



akuiteo
BUSINESS SOFTWARE

Setup Guide

DEMATERIALIZED INVOICES - CHORUS

Version 4.3

Revision number: 3

Published in: June 2021

Written by: Documentation team

Copyright (c) 2006-2021 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 Defining custom data | 7 |
| 3.3.1 Customer sites | 7 |
| 3.3.2 Sales business documents | 7 |
| 3.4 Transferring dematerialized invoices using the CHORUS PRO APIs | 8 |
| 3.4.1 Adding the Certigna certificate to the Tomcat server | 8 |
| 3.4.2 Setting up the Administration console | 9 |
| 3.4.3 Setting up the CHORUS PRO account | 11 |
| 3.4.4 Defining custom data for the CHORUS PRO APIs | 11 |
| 3.4.5 Setting up the CheckFacturesClientChorus batch | 12 |
| 4 Description of the generated XML file | 13 |
| 4.1 Information about file format | 13 |
| 4.2 Start of file - Header | 14 |
| 4.3 Commitment | 16 |
| 4.4 Main attachment | 17 |
| 4.5 Additional attachments | 18 |
| 4.6 Supplier | 19 |
| 4.7 Debtor (customer) | 21 |
| 4.8 Payment | 22 |
| 4.9 Amounts | 23 |
| 4.10 Tax overview | 24 |
| 4.11 Lines | 25 |

1 Preface

1.1 REVISIONS

| | |
|-------------------|---|
| Revision 3 | Published in June 2021 <ul style="list-style-type: none">• Updated Setting up the Administration console (p. 9) sub-chapter.• Added Adding the Certigna certificate to the Tomcat server (p. 8) sub-chapter.• Added Setting up the CHORUS PRO account (p. 11) sub-chapter. |
| Revision 2 | Published in November 2020 <ul style="list-style-type: none">• Added Transferring dematerialized invoices using the CHORUS PRO APIs (p. 8) sub-chapter.• Updated rules for the <code>PayeeFinancialAccount > ID et PayeeFinancialAccount > FinancialInstitutionBranch > ID</code> tags (see Payment (p. 22)). |
| Revision 1 | Published in November 2019 |

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:

- 2018/01/01 for intermediate-sized companies;
- 2019/01/01 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 the CHORUS PRO APIs.

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

The **VEN_RUB_LIB** management rule must be active to dematerialize invoices. This rule activates custom data in all the Sales module.


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.
- Activate the **CHORUS** application control to display dematerialization information in the modification screen of the billing address from a sales order.

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 |
|--------------------------|---|
| Code | Enter the CHORUS code. |
| Label | Enter a label for the dematerialization code to identify it. |
| Storage directory | 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 DEFINING CUSTOM DATA

3.3.1 Customer sites

- 1 In the setup window, go to **General setup > Sales > Custom data for customer sites**.
- 2 In the **Sequences** section, add sequences until the **Rank # 9**.
- 3 Create the following custom data:

| | |
|--------------|--|
| Service code | Label: Service code Section code: Alphanum01 Sequence: 9 Max. size: 100 |
| Service name | Label: Service name Section code: Alphanum02 Sequence: 9 Max. size: 100 |
| Cash journal | Label: Cash journal Section code: Alphanum03 Sequence: 9 |

3.3.2 Sales business documents

- 1 In the setup window, go to **General setup > Sales > Custom data for sales**.
- 2 In the **Sequences** section, add sequences until the **Rank # 9**.
- 3 Create the following custom data:

| | |
|-------------------|--|
| Contract number | Label: Contract number Section code: Alphanum01 Sequence: 9 Max. size: 50 |
| Market number | Label: Market number Section code: Alphanum02 Sequence: 9 Max. size: 50 |
| Commitment number | Label: Commitment number |

| | |
|----------------|---|
| | Section code: Alphanum03 Sequence: 9 Max. size: 50 |
| Cash journal | Label: Cash journal Section code: Alphanum04 Sequence: 9 Max. size: 50 |
| Effective date | Label: Effective date Section code: Date01 Sequence: 9 |

3.4 TRANSFERRING DEMATERIALIZED INVOICES USING THE CHORUS PRO APIS

Akuiteo can be used with the CHORUS PRO APIs 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 use the CHORUS PRO APIs, please contact your Akuiteo project manager.

3.4.1 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 use the CHORUS PRO APIs, 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.

3.4.2 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.

The setup of the Administration console is done for a single CHORUS PRO structure.

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

| Field | Description |
|----------------------------------|---|
| Chorus Active | Check this box to activate the connection to the CHORUS PRO APIs. |
| Chorus Authentication URL | Specify the authentication URL to the CHORUS PRO APIs, provided by Akuiteo. |
| Customer ID | Specify the customer ID allowing the authentication to the CHORUS PRO APIs. This ID is retrieved from the PISTE account and is provided by Akuiteo. |
| Secret | Specify the password allowing the authentication to the CHORUS PRO APIs. This password is linked to the ID retrieved from the PISTE account and is provided by Akuiteo. |
| Url | Specify the URL to connect to the CHORUS PRO APIs, provided by Akuiteo. |

| Field | Description |
|--|--|
| Chorus Login | Specify the login from the CHORUS PRO technical account. |
| Chorus password | Specify the password associated with the login from the CHORUS PRO technical account. |
| ID | Specify the ID of the CHORUS PRO structure linked to the technical account. |
| List of valid dematerialization CODES | Specify the CHORUS_DEMATERIALIZED code, which can be used by the CHORUS PRO APIs. |

Important

The **Chorus Authentication URL**, **Customer ID**, **Secret** and **Url** fields, with values provided by Akuiteo, must be different if it is a production or a test environment.

- 2 Fill in the following fields to configure the use of the CHORUS PRO APIs in Akuiteo:

| Field | Description |
|--|---|
| Akuiteo user | Specify the login of the Akuiteo's technical user. |
| Akuiteo user's password | Specify the password associated with the Akuiteo login. |
| Akuiteo user's company code | Specify the code of the company used for connection. |
| Number of successive test runs in case of an error | <p>The number of successive test runs enables to specify, in case of an error when transferring dematerialized invoices, the number of times that Akuiteo will re-run the transfer.</p> <p>By default, Akuiteo performs 3 successive test runs in case of an error.</p> |
| Time period in seconds between two successive test runs | <p>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.</p> <p>By default, Akuiteo waits 10 seconds between two successive test runs</p> |

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.
- 4 Click on the **Test** button to test the connection to the CHORUS PRO APIs using the values specified.

3.4.3 Setting up the CHORUS PRO account

In order to use the CHORUS PRO APIs with Akuiteo, you must add the PISTE application from your CHORUS PRO account.

Note

You must add as many PISTE applications as there are environments on which you wish to use the CHORUS PRO APIs. If you are using a production environment and another one for testing, you must add two PISTE applications.

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

| Field | Description |
|-------------------------------|--|
| PISTE application name | Fill in the name of the PISTE application. This application is created by Akuiteo so its name must be provided by Akuiteo. |
| Status | Select Active from the drop-down list. |
| Type of use | Select User of a publisher or operator solution from the drop-down list. |
| Editor / solution | Fill in AKUITEO . |

- 3 Validate the creation of the PISTE application.

3.4.4 Defining custom data for the CHORUS PRO APIs

- 1 In the setup window, go to **General setup > Sales > Custom data for sales**.
- 2 Create the following custom data:

| | |
|------------------|---|
| Transaction ID | Label: Transaction ID Section code: Alphanum06 Sequence: 9 Max. size: 50 |
| Invoice status | Label: Invoice status Section code: Alphanum07 Sequence: 9 Max. size: 50 |
| Transaction date | Label: Transaction date Section code: Date02 Sequence: 9 |

For each dematerialized invoice that is transferred using the CHORUS PRO APIs, the information retrieved from CHORUS PRO are then stored in these custom data on the Application Desktop:

- The transaction ID is retrieved for each dematerialized invoice that is successfully transferred to the CHORUS PRO portal.
- The transaction date indicates when the dematerialized invoice was successfully transferred to the CHORUS PRO portal.
- The invoice status refers to the CHORUS PRO status and enables to follow the progress of each dematerialized invoice from the Application Desktop. To update these statuses, refer to [Setting up the CheckFacturesClientChorus batch \(p. 12\)](#).

These custom data are automatically filled in by Akuiteo and cannot be modified manually. However, if the transmission status of an invoice is updated manually and set to **1 - To transmit**, these fields are emptied.

3.4.5 Setting up the CheckFacturesClientChorus batch

When a dematerialized invoice is transferred using the CHORUS PRO APIs, the CheckFacturesClientChorus batch enables to retrieve the CHORUS status in order to follow the progress of a dematerialized invoice 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 Akuiteo that could be transferred via the CHORUS PRO platform.

The invoice type flow allowed by the CHORUS PRO platform and used by Akuiteo 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 Akuiteo. 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 Akuiteo 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 Akuiteo.

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">"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. |
| 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 CHORUS_BA_COMPLEM 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"> • "380" value if it is an invoice • "381" value if it is a credit note |
| DocumentCurrencyCode | Invoice currency |
| InvoicePeriod | <p>StartDate: Period start date. This start date is determined by taking the earliest date among all the ones entered in the Period from field in the invoice lines.</p> <p>EndDate: Period end date This end date is determined by taking the most recent date among all the ones entered in the to 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 | Custom data for sales business documents (Sequence 9, Alphanum01). |
| ContractDocumentReference > DocumentTypeCode | "Contract" value. |
| ContractDocumentReference > IssueDate | Custom data for sales business documents (Sequence 9, Date01). Tag only displayed if the date is entered. |
| ContractDocumentReference | Market number. |
| ContractDocumentReference > ID | Custom data for sales business documents (Sequence 9, Alphanum02). |
| ContractDocumentReference > DocumentTypeCode | "Market" value. |
| ContractDocumentReference > IssueDate | Custom data for sales business documents (Sequence 9, Date01). Tag only displayed if the date is entered. |
| OrderReference | Commitment number. |
| OrderReference > ID | Custom data for sales business documents (Sequence 9, Alphanum03). |
| OrderReference > IssueDate | Custom data for sales business documents (Sequence 9, Date01). 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">
      G4BALWUAQAVAAAARkFDVFVSRV9GQzE3MDg1MjMucGRmnLoFWF1Ls
      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 CHORUS_BA_COMPLEM application control is active. |
| Contact > Note | Company's legal form. This tag is only displayed if the CHORUS_BA_COMPLEM application control is active. |
| Contact > Note | Company's share capital. This tag is only displayed if the CHORUS_BA_COMPLEM 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 | Custom data for sites (Sequence 9, Alphanum01). |
| AccountingContact > Name | Custom data for sites (Sequence 9, Alphanum02). 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"> • "01" if it is empty or for AU (Others) • "10" for ES (Cash) • "20" for CH or LC (Check) • "30" for VI or VD (Bank transfer) • "48" for CB (Credit card) • "49" for PR, CB or TR (Direct debit) • "97" if the amount left to be paid is equal to zero in cases where the invoice or credit note has already been paid. |
| PaymentDueDate | First payment date scheduled in the due dates lines (unpaid lines). Tag only displayed if the date is entered. |
| PaymentChannelCode | IBAN number. This number is required. |
| PayeeFinancialAccount > ID | Custom data for the invoice (Sequence 9, Alphanum04). If the custom data is not specified, the code of the cash journal specified in the billing site's custom data (Sequence 9, Alphanum03) is used and enables to determine the IBAN number. The tag value concatenates the 3 following fields: PAY_CODE, BPR_CLE_IBAN, BPR_NUMCPTE_IBAN. |
| PayeeFinancialAccount > FinancialInstitutionBranch > ID | Custom data for the invoice (Sequence 9, Alphanum04). If the custom data is not specified, the code of the cash journal specified in the billing site's custom data (Sequence 9, Alphanum03) is used and enables to determine the BIC number. |

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 \(p. 14\)](#)) 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 > TaxableAmount</code> | Sum of the pre-tax amounts for all the relevant lines of the invoice. |
| <code>TaxSubtotal > TaxAmount</code> | Sum of the tax amounts for all the relevant lines of the invoice. |
| <code>TaxSubtotal > Percent</code> | Tax percentage for all the relevant lines of the invoice. This percentage is displayed without the "%" character. |
| <code>TaxSubtotal > TaxCategory > TaxScheme > 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 CHORUS_BA_COMPLEM 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". |