

Dynamic Tables reinvented

Do you know how to create a dynamic table for an invoice and use nested detail data to show complementary information? It is a simple question but most will be challenged to accomplish this seemingly simple task.

OL Connect 2020.1 introduces a range of new features to simplify this scenario and provides ways to create more sophisticated dynamic tables. It comes with a **new Insert Dynamic Table wizard** and puts **new HTML attributes** at your disposal. These attributes let you add data to your dynamic table and modify the table structure **without writing a single script**. And yes... it **supports nested detail data**.

In this first article you'll explore the new Insert Dynamic Table wizard and learn about the underlying HTML attributes. We will elaborate on various techniques to place product *A ributes* below the name of the product in an invoice template.

Note! The screenshots in this article are taken from a early version of OL Connect 2020.1. The implementation could be different in the final implementation.

Exploring the Insert Dynamic Table wizard

We'll start with a basic invoice data file. It contains information about the recipient and the line items are stored in a detail table. The product may have an arbitrary number of *A ributes*. These are provided as a HTML string and stored in a single field of the detail data (the *A ributes* field). The Data Model looks like this:

Additional attributes (optional) will be introduced to repeat the *product* lines in case a page break occurs within the virtual group.

CSS tricks

In the example above we assigned a custom CSS classes to the product and attributes rows. One can use the **page-break-before** and **page-break-after** CSS properties to influence the position where page breaks are allowed. The following example prevents page-breaks to happen before attributes rows (e.g. no split will happen between rows with the *attributes* class). Using this technique product and attribute lines always stay together.

```
tr.attributes {  
  page-break-before: avoid;  
}
```

Backward compatibility

Dynamic tables created in older versions of OL Connect do not use the new table expander logic. Therefore these tables render as the did before.

What is next?

This article describes the essentials of the new table expander logic. Additional articles will be published on how to repeat rows as in-table headers (for example our *product* row), how to apply formatting to data-field elements, write custom scripts and calculate totals. Stay tuned!