File Layout IPCXML v 1.0

Import Payment Complex

Date: 2022-04-13

aptic

Event log

Version	Date	Change	Changed by
01.00	2021-02-15	Updated document with new Word template and adding new attributes and elements, including match payment reservation fields.	Jeanette Fridolf
01.01	2021-02-16	Updating information about debtref and reservationnumber attribute.	Jeanette Fridolf
01.02	2022-01-31	Updating description for attribute reference under spec element.	Jeanette Fridolf
01.03	2022-04-13	Updating description for spec element and preferred amount.	Jeanette Fridolf

Content

IMPXML – Import Match Payment Reservations

1 Introduction

The file layout IPCXML (Import Payments Complex) is used for importing payments into the system Aptic ARC. For ARC to read and validate this file the file layout is limited by a specific XML schema.

The XML schema can be downloaded from http://schema.aptic.net/aptic-link-import-payments-complex.xsd. Every element and its restrictions can be deduced in that file.

2 Example

This file example contains one payment from just one bank account and is an example of a simple form of this file.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<payments xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="http://schema.aptic.net/aptic-link-
import-payments-complex.xsd">
  <journal date="2013-11-08" totalamount="2000.00" bankaccount="44556677">
    <payment amount="2000.00" refno="554412" paidbyid="1">
    </payment>
  </journal>
</payments>
```

Another example is a little bit more complex showing several journals and different currencies amongst other things.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<payments xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="http://schema.aptic.net/aptic-link-
import-payments-complex.xsd">
  <journal date="2013-11-09" totalamount="5000" bankaccount="11133">
    <payment amount="4000" refno="133444" currency="EUR" paidbyid="1">
      <message>Payment from another person</message>
    </payment>
    <payment amount="1000" refno="31464" currency="EUR" paidbyid="1">
      <spec amountcode="P" principal="1000" interest="0"/>
    </payment>
  </journal>
  <journal date="2013-11-09" totalamount="2500.00" bankaccount="44556677">
    <payment amount="2500.00" refno="554412" paidbyid="1" currency="SEK">
    </payment>
  </journal>
</payments>
```

3 Element description

The following section contains information about the elements used in the XML-file. The ones that are prefixed with @ are attributes in the XML document.

Element /Attribute	Type	Required	Description
payments			
@counterkey	string (15)	No	File count identity. The name of the file counter. This is used to control that files are imported in the correct sequence. "counterkey" and "countervalue" are used as an option file counter to ensure that files are imported in correct order and not skipped. "counterkey" is the chosen name of the counter for this import. It is possible to share between different import files by using the same name. "countervalue" is the expected counter value that the system should have after increasing the last value by one. If it is the first time a counter key is used, the counter value will be used to initialize the counter. If the expected counter value does not match up with the saved value, the import will fail.
@countervalue	int	No	File number. This is used to control that files are imported in the correct sequence. Used together with counterkey.
@recordcount	int	No	Number of items in the file. Saved in log file but not used otherwise.
@extradata	string (500)	No	Additional information. Is saved in log file, not used for any other purpose.
@senderpartyref	string (50)	No	Sender party reference of the file. If filled, it needs to match a party reference code in ARC. Should be filled if the file contains values in element idnumber (SSN) for natural persons, due to GDPR.
@checkrefno	string	No	If set to true, check validity of the reference.
payments / journal			
@date	date	Yes	Date when the journal is created. Date format YYYY-MM-DD.
@accountingdate	date	No	
@totalamount	money	Yes	The total amount of the payments in this journal (ex 1344.10). Will be compared to the sum of the payments in the journal.
@bankaccount	string (50)	Yes	The bank account number in ARC where the payments will be booked on.
@comment	string (160)	No	A comment that is saved in the journal.

△ aptic

@externaljournalno	string (60)	No	Other unique reference number for this journal. Not used by ARC, only stored in database.
@bankaccountid	Int	No	If specified, Aptic ARC's internal bankaccountid (in database table dat_bankaccount).
@type	Int	No	From cfg_journaltype, normally not used: 1 = Manual 2 = Import 3 = Failed
@approved	boolean	No	If journal is pre-approved. If this is set to False, a user needs to approve the journal in ARC before it can be imported.
@hiddenfromui	boolean	No	If the journal should be visible in ARC GUI or not.
payments / journal / payment			
@amount	money	Yes	The initial amount of the payment.
@refno	string (100)	Yes	The payment reference. The value set here will match on case level or debt level. Example of value that can be used: • Payment reference (e.g., OCR in Sweden) • Debt reference (invoice number for debt) • Account payment reference • If payment from authority legal reference number can be used. • Payment reference in debtor link table • Old account reference on debt When a match on refno, payment will be deducted on case level even if a hit is found on a specific debt.
@currency	string (3)	No	Currency code according to ISO 4217.
@receivedamount	money	No	The amount that the system will receive.
@exchangedate	date	No	The exchange date to use if the currency differs from the bank account currency.
@exchangerate	money	No	The exchange rate to be used, if not supplied the rate in ARC will be used.
@exchangefee	money	No	The fee for the exchange.
@paidbyid	integer	Yes	Payer, the person/authority/company type that did the payment. Should match paidbyid in database table cfg_paidby. Available values: • 1 - Debtor/Customer • 2 - Other • 3 - Authority • 4 - Social welfare office • 5 - Insurance company • 6 - Debt relief
@paymentdate	date	No	The date of the payment.
@accountingdate	date	No	The accounting date the payment should be accounted.
@refno2	string (60)	No	Second reference number, is used for reference for legal payments among other things.
@debtorref	string (50)	No	The reference of the debtor.
@customerref	string (60)	No	The reference of the customer.
@debtref	string (60)	No	The reference of the debt/invoice. Normally invoice number. If this is added, the payment will be deducted on the specific debt with this debt reference. This can be used on it's own or together with refno (with for example payment reference filled), to match a debt.
@creditorref	string (100)	No	Reference code of creditor that is linked to the case/debt. If this is filled the payment will only match if it is the correct debtor linked to the refno/debtref.

@approved	boolean	No	If the payment is pre-approved. True is default.
@filmid	string (80)	No	Film ID for data purpose.
@bankaccountno	string (100)	No	The source of the payments bank account number. If parameter "Save bank account number when importing payment" (SV: "Spara bankkontonummer vid import av betalning") is set to true, this bank account number will be saved at debtor/customer. If import is used for payment reporting from creditor, do not add creditors bank account no in this field since this will update bank account information on customer/debtor.
@ swift	string (200)	No	SWIFT (Society for Worldwide Interbank Financial Telecommunication) for source of payments bank account number.
@iban	string (86)	No	IBAN (International Bank Account Number) for source of payments bank account number.
@blz	string (200)	No	BLZ (Bank Identifier Code) for source of payments bank account number. Is used in Germany and some other countries.
@bic	string (22)	No	BIC (Bank Identifier Code) for source of payments bank account number.
@status	integer	No	Status for payment. Possible values: • 0 - Unknown • 1 - Paymentplan • 2 - Full • 3 - Deviant • 4 - Overpayment • 5 - Other currency • 6 - Offer • 7 - Missing offer • 8 - Missing plan • 100 - Rejected no agreement • 101 - Rejected no coverage • 102 - Rejected retrying • 103 - Rejected
@idnumber	string (30)	No	ID-number (SSN/organization number) of person paying.
@firstname	string (200)	No	First name of payer.
@infix	string (30)	No	Infix of payer.
@lastname	string (120)	No	Last name of payer.
@initials	string (20)	No	Initials of payer.
@street	string (400)	No	Street of payer's address.
@city	string (70)	No	City of payer's address.
@statecode	string (4)	No	State code of payer's address.
@zipcode	string (20)	No	Zip code of payer's address
@countrycode	string (2)	No	Country code of payer's address, according to 2 chars ISO 3166-1 alpha-2.
@source	string (100)	No	The source of the payment, for example sending system or bank.
@finalpayment	boolean	No	If this is a final payment or not.
@externallyaccountedfor	boolean	No	If this payment is externally accounted for, for example debt collection outside Aptic ARC or direct payment. When this flag is set no reporting back will be sent to systems (e.g., ledger systems). This is to prevent the payment to be sent back to reporting system that already has the information.

@autotransferbatchno	integer	No	Batch number for auto transfer (SV: autogiro).
@autotransferrejectioncode	string (20)	No	Rejection code for auto transfer (SV: autogiro).
@autotransferrejectiontext	string (200)	No	Rejection text for auto transfer (SV: autogiro).
@retroactive	boolean	No	If the payment is retroactive or not.
@type	string (100)	No	Payment type
@paymentperiodfrom	date	No	Payment period from date.
@paymentperiodto	date	No	Payment period to date.
@reservationnumber	string (100)	No	Reservation number. Is used to match a "real" payment (in the bank file) with an existing payment reservation in ARC. Is for example used for Swish payments.
@deductcreditinvoice	boolean	No	If credit invoice should be deducted or not.
@lumpsum	boolean	No	The payment is a lumpsum for matching one or several payment reservations. The payment will always have a paymentreservations element if this flag is set to True. The payment can be a lumpsum despite the flag is set to False, but then the reservations already are matched in ARC in an earlier stage.
message	string (8000)	No	Message linked to the payment.

payments / journal / payment / spec
* For more information about deduction order and what to think about when using spec elements, see 0
***Spec element* functionality** *below.*

@amountcode	string (15)	Yes	The amount code for this deduction: • P - Principal • I - Interest • O - Outlay • F – Fee
@reference	string (50)	No	Reference to invoice number (dat_debthead.debtref), for this deduction. Can both be used for invoices but also to refer to a specific debt in collection using the original invoice number.
@debtid	integer	No	ARC's internal debt ID to map this deduction.
@principal	money	Yes	The principal amount of this deduction.
@interest	money	Yes	The interest amount of this deduction.

payments / journal / payment / reference

~~@refno~~			~~Not applicable at this moment.~~
~~@amount~~			~~Not applicable at this moment.~~

payments / journal / payment / paymentreservations

paymentserviceprovider	string (40)	No	Payment reference to be compared to payment reference at payment reservation in ARC (dat_payment.paymentrefstr). Can be empty if reservationnumber has a value. If both reservationpaymentreference and reservationnumber is filled, both must match the payment to be a match.

payments / journal / payment / paymentreservations / paymentreservation

reservationpaymentreference	string (100)	No	Payment reference to be compared to payment reference at payment reservation in ARC (dat_payment.paymentrefstr). Can be empty if reservationnumber has a value. If both reservationpaymentreference and reservationnumber is filled, both must match the payment to be a match.
reservationnumber	string (100)	No	Reservation number to be compared to reservation number at payment reservation in ARC (dat_payment.reservationnumber). Could have been sent from WIP call BookPaymentReservation, field ReservationNumber. Can be empty if reservationpaymentreference has a value. This needs to be

			filled if payment reservation should be added automatically, if the payment reservation does not exist in ARC since before.
reservationdate	date	No	Reservation date for payment reservation. • This needs to be filled if payment reservation should be added automatically if the payment reservation does not exist in ARC since before. • If this is filled and the payment reservation does exist in ARC this date needs to match the date of the payment reservation in ARC.
reservationamount	money	No	Reservation amount for the payment reservation. • This needs to be filled if payment reservation should be added automatically if the payment reservation does not exist in ARC since before. • If this is filled and the payment reservation does exist in ARC this amount needs to match the amount of the payment reservation in ARC. • Is only used for matching with reserved amount in ARC, whole payment will be booked. It is not possible to book part payment.
payments / journal / payment / paymentreservations / paymentreservation / caseidentifications			
idnumber	string (15)	No	Debtor ID number (SSN or organization number). Is used, when a missing payment reservation should be added, to link the payment reservation to correct debtor. If this is empty the payment will be linked to the main debtor. If this field is used and contains SSN of a natural person, @senderpartyref should be filled due to GDPR.
note	string (500)	No	Field for notes linked to the reservation. Can be used for manually identifying payment if not matched automatically. Only saved in import database table. If used it should be placed after caseidentifications element.
payments / journal / payment / paymentreservations / paymentreservation / caseidentifications / caseidentification			
caseguid	string	No	Unique GUID for debt collection case, stored in ARC. If one of the other case identifiers is filled, this can be empty.
casenumber	string (30)	No	Case number in ARC. If one of the other case identifiers is filled, this can be empty.
accountreferencenumber	string (100)	No	Account reference number provided by the client. Element creditorref needs to be filled if this case identifier is used. If one of the other case identifiers is filled, this can be empty.
invoicenumber	string (50)	No	Invoice number of one on the debts provided by the client. Element creditorref needs to be filled if this case identifier is used. If one of the other case identifiers is filled, this can be empty.
customernumber	string (50)	No	Creditor's customer number at the debtor. Element creditorref needs to be filled if this case identifier is used. If one of the other case identifiers is filled, this can be empty.
creditorref	string (30)	No	Creditor reference in ARC. This element needs to have a value if one of these elements are used: • accountreferencenumber • invoicenumber • customernumber

3.1 Spec element functionality

Functionality when using spec element under payment:

1. Distribute the amounts specified as "preferred amount" in the spec element if the paid amount covers preferred amounts (but not more than the remaining amount per debt and amount type).
2. Any remaining amount will be deducted at other debts (oldest first) with the amount type that is **not** specified in spec element. Note that also 0.00 is a specified amount in this context.
3. Any remaining amount will be deducted according to deduction order, on the oldest debt first, and the amount types that were specified as "preferred amounts".
4. Any remaining amount will be placed as over payment.

It is important to think that "preferred amount" with 0.00 is a choice to **not** deduct any money on that specific debt and amount type.

3.1.1 Examples

Example 1

One debt collection case with two debts (debt 1 is oldest). No debtref is given in spec, but all will be deducted "equally" on the case with payment reference 12345678911.

Debt	Principal amount	Fee amount	Interest amount
1	**200**	**60**	
2	**200**	**60**	

File, total payment is 400 SEK.

```xml
<?xml version="1.0" encoding="utf-8"?>
<payments xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSch
  <journal date="2022-04-13" totalamount="900" bankaccount="123456-7" comment="Exa
    <payment amount="400" refno="12345678911" currency="SEK" receivedamount="400"
      <message>Payment with specification.</message>
      <spec amountcode= "P" interest="0" principal="340.00" />
      <spec amountcode= "F" interest="0" principal="60.00" />
    </payment>
  </journal>
</payments>
```

How amount will be deducted. Green is deducted amount; red amount is not deducted.

Debt	Principal amount	Fee amount	Interest amount
1	**200**	**60**	
2	**140** + **60**	**60**	

Example 2

One debt collection case with three debts (debt 1 is oldest, 2 is older than 3). Debtref 2 is specified in spec element.

Debt	Principal amount	Fee amount	Interest amount
1	**100**	**60**	
2	**200**	**60**	
3	**300**		

File, total payment is 600 SEK.

```xml
<?xml version="1.0" encoding="utf-8"?>
<payments xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSc
  <journal date="2022-04-13" totalamount="600" bankaccount="123456-7" comment="Exa
    <payment amount="600" refno="12345678911" currency="SEK" receivedamount="600"
      <message>Payment with specification.</message>
      <spec amountcode= "P" reference= "2" interest="0" principal="200.00" />
      <spec amountcode= "F" reference= "2" interest="0" principal="0" />
    </payment>
  </journal>
</payments>
```

How amount will be deducted. Green is deducted amount; red amount is not deducted.

Debt	Principal amount	Fee amount	Interest amount
1	**100**	**60**	
2	**200**	**60**	
3	**240** + **60**		