist-vol`	FX spot rates
`fx-index-corr`	FX spot rates with equity index levels
`fx-ir-corr`	FX spot rates with IR cash rates
`interbank-pfandbrief-corr`	Interbank zeros with pfandbrief zeros
`interbank-treasury-corr`	Interbank zeros with treasury zero yields
`interbank-zero-corr`	Interbank zeros
`interbank-zero-hist-vol`	Interbank zeros
`ir-cap-corr`	Cash deposit rates with quotes on implied volatility of caps
`ir-corr`	Cash deposit rates
`ir-hist-vol`	Cash deposit rates
`ir-index-corr`	Cash deposit rates with equity index levels
`ir-interbank-corr`	Cash deposit rates with interbank zero yields
`ir-pfandbrief-corr`	Cash deposit rates with pfandbrief zeros
`ir-swaption-corr`	Cash deposit rates with quotes on implied volatility of swaptions
`ir-treasury-corr`	Cash deposit rates with treasury zero yields
`pfandbrief-zero-corr`	Pfandbrief zeros
`pfandbrief-zero-hist-vol`	Pfandbrief zeros
`swaption-corr`	Quotes of implied volatility of swaptions
`swaption-hist-vol`	Quotes of implied volatility of swaptions
`swaption-interbank-corr`	Swaptions with interbank zeros
`swaption-pfandbrief-corr`	Swaptions with pfandbrief zeros
`swaption-treasury-corr`	Swaptions with treasury zeros
`treasury-pfandbrief-corr`	Treasury zeros with pfandbrief zeros
`treasury-zero-corr`	Treasury zeros
`treasury-zero-hist-vol`	Treasury zeros