

Cube Load

Related link: [Cube Load with Drillthrough](#)

The cube load writes data in a cube in the Jedox In-Memory DB (OLAP). If the cube is not yet present in the database, it will be created during the load. In this way, the modeling can be carried out within Jedox Integrator. A complete database can even be created from scratch.

Main Settings

Data Source: The source is an extract or transform that defines in its rows the data to be written to the cube. The first columns of the source must contain the names of the dimension elements of the cube (except in delete mode; see mode explanations below). The last column contains the value that is written in the cube cell.

In the following example, the cube has eight dimensions, one less than the number of columns in the source:

Orderlines	Years	Months	Products	Customers	Channels	Datatypes	Measures	#Value
202921	2008	Jan	Cross-650 White 58	Downtown Hotel	Online Shop	Actual	Units	1.0
202921	2008	Jan	Cross-650 White 58	Downtown Hotel	Online Shop	Actual	Cost of Sales	146.01198
202921	2008	Jan	Cross-650 White 58	Downtown Hotel	Online Shop	Actual	Sales	243.0399604

Target Connection: Connection to a Jedox In-Memory DB

Target Cube: Cube in the Jedox In-Memory DB

Mode: The load mode selected depends on the existing structures and data in the target system. Cube load modes are:

- create** The existing cube is deleted and created anew with the dimension order as defined in the source. The values from the data source are aggregated and written in the cube. Existing rules are deleted.
- update** The existing cube is emptied; values from the data source are aggregated and written to the cube.
- add** Values from the data source are aggregated and added to the existing values in the cube.
- insert** Existing values in the cube are overwritten with the values from the data source. The values from the data source are not aggregated in the load; they must be aggregated in the extract/transformation, otherwise the last cell value is written.
- delete** Values from the data source are deleted in the cube; other values that exist in the cube remain. For values in consolidated cells, the respective cube slice is deleted. Dimensions can be omitted in the source; in this case, the complete cube slice according to this dimension is deleted. The value column must be present as the last column, and the value itself is ignored for deletion (can be 0). See also [Cube Slice Extract](#).

Splash Mode: Can be set to write on consolidated elements. Possible values are:

- default** If value = 0.0: clears all base paths.
If value <> 0.0 and old_value = 0.0: computes the splash value according to the weights of the path and sets this value to all base paths.
If value <> 0.0 and old_value <> 0.0: scales all value so that the sum is the new value
This mode corresponds to [splash parameter #](#).
- set** Sets all base paths to the given value. This mode corresponds to [splash parameter !](#).
- add** Adds given value to all base paths. This mode corresponds to [splash parameter !!](#).
- disabled** No splashing is done on the In-Memory DB. This will result in a warning message for consolidated elements.

The Splash mode is set to "disabled" unless otherwise indicated. Splash modes "add" and "set" are only possible in combination with load modes "insert" and "delete". Further information on the topic of splashing can be found in the article [Splashing: Data Entry on Consolidated Cells](#)

Handling of missing elements:	In general, if you write data to a cube cell, the dimension elements of the cell path must exist for all dimensions. You can configure how the cube load should react if this is not the case for one (or several) dimensions:
warning	A warning message is written in the log file indicating which dimension element is missing. All values for the cube cells with missing elements are not written to the cube. Note: If a warning on missing elements occurs when drillthrough has been enabled, the drillthrough data on this element is nevertheless written in persisted drillthrough. It will be considered in a drillthrough request on the root element of the dimension.
mapToDefault	A default element has to be defined in the load which is used for all dimensions. The load then writes the value on this default element instead of the missing element.
mapToDefaultWrite	The Default Write Element has to be defined in the dimension properties. The load then writes the value to this Default Write Element instead of the missing element.
createUnderDefault	A default element that is used for all dimensions has to be defined in the load. The load then creates the missing element in the dimension, consolidates it under the default element, and writes the value on the newly created element.
createUnderDefaultParent	The Default Parent Element has to be defined in the dimension properties. The load then creates the missing element in the dimension, consolidates it under this Default Write Element element, and writes the value on the newly created element.

With options `mapToDefault` and `createUnderDefault` you will not lose the values of items which do not exist in the dimension. The default element is the same for all dimensions and has to exist before the load for all dimensions for which this logic should apply.

Dimension Mapping

In the Dimension Mapping table, any dimension of the cube can be assigned for each source column. For columns that should not be loaded to the cube, the entry `""` is used. Thus it is possible to rename and skip columns. The dimension order cannot be changed, which is only relevant if the cube does not yet exist. The last source column is not displayed in the Dimension Mapping, as it always contains the cube cell value.

By default, if no Dimension Mapping is given, the names of the cube dimensions are entirely defined by the names of the source column names. In this case, the number of dimensions must be equal to the number of columns in the source minus 1.

Advanced Settings

Cube Layout Change Mode:	This option defines the behavior if the source columns and the cube layout don't match for load modes "add", "insert", and "update". It allows writing to a default element for missing source columns and the automatic change of the cube layout before the data is loaded.	
none	columns of source and dimensions of cube have to match exactly. (default)	
noneWithDefaultWrite	If dimensions of the cube are not in the source, the values are written to the Default Write Element of the dimension. The cube layout is not changed.	
addDimensions	If columns in the source are not in the cube, they are added as cube dimensions at the end. If dimensions of the cube are not in the source, the values are written to the Default Write Element of the dimension.	
updateDimensions	the cube layout is modified exactly to the columns of the source. Note that dimensions may be removed from the cube, in which case the Default Read Element of the dimension is used and no data is lost.	
reorderDimensions	the columns of the source and the dimensions of the cube have to match, but not the order. If the order does not match, the ordering of the cube is changed.	
Deactivate while Loading:		
Cube access	This option can only be used in load mode "insert". If activated, then the data cube is locked for read and write access for all other In-Memory DB users. With this option you can prevent seeing inconsistent cube values during the cube load.	
Rules calculation	The calculation of rules in the In-Memory DB can be deactivated for this cube during the load. It may have a positive influence on rules with markers only when loading to consolidated cells.	
SVS event	The Supervision Server (SVS) event <code>initCubeWorker</code> , which is executed at changes of cell values in a cube, can be deactivated for the cube during the load. See article Supervision Server Handlers for detailed information on this event. It is disabled by default in cube loads.	
Bulk Size:	Size of blocks for OLAP write requests:	
	<ul style="list-style-type: none"> • If this field is filled with a value, this value will be used. • If this field is empty, the value of the key "integrator.loads.cube_bulk_size" in "Administration - Settings" will be used when the key value is greater than 100,000. • If this field is empty and the value of the key "integrator.loads.cube_bulk_size" in "Administration - Settings" is smaller than 100,000 or that key doesn't exist, the default value of 100,000 will be used. 	
Settings for Drillthrough	Default setting is None. For details on drillthrough options, see Cube Load with Drillthrough .	
Settings for semi-additive measures	For the use of semi-additive measures (especially time dimensions), you can define the dimensions for measure declaration and for semi-additive aggregations here.	