Package com.opengamma.strata.basics.date

Class DayCounts

java.lang.Object
com.opengamma.strata.basics.date.DayCounts

```
public final class DayCounts
extends Object
```

Constants and implementations for standard day count conventions.

The purpose of each convention is to define how to convert dates into numeric year fractions. The is of use when calculating accrued interest over time.

Field Summary

Fields

Modifier and Type	Field	Description
`static DayCount`	`ACT_360`	The 'Act/360' day count, which divides the actual number of days by 360.
`static DayCount`	`ACT_364`	The 'Act/364' day count, which divides the actual number of days by 364.
`static DayCount`	`ACT_365_25`	The 'Act/365.25' day count, which divides the actual number of days by 365.25.
`static DayCount`	`ACT_365_ACTUAL`	The 'Act/365 Actual' day count, which divides the actual number of days by 366 if a leap day is contained, or by 365 if not.
`static DayCount`	`ACT_365F`	The 'Act/365F' day count, which divides the actual number of days by 365 (fixed).
`static DayCount`	`ACT_365L`	The 'Act/365L' day count, which divides the actual number of days by 365 or 366.
`static DayCount`	`ACT_ACT_AFB`	The 'Act/Act AFB' day count, which divides the actual number of days by 366 if a leap day is contained, or by 365 if not, with additional rules for periods over one year.
`static DayCount`	`ACT_ACT_ICMA`	The 'Act/Act ICMA' day count, which divides the actual number of days by the actual number of days in the coupon period multiplied by the frequency.
`static DayCount`	`ACT_ACT_ISDA`	The 'Act/Act ISDA' day count, which divides the actual number of days in a leap year by 366 and the actual number of days in a standard year by 365.

DayCount		omitting leap days by 360.
static DayCount	NL_365	The 'NL/365' day count, which divides the actual number of days omitting leap days by 365.
static DayCount	ONE_ONE	The '1/1' day count, which always returns a day count of 1.
static DayCount	THIRTY_360_ISDA	The '30/360 ISDA' day count, which treats input day-of-month 31 specially.
static DayCount	THIRTY_360_PSA	The '30/360 PSA' day count, which treats input day-of-month 31 and end of February specially.
static DayCount	THIRTY_E_360	The '30E/360' day count, which treats input day-of-month 31 specially.
static DayCount	THIRTY_E_360_ISDA	The '30E/360 ISDA' day count, which treats input day-of-month 31 and end of February specially.
static DayCount	THIRTY_E_365	The '30E/365' day count, which treats input day-of-month 31 and end of February specially.
static DayCount	THIRTY_EPLUS_360	The '30E+/360' day count, which treats input day-of-month 31 specially.
static DayCount	THIRTY_U_360	The '30U/360' day count, which treats input day-of-month 31 and end of February specially.
static DayCount	THIRTY_U_360_EOM	The '30U/360 EOM' day count, which treats input day-of-month 31 and end of February specially.

Method Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

ONE_ONE

```
public static final DayCount ONE_ONE
```

The '1/1' day count, which always returns a day count of 1.

ACT_ACT_ISDA

```
public static final DayCount ACT_ACT_ISDA
```

The 'Act/Act ISDA' day count, which divides the actual number of days in a leap year by 366 and the actual number of days in a standard year by 365.

The result is calculated in two parts. The actual number of days in the requested period that fall in a leap year is divided by 366. The actual number of days in the requested period that fall in a standard year is divided by 365. The result is the sum of the two. The first day in the period is included, the last day is excluded.

Also known as 'Actual/Actual'. Defined by the 2006 ISDA definitions 4.16b.

ACT_ACT_ICMA

```
public static final DayCount ACT_ACT_ICMA
```

The 'Act/Act ICMA' day count, which divides the actual number of days by the actual number of days in the coupon period multiplied by the frequency.

The result is calculated as follows.

First, the underlying schedule period is obtained treating the first date as the start of the schedule period.

Second, if the period is a stub, then nominal regular periods are created matching the schedule frequency, working forwards or backwards from the known regular schedule date. An end-of-month flag is used to handle month-ends. If the period is not a stub then the schedule period is treated as a nominal period below.

Third, the result is calculated as the sum of a calculation for each nominal period. The actual days between the first and second date are allocated to the matching nominal period. Each calculation is a division. The numerator is the actual number of days in the nominal period, which could be zero in the case of a long stub. The denominator is the length of the nominal period multiplied by the frequency. The first day in the period is included, the last day is excluded.

Due to the way that the nominal periods are determined ignoring business day adjustments, this day count is recommended for use by bonds, not swaps.

The method `DayCount.yearFraction(LocalDate, LocalDate)` will throw an exception because schedule information is required for this day count.

Also known as 'Actual/Actual ICMA' or 'Actual/Actual (Bond)'. Defined by the 2006 ISDA definitions 4.16c and ICMA rule 251.1(iii) and 251.3 as later clarified by ISDA 'EMU and market conventions' http://www.isda.org/c_and_a/pdf/mktc1198.pdf.

ACT_ACT_AFB

```
public static final DayCount ACT_ACT_AFB
```

denominator is 366 if the schedule period contains February 29th, if not it is 365. The first day in the schedule period is included, the last day is excluded.

Also known as 'Actual/Actual AFB' or 'Actual/Actual (Euro)'. Defined by the Association Francaise des Banques in September 1994 as 'Base Exact/Exact' in 'Definitions Communes plusieurs Additifs Techniques'.

OpenGamma implements this day count based on the original French documentation without the ISDA clarification. The ISDA document translates "Periode d'Application" to "Calculation Period" and then assigns the regular ISDA meaning of "Calculation Period". Examination of the original French indicates that "Periode d'Application" simply means the period that the day count is applied to, not a regular periodic schedule.

In addition, the ISDA document adds a roll back rule stating that if the period ends on the 28th February it should be rolled back to the 28th, or to the 29th in a leap year. Unfortunately, this rule has a strange effect when implemented, with one day receiving two days interest and the next receiving no interest:

```
 From 2004-02-28 to 2008-02-27, ISDA rule = 3 + 365 / 366
 From 2004-02-28 to 2008-02-28, ISDA rule = 4 + 1 / 366
 From 2004-02-28 to 2008-02-29, ISDA rule = 4 + 1 / 366
```

(Other strange examples occur from 2004-02-29 and 2003-03-01).

OpenGamma interprets the roll back rule to be that if the period ends on the *29th* February it should be rolled back to the 28th, or to the 29th in a leap year. This change (which can be argued is closer to the original French than the ISDA "clarification") results in the following:

```
 From 2004-02-28 to 2008-02-27, OpenGamma interpretation = 3 + 365 / 366
 From 2004-02-28 to 2008-02-28, OpenGamma interpretation = 4
 From 2004-02-28 to 2008-02-29, OpenGamma interpretation = 4 + 1 / 366
```

Original French (from 1999 as 1994 version cannot be found): http://www.banque-france.fr/fileadmin/user_upload/banque_de_france/archipel/publications/bdf_bof/bdf_bof_1999/bdf_bof_01.pdf
ISDA "clarification": http://www.isda.org/c_and_a/pdf/ACT-ACT-ISDA-1999.pdf

ACT_ACT_YEAR

```
public static final DayCount ACT_ACT_YEAR
```

The 'Act/Act Year' day count, which divides the actual number of days by the number of days in the year from the start date.

The result is calculated as follows in two parts - a number of whole years and the remaining part.

If the period is over one year, a number of years is added to the start date to reduce the remaining period to less than a year. If the start date is February 29th, then each time a year is added the last valid day in February is chosen.

The remaining period is then processed by a simple division. The numerator is the actual number of days in the remaining period. The denominator is the actual number of days in the year from the adjusted start date.

This is a variation of the 'Act/Act ICMA' day count. If 'Act/Act ICMA is called with a frequency of yearly, the next coupon date equal to the start date plus one year and the end-of-month flag set to false, then the result will be the same for periods less than a year.

ACT_365_ACTUAL

```
public static final DayCount ACT_365_ACTUAL
```

The 'Act/365 Actual' day count, which divides the actual number of days by 366 if a leap day is contained, or by 365 if not.

The result is a simple division. The numerator is the actual number of days in the requested period. The denominator is 366 if the period contains February 29th, if not it is 365. The first day in the period is excluded, the last day is included.

Also known as 'Act/365A'.

ACT_365L

```
public static final DayCount ACT_365L
```

The 'Act/365L' day count, which divides the actual number of days by 365 or 366.

The result is a simple division. The numerator is the actual number of days in the requested period. The denominator is determined by examining the frequency and the period end date (the date of the next coupon). If the frequency is annual then the denominator is 366 if the period contains February 29th, if not it is 365. The first day in the period is excluded, the last day is included. If the frequency is not annual, the denominator is 366 if the period end date is in a leap year, if not it is 365.

The method `DayCount.yearFraction(LocalDate, LocalDate)` will throw an exception because schedule information is required for this day count.

Also known as 'Act/365 Leap year'. Defined by the 2006 ISDA definitions 4.16i and ICMA rule 251.1(i) part 2 as later clarified by ICMA and Swiss Exchange.

ACT_360

```
public static final DayCount ACT_360
```

The 'Act/360' day count, which divides the actual number of days by 360.

The result is a simple division. The numerator is the actual number of days in the requested period. The denominator is always 360.

ACT_364

```
public static final DayCount ACT_364
```

The 'Act/364' day count, which divides the actual number of days by 364.

The result is a simple division. The numerator is the actual number of days in the requested period. The denominator is always 364.

Also known as 'Actual/364'.

ACT_365F

```
public static final DayCount ACT_365F
```

The 'Act/365F' day count, which divides the actual number of days by 365 (fixed).

The result is a simple division. The numerator is the actual number of days in the requested period. The denominator is always 365.

Also known as 'Act/365', 'Actual/365 Fixed' or 'English'. Defined by the 2006 ISDA definitions 4.16d.

ACT_365_25

```
public static final DayCount ACT_365_25
```

The 'Act/365.25' day count, which divides the actual number of days by 365.25.

The result is a simple division. The numerator is the actual number of days in the requested period. The denominator is always 365.25.

NL_360

```
public static final DayCount NL_360
```

The 'NL/360' day count, which divides the actual number of days omitting leap days by 360.

The result is a simple division. The numerator is the actual number of days in the requested period minus the number of occurrences of February 29. The denominator is always 360. The first day in the period is excluded, the last day is included.

Also known as 'NoLeap/360', 'Actual/360 No Leap' or 'Actual (no leap year)/360'.

The result is a simple division. The numerator is the actual number of days in the requested period minus the number of occurrences of February 29. The denominator is always 365. The first day in the period is excluded, the last day is included.

Also known as 'NoLeap/365', 'Actual/365 No Leap' or 'Actual (no leap year)/365'.

THIRTY_360_ISDA

`public static final DayCount THIRTY_360_ISDA`

The '30/360 ISDA' day count, which treats input day-of-month 31 specially.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 360`. The deltaDay is calculated once day-of-month adjustments have occurred. If the second day-of-month is 31 and the first day-of-month is 30 or 31, change the second day-of-month to 30. If the first day-of-month is 31, change the first day-of-month to 30.

Also known as '30/360 U.S. Municipal' or '30/360 Bond Basis'. Defined by the 2006 ISDA definitions 4.16f.

THIRTY_U_360

`public static final DayCount THIRTY_U_360`

The '30U/360' day count, which treats input day-of-month 31 and end of February specially.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 360`. The deltaDay is calculated once day-of-month adjustments have occurred. If the schedule uses EOM convention and both dates are the last day of February, change the second day-of-month to 30. If the schedule uses EOM convention and the first date is the last day of February, change the first day-of-month to 30. If the second day-of-month is 31 and the first day-of-month is 30 or 31, change the second day-of-month to 30. If the first day-of-month is 31, change the first day-of-month to 30.

This day count has different rules depending on whether the EOM rule applies or not. The EOM rule is set in the `DayCount.ScheduleInfo`. The default value for EOM is true, as used by `DayCount.yearFraction(LocalDate, LocalDate)`.

There are two related day counts. The '30U/360 EOM' rule is identical to this rule when the EOM convention applies. The '30/360 ISDA' rule is identical to this rule when the EOM convention does not apply.

Also known as '30/360 US', '30US/360' or '30/360 SIA'.

History note. It appears that the US 30/360 day count originally started with just the two rules of '30/360 ISDA'. At some later point, the last day of February EOM rules were added.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 360`. The deltaDay is calculated once day-of-month adjustments have occurred. If both dates are the last day of February, change the second day-of-month to 30. If the first date is the last day of February, change the first day-of-month to 30. If the second day-of-month is 31 and the first day-of-month is 30 or 31, change the second day-of-month to 30. If the first day-of-month is 31, change the first day-of-month to 30.

This day count is not dependent on the EOM flag in `DayCount.ScheduleInfo`.

This is the same as '30U/360' when the EOM convention applies. This day count would typically be used to be explicit about the EOM rule applying. In most cases, '30U/360' should be used in preference to this day count.

The method `DayCount.yearFraction(LocalDate, LocalDate)` will assume that the end-of-month rule applies.

See Also:

`THIRTY_U_360`

THIRTY_360_PSA

```
public static final DayCount THIRTY_360_PSA
```

The '30/360 PSA' day count, which treats input day-of-month 31 and end of February specially.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 360`. The deltaDay is calculated once day-of-month adjustments have occurred. If the first date is the last day of February, change the first day-of-month to 30. If the second day-of-month is 31 and the first day-of-month is 30 or 31, change the second day-of-month to 30. If the first day-of-month is 31, change the first day-of-month to 30.

Also known as '30/360 PSA' (PSA is the Public Securites Association, BMA is the Bond Market Association).

THIRTY_E_360_ISDA

```
public static final DayCount THIRTY_E_360_ISDA
```

The '30E/360 ISDA' day count, which treats input day-of-month 31 and end of February specially.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 360`. The deltaDay is calculated once day-of-month adjustments have occurred. If the first day-of-month is 31, change the first day-of-month to 30. If the second day-of-month is 31, change the second day-of-month to 30. If the first date is the last day of February, change the first day-of-month to 30. If the second date is the last day of February and it is not the maturity date, change the second day-of-month to 30.

The method `DayCount.yearFraction(LocalDate, LocalDate)` will throw an exception because schedule information is required for this day count.

Also known as '30E/360 German' or 'German'. Defined by the 2006 ISDA definitions 4.16h.

The '30E/360' day count, which treats input day-of-month 31 specially.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 360`. The deltaDay is calculated once day-of-month adjustments have occurred. If the first day-of-month is 31, it is changed to 30. If the second day-of-month is 31, it is changed to 30.

Also known as '30/360 ISMA', '30/360 European', '30S/360 Special German' or 'Eurobond'. Defined by the 2006 ISDA definitions 4.16g and ICMA rule 251.1(ii) and 252.2.

THIRTY_EPLUS_360

```
public static final DayCount THIRTY_EPLUS_360
```

The '30E+/360' day count, which treats input day-of-month 31 specially.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 360`. The deltaDay and deltaMonth are calculated once adjustments have occurred. If the first day-of-month is 31, it is changed to 30. If the second day-of-month is 31, it is changed to 1 and the second month is incremented.

THIRTY_E_365

```
public static final DayCount THIRTY_E_365
```

The '30E/365' day count, which treats input day-of-month 31 and end of February specially.

The result is calculated as `(360 * deltaYear + 30 * deltaMonth + deltaDay) / 365`. The deltaDay is calculated once day-of-month adjustments have occurred. If the first day-of-month is the last day-of-month, change the first day-of-month to 30. If the second day-of-month is the last day-of-month, change the second day-of-month to 30.

Also known as '30/365 German'.

Additional documentation can be found at strata.opengamma.io.