



FILE LAYOUT IREXML V 1.0

**IMPORT REMITTANCE
2019-05-13**

VERSIONS

Version	Date	Description of change	Changed by
1.0	2019-05-13	First version	Seva Posviatenko
1.1	2019-05-22	Changing file example	Jeanette Fridolf

CONTENTS

1.	INTRODUCTION	3
2.	EXAMPLE.....	3
3.	ELEMENT DESCRIPTIONS	7

1. INTRODUCTION

File layout IREXML, import remittance, is used to import the party remittance information into all ARC's modules. The import file is an XML-file that must follow a specific scheme.

XML-scheme can be found at <http://schema.aptic.net/aptic-link-import-remittance.xsd>.

In the scheme, all fields that are available can be interpreted; we can determine the order of these and even point out which ones are required to be filled in. Many fields are optional and some of those fields might be replaced with default values if empty.

2. EXAMPLE

The example file below shows the file with party remittance information that will be inserted to the ARC.

```
<?xml version="1.0" encoding="utf-8"?>
<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://schema.aptic.net/aptic-link-import-remittance.xsd">
  <counterkey>remittance001</counterkey>
  <recordcount>1</recordcount>
  <extradata>extradata_remittance001</extradata>
  <journal>
    <sendingpartytype>EXTDEBTCOL</sendingpartytype>
    <sendingpartyref>DebtcollectionX</sendingpartyref>
    <journaldate>2019-04-05</journaldate>
    <currencycode>SEK</currencycode>
    <remittedamount>1000.0000</remittedamount>
    <invoicenum>209582739</invoicenum>
    <paymentreferencenum>2095827396931</paymentreferencenum>
    <duedate>2019-02-27</duedate>
    <referencenum>referencenum001</referencenum>
    <externaljournalno>externaljournalno001</externaljournalno>
    <comment>comment_remittance001</comment>
    <journaldebitsum>15700.0000</journaldebitsum>
    <journalcreditsum>14700.0000</journalcreditsum>
    <cases>
      <case>
        <casenum>1299</casenum>
        <creditor>1000</creditor>
        <currencycode>SEK</currencycode>
        <final>true</final>
        <collectedamount>15700.0000</collectedamount>
        <remittanceamount>1000.0000</remittanceamount>
        <closuredate>2019-05-08</closuredate>
        <closurecode>PAIDINFULL</closurecode>
        <closurereason>Paid in full with success</closurereason>
        <transactions>
          <transaction>
            <transtype>Collected</transtype>
            <transdate>2019-05-07</transdate>
            <principal>14700.0000</principal>
            <vat>0.0000</vat>
            <interest>0.0000</interest>
          </transaction>
        </transactions>
      </case>
    </cases>
  </journal>
</import>
```

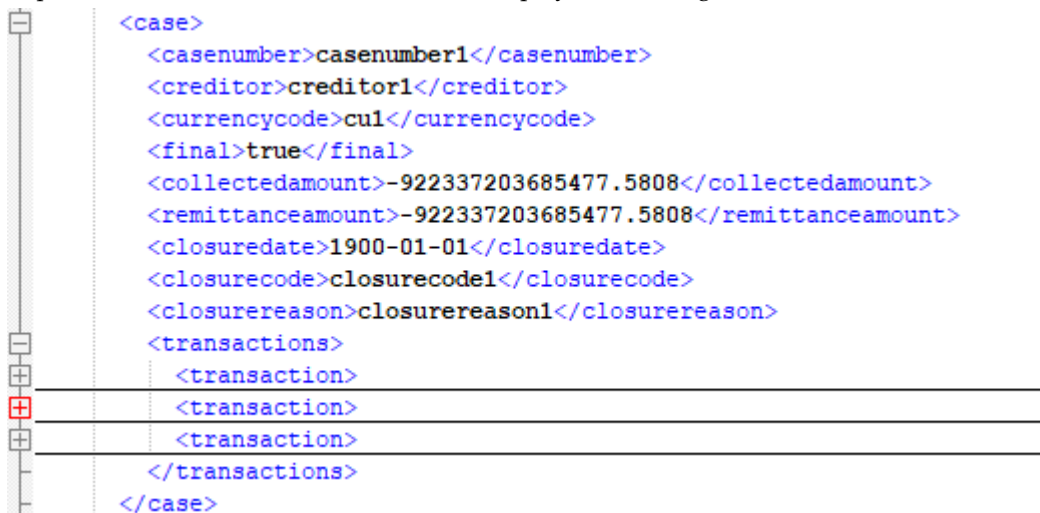
```

        <description>description_remittance001</description>
        <debtref></debtref>
        <feecode></feecode>
        <feetype></feetype>
        <creditorfeecode></creditorfeecode>
        <outlayfeecode></outlayfeecode>
    </transaction>
</transactions>
</case>
</cases>
</journal>
</import>

```

The XML example above displays the import file having data for one case (collection account) with one transaction. The structure of the XML file can be extended and there are three type of extensions for an XML file structure.

This means that each case in an import XML file can store set of transactions (the first type of extension). It's only needed to multiply the <transaction> block and populate transaction values. The basic structure of an import XML file in such case will look like displayed on the figure below.



```

<case>
  <casenumber>casenumber1</casenumber>
  <creditor>creditor1</creditor>
  <currencycode>cu1</currencycode>
  <final>true</final>
  <collectedamount>-922337203685477.5808</collectedamount>
  <remittanceamount>-922337203685477.5808</remittanceamount>
  <closuredate>1900-01-01</closuredate>
  <closurecode>closurecode1</closurecode>
  <closurereason>closurereason1</closurereason>
  <transactions>
    <transaction>
  _____
    <transaction>
  _____
    <transaction>
  _____
  </transactions>
</case>

```

Also, an import XML file can be extended to have the set of cases in it (the second type of extension). The basic structure of an import XML file in this case is shown on the figure below.

```

<journal>
  <sendingpartytype>DEBTORSOLICITOR</sendingpartytype>
  <sendingpartyref>sendingpartyref1</sendingpartyref>
  <journaldate>1900-01-01</journaldate>
  <currencycode>cul</currencycode>
  <remittedamount>-922337203685477.5808</remittedamount>
  <invoicenum>invoicenum1</invoicenum>
  <paymentreferencenum>paymentreferencenum1</paymentreferencenum>
  <duedate>1900-01-01</duedate>
  <referencenum>referencenum1</referencenum>
  <externaljournalno>externaljournalno1</externaljournalno>
  <comment>comment1</comment>
  <journaldebitsum>-922337203685477.5808</journaldebitsum>
  <journalcreditsum>-922337203685477.5808</journalcreditsum>
  <cases>
    <case>
      <casenum>casenum1</casenum>
      <creditor>creditor1</creditor>
      <currencycode>cul</currencycode>
      <final>true</final>
      <collectedamount>-922337203685477.5808</collectedamount>
      <remittanceamount>-922337203685477.5808</remittanceamount>
      <closuredate>1900-01-01</closuredate>
      <closurecode>closurecode1</closurecode>
      <closurereason>closurereason1</closurereason>
      <transactions>
        <transaction>
          <transtype>Collected</transtype>
          <transdate>1900-01-01</transdate>
          <principal>-922337203685477.5808</principal>
          <vat>-922337203685477.5808</vat>
          <interest>-922337203685477.5808</interest>
          <description>description1</description>
          <debtref>debtref1</debtref>
          <feecode>feecode1</feecode>
          <feetype>feetype1</feetype>
          <creditorfeecode>creditorfeecode1</creditorfeecode>
          <outlayfeecode>outlayfeecode1</outlayfeecode>
        </transaction>
        <transaction>
        </transaction>
      </transactions>
    </case>
    <case>
    </case>
  </cases>
</journal>

```

The third type of extension is connected to an XML file having set of journals in one file. In such case the basic XML structure of the import file will look like displayed on the figure below.

```

<?xml version="1.0" encoding="utf-8"?>
<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://schema.aptic.net/aptic-link-import-remittance.xsd">
  <counterkey>counterkey1</counterkey>
  <recordcount>1</recordcount>
  <extradata>extradata1</extradata>
  <journal>
    <sendingpartytype>DEBTORSOLICITOR</sendingpartytype>
    <sendingpartyref>sendingpartyref1</sendingpartyref>
    <journaldate>1900-01-01</journaldate>
    <currencycode>cul</currencycode>
    <remittedamount>-922337203685477.5808</remittedamount>
    <invoicenumber>invoicenumber1</invoicenumber>
    <paymentreferencenumber>paymentreferencenumber1</paymentreferencenumber>
    <duedate>1900-01-01</duedate>
    <referencenumber>referencenumber1</referencenumber>
    <externaljournalno>externaljournalno1</externaljournalno>
    <comment>comment1</comment>
    <journaldebitsum>-922337203685477.5808</journaldebitsum>
    <journalcreditsum>-922337203685477.5808</journalcreditsum>
    <cases>
      <case>
        <casenumber>casenumber1</casenumber>
        <creditor>creditor1</creditor>
        <currencycode>cul</currencycode>
        <final>true</final>
        <collectedamount>-922337203685477.5808</collectedamount>
        <remittanceamount>-922337203685477.5808</remittanceamount>
        <closuredate>1900-01-01</closuredate>
        <closurecode>closurecode1</closurecode>
        <closurereason>closurereason1</closurereason>
        <transactions>
          <transaction>
            <transtype>Collected</transtype>
            <transdate>1900-01-01</transdate>
            <principal>-922337203685477.5808</principal>
            <vat>-922337203685477.5808</vat>
            <interest>-922337203685477.5808</interest>
            <description>description1</description>
            <debtref>debtref1</debtref>
            <feecode>feecode1</feecode>
            <feetype>feetype1</feetype>
            <creditorfeecode>creditorfeecode1</creditorfeecode>
            <outlayfeecode>outlayfeecode1</outlayfeecode>
          </transaction>
          <transaction>
          </transaction>
        </transactions>
      </case>
    </cases>
  </journal>
</journal>
</import>

```

3. ELEMENT DESCRIPTIONS

Below you can find a short element description of the elements that are available.

Element	Type	Required	Description
import			
@counterkey	String(15)	Optional	File count identity. The name of the file counter. This is used to control that files are loaded in correct sequence.
@recordcount	Integer	Optional	The number of records in the file.
@extradata	String(500)	Optional	Additional information. Is saved in log file, is not used for other purposes.
import / journal			
@sendingpartytype	String(15)	Required	Type of the party sending the remittance.
@sendingpartyref	String(30)	Required	Reference code of the party sending the remittance.
@journaldate	Date	Required	Date of the journal to be imported.
@currencycode	String(3)	Required	Currency code according to ISO 4217.
@remittedamount	Money	Required	Remitted amount.
@invoicenummer	String(100)	Required	Invoice number.
@paymentreferencenummer	String(100)	Required	Payment reference number.
@duedate	Date	Optional	Debt due date.
@referencenummer	String(100)	Optional	Remittance reference number.
@externaljournalno	String(60)	Optional	External journal reference number.
@comment	String(160)	Optional	Remittance comment information.
@journaldebitsum	Money	Optional	Total debit amount for journal.
@journalcreditsum	Money	Optional	Total credit amount for journal.
import / journal / cases / case			
@casenummer	String(30)	Required	Collection account reference.
@creditor	String(15)	Optional	Creditor reference code.
@currencycode	String(3)	Required	Currency code according to ISO 4217.
@final	Boolean	Required	Element values are TRUE or FALSE. TRUE – in case when final remittance imported. FALSE – in case when intermediate remittance imported.
@collectedamount	Money	Required	Collected amount value on collection account.
@remittanceamount	Money	Required	Remittance amount value on collection account.
@closuredate	Date	Optional	Date of collection account closure.
@closurecode	String(15)	Optional	Collection account closure code.
@closurereason	String(400)	Optional	Collection account closure reason.
import / journal / cases / case / transactions / transaction			

@transtype	String(100)	Required	Transaction type code.
@transdate	Date	Required	Transaction date.
@principal	Money	Optional	Principal amount.
@vat	Money	Optional	VAT amount.
@interest	Money	Optional	Interest amount.
@description	String(400)	Optional	Transaction description.
@debtref	String(100)	Optional	Debt reference code.
@feecode	String(15)	Optional	Fee code.
@feetype	String(15)	Optional	Fee type.
@creditorfeecode	String(100)	Optional	Creditor fee type.
@outlayfeecode	String(100)	Optional	Outlay fee code.