

# Algorithmica Research AB

## Database tool for Quantlab 3.1

**User's manual**

Last update: 2016-06-15

© 2009-2017 Algorithmica Research AB. All rights reserved.

## Contents

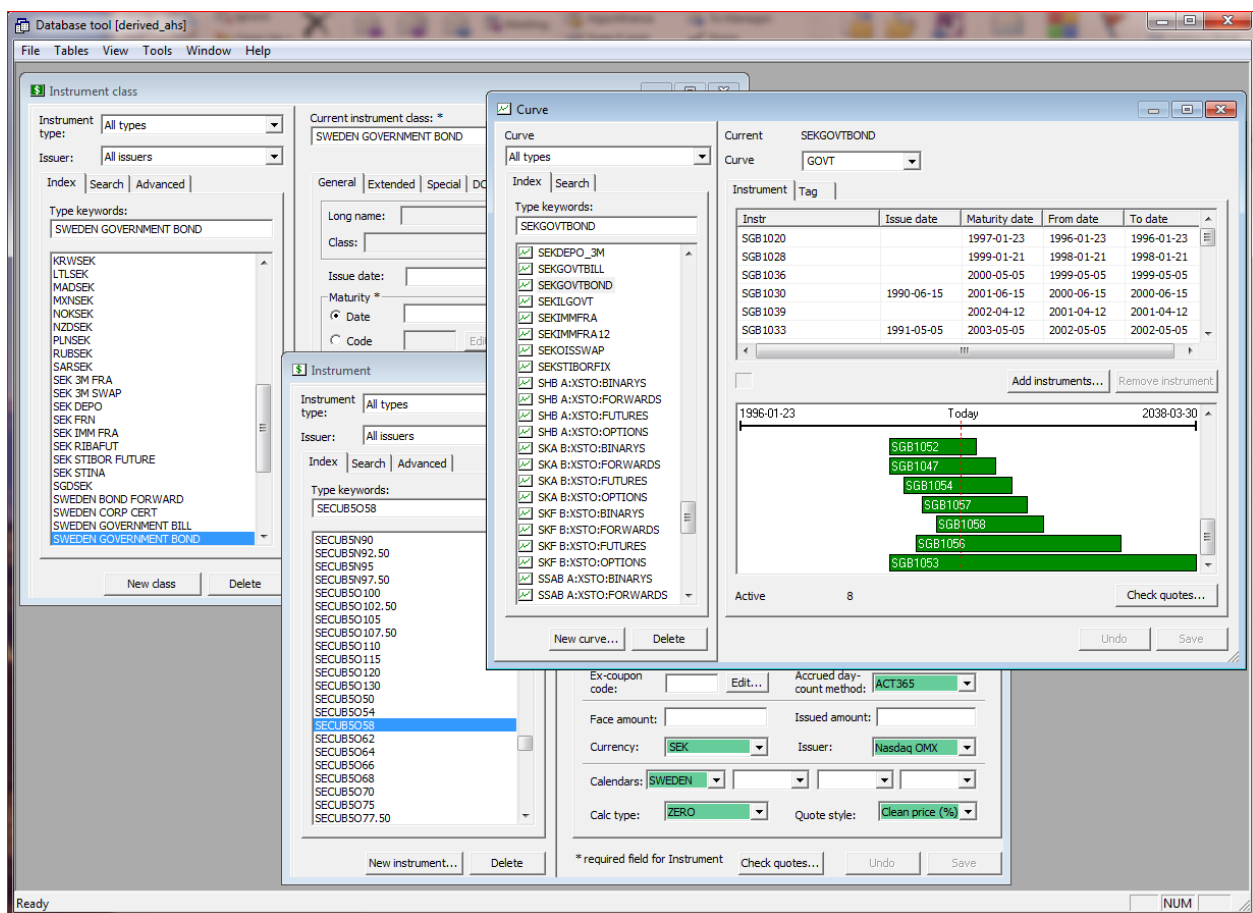
<b>1</b>	<b><i>Introduction to the Database tool</i></b>	<b>4</b>
<b>2</b>	<b><i>Work environment</i></b>	<b>5</b>
<b>3</b>	<b><i>Instruments</i></b>	<b>6</b>
<b>3.1</b>	<b>Instrument classes</b>	<b>6</b>
3.1.1	What is a class?	6
3.1.2	Adding, editing and deleting	7
<b>3.2</b>	<b>Instruments</b>	<b>8</b>
3.2.1	Adding, editing and deleting instruments	8
3.2.2	Quick example	9
3.2.3	Changing real time link data	10
3.2.4	Check historical quotes	11
<b>3.3</b>	<b>Field definitions for instruments</b>	<b>11</b>
<b>3.4</b>	<b>Coupon fixings for Floating Rate Notes</b>	<b>11</b>
<b>3.5</b>	<b>Instrument validation settings</b>	<b>11</b>
<b>3.6</b>	<b>Adding extra instrument properties</b>	<b>12</b>
<b>4</b>	<b><i>Curves</i></b>	<b>15</b>
<b>4.1</b>	<b>What is a curve?</b>	<b>15</b>
4.1.1	Combined curves	15
<b>4.2</b>	<b>Adding, editing and deleting curves</b>	<b>16</b>
4.2.1	Check for missing data	17
4.2.2	Having multiple date ranges	17
4.2.3	Adding instruments to a curve	18
<b>5</b>	<b><i>Holiday calendars</i></b>	<b>19</b>
<b>5.1</b>	<b>Adding, editing and deleting holiday calendars</b>	<b>19</b>
5.1.1	Creating a new market and setting holidays	20
<b>6</b>	<b><i>Indices</i></b>	<b>21</b>
<b>6.1</b>	<b>Adding, editing and deleting indices</b>	<b>21</b>
<b>7</b>	<b><i>Appendix 1. Database tables</i></b>	<b>23</b>
7.1.1	Tag tables	23
7.1.2	The QuoteSource table: Extending the Quote table and adding real-time quote fields	24
<b>7.2</b>	<b>Example of multiple real time sources</b>	<b>28</b>

<b>8</b>	<b><i>Appendix 2: Instrument definition</i></b>	<b>29</b>
	General instrument information	29
	Extended instrument information	35
	Special instrument information	38
	FRN information	38
	Real time link information	40
<b>9</b>	<b><i>Appendix 3: To define instruments using instrument classes</i></b>	<b>42</b>
<b>10</b>	<b><i>Appendix 4. CalcMethod attributes</i></b>	<b>43</b>
	Bonds	43
	Inflation Linked Bonds	45
	Treasury Bills	46
<b>11</b>	<b><i>Appendix 5. Bond definition examples</i></b>	<b>48</b>
	Generic example: USTreasury 1	48
	Generic example: USTreasury 2	49
	Generic example: USTreasury 3	51
	Example: Swedish Treasury	53
	Example: Spanish Treasury	54
	Example: Bund	55
	Example: OATi	56

# 1 Introduction to the Database tool

The Database tool is a common database administration platform for MarketWatch, Quantlab and History Server. By using the Database tool, information such as instrument static data, real time instrument identification codes, and holiday calendars can be managed and shared from a central data source.

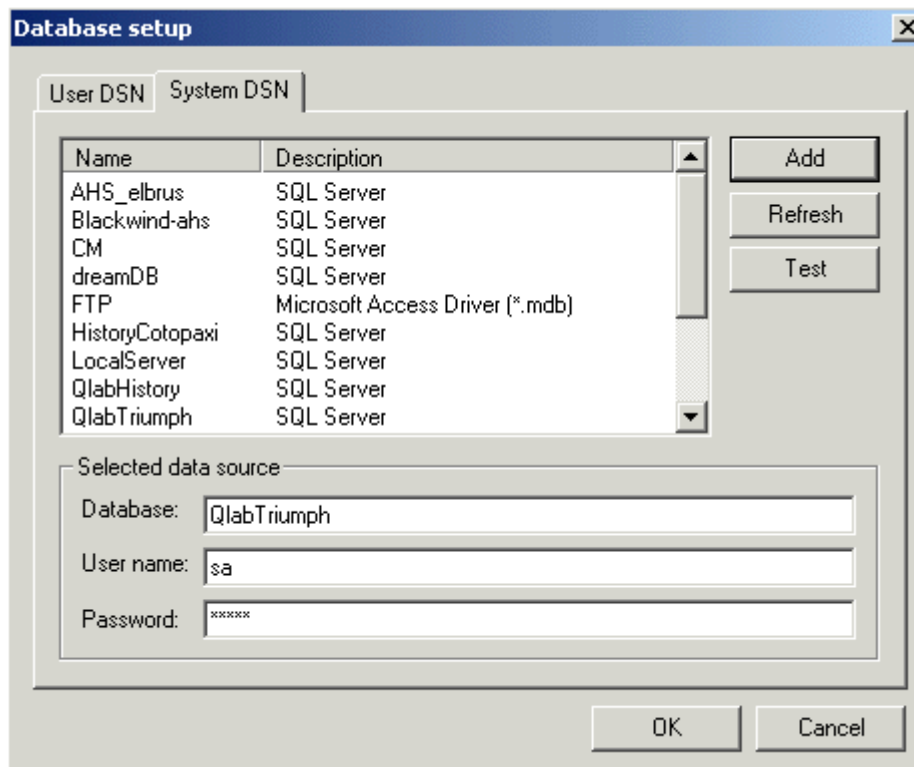
Having a common platform for administration of supportive data will drastically reduce time for maintenance. Corrections made will automatically come to the benefit of the whole organization. Wizards and templates will enable fast instrument and curve maintenance.



## 2 Work environment

On opening the database tool, a dialog asking the user to choose current database will open. If the ODBC source is not yet configured, pressing Add will take the user into Windows ordinary environment for ODBC sources. (Some users may not have permission to add or configure the ODBC source, and then IT-support must be involved before continuing.)

If the correct ODBC source is set-up for the database it can be selected and by pressing OK the database will be on-line.



*Database source selection show on start-up*

## 3 Instruments

Currently, the Database tool support instruments used by MarketWatch and Quantlab. Future releases of Quantlab will support a more extensive range of instrument classes and derivatives and so will the Database tool.

### 3.1 *Instrument classes*

The instrument class dialog can be found in the menu Tables | Instrument class or by pressing Alt+2.

#### 3.1.1 What is a class?

Instrument classes are templates used to minimize maintenance of information that many objects share. More specifically, an instrument class represents the common information for a group of instruments.

---

**Note!** A class can be viewed as "fall back" information. This means that if a piece of information is unknown or missing for a specific instrument, and the instrument has a class defined, the missing information is taken from the class instead. Class information is not saved to the instrument itself. As soon as an instrument have any field filled in with information this will override the class information.

---

When setting up a new class one must consider which information the instruments have in common and whether to provide certain information at the instrument level instead. General advice on this subject is given in Appendix 3: To define instruments using instrument classes.

Let's look at a typical instrument class question.

We want to add government bonds and bills to the instrument database. In fact, the bonds and bills come from some different markets as well.

At this point we find that bonds and bills are different enough to render their own classes. Of course this is due to the fact that calculation conventions differ between bonds and bills. Further we identify that bonds and bills from different markets are too different to yield their own classes as well.

Our class library will look like this:

1. Swedish government bonds
2. Swedish government bills
3. etc.

### 3.1.2 Adding, editing and deleting

The screenshot shows the 'Instrument class' dialog box. On the left, there's a list of instrument classes under the 'Index' tab. The list includes various Swedish government bonds and corporate certificates. 'SWEDEN GOVERNMENT BOND' is selected. On the right, the details for the selected class are shown. The 'Current instrument class' is 'SWEDEN GOVERNMENT BOND'. The 'General' tab is active, showing fields for 'Long name', 'Class', 'Type' (set to 'FIXED CPN BON'), 'Issue date', 'Maturity' (set to 'Date'), 'Coupon' (set to 'BD4'), 'Ex-coupon code' (set to 'BD4'), 'Face amount' (set to '100'), 'Currency' (set to 'SEK'), 'Calendars' (set to 'SWEDEN'), 'Calc type' (set to 'SEKGOVT'), 'First coupon date', 'Settle' (set to 'BD2'), 'Coupon freq' (set to '1'), 'Accrued day-count method' (set to 'EU30360'), 'Issued amount', 'Issuer' (set to 'Riksgäldskontore'), and 'Quote style' (set to 'Yield'). There are 'New class' and 'Delete' buttons at the bottom left, and 'Undo' and 'Save' buttons at the bottom right. A note at the bottom states '\* required field for Instrument'.

*Instrument class dialog*

The left side of the dialog lists the current classes. Extensive search functions are available when the list of classes extends.

The right hand side of the dialog displays the instrument class details. Available parameters in the class are identical to those of the instrument itself.<sup>1</sup>

---

**Note!** Fields marked with \* - required fields - does not apply to the class, but rather to the instrument itself in order for Quantlab to perform calculations correctly. This implies that when adding a new instrument the required fields must exist in either the class or the instrument itself.

---

Important fields are placed in the 'General' tab. Less frequently used fields are found under the 'Extended' tab.

---

**Tip!** By carefully adding as many fields as possible to the class, all instruments connected to the class will benefit by being fully specified.

---

---

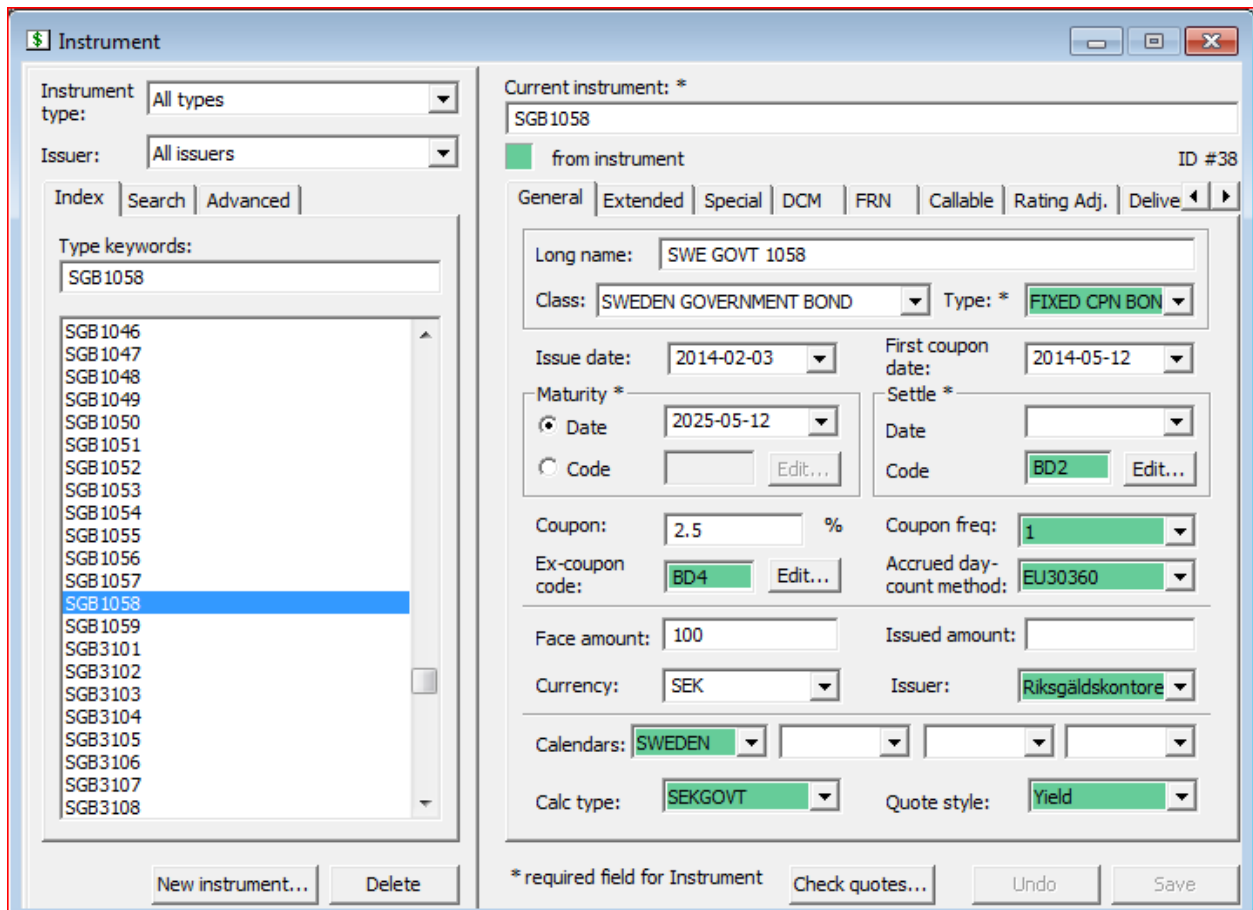
<sup>1</sup> With the exception that the instrument has a reference to the instrument class.

For help on filling in fields and valid choices see the section about Field definitions. This section applies to both instrument classes and instruments.

## 3.2 Instruments

### 3.2.1 Adding, editing and deleting instruments

If you have not read about how instruments and instrument classes work, it is recommended that you start with the instrument class section before you add any instruments to the database.



The screenshot shows the 'Instrument' dialog box. On the left, there is a list of instruments with 'SGB1058' selected. The right pane shows the configuration for 'SGB1058'. The 'Current instrument' is 'SGB1058'. The 'Long name' is 'SWE GOVT 1058'. The 'Class' is 'SWEDEN GOVERNMENT BOND' and the 'Type' is 'FIXED CPN BON'. The 'Issue date' is '2014-02-03' and the 'First coupon date' is '2014-05-12'. The 'Maturity' is '2025-05-12' and the 'Settle' is 'BD2'. The 'Coupon' is '2.5 %' and the 'Coupon freq' is '1'. The 'Ex-coupon code' is 'BD4' and the 'Accrued day-count method' is 'EU30360'. The 'Face amount' is '100' and the 'Issued amount' is empty. The 'Currency' is 'SEK' and the 'Issuer' is 'Riksgäldskontore'. The 'Calendars' are 'SWEDEN' and the 'Calc type' is 'SEKGOVT'. The 'Quote style' is 'Yield'. The bottom of the dialog has buttons for 'New instrument...', 'Delete', 'Check quotes...', 'Undo', and 'Save'.

*Instrument dialog with example of Swedish Government bond*

Just as in the instrument class dialog the left section lists all instruments with extensive search capabilities on the top. By activating a specific instrument in the list, its information will display on the right.

Any information that is not saved for the instrument itself will "fall back" on the parameter definitions of the selected instrument class. The corresponding fields are shown with a green background color. If the information shown with normal white background color the information is saved for the instrument itself.



---

**Speed tip!** By right clicking on the instrument name in the list box you can copy the instrument. By default the new instrument will be named as “Copy of ...” the previous instrument name. The new name can be entered in the “Current instrument” field.

---

The instrument dialog is divided into the following five tabs:

<b>General</b>	Displays the most important fields for an instrument and all required fields <sup>2</sup>
<b>Extended</b>	Includes the less frequently used fields for specific instrument types.
<b>Special</b>	Includes special purpose fields for certain instruments like index-linked bonds using a CPI reference.
<b>Frn</b>	Includes special purpose fields for floating rate notes. See also 3.4 for FRN coupons.
<b>Tag</b>	
<b>Realtime link</b>	For each instrument, the settings used to connect to the real time feed and historical time series data.
<b>Comments</b>	A free text field for comments about a specific instrument.

### 3.2.2 Quick example

Let's take an example where we already have specified instrument classes sufficient to cover our new instrument needs.

We want to add a newly issued benchmark German government bond. What will be unique for this bond?

1. Press the 'New instrument' button on instrument dialog.
2. Enter the short name for the new bond (used in most displays in Quantlab and MarketWatch).
3. Choose the correct instrument class for this bond.
4. Skip the long name because it's not required.
5. Add the maturity date e.g., 2005-05-05 (unique for this bond).
6. Add the coupon size e.g., 5.5% (unique for this bond).
7. Add the issue date e.g., 2002-05-05.
8. Quick look at the green fields taken from the class. Ok? Yes – then first part done.

---

<sup>2</sup> An exception is the real time link fields, which are all required for Quantlab to accept the instrument.

Since we're doing a fast-track instrument insert, we completely skip the 'Extended' tab fields. Last part of add-new-instrument process is to define the how the instrument should be linked to the real time feed.

1. Switch to realtime link tab.
2. Press 'Add' button in the center right dialog.
3. In field **Ric** – add the name of the instrument as it should be referenced in the historical database. Default for Quantlab is the name used in the Quote table.
4. In the field **Realtime ric**, add the identifier in the real time feed you are using. If a contributor is used this will be set later.
5. **From date** – we set 2002-02-05\*
6. **To date** – we set 2005-05-05\*
7. **Bid code** – for the Reuters feed this is 22 or 393 depending on the market.
8. **Ask code** – for the Reuters feed this is 25 or 275 depending on the market.
9. **Feed** – name of the feed commonly used is IDN\_SELECTFEED.
10. **Quote table** – name of the table where the historical values are stored – normally "Quote".
11. Press insert. -> the instrument will appear as with blue background indicating it has not yet been saved to the database.
12. Since we skip the comments tab, we are now done.
13. Press save.

---

**Speed tip!** Many instruments have identical settings in the real time link dialog. It is possible to save a preset for the most common choices. Press the “Create preset” button to store a preset for realtime link information. Next time around, only the Ric and Realtime ric will be necessary together with the preset.

---

### 3.2.3 Changing real time link data

In the Real time link tab, double clicking on the cell where the information is displayed will open it for changes. This will enable the user to change already stored data. Before the data is saved to the database it is shown with a blue background. To commit the data to the database – press 'Save' at the bottom of the dialog.

---

\* The from- and to date are used to indicate for which period the ric and realtime ric are valid. For some instruments having rolling rics, more than one row is needed to specify the complete period. For this simple example the ric is always valid and hence we choose the 2002-02-05 start date and the 2005-05-05 end date.

### 3.2.4 Check historical quotes

By pressing the 'Check quotes' button, it is possible to check the historical database for missing data for the selected instrument. The resulting table can be copied and saved to for example MS Excel for further examination. Correct holiday calendars will be applied by using the instrument's market definitions.

## 3.3 Field definitions for instruments

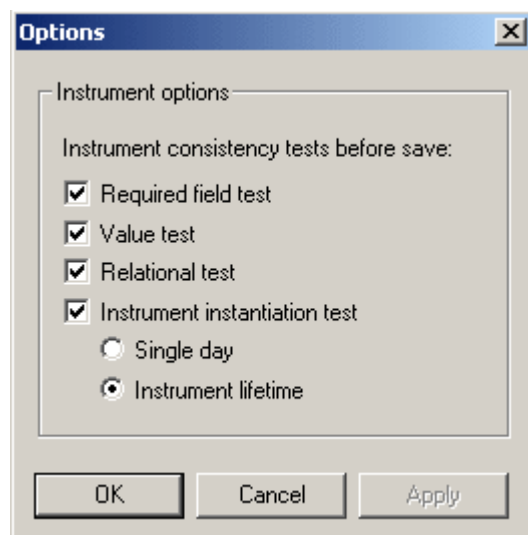
Please refer to the Appendix 1 for specifications on the instrument fields to fill in. In Appendix 4, there are some examples from large bond markets and Appendix 2 contains guidelines about what data to store in the Instrument Class. In Appendix 3 there is detailed information about the calculation methods.

## 3.4 Coupon fixings for Floating Rate Notes

In the menu Table | FRN Coupon it is possible to set fixings for all FRN bonds in the database. You can search for FRN using the filter option to the left and then set the fixing rate for all your selected instruments. The fixed margin is added to the fixing rate to give the coupon rate. It is also possible to edit each fixing manually in the table to the right.

## 3.5 Instrument validation settings

When adding new instruments it can sometimes be difficult to verify if all fields are filled in correctly. There are numerous ways to try to validate the correctness of the instrument specification. Some user defined validation levels can be set using the menu Tools | Options.



*Instrument validation settings*

The first three validation options are straightforward.

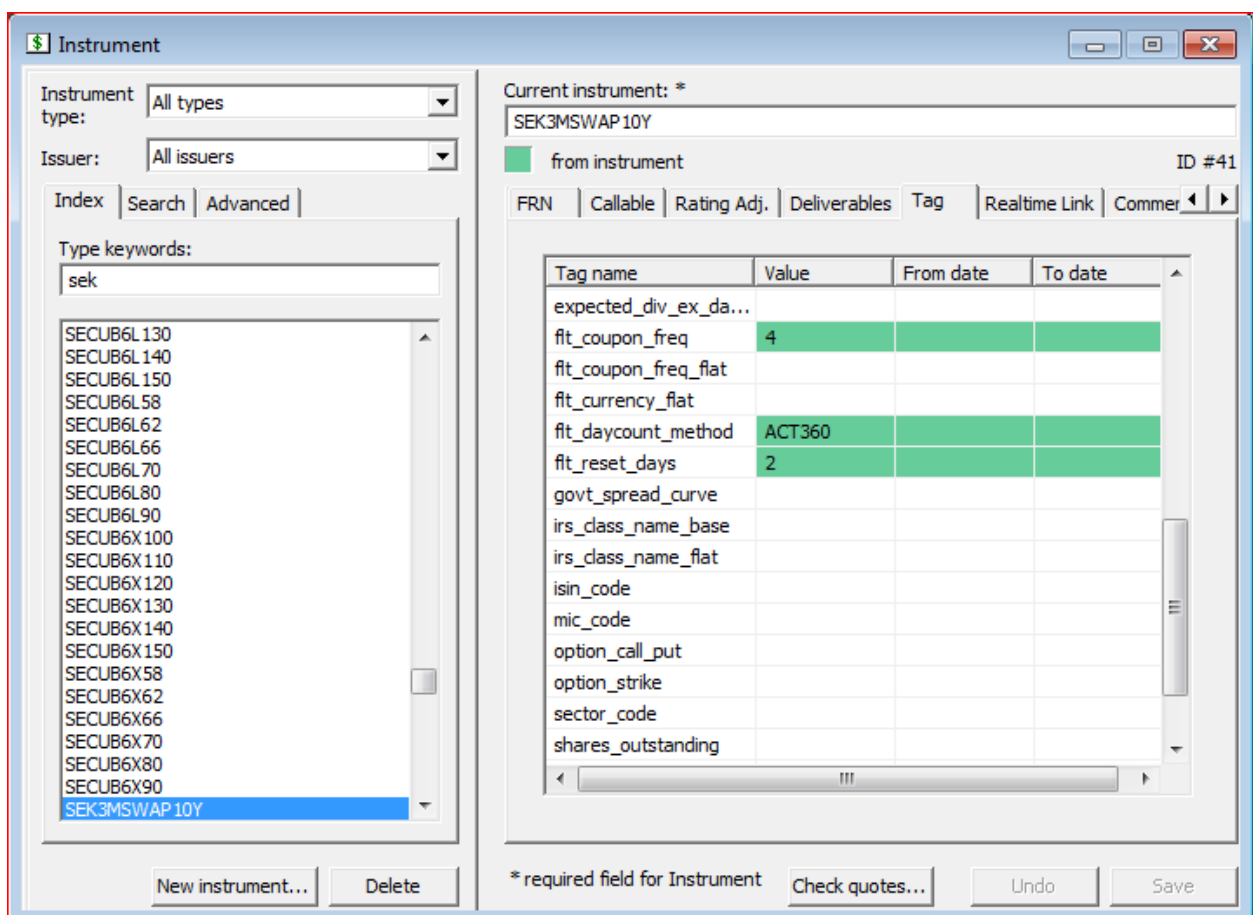
Test	Explanation
Required field test	Test that no required fields are missing (e.g. 'Calc type' must be filled in)
Value test	Test that all numeric fields are accurate (e.g. 'Coupon' $\geq 0$ )

Relational test	Test that important relations are accurate (e.g. 'Issue date' < 'Maturity date')
Instrument instantiation test	Simulate an instrument instantiation to find possible run-time errors. This can be performed on a single day or on the instruments entire life span.

The last test refers specifically to the use of the instruments in either MarketWatch or the Quantlab applications. For an instrument to instantiate correctly it must be a part of the instrument universe currently supported by these applications.

### 3.6 Adding extra instrument properties

In general, all settings for instruments are handled in dialog boxes either in the class or directly for the instrument as shown above. However, some new instruments require additional properties in order to function correctly. As an interim user-interface, the tag table is used. It can be found as a tab on both class-level as well as directly on the instrument.



Tag table showing extra settings for an interest rate swap, floating leg.

A user can also add tags as user-defined properties freely. By opening Tables | Tag... a list of all currently added tags appear.

Note that as this is a list of both instrument specific tags and user-defined tags, the list of properties will differ from installation to installation.

The screenshot shows a 'Tag' dialog box with a list of tags on the left and a table of values on the right. The 'Current' tag is 'flt\_coupon\_freq'. The 'Tag type' is 'number' and the 'Key type' is 'instrume'. The table lists the following keys and values:

Key	Value
SEK 3M FRA	4
SEK 3M SWAP	4
SEK IMM FRA	4
SEK STINA	365

The list of tags on the left includes:

- ZC\_ADD\_TO\_BASECURVE
- ZC\_IGNORE\_ITEM
- ann\_extra\_div\_amount
- ann\_extra\_div\_ann\_date
- ann\_extra\_div\_ex\_date
- ann\_extra\_div\_pay\_date
- ann\_ord\_div\_amount
- ann\_ord\_div\_ann\_date
- ann\_ord\_div\_ex\_date
- ann\_ord\_div\_pay\_date
- bench\_spread\_curve
- capital\_adj\_factor
- contract\_size
- curve\_underlying\_name
- default\_disc\_curves
- delivery\_code
- delivery\_style
- derivative\_type
- div\_currency
- div\_date\_freq
- exercise\_style
- expected\_div\_ex\_dates
- flt\_coupon\_freq
- flt\_coupon\_freq\_flat
- flt\_currency\_flat
- flt\_daycount\_method
- flt\_reset\_days
- govt\_spread\_curve
- irs\_class\_name\_base
- irs\_class\_name\_flat
- isin\_code
- mic\_code
- option\_call\_put
- option\_strike
- sector\_code
- shares\_outstanding
- swap\_spread\_curve
- underlying\_name
- underlying\_type

Buttons at the bottom include 'New tag...', 'Delete', and 'Save'.

List of tags listing the members having information associated with the property *flt\_cpn\_freq*.



## 4 Curves

### 4.1 What is a curve?

A curve defines which instruments that should be included on a given date when calculating zero coupon curves or similar financial metrics. Usually the user wants to define a curve with only instruments having certain liquidity or benchmark status. Further, the user might want instruments maturing within a certain period to leave the curve because they are not longer considered liquid. This makes the curve a user defined entity and must be specified explicitly in the Database tool.

Instead of saving information about which instruments are on a curve for every given business day, the system will store the dates between which the instrument is on the curve.

#### 4.1.1 Combined curves

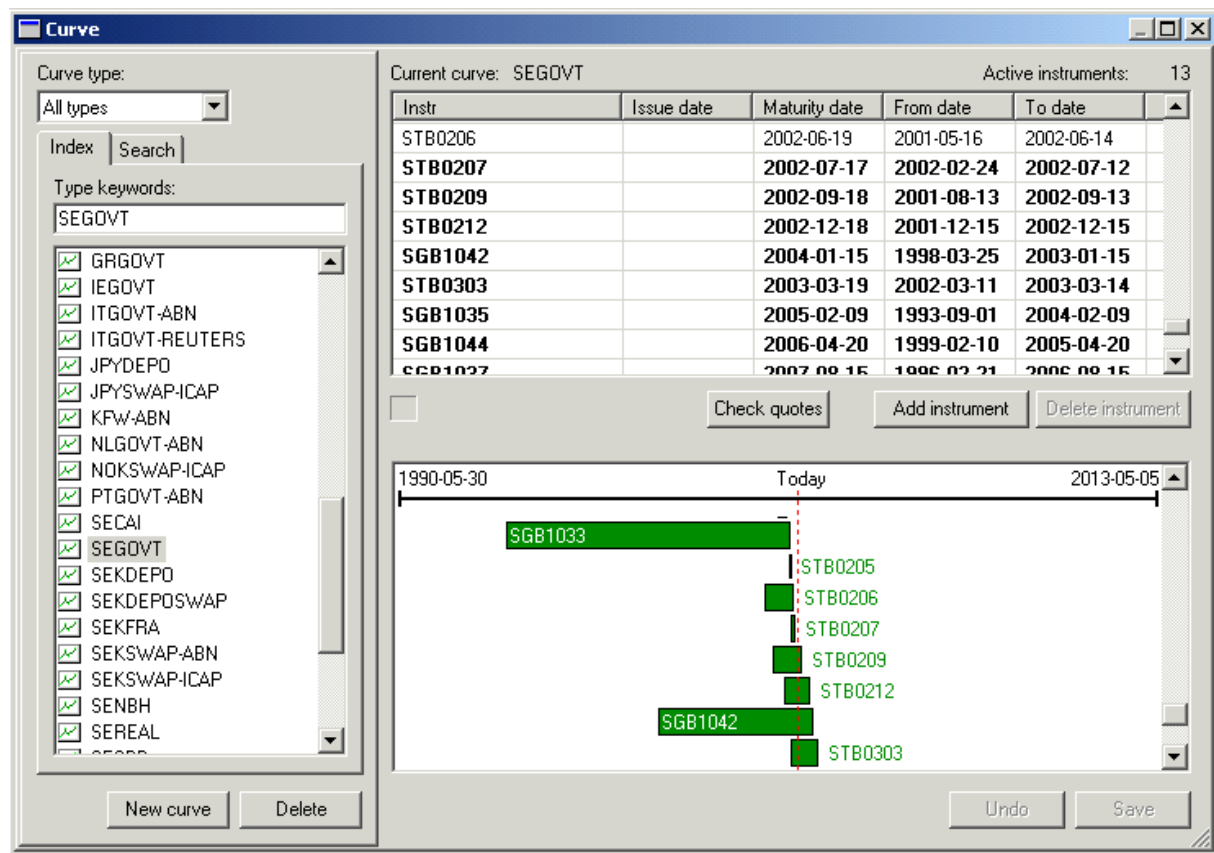
Some curves are made up of other curves blended together at certain user-defined maturities. Typical examples of this are blending a government bond curve together with deposit rates or blending swap rates and future rates together.

There are two ways of handling combining curves:

1. Defining the curve having all instruments in the same one. Can be useful for government + depo for instance.
2. Defining curves separately and blending them together using the built-in functions in Quantlab or MarketWatch. Useful for swap – FRA or swap-future curves.

## 4.2 Adding, editing and deleting curves

The curve dialog can be found under Tables | Curves or by pressing Alt+3.



Curve dialog with sample data

As in the instrument dialog, the curve names appear in the list box. A simple search function is available to find certain types of curves. The curve type is used only for searching purposes, it does not affect any calculations.

**Note!** By right clicking on the curve name in the list box you can change name, type and other parameter settings for each curve.

In the upper right window of the dialog, the curve's instrument list is shown. By double clicking on a cell in the instrument, the From date or To date data can be changed.

Sorting data by columns is achieved by clicking on the column header.

Quantlab and MarketWatch will regard any date left empty as infinite and correct. This implies that an instrument without From or To dates always will be on the curve. This is the typical standard for such instruments as swaps and money market deposits.

The lower right window shows in a graphical manner the same instruments as the upper window. A green horizontal bar indicates the period for which each instrument is on the curve. By dragging the vertical time line left and right, it is possible to see which instruments are on the curve for a given date. Instruments on the curve will be marked with bold in the upper window.



---

**Speed tip!** By double clicking on date above the vertical time line (or start and end dates), the dates can be written explicitly. This can be helpful in order to get accuracy for a specific date.

---

#### **4.2.1 Check for missing data**

By pressing the 'Check quotes' button, it is possible to check a complete curve for missing data in the database. The resulting table can be copied and saved to for example MS Excel for further examination. Correct holiday calendars will be applied by using the instrument's market definitions.

#### **4.2.2 Having multiple date ranges**

An instrument can go on and off the curve multiple times with different from- and to dates. This can be necessary for instruments having rolling ric:s or for blended curves where instruments are needed to be taken on and off the curve multiple times.

By adding a new row for the same instrument with a different from- and to date this is easily achieved.

### 4.2.3 Adding instruments to a curve

One or several instruments can be added to a curve at a time. Pressing 'Add' will open an instrument dialog with some special features.

A search function will filter out the instrument or instrument group that will be added to the curve.

In the list box one or many instruments can be marked for addition.

For instruments that require a from- and to date for when they should be on the curve, the lower controls will assist.

---

**Note!** For swaps and other instruments that should perpetually exist on the curve, the from-date and to-date should be left empty.

---

A from- or to date can take an instrument's

- issue date
- maturity date
- today's date
- first settlement date

as starting point and having +/- a numeric number of calendar days, months, or years.

---

**Tip!** As discussed in the short example about adding a German government bond, in the instrument chapter, it should now be apparent that having set the issue date of an instrument is quite convenient.

---

**Insert instrument**

Search

Instrument type: **FIXEDCPNBOND** Currency: **EUR**

Free text:  Search

☐ Match whole word

Instruments:

- EIB 4.875 15/04/06
- EIB 5.0 15/04/08
- EIB 5.25 15/04/2004
- EIB 5.625 15/10/10
- EIB 5.75 15/02/07
- FINLAND 1/00
- FINLAND 1/01
- FINLAND 1/93
- FINLAND 1/96
- FINLAND 1/97
- FINLAND 2/98
- FINLAND 3/98
- GermanGovtBond**
- GREECE 10Y FIXED 1/2008
- GREECE 10Y FIXED 1/2009
- GREECE 10Y FIXED 10/2007

Mark all

Unmark all

Set 'From date' as:

Issue date ☐ + ☐ -  **day(s)**

Set 'To date' as:

Maturity date ☐ + ☐ -  **year(s)**

Cancel Insert

After instruments are inserted into the upper window they are marked with a blue background. This means the instrument is not yet saved to the database. To alter any date information after insert has been made simply double click on the data and change.

Press 'Save' to commit changes or new data to the database.

## 5 Holiday calendars

Any calculation that depends on an exact calendar day to accurately price or adjust some other information about an instrument will need a holiday calendar. When working with time series it is also important to know which dates that should be excluded in graphs and statistical calculations. For these purposes it is important to have accurate holiday calendars for all markets for which there are instruments specified.

Quantlab needs only the market holiday dates. Weekends do not have to be specified and will be excluded automatically in the system.

### 5.1 Adding, editing and deleting holiday calendars

H.	Date	Weekday
<input checked="" type="checkbox"/>	2001-01-01	Monday
<input checked="" type="checkbox"/>	2001-04-13	Friday
<input checked="" type="checkbox"/>	2001-04-16	Monday
<input checked="" type="checkbox"/>	2001-05-07	Monday
<input checked="" type="checkbox"/>	2001-05-28	Monday
<input checked="" type="checkbox"/>	2001-08-27	Monday
<input checked="" type="checkbox"/>	2001-12-25	Tuesday
<input checked="" type="checkbox"/>	2001-12-26	Wednesday
<input checked="" type="checkbox"/>	2002-01-01	Tuesday
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		

*Holiday calendar dialog*

All defined markets are shown in the left window. In the right window all dates specified as market holidays will be ticked.

Top right is a search and select function where from and to date can be chosen. Ticking the "Show only holidays" box will display only holidays. Two tick boxes are available to hide Saturdays and Sundays.

To let any specific day of the week to be a constant holiday, select the day in the bottom list box and press the Set button.

### **5.1.1 Creating a new market and setting holidays**

Below is a “walk-through” example to create a new holiday calendar:

1. Click the Insert button and write a new market name. (This name will appear in the combo box for the market name used in the Instrument class or Instrument form.)
2. Activate the new market name in the left list window.
3. Un-tick the "Show only holidays" tick box. Keep the Saturdays and Sundays hidden.
4. Scroll to all dates and tick where appropriate for all market holidays.
5. When finished press "Commit" to save the holiday calendar to the database.

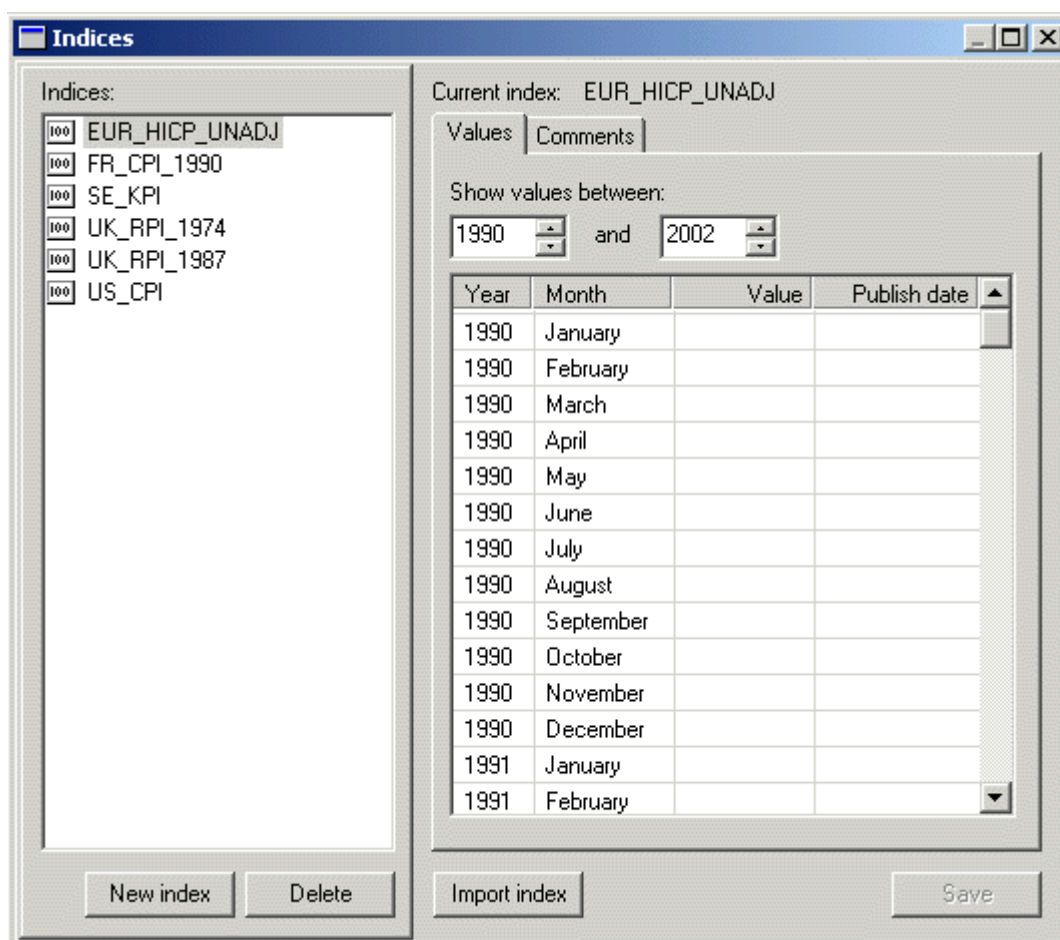
## 6 Indices

It is possible to add different kinds of indices to the Quantlab database. A typical required index is the Consumer Price Index, CPI for different markets. These monthly numbers are then used for certain instruments like the index-linked bonds.

### 6.1 Adding, editing and deleting indices

The menu Tables | Indices or Alt+6 will open the indices dialog. As with all the Database Tool's dialogs the list of current indices are presented on the left hand side. Index values will display on the right. Comments can be freely input under the comments tab.

To enter values in the "Value" and "Publish date" columns, simply double-click on the cell and enter the data. The same method will apply when editing old data. The Publish date is required only for UK Index-linked bonds.



The dialog box titled "Indices" has a left pane labeled "Indices:" containing a list of indices, each with a small icon and a name: EUR\_HICP\_UNADJ, FR\_CPI\_1990, SE\_KPI, UK\_RPI\_1974, UK\_RPI\_1987, and US\_CPI. Below this list are buttons for "New index", "Delete", "Import index", and "Save". The right pane shows the "Current index: EUR\_HICP\_UNADJ" and has two tabs: "Values" (selected) and "Comments". Under the "Values" tab, there is a section "Show values between:" with two date pickers set to "1990" and "2002". Below this is a table with four columns: "Year", "Month", "Value", and "Publish date". The table contains data for the years 1990 and 1991, with months listed from January to December for 1990 and January to February for 1991. The "Value" and "Publish date" columns are empty.

Year	Month	Value	Publish date
1990	January		
1990	February		
1990	March		
1990	April		
1990	May		
1990	June		
1990	July		
1990	August		
1990	September		
1990	October		
1990	November		
1990	December		
1991	January		
1991	February		

*Indices dialog*

---

**Tip!** To import a whole range of values for an index, use the "Import index" button.

---

By formatting a comma separated text file in the style YEAR, MONTH, VALUE, [PUBLISH DATE] the data can be directly imported. Some different formats for MONTH are allowed ("January", "1", "Jan" are all valid MONTH formats).

An example file might look like this:

1990, January, 100

1990, February, 101

1990, March, 102

## 7 Appendix 1. Database tables

### 7.1.1 Tag tables

The original idea with these tables is to allow for extension of static data for instruments and curves, but it can be used in many other cases as well. The tables work as dynamic four-column tables with a key identifier, a value, a from-date and a to-date. Instead of having several such tables with different names you define a tag name that corresponds to a hypothetical table.

For example, the table TagValueStr could look like this:

TAG_NAME	TAG_KEY	FROM_DATE	TO_DATE	TAG_VALUE
instr_rating	Corp A 453	2002-01-01	2003-06-05	AA
instr_rating	Corp A 453	2003-06-06		A
instr_rating	Corp B 765	2002-01-01		BB
curve_rating	Corp C	2002-01-01		AA

given that the TagDef table has the following entries:

TAG_NAME	TAG_TYPE	KEY_TYPE
instr_rating	string	instrument
curve_rating	string	curve

In this example we have used the tag table TagValuesStr to save rating data, both for instruments and for curves, and the tags needed for this are defined in the TagDef table. For example, the bond called Corp A 453 has had a AA rating from the first of January 2002 to the fifth of June 2003 when it changed to single A. In order to actually be able to interpret these values in Quantlab, you must define functions that retrieves this data. For an instrument it is most convenient to define a member function in a library file:

```
string instrument.rating(instrument i)
{
    string rating = get_tag_str('instr_rating', i.name, i.trade_date);
    if (null(rating)) return '';
    else return rating;
}
```

This function will be added to the instrument object when compiled in a library file. See the Quantlab User Manual for further information about library files and member functions.

Similar functions can be added for variables of date and number type after having inputted the necessary rows in the tag tables. Each new tag has to be defined in the TagDef table where you set the name and the return type: Date, number or string.

Tags can be edited from DatabaseTool either by using the tag tabs in the instrument, curve and issuer dialogs, or by using the general Tag dialog. In both cases you have to define each tag using the Tag dialog, however.

### 7.1.2 The QuoteSource table: Extending the Quote table and adding real-time quote fields

It is possible to add columns to the Quote table and also to add real-time quote fields to instruments. In this context, "quote" has a broad meaning: Any type of data, associated to an instrument, that could be saved on a daily basis and/or retrieved in real time.

In the table QuoteSource all quote fields are defined. In RealtimeLink you define the specifications for the real time sources and in QuoteSourceComp you are able to make comparisons between two quotes. This means that you might have to set up new columns in RealtimeLink, and also a new Quote table, depending of what you want to do. In this paragraph we give a description of the tables QuoteSource and QuoteSourceComp and in 7.1.2 there are examples that include the RealtimeLink table.

Below are the columns in the QuoteSource table.

Column name	Description
QUOTE_SOURCE_NAME	The name of the quote_side, e.g. used in instrument creation. E.g. "bid" or "ask".
QUOTE_TYPE	The type of the quote: "number", "date" or "string".
REAL_QUOTE	A boolean, true (nonzero) if this quote_side may be used in instrument creation, false otherwise. If true, QUOTE_TYPE must be "number".
LINK_TABLE_HISTORIC_TABLE_COLUMN	The name of the column in RealtimeLink containing the name of the table from which historic quotes are fetched for the specific instrument. E.g. "QUOTE_TABLE".
LINK_TABLE_HISTORIC_KEY_COLUMN	The name of the column in RealtimeLink containing the key used for looking up historic quotes for the specific instrument. E.g. "RIC".
LINK_TABLE_RIC_COLUMN	The name of the column in RealtimeLink containing the ric used for fetching realtime quotes for the specific instrument. E.g. "RIC" or "REALTIME_RIC".
LINK_TABLE_FID_COLUMN	The name of the column in RealtimeLink containing the fid used for fetching realtime quotes for the specific instrument. E.g. "BID_CODE" or "ASK_CODE".
REALTIME_FEED	The feed used for fetching realtime quotes (if any). E.g. "IDN_SELECTFEED".
HISTORIC_TABLE_KEY_COLUMN	The name of the column in the historic table for this instrument containing the key to be matched against the key from RealtimeLink in the column specified by LINK_TABLE_HISTORIC_KEY_COLUMN. E.g. "RIC".
HISTORIC_TABLE_QUOTE_COLUMN	The name of the column in the historic table for this instrument containing the quote. E.g. "BID" or "ASK".



If `REALTIME_FEED` is set, quotes for "today" will be fetched from the corresponding realtime source, otherwise "today" will have no special meaning and the quotes will be fetched from history (if available).

If any of

`LINK_TABLE_HISTORIC_TABLE_COLUMN`,  
`LINK_TABLE_HISTORIC_KEY_COLUMN`,  
`HISTORIC_TABLE_KEY_COLUMN`,  
or `HISTORIC_TABLE_QUOTE_COLUMN`,

is not set, the `quote_side` cannot be used for fetching historic quotes.

If `LINK_TABLE_RIC_COLUMN` or `LINK_TABLE_FID_COLUMN` is not set, the `quote_side` cannot be used for fetching realtime quotes.

The former contributor fields in Realtime quotes are deprecated. Instead you can define several parallel quote sources for the different contributors. One advantage is that in Quanlab workspaces, they will automatically turn up in list boxes with quote sides.

As a first example, one row in the table `QuoteSource` could look like this (transposed):

Column name	Value
<code>QUOTE_SOURCE_NAME</code>	Bid
<code>QUOTE_TYPE</code>	number
<code>REAL_QUOTE</code>	1
<code>LINK_TABLE_HISTORIC_TABLE_COLUMN</code>	<code>QUOTE_TABLE</code>
<code>LINK_TABLE_HISTORIC_KEY_COLUMN</code>	<code>RIC</code>
<code>LINK_TABLE_RIC_COLUMN</code>	<code>REALTIME_RIC</code>
<code>LINK_TABLE_FID_COLUMN</code>	<code>BID_CODE</code>
<code>REALTIME_FEED</code>	<code>IDN_SELECTFEED</code>
<code>HISTORIC_TABLE_KEY_COLUMN</code>	<code>RIC</code>
<code>HISTORIC_TABLE_QUOTE_COLUMN</code>	<code>BID</code>

The first column corresponds to the `quote_field` parameter in the `get_quote`-functions mentioned above, the second column refers to the type (number, date or string) which determines which function to use, the third column is 1 (true) if the quote can be used to price instruments (a "real" quote) as opposed to for example a volume or a spread, the following four columns correspond to the column names in the `RealtimeLink` table, the `Realtime_feed` column determines what realtime feed to use and the last two columns corresponds to column names in the historic quote table. In this way you may have several quote tables at the same time and the realtime link table can be extended with additional quote sides.

If there is no entry in the `LINK_TABLE_FID_COLUMN` column the database will always be used when retrieving data. This has been exploited in the example below as there are two quote sides, `bid_db` and `ask_db`, that are always the same as the usual bid and ask except that they don't use real time data when today's date is chosen. So if there are quotes written to the database during the day, these can be used alternatively with real time quotes.

Column name	Value	Value
-------------	-------	-------

QUOTE_SOURCE_NAME	Bid_db	Ask_db
QUOTE_TYPE	number	number
REAL_QUOTE	1	1
LINK_TABLE_HISTORIC_TABLE_COLUMN	QUOTE_TABLE	QUOTE_TABLE
LINK_TABLE_HISTORIC_KEY_COLUMN	RIC	RIC
LINK_TABLE_RIC_COLUMN		
LINK_TABLE_FID_COLUMN		
REALTIME_FEED		
HISTORIC_TABLE_KEY_COLUMN	RIC	RIC
HISTORIC_TABLE_QUOTE_COLUMN	BID	ASK

A last example shows a quote field that is not a real quote, i.e., not a quote\_side in Quantlab. This field corresponds to an extra column in the Quote table, called "acvol".

Column name	Value
QUOTE_SOURCE_NAME	AcVol
QUOTE_TYPE	number
REAL_QUOTE	0
LINK_TABLE_HISTORIC_TABLE_COLUMN	QUOTE_TABLE
LINK_TABLE_HISTORIC_KEY_COLUMN	RIC
LINK_TABLE_RIC_COLUMN	REALTIME_RIC
LINK_TABLE_FID_COLUMN	ACVOL_CODE
REALTIME_FEED	IDN_SELECTFEED
HISTORIC_TABLE_KEY_COLUMN	RIC
HISTORIC_TABLE_QUOTE_COLUMN	acvol

Similarly to the tag fields described in 7.1.1, it is convenient to define instrument member functions for the quote fields you have defined in the database. For instance, you could in a library file, write

```
number instrument.AcVol(instrument i)
{
    number spread = i.get_quote_num('AcVol');
    if (null(spread)) return 0;
    else return spread/100;
}
```

to create a function that gives the accumulated volume for an instrument, given it is written to the database in the Quote table column "acvol" and defined in the QuoteSource table as AcVol.

In addition to the QuoteSource table there is a table QuoteSourceComp that combines two quote sources. It is used for taking averages and for making prioritizations between quote sources.

Column name	Description
QUOTE_SOURCE_NAME	The name of the combined quote_side, e.g. used in instrument creation. E.g. "mid".
REAL_QUOTE	A boolean, true (nonzero) if this quote_side may be used in instrument creation, false otherwise. If true, QUOTE_TYPE must be "number".
OPERATION	A string enumerating the operation that is used to combine two quote-

	sides to form a new one. Initially supports the following operations: 'avg' - the average between the two quote-sides, 'prio' - the first available of two quote-sides (used for realtime-feeds).
QUOTE_SOURCE_1	The name of the first quote-side. E.g. "BID".
QUOTE_SOURCE_2	The name of the second quote-side. E.g. "ASK".

For example, you can construct a mid quote of a bid and an ask quote, or letting the bid-sw quote prioritize the bid-bb quote and, if that is null, take the bid quote instead:

Column name	Value	Value	Value
QUOTE_SOURCE_NAME	mid	mid-bb	bid-sw
REAL_QUOTE	1	1	1
OPERATION	avg	avg	prio
QUOTE_SOURCE_1	bid	bid-bb	bid-bb
QUOTE_SOURCE_2	ask	ask-bb	bid

For further examples including the RealtimeLink table see 7.2.

## 7.2 Example of multiple real time sources

By using the QuoteSource table and RealtimeLink table as described above it is possible to set up multiple real time sources for instruments. For example you may have Reuters and Bloomberg connections at the same time and use the following setup in the QuoteSource table:

QUOTE_SOURCE_NAME	QUOTE_TYPE	REAL_QUOTE	LINK_TABLE_HISTORIC_TABLE_COLUMN	LINK_TABLE_HISTORIC_KEY_COLUMN	LINK_TABLE_REALTIME_COLUMN	LINK_TABLE_FID_COLUMN	REALTIME_FEED	HISTORIC_TABLE_KEY_COLUMN	HISTORIC_TABLE_QUOTE_COLUMN
bid	number	1	QUOTE_TABLE	RIC	REALTIME_RIC	BID_CODE	IDN_SELECTFEED	RIC	BID
ask	number	1	QUOTE_TABLE	RIC	REALTIME_RIC	ASK_CODE	IDN_SELECTFEED	RIC	ASK
bid-bb	number	1	QUOTE_TABLE	RIC	BB_ID	BB_BID_CODE	BLOOMBERG	RIC	BID
ask-bb	number	1	QUOTE_TABLE	RIC	BB_ID	BB_ASK_CODE	BLOOMBERG	RIC	ASK
last-bb	number	1	QUOTE_TABLE	RIC	BB_ID	BB_LAST_CODE	BLOOMBERG	RIC	LAST

And the following setup in the RealtimeLink table:

INSTRUMENT_ID	PRIORITY	FEE	RIC	BID_CODE	ASK_CODE	FROM_DATE	TO_DATE	QUOTE_TABLE	ALLOW_CONTRIBUTOR	DEF_CONTRIBUTOR	REALTIME_RIC	ACVOL_CODE	BIDSIZE_CODE	ASKSIZE_CODE	OPENINTR_CODE	NUMMOVES_CODE	BB_ID	BB_BID_CODE	BB_ASK_CODE	BB_LAST_CODE
7653		RSF	FR081205B=	393	275	1901-01-01	2027-01-01	Quote	1		FR081205B=	32	30	31	64	77	ISIN=FR0107673925	2	3	1
7654		RSF	FR220905B=	393	275	1901-01-01	2027-01-01	Quote	1		FR220905B=	32	30	31	64	77	ISIN=FR0107673891	2	3	1
7655		RSF	IT387292=	393	275	1901-01-01	2027-01-01	Quote	1		IT387292=	32	30	31	64	77	ISIN=IT0003872923	2	3	1
7656		RSF	JP00780069=	393	275	1901-01-01	2027-01-01	Quote	1		JP00780069=	32	30	31	64	77	ISIN=JP1200781563	2	3	1
7657		RSF	NL10224=	393	275	1901-01-01	2027-01-01	Quote	1		NL10224=	32	30	31	64	77	ISIN=NL0000102242	2	3	1
7658		RSF	NO1027289=OL	393	275	1901-01-01	2027-01-01	Quote	1		NO1027289=OL	32	30	31	64	77	ISIN=NO0010272891	2	3	1
7659		RSF	US912828DX5=	393	275	1901-01-01	2027-01-01	Quote	1		US912828DX5=	32	30	31	64	77	ISIN=US912828DX53	2	3	1

## 8 Appendix 2: Instrument definition

The table below shows the valid choices and explanations for all instrument fields.

### General instrument information

Dialog caption	Valid choices	Re-quired	Explanation	Database column
Current instrument	String	Y	The (short) name of the instrument. This is the name that is normally used to identify the instrument and, hence, must be unique.	Name
Long name	String		A long name of the instrument.	Name_long
Class	Instrument class from list		An instrument class name. The instrument class that an instrument points to is used for supplying default values should any data not be specified.	Class_id
Type	String	Y	Type of instrument.	Instr_type
Issue date	Date		The issue date for the instrument i.e. the date on which the instrument begins trading in the secondary market. Normally this is also the date on which the instrument begins accruing interest (if not see [Coupon start date]). The issue date can also be useful when defining start date for curves.	Issue_date
First coupon date	Date		The first coupon payment date for the instrument. Together with [Issue date] this date should be specified in the case the instrument has an odd first coupon period. If this date is not specified it will be calculated according to the regular coupon payment cycle for the instrument.	First_cpn_date
Maturity date	Date	Y	Maturity date for the instrument.  Either [Maturity date] or [Maturity code] must be specified.	Maturity

Maturity code	Integer >0 and Y, M, W or D and optionally _F, _MF, _P, _MP or NONE	Y	Maturity code is divided into two or three parts, an integer number greater than zero and 'Y' for year(s), 'M' for month(s), 'W' for week(s), 'D' for day(s). The optional third part decides the business day convention for holidays. It can be one of '_F' for Following, '_MF' for Modified following, '_P' for Previous, '_MP' for Modified previous and NONE for no adjustment.  Either [Maturity date] or [Maturity code] must be specified.	Maturity_code
Settle date	Date	Y	Settlement date for the instrument. If [Settle code] is <u>not</u> specified this will be the settlement date for the instrument. If <u>both</u> [Settle code] <u>and</u> [Settle date] are specified then [Settle date] will define the first settlement date for the instrument. This date should be specified, for example, for forward starting instruments. Also, to define bonds that trade "when issued" this date should normally be set equal to [Issue date]. The default value of [Settle date] is [Issue date]. Either [Settle date] or [Settle code] must be specified.	First_settle_date
Settle code	( BD, CD and Integer >=0 or Integer >0 and Y, M, W or D or WD+weekday and integer > 0 ) and optionally _F, _MF, _P or _MP	Y	A generic settlement code for the instrument defining the number of days from the trade date to the settlement date.  It can be assembled in one of three ways: 1) First part; - BD for business days, - CD for calendar days.  Second part; any integer larger than or equal to zero. 2) Code of the same type as [Maturity code]. 3) WD for Weekday + code for the weekday (WDmon, WDtue, WDwed etc.) and an integer for the number of weeks (E.g. WDFri3 means Friday three weeks from now)  All these three codes (except for BD which needs no business day convention) can optionally be followed by a business day convention in the same way as for [Maturity code].	Settle_code

Coupon	Numeric >=0		Coupon size expressed as a percentage. Example 5.5 %	Coupon
Coupon freq	1 2 4	Y	Number of coupons per year. Valid choices are: 1 for annual, 2 for semi-annual and 4 for quarterly coupon payments.	Cpn_freq
Accrued day count method	ACT360 ACT365 ACTACT (*) ACT365NL (*) EU30360 EU30360P1 (*) EU30D360 (*) EU30P360 US30360 SIA30360 (*) PSA30360 (*)  * denotes day count conventions not explicitly available in QLang functions.	Y	Day count conventions available.  - ACT360 for actual number of days divided by 360 days per year,  - ACT365 actual days divided by 365 days per year,  - ACTACT actual days divided by actual days in coupon period,  - ACT365NL for ignoring leap days,  - EU30360 for ISMA version of 30 days a month divided by 360 days per year,  - EU30D360 for German version of 30 days a month divided by 360 days per year,  - EU30360P1 for Italian bonds with an extra day of accrued interest,  - US30360 for US (ISDA) version of 30 days a month divided by 360 days per year,  - SIA30360 for SIA (Securities Industry Association) version of 30 days a month divided by 360 days per year,  - PSA30360 for PSA (Public Securities Association) version of 30 days a month divided by 360 days per year.	Accr_dc_method
Face amount	Numeric		Face amount of the instrument. Will default to 100 if left empty.	Face_amount
Issued amount	Numeric		The currently issued amount of the instrument.	Issued_amnt

Ex-coupon code	BD, CD and ( integer >=0 or MB + FI, LA, SA ) and optionally _F, _MF, _P or _MP	Y	<p>If the ex-coupon code is:</p> <ul style="list-style-type: none"> <li>- "BD" for business days or</li> <li>- "CD" for calendar days,</li> </ul> <p>an integer larger than or equal to zero is used in the second field. This field should be interpreted as the number of days between the ex-coupon date and the coupon date.</p> <p>If the ex-coupon code is:</p> <ul style="list-style-type: none"> <li>- "MB"</li> </ul> <p>possible values for the second field are:</p> <ul style="list-style-type: none"> <li>- "FI" for first day in the month before,</li> <li>- "LA" for last day in the month before,</li> <li>- "SA" for same as coupon day, or</li> <li>- a numeric value interpreted as the day in the month before.</li> </ul> <p>The optional third part decides the business day convention for holidays (not available for BD). It can be one of '_F' for Following, '_MF' for Modified following, '_P' for Previous and '_MP' for Modified previous.</p> <p>Note that the ex-coupon day is the day when the instrument trades exclusive of the coupon.</p>	Excpn_code
Currency	CCY ISO-codes	Y	Main currency of the instrument	Principal_currency
Issuer	Issuers from list	Y	Having first entered issuers in the Issuer table, choices become available in the combo box.	Issuer_id
Market	String from list		A user defined name of a market. Used to connect with a holiday calendar. All market names must match those chosen in the Holiday calendar dialog.	Market



Calc type	CCY + BILL CCY + GOVT CCY + CORP CCY + STRIP IL + CCY + GOVT <u>Special types:</u> DEPO SWAP SWAPEFF ESPGOVTODD SEKGOVTODD SEKOM BOND BONDEFF BILL (equivalent to SIMP) FRFOAT, FRFBTAN, USDGOVTTR, USDCORPSTRIP USDAGENCY USDAGENCYSTRIP USDMUNI USDMUNISTRIP ZERO (equivalent to EFF) AUSGOVTOLD BEFGOVTOLD DEMGGOVTOLD ESPGOVTOLD EURGOVTOLD FIMGGOVTOLD GBPGGOVTOLD IEPGGOVTOLD ITLGOVTOLD JPYGOVTOLD NLGGGOVTOLD PTEGOVTOLD se Appendix 2	Y	<p>Calculation type that determines all calculation (secondary market) conventions specific for the market and type of instrument.</p> <p>Note that only the available (in the drop down list) combinations of currency codes and BILL, GOVT, CORP, and STRIP list can be used. Other combinations may exist but are untested in the current version of MarketWatch and Quantlab.</p> <p>Some special types are available:</p> <ul style="list-style-type: none"> <li>- DEPO for deposits,</li> <li>- SWAP for regular swaps,</li> <li>- SWAPEFF for regular swaps with annual effective compounding,</li> <li>- ESGOVTODD for Spanish bonds with an odd first coupon period with a regular coupon payment,</li> <li>- SEKGOVTODD for Swedish bonds with an odd first (proportional) coupon,</li> <li>- SEKOM for futures traded at OM,</li> <li>- SEKSTRIP for strip issues,</li> <li>- BOND for generic bonds with compounding frequency equal to coupon frequency,</li> <li>- BONDEFF for generic bonds with annually compounded yields,</li> <li>- BILL (SIMP) for single cash flow instruments with simple rates,</li> <li>- ZERO (EFF) for single cash flow instruments with annual effective rates,</li> <li>- USDGOVTTR for USD government bonds with treasury method,</li> <li>- USDAGENCY for US agency bonds,</li> <li>- USDMUNI for US municipal bonds,</li> <li>- *OLD for old style definitions</li> </ul>	Calc_type
-----------	--	---	--	-----------

Quote style	CLEAN CLEAN_PCT DIRTY DIRTY_PCT YLD EDFUT CLEAN_PCT_EXINFL	Y	Field defining by which convention an instrument's price is quoted in the market. Used for the interpretation of all historical and real-time data.  Valid choices are:  - CLEAN for clean price, - CLEAN_PCT for clean price percentage (most common case), - DIRTY for dirty price, - DIRTY_PCT for dirty price percentage and, - YLD quotations in yield, - EDFUT for quotations with (100 – yield) - CLEAN_PCT_EXINFL for clean price excluding inflation accrual (applicable only for inflation linked bonds)	Quote_style
-------------	--	---	--	-------------

## Extended instrument information

Dialog caption	Valid choices	Req	Explanation	DBase column
Announce date	Date		The date when the instrument issue was announced.	Announce_date
Auction date	Date		The date when the auction is held. An instrument can trade “when-issued” between the announce date and the auction date.	Auction_date
Coupon start date	Date		The date on which a new instrument begins accruing interest. This date is sometimes referred to as the dated date.  For instruments which are issued including accrued interest, i.e. the accrual start period is before the issue date, the Calc type setting determines the coupon. See the example for a Swedish Government Bond below.	Cpn_start_date
Last regular coupon date	Date		Last regular coupon date i.e. the last date that falls in the regular coupon cycle. This date should be specified if the instrument has an odd last coupon period.	Last_reg_cpn_date
Value code	See [Settle code]		A generic value date code for the instrument defining the number of days from the trade date to the value date. The value date is the “settlement date” used for calculating accrued interest, which for some instruments is different from the settlement date used for calculating price or yield. Will default to settlement code if left empty.	Value_code
Coupon currency	CCY ISO-codes		Currency of the coupon payments if different from the instruments principal currency.	Cpn_currency
Capital gains tax rate	Numeric		Tax rate for capital gains on the instrument.	Capgain_tax
Coupon tax rate	Numeric		Tax rate for coupon payments.	Cpn_tax
Issue price	Numeric		Price at which the instrument was issued. If left empty, will default to 100.	Issue_price

Issue price for tax	Numeric		Price at which the instrument was issued for the purpose of calculating taxes on the capital gain. If left empty, will default to 100.	Issue_price_tax
Redemption price	Numeric		The redemption price of the instrument in percentage terms. Normally this is equal to 100.	redemp_price
First coupon payment	Numeric		The size of the first coupon payment. This value overrides any calculated first coupon payment and, hence, needs only be specified in case the type of first coupon payment is not handled by the financial library.	First_coupon
Rounding convention   Clean price	R+number, T+number or RT+number ,number		<p>The market rounding conventions on clean price. 3 cases</p> <p>1) R + number: rounding to specified number of decimals (ex R3 for rounding to 3 decimals),</p> <p>2) T + number: truncating to specified number of decimals (ex T3 for truncating to 3 decimals), and</p> <p>3) RT + number,number: first apply rounding to specified number of decimals then truncating to specified number of decimals (ex RT5,4 for rounding to 5 decimals and thereafter truncating to 4 decimals).</p> <p>Note that first truncating and then rounding is equivalent to simple rounding hence a "TR" code is not defined.</p>	Price_round_dec
Rounding convention   Yield	see Rounding convention   Clean price		<p>The market rounding conventions on yield.</p> <p>see Rounding convention   Clean price for explanation.</p>	Yield_round_dec
Rounding convention   Accrued interest	see Rounding convention   Clean price		<p>The market rounding conventions on accrued interest.</p> <p>see Rounding convention   Clean price for explanation.</p>	Accr_round_dec

Daycount method   To next coupon	See [Accrued day count method]		Used if the day count method for the period to the next coupon payment, i.e. for yield/price calculations, is different from the [Accrued day count method]. Will default to [Accrued day count method] if left empty.	Next_dc_method
Daycount method   Accrued (in last period)	See [Accrued day count method]		Used if the accrued day count method in the last coupon period is different from the regular periods. Will default to [Accrued day count method] if left empty.	Accr_dc_last_method
Daycount method   To next coupon (in last period)	See [Accrued day count method]		Used if the day count method in the last coupon period (for the period to the next coupon payment) is different from the [Next dc method]. Will default to [Next dc method] if left empty.	Next_dc_last_method
End of month	Yes / No (ticked for yes)		If adjustment for end of month should be made for the instrument. Yes means that the last day in month principle will be used and no means that the same day principle will be used.	Eom
Allow zero quote	Yes / No (ticked for yes)		If the automatic warning checks should allow prices/yields being zero for an instrument. Example: Japanese government bond yields should be allowed to take zero as valid quote values.	Allow_zero_quote
Business day convention for payments	F, MF, P, MP, NONE		<p>Defines the date adjustment should a coupon day occur on a non-business day. Relevant for swaps and true-yield calculations.</p> <p>F=following,  MF=modified following,  P=previous,  MP=modified previous  NONE=no adjustment</p> <p>Default value for bonds: F  Default value for swaps: MF</p>	Pmt_bus_day

## Special instrument information

Dialog caption	Valid choices	Req	Explanation	DBase column
CPI Name	String from list	Y	Name of the inflation index (the Consumer Price Index or similar index measuring inflation).	Cpi_name
CPI Lag 1	Integer	Y	First monthly lag of CPI (for example for ILSEKGOVT and ILFRFGOVT this is equal to 2 and for ILGBPGOVT it's equal to 8)	Cpi_lag1
CPI Lag 2	Integer		Second monthly lag of CPI (for example for ILSEKGOVT and ILFRFGOVT this is equal to 3 but for ILGBPGOVT it's not applicable).	Cpi_lag2
CPI Base Date	Date		CPI base date for the instrument. Required only for UK index linked bonds.	Cpi_date
CPI Base Index	Numeric	Y	CPI base index for the instrument. The base index is taken from this field, it is not calculated from the index series and the CPI base Date.	Cpi_Base
Rounding Code	see Rounding convention   Clean price		Rounding code for calculated cash flows. see Rounding convention   Clean price for explanation.	Cash_round_dec
Minimum redemption at par	Yes / No (ticked for yes)		Defines if the instrument has a minimum redemption at par.	Min_redemp_at_par

## FRN information

Dialog caption	Valid choices	Req	Explanation	DBase column
Fixed margin	Numeric	Y	Fixed margin for the FRN.	Fix_margin
Roll Style	Straight IMM 3rdWed	Y	Straight = roll on straight periods. IMM = roll on IMM-dates. 3rdWed = roll on 3 <sup>rd</sup> Wednesday of the month.	Roll_style
Index	String from list	Y	Name of frn index.	Frn_index

Fixing code	( BD, CD and Integer ≥0 or Integer >0 and Y, M, W or D ) and optionally _F, _MF, _P or _MP	Y	<p>A generic fixing code for the instrument defining the number of days (going back) from the coupon payment date to the fixing date.</p> <p>It can be assembled in one of three ways:</p> <p>1) First part;</p> <ul style="list-style-type: none"> <li>- BD for business days,</li> <li>- CD for calendar days.</li> </ul> <p>Second part;</p> <p>any integer larger than or equal to zero.</p> <p>2) Code of the same type as [Maturity code].</p> <p>All these three codes (except for BD which needs no business day convention) can optionally be followed by a business day convention in the same way as for [Maturity code].</p>	Fixing_code
-------------	---	---	--	-------------

*Note: FRN coupon fixings are set using the menu Table/FRN coupon*

## Real time link information

Dialog caption	Valid choices	Req	Explanation	DBase column
Ric	String	Y	The identification code used in the historical time series databases. Any string value is allowed as long as it matches the name/id used in the table where the data is stored. Commonly the historical data is saved using the ric as the unique identifier; in this case the real time feed can also use the name in this column.	Ric
Realtime ric	String	Y	The identification code used by the Reuters real time feed to identify the instrument. This can be different from the name under which the instrument is stored in the historical database. For Reuters feed a ric should be entered without any contributor in this field. Example: SE1044=	Realtime_ric
From date	Date	Y	Date from which the realtime_ric is valid for the instrument. If the realtime ric always will be valid, an early dummy date should be used. Example: 1901-01-01	From_date
To date	Date	Y	Date to which the realtime ric is valid for the instrument. If the realtime ric will always be valid, a late dummy date should be used. Example: 2027-01-01	To_date
Bid code	String/Integer	Y	Code identifying the field used in the real time for the bid quote. Using Reuters this can for example be: 22 for Swedish government bonds and 393 for European government bonds.	Bid_code
Ask code	String/Integer	Y	Code identifying the field used in the real time for the ask quote. Using Reuters this can for example be; 25 for Swedish government bonds and 275 for European government bonds.	Ask_code
Feed	String [Valid feed names]		RealtimeFeed name, see 7.1.2.	QuoteSource table: REALTIME_FEED
Quote table	String [Valid database table name]	Y	Name of the table where the instrument's historical time series is stored.	Quote_table
Allow contributor	Yes/No (ticked for yes)		Not used - Deprecated. Use alternative quote sides instead – see 7.1.2.	
Default contributor	String [Valid contributor name]		Not used - Deprecated. Use alternative quote sides instead – see 7.1.2.	
Bloomberg ID	String		Identification code in Bloomberg.	BB_ID
Bloomberg Ask Code	String		Ask code field identifier.	BB_ASK_CODE
Bloomberg Bid Code	String		Bid code field identifier.	BB_BID_CODE
Bloomberg Last Code	String		Last code field identifier.	BB_LAST_CODE

For Bloomberg real time source the following field identifiers are available:



<b>Field name</b>	<b>Description</b>
price_open	open price
price_high	high price
price_low	low price
price_last	last price
price_settle	settlement price
price_bid	bid price
price_ask	ask price
yield_bid	bid yield
yield_ask	ask yield
limit_up	up limit (futures)
limit_down	down limit (futures)
open_interest	open interest (futures)
yest_price_last	last price of previous session
scale	scale factor
time_last	time of last price
exchange_last	exchange of last trade
tick_direction	up/down/zero tick
size_bid	bid size
size_ask	ask size
condition_bid	bid condition
condition_ask	ask condition
condition_last	last trade condition
condition_market	market condition
monitable	monitable
volume_total	total trade volume
ticks_total	total number of ticks
time_start	session start time
time_end	session end time

## 9 Appendix 3: To define instruments using instrument classes

When defining new instruments, whether it is done by manual input or by automatic data transfer from another system, the use of instrument classes is recommended, see 3.1. The general advice is to use all information from the corresponding instrument class *except the information in the following table*:

<i>Caption in DatabaseTool</i>	<i>Field in table INSTRUMENT</i>
<b>General</b>	
Class	CLASS_ID
Maturity Date* or Code*	MATURITY or MATURITY_CODE
Coupon	COUPON
(Coupon freq)	CPN_FREQ
(Issuer*)	ISSUER_ID
Issue date	ISSUE_DATE
First coupon date	FIRST_CPN_DATE
Issued Amount	ISSUED_AMNT
<b>Extended</b>	
Announce date	ANNOUNCE_DATE
Auction date	AUCTION_DATE
Coupon start date	CPN_START_DATE
Last regular coupon date	LAST_REG_CPN_DATE
First coupon payment	FIRST_COUPON
Issue price	ISSUE_PRICE
Issue price for tax	ISSUE_PRICE_TAX
<b>Special</b>	
CPI base date	CPI_DATE
CPI base index	CPI_BASE
<b>Frn</b>	
Fixed margin	FIXED_MARGIN

Fields marked with \* are required for all instruments.

Fields in () could equally well be taken from instrument classes, if those are defined accordingly.

## 10 Appendix 4. CalcMethod attributes

The calculation methods are constructed using a number of attributes. These are combined to form a number of regional calculation types.

### Bonds

For bonds the following attributes are relevant.

Attribute	Description	Values
OddFirstCpnStyle	Determines the size of the first coupon payment when there is an odd first period.	Proportional: the coupon is proportional to the length of the period Regular: the coupon is set to the regular coupon payment
Compound. Freq	Compounding frequency	Number of periods per year (1,2 or 4)
NextPeriodStyle	Rate type used when discounting from next coupon payment to the settle date.	Effective, Simple, BankDisc
LastPeriodStyle	Rate type used when the bond is in the final coupon period.	Effective, Simple, BankDisc

The following calculation methods are currently available. For some calculation methods there is an old version used before a convention shift (for example the introduction of the euro).

CalcType	OddFirstCpn Style	Compound. Freq	NextPeriod Style	LastPeriod Style	Comment
ARSGOVT	regular	2	Effective	Simple	
AUDGOVT	proportional	2	Effective	Simple	
AUSGOVT	proportional	1	Effective	Simple	
AUSGOVTOLD	proportional	1	Effective	Simple	
BEFGOVT	proportional	1	Effective	Simple	
BEFGOVTOLD	proportional	1	Effective	Simple	
BGLGOVT	proportional	2	Effective	None	
CADGOVT	proportional	2	Effective	None	>=3years
CADGOVT2	proportional	2	Effective	Simple	<3years
CHFGOVT	proportional	1	Effective	Simple	
CLPGOVT	proportional	As CpnFreq	Effective	Simple	
CZKGOVT	proportional	1	Effective	None	
CZKCORP	proportional	1	Effective	None	
CNYGOVT	proportional	As CpnFreq	Effective	None	
DEMGVT	proportional	1	Effective	Simple	
DEMGVTOLD	see EURGOVTOLD				
DKKGOVT	regular	1	Effective	None	
DKKGOVTOLD	regular	1	Effective	None	before 8-Feb-2001

ESPGOVT	regular	1	Effective	None	
ESPGOVTODD	regular	1	Effective	Simple	
ESPGOVTOLD	regular	1	Effective	None	
EURGOVT	proportional	1	Effective	None	
EURGOVTOLD	proportional	1	Effective	Simple	
FIMGGOVT	see EURGOVT				
FIMGGOVTOLD	regular	1	Effective	Simple	
FRFBTAN	proportional	1	Effective	Simple	
FRFOAT	proportional	1	Effective	Simple	
GBPGOVT	proportional	2	Effective	None	
GBPGOVTOLD	proportional	2	Effective	None	
GRDGOVT	proportional	1	Effective	None	
HKDGOVT	proportional	2	Effective	Simple	
IEPGOVT	proportional	1	Effective	Simple	Sept 1997 -
IEPGOVTOLD	proportional	1	Effective	Simple	1993 - Sept 1997
IEPGOVTOLD2	proportional	2	Effective	Simple	- 1992
INRGOVT	proportional	2	Effective	None	
ITLGOVT	regular	1	Effective	Simple	True yield
ITLGOVT2	regular	1	Effective	Simple	1) True yield, 2) Lastperiod is the two last coupon periods
ITLGOVTOLD	proportional	1	Effective	Simple	1) True yield, 2) Lastperiod is the two last coupon periods
JPYGOVT	proportional	n/a	n/a	None	
KRWGOVT	proportional	As CpnFreq	Effective	Simple	
LUFGOVT	proportional	As CpnFreq	Effective	Simple	
MTLGOVT	proportional	2	Effective	None	
NLGGGOVT	see EURGOVT				
NLGGGOVTOLD	see EURGOVTOLD				
NOKGOVT	regular	1	Effective	None	
NZDGOVT	regular	2	Effective	Simple	
PHPGOVT	proportional	2	Effective	None	
PLZGOVT	proportional	1	Effective	Simple	
PLZGOVTQ	proportional	1	Effective	None	coupon quarterly
PTEGOVT	proportional	1	Effective	None	
PTEGOVTOLD	see EURGOVTOLD				
PTEGOVTOLD2	proportional	1	Effective	None	
RUBGOVT	regular	1	Effective	Simple	
SEKGOVT	regular	1	Effective	Simple	
SEKGOVTODD	proportional	1	Effective	Simple	

SEKOM	regular	1	Effective	Simple	
SGDGOVT	proportional	2	Effective	Simple	
SKKGOVT	proportional	1	Effective	None	
SKKCORP	proportional	1	Effective	None	
THBGOVT	proportional	2	Effective	Simple	
TWDGOVT	proportional	as CpnFreq	Effective	Simple	
USDAGENCY	proportional	2	Effective	Simple	
USDCORP	proportional	2	Effective	Simple	
USDGOVT	proportional	2	Effective	Simple	Street Method
USDGOVTTTR	proportional	2	Simple	Simple	Treasury Method
USDMUNI	proportional	2	Effective	Simple	
ZARGOVT	proportional	as CpnFreq	Effective	None	
BOND	proportional	as CpnFreq	Effective	None	
BONDEFF	proportional	1	Effective	None	

## Inflation Linked Bonds

For inflation-linked bonds the following attributes are relevant.

CalcType	OddFirstCpn Style	Compound. Freq	NextPeriod Style	LastPeriod Style	Comment
ILEURGOVT	proportional	1	Effective	None	"CAD-Style"
ILFRFGOVT	proportional	1	Effective	None	"CAD-Style"
ILGBPGOVT	proportional	2	Effective	None	-
ILGBPGOVT3M	proportional	2	Effective	None	"CAD-Style"
ILSEKGOVT	regular	1	Effective	Simple	"CAD-Style"
ILUSDGOVT	proportional	2	Effective	Simple	"CAD-Style"

"CAD-Style" means Canadian index-linked bond style.

## Treasury Bills

For T-bills and similar instruments the following attributes are relevant.

Attribute	Description	Values
RateStyle	Rate type used for discounting.	Effective, Simple, BankDisc
Rate Type > 1yr	Rate type used for discounting when longer than 1 year.	Effective, Simple, BankDisc
Coupon	Restriction on coupon	
Coupon Style	Determines the size of the first coupon	Proportional

The following calculation methods are currently available.

CalcType	Rate Type	Rate Type > 1yr	Coupon	Coupon Style
AUDBILL	Simple	Simple	must be zero	n/a
AUSBILL	BankDiscount	BankDiscount	must be zero	n/a
BEFBILL	BankDiscount	BankDiscount	must be zero	n/a
BGLBILL	Simple	Simple	must be zero	n/a
CHFBILL	BankDiscount	BankDiscount	must be zero	n/a
CLPBILL	Simple	Simple	must be zero	n/a
CADBILL	Simple	Simple	must be zero	n/a
CZKBILL	BankDiscount	BankDiscount	must be zero	n/a
DKKBILL	Simple	Simple	must be zero	n/a
DEMBILL	see EURBILL			
GRDBILL	Simple	Simple	must be zero	n/a
EURBILL	Simple	Simple	must be zero	n/a
ESPBILL	Simple	Effective	must be zero	n/a
FIMBILL	Simple	Simple	must be zero	n/a
FIMBILLOLD	Simple	Simple	must be zero	n/a
FRFBILL	Simple	Simple	must be zero	n/a
GBPBILL	Simple	Simple	must be zero	n/a
GRDBILL	Simple	Simple	must be zero	n/a
HKDBILL	BankDiscount	BankDiscount	must be zero	n/a
ITLBILL	see EURBILL			
ITLBILLOLD	Simple	Simple	must be zero	n/a
JPYBILL	Simple	Simple	must be zero	n/a
MXPBILL	Simple	Simple	must be zero	n/a
NLGBILL	BankDiscount	BankDiscount	must be zero	n/a
NOKBILL	Effective	Effective		proportional
NZDBILL	Simple	Simple	must be zero	n/a

PHPBILL	Simple	Simple	must be zero	n/a
PLZBILL	Simple	Simple	must be zero	n/a
PTEBILL	Simple	Simple	must be zero	n/a
ROLBILL	Effective	Effective	must be zero	n/a
RUBBILL	BankDiscount	BankDiscount	must be zero	n/a
UAHBILL	BankDiscount	BankDiscount	must be zero	n/a
SEKBILL	Simple	Simple	must be zero	n/a
SGDBILL	BankDiscount	BankDiscount	must be zero	n/a
SKKBILL	Simple	Simple	must be zero	n/a
THBBILL	Simple	Simple	must be zero	n/a
TWDBILL	Simple	Simple	must be zero	n/a
TRLBILL	Effective	Effective	must be zero	n/a
UAHBILL	Simple	Simple	must be zero	n/a
USDBILL	BankDiscount	BankDiscount	must be zero	n/a
BANKDISC	BankDiscount	BankDiscount		proportional
BILL	Simple	Simple		proportional
ZERO	Effective	Effective		proportional
DEPO	Simple	Simple		proportional

## 11 Appendix 5. Bond definition examples

In this Appendix a number of bonds are described using the relevant instrument information. Blank or irrelevant fields have been omitted.

### Generic example: USTreasury 1

This bond has a normal first and normal last coupon period.

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example with a normal first period and a normal last period
G	Type	INSTR_TYPE	FIXED CPN BOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	6.875
G	Coupon freq	CPN_FREQ	2
G	Issue date	ISSUE_DATE	15-Dec-85
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 15-Dec-85)
E	Coupon start date	CPN_START_DATE	(defaults to 15-Dec-85)
G	First coupon date	FIRST_CPN_DATE	15-Jun-86
E	Last regular coupon date	LAST_REG_CPN_DATE	(defaults to 15-Sep-04)
G	Maturity   Date	MATURITY	15-Sep-04
G	Settle   Code	SETTLE_CODE	BD1
E	Value code	VALUE_CODE	(defaults to BD1)
E	End of month	EOM	TRUE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	USDGOVT
E	Rounding convention   Clean price	PRICE_ROUND_DEC	
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT
E	Coupon currency	CPN_CURRENCY	USD
G	Currency	PRINCIPAL_CURRENCY	USD
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to 6.875)

\*G= General, E = Extended, C = Comments



## Generic example: USTreasury 2

These bonds have a normal first and odd last coupon period

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example with a normal first period and an odd <i>long</i> last period
G	Type	INSTR_TYPE	FIXED CPN BOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	3.750
G	Coupon freq	CPN_FREQ	2
G	Issue date	ISSUE_DATE	15-Oct-85
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 15-Oct-85)
E	Coupon start date	CPN_START_DATE	(defaults to 15-Oct-85)
G	First coupon date	FIRST_CPN_DATE	15-Apr-86
E	Last regular coupon date	LAST_REG_CPN_DATE	15-Oct-92
G	Maturity   Date	MATURITY	15-Jun-93
G	Settle   Code	SETTLE_CODE	BD1
E	Value code	VALUE_CODE	(defaults to BD1)
E	End of month	EOM	TRUE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	USDGOVT
E	Rounding convention   Clean price	PRICE_ROUND_DEC	
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT
E	Coupon currency	CPN_CURRENCY	USD
G	Currency	PRINCIPAL_CURRENCY	USD
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to 3.750)

\*G= General, E = Extended, C = Comments

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example with a normal first period and an odd <i>short</i> last period
G	Type	INSTR_TYPE	FIXED CPN BOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	6.875
G	Coupon freq	CPN_FREQ	2
G	Issue date	ISSUE_DATE	15-Dec-85
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 15-Dec-85)
E	Coupon start date	CPN_START_DATE	(defaults to 15-Dec-85)
G	First coupon date	FIRST_CPN_DATE	15-Jun-86
E	Last regular coupon date	LAST_REG_CPN_DATE	15-Dec-03
G	Maturity   Date	MATURITY	15-Sep-04
G	Settle   Code	SETTLE_CODE	BD1
E	Value code	VALUE_CODE	(defaults to BD1)
E	End of month	EOM	TRUE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	USDGOVT
E	Rounding convention   Clean price	PRICE_ROUND_DEC	
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT
E	Coupon currency	CPN_CURRENCY	USD
G	Currency	PRINCIPAL_CURRENCY	USD
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to 6.875)

\*G= General, E = Extended, C = Comments

### Generic example: USTreasury 3

These bonds have an odd first and odd last coupon period.

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example with an odd <i>long</i> first period and an odd <i>long</i> last period
G	Type	INSTR_TYPE	FIXEDCPNBOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	5.000
G	Coupon freq	CPN_FREQ	2
G	Issue date	ISSUE_DATE	1-Jan-92
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 1-Jan-92)
E	Coupon start date	CPN_START_DATE	(defaults to 1-Jan-92)
G	First coupon date	FIRST_CPN_DATE	15-Dec-92
E	Last regular coupon date	LAST_REG_CPN_DATE	15-Dec-22
G	Maturity   Date	MATURITY	15-Dec-23
G	Settle   Code	SETTLE_CODE	BD1
E	Value code	VALUE_CODE	(defaults to BD1)
E	End of month	EOM	TRUE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	USDGOVT
E	Rounding convention   Clean price	PRICE_ROUND_DEC	
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT
E	Coupon currency	CPN_CURRENCY	USD
G	Currency	PRINCIPAL_CURRENCY	USD
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to a calculated coupon)

\*G= General, E = Extended, C = Comments

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example with an odd <i>short</i> first period and an odd <i>short</i> last period
G	Type	INSTR_TYPE	FIXEDCPNBOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	4.000
G	Coupon freq	CPN_FREQ	2
G	Issue date	ISSUE_DATE	1-Mar-93
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 1-Mar-93)
E	Coupon start date	CPN_START_DATE	(defaults to 1-Mar-93)
G	First coupon date	FIRST_CPN_DATE	1-Jul-93
E	Last regular coupon date	LAST_REG_CPN_DATE	1-Jan-20
G	Maturity   Date	MATURITY	1-Mar-20
G	Settle   Code	SETTLE_CODE	BD1
E	Value code	VALUE_CODE	(defaults to BD1)
E	End of month	EOM	TRUE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	USDGOVT
E	Rounding convention   Clean price	PRICE_ROUND_DEC	
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT
E	Coupon currency	CPN_CURRENCY	USD
G	Currency	PRINCIPAL_CURRENCY	USD
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to a calculated coupon)

\*G= General, E = Extended, C = Comments

## Example: Swedish Treasury

Example of a regular Swedish Treasury Bond

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example of a regular Swedish Treasury Bond
G	Type	INSTR_TYPE	FIXED CPN BOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	5.250
G	Coupon freq	CPN_FREQ	1
G	Issue date	ISSUE_DATE	15-Mar-01
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 15-Mar-01)
E	Coupon start date	CPN_START_DATE	(defaults to 15-Mar-01)
G	First coupon date	FIRST_CPN_DATE	15-Mar-02
E	Last regular coupon date	LAST_REG_CPN_DATE	(defaults to 15-Mar-11)
G	Maturity   Date	MATURITY	15-Mar-11
G	Settle   Code	SETTLE_CODE	BD3
E	Value code	VALUE_CODE	(defaults to BD3)
E	End of month	EOM	TRUE
G	Accrued daycount method	ACCR_DC_METHOD	EU30360
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to EU30360)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to EU30360)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to EU30360)
G	Calc type	CALC_TYPE	SEKGOVT
E	Rounding convention   Clean price	PRICE_ROUND_DEC	R3
G	Ex-coupon code	EXCPN_CODE	BD4
G	Quote style	QUOTE_STYLE	YLD
E	Coupon currency	CPN_CURRENCY	SEK
G	Currency	PRINCIPAL_CURRENCY	SEK
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to 5.250)

\*G= General, E = Extended, C = Comments

## Example: Spanish Treasury

Spanish Treasury Bonds with an odd long first period but a regular coupon.

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example of a Spanish Treasury Bond with an odd long first period with a regular first coupon
G	Type	INSTR_TYPE	FIXEDCPNBOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	5.150
G	Coupon freq	CPN_FREQ	1
G	Issue date	ISSUE_DATE	7-Jul-98
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 7-Jul-98)
E	Coupon start date	CPN_START_DATE	30-Jul-99
G	First coupon date	FIRST_CPN_DATE	30-Jul-00
E	Last regular coupon date	LAST_REG_CPN_DATE	(defaults to 30-Jul-09)
G	Maturity   Date	MATURITY	30-Jul-09
G	Settle   Code	SETTLE_CODE	BD3
E	Value code	VALUE_CODE	(defaults to BD3)
E	End of month	EOM	FALSE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	ESPGOVTODD
E	Rounding convention   Clean price	PRICE_ROUND_DEC	
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT
E	Coupon currency	CPN_CURRENCY	EUR
G	Currency	PRINCIPAL_CURRENCY	EUR
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to 5.150)

\*G= General, E = Extended, C = Comments

## Example: Bund

Example of a Bund bond with an odd short first period with a regular first coupon

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example of a Bund with an odd short first period with a regular first coupon
G	Type	INSTR_TYPE	FIXEDCPNBOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	3.750
G	Coupon freq	CPN_FREQ	1
G	Issue date	ISSUE_DATE	1-Aug-99
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 1-Aug-99)
E	Coupon start date	CPN_START_DATE	1-Apr-99
G	First coupon date	FIRST_CPN_DATE	1-Apr-00
E	Last regular coupon date	LAST_REG_CPN_DATE	(defaults to 1-Apr-09)
G	Maturity   Date	MATURITY	1-Apr-09
G	Settle   Code	SETTLE_CODE	BD3
E	Value code	VALUE_CODE	(defaults to BD3)
E	End of month	EOM	FALSE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	DEMGVOT
E	Rounding convention   Clean price	PRICE_ROUND_DEC	R3
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT
E	Coupon currency	CPN_CURRENCY	EUR
G	Currency	PRINCIPAL_CURRENCY	EUR
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to 3.750)

\*G= General, E = Extended, C = Comments

## Example: OATi

Example of a French inflation indexed-linked bond

Tab*	Dialog caption	Database column	Value
C	Comment	COMMENT	Example of a French inflation indexed bond
G	Type	INSTR_TYPE	FIXEDCPNBOND
G	Face amount	FACE_AMOUNT	100
G	Coupon	COUPON	3.40
G	Coupon freq	CPN_FREQ	1
G	Issue date	ISSUE_DATE	25-Jul-1999
G	Settle   Date	FIRST_SETTLE_DATE	(defaults to 25-Jul-1999)
E	Coupon start date	CPN_START_DATE	25-Jul-1999
G	First coupon date	FIRST_CPN_DATE	25-Jul-2000
E	Last regular coupon date	LAST_REG_CPN_DATE	(defaults to 25-Jul-2027)
G	Maturity   Date	MATURITY	25-Jul-2027
G	Settle   Code	SETTLE_CODE	BD3
E	Value code	VALUE_CODE	(defaults to BD3)
E	End of month	EOM	FALSE
G	Accrued daycount method	ACCR_DC_METHOD	ACTACT
E	Daycount method   Accrued (in last period)	ACCR_DC_LAST_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon	NEXT_DC_METHOD	(defaults to ACTACT)
E	Daycount method   To next coupon (in last period)	NEXT_DC_LAST_METHOD	(defaults to ACTACT)
G	Calc type	CALC_TYPE	ILFRFGOVT
E	Rounding convention codes   Clean price:	PRICE_ROUND_DEC	R2
E	Rounding convention codes   Accrued interest:	ACCR_ROUND_DEC	R3
G	Ex-coupon code	EXCPN_CODE	BD0
G	Quote style	QUOTE_STYLE	CLEAN_PCT_EXINFL
E	Coupon currency	CPN_CURRENCY	EUR
G	Currency	PRINCIPAL_CURRENCY	EUR
E	Coupon tax rate	CPN_TAX	0
E	Capital gains tax rate	CAPGAIN_TAX	0
E	Redemption price	REDEMP_PRICE	100
E	First coupon payment	FIRST_COUPON	(defaults to 3.40)
S	CPI Name	CPI_NAME	The name of the French CPI in your database
S	CPI base date	CPI_DATE	25-Jul-1999
S	CPI lag 1:	CPI_LAG1	2
S	CPI lag 2:	CPI_LAG2	3



<b>S</b>	<b>Quote style ex. inflation:</b>		TRUE
<b>S</b>	<b>Minimum redemption at par</b>	<b>MIN_REDEMP_AT_PAR</b>	TRUE

\*G= General, E = Extended, S = Special, C = Comments