



**akuiteo**  
BUSINESS SOFTWARE

Setup Guide

# DEMATERIALIZED INVOICES - CHORUS

Version 4.6

Revision number: 1

Published in: January 2022

Written by: Documentation team

Copyright (c) 2006-2022 Akuiteo S.A.S. All Rights Reserved.

Any total or partial reproduction of this material, whether its form or content, without prior written permission from the author, is strictly prohibited. The French law only allows, on one hand, copies or reproductions strictly reserved for private usage of the copyist and not destined for collective usage and, on the other hand, analysis and short quotes for the purpose of illustration.

The Akuiteo designation and logos are registered trademarks of the Akuiteo S.A.S. company. Any use of the trademarks without the authorization of the Akuiteo S.A.S. company is prohibited.

Visit: <http://www.akuiteo.com> and <http://www.akuiteo.com/blog/>

# Table of Contents

1 Preface .....	4
1.1 Revisions .....	4
1.2 Help desk .....	4
2 Presentation .....	5
2.1 Dematerializing invoices for CHORUS .....	5
2.2 Your activity .....	5
2.3 Steps to follow .....	5
3 Setting up the dematerialization of invoices .....	6
3.1 Activating management rules .....	6
3.2 Creating a dematerialization code .....	6
3.3 Transferring dematerialized invoices automatically to CHORUS PRO .....	7
3.3.1 Adding the Certigna certificate to the Tomcat server .....	7
3.3.2 Setting up the Administration console .....	8
3.3.3 Setting up the PISTE APIs and the CHORUS PRO account .....	9
3.3.4 Setting up the login information per company .....	10
3.3.5 Setting up the CheckFacturesClientChorus batch .....	11
4 Description of the generated XML file .....	12
4.1 Information about file format .....	12
4.2 Start of file - Header .....	13
4.3 Commitment .....	15
4.4 Main attachment .....	16
4.5 Additional attachments .....	17
4.6 Supplier .....	18
4.7 Debtor (customer) .....	20
4.8 Payment .....	21
4.9 Amounts .....	22
4.10 Tax overview .....	23
4.11 Lines .....	24

# 1 Preface

## 1.1 REVISIONS

**Revision 1**

Published in January 2022

- Updated sub-chapter [Transferring dematerialized invoices automatically to CHORUS PRO](#) (p. 7) for the multi-company.

## 1.2 HELP DESK

Akuiteo S.A.S. highly values your satisfaction.

To share your feedback or contact the help desk, feel free to visit our website page:

<https://www.akuiteo.fr/akuiteo.clients/>

## 2 Presentation

### 2.1 DEMATERIALIZING INVOICES FOR CHORUS

Since January 1, 2017, all French public entities have been required to accept dematerialized invoices sent by their suppliers. The issuers of invoices destined to the public sphere are also concerned by this requirement and need to progressively transition to dematerialized invoices.

These provisions have been mandatory since:

- 01/01/2018 for intermediate-sized companies;
- 01/01/2019 for small and medium-sized companies.

### 2.2 YOUR ACTIVITY

In this context, as a private or public entity sending invoices to a public entity (State, local public sector entities and public institutions), you must comply with some billing requirements.

There are two options available depending on the number of invoices generated by your company for public entities:

- If you have a few invoices to submit, you can manually save them via the CHORUS PRO portal.
- If you have a lot of invoices to submit, you can set up Akuiteo in order to dematerialize these invoices and you can then transfer them via the CHORUS PRO portal.

### 2.3 STEPS TO FOLLOW

- 1 Sign up to CHORUS PRO.
- 2 Set up Akuiteo with the help of a consultant.
- 3 Create dematerialized invoices.
- 4 Generate invoices in XML format.
- 5 Transfer invoices to CHORUS PRO:
  - Either manually via the CHORUS PRO portal;
  - Or by asking your IT service to connect to the CHORUS platform and transfer the XML files in "EDI mode";
  - Or automatically by using CHORUS PRO.

# 3 Setting up the dematerialization of invoices

## Reference

This document explains how to set up the dematerialization of invoices in Akuiteo. For details about how to dematerialize invoices for CHORUS PRO, refer to the *User Guide Dematerialized invoices - Chorus*.


## 3.1 ACTIVATING MANAGEMENT RULES

To be able to dematerialize invoices, you must activate the **CHORUS** application control to display the **CHORUS** tabs in sites and sales management objects. This control also displays dematerialization information in the modification screen of the billing address from a sales order.


The following items are optional:

- Activate the **CHORUS\_ORIGINAL\_OBIL** management rule to make the main attachment required in order to generate the XML file.
- Activate the **CHORUS\_BA\_COMPLEM** application control to display the following tags in the XML file: CopyIndicator, Note (Contact) and StandardItemIdentification.

## 3.2 CREATING A DEMATERIALIZATION CODE

- 1 In the setup window, go to **General setup > Sales > Dematerialization codes**.
- 2 From the search results, click on  in the action bar.
- 3 In the **Dematerialization setup** screen, fill in the following fields (only the fields required to create a new dematerialization code):

Field	Description
<b>Code</b>	Enter the <b>CHORUS</b> code.
<b>Label</b>	Enter a label for the dematerialization code to identify it.
<b>Storage directory</b>	Enter the path to the storage directory. This directory will be used to store the XML files generated for CHORUS.

- 4 Save the dematerialization code.  
 This code can be used to generate dematerialized invoices.

## 3.3 TRANSFERRING DEMATERIALIZED INVOICES AUTOMATICALLY TO CHORUS PRO

Akuiteo interfaces with CHORUS PRO in order to automatically transfer dematerialized invoices from the Application Desktop to the CHORUS PRO portal. This way, the user does not have to manually transfer dematerialized invoices to the CHORUS PRO portal.

### Important

In order to interface with CHORUS PRO, please contact your Akuiteo project manager.

### 3.3.1 Adding the Certigna certificate to the Tomcat server

The Java Runtime Environment (JRE) has a configuration file (keystore) that includes root certificates from the different renowned certification authorities. When a connection is established to another system using https, this list of certificates is used to validate the secured connection.

Some certification authorities are not included in the file provided by default with the JRE, so they must be added manually. To interface with CHORUS PRO, the Certigna certificate must be added to certify the CHORUS connections.

### Note

For SaaS customers, this certificate is added by Akuiteo.

#### Identifying the JRE's location

- 1 Connect to the server that hosts the Akuiteo environment.
- 2 Launch the Tomcat Manager for that environment.
- 3 From the **Java** tab, take note of the location of the JRE used by Tomcat.

#### Retrieving the Certigna's Racine certificate

- 1 Go to the Certigna's website: <https://www.certigna.com/autorite-crl>.
- 2 Download the Certigna ROOT CA's authority certificate: *certignarootca.der*.

#### Importing the certificate into the JRE's keystore

- 1 Launch the command prompt as an administrator.
- 2 Launch the following command:

```
[JRE_LOCATION]\bin\keytool.exe" -import -alias "certigna-root-ca" -keystore "[JRE_LOCATION]\lib\security\cacerts" -trustcacerts -file "[CERTIFICATE_LOCATION]\certignarootca.der" -storepass changeit
```

In this command, you must replace:

- **[JRE\_LOCATION]** with the location of the JRE used by Tomcat
- **[CERTIFICATE\_LOCATION]** with the location of the *certignarootca.der* certificate downloaded.

- 3 Restart the Tomcat server of the targeted Akuiteo environment to take into account the new certificate.

### 3.3.2 Setting up the Administration console

The automatic transfer of dematerialized invoices is set up from the Administration Console, from the menu **Configuration > Dematerialization**.

#### Notes

For SaaS customers, the setup of the Administration console is done by Akuiteo.

#### Reference

The login information to CHORUS PRO must be specified from the setup of the Application Desktop to enable you to use different CHORUS PRO accounts depending on each company. For more information, refer to [Setting up the login information per company \(p. 10\)](#).

- 1 Fill in the following fields to configure the connection to CHORUS PRO:

Field	Description
<b>Chorus Active</b>	Check this box to activate the connection to CHORUS PRO.
<b>Chorus Authentication URL</b>	Specify the URL to authenticate to CHORUS PRO: <ul style="list-style-type: none"><li>• <a href="https://sandbox-oauth.aife.economie.gouv.fr/api/oauth/token">https://sandbox-oauth.aife.economie.gouv.fr/api/oauth/token</a> for test environments,</li><li>• <a href="https://oauth.aife.economie.gouv.fr/api/oauth/token">https://oauth.aife.economie.gouv.fr/api/oauth/token</a> for production environments.</li></ul>
<b>Url</b>	Specify the URL to connect to CHORUS PRO: <ul style="list-style-type: none"><li>• <a href="https://sandbox-api.aife.economie.gouv.fr/">https://sandbox-api.aife.economie.gouv.fr/</a> for test environments,</li><li>• <a href="https://api.aife.economie.gouv.fr/">https://api.aife.economie.gouv.fr/</a> for production environments.</li></ul>

- 2 Fill in the following fields to configure the interface between Akuiteo and CHORUS PRO:

Field	Description
<b>Akuiteo user</b>	Specify the login of the Akuiteo's technical user.
<b>Akuiteo user's password</b>	Specify the password associated with the Akuiteo login.
<b>Akuiteo user's company code</b>	Specify the code of the company used for connection.
<b>Number of successive test</b>	The number of successive test runs enables to specify, in case of an error when



Field	Description
<b>runs in case of an error</b>	transferring dematerialized invoices, the number of times that Akuiteo will re-run the transfer.  By default, Akuiteo performs 3 successive test runs in case of an error.
<b>Time period in seconds between two successive test runs</b>	The time period between two successive test runs enables to specify, in seconds, the waiting period before another transfer is attempted in case of an error.  By default, Akuiteo waits 10 seconds between two successive test runs
<b>Maximum number of calls to Chorus per second</b>	Specify a maximum number of simultaneous calls to CHORUS PRO per second.  By default, there is a maximum of 20 calls per second. For a test environment, the number must be set on maximum 1 call per second.

#### Note

The CHORUS PRO portal has quotas for transferring dematerialized invoices:

- On the test environment: 5 queries per second with a maximum of 50,000 queries per day
- On the production environment: 20 queries per second with a maximum of 1 million queries per day

When these quotas are reached, the invoices can no longer be transferred. You should adapt the values in the **Number of successive test runs in case of an error** and **Time period in seconds between two successive test runs** fields if you regularly have errors when transferring invoices.

- 3 Click on **Save** for each field that is filled in or modified to take into account the value specified.

### 3.3.3 Setting up the PISTE APIs and the CHORUS PRO account

To interface CHORUS PRO with Akuiteo, you must have a PISTE application and connect it to your CHORUS PRO account.

#### Note

For a test environment, you must use the PISTE sandbox account and connect it to a CHORUS PRO qualification environment, in order to not transfer test invoices to a CHORUS production environment.

#### PISTE application

Each customer must set up their own PISTE application, and the following APIs must be activated:

- AIFE - Chorus Pro - Factures
- AIFE - Chorus Pro - Transverses

#### CHORUS PRO account

The PISTE application must be connected to the CHORUS PRO account.

### Note

You can only connect one PISTE application per CHORUS PRO account. If you have already connected your CHORUS PRO account to a PISTE application, you must set up this PISTE application as indicated above.

- 1 From the CHORUS PRO account, click on the menu **EDI and API connections > Manage API connection**.
- 2 Add a connection declaration by filling in the following information:

Field	Description
<b>PISTE application name</b>	Fill in the name of the PISTE application.
<b>Status</b>	Select <b>Active</b> from the drop-down list.
<b>Type of use</b>	Select <b>User of a publisher or operator solution</b> from the drop-down list.
<b>Editor / solution</b>	Fill in <b>AKUITEO</b> .


- 3 Validate the connection with the PISTE application.

### 3.3.4 Setting up the login information per company

DMF	090175 ADMINISTRATOR SETUP CHORUS
-----	-----------------------------------

The specific Chorus parameters enable you to specify the login information to CHORUS PRO, and therefore to automatically transfer the dematerialized invoices generated by Akuiteo to the CHORUS PRO portal.

The specific Chorus parameters are set up from **Special parameters > Chorus**.

This setup screen has several tabs: one tab for the \*\*\*\*\* company and other tabs for each of the companies set up in Akuiteo. These different tabs enable you to use different Piste accounts depending on each user's company. Click on  at the top right of the screen to enable the specific setup for a given company.

### Important

The setup on the \*\*\*\*\* company must not be used since each company must have a different CHORUS PRO account set up.

When dematerialized invoices are transferred to CHORUS PRO, Akuiteo checks the setup depending on each invoice's company:

- If a specific setup is enabled for the company, dematerialized invoices are automatically transferred to the CHORUS PRO portal using the specified connection parameters.
- If there is no specific setup for this company, Akuiteo uses the setup for \*\*\*\*\*.
- If the setup is not enabled, invoices are not transferred.

Click on  at the top right of a company's tab to disable a specific setup.

### Tip

To transfer the information specified on a company to another company, click on → at the top right of the screen then select the **Destination company** from the help list and click on **Validate**. The information is emptied in the source company and is added to the destination company.

To set up the login information to CHORUS PRO, fill in the following fields from the tab of the relevant company:

Field	Description
<b>Piste Client ID</b>	Specify the customer ID allowing the authentication to CHORUS PRO. This ID is retrieved from the PISTE account.
<b>Piste Secret</b>	Specify the client secret allowing the authentication to CHORUS PRO. This client secret is linked to the ID retrieved from the PISTE account.
<b>Chorus Login</b>	Specify the login from the CHORUS PRO technical account.
<b>Chorus Password</b>	Specify the password associated with the login from the CHORUS PRO technical account.
<b>ID</b>	Specify an ID to identify the invoices transferred to CHORUS using the APIs, in order to differentiate them from the invoices transferred manually.  This ID is free and is optional.

After saving the setup screen, click on the **Test** button from the relevant tab to test the connection to CHORUS PRO using the saved values.

### Important

The **Piste Client ID** and **Piste Secret** fields must be different between a production and a test environment.

## 3.3.5 Setting up the CheckFacturesClientChorus batch

When a dematerialized invoice is transferred automatically to CHORUS PRO, the CheckFacturesClientChorus batch enables to retrieve the CHORUS status in order to follow the progress of an invoice dematerialized from the Application Desktop. This status is stored on the invoice, in the **Chorus status** field of the **CHORUS** tab.

The status of a dematerialized invoice is no longer updated once the invoice has reached a status considered as final: A\_RECYCLER, SUSPENDUE, REJETEE or MISE\_EN\_PAIEMENT.

### Reference

For more information about how to set up the CheckFacturesClientChorus batch, refer to the *Advanced Setup Guide*.

# 4 Description of the generated XML file

This chapter describes the XML file generated by Akuteo that could be transferred via the CHORUS PRO platform.

The invoice type flow allowed by the CHORUS PRO platform and used by Akuteo to generate the XML files is the *(E1) Invoice structured flow*. All the data required to process the invoices in Chorus is included in the XML file generated by Akuteo. The data expected for this *(E1) Invoice structured flow* is described in the EDI appendix of the external specifications provided by Chorus.

An invoice structured flow can contain one or multiple XML files. Each file corresponds to an invoice. This flow can contain 500 invoices at most and has a total maximum size of 1 GB.

An XML file includes:

- Data structured in XML which represents the whole invoice;
- Attachments for additional documents linked to the invoice.

The types of invoices that are managed by Akuteo only correspond to the following billing types:

- **A1** (submission of the invoice by the supplier)
- **A2** (submission of the already paid invoice by the supplier)

## Note

The other types (**A4, A7, A8, A9, A10, A12, A13, A14, A22**) are not managed by Akuteo.

## 4.1 INFORMATION ABOUT FILE FORMAT

All the date fields must have the YYYY-MM-DD format.

The number fields are expressed by a 19-digit number, with a maximum of six decimals. The separator used for decimals is the "." point.

The following chapters are ordered the same way the tags are ordered in the XML file.

## 4.2 START OF FILE - HEADER

```
<?xml version="1.0" encoding="UTF-8"?>
<Invoice
xmlns:qdt="urn:oasis:names:specification:ubl:schema:xsd:QualifiedDatatypes-
2"
xmlns:ccts="urn:oasis:names:specification:ubl:schema:xsd:CoreComponentParame
ters-2"
xmlns:stat="urn:oasis:names:specification:ubl:schema:xsd:DocumentStatusCode-
1.0"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponent
s-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateCompo
nents-2"
xmlns:udt="urn:un:unece:uncefact:data:draft:UnqualifiedDataTypesSchemaModule
:2"
xmlns:cec="urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionCompo
nents-2">
  <cec:UBLExtensions>
    <cec:UBLExtension>
      <cec:ExtensionContent>
        <CategoryCode>A1</CategoryCode>
      </cec:ExtensionContent>
    </cec:UBLExtension>
  </cec:UBLExtensions>
  <cbc:UBLVersionID>2.1</cbc:UBLVersionID>
  <cbc:ID>F1001_2016001</cbc:ID>
  <cbc:CopyIndicator>false</cbc:CopyIndicator>
  <cbc:IssueDate>2016-04-25</cbc:IssueDate>
  <cbc:InvoiceTypeCode>380</cbc:InvoiceTypeCode>
  <cbc:DocumentCurrencyCode>USD</cbc:DocumentCurrencyCode>
  <cac:InvoicePeriod>
    <cbc:StartDate>2016-01-01</cbc:StartDate>
    <cbc:EndDate>2016-12-31</cbc:EndDate>
  </cac:InvoicePeriod>
```

Tag	Rules
<?xml version="1.0" encoding="UTF-8"?>	First line.
Invoice	Root tag.
CategoryCode	Framework (billing type): <ul style="list-style-type: none"><li>"A1" value if the amount left to be paid is different than zero.</li><li>"A2" value if the amount left to be paid is equal to zero.</li></ul>
ID	Invoice number.
CopyIndicator	Value always equals to "false" because Akuiteo never transfers duplicates of invoices to CHORUS PRO.  This tag is only displayed if the <b>CHORUS_BA_COMPLEM</b> application control is active.

Tag	Rules
IssueDate	<p>Invoice date.</p> <p>The date when the invoice was created must be before or the same as the date when the invoice is transmitted to Chorus.</p>
InvoiceTypeCode	<p>Authorized invoice type:</p> <ul style="list-style-type: none"> <li>• "380" value if it is an invoice</li> <li>• "381" value if it is a credit note</li> </ul>
DocumentCurrencyCode	Invoice currency
InvoicePeriod	<p><b>StartDate:</b> Period start date. This start date is determined by taking the earliest date among all the ones entered in the <b>Period from</b> field in the invoice lines.</p> <p><b>EndDate:</b> Period end date This end date is determined by taking the most recent date among all the ones entered in the <b>to</b> field in the invoice lines.</p> <p>The <code>InvoicePeriod</code> tag is not displayed if there is no period entered for the invoice.</p>

## 4.3 COMMITMENT

```
<cac:ContractDocumentReference>
  <cbc:ID>123456789123</cbc:ID>
  <cbc:DocumentTypeCode>Contract</cbc:DocumentTypeCode>
</cac:ContractDocumentReference>
<cac:ContractDocumentReference>
  <cbc:ID>123456789123</cbc:ID>
  <cbc:DocumentTypeCode>Market</cbc:DocumentTypeCode>
</cac:ContractDocumentReference>
<cac:OrderReference>
  <cbc:ID>1400000000</cbc:ID>
</cac:OrderReference>
```

Depending on the data entered, the following tags will be added. If none of these 3 alphanumeric values is entered, no tags are displayed.

Tag	Rules
ContractDocumentReference	Contract number.
ContractDocumentReference > ID	<b>Contract number</b> field in the <b>CHORUS</b> tab of sales management objects.
ContractDocumentReference > DocumentTypeCode	"Contract" value.
ContractDocumentReference > IssueDate	<b>Effective date</b> field in the <b>CHORUS</b> tab of sales management objects.  Tag only displayed if the date is entered.
ContractDocumentReference	Market number.
ContractDocumentReference > ID	<b>Market number</b> field in the <b>CHORUS</b> tab of sales management objects.
ContractDocumentReference > DocumentTypeCode	"Market" value.
ContractDocumentReference > IssueDate	<b>Effective date</b> field in the <b>CHORUS</b> tab of sales management objects.  Tag only displayed if the date is entered.
OrderReference	Commitment number.
OrderReference > ID	<b>Commitment number</b> field in the <b>CHORUS</b> tab of sales management objects.
OrderReference > IssueDate	<b>Effective date</b> field in the <b>CHORUS</b> tab of sales management objects.  Tag only displayed if the date is entered.

# 4.4 MAIN ATTACHMENT

```
<cac:AdditionalDocumentReference>
  <cbc:ID>PJP_FACTURE_FC1708523.pdf</cbc:ID>
  <cbc:DocumentTypeCode>application/pdf</cbc:DocumentTypeCode>
  <cbc:DocumentType>01</cbc:DocumentType>
  <cac:Attachment>
    <cbc:EmbeddedDocumentBinaryObject
mimeCode="application/zip">
      G4BALWUAQAVAAAARkFDVFVSRV9GQzeE3MDg1MjMucGRmnLoFWF1Ls
      jYMITiEAMHdXbbgFtzdLchm4
    </cbc:EmbeddedDocumentBinaryObject>
  </cac:Attachment>
</cac:AdditionalDocumentReference>
```

Tag	Rules
ID	Main attachment ID. The ID value contains:  PJP_(hard-coded) + file name + .pdf extension  PJP designates the main attachment.
DocumentTypeCode	File extension that must contain:  application/pdf (hard-coded)
DocumentType	File type that must contain the "01" value.  "01" = main attachment (invoice / credit note)
EmbeddedDocumentBinaryObject	Contains the attachment encoded in base64.  The mimeCode="application/zip" property is hard-coded.



# 4.5 ADDITIONAL ATTACHMENTS

```
<cac:AdditionalDocumentReference>
  <cbc:ID>PJC_INVOICE_FC1708523_Appendix1.pdf</cbc:ID>
  <cbc:DocumentTypeCode>application/pdf</cbc:DocumentTypeCode>
  <cbc:DocumentType>02</cbc:DocumentType>
  <cac:Attachment>
    <cbc:EmbeddedDocumentBinaryObject
mimeCode="application/zip">
      G4BALWUAQAVAAAARkFDVFVSRV9GQzE3MDg1MjMucGRmnLoFWF1Ls
      jYMITiEAMHdXbbgFtzdLchm4
    </cbc:EmbeddedDocumentBinaryObject>
  </cac:Attachment>
</cac:AdditionalDocumentReference>
```

Tag	Rules
ID	Additional attachment ID. The ID value contains:  PJC_(hard-coded) + file name + .pdf extension  PJC designates the additional attachment.
DocumentTypeCode	File extension that must contain:  application/pdf (hard-coded)
DocumentType	File type that must contain the "02" value.  "02" = additional attachment
EmbeddedDocumentBinaryObject	Contains the attachment encoded in base64.  The mimeCode="application/zip" property is hard-coded.

## 4.6 SUPPLIER

```

<cac:AccountingSupplierParty>
  <cac:Party>
    <cac:PartyIdentification>
      <cbc:ID schemeName="1">000000000000347</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name>REC005SUPPLIER</cbc:Name>
    </cac:PartyName>
    <cac:PartyLegalEntity>
      <cbc:RegistrationName>REC005SUPPLIER</cbc:Registrati
onName>

      <cac:RegistrationAddress>
        <cbc:StreetName>Fictitious
Street</cbc:StreetName>

        <cbc:CityName>PostOffice</cbc:CityName>
        <cbc:PostalZone>87016</cbc:PostalZone>
        <cac:Country>
          <cbc:IdentificationCode>US</cbc:Identifi
cationCode>

        </cac:Country>
      </cac:RegistrationAddress>
    </cac:PartyLegalEntity>
    <cac:PartyTaxScheme>
      <cbc:CompanyID>US93 533 335 303</cbc:CompanyID>
      <cac:TaxScheme>
        <cbc:TaxTypeCode>TVA
ENCAISSEMENT</cbc:TaxTypeCode>
      </cac:TaxScheme>
    </cac:PartyTaxScheme>
    <cac:Contact>
      <cbc:Note>REC005SUPPLIER</cbc:Note>
      <cbc:Note>SAS</cbc:Note>
      <cbc:Note>100000</cbc:Note>
    </cac:Contact>
  </cac:Party>
</cac:AccountingSupplierParty>

```

Tag	Rules
PartyIdentification > ID schemeName="1"	SIRET number of the invoice's company.
PartyName > Name	Company's full name.
PartyLegalEntity > RegistrationName	Company's full name.
PartyLegalEntity > RegistrationAddress > StreetName	Company's address. The company's address is required.
PartyLegalEntity > RegistrationAddress > PostalZone	Company's postcode.
PartyLegalEntity > RegistrationAddress > CityName	Company's city.
PartyLegalEntity > RegistrationAddress >	Company's country.

Tag	Rules
Country > IdentificationCode	
PartyTaxScheme > CompanyID	Company's EU VAT number.
PartyTaxScheme > TaxScheme > TaxTypeCode	VAT regime. The value is always "TVA ENCAISSEMENT".
Contact > Note	Company's full name. This tag is only displayed if the <b>CHORUS_BA_COMPLEM</b> application control is active.
Contact > Note	Company's legal form. This tag is only displayed if the <b>CHORUS_BA_COMPLEM</b> application control is active.
Contact > Note	Company's share capital. This tag is only displayed if the <b>CHORUS_BA_COMPLEM</b> application control is active.

## 4.7 DEBTOR (CUSTOMER)

```
<cac:AccountingCustomerParty>
  <cac:Party>
    <cac:PartyIdentification>
      <cbc:ID schemeName="1">12345678200051</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name>CLI001CUSTOMER</cbc:Name>
    </cac:PartyName>
    <cac:PostalAddress>
      <cbc:StreetName>Test Street 1</cbc:StreetName>
      <cbc:AdditionalStreetName>A2 building</cbc:AdditionalStreetName>
      <cbc:CityName>Test</cbc:CityName>
      <cbc:PostalZone>75000</cbc:PostalZone>
      <cac:Country>
        <cbc:IdentificationCode>US</cbc:Identification
Code>
      </cac:Country>
    </cac:PostalAddress>
  </cac:Party>
  <cac:AccountingContact>
    <cbc:ID>SERVICE_DEST_EJ_OBL</cbc:ID>
    <cbc:Name>Service name</cbc:Name>
  </cac:AccountingContact>
</cac:AccountingCustomerParty>
```

Tag	Rules
PartyIdentification > ID schemeName="1"	Billing site's SIRET number.
PartyName > Name	Billing site's full name.
PostalAddress > StreetName	Billing site's address 1.
PostalAddress > AdditionalStreetName	Billing site's address 2.
PostalAddress > CityName	Billing site's city.
PostalAddress > PostalZone	Billing site's postcode.
PostalAddress > Country > IdentificationCode	Billing site's country.
AccountingContact	Service code.  Block only displayed if the service code is entered.
AccountingContact > ID	<b>Service code</b> field in the <b>CHORUS</b> tab of the customer site.
AccountingContact > Name	<b>Service label</b> field in the <b>CHORUS</b> tab of the customer site.  Tag only displayed if the name is entered.

## 4.8 PAYMENT

```
<cac:PaymentMeans>
  <cbc:PaymentMeansCode listID="UN/ECE 4461 Subset"
    listAgencyID="NES" listAgencyName="Northern European Subset"
    listName="Payment Means">30</cbc:PaymentMeansCode>
  <cbc:PaymentDueDate>2016-04-26</cbc:PaymentDueDate>
  <cbc:PaymentChannelCode>IBAN</cbc:PaymentChannelCode>
  <cac:PayeeFinancialAccount>
    <cbc:ID>DK121234123412343412</cbc:ID>
    <cac:FinancialInstitutionBranch>
      <cac:FinancialInstitution>
        <cbc:ID>DKDKABCD</cbc:ID>
      </cac:FinancialInstitution>
    </cac:FinancialInstitutionBranch>
  </cac:PayeeFinancialAccount>
</cac:PaymentMeans>
```

Tag	Rules
PaymentMeansCode	Value determined by the value of the payment method: <ul style="list-style-type: none"> <li>• "01" if it is empty or for AU (Others)</li> <li>• "10" for ES (Cash)</li> <li>• "20" for CH or LC (Check)</li> <li>• "30" for VI or VD (Bank transfer)</li> <li>• "48" for CB (Credit card)</li> <li>• "49" for PR, CB or TR (Direct debit)</li> <li>• "97" if the amount left to be paid is equal to zero in cases where the invoice or credit note has already been paid.</li> </ul>
PaymentDueDate	First payment date scheduled in the due dates lines (unpaid lines).  Tag only displayed if the date is entered.
PaymentChannelCode	If the invoice originates from a contract: Details about the bank with the code specified in the <b>Cash journal</b> field of the contract.
PayeeFinancialAccount > ID	If the invoice does not originate from a contract: Details about the bank with the code specified in the <b>Cash journal</b> field of the customer site or, if not specified, with the <b>Bank journal</b> code specified on the next due date's line in the <b>Due dates</b> tab.
PayeeFinancialAccount > FinancialInstitutionBranch > ID	
	If none of these fields are specified: The tags are not filled in.

## 4.9 AMOUNTS

```
<cac:LegalMonetaryTotal>
  <cbc:TaxExclusiveAmount
currencyID="USD">83.4000</cbc:TaxExclusiveAmount>
  <cbc:TaxInclusiveAmount
currencyID="USD">100.0800</cbc:TaxInclusiveAmount>
  <cbc:PayableAmount currencyID="USD">100.0800</cbc:PayableAmount>
</cac:LegalMonetaryTotal>
```

The `CategoryCode` tag (see [Start of file - Header \(p. 13\)](#)) can be one of the two following values:

- "A1" value if the amount left to be paid is different than zero.
- "A2" value if the amount left to be paid is equal to zero.

Depending on this value, the `PayableAmount` doit toujours être inférieure ou égale à `TaxInclusiveAmount`.

Tag	Rules
<code>TaxExclusiveAmount</code> <code>currencyID="???"</code>	Currency (hard-coded in the tag) and value of the pre-tax amount.  If it is a credit note, the negative amount is changed into a positive one (credit notes have negative amounts in Akuiteo whereas they are positive in Chorus).
<code>TaxInclusiveAmount</code> <code>currencyID="???"</code>	Currency (hard-coded in the tag) and value of the after-tax amount.  If it is a credit note, the negative amount is changed into a positive one (credit notes have negative amounts in Akuiteo whereas they are positive in Chorus).
<code>PayableAmount</code> <code>currencyID="???"</code>	Currency (hard-coded in the tag) and amount of the unpaid due dates.  If it is a credit note, the negative amount is changed into a positive one (credit notes have negative amounts in Akuiteo whereas they are positive in Chorus).

## 4.10 TAX OVERVIEW

```
<cac:TaxTotal>
  <cbc:TaxAmount currencyID="EUR">5.2</cbc:TaxAmount>
  <cac:TaxSubtotal>
    <cbc:TaxableAmount currencyID="EUR">24</cbc:TaxableAmount>
    <cbc:TaxAmount currencyID="EUR">1.2</cbc:TaxAmount>
    <cbc:Percent>5</cbc:Percent>
    <cac:TaxCategory>
      <cac:TaxScheme>
        <cbc:TaxTypeCode>TVA</cbc:TaxTypeCode>
      </cac:TaxScheme>
    </cac:TaxCategory>
  </cac:TaxSubtotal>
  <cac:TaxSubtotal>
    <cbc:TaxableAmount currencyID="EUR">20</cbc:TaxableAmount>
    <cbc:TaxAmount currencyID="EUR">4</cbc:TaxAmount>
    <cbc:Percent>20</cbc:Percent>
    <cac:TaxCategory>
      <cac:TaxScheme>
        <cbc:TaxTypeCode>TVA</cbc:TaxTypeCode>
      </cac:TaxScheme>
    </cac:TaxCategory>
  </cac:TaxSubtotal>
</cac:TaxTotal>
```

All the invoice lines with a tax amount that is not empty are grouped by tax rate.

Each tax rate has its own separate `TaxSubtotal` tag.

Tag	Rules
<code>TaxAmount</code>	Total tax amount.
<code>TaxSubtotal</code>	Tag repeated as many times as the number of different tax rates.
<code>TaxSubtotal &gt; TaxableAmount</code>	Sum of the pre-tax amounts for all the relevant lines of the invoice.
<code>TaxSubtotal &gt; TaxAmount</code>	Sum of the tax amounts for all the relevant lines of the invoice.
<code>TaxSubtotal &gt; Percent</code>	Tax percentage for all the relevant lines of the invoice. This percentage is displayed without the "%" character.
<code>TaxSubtotal &gt; TaxCategory &gt; TaxScheme &gt; TaxTypeCode</code>	Task type. The "TVA" value is used.

## 4.11 LINES

```

<cac:InvoiceLine>
  <cbc:ID>1</cbc:ID>
  <cbc:InvoicedQuantity unitCode="EA">1</cbc:InvoicedQuantity>
  <cbc:LineExtensionAmount currencyID="USD">
    83.4000</cbc:LineExtensionAmount>
  <cac:Item>
    <cbc:Description>Description of the invoice
line</cbc:Description>
    <cbc:Name>Intellectual service</cbc:Name>
    <cac:StandardItemIdentification>
      <cbc:ID>SERVICES</cbc:ID>
    </cac:StandardItemIdentification>
    <cac:ClassifiedTaxCategory>
      <cbc:Percent>20</cbc:Percent>
    </cac:ClassifiedTaxCategory>
  </cac:Item>
  <cac:Price>
    <cbc:PriceAmount
currencyID="USD">83.4000</cbc:PriceAmount>
    <cbc:BaseQuantity unitCode="EA">1</cbc:BaseQuantity>
  </cac:Price>
</cac:InvoiceLine>
</Invoice>

```

Tag	Rules
ID	Line number.  The lines are numbered in ascending order.
InvoicedQuantity	Line quantity, always set to "1".
LineExtensionAmount currencyID="???"	Currency (hard-coded in the tag) and value of the line amount.  If it is a credit note, the negative amount is changed into a positive one (credit notes have negative amounts in Akuiteo whereas they are positive in Chorus).
Item > Description	Line's additional label (maximum 1,000 characters).
Item > Name	Line's label (maximum 40 characters).
Item > StandardItemIdentification > ID	Product reference (maximum 40 characters).  This tag is only displayed if the <b>CHORUS_BA_COMPLEM</b> application control is active.
Item > ClassifiedTaxCategory > Percent	Tax rate.
Price > PriceAmount currencyID="???"	Currency (hard-coded in the tag) and value of the line amount (unit price).  If it is a credit note, the negative amount is changed into a positive one (credit notes have negative amounts in Akuiteo whereas they are positive



Tag	Rules
	in Chorus).
Price > BaseQuantity	Quantity associated with the unit price, always set to "1".