

(三)Index 物件重要方法

◆ 建構在 Index 物件上的方法

● Function : name():

✧ 語法

```
def name(self):  
    return _QuantLib.Index_name(self)
```

✧ 使用

● Function : fixingCalendar():

✧ 語法

```
def fixingCalendar(self):  
    return _QuantLib.Index_fixingCalendar(self)
```

✧ 使用

◆ 比價利率的增刪

● Function : fixing():

✧ 語法

```
def fixing(self, fixingDate, forecastTodaysFixing=False):  
    return _QuantLib.Index_fixing(self, fixingDate, forecastTodaysFixing)
```

✧ 使用

- ✓ 根據輸入的比價日向量，傳回相對應的比價向量，如果事先沒有輸入該日的資料，則以 Index 物件中的利率曲線物件來估計。
- ✓ 第三個參數是控制如果沒有今日的比價日資訊，是否要以 Index 物件中的利率曲線物件來估計，預設值為 False。
- ✓ 下圖中，我們要求 2009/8/13 的比價利率，傳回 4.00%。這是我們之前已經加入的比價日資訊。當我們要求 2011/8/13 的比價利率，則傳回 4.9091%。這是根據 Index 物件中的利率曲線物件來估計。
- ✓ 雖然我們利率曲線是以 5% Flat Forward 建立的物件，但是由於 Index 的複利方式與利率曲線的複利方式不同，因此有一些差距。

	J	K	L	M	N	O	P
25							
26		IndexFixing	4.0000%				
27	string	ObjectID	TWDM3M#0004				
28	vec<long>	FixingDates	2009/8/13				
29	bool	ForecastToday	FALSE				
30	any	Trigger					
31							
32		IndexFixing	4.9091%				
33	string	ObjectID	TWDM3M#0004				
34	vec<long>	FixingDates	2011/8/13				
35	bool	ForecastToday	FALSE				
36	any	Trigger					
37							
38							
39							
40							
41							
42							
43							
44							
45							

● Function : isValidFixingDate():

✧ 語法

```
def isValidFixingDate(self, fixingDate):
    return _QuantLib.Index_isValidFixingDate(self, fixingDate)
```

✧ 使用

- ✓ 根據輸入的比價日向量，傳回比價日向量是否為有效日期的向量。

● Function : clearFixings():

✧ 語法

```
def clearFixings(self):
    return _QuantLib.Index_clearFixings(self)
```

✧ 使用

- ✓ 輸入的 Index 物件的所有比價資料清除。

- Function : addFixing():

- ◇ 語法

```
def addFixing(self, fixingDate, fixing, forceOverwrite=False):
    return _QuantLib.Index_addFixing(self, fixingDate, fixing, forceOverwrite)
```

- ◇ 使用

- ✓ 輸入的比價日向量以及比價利率大小，加到輸入的 Index 物件之內。第四個參數控制是否要把已有日期的比價利率以新的資料覆蓋上去。

- Function : addFixings():

- ◇ 語法

```
def addFixings(self, fixingDates, fixings, forceOverwrite=False):
    return _QuantLib.Index_addFixings(self, fixingDates, fixings, forceOverwrite)
```

- ◇ 使用

- ✓ 輸入的比價日向量以及比價利率大小，加到輸入的 Index 物件之內。第四個參數控制是否要把已有日期的比價利率以新的資料覆蓋上去。
 - ✓ 在此例中，我們加入一些已經比價的資料，2009/8/13、2009/11/13、2010/2/13、2010/5/13、...、2011/2/13，他們的比價利率皆設為 4.00%。注意，由於我們評價日為 2011/7/11，只有此日前的日期有比價資料。

Andy M. Dong 27

	J	K	L	M	N	O	P
12							
13		IndexAddFixings	=qlIndexAddFixing		TSFixings#0010	Fixing Dates	Fixing Rates
14	string	Objectid	TWDMM3M#0002			2009/8/13	4.00%
15	vec<long>	FixingDates	N14:N20			2009/11/13	4.00%
16	vec<double>	FixingValues	O14:O20			2010/2/13	4.00%
17	bool	ForceOverwrite	TRUE			2010/5/13	4.00%
18	any	Trigger				2010/8/13	4.00%
19						2010/11/13	4.00%
20		IndexAddFixings2	TRUE			2011/2/13	4.00%
21	string	Objectid	TWDMM3M#0002				
22	string	TimeSeries	TSFixings#0010				
23	bool	ForceOverwrite	TRUE		TRUE		
24	any	Trigger					
25							
26		IndexFixing	0.04				
27	string	Objectid	TWDMM3M#0002				
28	vec<long>	FixingDates	2009/8/13				
29	bool	ForecastToday	FALSE				
30	any	Trigger					
31							
32		IndexFixing	0.049091403				
33	string	Objectid	TWDMM3M#0002				

Andy M. Dong 28

- Function : timeSeries():

- ✧ 語法

```
def timeSeries(self):  
    return _QuantLib.Index_timeSeries(self)
```

- ✧ 使用

- ✓ 傳回比價日與比價利率的時間數列。