



Professional Services Workbench

Ledger Export XML Processing Guide

Version 6.0

August 2016

© 2016 systems@work. All rights reserved

Contents

| | |
|------------------------------------|---|
| Introduction | 3 |
| Example – Exporting Invoices | 5 |

Introduction

systems@work Ledger Export selects transactions and fields from the systems@work ledgers and formats them usually for import into other software systems.

The process involves many steps, including:

- The selection of transactions for export
- The selection of fields for export from the selected transactions
- The renaming of fields
- The preparation of transactions for import into another software system

This last step, the preparation of transactions for import into another software system, itself involves many steps, especially if the result is to be an accounting journal. In this case, the processes involve:

- The creation of at least two, and often three, rows of a journal from one systems@work ledger transaction (for example, in the case of an invoice, a debtor row, a tax row and a revenue row)
- The summarising of values for a particular row (for example, an invoice journal will usually contain just one debit to the debtor account, a credit for each tax category to the tax account, and as any revenue credits to the revenue account as preserve the analytical fields that are required for management reporting from the general ledger, such as employee, project, profit centre, etc.)
- The elimination from each type of debit or credit of unnecessary fields
- The insertion of additional field values not present in systems@work, such as a Journal Type
- The formatting of a file. This may be a text file, or an XML file or various other types of file
- The renaming of fields, which, by default, were output from Ledger Export into an XML file using the standard database names of all fields

Until systems@work version 4.8.3 all of these tasks were managed using XSL. This resulted in XSL of great complexity.

From systems@work version 4.8.3 some of these tasks can alternatively be managed using a suite of programs that can be called from Ledger Export.

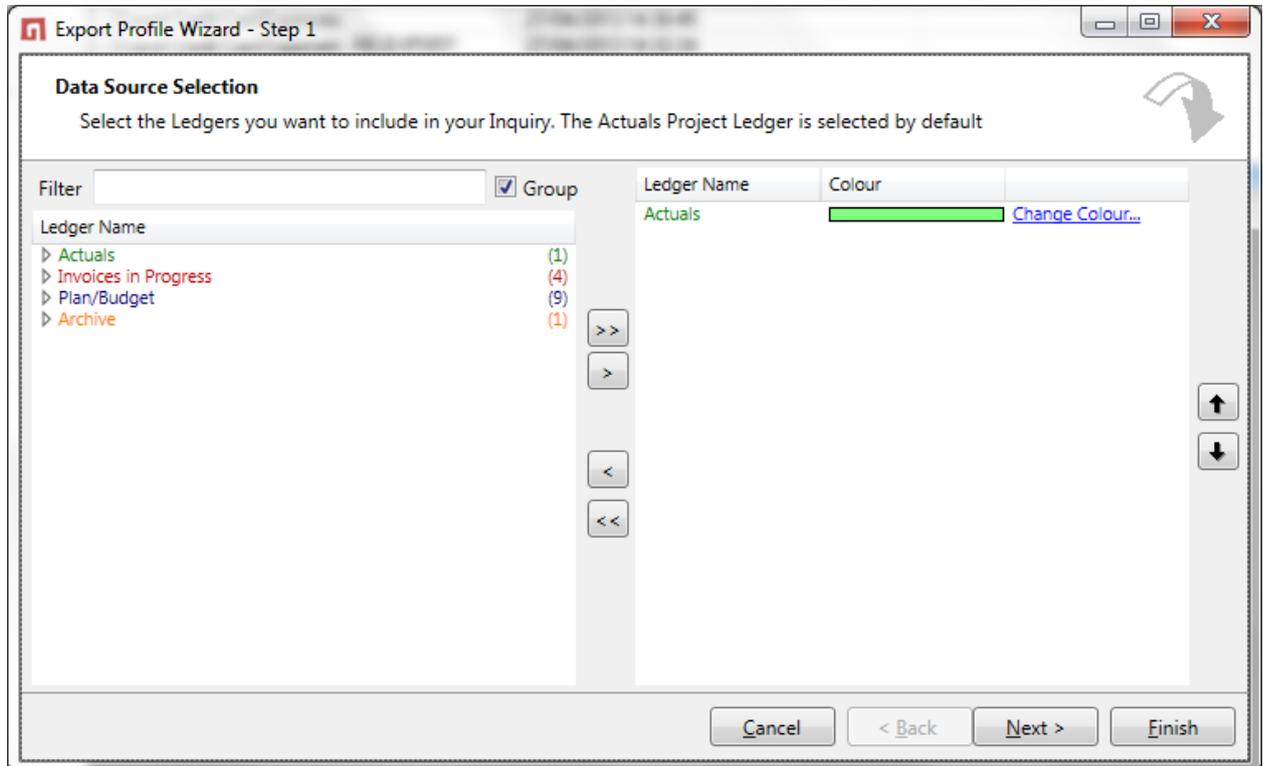
| | |
|----------|--|
| MULTIPLY | Creates a specified number of rows from each systems@work ledger row |
| GROUP | Summarises specified calculation values on the basis of a list of fields, all other fields being eliminated |
| SORT | Sorts the resulting file based on a list of fields |
| RENAME | Ensures that the names given in the Ledger Export profile are the names of the fields in the output XML file |

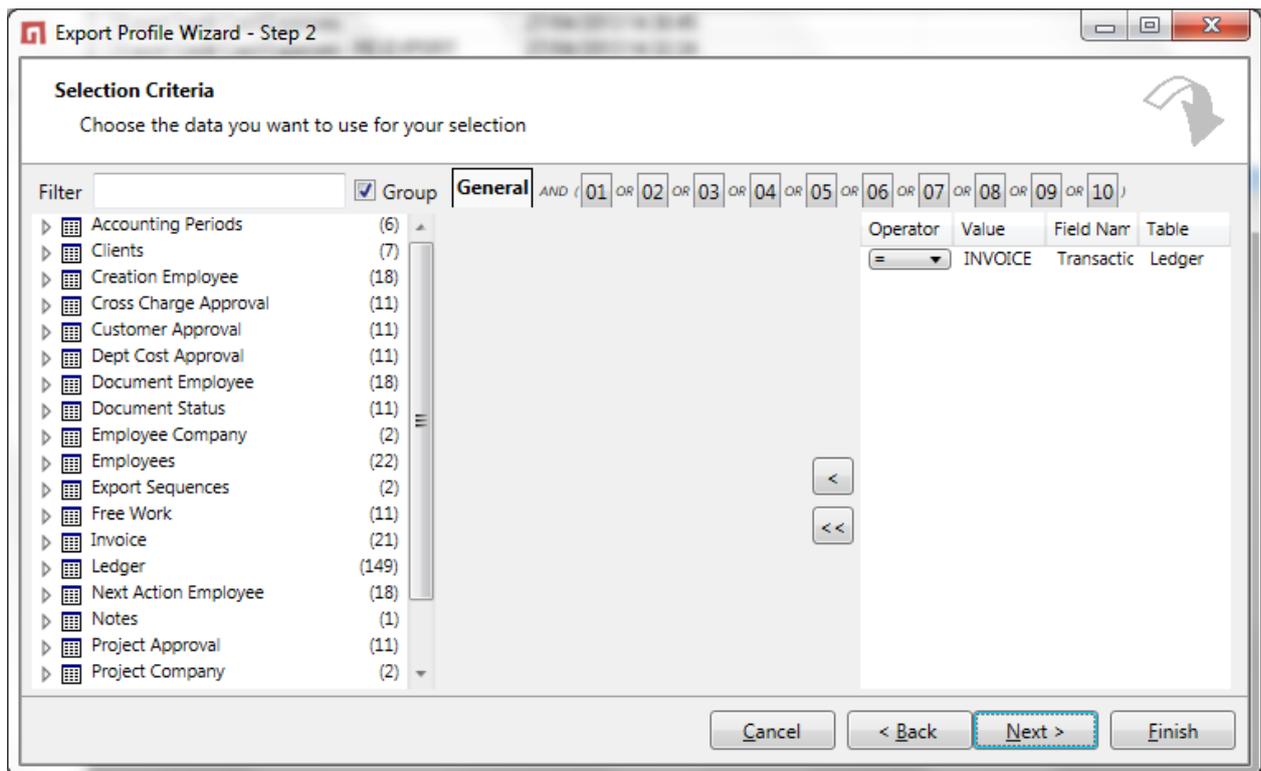
This means that XSL is needed only for the insertion of additional field values and for the formatting of the resulting file. The result is a greatly simplified and easier-to-maintain XSL stylesheet.

Example – Exporting Invoices

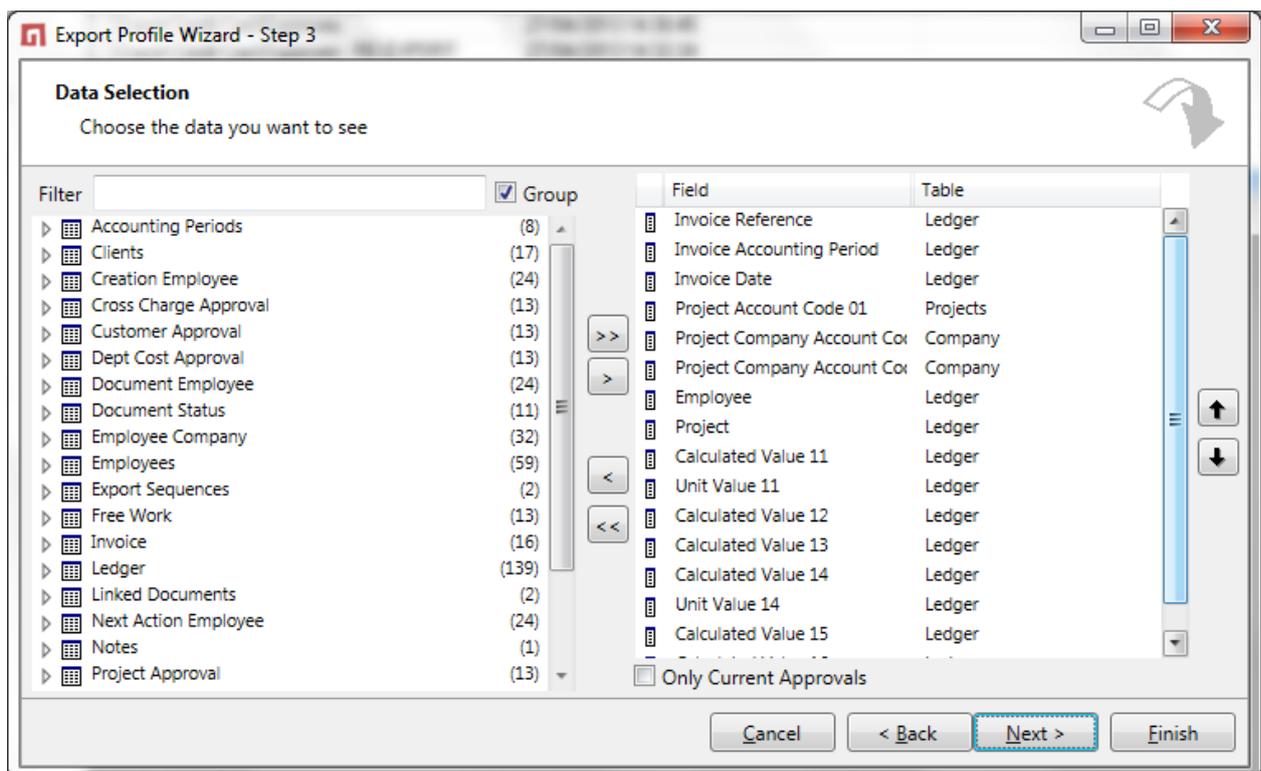
The following example is based on a typical export of invoices from time@work for import into an accounting system.

The selected transactions from the Actuals ledger will be of type INVOICE.





The selected fields will be:



These are renamed as:

Export Profile Wizard - Step 5

Set columns
Set column properties

| Default name | New name | Width | Suppress Total |
|---------------------------------|------------|-------|--------------------------|
| Invoice Reference | Reference | | |
| Invoice Accounting Period | Acc Period | | |
| Invoice Date | Date | | |
| Project Account Code 01 | Debtor Acc | | |
| Project Company Account Code 02 | Sales Acc | | |
| Project Company Account Code 03 | Tax Acc | | |
| Employee | Emp | | |
| Project | Proj | | |
| Calculated Value 11 | FX Net | | <input type="checkbox"/> |
| Unit Value 11 | FX Curr | | |
| Calculated Value 12 | FX Tax | | <input type="checkbox"/> |

Buttons:

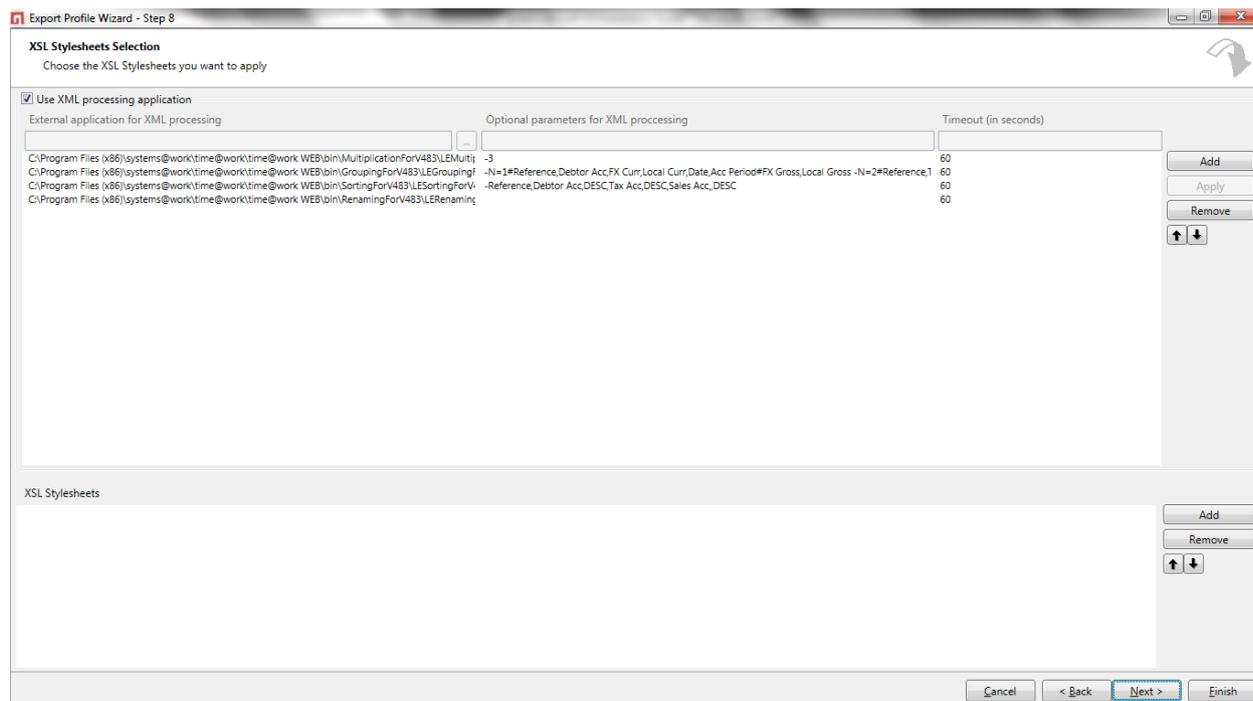
Export Profile Wizard - Step 5

Set columns
Set column properties

| Default name | New name | Width | Suppress Total |
|---------------------------------|-------------|-------|--------------------------|
| Project Company Account Code 03 | Tax Acc | | |
| Employee | Emp | | |
| Project | Proj | | |
| Calculated Value 11 | FX Net | | <input type="checkbox"/> |
| Unit Value 11 | FX Curr | | |
| Calculated Value 12 | FX Tax | | <input type="checkbox"/> |
| Calculated Value 13 | FX Gross | | <input type="checkbox"/> |
| Calculated Value 14 | Local Net | | <input type="checkbox"/> |
| Unit Value 14 | Local Curr | | |
| Calculated Value 15 | Local Tax | | <input type="checkbox"/> |
| Calculated Value 16 | Local Gross | | <input type="checkbox"/> |

Buttons:

The suite of four programs is then called.



Each of these programs expects parameters to be supplied in specific ways.

| | |
|----------|----|
| MULTIPLY | -N |
|----------|----|

Where N is the number of rows to be created from each systems@work transaction.

| | |
|-------|---|
| GROUP | -N=RowID#FieldName1,FieldName2,FieldName3#CalcName1,CalcName2,CalcName3 |
|-------|---|

Where RowID identifies one of the rows resulting from multiplication

Where # is a separator

Where FieldNameN identifies a field to be preserved in the output row and to be used as a basis for summarising the calculation values, each field name separated by a comma

Where # is a separator

Where CalcNameN identifies a calculation value to be preserved in the output row and summarised

Where a trailing space precedes the specification for the next RowID

| | |
|------|---|
| SORT | -FieldName1,SEQ;FieldName2,SEQ;FieldName3,SEQ |
|------|---|

Where FieldNameN identifies a field on which the output file is to be sorted

Where SEQ is either ASC or DESC to determine the direction of sorting. If absent, ASC is assumed.

A semicolon must separate each FieldName.

In this example, where we are creating an invoice journal, the parameters are set up as follows:

MULTIPLY

-3

This creates three rows of types 1, 2 and 3 from each systems@work ledger row.

GROUP

-N=1#Reference,Debtor Acc,FX Curr,Local Curr,Date,Acc Period#FX Gross,Local Gross

-N=2#Reference,Tax Acc,FX Curr,Local Curr,Date,Acc Period#FX Tax,Local Tax

-N=3#Reference,Sales Acc,FX Curr,Local Curr,Date,Acc Period,Emp,Proj#FX Net,Local Net

This summarises FX Gross and Local Gross values from all rows of type 1 on the basis of Reference, Debtor Acc, FX Curr, Local Curr, Date and Acc Period.

This summarises FX Tax and Local Tax values from all rows of type 2 on the basis of Reference, Tax Acc, FX Curr, Local Curr, Date and Acc Period.

This summarises FX Net and Local Net values from all rows of type 3 on the basis of Reference, Sales Acc, FX Curr, Local Curr, Date, Acc Period, Emp and Proj.

SORT

-Reference;Debtor Acc,DESC;Tax Acc,DESC;Sales Acc,DESC

This sorts the output file by Reference (ascending), Debtor Acc (descending), Tax Acc (descending) and Sales Acc (descending).

In this example the result should be, for each invoice:

- A single debtor transaction, followed by
- A single tax transaction, followed by
- As many revenue transactions as there are combinations of Employee and Project

RENAME

This program applies the names given in Step 5 to the fields in the final XML file.

The final XML looks like this (one debtor row, one tax row and two revenue rows):

```
<PROJECT_LEDGER_LINE>
  <N>
</N>
  <Proj VALUE="">
    <Debtor_Acc>D-002</Debtor_Acc>
  </Proj>
  <Emp VALUE="" />
  <Reference>INFR000001</Reference>
  <Date>2013-01-31</Date>
  <Acc_Period>2013001</Acc_Period>
  <FX_Net />
  <FX_Curr>EUR</FX_Curr>
  <FX_Tax />
  <FX_Gross>-22724</FX_Gross>
  <Local_Net />
  <Local_Curr>EUR</Local_Curr>
  <Local_Tax />
  <Local_Gross>-22724</Local_Gross>
  <ExportSequence>
</ExportSequence>
  <ExportSequenceNo>
</ExportSequenceNo>
  <Project_Company_Accounts>
    <Sales_Acc />
    <Tax_Acc />
  </Project_Company_Accounts>
</PROJECT_LEDGER_LINE>
<PROJECT_LEDGER_LINE>
  <N>
</N>
  <Proj VALUE="">
    <Debtor_Acc />
  </Proj>
  <Emp VALUE="" />
  <Reference>INFR000001</Reference>
  <Date>2013-01-31</Date>
  <Acc_Period>2013001</Acc_Period>
  <FX_Net />
  <FX_Curr>EUR</FX_Curr>
  <FX_Tax>-3724</FX_Tax>
  <FX_Gross />
  <Local_Net />
  <Local_Curr>EUR</Local_Curr>
  <Local_Tax>-3724</Local_Tax>
  <Local_Gross />
  <ExportSequence>
</ExportSequence>
  <ExportSequenceNo>
</ExportSequenceNo>
  <Project_Company_Accounts>
    <Sales_Acc />
    <Tax_Acc>10002</Tax_Acc>
  </Project_Company_Accounts>
</PROJECT_LEDGER_LINE>
<PROJECT_LEDGER_LINE>
  <N>
</N>
```

```

<Proj VALUE="101-001">
  <Debtor_Acc />
</Proj>
<Emp VALUE="0011" />
<Reference>INFR000001</Reference>
<Date>2013-01-31</Date>
<Acc_Period>2013001</Acc_Period>
<FX_Net>-8735.64</FX_Net>
<FX_Curr>EUR</FX_Curr>
<FX_Tax />
<FX_Gross />
<Local_Net>-8735.64</Local_Net>
<Local_Curr>EUR</Local_Curr>
<Local_Tax />
<Local_Gross />
<ExportSequence>
</ExportSequence>
<ExportSequenceNo>
</ExportSequenceNo>
<Project_Company_Accounts>
  <Sales_Acc>10010</Sales_Acc>
  <Tax_Acc />
</Project_Company_Accounts>
</PROJECT_LEDGER_LINE>
<PROJECT_LEDGER_LINE>
  <N>
  </N>
  <Proj VALUE="101-001">
    <Debtor_Acc />
  </Proj>
  <Emp VALUE="0012" />
  <Reference>INFR000001</Reference>
  <Date>2013-01-31</Date>
  <Acc_Period>2013001</Acc_Period>
  <FX_Net>-10264.36</FX_Net>
  <FX_Curr>EUR</FX_Curr>
  <FX_Tax />
  <FX_Gross />
  <Local_Net>-10264.36</Local_Net>
  <Local_Curr>EUR</Local_Curr>
  <Local_Tax />
  <Local_Gross />
  <ExportSequence>
  </ExportSequence>
  <ExportSequenceNo>
  </ExportSequenceNo>
  <Project_Company_Accounts>
    <Sales_Acc>10010</Sales_Acc>
    <Tax_Acc />
  </Project_Company_Accounts>
</PROJECT_LEDGER_LINE>

```