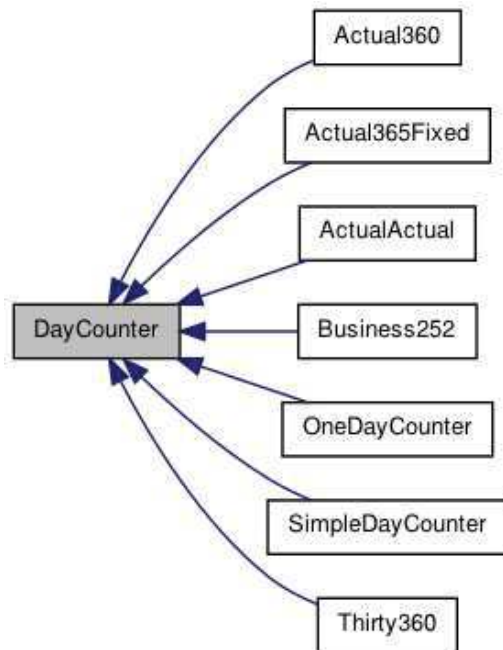


三、DayCounter 类别

◆ 在 QuantLibXL 中实作的计日类别有七大类，如图所示。



Andy M. Dong 82

► DayCounter 类别

- CustomRegion
- CYPCurrency
- CzechRepublic
- CzechRepublic.Market
- CZKCurrency
- Date
- DatedOISRateHelper
- DateGeneration
- DateGeneration.Rule
- DateParser
- DateVector
- DateVector.DateVectorEnumerator
- DayCounter
- DefaultDensity
- DefaultDensityCurve
- DefaultLogCubic
- DefaultProbabilityHelper
- DefaultProbabilityHelperVector
- DefaultProbabilityHelperVector.DefaultProbability
- DefaultProbabilityTermStructure
- DefaultProbabilityTermStructureHandle
- _DeltaVolQuote
- _DeltaVolQuote.AtmType
- _DeltaVolQuote.DeltaType
- DeltaVolQuote
- DeltaVolQuoteHandle
- DEM.Currency
- Denmark
- DepositRateHelper
- DifferentialEvolution
- DirichletBC
- Discount
- DiscountCurve
- DiscountingBondEngine
- DiscountingSwapEngine
- DiscreteAveragingAsianOption
- Dividend
- DividendSchedule
- DividendSchedule.DividendScheduleEnumerator
- DividendVanillaOption
- DKKCurrency

```
~DayCounter()
dayCount(QuantLib.Date, QuantLib.Date)
DayCounter(System.IntPtr, bool)
Dispose()
getCPtr(QuantLib.DayCounter)
name()
_str_0
yearFraction(QuantLib.Date, QuantLib.Date)
yearFraction(QuantLib.Date, QuantLib.Date, QuantLib.Date)
yearFraction(QuantLib.Date, QuantLib.Date, QuantLib.Date, QuantLib.Date)
swigCMemOwn
swigCPtr

public double yearFraction(QuantLib.Date d1, QuantLib.Date d2)
QuantLib.DayCounter 的成员
```

Andy M. Dong 83

◆ 在台湾市场上，主要是以 Actual365Fixed 为主，

- 此计日类别是假设一年都是 365 天(不管闰年与否)，两个日期(Date1≤Date2)间的日时距则为两日期的实际日历天差距。
- Actual365Fixed 的年时距(yearfraction)可表示为，

$$\text{年时距} = \frac{\text{Date2} - \text{Date1}}{365}$$

◆ 另一个常用的计日类别 Actual360，

- 此计日类别是假设一年都是 360 天(不管闰年与否)，两个日期间的日时距则为两日期的实际日历天差距。Actual360 的年时距可表示为，

$$\text{年时距} = \frac{\text{Date2} - \text{Date1}}{360}$$

◆ 在内地市场，Actual360、Actual365Fixed 与 ActualActual 都有在用。

- Swap Floating Side 之 Shibor Index 用 Actual360、Swap Fixed Side 用 Actual365Fixed。
- 债券会用 ActualActual。

Andy M. Dong 84



```
actual360.hpp - x
QuantLib
QuantLib:Actual360

28 namespace QuantLib {
29     //! Actual/360 day count convention
30     //! Actual/360 day count convention, also known as "Act/360", or "A/360".
31     \ingroup daycounters
32     */
33     class Actual360 : public DayCounter {
34     private:
35         class Impl : public DayCounter::Impl {
36         private:
37             bool includeLastDay_;
38         public:
39             explicit Impl(const bool includeLastDay)
40                 : includeLastDay_(includeLastDay) {}
41             std::string name() const {
42                 return includeLastDay_ ?
43                     std::string("Actual/360 (inc)")
44                     : std::string("Actual/360");
45             }
46             Date::serial_type dayCount(const Date& d1,
47                                       const Date& d2) const {
48                 return (d2-d1) + (includeLastDay_ ? 1 : 0);
49             }
50             Time yearFraction(const Date& d1,
51                              const Date& d2,
52                              const Date&,
53                              const Date&) const {
54                 return (daysBetween(d1,d2)
55                     + (includeLastDay_ ? 1.0 : 0.0))/360.0;
56             }
57         };
58     public:
59         explicit Actual360(const bool includeLastDay = false)
60             : DayCounter(ext::shared_ptr<DayCounter::Impl>{
61                 new Actual360::Impl(includeLastDay)}) {}
62     };
63 }
```

Andy M. Dong 85

◆ 下表列示 QuantLib 有实作的计日类别，及其代表的字符串文字。

- 例如 A/360 代表 Actual360 类别，A/365F 代表 Actual365Fixed 类别。
- 至于每一种类别的计算逻辑，请自行参考 QuantLib 的文件，在此不一一说明。

```
class Actual360 : public DayCounter{
public:
    Actual360(const bool includeLastDay = false);
};

class Actual365Fixed : public DayCounter{
public:
    enum Convention { Standard, Canadian, NoLeap };
    Actual365Fixed(Convention c = Standard);
};

class Thirty360 : public DayCounter{
public:
    enum Convention { USA, BondBasis, European, EurobondBasis, Italian };
```

Andy M. Dong 86

```
    Thirty360(Convention c = USA);
};

class ActualActual : public DayCounter{
public:
    enum Convention { ISMA, Bond, ISDA, Historical, Actual365, AFB, Euro };
    ActualActual(Convention c = ISDA, const Schedule& schedule = Schedule());
};

class OneDayCounter : public DayCounter {};

class SimpleDayCounter : public DayCounter {};

class Business252 : public DayCounter{
public:
    Business252(Calendar c = Brazil());
};
```

Andy M. Dong 87