Column based files

Uploading data from Spreadsheets · https://paiyroll.com/kb/column-based-files/

Follow this guide if you need to upload **Column-based** files. These files have one row per worker with multiple columns. Columns will typically be mapped to existing user-defined Pay Definitions. The single value from each column is used to update a Pay Item input on the worker. You can upload in CSV or XLSX formats. Using XLSX, the works_id column must be formatted as text, not numeric.

There are two steps required: creating the pay definitions and then creating the data feed uploader..

Before you start

Dots are dropped from headings and repalced by a space, so **Att.Salary** is defined as **{Att Salary}**

Headings must match case and spaces exactly.

Only the first sheet of a workbook is read

1. Pay Definitions

The first step is to create pay definitions that the data-feed uploader will require. For any columns you wish to upload, you will need a pay definition. For example, if you have a column in your file called "Monthly Salary", you will need a Pay Definition that will be used. In most cases, you would use the same name, such as "Monthly Salary."

Single Input

When creating Pay Definitions, the most commonly used cases will involve Pay Definitions based on the single-input **Payment** Pay Template. If a column contains negative values, the **Payment** Pay Template will also work for these.

If your sheet has positive values as deductions, you will need to use the **Deduction** Pay Template.

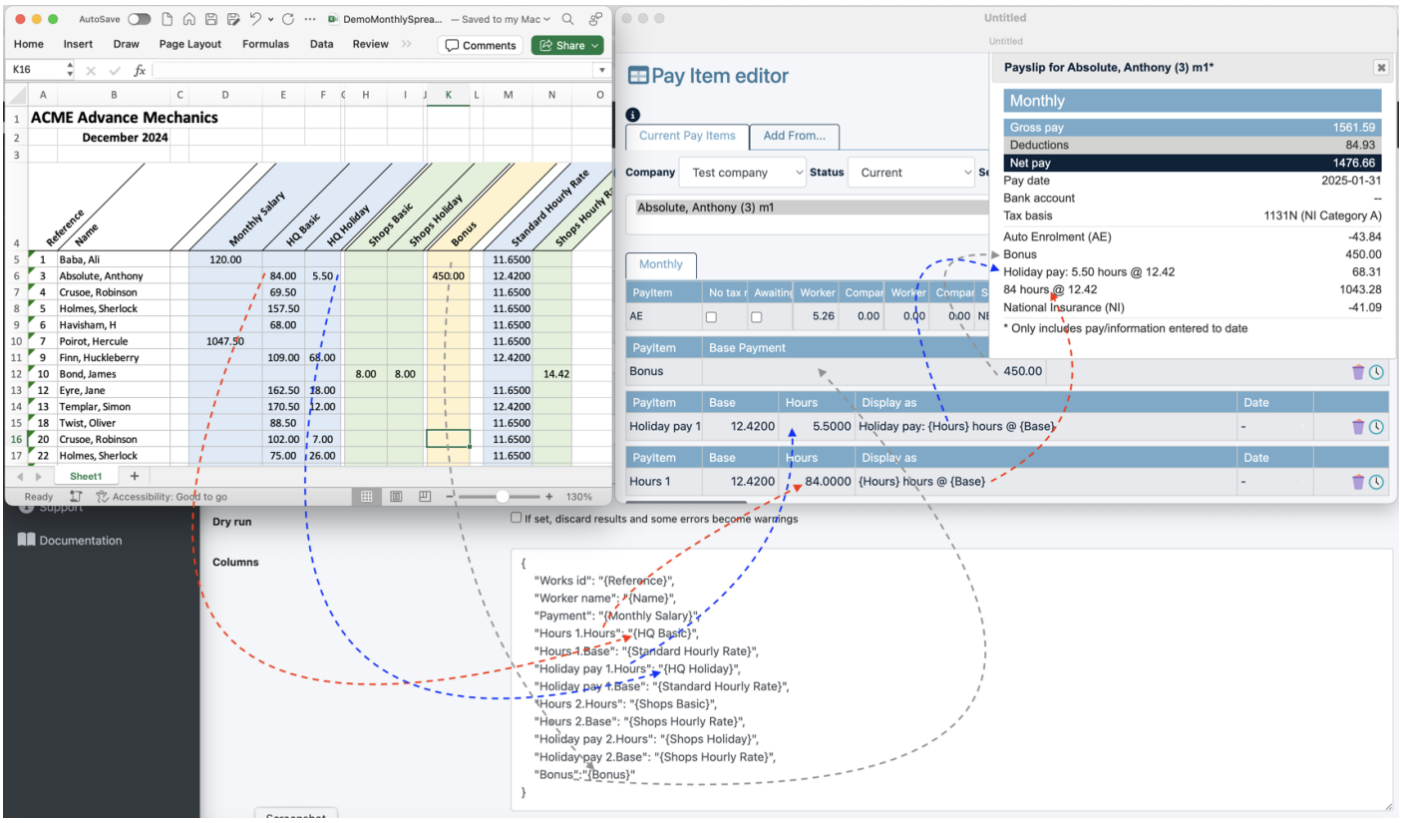
Take care to ensure that Gross and Net Deductions/Payments are set up correctly.

Double inputs

Sometimes you will need to create pay definitions with two inputs, typically based on the **Per unit** Pay Template. For example, if you need to read hours from one column and rate from another column

Example

Below is an example showing how Monthly Salary, Basic Pay and Holiday pay can be uploaded for 2 locations (HQ and Shops):



2. Column Data feed

The second step is to create the data feed column uploader.

Setup > **Report Definitions** > **Add** > **Upload pay item Columns from CSV**

Enter your preferred **Name** and **Description**.

Columns: Carefully edit the **Columns** JSON to map each file column cell to a Pay Item input.

Columns

The column mapping may appear complicated, but upon closer examination, it proves to be relatively straightforward. It essentially maps what paiyrol needs to where it can be found in the file.

Each line consists of two parts: one to the left of the colon and one to the right of the colon. The left is what is needed by paiyrol® Payroll, and the right is how it can be read from the file. For example, apiyroll® needs **Works Id**, which comes from a column in the file called {**WorkerId**}. It also uses the **Worker name**, which comes from a column called **{WorkerName}.** Note how the columns are always wrapped in curly brackets:

```
{
    "Works id":         "{WorkerId}",
    "Worker name":  "{WorkerName}",
    "Payment":          "{Monthly Salary}",
    "Hours 1.Hours": "{HQ Basic}",
    "Hours 1.Base":   "{Standard Hourly Rate}",
}
```

The remainder of the mapping will map the needed columns to the Pay Definitions: We have a **Payment** Pay Definition, which is mapped to a column {Monthly Salary}. We also have an Hours 1 Pay Definition, which is mapped to 2 columns in the file: **{HQ Basic}** and **{Standard Hourly Rate}**. Note the dots used to split the Pay Definition name and the name of an input – **Hours** or **Base** in this case. Please refer to the full example above.

You may find it easier to create the column mapping text in a text editor and then paste it in when complete.

Select the pay **Schedule**.

Skip rows: If the header starts on a different row after the first row, enter the number of rows to skip.

Dry-run. Determines whether the data feed will create Pay Items or just tell you what it would do.

Operation

Once you have completed steps one through three above, you can test your data feed. We always recommend testing in dry run mode first.

Data feeds > Upload pay item Columns from CSV > Upload

Detailed operation

This Universal Upload pay items Columns facility can upload:

- **Hours-only shifts.** Similar to Universal_Upload_shifts, but uploads the Hours only. The Rates are held and managed in paiyroll®, and only the Hours are exported from a T&A system.
- **Payments** and **Deductions**.
- All other Pay Items by named input

Each row specifies an employee and input values for multiple pay items. For each Pay Item:

- If all the inputs are blank, nothing is uploaded.
- Nothing is uploaded if all the numeric inputs are zero and the Pay Item does not exist.
- If the Pay Item does not exist, the given inputs will be supplemented with the inputs from the pay definition (i.e. as a set of defaults).
- The Pay Item will be created if necessary, and input values will be uploaded to it.

1. If the Columns setting maps the column heading to the name of a Pay Definition, which is Shift-like, for example:

```
"Basic": "{BasicHours}",
"Overtime15": "{OvertimeHours}",
"Overtime20": "{WeekendHours}",
```

Then, a single cell value is applied to the “Hours” input.

Each column with an input value to be uploaded is processed as follows:

- Note that if the Pay Item has to be created, the Rate will be defaulted from the Pay Definition, so if per-employee custom Rates are required, the Pay Item must be edited to set the required Rate. The Pay Item upload wizard (and the associated download capability) can be used to streamline this process.
- Custom pre-processing allows each Shift-like Pay Item to be presented on payslips using a tailored, and automatically completed, “display-as” format. This can be used to show the details of the Shift worked (and so meet any statutory requirement for such granular information).
-

2. Otherwise, if the Columns setting maps a column heading to a Payment or Deduction Pay Definition name, for example:

```
"Expenses": "{Expenses}",
```

The single cell value is applied to the “basis” input.

3. Otherwise, the Columns setting(s) must map one or more column headings to the Pay Definition name and input name pairs with a “.” in between, for example:

```
"Annual Salary.Next Base Salary": "{Salary}",
"Annual Salary.Next Effective Date": "{EffectiveFrom}",
```

The cell values are applied to the named inputs.