

Calendar Extract

This extract produces a tree that can consist of years, half years, quarters, months, weeks, days, hours, minutes, and/or seconds. Connection to a source system is not required, so time dimensions can be easily created in Jedox Integrator and imported later into a Jedox database.

The language of the output can be set according to standard language codes, which can be found at

http://www.loc.gov/standards/iso639-2/php/English_list.php.

Date hierarchy levels

Level	The supported hierarchy levels are root (for a single constant root node), years, half years, quarters, months, weeks, days, hours, minutes, and seconds.
Start	Start of the levels (required for years)
End	End of the levels (required for years)
Pattern	Pattern for the representation as a text

Below is an overview of the most important patterns:

Pattern	Meaning	Result for 31.12.2014
d	Day in month	31
D	Day in year	365
E	Day in week	EEE: Wed, EEEE: Wednesday
M	Month in year	MM: 12, MMM: Dec, MMMM: December
q	Half year	2
Q	Quarter in year	4

v	Week in year (corresponding to y)	53
w	Week in year (corresponding to Y)	1
W	Week in month	5
y	Year	yyyy: 2014, yy: 14
Y	Week year	2015

The pattern symbols q, Q, and v are Jedox Integrator-specific enhancements of the standard SimpleDateFormat, which can be found under

<http://download.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>.

Fixed text elements in calendar

When the output should contain fixed text elements, then these can be defined with a single quote ('text'). For the root level, the output can only consist of fixed text elements.

+ **Example**

If you want to create a year dimension with half years, quarters, and months, then the setting looks like this:

Date hierarchy level	Start	End	Pattern
root			'Year'
months			MMM
quarters			'Qtr. 'Q
halfyears			'Half year 'q

Result:

level1	level2	level3	level4
Year	Half year 1	Qtr. 1	Jan
Year	Half year 1	Qtr. 1	Feb
Year	Half year 1	Qtr. 1	Mar
Year	Half year 1	Qtr. 2	Apr
Year	Half year 1	Qtr. 2	May
Year	Half year 1	Qtr. 2	Jun
Year	Half year 2	Qtr. 3	Jul
Year	Half year 2	Qtr. 3	Aug
Year	Half year 2	Qtr. 3	Sep
Year	Half year 2	Qtr. 4	Oct
Year	Half year 2	Qtr. 4	Nov
Year	Half year 2	Qtr. 4	Dec

Week-specific settings

First day of the week	The first day of the week. The numerical values of option in older versions are replaced with text representations. If not set, the language-specific value is used, e.g. "Sunday" in US and "Lundi" in France.
------------------------------	---

Minimum of days in first week The minimum of days required in the first week of the year. For example, if the first week is defined as one that contains the first day of the first month of a year, it is 1. If it must be a full week, it is 7. If not set, the language-specific value is used.

Weeks can be generated in two different ways:

- With pattern letters “v” and “y”: in this case, the weeks are ascending in one calendar year, starting from week 0 or 1. All days under the week belong to the same calendar year.



no overlapping of weeks & days with Pattern "vv yyyy"

Click to enlarge screenshot

- With pattern letters “w” and “Y”: in this case, the weeks may start with 53 or 54 and end with 1 in one calendar year. Some of the days under the week may belong to a different calendar year.



overlapping weeks & days with Pattern "ww YYYY"

Click to enlarge screenshot

Note: you should not use the combinations v/Y and w/y.

+ Example with week level

Pattern	'CW' w YYYY	'CW' v yyyy
01.01.2010	CW 53 2009	CW 00 2010
31.12.2014	CW 01 2015	CW 53 2014

Time-to-date hierarchy

The Calendar extract allows the generation of hierarchical nodes for a

time-to-date (TTD) analysis, which is a general term for year-to-date, month-to-date, etc.

The levels for which additional TTD nodes should be generated have to be defined in the time-to-date levels table in Integrator. They must have a different pattern than the “normal” date hierarchy levels so that the nodes can be distinguished. In most cases, one TTD level should be sufficient.

There are a number of generation nodes available:

timeToDate: sums up all nodes starting from 1 up to the current node. This is the most-used version for a year-to-date hierarchy.

timeFromDate: sums up all nodes starting from the current node up to the last node. This can be used in year-to-go scenarios.

timeToDateDeep: similar to timeToDate node, but uses a recursive definition of the consolidations, which leads to a lower number of total nodes but a deeper hierarchy. This generation pattern was used in ETL up to version 5.1 for TTD hierarchies.

timeFromDateDeep: similar to timeFromDate node, but uses a recursive definition of the consolidations, which leadsto a lower number of total nodes but a deeper hierarchy.

timeToDateCompact: generates two separate hierarchies: one for standard nodes and one for time-to-date nodes in a compact

form. The TTD nodes are created under the specified root element and sum up all nodes starting from 1 up to the current node.

timeFromDateCompact: generates two separate hierarchies: one for standard nodes and one for time-to-date nodes in a compact form. The TTD nodes are created under the specified root element and sum up all nodes starting from the current node up to the last node.

storedTTDValues: used when the base node itself contains aggregated time-to-date values. The generated consolidated nodes then represent aggregated values by subtracting the previous node from the current node with a consolidation factor of -1.

rolling: generates a separate hierarchy for rolling periods under the specified root element. The global parameter “Rolling periods” defines the number of preceding month nodes, e.g. 12 for a 12-month rolling hierarchy.

Reference level: each TTD hierarchy has its own reference level that defines the start and end points for the aggregation of a TTD node. For example, for a year-to-date hierarchy on several years, the reference level should be “year”. For the reference level “root”, it is possible to create the TTD nodes in a completely separated hierarchy by specifying the name of its root node.

Note that in ETL up to version 5.1, the time-to-date reference “toNext”

specified the next higher date hierarchy level and the time-to-date reference "toRoot" specified the reference level "root".

+Example of different modes

Level: months

ReferenceLevel: years

Pattern: MM.yyyy 'TTD'

Child nodes from node 04.2015 TTD

timeToDate	timeFromDate	timeToDateDeep	timeFromDateDeep	storedTTDValues
01.2015	04.2015	04.2015	04.2015	04.2015
02.2015	05.2015	03.2015 TTD	05.2015 TTD	03.2015 (*)
03.2015	06.2015			
04.2015	07.2015			
	08.2015			
	09.2015			
	10.2015			
	11.2015			
	12.2015			

(*) consolidation factor -1

timeToDate timeFromDate

+Example for YTD hierarchy on months for the first half of a year

Date hierarchy levels

Level	Start	End	Pattern
root			'All Months'
quarters	1	2	'Qtr. 'Q
months			MMM

Time-to-date levels

Level	Pattern	Mode	Reference level	Root node
months	MMM' acc'	timeToDateDeep	root	All YTD

This results in a year-to-date hierarchy for months, with additional nodes for quarters.

level1	level2	level3	level4	level5	level6	level7	level8
JAll YTD	Jun acc	May acc	Apr acc	Mar acc	Feb acc	Jan acc	Jan
All YTD	Jun acc	May acc	Apr acc	Mar acc	Feb acc	Feb	
All YTD	Jun acc	May acc	Apr acc	Mar acc	Mar		
All YTD	Jun acc	May acc	Apr acc	Apr			
All YTD	Jun acc	May acc	May				
All YTD	Jun acc	Jun					
All Months	Qtr. 1	Jan					
All Months	Qtr. 1	Feb					
All Months	Qtr. 1	Mar					
All Months	Qtr. 2	Apr					
All Months	Qtr. 2	May					
All Months	Qtr. 2	Jun					

Months offset

For the calendar, a month offset can be specified, e.g. to handle fiscal

years that differ from a calendar year (non-calendar fiscal year). In this case, the additional pattern symbol “O” has to be set as prefix to the symbols M, Q, q, and y. The month offset is not available in combination with the weeks level. It shouldn’t be used in combination with days.

Instead of this global calendar setting, a month offset can be set also locally inside of the pattern with syntax “O(<offset)”. This allows the use of different offsets in one calendar, e.g. for previous and next month.

If the resulting calendar entity is outside of the time range of the calendar (as defined by the start and end parameters of the years level), the result of the pattern is empty; i.e., the node or attribute value is not created. For locally defined patterns, this logic can be turned off with the syntax “O(<offset>,false)”.

+ Examples

(1) Global months offset, starting year 2015:

Pattern	Month offset 2	Month offset -3
MMM yyyy	Jan 2015	Jan 2015
OMMM Oyyyy	Mar 2015	Oct 2014
M.yy 'Q'Q ('OM Qyy 'Q'OQ')	1.15 Q1 (3.15 Q1)	1.15 Q1 (10.14 Q4)

(2) Local months offset, start year 2015, end year 2016:

Pattern MMMM yyyy	Pattern	Result
-------------------	---------	--------

January 2015	OMMM(1) Oyyyy(1)	Feb 2015
January 2015	OMMM(-12) Oyyyy(-12)	
January 2015	OMMM(-12,false) Oyyyy(-12,false)	Jan 2014
December 2016	OMMM(1) Oyyyy(1)	
December 2016	OMMM(-12) Oyyyy(-12)	Dec 2015
December 2016	OMMM(1,false) Oyyyy(1,false)	Jan 2017

Rolling periods

This setting is required for time-to-date hierarchies of type “Rolling”.

Special elements

To add an element for an opening period or to add month elements for fiscal years with more than 12 periods, you can define special elements for each level. A pattern has to be defined for each individual element, separated by a semicolon (;). With the prefix “^” the special element is generated in front of all other elements; otherwise it is generated after all other elements.

+ **Examples**

1. Level years, pattern: ‘~’
One additional element “~” is generated after all other years
2. Level months, pattern: ^yyyy ‘~’; yyyy 13; yyyy 14; yyyy

15; yyyy 16

With current year 2015, one additional element “2015 ~” is generated before the first month element (e.g. Jan 2015). After the last month element (e.g. Dec 2015), the elements “2015 13”, “2015 14”, “2015 15”, and “2015 16” are generated.

Attributes

For each level, the following attributes can be defined:

Attribute: name of the attribute

Pattern: the generation pattern

Language

Time-to-date pattern: optional attributes for the nodes generated for time-to-date hierarchies

Special element pattern: optional attributes for special elements