

Provisional Upload Procedure to Celonis OCDM

Due to recent changes breaking compatibility, the uploader from OCEL to Celonis OCDM is not working anymore. This document specifies a provisional upload procedure, waiting for automated support by the new APIs.

The first step is about downloading the file **splitter.py**:

<https://github.com/Javert899/ocel20-celonis-connector/blob/main/splitter.py>

And modifying the bottom part (just after `__main__`) to ingest your OCEL 1.0 / 2.0 object-centric event log using pm4py. Executing the file, a collection of CSVs is stored in the folder “target”, which are the base for the next steps of the upload procedure.





For a minimal example, we loaded the OCEL 1.0 event log available at:

https://github.com/pm4py/pm4py-core/blob/release/tests/input_data/ocel/example_log.jsonocel

and filtered that on a minimal configuration of object and event types:

- Object types: order, element
- Event types: Create Order
- Relationships:
 - (Create Order, order) 1:1
 - (Create Order, element) 1:N

Executing the script, we get the following files inside the “target” folder:

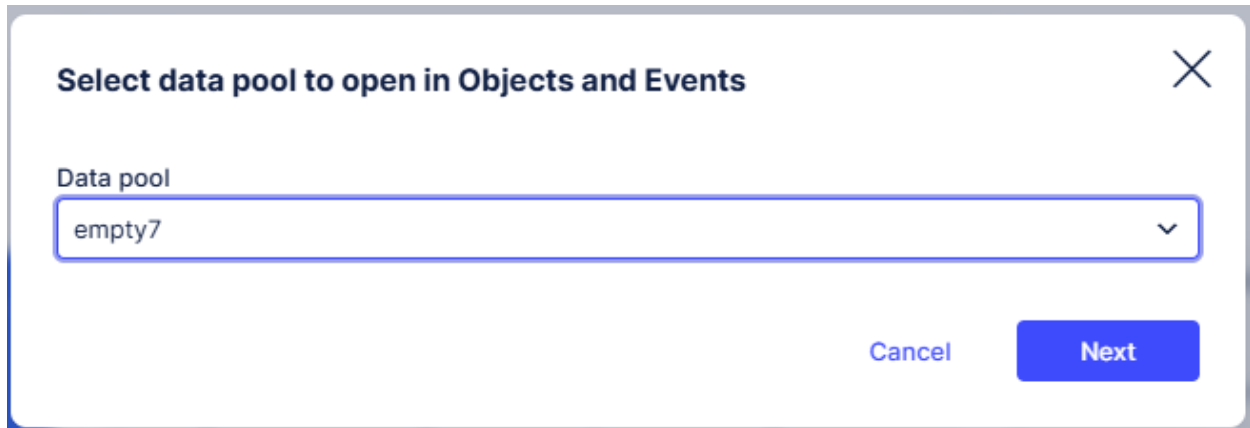
Name	Date modified	Type	Size
 CreateOrder_Element_relations.sql	11/20/2024 10:16 AM	SQL Source File	1 KB
 CreateOrder_events.sql	11/20/2024 10:16 AM	SQL Source File	1 KB
 Element_objects.sql	11/20/2024 10:16 AM	SQL Source File	1 KB
 Order_objects.sql	11/20/2024 10:16 AM	SQL Source File	1 KB

We see the two object types stored as SQL files, the activity stored as SQL file, and the (potentially) many-to-many relationships stored in a separate “relations” SQL file.

The next steps are done inside Celonis.

First, create an empty data pool (in our case, “empty7”).

Then, reach the “Objects & Events” feature



The screenshot shows a dialog box with the title "Select data pool to open in Objects and Events" and a close button (X) in the top right corner. Below the title is a label "Data pool" and a dropdown menu containing the text "empty7" and a downward arrow. At the bottom right of the dialog, there are two buttons: "Cancel" and "Next".

empty7 | Objects and Events > Dashboard

Search (Ctrl + /)

Overview

- 0 Object Types
- 0 Event Types
- 0 Perspectives
- 0 Transformations

Catalog Processes (0) | [View Catalog](#)

[+](#)
Add from catalog

Documentation & Help

Documentation for OCPM

- [Overview](#)

click on “Object Types” and start the creation of an object type.

Let’s start by “Order”, adding the additional attributes that are associated to an “Order” in the given OCEL.

Create object type

Name

Color

Description Optional

^ Tags and categories

Custom Processes (0)

Catalog Processes (0)

Metadata (0)

^ Attributes (3)

[+ Add](#)

Name	Data Type	
<input type="text" value="Oattr2"/>	<input type="text" value="Floating point"/>	
<input type="text" value="Oattr1"/>	<input type="text" value="String"/>	
ID	String	

Cancel

Save

To “inject” the actual objects, reach the “Transformations” page by clicking the corresponding button on the right of the page. Proceed to “Add transformation” and copy-paste inside the SQL instructions. Click “Save” and then “Preview”.

```
1 SELECT
2   'o1' AS "ID",
3   'uno' AS "Oattr1",
4   1.0 AS "Oattr2"
5 FROM (SELECT 1) AS dummy
6 WHERE 1=1
7
8 UNION ALL
9
10 SELECT
11   'o2' AS "ID",
12   NULL AS "Oattr1",
13   NULL AS "Oattr2"
14 FROM (SELECT 1) AS dummy
15 WHERE 1=1
16
17 UNION ALL
18
19 SELECT
20   'o3' AS "ID",
21   NULL AS "Oattr1",
22   NULL AS "Oattr2"
23 FROM (SELECT 1) AS dummy
24 WHERE 1=1
```

STRING	ID	STRING	Oattr1	STRING	Oattr2
	o1		uno		1
	o2		<null>		<null>
	o3		<null>		<null>

Then, come back to the “Object Types” page of the “Objects and Events” feature. Let’s proceed to add the other object type, “Element”.

Create object type

Name: Element Color

Description: Optional

Tags and categories

Custom Processes (0)

Catalog Processes (0)

Metadata (0)

Attributes (3) + Add

Name	Data Type
Oattr2	Floating point
Oattr1	String
ID	String

Cancel Save

After creating the attributes, let's add the transformation in a similar way to before.

The screenshot shows a data transformation tool interface. At the top, there is a tab labeled "ElementAttributesX". Below the tab is a text area containing SQL code. The code consists of three SELECT statements, each followed by a UNION ALL, resulting in a single large query. The first SELECT statement (lines 46-51) has '16' as the ID, 'due' as Oattr1, and 2 as Oattr2. The second SELECT statement (lines 56-61) has '17' as the ID, 'tre' as Oattr1, and 3 as Oattr2. The third SELECT statement (lines 65-70) has '18' as the ID, 'quattro' as Oattr1, and 4 as Oattr2. All other Oattr1 and Oattr2 values are NULL.

Below the SQL editor is a "Preview" section. It contains a table with three columns: "ID" (type STRING), "Oattr1" (type STRING), and "Oattr2" (type FLOAT). The table has 7 rows, corresponding to the data generated by the SQL query.

(STRING) ID	(STRING) Oattr1	(FLOAT) Oattr2
i1	due	2
i2	tre	3
i3	quattro	4
i4	<null>	<null>
i5	<null>	<null>
i6	<null>	<null>
i7	<null>	<null>

Now, reach the "Event Types" part. Let's add the event named "CreateOrder".

CreateOrder

[View](#)

Created 11/20/2024 10:45 Modified 11/20/2024 10:45 Namespace custom

Description Optional

Tags and categories

Custom Processes (0)

Catalog Processes (0)

Attributes (4)

[+ Add](#)

Name	Data Type	
Prova2	Floating point	
Prova	String	
ID	String	
Time	Datetime	

The next step is about adding the relationships to objects. Remind that this event type has 1:1 relation with Order, and 1:N relationship with Element. Use the “Relationships to objects” to add the two relationships.

Add Related Objects

CreateOrder

Involves one Involves many

Object

Relationship name

Add Related Objects

CreateOrder

Involves one Involves many

Object

Element ▼

Relationship name

+ Add another object

Then, reach the “Transformation” page and click “Add transformations”.

The first transformation that you need to add is the content of the event table, which also contains the “Order” as column (so, the 1:1 relationship is defined). Copy-paste the corresponding content of the SQL file, save and preview.

CreateOrderAttributes X

```

6      'o1' AS "Order"
7      FROM (SELECT 1) AS dummy
8      WHERE 1=1
9
10     UNION ALL
11
12     SELECT
13         'e11' AS "ID",
14         TIMESTAMP '1981-01-01 00:00:00' AS "Time",
15         NULL AS "Prova",
16         NULL AS "Prova2",
17         'o2' AS "Order"
18     FROM (SELECT 1) AS dummy
19     WHERE 1=1
20
21     UNION ALL
22
23     SELECT
24         'e14' AS "ID",
25         TIMESTAMP '1981-01-04 00:00:00' AS "Time",
26         NULL AS "Prova",
27         NULL AS "Prova2",
28         'o3' AS "Order"
29     FROM (SELECT 1) AS dummy
30     WHERE 1=1

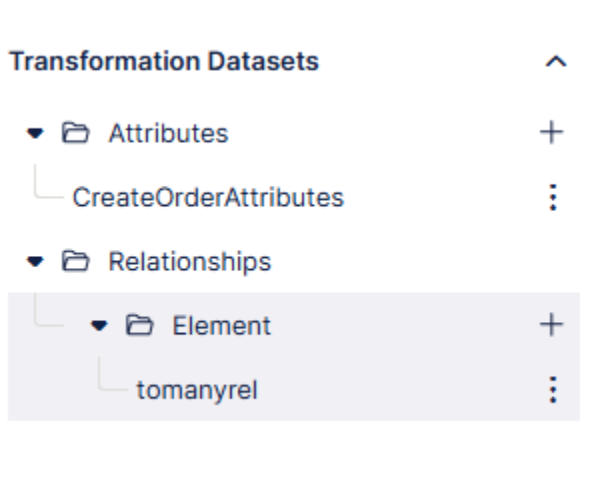
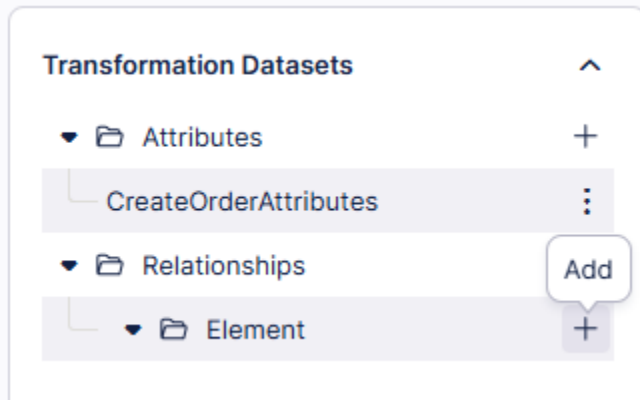
```

=

Preview + Edit attributes Preview

(STRING) ID	(DATETIME) Time	(STRING) Prova	(FLOAT) Prova2	(STRING) Order_ID
e1	1980-01-01 00:00:00	ciao	456	o1
e11	1981-01-01 00:00:00	<null>	<null>	o2
e14	1981-01-04 00:00:00	<null>	<null>	o3


The second transformation that you need to add is the 1:N relationship with Element. For this click the add button near “Relationships – Element” on the top left part, and give a name to the relationship:



Then, copypaste the corresponding content of the “relationship” SQL file, save, and preview.

CreateOrderAttributes X tomanyrel X

```
1 SELECT
2   '14' AS "Element",
3   'e1' AS "ID"
4 FROM (SELECT 1) AS dummy
5 WHERE 1=1
6
7 UNION ALL
8
9 SELECT
10  '11' AS "Element",
11  'e1' AS "ID"
12 FROM (SELECT 1) AS dummy
13 WHERE 1=1
14
15 UNION ALL
16
17 SELECT
18  '13' AS "Element",
19  'e1' AS "ID"
20 FROM (SELECT 1) AS dummy
21 WHERE 1=1
22
23 UNION ALL
24
25 SELECT
```

Preview  Preview

(STRING) Element_ID	(STRING) ID
14	e1
11	e1
13	e1
12	e1
16	e11
15	e11
18	e14

The data modeling part concludes with the creation of an analysis perspective. Reach the “Perspectives” section of the “Objects and Events” feature, click “Create Perspective”, and add the two object types.

saddsads

Objects Event logs

Search

Included in the perspective (2) ^

Element custom ⋮

Order custom ⋮

All objects (0) ^

⚠ Not all objects are connected. Data permissions and filters may not apply to every object.

Element

Order

And then, click on “Save”.

Then, click “Publish” on the top right corner, and select “Publish to development and production”.

Dashboard Objects Events **Perspectives** Transformations

Development ▾ Publish ▾

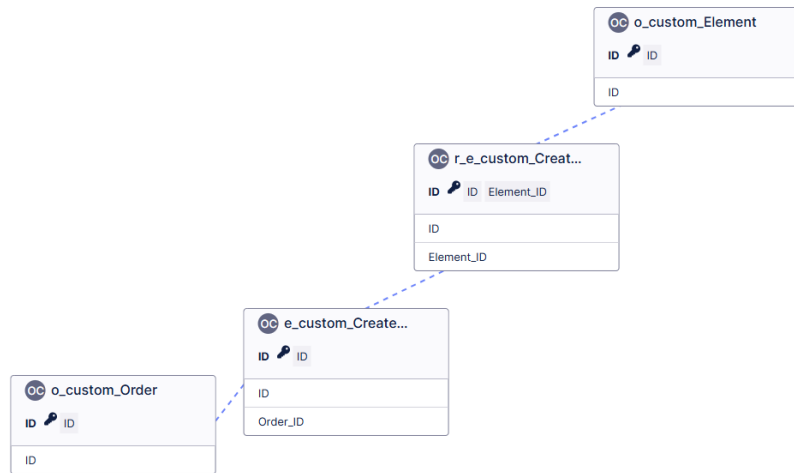
Publish to development

Publish to development and production

When the publishing is done, reach the “Data – Data Integration” component of Celonis, click on the data pool, reach the data models, and click on the data name starting with “perspective_”:

Data Models ▾ + Add Data Model ×

Name	Status	Loaded rows	Created By	Last Edited By	Last Edit	Last Execution	Data Permissions	Multi-Event Log
perspective_custom_saddsads	✔		Javer1899	Javer1899	2024-11-20 10:54:36		Inactive	No
test:perspective_custom_saddsads	✔		Javer1899	Javer1899	2024-11-20 10:54:06		Inactive	No



At this point, trigger the load of the data model. Then, the data model is ready for usage in the “Studio” component of Celonis.