

# XML data property file - definition

## Description

The Data Intelligence XML Data profile file provides the required information to:

- Read the information out of the data sent to the Data Intelligence (typically in a archive zip file containing several CSV files)
- Generate the required properties out of the data provided (e.g.: product titles comes from file A.csv, French label is in column "title\_fr" and English label is in column "title\_en")
- Prepare all the account configuration that any sending of the data will be automatically processed and loaded in Boxalino Intelligent Engine

A full example is provided here in attachment of this page (of the XML file as well as of demo CSV files in a zip archive which follow the specification of the XML file).

- [properties.xml](#)
- [data.zip](#)

Here is the base structure of the XML file:

```
<?xml version="1.0" encoding="UTF-8"?>

<root>

  <languages>

    ...

  </languages>

  <containers>

    <container id="products" type="products">

      <sources>

        ...

      </sources>

      <properties>

        ...

      </properties>

    </container>

    ...

  </containers>

</root>
```

## <root>

The <root> tag is required but is only a container for the inner elements.

#Attributes: none

#Children tags (\* required, + at least one)

- [languages\\*](#)
- [containers\\*](#)

## <languages>

The <languages> tag is required but is only a container for the inner elements.

#Attributes: none

#Children tags (\* required, + at least one)

- [language+](#)

Example:

```
<languages>
```

```
<language id="fr" />
<language id="de" />
</languages>
```

## <language>

The <language> tag indicate a language the must be managed by the system.

#Attributes:

- **id**: a unique identifier for the language, must be one of the following values:
  - en (for English)
  - fr (for French)
  - de (for German)
  - it (for Italian)
  - es (for Spanish)
  - zh (for Chinese)
  - cz (for Czech)
  - ru (for Russian)

#Children tags (\* required, + at least one): none

NB: For other languages, it is possible, please contact with Boxalino Customer Success team.

## <containers>

The <containers> tag is required but is only a container for the inner elements.

#Attributes: none

#Children tags (\* required, + at least one)

- container+

## <container>

The <container> tag describes the data configuration for a specific type of data.

Typically, there is up to 3 containers, each with a different type.

#Attributes:

- **id**: a unique identifier for the container
- **type**: the type of the container, must be one of the following values
  - products (for products and content-like data)
  - customers (for customer data)
  - transactions (for transaction logs defining what products what bought when by which customer)

#Children tags (\* required, + at least one)

- sources\*
- properties\*

Example:

```
<container id="products" type="products">
```

...

```
</container>
```

## <sources>

The <sources> tag is required but is only a container for the inner elements.

#Attributes: none

#Children tags (\* required, + at least one)

- sources+

## <source>

The <source> tag define a source of data (typically one CSV file).

#Attributes:

- **id**: a unique identifier for this source
- **type**: the type of the source, must be one of the following values:
  - **item\_data\_file** (this is the typical case where the CSV contains one columns indicating the item id and other columns providing information about the item)
  - **combined\_options** (this is similar to **item\_data\_file**, but for options combination typically for a product; e.g.: different colors and sizes linked with different prices and stock status, with possibly many rows for one single product)
  - **hierarchical** (this is to describe a global hierarchical resource, such as a category tree)
  - **resource** (this is similar to hierarchical, but for flat resources, such as a global mapping between ids and labels; e.g.: mapping between color codes and labels in different languages)

#Children tags (\* required, + at least one) :

FOR ALL TYPES OF SOURCE

- **file\***
  - required attribute **value** to define: the file name with extension (e.g.: "item\_properties.csv")
  - no children
- **format\***
  - required attribute **value** to define: the file format. Suggested value: **CSV**, for other values, please contact Boxalino Customer Success Team
  - no children
- **encoding\***
  - required attribute **value** to define: the file encoding. Suggested value: **UTF-8**, for other values, please contact Boxalino Customer Success Team
  - no children
- **delimiter\***
  - required attribute **value** to define: the csv file delimiter. Suggested value: **,** (coma), for other values, please limit to delimiter supported [here](#)
  - no children
- **enclosure\***
  - required attribute **value** to define: the csv file enclosure. Suggested value: **"** (double quote), for other values, please limit to delimiter supported [here](#)
  - no children
- **escape\***
  - required attribute **value** to define: the csv file escape. Suggested value: **\** (backslash): please consider in XML escape is required and you will need to write **"\"**, for other values, please limit to delimiter supported [here](#)
  - no children
- **lineSeparator\***
  - required attribute **value** to define: the file line separator. Suggested value: **\n**, for other values, please contact Boxalino Customer Success Team
  - no children

IF SOURCE TYPE IS: **item\_data\_file**

- **itemIdColumn\***
  - required attribute **value** to define: the file format. Value: the exact name of the column (as must be indicated in the first row of the CSV file) indicating the unique product identifier  
make sure this item id is unique, as the system will group data based on this criteria.  
NB: must be the same identifier in all your **item\_data\_file** and **combined\_options** sources
  - no children

Example:

```
<source type="item_data_file" id="item_vals">
```

```
<file value="item_properties.csv" />
```

```
<itemIdColumn value="item_id" />
```

```
<format value="CSV" />
```

```
<encoding value="UTF-8" />
```

```

<delimiter value="," />
<enclosure value="&quot;" />
<escape value="\" />
<lineSeparator value="\\n" />
</source>

```

IF SOURCE TYPE IS: hierarchical

- referenceIdColumn\*
  - required attribute value to define: the exact name of the column (as must be indicated in the first row of the CSV file) indicating the unique category id (this category id should only appear on this row of the CSV and no other rows)  
NB: this file should not include any mapping between item ids and the reference ids, this file is only for the hierarchical data (for example the category tree, without any connection to item ids).  
For the connection to item ids, please use another source of the type: item\_data\_file
  - no children
- parentIdColumn\*
  - required attribute value to define: the file format. Value: the exact name of the column (as must be indicated in the first row of the CSV file) indicating the unique parent category id of the category on this row
  - no children
- labelColumns\*
  - no attributes
  - children: a <language> tag for each language declared in the <languages> tag before, with two required attributes: name with the same value as in the languages ids and a required attribute value to define the exact name of the columns (as must be indicated in the first row of the CSV file) indicating the category label in this language

Example:

```

<source type="hierarchical" id="categories">
  <file value="categories.csv" />
  <referenceIdColumn value="cat_id" />
  <parentIdColumn value="parent_id" />
  <labelColumns>
    <language name="fr" value="value_fr" />
    <language name="de" value="value_de" />
  </labelColumns>
  <format value="CSV" />
  <encoding value="UTF-8" />
  <delimiter value="," />
  <enclosure value="&quot;" />
  <escape value="\" />
  <lineSeparator value="\\n" />
</source>

```

IF SOURCE TYPE IS: combined\_options

- combinationIdColumn\*
  - required attribute value to define: the exact name of the column (as must be indicated in the first row of the CSV file) indicating the unique combination id (specific for this combination and for this item\_id) often called "sku" (this combination id should only appear on this row of the CSV and no other rows)
  - no children
- itemIdColumn\*
  - required attribute value to define: the item id  
NB: must be the same identifier in all your item\_data\_file and combined\_options sources
  - no children
- combinationColumns\*
  - no attributes
  - at least one children tag <column> with a required attribute value identifying a combination column (as must be indicated in

the first row of the CSV file).

For example: if you have different prices and stocks according to the size and the color, both size and color must be declared as combination columns.

- targetColumns\*
  - no attributes
  - at least one children tag <column> with a required attribute value identifying a target column (as must be indicated in the first row of the CSV file).  
For example: if you have different prices and stocks according to the size and the color, both price and stock must be declared as target columns.

Example:

```
<source type="combined_options" id="item_options">
  <file value="item_options.csv" />
  <combinationIdColumn value="sku" />
  <itemIdColumn value="item_id" />
  <combinationColumns>
    <column value="size" />
    <column value="color_id" />
  </combinationColumns>
  <targetColumns>
    <column value="price" />
    <column value="discounted_price" />
    <column value="availability" />
  </targetColumns>
  <format value="CSV" />
  <encoding value="UTF-8" />
  <delimiter value="," />
  <enclosure value=""" />
  <escape value="\" />
  <lineSeparator value="\\n" />
</source>
```

IF SOURCE TYPE IS: resource

- referenceIdColumn\*
  - required attribute value to define: the exact name of the column (as must be indicated in the first row of the CSV file) indicating the unique resource id (this id should only appear on this row of the CSV and no other rows)  
NB: this file should not include any mapping between item ids and the reference ids, this file is only for the mapping data (for example the mapping between ids and labels for different "stock" status).  
For the connection to item ids, please use another source of the type: item\_data\_file
  - no children
- labelColumns\*
  - no attributes
  - children: a <language> tag for each language declared in the <languages> tag before, with two required attributes: name with the same value as in the languages ids and a required attribute value to define the exact name of the columns (as must be indicated in the first row of the CSV file) indicating the resource label in this language

Example:

```
<source type="resource" id="colors">
  <file value="colors.csv" />
  <referenceIdColumn value="color_id" />
  <labelColumns>
    <language name="fr" value="value_fr" />
  </labelColumns>
</source>
```

```

    <language name="de" value="value_de" />

</labelColumns>

<format value="CSV" />

<encoding value="UTF-8" />

<delimiter value="," />

<enclosure value="&quot;" />

<escape value="\\" />

<lineSeparator value="\n" />

</source>

```

## <properties>

The <properties> tag is required but is only a container for the inner elements.

#Attributes: none

#Children tags (\* required, + at least one)

- property+

## <property>

The <property> tag describes the data configuration for a specific item property to be generated out of the provided data.

Typically, a product has at least the following properties defined: title, description, price, category... but can have many more.

#Attributes:

- id: a unique identifier for the property (unique only per container)
- type: the type of the container, must be one of the following values
  - id: only one per container, is required, provides the information about the item id to be used (usually, is the same as the item id defined as itemColumnId in the source)
  - title: only one per container, not required, but strongly suggested for containers of the type products, is localized (same configuration as type text)
  - body: only one per container, not required, but strongly suggest for containers of the type products, is localized (same configuration as type text)
  - price: only one per container, contains the original price of the product, not required, but strongly suggest for containers of the type products, numeric (same configuration as number)
  - discounted: only one per container, contains the final price with discount of a product, not required, but strongly suggest for containers of the type products, numeric (same configuration as number)
  - hierarchical: contains the connection between an item and a hierarchical resource (requires a hierarchical resource, like example above for categories, to be defined)
  - text: for localized string (different values for each language)
  - string: for non-localized string (same value for each language)
  - number: for numerical values
  - date: for date values

#Children tags (\* required, + at least one)

- transform\*
- params\*

## <transform>

The <transform> tag is required but is only a container for the inner elements.

#Attributes: none

#Children tags (\* required, + at least one)

- logic+

## <logic>

The <logic> tag defines a mapping logic between your source and the property (how the data is taken from your source to be transformed into a property).

#Attributes:

- source: the id of the source (as defined in the <sources> tag above)
- type: the type of the logic to apply for the mapping. Must be one of the following values:
  - direct
    - for a direct mapping (simple copy of your source value into the property value);
    - requires one <field> child for properties of the same type as string and two for properties of the same type as text
  - reference
    - for an indirect mapping through a reference source
    - requires a param child tag <referenceSource> with a value attribute with the id of the reference source

#Children tags (\* required, + at least one)

- field[1 or 2] (if the property is localized, then as many as they are languages defined in <languages> tag (see above), otherwise 1)
  - If the property is localized, an attribute language is required with the appropriate id as value
  - In all cases: an attribute column is required with the exact name of the column (as must be indicated in the first row of the CSV file) for this property value (in this language if localized)
  - It is possible to map the same column to several languages if desired (but the mapping must be explicitly set for each language if the field is localized)
  - No children tags

## <params>

The <params> provides logic specific parameters depending on the type of the logic (see above in <logic>).

no attributes, only children tags.

Example:

```
<property id="category" type="hierarchical">
  <transform>
    <logic source="item_vals" type="reference">
      <field column="item_category_id" />
    </logic>
  </transform>
  <params>
    <referenceSource value="categories" />
  </params>
</property>
```