SWX Swiss Exchange

Accrued Interest & Yield Calculations and Determination of Holiday Calendars

Author(s):	David Christie	Date:	Version, 03.11.2003
Classification:	Unrestricted	Reference:	SWX-SBD-MAN-AIC-202_/E
Keywords:	Accrued interest, coupon payments, SPD, SBD		
Approval:	Peter Keller	Distribution:	SWX Participant Back Offices, SIS, and others interested in the SWX interpretation of the ISMA actual/actual rules
Filename:	\\szh029\team\gbm\sbd\allgemein\Clearing & Settlement\Accrued Interest & Yields\V2.2\SWX-SBD-MAN-AIC-202_E.doc	Documentum:	

Converted on 25.09.2002 by dc. Old Reference-No: M-SPR-AIC-105/E

Table of Contents

1. Introduction

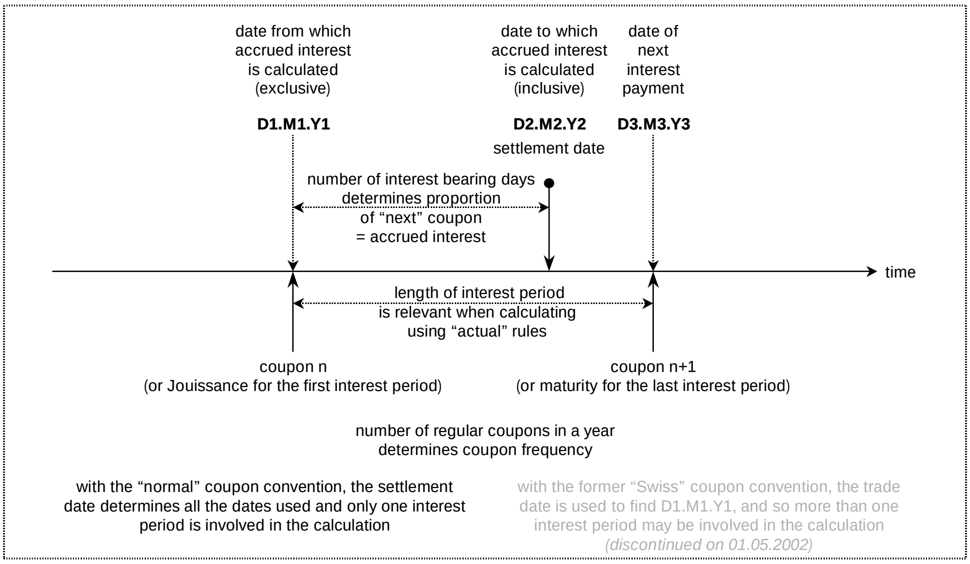
The introduction of the Euro and the start of SWX Eurobond trading required a revision of the available SWX accrued interest and yield calculation methods, since new day count methods were introduced for both from 4.1.1999 onward. The new SWX ISMA-99 methods are based on the ISMA rules which come into force at the beginning of 1999 (see the ISMA book “Bond Markets: Structures and Yield Calculations”, ISBN 1 901912 02 7, and ISMA’s Circular 14 of 1997). The US method is based on the 1994 edition of “Standard Securities Calculation Methods”, ASIN 1882936019, published by the Securities Industry Association.

This brief document covers all the accrued interest calculation methods supported by SWX and gives the SWX interpretation of the ISMA rules. ISMA has approved this interpretation. Because the determination of holiday calendars is so important to the correct calculation of accrued interest, the way this is carried out in SWX is covered in Appendix D. The document also incorporates a description of the yield calculation as implemented in the Trading System.

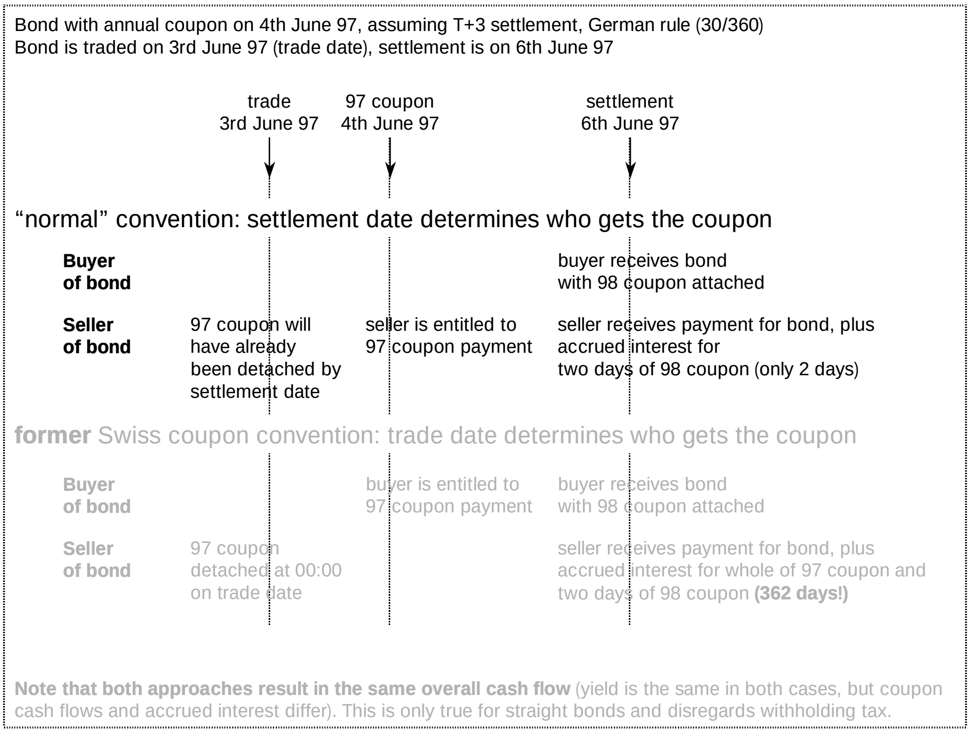
This is Version 2.2 of this document and has been updated to add NZD holiday information. There are no material changes to the calculation methods defined in this document compared with the previous version.

1.1 Definitions

The definitions in the following table should be read in conjunction with the following figure.


date from which accrued interest is calculated (exclusive)
date to which accrued interest is calculated (inclusive)
date of next interest payment
D1.M1.Y1
D2.M2.Y2
D3.M3.Y3
settlement date
number of interest bearing days determines proportion of "next" coupon = accrued interest
time
length of interest period is relevant when calculating using "actual" rules
coupon n (or Jouissance for the first interest period)
coupon n+1 (or maturity for the last interest period)
number of regular coupons in a year determines coupon frequency
with the "normal" coupon convention, the settlement date determines all the dates used and only one interest period is involved in the calculation
with the former "Swiss" coupon convention, the trade date is used to find D1.M1.Y1, and so more than one interest period may be involved in the calculation
(discontinued on 01.05.2002)

term	definition
accrued interest	The accrued interest represents the proportion of the coupon amount to which the seller is entitled. The basis of the calculation is the assumption that the buyer receives the full coupon payment and must pay the seller that part of the coupon representing the period between the previous coupon payment and the settlement date
coupon frequency	The number of (regular) coupon payments in a year
CSD	Central Securities Depository
D1.M1.Y1	The date from which accrued interest is calculated
D2.M2.Y2	The date to which accrued interest is calculated
D3.M3.Y3	The date of the next relevant interest payment
date ranges	Where the actual number of days between two dates as per calendar is required, the earlier date is **excluded** from the range and the later date is **included.** By convention, bonds are "ex coupon" on an interest payment date
flat flag	Indication of whether or not accrued interest is included in the calculation of settlement amount for a trade in a given security
"in default from" date	The date from which no more interest payments are expected as a result of default of the company issuing the debt. No accrued interest is paid for trades occurring on or after this date (if set)
Jouissance	The first date of interest entitlement In the case of a re-opening of a bond, Jouissance for the new tranche is set to the most recent interest payment date that has just passed for the original bond, whereas Liberierung (q.v.) is in the future. This ensures that the correct amount of accrued interest is calculated for the new tranche (the same amount as for the original bond, but from Liberierung onwards). SWX Platform field name: Interest_Entitlement_Start_Date
Liberierung	First date when a bond is officially traded: the date on which subscription payments are due SWX Platform field name: Subscription_Payment_Due_Date
maturity	The last date of interest entitlement
non-versé	A factor to take account of partly paid-up issues
settlement date	The date on which the trade will be settled. For SWX trades, this is identical to the value date for the transaction, by convention (delivery versus payment)

term	definition
Swiss coupon convention	In Switzerland, the right to future coupon payments formerly went to the buyer on the trade date, whereas in most other markets, the right to future coupon payments goes to the buyer on the settlement date. The Swiss convention only applied to Swiss domestic bonds listed on SWX, and was discontinued in the Swiss market as of 01.05.2002. In both cases, the seller has a right to compensation for interest (and coupons) accrued. The result of the former Swiss coupon convention is that accrued interest additionally compensated an entire coupon if the coupon date fell between the trade date (exclusive) and the settlement date (inclusive). See the following example:  Bond with annual coupon on 4th June 97, assuming T+3 settlement, German rule (30/360) Bond is traded on 3rd June 97 (trade date), settlement is on 6th June 97 trade 3rd June 97 97 coupon 4th June 97 settlement 6th June 97 “normal” convention: settlement date determines who gets the coupon **Buyer of bond** buyer receives bond with 98 coupon attached **Seller of bond** 97 coupon will have already been detached by settlement date seller is entitled to 97 coupon payment seller receives payment for bond, plus accrued interest for two days of 98 coupon (only 2 days) **former** Swiss coupon convention: trade date determines who gets the coupon **Buyer of bond** buyer is entitled to 97 coupon payment buyer receives bond with 98 coupon attached **Seller of bond** 97 coupon detached at 00:00 on trade date seller receives payment for bond, plus accrued interest for whole of 97 coupon and two days of 98 coupon **(362 days!)** **Note that both approaches result in the same overall cash flow** (yield is the same in both cases, but coupon cash flows and accrued interest differ). This is only true for straight bonds and disregards withholding tax.
trade date	The date on which the trade took place

1.2 Example Microsoft Excel Spreadsheet & MS Visual Basic Implementation

The example Microsoft Excel spreadsheet supplied with this document implements the rules given here for accrued interest calculations and yields[1]. It also covers the currently known holiday calendars for the various settlement currencies supported by the SWX platform[2]. Values can be entered into the fields with a coloured background.

A number of simplifications have been made in order to keep the spreadsheet straightforward, as follows:

- The spreadsheet “generates” the series of coupons for a given bond based on the Jouissance, First and Last Regular Coupon, Maturity dates and Ultimo dating flag values supplied. Up to 199 coupons can be handled correctly. If the “Ultimo dating” flag is set to “Yes”, then the assumption is made that coupon dates fall at the end of the month. The ISMA-99 rule that is not appropriate is greyed out accordingly.
- Only regularly repeating coupon periods are supported: if the period length is not a multiple of a number of months, then an approximation is made, by counting in months and days.
- The settlement date is calculated from the trade date, according to the settlement period and business day setting supplied. Officially published currency holidays in the period 1.1.1999 to 31.12.2001 are taken into account in the business day calculation. Otherwise, a currency holiday calendar based on the “standard” approach followed in the year 2000 is used. All current SWX trading currencies are included.
- When calculating notional periods, up to 15 periods can be handled.

The MS Visual Basic procedure given in Appendix C, which fully implements the accrued interest calculations described in this document, has been tested against the spreadsheet and gives identical results within the limitations noted above.

The “rules” used in the accrued interest calculation are indicated by a rule number in the margin. X

The rules are cross-referenced to the source code. For the ISMA-99 rules, Appendix A contains the detailed rules, and §3.4 a summary. X

[1] The spreadsheet was developed using MS Excel for Windows 97 (Excel Version 8.0 and Visual Basic Version 5.0). It may not work correctly under other versions.

[2] The year and currency selected are based on the Trade Date and Settlement Currency entered on the Accrued Interest sheet

2. Accrued Interest Methods Supported by SWX

The SWX supports the following day count methods for the purposes of accrued interest calculations. The two ISMA-99 methods only differ in their handling of coupon dates: otherwise they are the same.

NB: Throughout, where the actual number of days between two dates as per calendar is required**, the earlier date is excluded** from the range and **the later date is included**. In most cases the earlier date is an interest payment date and represents a day on which a coupon has already been detached[3] (and so should not be counted towards accrued interest). The number of days can be simply calculated as the difference between two date values on most computer systems.

day count method *[ISO 15022 keyword in field :22F::MICO//**aaaa**]*	**basic rule for determining number of interest bearing days (see §3.3)**	**basic rule for determining the length of a "year" (see §3.4)**	**remarks**
Flat (No Accrued Interest) *[A000]*	–	–	no accrued interest
German (30/360) *[A011]*	30 days/month	360 days/year	This is the method currently used in the Swiss marketplace
Special German (30S/360) *[A007]*	30 days/month, except for February which is 28 or, in leap years, 29 days	360 days/year	This is the method formerly used in the Eurobond markets (the ISMA rule for bonds issued until the end of 1998)
English (Actual/365) *[A005]*	days/month as per calendar	365 days/year	This is the method used in the UK Gilt Edged Market (which moved to actual/actual from 1.11.98)
French (Actual/360) *[A004]*	days/month as per calendar	360 days/year	This method is used in the French money markets
US (30U/360) *[A001]*	30 days/month, US variant	360 days/year	This is the 30/360 day count method typically used in the US
ISMA-Year (Actual/365L) *[A009]*	days/month as per calendar	365 or 366 days/year	The number of days/year used depends on the coupon frequency and whether a leap year is involved
ISMA-99 Normal (Actual/Actual) *[A006]* ISMA-99 Ultimo (Actual/Actual) *[A010]*	days/month as per calendar	days/year as per calendar (based on interest period length)	These methods apply to new issues in the Eurobond markets from the beginning of 1999 (hence the SWX names). They also apply to new Euro denominated bonds from the same date. "Converted" Euro denominated bonds need not change convention: the recommendation is to change convention on a coupon date, if at all. See Appendix A for a more detailed discussion of the two ISMA-99 methods.
	*Note that the ISMA-99 rules given in this document are the **SWX interpretation** of the ISMA rules, and cover exceptional cases not fully addressed by those rules – this can mean that the accrued interest calculated by SWX may differ from that calculated in other markets under exceptional circumstances.*		

[3] By convention, bonds are "ex coupon" on an interest payment date

3. Calculation of Accrued Interest

The calculation of the accrued interest amount for a particular trade date proceeds as follows:

- Determination of the settlement date
- Determination of the date of the previous coupon payment and date to which accrued interest is calculated
- Determination of the number of days for which interest is to be accrued
- Calculation of the accrued interest, including the effects of non-versé.

If the “flat flag” is set for the bond in question, then no accrued interest is calculated.

3.1 Determination of the Settlement Date

The settlement date for spot trades in a security for a particular trade date is determined using:

- The defined settlement cycle for the security (e.g. T+3, which means that the settlement date is the trade date plus three business days).
- The (so-called) weekend day calendar
- The settlement currency holiday calendar for the security in question
- The clearing organisation holiday calendar for the security in question
- The date upon which subscription payments are due ("Liberierung").

When counting days forward from the trade date to determine the settlement date, all non-business days are skipped. For these purposes, a business day is defined as a weekday on which neither the settlement currency nor the settlement organisation has a holiday. If the trade date is a non-business day, and a T+0 settlement cycle is used, then the next following business day is the settlement day. See Appendix D for further details.

For example, for a security settled in USD (the 4th July is a currency holiday in USD), if the trade date is Monday, 1st July, the settlement date is Friday 5th July. For a trade date of Tuesday 2nd July, the settlement date is Monday 8th July. For a security settled in CHF and traded on Monday 1st July, however, the settlement date is Thursday 4th July.

If the calculated settlement date for a security is earlier than “Liberierung”, then “Liberierung” is used.

If one of the following conditions is true, there is **no** accrued interest:

- The settlement date is earlier than or falls on the first date of interest entitlement ("Jouissance")
- The former Swiss coupon convention applies and the trade date is later than or equal to maturity
- The former Swiss coupon convention does not apply and the settlement date falls on an interest payment date
- The former Swiss coupon convention does not apply and the settlement date is later than or equal to maturity
- The trade date is equal to the “in default from” date for the bond (where it has been set) or after it
- The “null” day count method is selected.

3.2 Determination of Accrued Interest Dates

The date from which accrued interest is calculated **(D1.M1.Y1)** depends on whether the former Swiss coupon convention applies or not:

- If the former Swiss coupon convention applies, then it is the date of the start of the interest period within which the **trade date** falls. If the trade date is before Jouissance, but the settlement date falls after it or on it, then Jouissance is used for D1.M1.Y1
- Otherwise, it is the date of the start of the interest period within which the **settlement date** falls.

The date to which accrued interest is calculated **(D2.M2.Y2)** is by default the **settlement date,** unless it is later than or equal to the maturity date of the bond, in which case D2.M2.Y2 is set to the maturity date.

For the ISMA-Year, ISMA-99 Normal and ISMA-99 Ultimo day count methods, it is necessary to use the next relevant interest payment date **(D3.M3.Y3)** in the calculation. This is defined as follows:

- For the ISMA-Year day count method, D3.M3.Y3 is defined as the next interest payment date following D1.M1.Y1[4]
- For the ISMA-99 Normal and ISMA-99 Ultimo day count methods, D3.M3.Y3 is defined as the coupon payment date for the interest payment period within which the **settlement date** falls, or maturity if the settlement date is later than or equal to the maturity date of the bond[5].

The SWX security reference data contains a table INTEREST_PAYMENT, with one row for each interest payment period (see the example in Appendix B). The appropriate interest payment period in each case is located by searching for the row where the reference date (trade date or settlement date as adjusted) is equal to or later than the start of interest period, and earlier than the coupon payment date.

The relevant INTEREST_PAYMENT record for determining coupon amount, interest currency, interest rate and interest exchange rate is the period starting with D1.M1.Y1.

Note that if changes to security reference data occur in the interval between the trade date and the settlement date, then the values valid on the **trade** date are used.

3.3 Determination of Number of Interest-bearing Days

The dates D1.M1.Y1 (start date) and D2.M2.Y2 (end date) define the period over which interest is accrued. The number of interest-bearing days depends on the day count method defined for the bond in question.

day count method	details	number of interest-bearing days, N
German (30/360)	30 days/month	**N = (D2 – D1) + 30 * (M2 – M1) + 360 * (Y2 – Y1)** 1 This method requires special handling for D1 or D2 for months with 31 days and for February: • If D1 or D2 is 31, then use the value 30 instead. Thus, on the 31st of a month the number of accrued interest days is the same as that on the 30th • The last day of February is treated as the 30th day of the month.

[4] D3.M3.Y3 is thus the coupon payment date for the INTEREST_PAYMENT record starting with D1.M1.Y1. Note that this is the case even if the next interest payment date falls before D2.M2.Y2. This can occur if the Swiss coupon convention applies.

[5] Note that it is not necessary to search for a new INTEREST_PAYMENT record where the normal coupon convention applies or if D2.M2.Y2 is equal to or later than D1.M1.Y1 and earlier than the coupon payment date for the INTEREST_PAYMENT record starting with D1.M1.Y1. In these cases, both D1.M1.Y1 and D2.M2.Y2 fall in the same interest period, and D3.M3.Y3 is simply equal to the coupon payment date for the selected INTEREST_PAYMENT record.

day count method	**details**	**number of interest-bearing days, N**	
Special German (30S/360)	30 days/month, except for February which is 28 or, in leap years, 29 days	**N = (D2 – D1) + 30 * (M2 – M1) + 360 * (Y2 – Y1)** This method requires special handling for D1 or D2 for months with 31 days: • If D1 or D2 is 31, then use the value 30 instead. Thus, on the 31st of a month the number of accrued interest days is the same as that on the 30th • The last day of February is not treated specially.	2
English (Actual/365)	days/month as per calendar	**N = number of days between D1.M1.Y1 and D2.M2.Y2**	3
French (Actual/360)	days/month as per calendar	**N = number of days between D1.M1.Y1 and D2.M2.Y2**	4
US (30U/360)	30 days/month, US variant	**N = (D2 – D1) + 30 * (M2 – M1) + 360 * (Y2 – Y1)** This method requires special handling for D1 or D2 for months with 31 days and for February. The following rules are applied in the following order: • if D2 is the last day of February (28 in a non leap year; 29 in a leap year) and D1 is the last day of February, change D2 to 30 • if D1 is the last day of February, change D1 to 30 • if D2 is 31 and D1 is 30 or 31, change D2 to 30 • if D1 is 31, change D1 to 30	5
ISMA-Year (Actual/365L)	days/month as per calendar	**N = number of days between D1.M1.Y1 and D2.M2.Y2**	6
ISMA-99 Normal (Actual/Actual) ISMA-99 Ultimo (Actual/Actual)	days/month as per calendar	**N = number of days between D1.M1.Y1 and D2.M2.Y2**	7

3.4 Calculation of the Basic Accrued Interest Amount

In SWX, the basic accrued interest amount is calculated for the smallest tradeable unit of the bond[6] and is expressed in the interest currency. It depends on the day count method, as follows. The calculation to take account of partially paid up issues, if required, is given in §3.5. Note that in the SWX reference data, coupon amounts are always expressed as an *annualised* value.

day count method	**details**	**basic accrued interest amount, A** (F = annual coupon frequency; N = number of interest-bearing days)	
German (30/360)	360 days/year	**A = coupon amount * (N / 360)**	8
Special German (30S/360)	360 days/year	**A = coupon amount * (N / 360)**	9
English (Actual/365)	365 days/year	**A = coupon amount * (N / 365)**	10
French (Actual/360)	360 days/year	**A = coupon amount * (N / 360)**	11

[6] The coupon amount value required by the calculation is not necessarily identical to the value stored in INTEREST_PAYMENT. Where the smallest tradeable unit of the bond is not equal to the smallest denomination, it is adjusted accordingly.

day count method	**details**	**basic accrued interest amount, A** (F = annual coupon frequency; N = number of interest-bearing days)
US (30U/360)	360 days/year	**A = coupon amount * (N / 360)** 12
ISMA-Year (Actual/365L)	365 or 366 days/year	**A = coupon amount * (N / Y)** 13 where **Y** is given by the following: • If the coupon frequency is annual, then **Y** is 366 if the 29 February is included in the interest period, else **Y** is 365[7] 14 • If the coupon frequency is not annual, then **Y** is 366 for each interest period where D3.M3.Y3 falls in a leap year, else **Y** is 365. 15
ISMA-99 Normal (Actual/Actual) ISMA-99 Ultimo (Actual/Actual)	days/year as per calendar (based on interest period length)	See Appendix A for full details and examples of the ISMA-99 methods and for a detailed definition of “regular” and “periodic”. 16 The only difference between the two ISMA-99 methods is the assumption made about when regular coupons 19 fall: • The *ISMA-99 Normal* method assumes that regular coupons always fall on the *same* day of the month where possible 23 • The *ISMA-99 Ultimo* method assumes that regular coupons always fall on the *last* day of the month. 24 Where the coupon period is *regular* (by definition, this only applies to *periodic* coupon frequencies): **A = (coupon amount / F) * (N / C)** 17 where **C** = number of days between D1.M1.Y1 and D3.M3.Y3. For *aperiodic* coupon frequencies, and in *irregular* interest periods, the approach used is to divide the interest period into notional interest periods. 18 Normally the notional periods are generated backwards from D3.M3.Y3, but if D3.M3.Y3 equals the maturity date, then the notional periods are generated forwards from D1.M1.Y1: • For *aperiodic* coupon frequencies, the notional interest periods are 12 months long, and the applicable coupon frequency **F’** is 1. 21 • For *irregular* interest periods with a *periodic* coupon frequency, the notional interest periods are [12/F] months long, and the applicable coupon frequency **F’** is F. 20 The basic accrued interest amount is given by the following formula: $\mathbf{A = (coupon\ amount / F') * \sum_i (N_i / C_i)}$ 22 where **F’** is the applicable coupon frequency, $\mathbf{N_i}$ is the number of days of accrued interest falling into period **i**, and $\mathbf{C_i}$ = the length of period **i** in days.

[7] Note that for regular annual periods this rule is equivalent to counting the actual number of days between D1.M1.Y1 and D3.M3.Y3**.** Only where this value is neither 365 nor 366 (note that it is possible to have non regular periods irregardless of day count method), is it necessary to search the interest period from D1.M1.Y1 (**ex**clusive) to D3.M3.Y3 (**in**clusive) for an occurrence of 29 February. If one is found, then **Y** is set to 366, else **Y** is set to 365.

3.5 Inclusion of Non Versé

SWX reference data includes an attribute for each security that defines whether or not any non-versé effects have to be included in the calculation of the settlement amount for a trade. This attribute affects both the calculation of the settlement amount from the trade size and price, and the calculation of accrued interest.

If non versé is to be included in settlement, then the basic accrued interest amount is multiplied by the fraction of the issue which has been paid up (interest is only paid for the paid up fraction):

$$\textbf{accrued interest amount} = \textbf{basic accrued interest amount} * \frac{\textbf{(100 – non versé)}}{\textbf{100}}$$

4. Calculation of Interest Payments

Although SWX does not organise the payment of coupons (this is carried out by SIS in Switzerland), it is helpful to indicate the amount of interest that is paid in the normal case, and in the case where an interest period has an “irregular” length. The amount is dependent on the day count method as shown in the following table. For the purposes of this section, D1.M1.Y1 is set to the date of the previous interest payment and *both* D2.M2.Y2 *and* D3.M3.Y3 are set to the current interest payment date. The coupon frequency is F. The basic interest payment amount is calculated for the smallest tradeable unit of the bond and is expressed in the interest currency. Coupon amounts are expressed as an *annualised* value. The calculation to take account of partially paid up issues, if required, is analogous to that given in §3.5.

	basic interest payment amount, P	
day count method	**regular interest period**[8]	**irregular interest period**
German (30/360)	**P = coupon amount / F**	**P = coupon amount * (N / 360)** where **N** is the number of days between D1.M1.Y1 and D2.M2.Y2 counted using the German method (see §3.3)
Special German (30S/360)	**P = coupon amount / F**	**P = coupon amount * (N / 360)** where **N** is the number of days between D1.M1.Y1 and D2.M2.Y2 counted using the Special German method (see §3.3)
English (Actual/365)	**P = coupon amount / F**	**P = coupon amount * (N / 365)** where **N** is the number of calendar days between D1.M1.Y1 and D2.M2.Y2
French (Actual/360)	**P = coupon amount / F**	**P = coupon amount * (N / 360)** where **N** is the number of calendar days between D1.M1.Y1 and D2.M2.Y2
US (30U/360)	**P = coupon amount / F**	**P = coupon amount * (N / 360)** where **N** is the number of days between D1.M1.Y1 and D2.M2.Y2 counted using the US method (see §3.3)
ISMA-Year (Actual/365L)	**P = coupon amount / F**	**P = coupon amount * (N / Y)** where **N** is the number of days between D1.M1.Y1 and D2.M2.Y2 and **Y** is determined as given for the ISMA-Year method in §3.4
ISMA-99 Normal (Actual/Actual) ISMA-99 Ultimo (Actual/Actual)	**P = coupon amount / F**	Notional interest periods are generated (see §3.4 and Appendix A for further details), and: **P = (coupon amount / F’) * $\sum_i$ (N_i / C_i)** where **F’** is the applicable coupon frequency, **N_i** is the number of days falling in period **i**, and **C_i** = the length of period **i** in days

[8] This only applies for **periodic** coupon frequencies where the interest payment period is of regular length (see definitions on page 14)

5. Calculation of Yields

The yield formula currently used by the SWX Trading System is as follows. It is based on the simple yield to maturity formula adjusted to take better account of the redemption value[9].

$$\textbf{Yield} = \frac{g + [\,(C - CP) / L\,]}{(C + CP) / 2} * 100$$

If the relevant redemption date has already been reached, then the following formula is used. It is based on the current yield to maturity formula.

$$\textbf{Yield} = \frac{g}{CP} * 100$$

where:

- **g** = Annual Coupon Rate (in percent)
- **CP** = Clean Price (in percent)
- **C** = Redemption Value at maturity date or at earliest redemption (in percent)
- **L** =Life to maturity/earliest redemption in years = **D** / **Y**

L is based on the day count scheme as follows. The number of days, **D**, from today, D1.M1.Y1, to the relevant redemption date, D2.M2.Y2, is counted using the applicable day count method (see the description in M-SPR-AIC)[10]. This is then divided by the applicable number of days in a year, **Y**. An extended rule is used for the ISMA-Year and ISMA-99 day count methods.

day count method	**number of days to relevant redemption date** (D2.M2.Y2)**, D**	**year divisor, Y**	**life to maturity/earliest redemption, L**
German (30/360)	count the number of days using the German day count method	360	**D** / **Y**
Special German (30S/360)	count the number of days using the Special German day count method	360	**D** / **Y**
English (Actual/365)	the number of days is the actual number of calendar days	365	**D** / **Y**
French (Actual/360)	the number of days is the actual number of calendar days	360	**D** / **Y**
US (30U/360)	count the number of days using the US day count method	360	**D** / **Y**

[9] See the UBS publication “Formulas, Ratios and Indexes used in Banking”, and §5 in the ISMA book “Bond Markets: Structures and Yield Calculations”, ISBN 1 901912 02 7 for further details.

[10] As with other date ranges, the period start date (i.e., today) is **ex**clusive, and the relevant redemption date is **in**clusive

day count method	**number of days to relevant redemption date** (D2.M2.Y2)**, D**	**year divisor, Y**	**life to maturity/earliest redemption, L**
ISMA-Year (Actual/365L) ISMA-99 Normal (Actual/Actual) ISMA-99 Ultimo (Actual/Actual)	Following the same procedure as for calculating the basic accrued interest amount for “aperiodic coupon frequencies”, split the period to the relevant redemption date into notional annual periods, counting *backwards* in years from the relevant redemption date (D2.M2.Y2). If the relevant redemption date is the last day in February, the generated dates depend on the day count method as follows: • For **ISMA-Year** and **ISMA-99 Normal**, each newly generated date is the same day in February as D2.M2.Y2, substituting 28 February only where necessary (if the relevant redemption date happens to be 29 February). As with the basic accrued interest calculation, for each new notional period the start date is always calculated with reference to D2.M2.Y2, and not by counting from each newly calculated date. This avoids systematic “drift” in the dates. • For **ISMA-99 Ultimo**, each newly generated date is always the last day in February. The notional periods break down into part of a year and full years: • The last generated period (chronologically the earliest) in general contains a broken part of a year • For this period, divide the actual number of days by 365 unless there is a leap day included in the period in which case divide by 366. Note that the divisor can be obtained by counting the actual number of days between D2’.M2.[Y2-n-1] and D2’.M2.[Y2-n]. • The remaining periods are full years.		The sum of the fraction from the broken period and the integral number of years from the remaining periods gives the result required.

Appendix A. The ISMA-99 Methods

A.1 Details of the ISMA-99 Methods

The ISMA-99 methods make a distinction between regular and irregular interest periods: 16

- A **regular** interest period is always an exact multiple of a number of months long

  Determining for the accrued interest calculation whether the coupon period in question is regular or not is carried out by comparing the length of the period between D1.M1.Y1 and D3.M3.Y3 with the length of a *single* regular interest period as implied by the coupon frequency[11].

  The basic accrued interest amount for regular interest periods is given by:

  $A = (\text{coupon amount} / F) * (N / C)$ 17

  where **C** = number of days between D1.M1.Y1 and D3.M3.Y3.

- For **irregular** interest periods, it is necessary to generate “notional” interest periods. The interest bearing days are then spread over the notional periods, each with an appropriate daily accrual rate. In irregular coupon periods, the daily accrued interest rate is not necessarily constant throughout the period, but “jumps” in value at each notional coupon date.

  Notional interest periods are normally generated by counting backwards from the next coupon date (D3.M3.Y3). For irregular last coupons however, where D3.M3.Y3 is equal to the maturity date, the notional periods count forwards from the date from which accrued interest is calculated (D1.M1.Y1). The reason for this approach is that it aligns the notional periods with actual interest payment dates for most known bonds (generally speaking only the first and last interest periods are irregular, if at all). The start date for the notional periods (D3.M3.Y3 or D1.M1.Y1) is termed the *anchor date* below. 18

  For each new notional period, the start date is always calculated with reference to the anchor date by counting in multiples of the notional interest period length, and not by counting from each newly calculated date. This avoids systematic “drift” in the dates.

  SWX makes a distinction between **periodic** and **aperiodic** coupon frequencies, as follows: 19

  - SWX only recognises coupon frequencies **F** of 1x, 2x, 3x, 4x, 6x, and 12x per year as **periodic** (where F >= 1 and INT[12/F] = [12/F])**.** In these cases, each regular interest period is always an exact number of months long.

    For *periodic* coupon frequencies where the interest period is irregular, notional interest periods are generated at intervals of [12/F] months, and the applicable coupon frequency **F’** is F. 20

  - For all other, **aperiodic,** coupon frequencies (where F < 1 or INT[12/F] ≠ [12/F]), each interest period is *always regarded as irregular.*

    Notional interest periods are generated at yearly intervals, using the same date in the year as the anchor date, and substituting 28 February if the anchor date is 29 February where necessary. The applicable coupon frequency **F’** is 1. 21

  The basic accrued interest amount for irregular interest periods is given by:

  $A = (\text{coupon amount} / F') * \sum_i (N_i / C_i)$ 22

[11] The coupon frequency must be periodic: note that SWX only recognises coupon frequencies **F** of 1x, 2x, 3x, 4x, 6x, and 12x per year as **periodic** (see definition on this page)

where **F’** is the applicable coupon frequency, $\mathbf{N_i}$ is the number of days of accrued interest falling into period **i**, and $\mathbf{C_i}$ = the length of period **i** in days.

The two ISMA-99 methods only differ in their handling of coupon dates, as follows:

- ISMA-99 Normal

  Regular coupons are all assumed to fall on the **same** day in the month, at exact monthly intervals, where possible. Otherwise, where the given date does not exist (e.g., 31 June), the last day in the given month is used. If the interest period concerned (D1.M1.Y1 to D3.M3.Y3) does not meet these criteria, it is regarded as irregular, requiring the generation of notional interest periods. 23

  When notional interest periods are generated, the same day in the month is used, where possible. Otherwise, the last day in the given month is substituted. For instance, if there are four coupons per year and the next coupon falls on 29.02.96, then the first few “notional coupon dates” counting backwards in steps of three months would be: 29.11.95, 29.08.95, 29.05.95, 28.02.95, 29.11.94.

- ISMA-99 Ultimo

  Regular coupons are all assumed to fall on the **last** day in the month, at exact monthly intervals. Otherwise, the interest period concerned (D1.M1.Y1 to D3.M3.Y3) is regarded as irregular, requiring the generation of notional interest periods. Note that the “ISMA-99 Ultimo” method *only* makes sense when regular coupons always fall at the end of a month. 24

  When notional periods are generated, the last day in the given month is always used, even if D3 is less than 28. For instance, if there are four coupons per year and the next coupon falls on 28.02.97, then the first few “notional coupon dates” counting backwards in steps of three months would be: 30.11.96, 31.08.96, 31.05.96, 29.02.96, 30.11.95.

When it is unclear from the prospectus or other information available to SWX which of the two methods is appropriate and the regular coupons fall on the last day in the month, then SWX will assume that the ISMA-99 Ultimo method applies, otherwise the ISMA-99 Normal method will be assumed.

Where the former Swiss coupon convention applies and a coupon falls between the trade and settlement dates, two or more consecutive interest periods are used together in the calculation of accrued interest. In this case, the combined interest period is always treated as irregular, and notional periods are generated (note that this will automatically be the case, since the number of days between D1.M1.Y1 and D3.M3.Y3 is always greater than one regular interest period).

A.2 ISMA-99 Examples

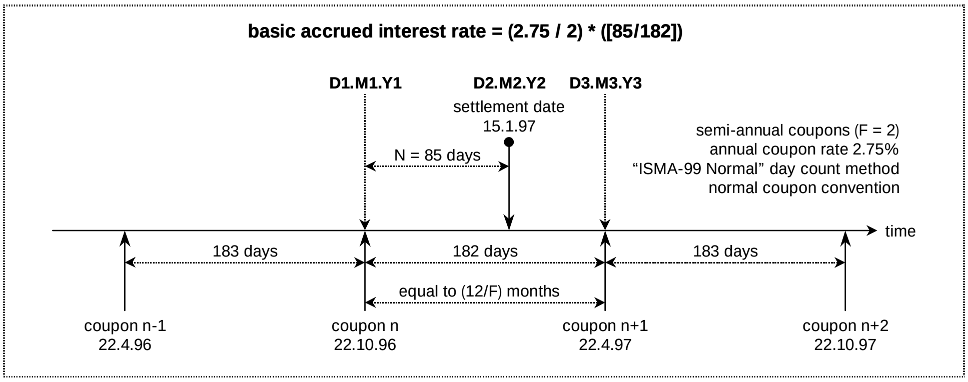


Figure 1 Example of regular interest period with semi-annual coupon payments

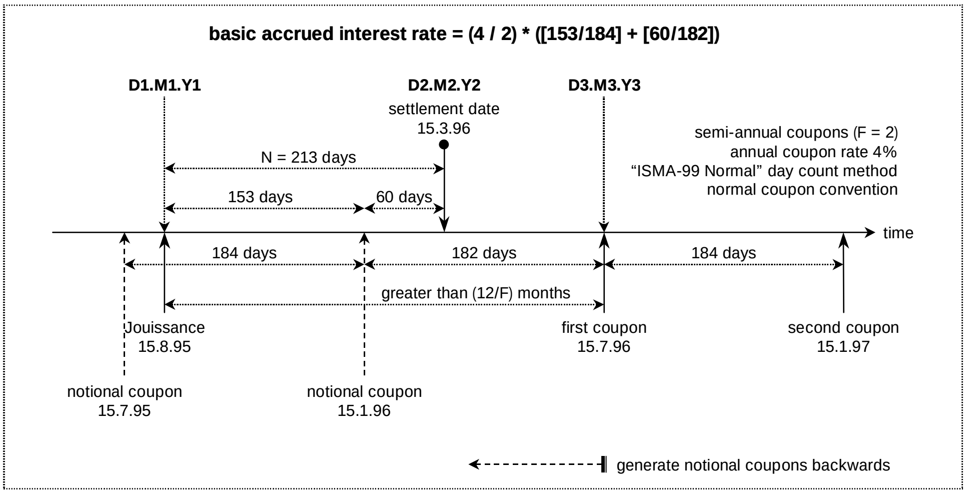


Figure 2 Example of irregular (long) initial interest period with semi-annual coupon payments

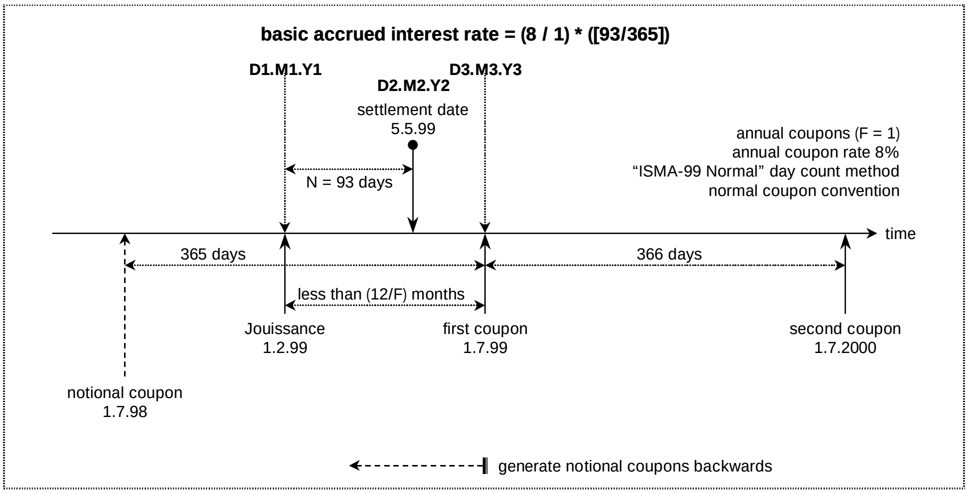


Figure 3 Example of irregular (short) initial interest period with annual coupon payments

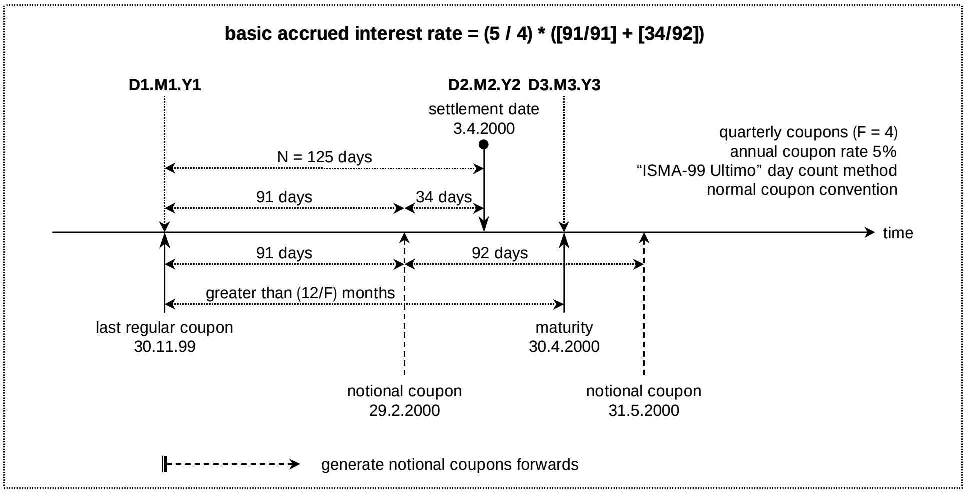


Figure 4 Example of irregular (long) final interest period with quarterly coupon payments

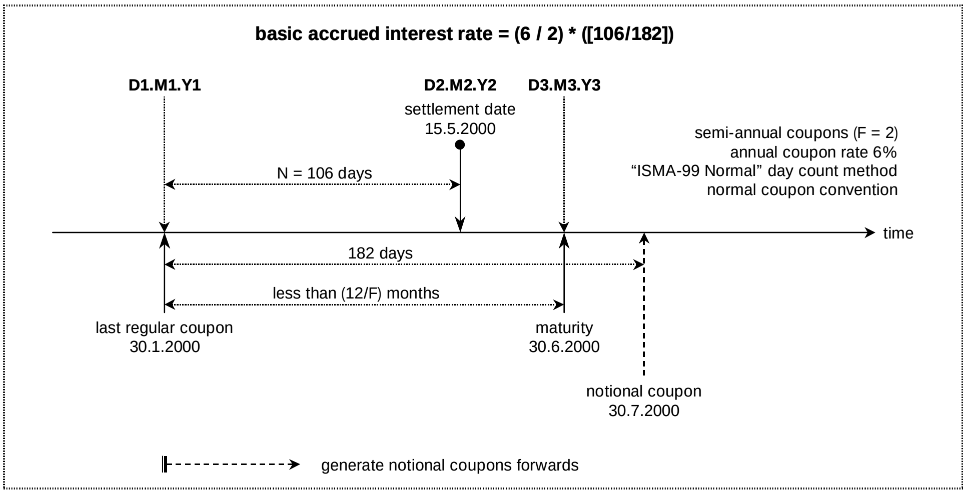


Figure 5 Example of irregular (short) final interest period with semi-annual coupon payments

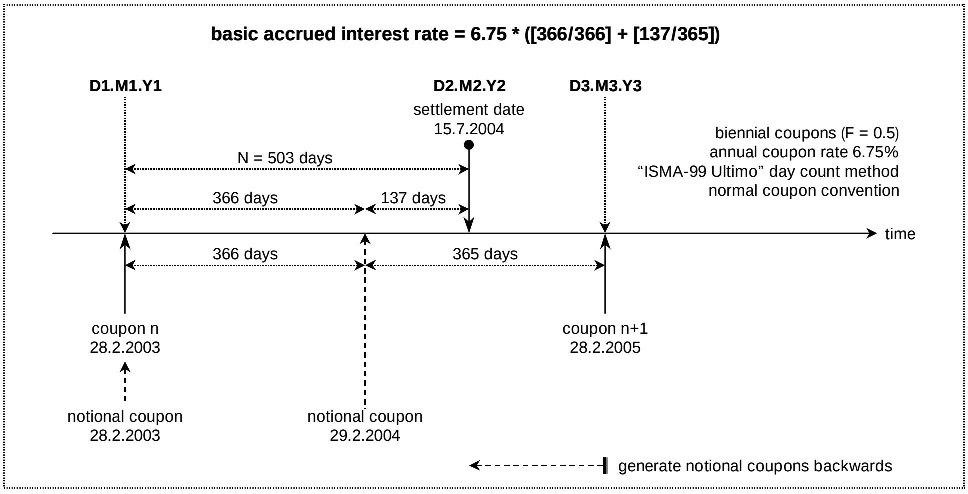


Figure 6 Example of aperiodic interest period with biennial coupon payments

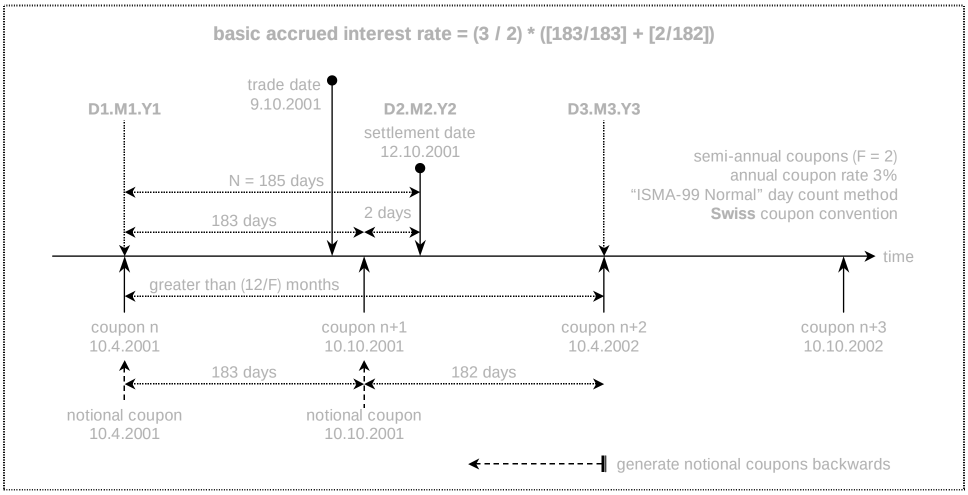


Figure 7 *Example of interest period with semi-annual coupon payments (Swiss convention)*

Appendix B. Interest_Payment Example

The following example is a single INTEREST_PAYMENT record for the Foreign Interest Payment Security "PepsiCo 7½%" (valor number 894 131) which uses a fixed exchange rate of CHF 1.8690 = USD 1. For further details, see the latest version of the “Member Interface Functional Description”, X-TSD-MID.

INTEREST_PAYMENT

Attribute	Meaning	Example
start of interest period *(Interest_Period_ Start_Date)*	The start date of the period for which the coupon applies. This is equal to the date of the previous coupon payment. The first interest period starts with Jouissance.	18.04.92
coupon payment date *(Interest_Period_ End_Date)*	The end date of the period for which the coupon applies. The last interest period ends on the maturity date.	18.04.93
interest currency *(Interest_Payment_ Currency_Code)*	The currency in which the interest is paid.	USD
coupon amount *(Coupon_Amount)*	The monetary amount of the (annualised) coupon payment for the smallest denomination expressed in the interest currency. In many cases this value is derived from the interest rate and the value of the smallest tradeable unit.	200.642055 (based on a smallest tradeable unit of CHF 5 000, the fixed exchange rate, and the annual interest rate)
interest rate *(Interest_Rate)*	The annualised interest expressed as a percentage.	7.5
interest exchange rate *(Interest_Exchange_ Rate)*	Used for Foreign Interest Payment Securities – this is the fixed exchange rate used to convert the coupon amount calculated from the interest rate applied to the nominal into an amount in the interest currency.	0.535045

Appendix C. MS Visual Basic Implementation of the Accrued Interest Calculations

The following implementation of the function AI_Factor has been tested against the example spreadsheet supplied[12]. AI_Factor returns the accrued interest factor by which the adjusted coupon amount should be multiplied to get the basic accrued interest amount. The function assumes that the input dates D1.M1.Y1 etc. have already been calculated. In the example, these dates are calculated in the supplied spreadsheet: in real-world systems, database access is required to find the correct values. AI_Factor can be provided in text form on request to SWX Swiss Exchange.

In order to understand this function, it is important to know that dates are held in “serial” form in an 8-byte integer variable in MS Visual Basic: serial dates monotonically increase, with each date exactly 1 greater than the date preceding. For example, the serial value for 28 Feb 1995 is 1 less than the serial value for 1 Mar 1995, whereas the serial values for 28 Feb 1996 and 1 Mar 1996 differ by 2. A similar approach is used for dates in most computer systems. MS Visual Basic provides a number of functions for handling serial dates. The ones used here are as follows:

- DateSerial (YYYY, MM, DD)

  This function converts the date DD.MM.YYYY into its serial form.
- Day (SD)
- Month (SD)
- Year (SD)

  These functions return the day, month, and year parts of the serial date SD respectively.

Here are a few notes to help the reader who is less accustomed to Visual Basic. The **Long** data type holds a 4-byte integer value. The **Double** data type holds an 8-byte floating-point number. The **Date** data type holds an 8-byte serial date value. The “\” operator is an integer division (fractional part of result truncated). The **mod** operator returns the remainder after dividing the first operand by the second (for example, 13 **mod** 12 = 1). The **Dim** statement declares storage for a given variable. Continuation from one line to the next is indicated by a trailing “ _” at the end of the preceding line.

[12] The spreadsheet was developed using MS Excel for Windows 97 (Excel Version 8.0 and Visual Basic Version 5.0). It may not work correctly under other versions.

```
'
' Calculate_accrued_interest
'
Public Const GERMAN = 1, SPEC_GERMAN = 2, ENGLISH = 3, FRENCH = 4, _
    US = 5, ISMA_YEAR = 6, ISMA_99N = 7, ISMA_99U = 8                   ' day count methods
Public Const ERROR_BAD_DCM = -1, ERROR_BAD_DATES = -2                   ' error returns

Public Function AI_Factor(ByVal DCM As Long, _
                    ByVal D1M1Y1 As Date, ByVal D2M2Y2 As Date, ByVal D3M3Y3 As Date, _
                    ByVal F As Double, _
                    ByVal Maturity As Date _
                    ) As Double
' DCM is day count method code
' D1M1Y1 is D1.M1.Y1 etc.
' F is the coupon frequency
' Maturity is the maturity date
    Dim D1, D1x, M1, Y1 As Long                 ' variables used to hold day, month and year
    Dim D2, D2x, M2, Y2 As Long                 '   values separately
    Dim D3, M3, Y3 As Long
    Dim Anchor As Date                          ' "anchor" date: start date for notional...
    Dim AD, AM, AY As Long                      '   ...periods
    Dim WM, WY As Long                          ' "working dates" in notional period loop
    Dim Target As Date                          ' end date for notional period loop
    Dim N, Nx As Long                           ' number of interest-bearing days
    Dim Y As Double                             ' length of a year (ISMA-Year)
    Dim L As Long                               ' regular coupon length in months
    Dim C, Cx As Double                         ' notional period length in days
    Dim Fx As Double                            ' applicable coupon frequency
    Dim Periodic, Regular As Boolean            ' various flags
    Dim Direction As Long
    Dim CurrC, NextC, TempD As Date             ' used for temporary serial date values
    Dim i As Long                               ' temporary loop variable

'
'   Check input dates
'
    If (D1M1Y1 = 0) Or (D2M2Y2 = 0) Or (D3M3Y3 = 0) Or _
            (D2M2Y2 < D1M1Y1) Or (D3M3Y3 <= D1M1Y1) Or (Maturity = 0) Then
        AI_Factor = ERROR_BAD_DATES
        Exit Function
    End If
'
'   Determine Number of Interest-bearing days, N
'
    Select Case DCM
    Case GERMAN                                                                 ' RULE 1
        D1 = Day(D1M1Y1): M1 = Month(D1M1Y1): Y1 = Year(D1M1Y1)
        D2 = Day(D2M2Y2): M2 = Month(D2M2Y2): Y2 = Year(D2M2Y2)
        If D1 = 31 Then
            D1x = 30
        ElseIf IsFebUltimo(D1M1Y1) Then         ' end of February
            D1x = 30
        Else
            D1x = D1
        End If
        If D2 = 31 Then
            D2x = 30
        ElseIf IsFebUltimo(D2M2Y2) Then         ' end of February
            D2x = 30
        Else
            D2x = D2
        End If
        N = (D2x – D1x) + 30 * (M2 – M1) + 360 * (Y2 – Y1)
    Case SPEC_GERMAN                                                              ' RULE 2
        D1 = Day(D1M1Y1): M1 = Month(D1M1Y1): Y1 = Year(D1M1Y1)
        D2 = Day(D2M2Y2): M2 = Month(D2M2Y2): Y2 = Year(D2M2Y2)
        If D1 = 31 Then
            D1x = 30
        Else
            D1x = D1
        End If
        If D2 = 31 Then
            D2x = 30
        Else
            D2x = D2
        End If
        N = (D2x – D1x) + 30 * (M2 – M1) + 360 * (Y2 – Y1)

    Case ENGLISH, FRENCH, ISMA_YEAR, ISMA_99N, ISMA_99U                   ' RULES 3, 4, 6, 7
        N = D2M2Y2 – D1M1Y1

Case US                                                                       ' RULE 5
        D1 = Day(D1M1Y1): M1 = Month(D1M1Y1): Y1 = Year(D1M1Y1)
        D2 = Day(D2M2Y2): M2 = Month(D2M2Y2): Y2 = Year(D2M2Y2)

        D1x = D1: D2x = D2
        If IsFebUltimo(D1M1Y1) And IsFebUltimo(D2M2Y2) Then
            D2x = 30
        End If
        If IsFebUltimo(D1M1Y1) Then
            D1x = 30
        End If
        If (D2x = 31) And (D1x >= 30) Then
            D2x = 30
        End If
        If D1x = 31 Then
            D1x = 30
        End If

        N = (D2x – D1x) + 30 * (M2 – M1) + 360 * (Y2 – Y1)

    Case Else
        AI_Factor = ERROR_BAD_DCM
        Exit Function
    End Select

'
'   Determine Basic Accrued Interest Factor
'
    Select Case DCM
    Case GERMAN, SPEC_GERMAN, FRENCH, US                                  ' RULES 8, 9, 11, 12
        AI_Factor = N / 360#                     ' force double precision arithmetic!

    Case ENGLISH                                                                  ' RULE 10
        AI_Factor = N / 365#                     ' force double precision arithmetic!
    Case ISMA_YEAR
        D1 = Day(D1M1Y1): M1 = Month(D1M1Y1): Y1 = Year(D1M1Y1)
        D3 = Day(D3M3Y3): M3 = Month(D3M3Y3): Y3 = Year(D3M3Y3)

        If F = 1 Then                                                             ' RULE 14
            i = (D3M3Y3 – D1M1Y1)
            If (i = 365) Or (i = 366) Then
                Y = i
            Else
                Y = 365
                For i = Y1 To Y3
                    TempD = GetUltimo(i, 2)      ' last day in February
                    If (Day(TempD) = 29) And (TempD > D1M1Y1) And (TempD <= D3M3Y3) Then
                        Y = 366
                        Exit For
                    End If
                Next i
            End If
        Else                                                                      ' RULE 15
            If ((Y3 Mod 4 = 0) And (Y3 Mod 100 <> 0)) Or (Y3 Mod 400 = 0) Then
                Y = 366
            Else
                Y = 365
            End If
        End If

        AI_Factor = N / Y                                                         ' RULE 13

    Case ISMA_99N, ISMA_99U
        D1 = Day(D1M1Y1): M1 = Month(D1M1Y1): Y1 = Year(D1M1Y1)
        D3 = Day(D3M3Y3): M3 = Month(D3M3Y3): Y3 = Year(D3M3Y3)

        ' check whether the frequency is periodic or not and look if the period is regular
        '   set up default values (assume aperiodic, irregular unless otherwise)
        Periodic = False                 ' aperiodic
        L = 12                           '   regular period length in months
        Fx = 1                           '   applicable coupon frequency
        Regular = False

        If F >= 1 Then                                                            ' RULE 21
            If (12 \ F) = (12 / F) Then                                   ' RULES 19, 20
                Periodic = True              ' periodic
                L = 12 \ F                   '   regular period length in months
                Fx = F                       '   applicable coupon frequency
                Regular = False              '   default: not regular

                If ((Y3 – Y1) * 12 + (M3 – M1)) = L Then                  ' RULES 23, 24
                    If DCM = ISMA_99N Then   ' ISMA-99 Normal
                        If (D1 = D3) Then
                            Regular = True
                        ElseIf InvalidDate(Y1, M1, D3) And IsUltimo(D1M1Y1) Then
                            Regular = True
                        ElseIf InvalidDate(Y3, M3, D1) And IsUltimo(D3M3Y3) Then
                            Regular = True
                        End If
                    Else                     ' ISMA-99 Ultimo
                        If IsUltimo(D1M1Y1) And IsUltimo(D3M3Y3) Then Regular = True
                    End If
                End If
            End If
        End If
        If Regular Then                                                         ' RULE 17
            C = (D3M3Y3 – D1M1Y1)
            AI_Factor = (1 / Fx) * (N / C)

        Else                            ' generate notional periods
            AI_Factor = 0#
            If D3M3Y3 = Maturity Then                                           ' RULE 18
                Direction = 1           '   ... forwards
                Anchor = D1M1Y1
                AY = Y1: AM = M1: AD = D1
                Target = D3M3Y3
            Else
                Direction = -1          '   ... backwards
                Anchor = D3M3Y3
                AY = Y3: AM = M3: AD = D3
                Target = D1M1Y1
            End If

            CurrC = Anchor              ' start notional loop
            i = 0
            While Direction * (CurrC – Target) < 0
                i = i + Direction
                WY = GetNewYear(AY, AM, (i * L))   ' next notional year and...
                WM = GetNewMonth(AM, (i * L))      '   ...month (handling year changes)

                If DCM = ISMA_99N Then          ' ISMA-99 Normal
                    If InvalidDate(WY, WM, AD) Then                             ' RULE 23
                        NextC = GetUltimo(WY, WM)
                    Else
                        NextC = DateSerial(WY, WM, AD)
                    End If
                Else                            ' ISMA-99 Ultimo
                    NextC = GetUltimo(WY, WM)                                   ' RULE 24
                End If

                Nx = Min(D2M2Y2, Max(NextC, CurrC)) – Max(D1M1Y1, Min(CurrC, NextC))
                Cx = Direction * (NextC – CurrC)
                If Nx > 0 Then                                                  ' RULE 22
                    AI_Factor = AI_Factor + (Nx / Cx)                           ' RULE 21
                End If
                CurrC = NextC
            Wend

            AI_Factor = AI_Factor / Fx                                          ' RULE 22
        End If
    Case Else
    End Select
End Function
```

```
Function Min(ByVal A As Date, ByVal B As Date) As Date      ' lesser of two dates
    If A < B Then Min = A Else Min = B
End Function

Function Max(ByVal C As Date, ByVal D As Date) As Date      ' greater of two dates
    If C > D Then Max = C Else Max = D
End Function

Public Function GetUltimo(ByVal YY As Long, _
                    ByVal MM As Long) As Date               ' last day in month MM.YY
                                                            '   NB: MM.YY must be valid
' if MM = 12 then handles new year boundary
    GetUltimo = DateSerial(YY + (MM \ 12), (MM Mod 12) + 1, 1) – 1
End Function

Public Function IsUltimo(ByVal DS As Date) As Boolean       ' is DS last day in month
    IsUltimo = (Day(DS + 1) = 1)
End Function

Public Function IsFebUltimo(ByVal DS As Date) As Boolean    ' is DS last day in February
    IsFebUltimo = (Day(DS + 1) = 1) And (Month(DS + 1) = 3)
End Function

Public Function InvalidDate(ByVal YY As Long, ByVal MM As Long, _
                    ByVal DD As Long) As Boolean            ' check if a valid date
    InvalidDate = (Month(DateSerial(YY, MM, 1) + DD – 1) <> MM)
End Function

Public Function GetNewMonth(ByVal MM As Long, _
                    ByVal Num As Long) As Long          ' new month MM +/- Num months
    Dim NM As Long                                      '   NB: MM must be valid
    NM = MM + Num
    If NM > 0 Then
        GetNewMonth = (NM – 1) Mod 12 + 1
    Else
        GetNewMonth = 12 + (NM Mod 12)
    End If
End Function

Public Function GetNewYear(ByVal YY As Long, _
                    ByVal MM As Long, _
                    ByVal Num As Long) As Long          ' get new year starting from MM.YY
    Dim NM As Long                                      '   going +/- Num months (MM.YY valid)
    NM = MM + Num
    If NM > 0 Then
        GetNewYear = YY + ((NM – 1) \ 12)
    Else
        GetNewYear = YY – 1 + (NM \ 12)
    End If
End Function
```

Appendix D. Determination of Holiday Calendars

D.1 Introduction

The SWX Platform uses a number of different holiday calendars when counting business days forward from the trade date.

- The (so-called) **Weekend Day** calendar
- The settlement **Currency Holiday** calendar for the security in question
- The **Clearing Organisation Holiday** calendar for the security in question.

A given day is regarded as a *non-business day* for the settlement of a given security if it is contained in *any* of the three relevant holiday calendars above. Note that the Market Holiday calendar (when the market is open) is *not relevant* for the purpose of determining the settlement date.

When counting days forward from the trade date to determine the settlement date, all non-business days are skipped. If the trade date is a non-business day, and a T+0 settlement cycle is used, then the next following business day is the settlement day.

The rest of this Appendix details how the various calendars are determined for the SWX Platform. Broadly, the following operational procedure is followed:

- Once a year latest in October, the holiday calendars for the next calendar year are established and entered into the SWX Platform

  The primary sources depend on the calendar type concerned and are detailed below
- Three times a year, in December (before Christmas and the New Year), early March (before Easter and Whitsun) and in June, the calendars are compared with market sources for possible discrepancies

  The comparison sources depend on the calendar type concerned and are detailed below. If unresolvable differences are discovered, then market participants are consulted.

D.2 Weekend Day Calendar

There is only one Weekend Day calendar, which only contains weekends.

- Primary sources

  The calendar is set up once a year to contain *only* the non weekdays
- Comparison sources

  The calendar is checked for the forthcoming period to ensure that it does indeed only contain weekends.

D.3 Currency Calendars

Each currency used in the SWX Platform has a defined currency holiday calendar. The *settlement* currency is relevant for the security in question.

The intention of the Currency Holiday calendar is to define on which days it is not possible to make a cash settlement in the currency in question. Since settlement of most currencies is carried out in the central bank system of the country in question, the holiday calendar is generally defined by the central bank, but there are exceptions (see below).

The currently known holidays can be found in the current version of the associated spreadsheet.

- Primary sources

  The primary sources used depend on the currency and are as follows:

  - AUD: Reserve Bank of Australia (the holidays in New South Wales apply; e-mail contact: mailto:southersm@rba.gov.au)
  - CAD: Federal Bank of Canada (see http://www.bank-banque-canada.ca) and Canadian Bankers Association
  - CHF: Swiss National Bank (see http://www.snb.ch)
  - DKK: Danmarks Nationalbank (see http://www.riksbank.com)
  - EUR: TARGET holidays (as published by the European Central Bank)
  - GBP: Department of Trade and Industry (England and Wales)
  - JPY: Bank of Japan (see http://www.boj.or.jp/en/about/holi.htm; note that the dates of the vernal and autumnal equinoxes play a role in defining Japanese holidays)
  - NOK: Norges Bank (see http://www.norges-bank.no)
  - NZD: Federal Reserve Bank of New Zealand (see http://www.rbnz.govt.nz/payment/ESAS/index.html)
  - SEK: Sveriges Riksbank (see http://www.nationalbanken.dk)
  - USD: Federal Reserve Bank of New York [Federal Holidays], **plus** Good Friday [NYSE] (see http://www.ny.frb.org/bankinfo/services/frsholi.html and http://www.ny.frb.org/bankinfo/circular/11087.html)
  - ZAR: Reserve Bank of South Africa (see http://www.resbank.co.za with email contact: mailto:info@resbank.co.za)

  As a cross-check, the very comprehensive International Bank Holidays calendar published by the Banque Générale du Luxembourg is used (see http://www.bgl.lu). ISMA publishes holiday information (see http://www.isma.org), and http://www.holidayfestival.com is also a good source.

- Comparison sources

  As comparison sources, the following are used:

  - Bloomberg and Reuters
  - the sub-custodian bank of SIS in the home market of the currency concerned (information available via http://www.sisclear.com).

D.4 Clearing Organisation Calendars

Each security in the SWX Platform is associated with a particular Clearing Organisation, which defines which *set* of CSDs can be used to settle the security in question.

The intention of the Clearing Organisation Holiday calendar is to define on which days it is not possible to settle in the CSDs concerned. Generally, most major European CSDs (including SIS) are open on all weekdays except January 1 and December 25.

- Primary sources

  Information provided by the CSDs themselves is used. The Clearing Organisation Holiday is the union of all individual holidays for the set of CSDs involved. See:

  - SIS: http://www.sisclear.com
  - CRESTCo: http://www.crestco.co.uk
  - Euroclear: http://www.euroclear.com
  - Clearstream Banking Frankfurt & Luxembourg: http://www.clearstream.com.

- Comparison sources

  A double-check is made, and SIS is also canvassed about the CSDs with which a direct link exists.