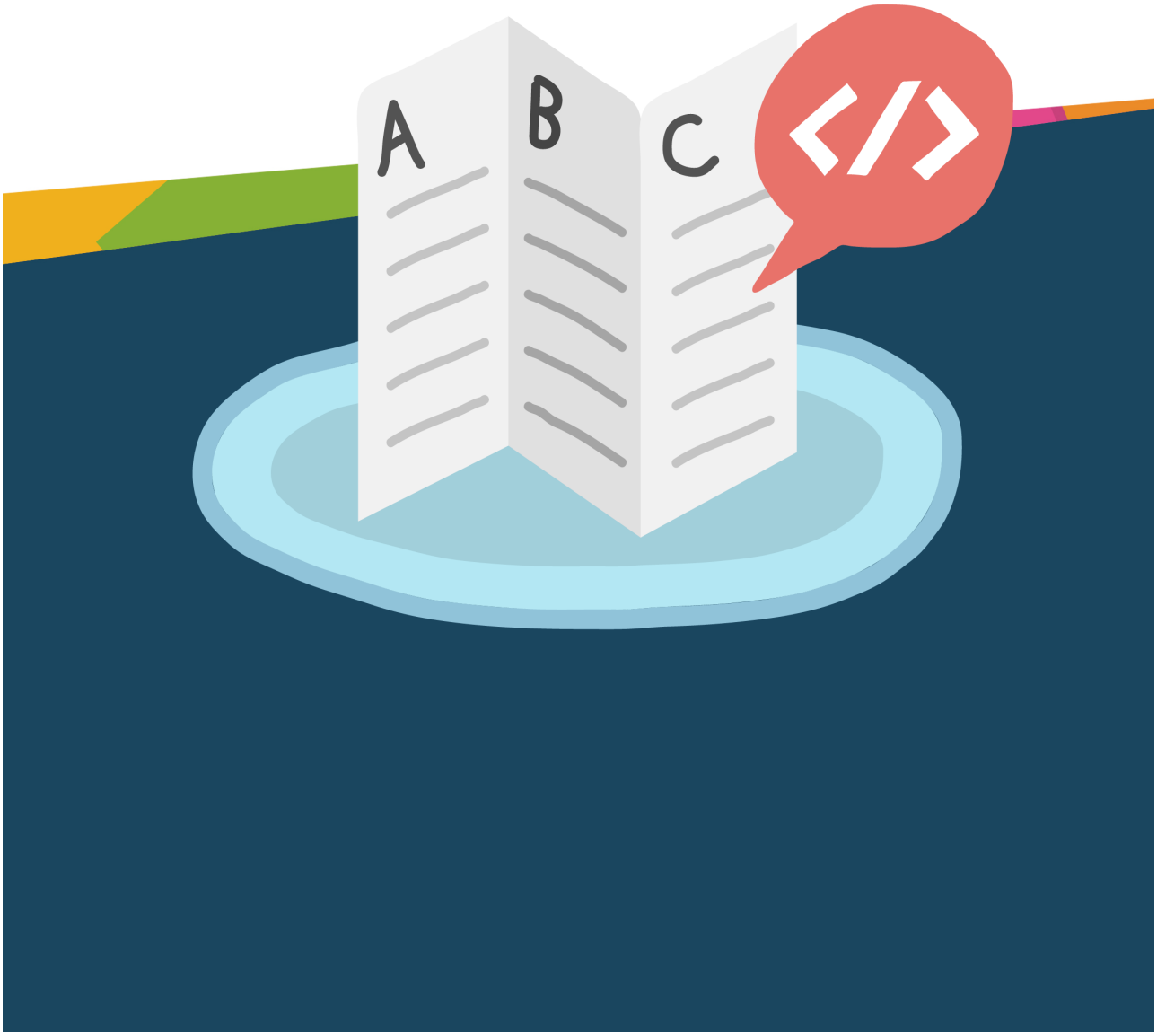
iObeya

Import/Export parameters

Version 4.21



iObeya

Export/Import parameters

Introduction

This section describes all the parameters of the elements that can be found in the export files.

> This section of the documentation is for advanced usage. Please ensure you have a complete understanding of this documentation and perform tests on a test platform before attempting to modify boards that are in production.

> Furthermore, please note that some data is critical. Each modification can lead to unexpected results using an imported element (e.g: if the width and height of a note is changed to a non supported size format, the rendering of the note will be broken).

> For advanced usage, the use of the *REST API* is advised.

There are three policies for element properties:

- **Mandatory**: data is mandatory. Import will fail if data is missing, or invalid.
- **Optional**: data can be missing, or invalid. It will be replaced by some default on import.
- **Information**: this is just information for the user, and will not be taken into account during the import operation.

A - Fields common to all board elements

These fields are common to all kinds of board elements.

Column	Description	Mandatory?
class	Defines element type, or kind of element. Every type of element has its own class name that defines it.	Mandatory
entityType	Type of element. If custom plugins like custom notes exist, this parameter specifies the custom note type.	Optional
id	ID of the element. This is a GUID (Globally Unique IDentifier) that identifies the instance of an element.	Optional
container	Name of the board	Optional if using import to a board. Mandatory if using import to a room.
isAnchored	Boolean that indicates if the element is locked	Optional
deleted	Boolean that indicates if the element must be deleted	Optional
zOrder	Position in the Z axis	Optional
linkLabel	Hyperlink name	Optional
linkURL	Hyperlink URL	Optional
height	Height of the element	Optional
width	Width of the element	Optional
x	Horizontal position of the element from the top left hand corner	Optional
y	Vertical position of the element from the top left hand corner	Optional

class

If this information is wrong or blank, import will lead to an error. The table below indicates the full name of each class of elements, and the default entity type when the class is extendable:

Element usual name	Full class name	Default entity type (some other entity types)
Note	**com.iobeya.entity.Note**	**BoardNote** (BoardVSMNote, BoardShape)
Card	**com.iobeya.entity.Card**	**BoardCard**
Sticker	**com.iobeya.entity.Sticker**	
Roll	**com.iobeya.entity.Roll**	
Freetext	**com.iobeya.entity.Freetext**	

Element usual name	Full class name	Default entity type (some other entity types)
Label	**com.iobeya.entity.Label**	
Image or Image gallery	**com.iobeya.entity.BoardImage**	
Graph	**com.iobeya.entity.Chart**	
Gauge	**com.iobeya.entity.Gauge**	
Team	**com.iobeya.entity.Team**	

entityType

Entity type is used to differentiate between board element variants of the same class. Elements of the same class, but with different entity types will generally have similar behavior, but it usually will not be possible to change one into another, or to put them together into the same tool. As an example, the **com.iobeya.entity.Note** class contains the note (entity type **BoardNote**), the VSM note (entity type **BoardVSMNote**) and the Shape (entity type **BoardShape**). Conversely, basic shapes, milestones and the VSM shapes all have the **BoardShape** entity type.

If blank, the default type of element will be set on import. It is only relevant for extendable classes of elements which -- as of version 3.6 -- only include note and cards.

id

- **if there is an element with the same ID on the board, it will update the element**
- if the field is blank a new ID will be created.
- if the ID already exists on another board, then a new ID will be generated (as an element cannot be on two different boards).

container

- **if using import at room level this field is mandatory in order to update or create the element on the right board**.
- if using import to a board this field is not used.

isAnchored

- if ***isAnchored=1***, then the element will be locked on the board.
- it will be ignored for elements that cannot be locked.

deleted

if ***deleted=1***, the element will be sent to the trash after being reimported.

zOrder

higher numbers are at the top of the pile, and lower numbers at the bottom.

linkLabel and linkURL

- if blank, no hyperlink on the element.
- linkUrl must respect URL standard scheme (http://..., https://...).
- it will be ignored for stickers (that cannot have hyperlinks).

width and height

for notes, stickers and cards, there are three fixed sizes. If these are not respected, then the default value of width/height will be the largest size.

x and y

- if x and y coordinates match a board position, the element will be positioned at these coordinates on the board.
- if x or y are empty or match a position outside the board, the element will be placed to the left or top (respectively) corner of the board.

B - Information fields

When exported, these values only provide information to the user. They will not be used during import operations.

Column	Description
creator	Creator of the element
creationDate	Creation date of the element
modifier	Last modifier name
modificationDate	Last modification date
score	Number of votes for this element
scoreRatio	Ratio of votes for this element

These fields will not be displayed in the different examples below.

C - Element specific fields

Each kind of element can also add new columns:

- *How to use note export?*.
- *How to use card export?*.
- *How to use roll export?*.
- *How to use sticker export?*.
- *How to use freetext and Label export?*.
- *How to use gauge export?*.
- *How to use graph export?*.
- *How to use image export?*.
- *How to use team export?*

How to use note export?

This section describes the export of notes, VSM notes and shapes, which are all in the same class **com.iobeya.entity.Note**.

Fields of the **com.iobeya.entity.Note** class

These fields are shared by all instances of the **com.iobeya.entity.Note** class:

Column	Description	Mandatory?	Information
name	Name of the element in toolset	Optional	If blank, then element will be set as new tool
setName	Name of the toolset	Optional	If blank, it will not match any known toolsets, and legend will track it as unknown
color	Note background color	Optional	A blank value will lead to a black note (color = 0)
height	Height of the note	Optional	If set to blank, the large size will be used
width	Width of the note	Optional	If set to blank, the large size will be used
asset	Asset reference	Optional	Only useful for relevant elements (ex: VSM shapes)

Fields of the **BoardNote** entity type

> ***Important:*** **To match the element with an existing tool, the corresponding value for the following must be set: color, name, setName, height, width and layoutId.**

As of iObeya 3.6, notes can have several layouts. This layout is determined by the **layoutId** field.

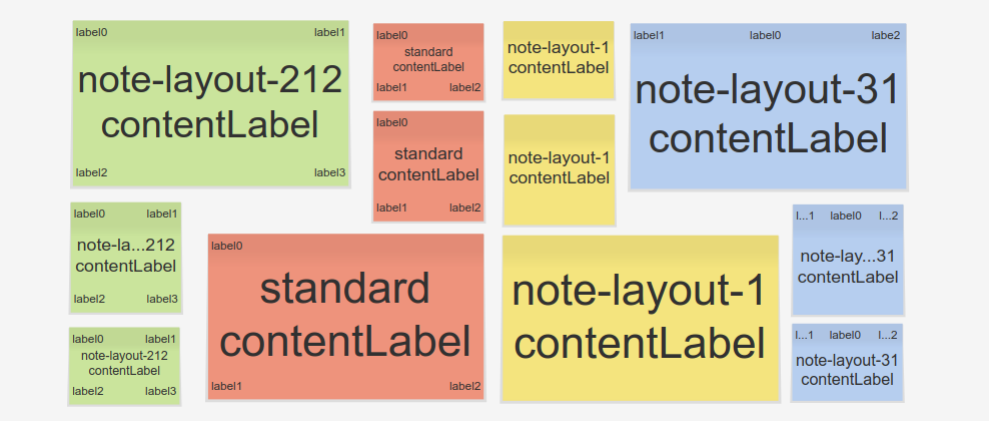
Column	Description	Mandatory?	Information
layoutId	Name (id) of the note layout	Optional	Only take values within layout ids given below. If blank or invalid, will be set to **standard**.

Each layout can contain a different number of labels. Each label has a corresponding column for its text content. They are all optional, leaving the corresponding note label empty when unset. The label columns taken into account for each note layout are given below:

Layout id	Label column names
note-layout-1	**contentLabel**
note-layout-31	**contentLabel**, **label0**, **label1**, **label2**
standard	**contentLabel**, **label0**, **label1**, **label2**
note-layout-212	**contentLabel**, **label0**, **label1**, **label2**, **label3**

These layouts are available in the following sizes:

- **large: 225x375 px**
- **medium: 150x150 px**
- **small: 105x150 px**



Fields of the **BoardVSMNote** entity type

> ***Important:*** **To match the element with an existing tool, the corresponding value for the following must be set: color, name, setName, height, width and layoutId.**

Similarly to notes, the layout of a VSM note is determined by a **layoutId** field.

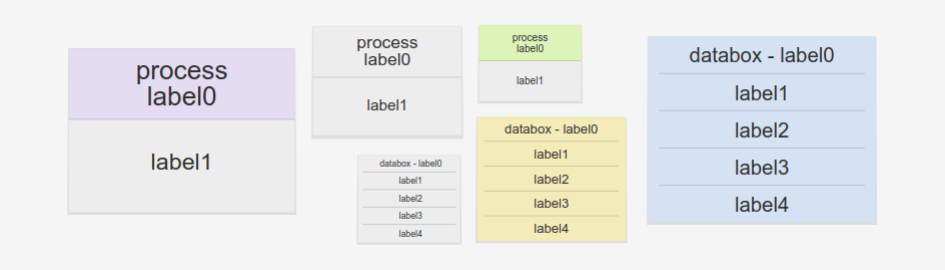
Column	Description	Mandatory?	Information
layoutId	Name of the note layout	Optional	Only take values within layout ids given below. If blank or invalid, will be set to **databox**.

Each layout can contain a different number of labels. Each label has a corresponding column for its text content. They are all optional, leaving the corresponding note label empty when unset. The label columns taken into account for each note layout are given below:

Layout id	Label column names
process	**label0**, **label1**
databox	**label0**, **label1**, **label2**, **label3**, **label4**

These layouts are available in the different following sizes:

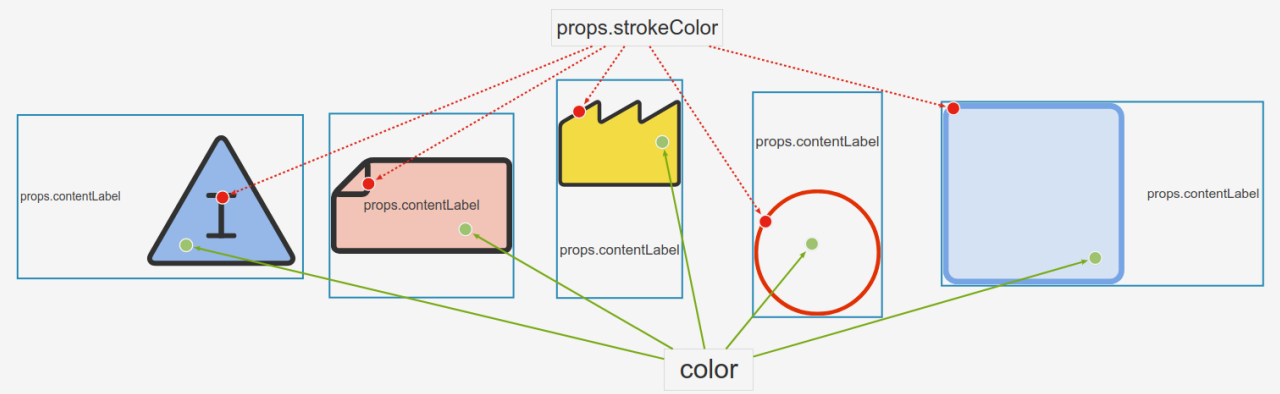
- For **process**:
  - **large: 310x225 px**
  - **medium: 204x150 px**
  - **small: 140x120 px**
- For **databox**:
  - **large: 312x255 px**
  - **medium: 204x170 px**
  - **small: 140x120 px**



Fields of the **BoardShape** entity type

> ***Important:*** **To match the element with an existing tool, the corresponding value for the following must be set: color, name, setName, height, width, props.textPosition, props.strokeColor.**

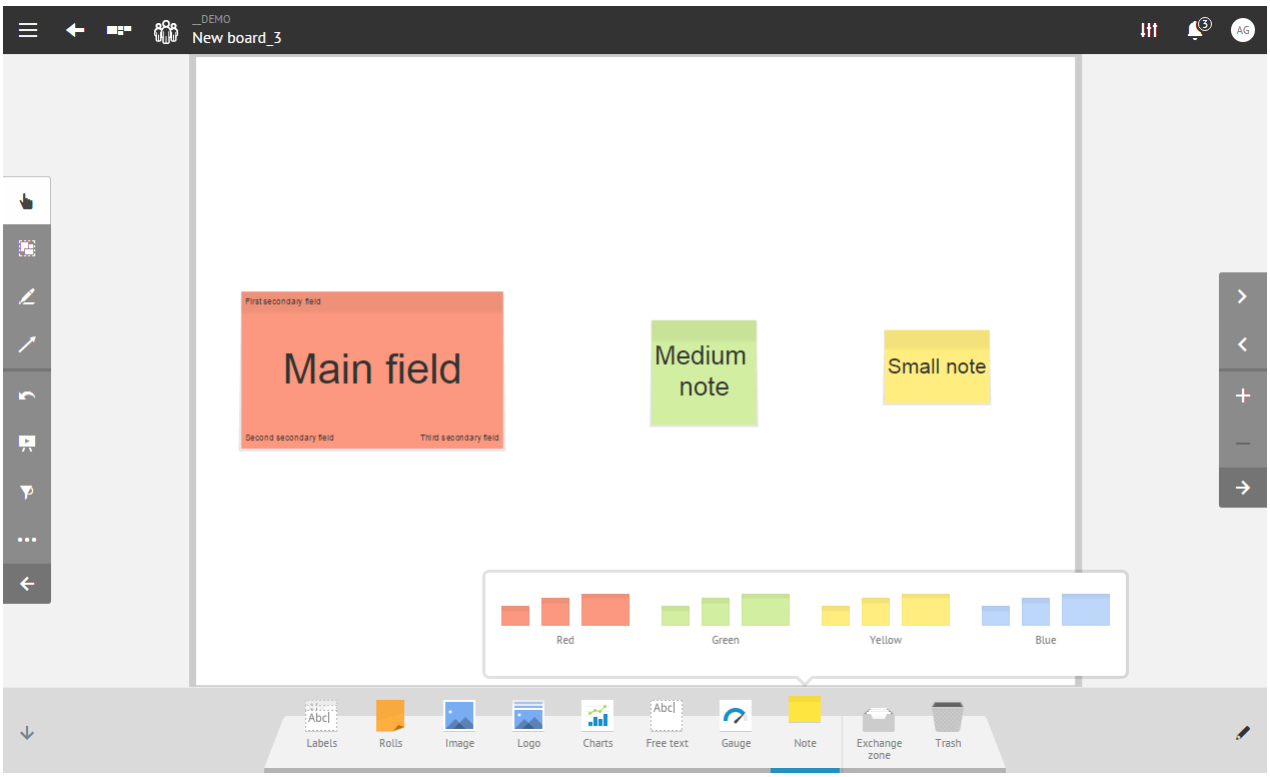
Column	Description	Mandatory?	Information
asset	Asset reference used for the shape	Mandatory	The format of asset reference is the same as for stickers or images, e.g.: **<com.iobeya.entity.Asset;13E82FA4-7F42-4bf3-A75D-EE1B11E961F4>** (see Asset export). The asset needs to be an SVG. For the **props.strokeColor** and **color** fields to work correctly, the SVG should have been created/modified accordingly, with the **stroke/fill** style being absent on SVG elements that should be respectively stroked/filled.
color	Fill color of this shape	Optional	Format: RGB as integer, or -1 for transparent. If blank, will be set to 16777215, meaning white.
props.strokeColor	Stroke color of this shape	Optional	Format: RGB as integer, or -1 for transparent. If blank, will be set to 0, meaning black.
props.textPosition	Position of text on this shape	Optional	Possible values: **top**, **left**, **right**, **bottom**, **inside**. If blank or invalid, will be set to **bottom**.
props.contentLabel	Text on this shape	Optional	



Example

For this example, three notes of different types and sizes have been added.

These three notes have been created from the notes toolset.



After exporting the board, there is one line per note in the CSV file:

class	id	color	container	deleted	height	width	contentLabel	label0	label1	label2	linkLabel	linkUrl	name	setName	x	y	zOrder
class com.iobeya.entity.Note	F8B535A7-60D0-F083-0DD5-3FE4F9F3F89B	16488575	<com.iobeya.entity.ElementContainer;dac5a8ef-d9e5-40cd-a210-3bc29fd47831>	0	225	375	Main field	Field #1	Field #2	Field #3			Red	Note	64	336	7
class com.iobeya.entity.Note	AE3C9D37-C499-C071-84D8-3FE5FF2F43FA	13823648	<com.iobeya.entity.ElementContainer;dac5a8ef-d9e5-40cd-a210-3bc29fd47831>	0	150	150	Medium note						Green	Note	653	377	8
class com.iobeya.entity.Note	90CA5053-3F75-F739-D557-3FE61B7F27D4	16772735	<com.iobeya.entity.ElementContainer;dac5a8ef-d9e5-40cd-a210-3bc29fd47831>	0	105	150	Small note						Yellow	Note	987	391	9

Note that:

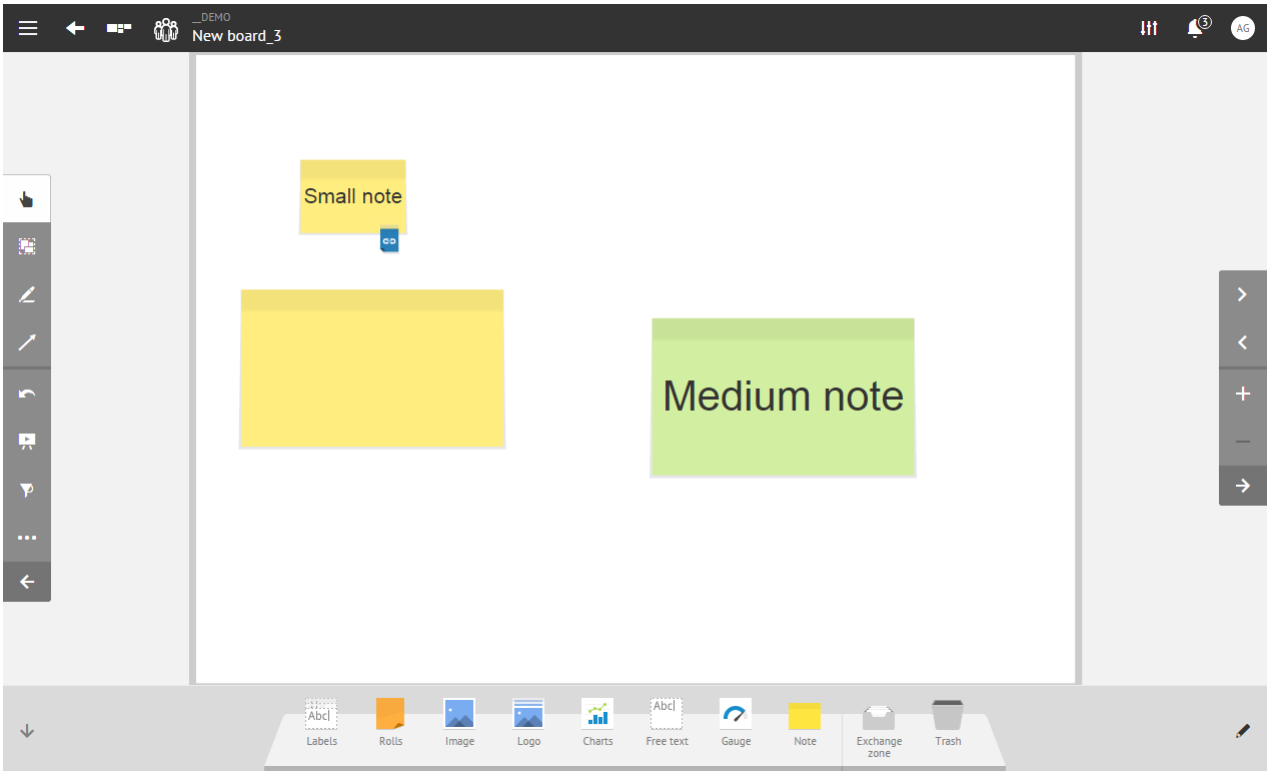
- there are 3 notes referencing the same tool set "Note": ***setName=Note***.
- each note references its own tool in the toolset:
  - the first note references the tool "Red": ***name=Red***.
  - the second note references the tool "Green": ***name=Green***.
  - the third note references the tool "Yellow": ***name=Yellow***.
- these three notes have different size formats and color backgrounds:
  - the first note is large and red: ***height=225***, ***width=375*** and ***color=16488575*** (in Hexadecimal format:#FB987F).
  - the second note is medium and green: ***height=150***, ***width=150*** and ***color=13823648*** (in Hexadecimal format:#D2EEA0).
  - the third note is small and yellow: ***height=105***, ***width=150*** and ***color=16772735*** (in Hexadecimal format: #FFEE7F).
- all the fields of the first note are populated:
  - the content of the main field is "Main field": ***contentLabel=Main field***.
  - the content of the top field is "Field #1": ***label0=Field#1***.
  - the content of the bottom-left field is "Field #2": ***label1=Field #2***.
  - the content of the bottom-right field is "Field #3": ***label2=Field #3***.

To import, the notes are modified as follows:

- the first note will become a yellow note and all fields will be emptied.
- the second note will become a large note.
- the third note will be moved and a hyperlink will be added.

class	id	color	container	deleted	height	width	contentLabel	label0	label1	label2	linkLabel	linkUrl	name	setName	x	y	zOrder
class com.iobeya.entity.Note	F8B535A7-60D0-F083-0DD5-3FE4F9F3F89B	16772735	<com.iobeya.entity.ElementContainer;dac5a8ef-d9e5-40cd-a210-3bc29fd47831>	0	225	375							Yellow	Note	64	336	7
class com.iobeya.entity.Note	AE3C9D37-C499-C071-84D8-3FE5FF2F43FA	13823648	<com.iobeya.entity.ElementContainer;dac5a8ef-d9e5-40cd-a210-3bc29fd47831>	0	225	375	Medium note						Green	Note	653	377	8
class com.iobeya.entity.Note	90CA5053-3F75-F739-D557-3FE61B7F27D4	16772735	<com.iobeya.entity.ElementContainer;dac5a8ef-d9e5-40cd-a210-3bc29fd47831>	0	105	150	Small note				iobeya.com	www.iobeya.com	Yellow	Note	150	150	9

After importing, the result will be as follows:



How to use card export?

Specific card fields

> ***Important:*** **To match the element with an existing tool, the corresponding value for the following must be set: color, name, setName.**

Card export generates several lines:

- the first one represents the object: class com.iobeya.entity.Card.
- additional lines represent items in the checklist of the card: class com.iobeya.entity.ChecklistItem.

Card object:

Column	Description	Mandatory?	Information
name	Name of the element in toolset	Optional	A blank value will lead to an unknown tool
setName	Name of the toolset	Optional	If blank, then it will not match any known toolset and legend will track it as unknown
color	Card background color	Optional	A blank value will lead to a black card (color = 0)
height	Height of the card	Optional	If set to blank, the large size will be used
width	Width of the card	Optional	If set to blank, the large size will be used
label0	Title of the card	Optional	If blank, the main field will be empty. However, when editing the card this field must be populated in order to save the card.

Column	Description	Mandatory?	Information
props.description	Field description	Optional	
props.endDate	Field due date	Optional	
props.metric	Field value	Optional	
props.priority	Flag that indicates if the card has been marked as high priority	Optional	If blank, value is set to false
isArchived	Flag that indicates if the card has been archived	Optional	If blank, value is set to false
assignments	List of usernames assigned to the card	Optional	Username must be separated by a space. No more than 5 usernames can be specified. If a username does not exist on the platform, the user will not be shown on the card.

Checklist item

Column	Description	Mandatory?	Information
class	Class of the element checklist item	Mandatory	Must be: class com.iobeya.entity.ChecklistItem
id	Indicates the ID of the card element that contains this checklist item and the index of this item in the list	Mandatory	
label	Content of the checklist item	Optional	If set to blank, the checklist item is emptied
status	Flag that indicates if the checklist item is solved	Optional	If blank, the value is set to false (not solved)

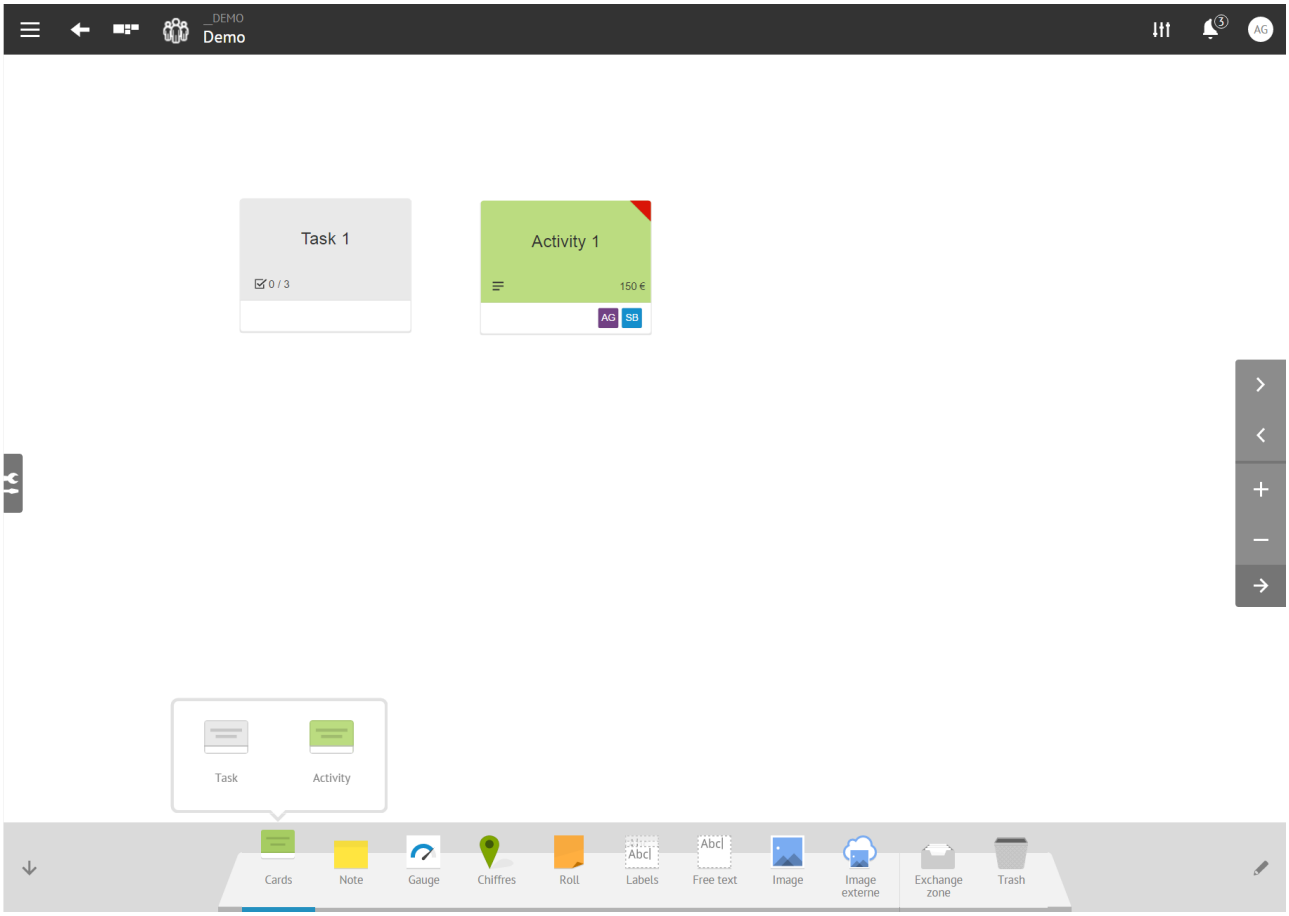
As of version 3.6, cards are available in the following sizes:

- **large: 379x297 px**
- **medium: 254x197 px**
- **small: 182x142 px**

Example

For this example, two cards of different types and with different properties have been added:

- task 1 is a card of type "Task" and contains 3 items in a checklist.
- activity 1 is a card of type "Activity", with a due date, a value, a description and 2 members.



After exporting the board, there is one line per card in the CSV file:

class	id	setName	name	label0	color	assignments	props.description	props.endDate	props.metric	props.priority	isArchived	deleted
class com.iobeya.entity.Card	67669440-97F3-2AB5-2F66-237BFEDB51CE	Cartes	Task	Task 1	15329769					false	false	0
class com.iobeya.entity.Card	3C3F56D2-A13D-A0DD-6D84-237D273FEA6C	Cartes	Activity	Activity 1	12311680	agehl sbourderon	Description of my card		150 €	true	false	0

And one line for each checklist item of Task 1:

class	id	label	status
class com.iobeya.entity.ChecklistItem	<com.iobeya.entity.ChecklistItem.Key;parentId:67669440-97F3-2AB5-2F66-237BFEDB51CE;index:0>	Element 1	false
class com.iobeya.entity.ChecklistItem	<com.iobeya.entity.ChecklistItem.Key;parentId:67669440-97F3-2AB5-2F66-237BFEDB51CE;index:1>	Element 2	false
class com.iobeya.entity.ChecklistItem	<com.iobeya.entity.ChecklistItem.Key;parentId:67669440-97F3-2AB5-2F66-237BFEDB51CE;index:2>	Element 3	false

Note that:

- there are 2 cards referencing the same toolset "Cards": setName=Cards.
- each card references its own tool in the toolset:
  - the first card references the tool "Task": name=Task.
  - the second card references the tool "Activity": name=Activity.
- there are 3 elements in the checklist of the task:
  - items referencing the ID of the task element and the index in the list: id=<com.iobeya.entity.ChecklistItem.Key;parentId:67669440-97F3-2AB5-2F66-237BFEDB51CE;index:0> with parentId=the ID of the card.
  - no items are resolved: status=false.
- the fields description and value of the activity are populated, and there are two assignees to this activity:
  - the content of the description is "Description of my card": props.description=Description of my card.
  - the content of the value field is "150": props.metric=150.
  - the activity is a high priority: priority=true.
  - the two members are the usernames agehl and sbourderon: assignments=agehl sbourderon.

To import, the cards are modified as follows:

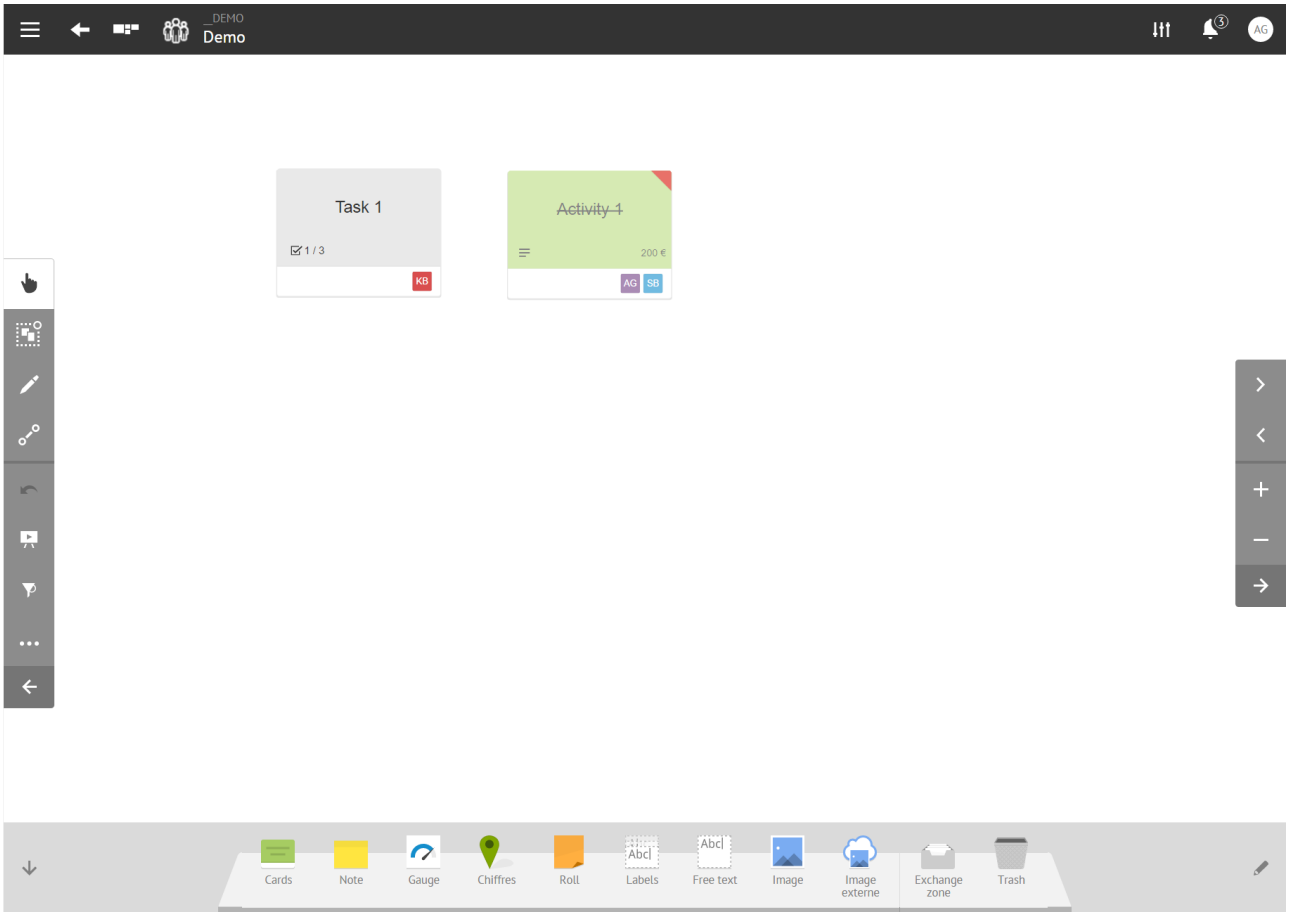
- user kbelmana is assigned to Task 1.
- the first element of Task 1's checklist is checked off.

- the value of Activity 1 is changed to 200.
- ... and Activity 1 is archived.

class	id	setName	name	label0	color	assignments	props.description	props.endDate	props.metric	props.priority	isArchived	deleted
class com.iobeya.entity.Card	67669440-97F3-2AB5-2F66-237BFEDB51CE	Cartes	Task	Task 1	15329769	kbelmana				false	false	0
class com.iobeya.entity.Card	3C3F56D2-A13D-A0DD-6D84-237D273FEA6C	Cartes	Activity	Activity 1	12311680	agehl sbourderon	Description of my card		200 €	true	true	0

class	id	label	status
class com.iobeya.entity.ChecklistItem	<com.iobeya.entity.ChecklistItem.Key;parentId	Element 1	true
class com.iobeya.entity.ChecklistItem	<com.iobeya.entity.ChecklistItem.Key;parentId	Element 2	false
class com.iobeya.entity.ChecklistItem	<com.iobeya.entity.ChecklistItem.Key;parentId	Element 3	false

After importing, the result will be as follows:



How to use roll export?

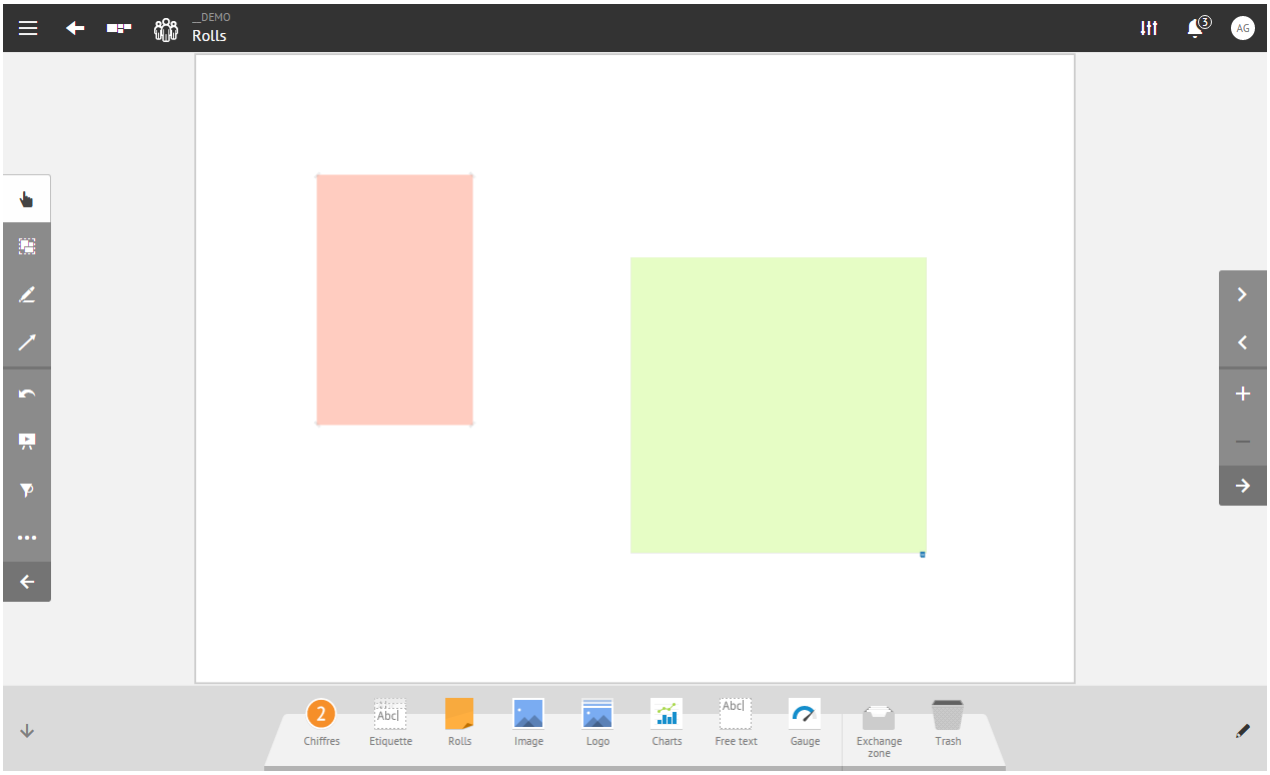
Specific roll fields

> ***Important:*** **To match the element with an existing tool, the corresponding value for the following must be set: color, name and setName.**

Column	Description	Mandatory?	Information
name	Name of the element in its toolset	Optional	A blank value will lead to an unknown tool
setName	Tool	Optional	If blank, then it will not match any known toolset and legend will track it as unknown
color	Color of the roll	Optional	If empty, value roll will be black (color 0)

Example

For this example, 2 rolls have been added to the board:



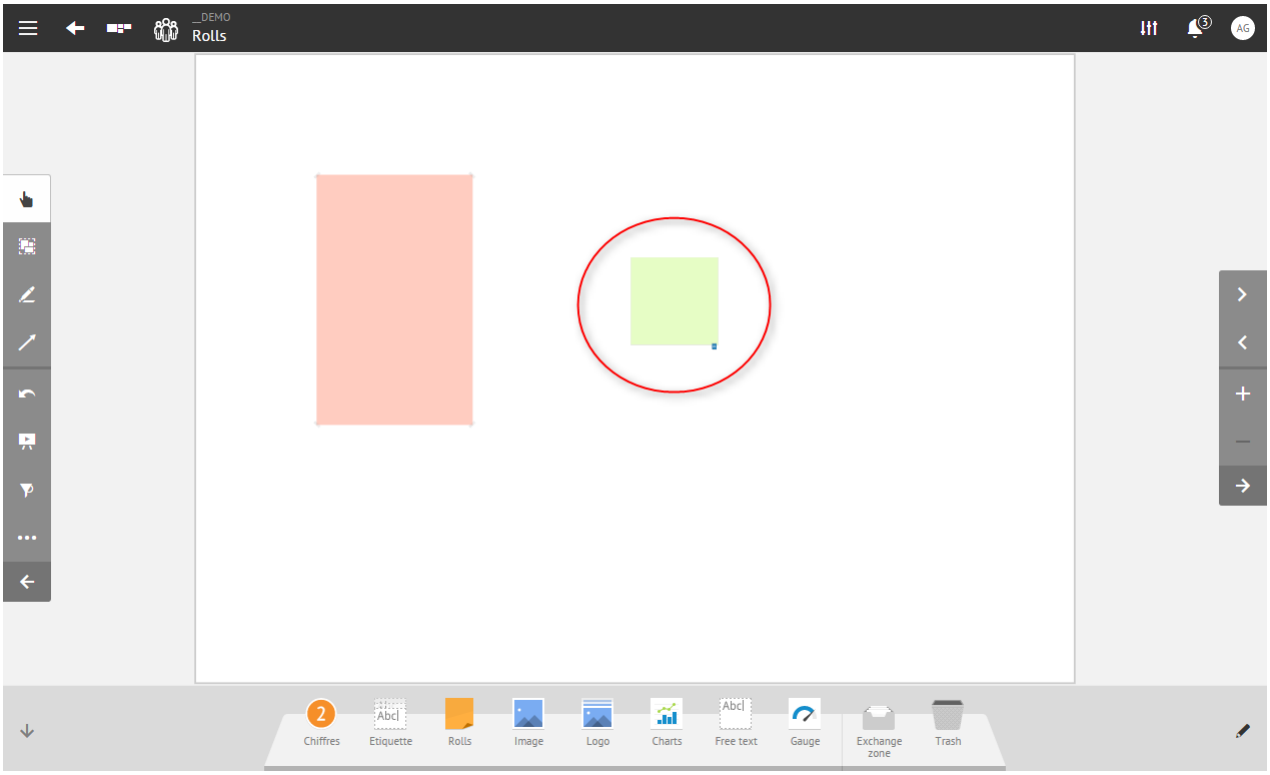
After exporting the board, there are two lines (one for each roll) in the CSV file:

class	color	container	deleted	height	id	isAnchored	linkLabel	linkUrl	name	setName	width	x	y	zOrder
class com.iobeya.entity.Roll	16764096	<com.iobeya.entity.ElementContainer;62b0fa16-c238-485b-b0d4-6ec15a09536a>	0	1436	30ABEA06-2294-BC3A-9023-35CF7B21B19D	true			Roll 1	Rolls	896	692	687	2
class com.iobeya.entity.Roll	15138245	<com.iobeya.entity.ElementContainer;62b0fa16-c238-485b-b0d4-6ec15a09536a>	0	1694	50E688D6-1FF4-70F1-521C-35CF93756E5B	false	iobeya.com	http://www.iobeya.com	Roll 2	Rolls	1696	2497	1162	4

Note that:

- the export file contains two rolls that reference two different tools (Roll 1, Roll2) from the toolset Rolls: ***name=Roll 1*** and ***name=Roll 2***.
- the first roll (Roll 1) is locked on the board: ***isAnchored=true***.
- the size of the first roll on the board is 896x1436 pixels: ***width=896*** and ***height=1436***.
- the second roll (Roll 2) contains a hyperlink with label iobeya.com and URL http://www.iobeya.com: ***linkLabel=iobeya.com*** and ***linkURL=http://www.iobeya.com***.
- the color of the first roll is red: ***color=16764096*** (integer code corresponding to #FFCCC0 in hexadecimal).

If there is a requirement to change the size of the second roll on the board, the values of height and width properties can be modified for this element before importing this roll to the board:

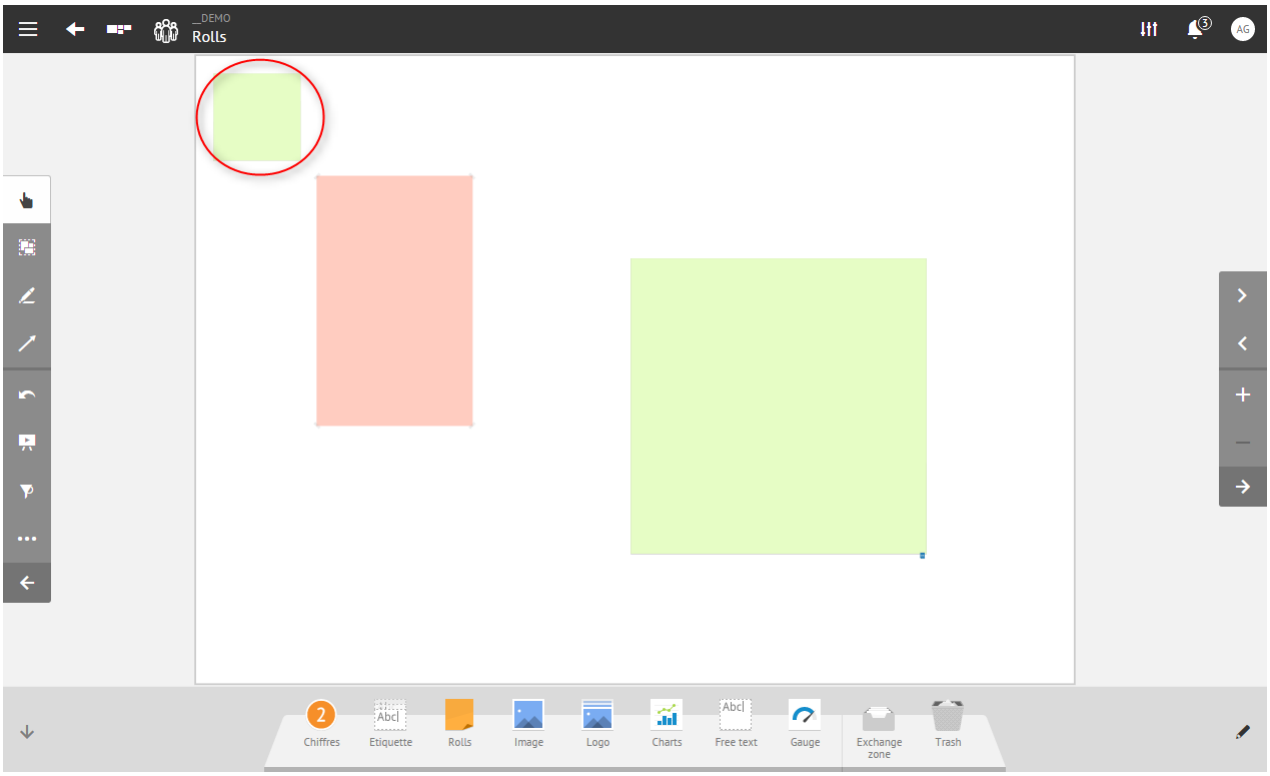


As the rolls have the same ID on the board and in the excel file, the imported roll will replace the existing roll.

In order to create a new roll that references the tool Roll 2 of the toolset rolls with a size of 500x500 px and 100x100px from the top-left corner:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	class	color	container	deleted	height	id	isAnchored	linkLabel	linkUrl	name	setName	width	x	y	zOrder
2	class com.iobeya.entity.Roll	16764096	<com.iobeya.entity.ElementContainer;62b0fa16-c238-485b-b0d4-6ec15a09536a>	0	1436	30ABEA06-2294-BC3A-9023-35CF7B21B19D	true			Roll 1	Rolls	896	692	687	2
3	class com.iobeya.entity.Roll	15138245	<com.iobeya.entity.ElementContainer;62b0fa16-c238-485b-b0d4-6ec15a09536a>	0	1694	50E688D6-1FF4-70F1-521C-35CF93756E5B	false	iobeya.com	http://www.iobeya.com	Roll 2	Rolls	1696	2497	1162	4
4	class com.iobeya.entity.Roll	15138245		0	500		false			Roll 2	Rolls	500	100	100	

The result will be as follows:



How to use sticker export?

Specific sticker fields

Sticker export generates several lines:

- the first line represents the sticker object.
- additional lines represent encoded data of the image asset of the sticker.

Sticker object

Column	Description	Mandatory?	Information
isColoredSticker	Is it a colored sticker, or a sticker of an image?	Mandatory	0 means the item is a sticker of an image; 1 means the item is a colored sticker
color	If it is a colored sticker, what color is it?	Mandatory	⚠ **Warning:** The specified color code should exist in the toolset
stickerImage	Id of the pattern/image of the stickers	Mandatory	⚠ **Warning:** Value should be an existing asset on the platform
name	Name of the element in its toolset	Optional	A blank value will lead to an unknown tool
setName	Toolset name	Optional	If blank, it will not match any known toolset and legend will track it as unknown
height	Height of the sticker	Optional	
width	Width of the sticker	Optional	

Sizes supported in iObeya are:

- **large: 112x112 px**
- **medium: 64x64 px**
- **small: 32x32 px**

Sticker asset

Column	Description	Mandatory?	Information
id	ID of the referent image object	Mandatory	A sticker asset can be composed of one or multiple image lots
data	Image data lot (base64 encoded)	Mandatory	
isSystem	Boolean that indicates if the asset used is a sticker included in iObeya standard catalog	Information	
category	Indicates the family of a catalog sticker	Information	
mimeType	Image format	Information	Automatically retrieved during the import
assetWidth	Width of the image	Information	Automatically retrieved during the import
assetHeight	Height of the image	Information	Automatically retrieved during the import

Column	Description	Mandatory?	Information
dataLocation	Storage type	Information	Automatically retrieved during the import
filesystem	Boolean that indicates if the element is stored in filesystem	Information	Automatically updated during the import
database	Boolean that indicates if the element is stored in database	Information	Automatically updated during the import

Sticker asset is useful when a user tries to re-import an export to a platform that did not recognise the sticker asset, in order to be able to create it automatically.

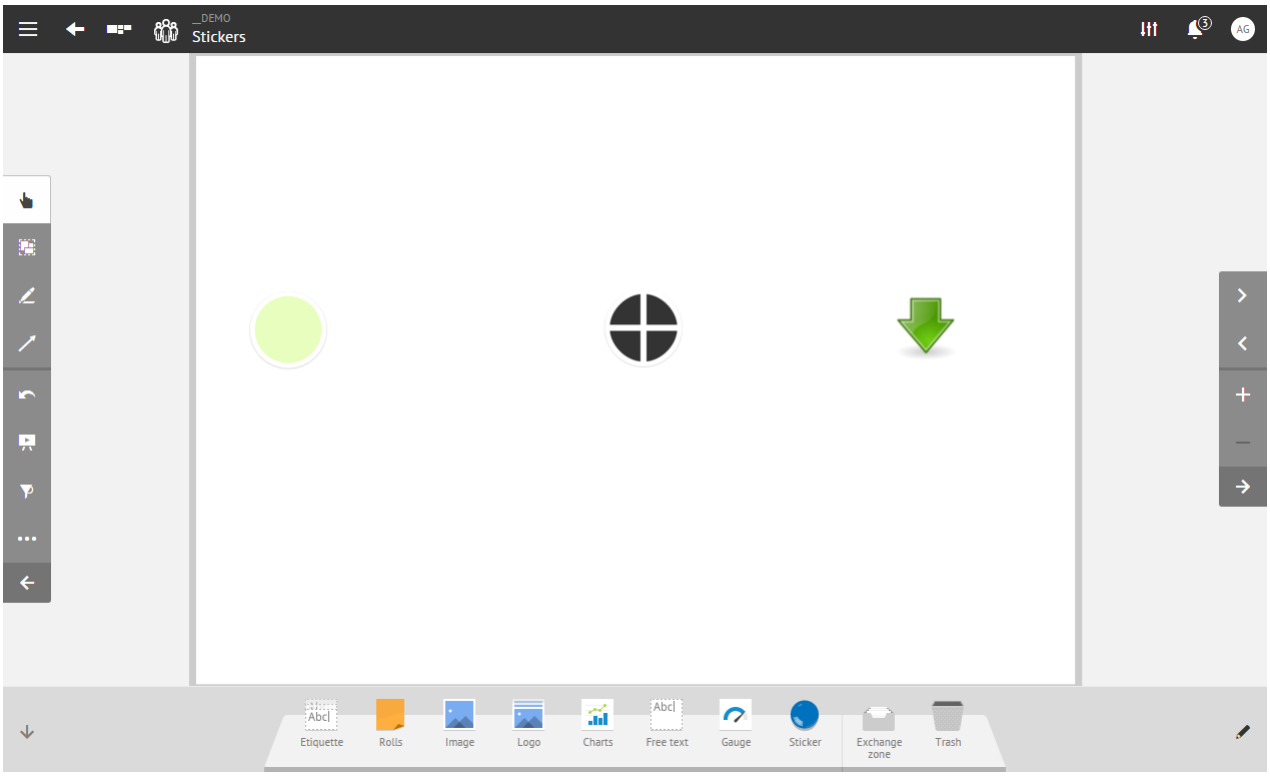
It is important to note that the asset line is especially useful for custom stickers using images uploaded by the user.

For system stickers, because they are known on all platforms of iObeya (of an identical version), it is not mandatory to specify asset line when attempting to re-import a CSV file.

Example

For this example, 3 different types of stickers have been added to the board:

- the first one is a colored sticker: using a color chosen when configuring the tool.
- the second one is a sticker using an icon from the icon catalog.
- the third one is a custom sticker using an uploaded personal icon.



For the colored sticker:

After exporting the board, one line corresponds to the colored sticker.

Sticker object

class	id	color	csvColumnTitle	csvColumnValue	deleted	height	isAnchored	isColoredSticker	name	setName	stickerImage	width	x	y	zOrder
class com.iobeya.entity.Sticker	A8728A0B-6655-BED9-6209-35FEB81F0395	15204030	Sticker	Colored sticker	0	112	false	true	Colored sticker	Sticker		112	75	336	1

Note that:

- this is a colored sticker: ***isColoredSticker=true***.

- the color of this sticker is light green (#E7FEBE): ***color=15204030*** (integer code).

For the icon catalog sticker:

After exporting the board, two lines correspond to the icon catalog sticker:

- the first line represents the image object.
- the second line represents encoded data of this image asset.

Sticker object

class	id	color	csvColumnTitle	csvColumnValue	deleted	height	isAnchored	isColoredSticker	name	setName	stickerImage	width	x	y	zOrder
class com.iobeya.entity.Sticker	D0591C85-EF7C-F005-923B-3F560D76BE48	0	Sticker	Catalog icon sticker	0	112	false	false	Catalog icon sticker	Sticker	<com.iobeya.entity.Asset;F63F9645-D9D4-4416-9D40-20D7867F42CF>	112	564	333	3

Note that:

- this sticker is not a colored sticker but an image sticker: ***isColoredSticker=false***.
- this sticker requires an asset and this asset is 13E82FA4-7F42-4bf3-A75D-EE1B11E961F4: ***stickerImage=<com.iobeya.entity.Asset;13E82FA4-7F42-4bf3-A75D-EE1B11E961F4>***.

Sticker asset

class	id	assetHeight	assetWidth	category	container	data	dataType	database	fileSystem	isSystem	maxRatio	mimeType
class com.iobeya.entity.Asset	13E82FA4-7F42-4bf3-A75D-EE1B11E961F4	128	128	SK_TOOL_STICKER_MAGNET		iVBORw0KGgoAA	filesystem	false	true	true	1	image/png

Note that:

- the asset used for this sticker is provided by iObeya: isSystem=true.

For the custom icon sticker:

After exporting the board, multiple lines correspond to the custom sticker:

- the first line represents the image object.
- the following line represents encoded data of the custom sticker asset.

Sticker object

class	id	color	csvColumnTitle	csvColumnValue	deleted	height	isAnchored	isColoredSticker	name	setName	stickerImage	width	x	y	zOrder
class com.iobeya.entity.Sticker	7510A811-2A46-A7EF-1472-3F56194A629C	0	Sticker	Custom icon sticker	0	112	false	false	Custom icon sticker	Sticker	<com.iobeya.entity.Asset;EB932DAC-86CA-5BAD-1D1A-35FDCB0687D8>	112	957	347	4

Note that:

- this sticker is an image sticker: ***isColoredSticker=false***.
- this sticker requires an asset and this asset is EB932DAC-86CA-5BAD-1D1A-35FDCB0687D8: ***stickerImage=<com.iobeya.entity.Asset;EB932DAC-86CA-5BAD-1D1A-35FDCB0687D8>***.

Sticker asset

class	id	assetHeight	assetWidth	category	container	data	dataType	database	fileSystem	isSystem	maxRatio	mimeType
class com.iobeya.entity.Asset	EB932DAC-86CA-5BAD-1D1A-35FDCB0687D8_1-2	256	256		<com.iobeya.entity.AssetContainer;d78aabcdS1ecb54031Sa411S25169b9323c7>	iVBORw0KGgoAA	filesystem	false	true	false	2	image/png
class com.iobeya.entity.Asset	EB932DAC-86CA-5BAD-1D1A-35FDCB0687D8_2-2	256	256		<com.iobeya.entity.AssetContainer;d78aabcdS1ecb54031Sa411S25169b9323c7>	AAVuHHc_Zoy2F	filesystem	false	true	false	2	image/png

Note that:

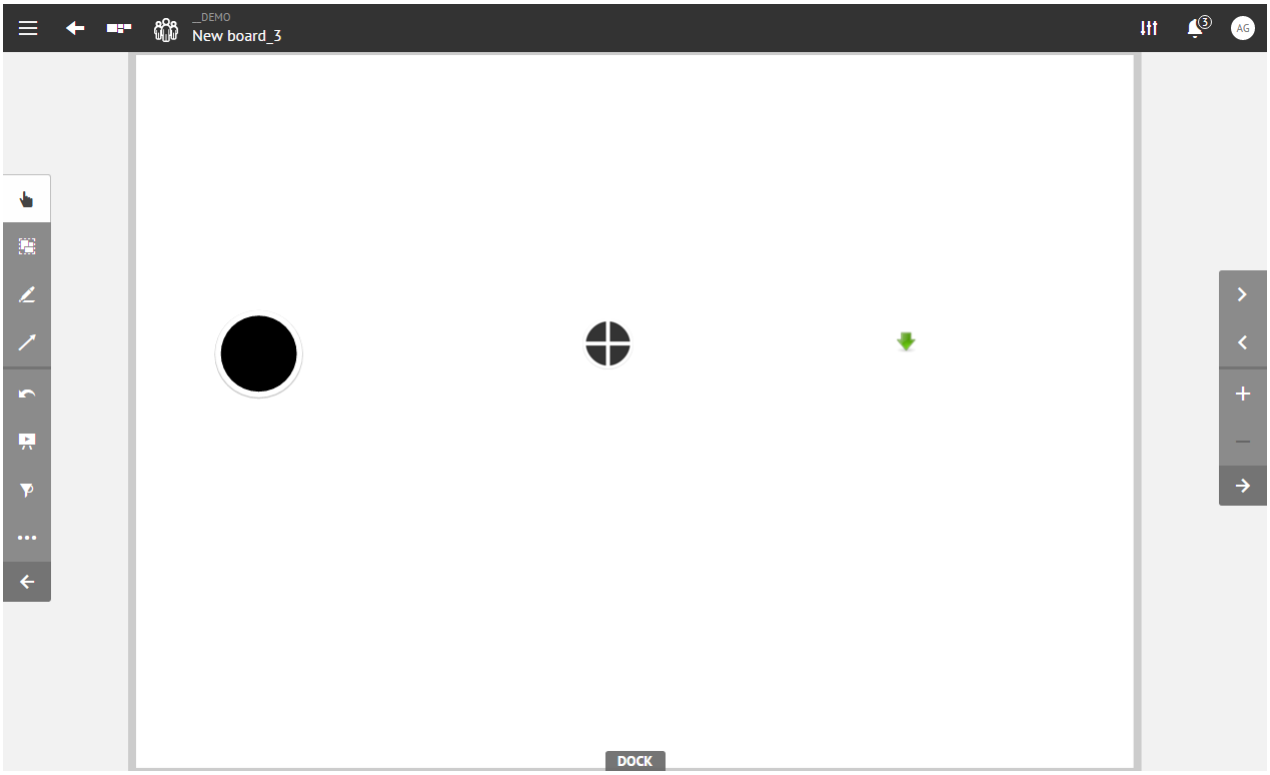
- the asset used for this sticker is a custom sticker using an image uploaded by a user: ***isSystem=false***.
- the asset is mandatory if there is a requirement to upload this sticker to a new platform.
- the asset of the sticker has been cut into 2 parts to support opening a CSV file in Excel (Excel has a limitation of number of characters in cells).

For import, the following will be modified:

- the color of the colored sticker to black.
- the sizes of the icon catalog, and custom stickers to medium and small.

class	id	color	csvColumnTitle	csvColumnValue	deleted	height	isAnchored	isColoredSticker	name	setName	stickerImage	width	x	y	zOrder
class com.iobeya.entity.Sticker	A8728A0B-6655-BED9-6209-35FEB81F0395	0	Sticker	Colored sticker	0	112	false	true	Colored sticker	Sticker		112	75	336	1
class com.iobeya.entity.Sticker	<com.iobeya.entity.ElementContainer;998	0	Sticker	Catalog icon sticker	0	64	false	false	Catalog icon sticker	Sticker	<com.iobeya.entity.Asset;F63F9645-D9D4-4416-9D40-20D7867F42CF>	64	564	333	3
class com.iobeya.entity.Sticker	<com.iobeya.entity.ElementContainer;998	0	Sticker	Custom icon sticker	0	32	false	false	Custom icon sticker	Sticker	<com.iobeya.entity.Asset;EB932DAC-86CA-5BAD-1D1A-35FDCB0687D8>	32	957	347	4

The result of the import is as follows:



How to use freetext and label export?

Specific freetext fields

Column	Description	Mandatory?	Information
contentLabel	Content of the label	Optional	If blank, then the content will be empty
fontSize	Text size	Optional	⚠ **Warning:** font size should match an existing size in the application
fontColor	Text color	Optional	The value of RGB color code is expressed in integer format (the hexadecimal value must be converted to an integer value)
backgroundColor	Background color	Optional	The value of RGB color code is expressed in integer format (the hexadecimal value must be converted to integer value)
textAlign	Horizontal text alignment property	Optional	Possible values supported: **right**, **center**, **left**
textVerticalAlign	Vertical text alignment property	Optional	Possible values supported: **top**, **middle**, **bottom**
fitToText	Boolean that indicates if the label adapt to the content	Optional	If true, the size of the label will be adapted to its content ⚠ **Warning:** this property cannot be true if **textAutoFit** is true; these properties are mutually exclusive.

Column	Description	Mandatory?	Information
textAutoFit	Boolean that indicates if the text adapts its size to the Label size	Optional	If true, the size of the text will be adapted to the label ⚠ **Warning:** this property cannot be true if **fitToText** is true; these properties are mutually exclusive.

Specific label fields

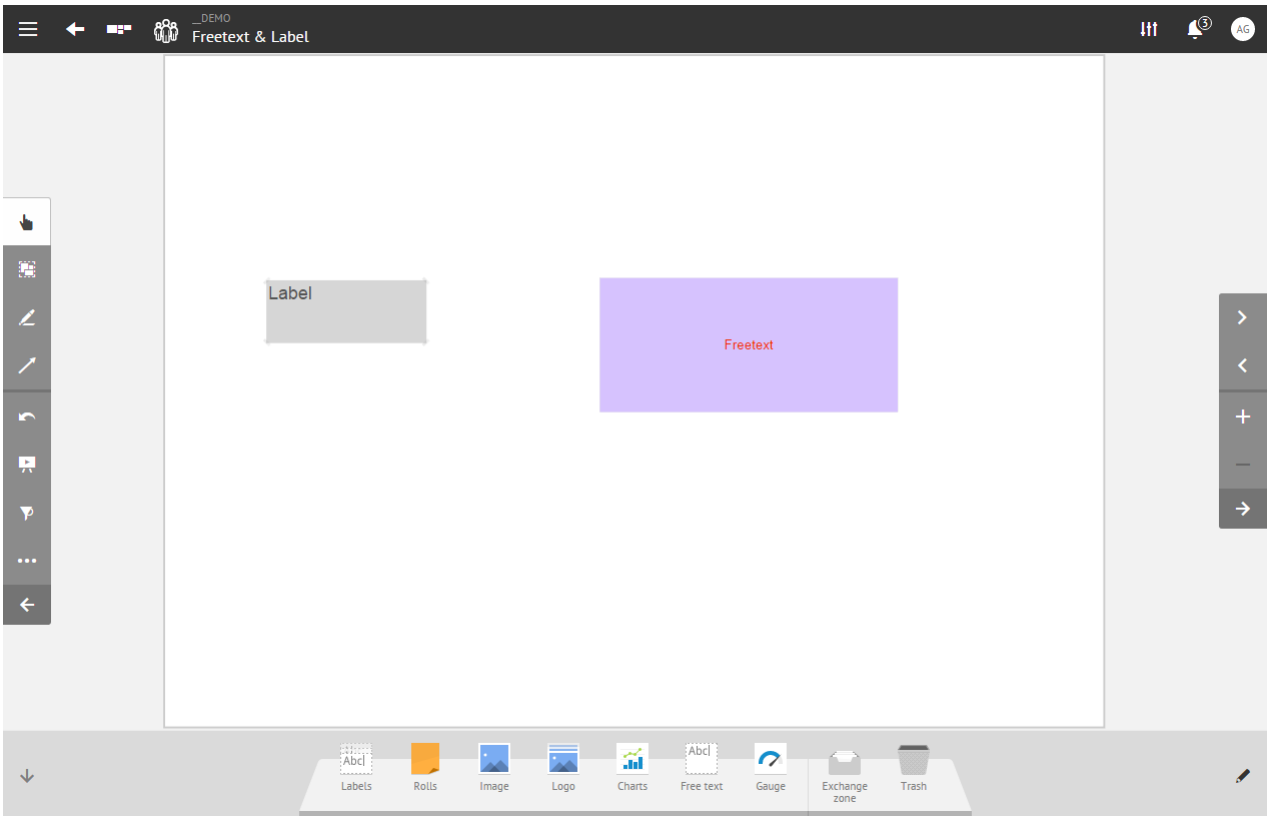
> ***Important:*** **To match the element to an existing tool, the same name as that existing tool in the dock must be used.**

Label and freetext parameters are exactly the same except for the below two parameters:

Column	Description	Mandatory?	Information
name	Name of the element in its toolset	Optional	Allows the linking of the element to an existing tool
setName	Toolset name	Optional	If blank, it will not match any known toolset, therefore its type will be unknown

Example

For this example, a freetext and a label have been added:



After exporting the board, two lines (one for the freetext and one for the label) have been created in the CSV file:

class	contentLabel	deleted	backgroundColor	fitToText	fontColor	fontFamily	fontSize	height	id	isAnchored	linkLabel	linkUrl	name	setName	textAlign	textAutofit	textVerticalAlign	width	x	y	zOrder
class com.iobeya.entity.Freetext	Freetext	0	14074622	false	15806465	arial	72	721	5C3CEDBF-5FD3-F210-9DB0-3A2FC7C0488B	false			Free text		center	false	middle	1601	2335	1191	16
class com.iobeya.entity.Label	Label	0	14079702	false	5592405	arial	96	337	CC8EA011-12FE-0DF0-9054-3A2F6C55CCAE	true			Big title	Labels	left	false	top	860	544	1204	26

Note that:

- the content of the freetext is "Freetext" and the content of the label is "Label": ***contentLabel=Freetext*** and ***contentLabel=Label***.
- the label is locked on the board: ***isAnchored=true***.

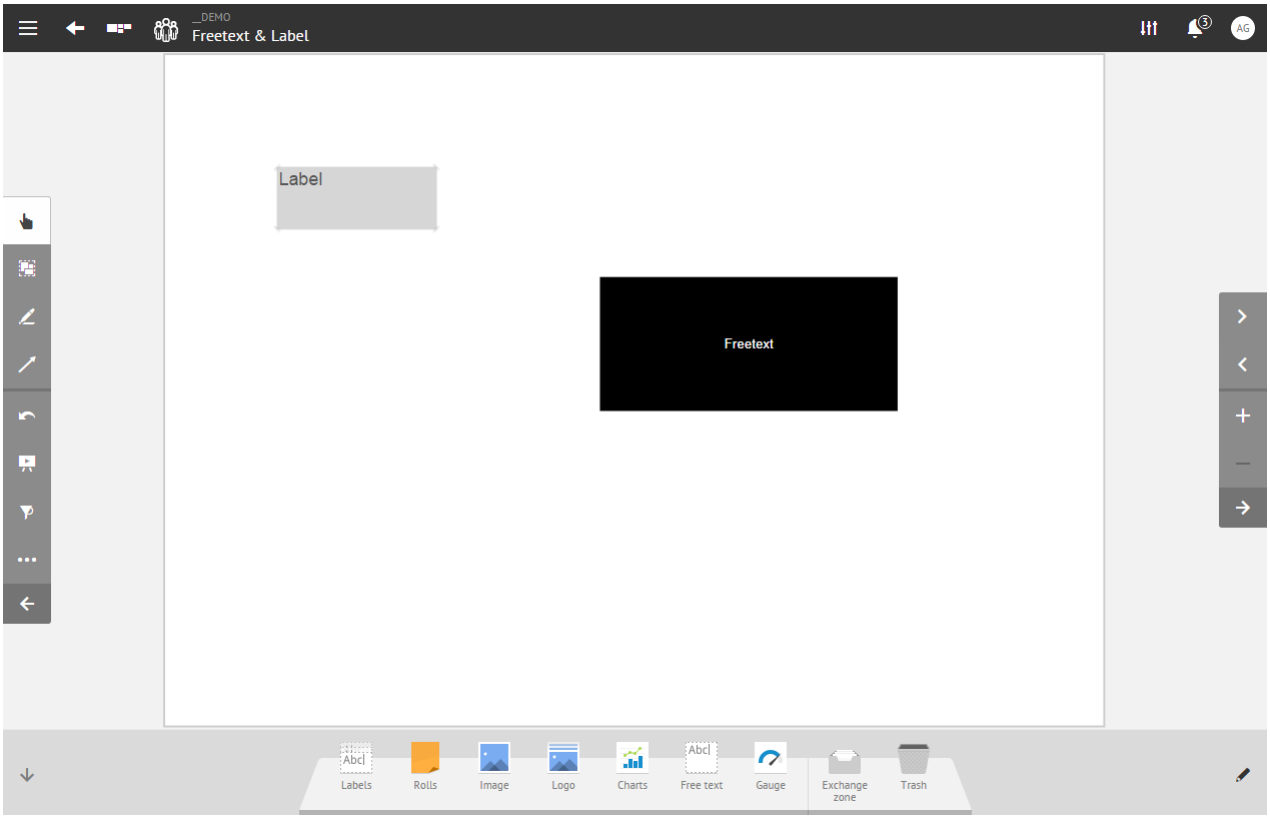
- the label references the tool "Big title" in the toolset "Labels": ***name=Big Title*** and ***setName=Labels***.
- the text of the freetext is aligned horizontally and vertically: ***textAlign=center*** and ***textVerticalAlign=middle***.
- the text of the label is aligned in the top-left corner: ***textAlign=left*** and ***textVerticalAlign=top***.

For import, the following changes will be made:

- the position of the label.
- some properties of the freetext:
  - background color to black.
  - text color to white.
  - fit-to-text to true.

class	contentLabel	deleted	backgroundColor	fitToText	fontColor	fontFamily	fontSize	height	id	isAnchored	linkLabel	linkUrl	name	setName	textAlign	textAutofit	textVerticalAlign	width	x	y	zOrder
class com.iobeya.entity.Freetext	Freetext	0	0	true	16777215	arial	72	721	5C3CEDBF-5FD3-F210-9DB0-3A2FC7C0488B	false			Free text		center	false	middle	1601	2335	1191	16
class com.iobeya.entity.Label	Label	0	14079702	false	5592405	arial	96	337	CC8EA011-12FE-0DF0-9054-3A2F6C55CCAE	true			Big title	Labels	left	false	top	860	600	600	26

The result will be as follows:



How to use gauge export?

Specific gauge fields

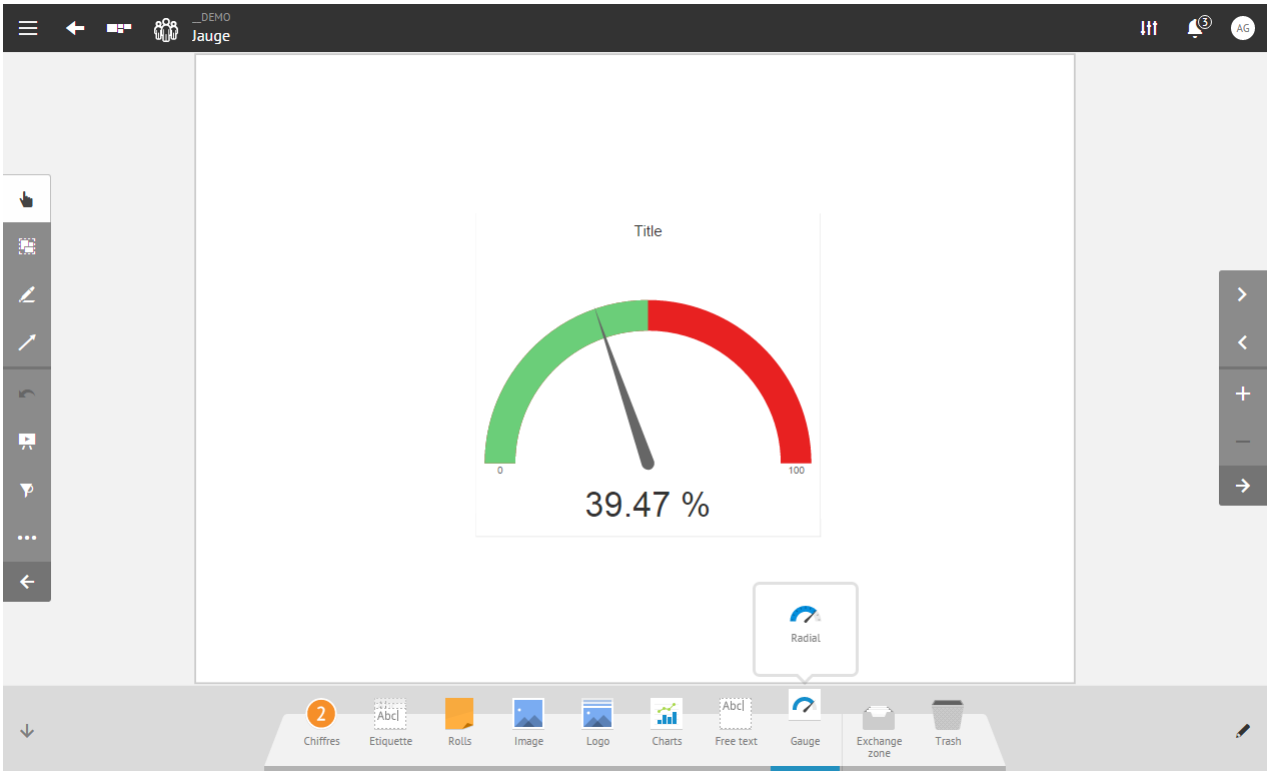
Column	Description	Mandatory?	Information
kind	Represents the kind of gauge	Mandatory	Only radial gauge is supported
label	Title of the gauge	Optional	If blank, no title on the gauge
name	Name of the element in its toolset	Optional	A blank value will lead to an unknown tool

Column	Description	Mandatory?	Information
setName	Toolset name	Optional	If blank, it will not match any known toolset and legend will track it as unknown
needleValue	Value of the gauge	Optional	If blank, the value is set to 0
maxScale	Represents the maximum scale value of the gauge	Optional	If blank, the value is set to 0 Make sure that the min value is lower than the max value
minScale	Represents the minimum scale value of the gauge	Optional	
themeIndex	Number of the color theme selected	Optional	There are four color themes that can be selected (between 0 and 3)
showMinMax	Boolean that indicates if the min and the max scale have to appear on the gauge	Optional	
showTicks	Boolean that indicates if the tick of threshold has to appear on the gauge	Optional	
showValue	Boolean that indicates if the tick of threshold has to appear on the gauge	Optional	
invertColors	Boolean that indicates if the color scheme used is inverted	Optional	

Column	Description	Mandatory?	Information
thresholdsNumber	Number of different colors in the gauge (define the number of thresholds in the gauge)	Mandatory	It is possible to define a value between 1 and 5 colors: - 1 color: threshold is defined by the min value scale and the max value scale - 2 colors: there is one threshold between the min and the max value scale - 3 colors: there are two thresholds between the min and the max value scale - 4 colors: there are three thresholds between the min and the max value scale - 5 colors: there are four thresholds between the min and the max value scale
threshold1	Value of the first threshold	Optional	This threshold value is used if the number of colors selected for the gauge is >=2 (thresholdNumber >=2). This value must be defined between the min value scale and the next value scale (min < threshold1 < threshold2).
threshold2	Value of the second threshold	Optional	This threshold value is used if the number of colors selected for the gauge is >=3 (thresholdNumber >=3). This value must be defined between the min value scale and the next value scale (threshold1 < threshold2 < threshold3).
threshold3	Value of the third threshold	Optional	This threshold value is used if the number of colors selected for the gauge is >=4 (thresholdNumber >=4). This value must be defined between the min value scale and the next value scale (threshold2 < threshold3 < threshold4).
threshold4	Value of the fourth threshold	Optional	This threshold value is used if the number of colors selected for the gauge is =5 (thresholdNumber >=5). This value must be defined between the min value scale and the next value scale (threshold3 < threshold4 < max).
valueUnit	Indicates the unit of gauge's data	Optional	If blank, there is no unit on the gauge.

Example

For this example, a gauge has been added to the board:



After exporting the board, there is one line in the CSV file:

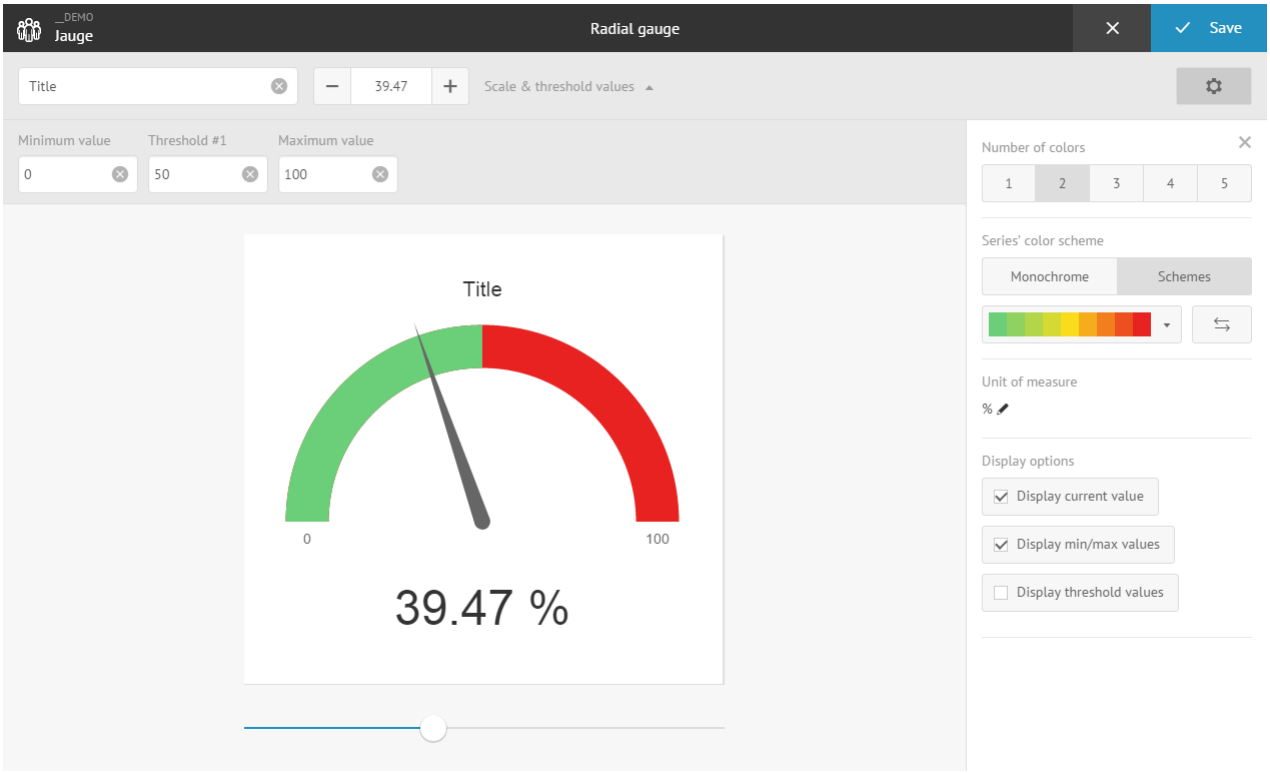
class	color	container	deleted	height	id	invertColors	isAnchored	isMultiColored	kind	label	linkLabel	linkUrl
class com.iobeya.entity.Gauge	298974	<com.iobeya.entity.ElementContainer;9481561e-6c85-4310-b517-b29b8aa6b761>	0	1853	A1609035-7829-D77B-EEED-304DE2CF1EA5	false	false	true	radial	Title		

maxScale	minScale	name	needleValue	setName	showMinMax	showTicks	showValue	themeIndex	threshold1	threshold2	threshold3	threshold4	threshold5	thresholdsN	valueUnit	width	x	y	zOrder
100	0	Radial	39.47	Gauge	true	false	true	1	50	0	0	0	0	1	%	1976	1609	909	7

Note that:

- the current value selected is 39.47: ***needleValue=39.47***.
- the min value of the gauge is 0: ***minScale=0*** and the maximum value of the gauge is 100: ***maxScale=100***.
- there is just one threshold on the gauge: ***thresoldsNumber=1*** and the threshold value is 50: ***threshold1=50***.
- the unit of the gauge is %: **valueUnit=%**.
- min, max and current values are shown on the gauge: ***showMinMax=true*** and ***showValue=true***.
- ranges of thresholds are not shown on the gauge: ***showTicks=false***.
- the gauge references its toolset and the kind of gauge which is radial.
- the parameter color is not used because a color theme is used: ***themeIndex =1***.

See below the configuration screen of this gauge:

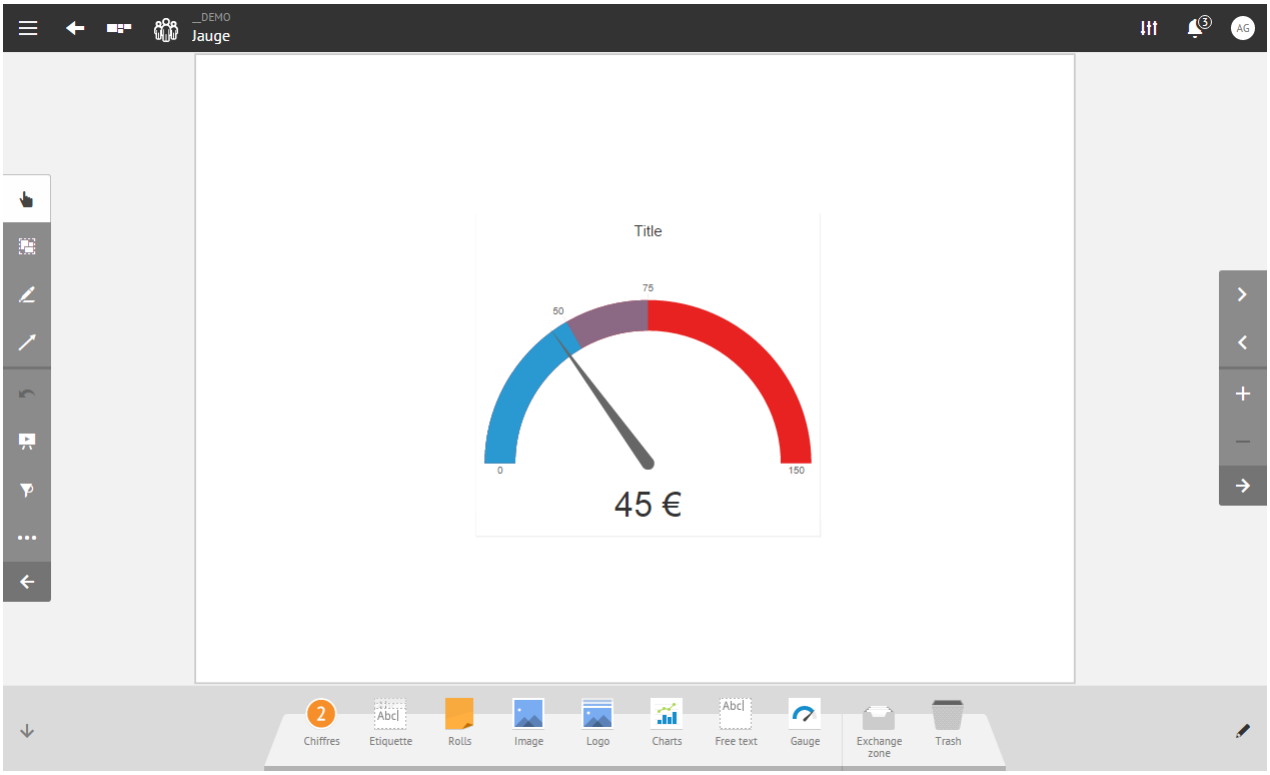


For import, the following changes will be made:

- the current value to 45.
- the number of thresholds to two, and add the new value for the new threshold to 75.
- the display threshold values parameter to true.

maxScale	minScale	name	needleValue	setName	showMinMax	showTicks	showValue	themeIndex	threshold1	threshold2	threshold3	threshold4	threshold5	thresholdsN	valueUnit	width	x	y	zOrder
150	0	Radial	45	Gauge	true	true	true	2	50	75	0	0	0	2	€	1976	1609	909	7

After importing, the result will be as follows:



How to use graph export?

Specific graph fields

Graph export generates several lines:

- the first one represents the object: class com.iobeya.entity.Chart.
- the following lines represent data in the graph.
- the last line represents the image generated from the chart data.

Chart object

Column	Description	Mandatory?	Information
kind	Represents the kind of chart	Optional	See details below
color	Color used	Optional	If isMultiColored=false, the parameter color is used
isMultiColored	Boolean that indicates if the color scheme of the selected graph uses multiple colors or just one color	Optional	If isMultiColored=true, the parameter color is ignored
maxScale	Maximum value of the representation of the graph	Optional	If maxScalePolicy is false, the value of maxScale is used to define the maximum limit to display the graph
maxScalePolicy	Indicates if the max scale is computed automatically or manually	Optional	If maxScalePolicy is true, the value of maxScale is ignored
minScalePolicy	Indicates if the min scale is computed automatically or manually	Optional	If minScalePolicy is true, the value of minScale is ignored
minScale	Minimum value of the graph	Optional	If minScalePolicy is false, the value of maxScale is used to define the minimum limit to display the graph
selectedRow	Number of selected rows when the graph is a pie chart (starting from 0)	Optional	If the graph is not a pie chart, this parameter is ignored
seriesInColumns	Indicates if the series is in columns, in a table, or in rows	Optional	If set to blank, the series is by default in rows. If set to true, the series is in columns
showDataLabels	Boolean that indicates if the labels of series have to appear in the graph	Optional	If true, labels of data are shown on the graph

There are five kind of charts:

- bar chart: **barChart**.
- stacked bar chart: **stackedBarChart**.
- area Chart: **areaChart**.

- line chart: **lineChart**.
- pie chart: **pieChart**.

Graph data

There are three different kinds of lines to represent the data of the graph:

- Rows
- Columns
- Cells

For rows:

Column	Description	Mandatory?	Information
label	Title of the row	Optional	If set to blank, the title of the row is set to "Untitled"
element	Indicates the ID of the chart element that contains this row	Mandatory	
index	Position of the row in the table	Mandatory	Index value begins at 0 (index=0 corresponds to the first line)

For columns:

Column	Description	Mandatory?	Information
label	Title of the column	Optional	If set to blank, the title of the row is set to "Untitled"
element	Indicates the ID of the chart element that contains this column	Mandatory	
index	Position of the column in the table	Mandatory	Index value begins at 0 (index=0 corresponds to the first column)

For Cells:

Column	Description	Mandatory?	Information
x	Row coordinate of the cell in the table of values	Mandatory	x value begins at 0 (x=0 corresponds to the first cell of the first row)
y	Column coordinate of the cell in the table of values	Mandatory	y value begins at 0 (y=0 corresponds to the first cell of the first column)
val	Value of the cell	Optional	If set to blank, the value is set to 0
element	Indicates the ID of the chart element that contains this cell	Mandatory	

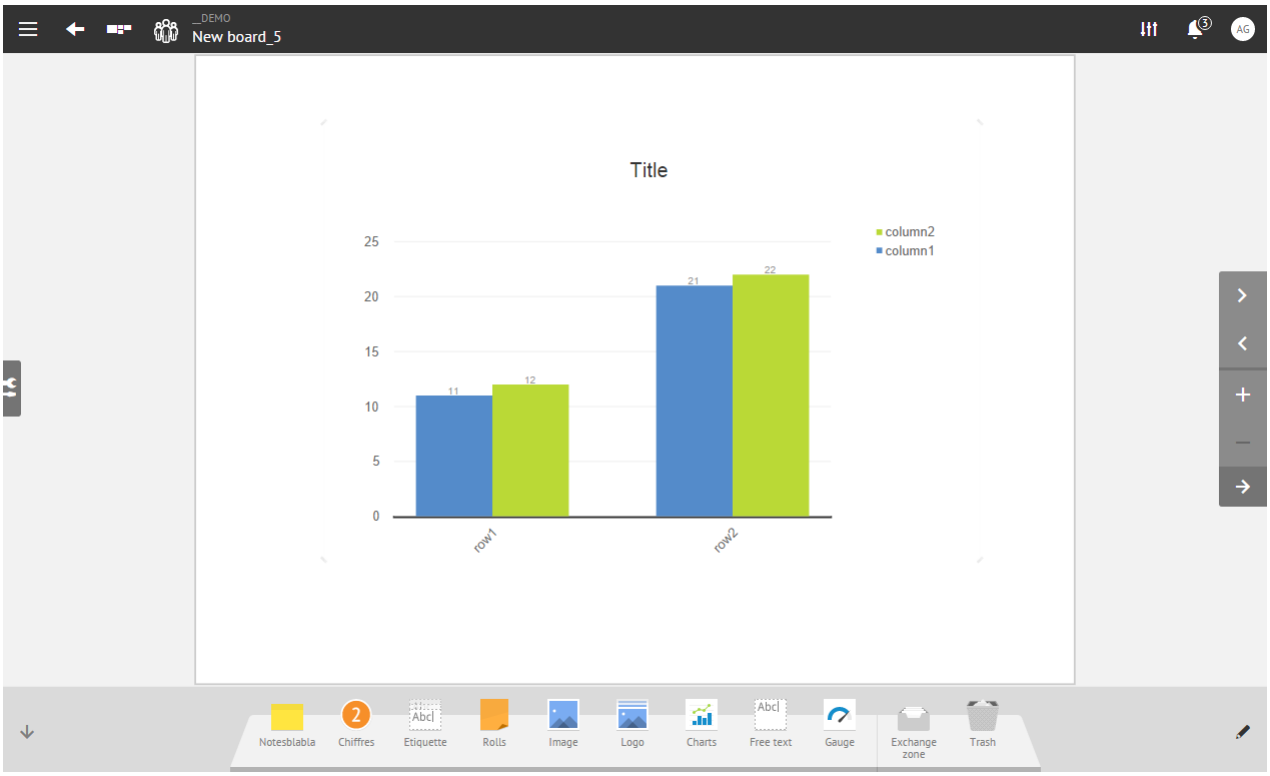
Graph asset

Graph asset is exported for information only. The data of this asset is not used during the import process, because the asset of the graph is automatically generated after the import process.

Column	Description	Mandatory?	Information
mimeType	Image format of the asset	Information	
data	Data of image asset generated	Information	Image (PNG or SVG) file encoded in BASE64 format

Example

For this example, a graph is added to the board:



After exporting the board, there are several lines in the CSV file:

Graph object

The first line below represents the graph object with the overall properties of the graph:

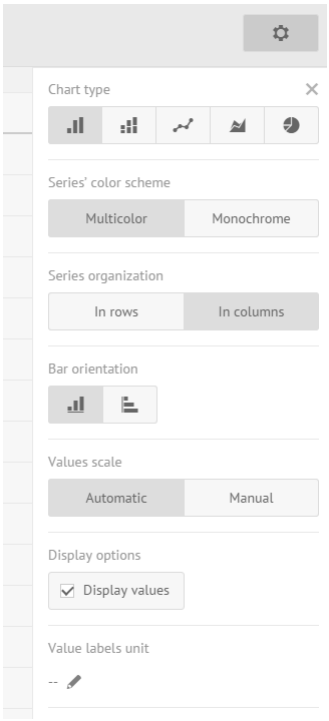
class	asset	color	deleted	height	id	isAnchored	isMultiColored	kind	label
class com.iobeya.entity.Chart	<com.iobeya.entity.Asset;ecb02cc4-7a33-4168-a18b-6c422ea04f88>	298974	0	2517	8F93C33A-F296-058E-B4A6-30494BA1F792	true	true	barChart	Title

maxScale	maxScalePolicy	minScale	minScalePolicy	orientation	selectedRow	seriesInColumns	showDataLabels	width	x	y	zOrder
10	Auto	5	Auto	Vertical	0	true	true	3775	730	377	15

Note that:

- the graph is a bar chart: ***type=barChart***.
- the label of the graph is "Title": ***label=Title***.
- the policy of the scale is automatic (the graph takes the min value and the max value and displays all data between min and max value): ***minScalePolicy=Auto*** and ***maxScalePolicy=Auto***.
- the series are the values of columns (and not rows): ***seriesInColumns=true***.
- the graph is displayed vertically and not horizontally: ***orientation=vertical***.
- the series of the graph can be configured to be displayed with any color of the spectrum: ***isMultiColored=true***.

Below, is the result of this configuration in the settings panel of the graph:



Graph data

The graph generated contains two rows, two columns and four cells.

Following the first object line, there is:

- one line for each row.
- one line for each column.
- one line for each cell.

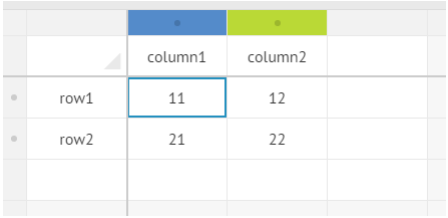
So, there are eight lines to describe the data of this graph:

class	color	element	id	index	label	val	x	y
class com.iobeya.entity.ChartDataRow	0	<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	7A45B330-3F34-FDD8-0DCE-304A46DB34C0	0	row1			
class com.iobeya.entity.ChartDataRow	0	<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	7ACD9666-96B6-2E3E-B15D-304A46DBDD5E	1	row2			
class com.iobeya.entity.ChartDataColumn	12245302	<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	37A933C5-4843-9251-2725-304A46DB2BEB	1	column2			
class com.iobeya.entity.ChartDataColumn	5540810	<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	AABC053A-41C8-A4E6-FEFA-304A46DB8A33	0	column1			
class com.iobeya.entity.ChartDataCell		<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	114F67AC-5D70-E60B-2733-304A46DAFCB0			11	0	0
class com.iobeya.entity.ChartDataCell		<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	120B1605-EA33-F8CC-CCDA-304A46DB646E			12	0	1
class com.iobeya.entity.ChartDataCell		<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	91D7FA9D-8BFD-D89C-F80A-304A46DB2F42			21	1	0
class com.iobeya.entity.ChartDataCell		<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>	DFD96E3D-18EC-C752-C3CE-304A46DBB692			22	1	1

Note that:

- the chart object referenced is the chart with id=8F93C33A-F296-058E-B4A6-30494BA1F792 (chart element's ID property): ***element=<com.iobeya.entity.Chart;8F93C33A-F296-058E-B4A6-30494BA1F792>***.
- the title of the rows are "row1" and "row2": ***label=row1*** and ***label=row2***.
- the title of the columns are "column1" and "column2": ***label=column1*** and ***label=column2***.
- colors are defined for columns and not for rows because the series are organized in columns (seriesInColumns parameter of the graph object).
- the contents of cells are:
  - 11 for row1, column1: ***val=11***, ***x=0*** and ***y=0***.
  - 12 for row1, column2: ***val=12***, ***x=0*** and ***y=1***.
  - 21 for row2, column1: ***val=21***, ***x=1*** and ***y=0***.
  - 22 for row2, column2: **val=22**, ***x=1*** and ***y=1***.

Below the data table of the graph:



How to use image export?

Specific Image and Image gallery fields

Image export generates several lines:

- the first one represents the image object.
- the following ones represent encoded data of this image's asset.

Image from filesystem

Image object:

Column	Description	Mandatory?	Information
fromLibrary	Boolean that indicates if this image has been created from an image gallery	Optional	If ***fromLibrary=true*** then the image has been created from an image gallery
asset	Reference to the image data	Mandatory for image from filesystem	The format of asset reference is the same as for stickers or shapes, ex: **<com.iobeya.entity.Asset;13E82FA4-7F42-4bf3-A75D-EE1B11E961F4>** (see following example and Asset export).

Image data:

Column	Description	Mandatory?	Information
id	ID of the referent image object	Mandatory	An image is formed with multiple image lots
data	Image data batch	Mandatory	PNG file encoded in BASE64 format
mimeType	Image format	Information	Automatically retrieved during the import
assetWidth	Width of the image	Information	Automatically retrieved during the import
assetHeight	Height of the image	Information	Automatically retrieved during the import
dataLocation	Storage type	Information	Automatically retrieved during the import
filesystem	Boolean that indicates if the element is stored in filesystem	Information	Automatically updated during the import
database	Boolean that indicates if the element is stored in database	Information	Automatically updated during the import

Image from an image gallery

Parameters of an image from an image gallery are exactly the same as those of a simple image, except for the below two parameters:

Column	Description	Mandatory?	Information
name	Name of the element in its toolset. Used only for image from image gallery to link this image to the image with the same name in the toolset.	Optional	A blank value will lead to an unknown tool

Column	Description	Mandatory?	Information
setName	Toolset name. Used only for image from image gallery to link this image to the toolset with the same name.	Optional	If blank, it will not match any known toolset and legend will track it as unknown

Example with image from filesystem

For this example, an image has been uploaded to the board from a filesystem using the image tool:

After exporting, a CSV file with several lines is generated:

Image object:

The first line below represents the image object with its own properties:

asset	container	fromLibrary	height	id	isAnchored	linkLabel	linkUrl	width	x	y	zOrder
<com.iobeya.entity.Asset;B992A970-C4CA-9572-4C4E-31C382EBAB83>	<com.iobeya.entity.ElementContainer;f2eb038e-6ca6-4f14-86ad-9bc2b4e67ad8>	false	853	434712FE-07B3-4826-6894-31C385A094C0	false			1720	1577	1231	8

Note that:

- the size of the image on the board is 1720x853: ***width=1720*** and ***height=853***.
- the position of the image on the board is at 1577 pixels from the left and 1231 pixels from the top: ***x=1577*** and ***y=1231***.
- this image is not locked on the board: ***isAnchored=false***.
- there is no hyperlink on the image: ***linkLabel*** and ***linkUrl*** are empty.
- this image is not an image from an image gallery: ***fromLibrary = false***.

Image data:

The four following lines represent the data of the asset. These lines will recreate the image asset during an import:

class	id	assetWidth	assetHeight	container	data	dataType	database	fileSystem	isSystem	maxRatio	mimeType
class com.iobeya.entity.Asset	B992A970-C4CA-9572-4C4E-31C382EBAB83_1-4	480	238	<com.iobeya.entity.AssetContainer;45e8b11b-3e17-414a-97ef-e09c2874b903>	iVBORw0KGgoAAA	filesystem	false	true	false	4	image/png
class com.iobeya.entity.Asset	B992A970-C4CA-9572-4C4E-31C382EBAB83_2-4	480	238	<com.iobeya.entity.AssetContainer;45e8b11b-3e17-414a-97ef-e09c2874b903>	g031IJXSrl278jvvvPf	filesystem	false	true	false	4	image/png
class com.iobeya.entity.Asset	B992A970-C4CA-9572-4C4E-31C382EBAB83_3-4	480	238	<com.iobeya.entity.AssetContainer;45e8b11b-3e17-414a-97ef-e09c2874b903>	yfr888-_yszMfNFTu	filesystem	false	true	false	4	image/png
class com.iobeya.entity.Asset	B992A970-C4CA-9572-4C4E-31C382EBAB83_4-4	480	238	<com.iobeya.entity.AssetContainer;45e8b11b-3e17-414a-97ef-e09c2874b903>	FzqWh_hWVFSsrKq	filesystem	false	true	false	4	image/png

Here, there are four lines that represent four parts of the asset's data (split into separate parts to overcome character limitation per cell in Excel):

Below, you can find the first part of data base64 encoded:

	A	B	C	D	E	F	G	H	I	J	K	L
1	class	id	assetWidth	assetHeight	container	data	dataType	database	fileSystem	isSystem	maxRatio	mimeType
2	class com.iobeya.entity.Asset	B992A970-C4CA-9572-4C4E-31C382EBAB83_1-4	480	238	<com.iobeya.entity.AssetContainer;45e8b11b-3e17-414a-97ef-e09c2874b903>	iVBORw0KGgoAAAA	filesystem	false	true	false	4	image/png

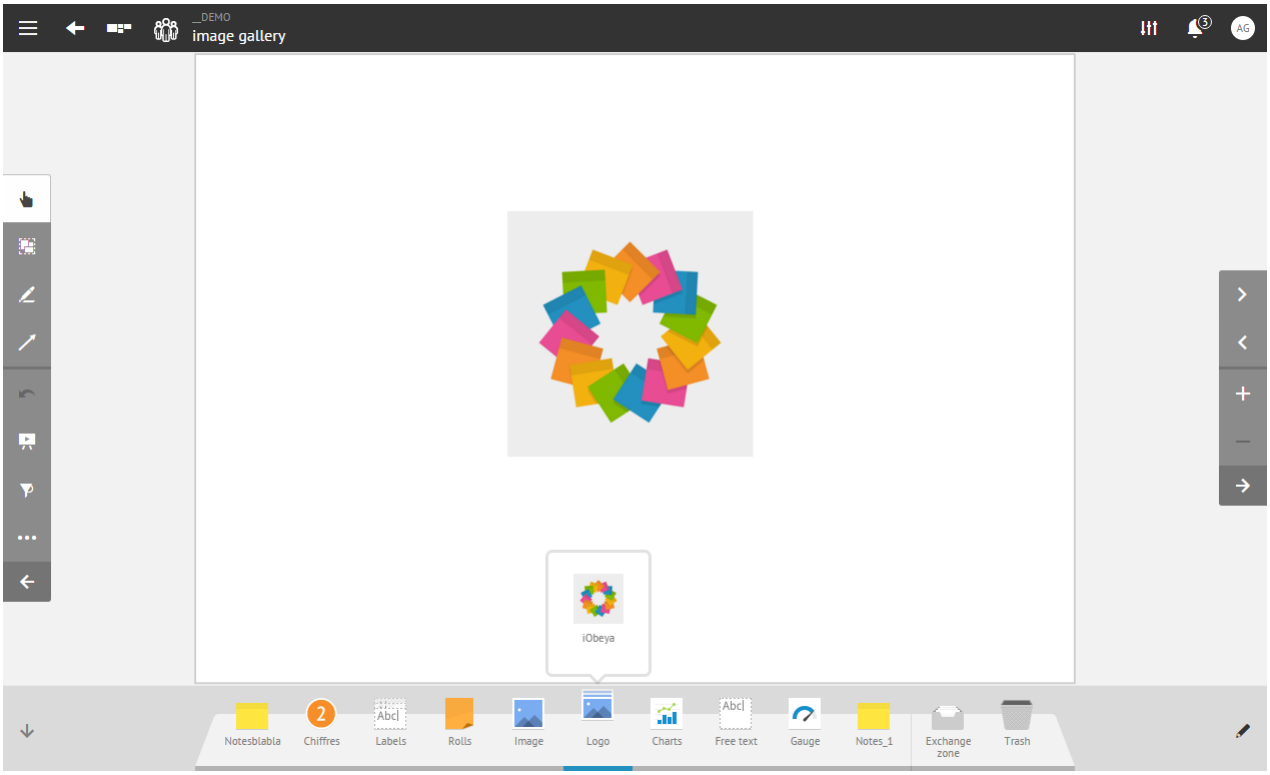
```
iVBORw0KGgoAAAANSUhEUgAAAeAAAADuCAYAAADhqdRRAAAACXBIWXMAAA7EAAAOxAGVKw4bAAAgAElEQVR4nOydd3wVVfr
```

Note that:

- this asset references the image object for id =0 B992A970-C4CA-9572-4C4E-31C382EBAB83: ***id=B992A970-C4CA-9572-4C4E-31C382EBAB83_1-4***.
- the size of the asset uploaded is 480*238 pixels: ***assetWidth=480*** and ***assetHeight=238***.
- the asset is a PNG file: ***mimeType=image/png***.
- the asset was exported from a filesystem: ***dataLocation=fileSystem***.
- it is possible to have 4 different sizes of this asset to be used in different contexts: ***maxRatio=4***.

Example with image from an image gallery

For this example, an image has been added to the board from an image gallery for a room:



An image export from a gallery is the same as the export of an image uploaded from a filesystem, with the exception that the image object references its own image gallery:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	asset	class	container	deleted	fromLibrary	height	id	isAnchored	linkLabel	linkUrl	name	setName	width	x	y	zOrder
2	<com.iobeya.entity.Asset;9F60B71B-9E52-5454-1E81-3587D7934417>	class com.iobeya.entity.BoardImage	<com.iobeya.entity.Element	0	true	1410	3D16AA38-C1CC-3261-376E-358800C8A2BA	false			iObeya	Logo	1410	1788	896	3

Note that:

- this image is an image from a gallery: fromLibrary=true.
- this image references the tool iObeya in the image gallery named Logo.

How to use team export?

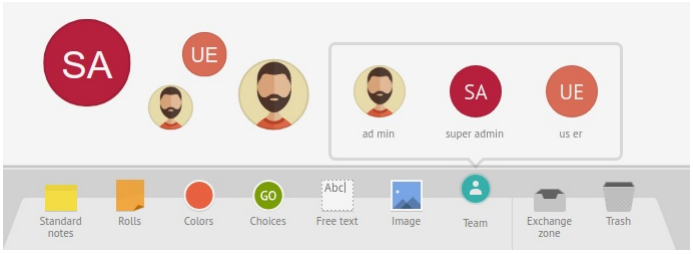
Specific team fields

> ***Important:*** **To match the element with an existing tool, the corresponding value for the following must be set: color, name, setName and username.**

Column	Description	Mandatory?	Information
username	User username	Mandatory	
avatar	Asset for user avatar	Optional	

Example

For this example we simply added four team members to the board, two of them being of the same user:



Exporting these four elements gives this CSV file:

	A	B	C	D	E	F	G	H	I	J
1	class	id	avatar	username	name	assetHeight	assetWidth	width	height	x
2	class com.iobeya.entity.Team	6681E1EE-C318-4E4A-A634-B467B3A0B645	<com.iobeya.entity.Asset;50997a5c-f4fc-4f7f-b2bf-486b871e2264>	admin	ad min			200	200	16
3	class com.iobeya.entity.Team	E0F615A9-59BB-9182-CD20-B467BAEA10A4	<com.iobeya.entity.Asset;170dd158-a872-4fff-b892-33fba14dfc57>	superadmin	super admin			389	389	12
4	class com.iobeya.entity.Team	4A36C0A9-C669-D421-E2B3-B4756CDF55A7	<com.iobeya.entity.Asset;50997a5c-f4fc-4f7f-b2bf-486b871e2264>	admin	ad min			314	314	20
5	class com.iobeya.entity.Team	0D8AD75D-E8E6-EE73-ED0F-B467C1359295	<com.iobeya.entity.Asset;09d3de60-157b-47f9-b9df-be3614534424>	user	us er			200	200	18
6	class com.iobeya.entity.Asset	170dd158-a872-4fff-b892-33fba14dfc57			super admin	0	0			
7	class com.iobeya.entity.Asset	09d3de60-157b-47f9-b9df-be3614534424			us er	0	0			
8	class com.iobeya.entity.Asset	50997a5c-f4fc-4f7f-b2bf-486b871e2264_1-3			ad min	138	138			
9	class com.iobeya.entity.Asset	50997a5c-f4fc-4f7f-b2bf-486b871e2264_2-3			ad min	138	138			
10	class com.iobeya.entity.Asset	50997a5c-f4fc-4f7f-b2bf-486b871e2264_3-3			ad min	138	138			
11										

D - Special cases

This section presents some special cases in which import/export behavior can change to ease usage or prevent errors:

- "Export notes" specific fields.
- Asset export.
- Synchronized elements.

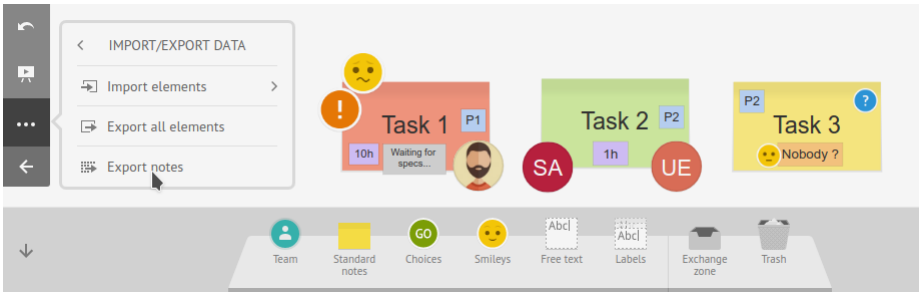
Export notes specific fields

When using the "Export notes" (or "Export selected notes") action, additional informational columns can be added to the CSV export of each note, depending on the board elements that are put on them. Only some of these such elements add columns, and those ones are described in the table below:

Element class	Column	Values
com.iobeya.entity.Label	Variable: the name of each label toolset item	The text content of the label elements on the note
com.iobeya.entity.Freetext	**Freetext**	The text content of the freetext elements on the note
com.iobeya.entity.Team	**members**	The usernames of the team member elements on the note
com.iobeya.entity.Sticker	Variable: the name of each sticker toolset	The names of the toolset items on the note

Example

For this example, we created three notes of different colors and put stickers, team members, labels and freetexts on them:



Exporting the notes gives us a CSV file whose content looks like this:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	class	entityType	name	contentLabel	Freetext	Smileys	Duration	Choices	Project	members	layoutId	width	heigh
2	class com.iobeya.entity.Note	BoardNote	Yellow	Task 3	Nobody ?	No voice		Question	P2		note-layout-1	375	2
3	class com.iobeya.entity.Note	BoardNote	Green	Task 2			1h		P2	superadmin,user	note-layout-212	375	2
4	class com.iobeya.entity.Note	BoardNote	Red	Task 1	Waiting for specs...	Worried	10h	Warning	P1	admin	standard	375	2
5													

We can, for example, see that the note with contentLabel "Task 1" holds a freetext containing "Waiting for specs...", a "Smiley/Worried" sticker, a "Duration" (name of the purple label item within the "Labels" toolset) label containing "10h", a "Choices/Warning" sticker, a "Project" (name of the blue label item within the "Labels" toolset) label containing "P1" and a team member whose username is "admin".

Asset export

When some elements using assets are exported, the assets they reference are automatically exported too. The format of asset reference is a combination of the asset class full name and its id, e.g.: **<com.iobeya.entity.Asset;13E82FA4-7F42-4bf3-A75D-EE1B11E961F4>**.

The data they contain is encoded in base64 in a column named **data**. Given this data may be longer than the maximum size of a cell in a spreadsheet like MS Excel or LibreOffice Calc, an asset could be split into several CSV rows, where its id will be suffixed with a 1-based index and the number of asset parts like this: "<ID>_<INDEX>-<NUMBER_OF_PARTS>". For example, an asset with id **50997a5c-f4fc-4f7f-b2bf-486b871e2264** could be split into three rows, all identical, except for the columns **id** and **data**. The **data** would simply be split into three and distributed over these rows, while the ids would be:

- **50997a5c-f4fc-4f7f-b2bf-486b871e2264_1-3**.
- **50997a5c-f4fc-4f7f-b2bf-486b871e2264_2-3**.
- **50997a5c-f4fc-4f7f-b2bf-486b871e2264_3-3**.

Synchronized elements

Element synchronization is preserved as most as possible during CSV import/export. This means that as long as an element synchronized with the exported element is still in the platform, the exported element would be re-synchronized with it on import (and synchronizable values of imported element would be replaced by values of element in platform, unless the imported element is already on the board).

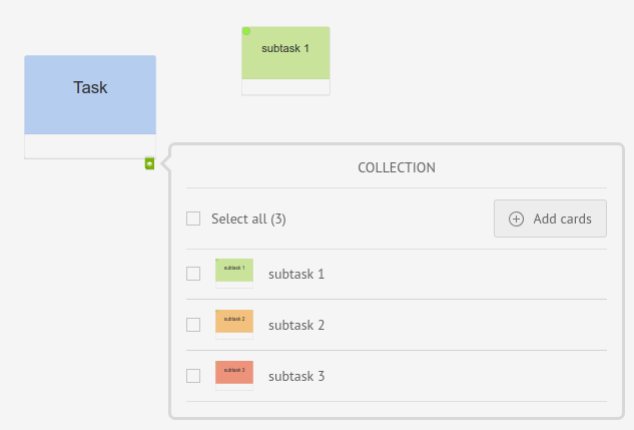
In order to understand CSV export of synchronized elements and collections, we need to precise that synchronization is ensured by a server-side board element called *master* which is referenced by synchronized elements through an object of class `com.iobeya.entity.ElementSyncInfo` and by collection items through objects of class `com.iobeya.entity.SyncElementReference`.

These master elements are identified in CSV by a value of `true` in the `master` column. Existing master elements cannot be directly modified during CSV import: this is to avoid hidden modifications during import of elements on a board. Also collection items do not exist by themselves and are only references to a master element. As such, they will either reference an existing master, ignoring any modification made to the master element in CSV, or reference a new master created from the master element in CSV.

In the case where an imported synchronized element is already in the target board, the element will be updated with values from CSV, and the synchronizable values will be propagated to the other synchronized instances. Else -- as described earlier -- the synchronized values of imported element will be set to the same values as other synchronized instances if there are, or as given in CSV if there isn't any.

Example

Here we take the example of exporting/importing a card with its collection as in the following screenshot:



Exporting the cards `Task` and `subtask 1` in CSV gives the following file (with reordered rows):

class	entityType	id	label0	element	boardElement	masterElement	syncInfo	itemElementRef	weakRef	master	x	y
class com.iobeya.entity.Card	BoardCard	aa034e38-0d1d-4d71-ad11-a2e11387d3da	subtask 1							true	0	0
class com.iobeya.entity.Card	BoardCard	bc61fb66-1c99-4557-8a9e-601e7ab02586	subtask 2							true	0	0
class com.iobeya.entity.Card	BoardCard	cf0dc9b5-2b99-4014-a3c1-3b9427af84f5	subtask 3							true	0	0
class com.iobeya.entity.Card	BoardCard	428E86B1-DFB7-8FB8-1D46-CAB2845D26A9	Task							false	745	698
class com.iobeya.entity.ElementSyncInfo		aa034e38-0d1d-4d71-ad11-a2e11387d3da				<com.iobeya.entity.Card;aa034e38-0d1d-4d71-ad11-a2e11387d3da>						
class com.iobeya.entity.Card	BoardCard	AF163296-BAC2-56D2-6A32-CAB29DCFCC40	subtask 1				<com.iobeya.entity.ElementSyncInfo;aa034e38-0d1d-4d71-ad11-a2e11387d3da>			false	1365	616
class com.iobeya.entity.SyncElementReference	CollectionItemElementRef	ad265ae9-6245-470b-9161-8a1d4df569f9			<com.iobeya.entity.Card;aa034e38-0d1d-4d71-ad11-a2e11387d3da>				false			
class com.iobeya.entity.SyncElementReference	CollectionItemElementRef	d87258db-4c7b-4894-a992-c9406c2955ab			<com.iobeya.entity.Card;bc61fb66-1c99-4557-8a9e-601e7ab02586>				false			
class com.iobeya.entity.SyncElementReference	CollectionItemElementRef	4d27b260-a660-4ec6-bf9e-c3ca1b9a687d			<com.iobeya.entity.Card;cf0dc9b5-2b99-4014-a3c1-3b9427af84f5>				false			
class com.iobeya.entity.BoardElementCollectionItem		43E8C7FC-5309-831E-5CC6-CAB357426F0B		<com.iobeya.entity.Card;428E86B1-DFB7-8FB8-1D46-CAB2845D26A9>				<com.iobeya.entity.SyncElementReference;ad265ae9-6245-470b-9161-8a1d4df569f9>				
class com.iobeya.entity.BoardElementCollectionItem		5D5645C8-D368-81EC-E649-CAB3574209C2		<com.iobeya.entity.Card;428E86B1-DFB7-8FB8-1D46-CAB2845D26A9>				<com.iobeya.entity.SyncElementReference;d87258db-4c7b-4894-a992-c9406c2955ab>				
class com.iobeya.entity.BoardElementCollectionItem		9895B95A-D5CB-AD9E-C1B1-CAB35743AF42		<com.iobeya.entity.Card;428E86B1-DFB7-8FB8-1D46-CAB2845D26A9>				<com.iobeya.entity.SyncElementReference;4d27b260-a660-4ec6-bf9e-c3ca1b9a687d>				

As we can see, the master cards (cards with value `true` in column `master`) are coming first (to be referenced by following objects). They are similar to other cards, excepted for the `master` column (`true`), and the `container` column (blank).

Next is the `Task` card, which has no particularity even though it carries a collection. Indeed, it does not need to reference the collection items, instead, the collection items themselves reference the card.

Next two rows describe the card instance of `subtask 1` on the board, the second row is a usual card, excepted it references the object of class `com.iobeya.entity.ElementSyncInfo` of the first of these rows in the column `syncInfo`. This `ElementSyncInfo` references the master card corresponding to `subtask 1`.

The last 6 rows represent the collection attached to the `Task` card.

The first 3 of these ones are objects of class `com.iobeya.entity.SyncElementReference` which are used to reference the masters corresponding to elements in collection. Masters are referenced by column `boardElement`. `SyncElementReference` is an extensible entity with default value for `entityType` being `CollectionItemElementRef`.

The last 3 of these ones are the collection items referencing:

- the board element to which they are attached (in column `element`),
- the reference to the board element they represent (in column `itemElemRef`).

E - Board specific fields

Import/export default behavior can be modified depending on the board kind on which elements are positioned:

- on export, additional data can appear as information or board context.
- on import, additional data can be taken into account to preserve any original contextual information within the target board, when meaningful.
- *Additional data with the planning board*.

Additional data with the planning board

When elements from a planning board are exported, new column values are specified if elements are positioned on the schedule. (rows, date, startDate and endDate)

Column	Description
rows	Describe for each element the row(s) it belongs to
date	The specific date of a non-resizable element
startDate	The start date of a resizable element
endDate	The end date of a resizable element

Import process

- the *rows* value just shows the row names associated with the element. This value only provides information to the user. It will not be used during import operations.
- the *date* / *startDate* / *endDate*:
  - if the date is specified and the targeted board is a planning board containing this date, the element will be positioned at this date on the board.
  - if the date is specified and the targeted board is a planning board excluding this date, the element will be placed in the trash.
  - if the date is specified and the targeted board is a non-planning board, the element will be positioned due to its x coordinate.
  - the same logic as above applies with the start date and the end date of resizable elements.