Temporal Expressions

Temporal Expressions

The Apache Open For Business recurring events are based on *Temporal Expressions* - a design proposed by Martin Fowler: Recurring Events for Calendars (pdf). The design is extremely flexible and it allows for arbitrarily complex recurring events.

Overview

Each temporal expression could be considered a rule - like "every Monday" or "the 15th of the month." Temporal expressions can used alone or they can be combined in any of three *collection expressions* - Union, Intersection, and Difference. A date will match a Union collection if any of its member expressions match (logical OR). A date will match an Intersection collection if all of its member expressions match (logical AND). A date will match a Difference collection if it matches the *included* member expression and doesn't match the *excluded* member expression (logical AND NOT).

Some of the basic OFBiz temporal expressions are ranges - which simplifies expression creation. For example, if an event occurs every Monday, Tuesday, and Wednesday, you *could* create an expression for each day and make them members of a Union expression, *or* you could use a single Day Of Week Range expression that includes Monday through Wednesday.

Sometimes recurring events need to be rescheduled - like when they fall on a holiday, or when maintenance is being performed. The Substitution expression can be used to reschedule recurring events.

In the temporal expression definitions that follow, the expressions will use this notation:

```
Expression(parameter1, parameter2, ...)

Collection:
  Member Expression(parameter1, parameter2, ...)
  Member Expression(parameter1, parameter2, ...)
```

Basic Expressions

Range Expressions

```
MinuteRange(start, end)
HourRange(start, end)
DayOfWeekRange(start, end)
DayOfMonthRange(start, end)
MonthRange(start, end)
DateRange(start, end)
```

Range Examples

An event that occurs during the 3 o'clock PM hour:

```
HourRange(15, 15)
```

An event that occurs on Monday, Tuesday, and Wednesday:

```
DayOfWeekRange(Monday, Wednesday)
```

Day In Month Expression

```
DayInMonth(day of week, occurance)
```

Day In Month Examples

An event that occurs on the first Monday of the month:

```
DayInMonth(Monday, 1)
```

An event that occurs on the fourth Thursday of the month:

```
DayInMonth(Thursday, 4)
```

An event that occurs on the last Saturday of the month:

```
DayInMonth(Saturday, -1)
```

Frequency Expression

```
Frequency(start date-time, frequency type, frequency count)
```

Frequency Examples

An event that occurs every day beginning January 1, 2010:

```
Frequency(2010-01-01, day, 1)
```

An event that occurs every two weeks beginning January 1, 2010:

```
Frequency(2010-01-01, day, 14)
```

Expression Collections

Basic temporal expressions by themselves are not very powerful. Truly powerful and complex recurring events can be created by combining basic expressions in expression collections.

Union Expression

```
Union:
  Expression(parameter1, parameter2, ...)
  Expression(parameter1, parameter2, ...)
  ...
```

Intersection Expression

```
Intersection:
  Expression(parameter1, parameter2, ...)
  Expression(parameter1, parameter2, ...)
  ...
```

Difference Expression

```
Difference:
  Include:
    Expression(parameter1, parameter2, ...)
  Exclude:
    Expression(parameter1, parameter2, ...)
```

Expression Collection Examples

An event that occurs at 8:00 AM:

```
Intersection:
  MinuteRange(0, 0)
  HourRange(8, 8)
```

An event that occurs Monday, Tuesday, Wednesday, and Saturday:

```
Union:
  DayOfWeekRange(Monday, Wednesday)
  DayOfWeekRange(Saturday, Saturday)
```

An event that occurs Monday, Tuesday, Wednesday, and Saturday at 8:00 AM:

```
Intersection:
  MinuteRange(0, 0)
  HourRange(8, 8)
  Union:
    DayOfWeekRange(Monday, Wednesday)
    DayOfWeekRange(Saturday, Saturday)
```

An event that occurs Monday, Tuesday, Wednesday, and Saturday at 8:00 AM *except* the last Saturday of the month:

```
Difference:
  Include:
    Intersection:
      MinuteRange(0, 0)
      HourRange(8, 8)
      Union:
        DayOfWeekRange(Monday, Wednesday)
        DayOfWeekRange(Saturday, Saturday)
  Exclude:
    DayInMonth(Saturday, -1)
```

The Substitution Expression

The Substitution temporal expression works a lot like a Difference expression, except it provides a substitute for the exclusion.

> The Substitution expression is not available in Release 9.04.

```
Substitution:
  Include:
    Expression(parameter1, parameter2, ...)
  Exclude:
    Expression(parameter1, parameter2, ...)
  Substitute:
    Expression(parameter1, parameter2, ...)
```

Substitution Examples

An event that occurs on the first Monday of the month, *except* on Labor Day (first Monday in September) - where it is rescheduled for the following Tuesday:

```
Substitution:
  Include:
    DayInMonth(Monday, 1)
  Exclude:
    Intersection:
      DayInMonth(Monday, 1)
      MonthRange(9, 9)
  Substitute:
    DayOfWeekRange(Tuesday, Tuesday)
```

An event that occurs Monday, Tuesday, Wednesday, and Saturday at 8:00 AM *except* the last Saturday of the month - where it is rescheduled for the following Sunday at 1:00 PM:

```
Substitution:
  Include:
    Intersection:
      MinuteRange(0, 0)
      HourRange(8, 8)
      Union:
        DayOfWeekRange(Monday, Wednesday)
        DayOfWeekRange(Saturday, Saturday)
  Exclude:
    DayInMonth(Saturday, -1)
  Substitute:
    Intersection:
      MinuteRange(0, 0)
      HourRange(13, 13)
      DayOfWeekRange(Sunday, Sunday)
```

What's Next?

If you would like to take the Temporal Expression tutorial, go here.

If you would like to see the Temporal Expression demo data, go here.

If you would like to see the Temporal Expression Java API, go here.