

# TRADESHIFT®

## commerce eXtensible Markup Language

### User Guide



### Seller Onboarding

November 2019

Version: 0.2

## Table of Contents

<b>Introduction</b>	<b>3</b>
Platform basics	3
Tradeshift as a multi-tenant platform	3
Network and identity	4
Integration basics	4
Business Firewall	5
Security and compliance	5
Auditing standards	5
General account security properties	6
Communication security	6
Web security	6
Integration services security	6
API-level security	6
Document level security	6
Tradeshift infrastructure	7
Scalability	7
Quality of service	7
Global reach	7
cXML, an XML Implementation	8
cXML Capabilities	8
cXML Sample Files Library	8
<b>on your mark, get set, go</b>	<b>9</b>
Master specification spreadsheets	9
Overview	9
Mandatory fields	10
Currency Code	11
Country Identification Code	11
Tax Category ID	11
Tax Scheme ID	11
Unit Of Measure Code	12
TS PartyID schemes	12
TS PartyID schemes names	12
Binary Object Mime Code	12
TS Document Types	12
TS Districts	12

<b>cXML Mapping for direct integration</b>	<b>12</b>
cXML to TSUBL spreadsheet	13
TSUBL to cXML spreadsheet	13
<b>cXML Structure &amp; Syntax</b>	<b>14</b>
Header	14
Request	15
Segment mapping examples	25
InvoiceNumber	25
InvoiceSenderPartyID	25
Sender Assigned (as identifier)	26
ReceiverPartyID (TSGLI, TSLEID and VAT)	27
Invoice line Order reference	27
InvoicePaymentDueDate	28
InvoiceAllowanceAmount	28
InvoiceChargeAmount	29
With shipping charges	29
With other charges	29

## Introduction

This document is the starting point for the technical integration between Tradeshift and our users. Its goal is to accelerate the integration by providing a high level summary, tying together other available documents, filling in any gaps, and making best practice recommendations.

### *Platform basics*

Tradeshift is a collaborative platform that supports any processes involving the exchange of structured, electronic business documents between businesses of any size, for internal employee collaboration and process support. As a platform it allows customers and 3rd parties to integrate or extend it in order to support new processes, collaborations and document exchanges.

At the heart of the platform is the network which allows businesses to build networks of sellers, customers, employees and other collaborators which they can use to speed up processes, reduce the noise, increase the quality of data, and use the combination of the network and real-time data to explore new business opportunities.

Technically, Tradeshift is realized as a massively scalable cloud-based SaaS (Software-as-a-Service) platform, providing rich integration points for both customers, and sellers, for in-house and cloud-based ERP-systems, external SaaS solutions, as well as in-UI application extensions.

### Tradeshift as a multi-tenant platform

Tradeshift is realized as a true multi-tenant platform. Each company on Tradeshift corresponds to a 'tenant' and is a separate administrative unit with its own users, data storage, localization and extensibility. Nobody outside of the account, unless explicitly authorized by the account holder, can access data or functions within the account. Any form of collaboration or message exchange involves explicitly exchanging the relevant business document over the network to other Tradeshift tenants.

The concept of a "strong tenant isolation" refers to the ability to see each account as a truly separate unit. This is what allows the platform to control the location of data storage, to manage security isolation of each account, to ensure strong access separation when the API is used for integrations or other extensions of the basic platform, for example in the form of apps.

## Network and identity

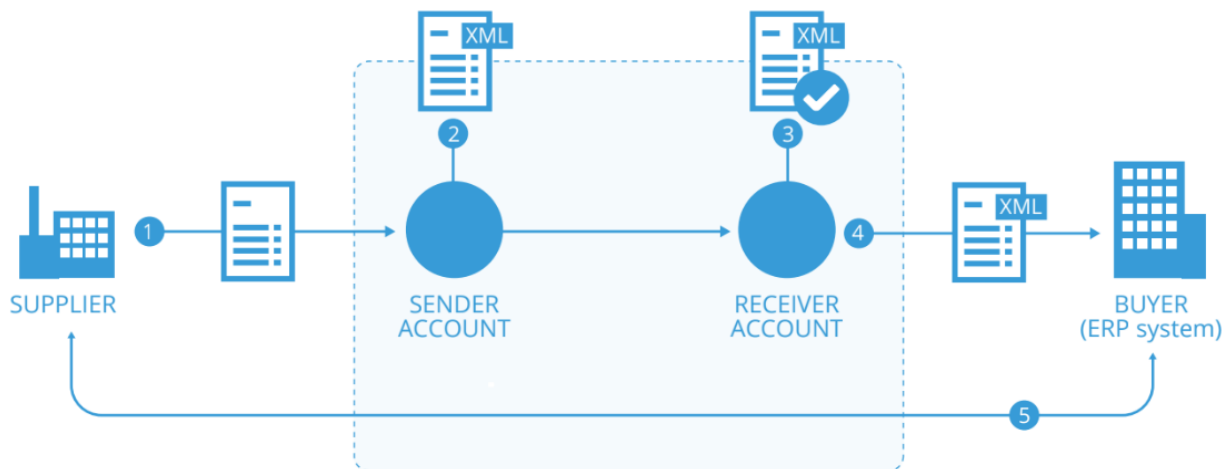
The Tradeshift network deals with the identities of companies and users within the network. For companies, a minimum set of information must be available for them to be visible and available in the network for other companies to request connections with them. Employees may also be visible on the network, for example for the purpose of directing invoices at specific employees.

The concept of the network is based on the principle that each participant controls their part of the network — which companies and individuals they wish to collaborate with.

## Integration basics

Different types of Integrations with Tradeshift are supported whether you are a large enterprise, small business, external service provider, or a developer of extensions and apps that are distributable through our apps marketplace.

Tradeshift architecture is based on accounts. A seller and a buyer will each have an account and can designate users to maintain that account. Sellers and Buyers must both accept their connection requests to allow for document exchange.



1. A seller (sender) creates and sends an invoice to a buyer (receiver) through Tradeshift, using the integration channels.
2. The invoice is converted to XML (UBL) and stored on the seller's account on Tradeshift (sender account).
3. A copy of the invoice is transmitted to the buyer's account (receiver account), together with any supporting documents (such as attachments, signatures and source documents in case a conversion was involved).
4. The invoice is either pushed to or pulled by the buyer's ERP system for further processing.

5. The statuses of the invoice (e.g. sent, accepted, paid) is communicated between the two parties.

However, in the end, Tradeshift converts all of these document formats to a canonical XML format which is used as the exchange between the different parties — before possibly being converted again to some other format upon delivery. Both formats (the original format as well as the XML format that Tradeshift creates regardless of the source format) will be available to the parties.

## Business Firewall

Tradeshift business firewall is set up on the Buyer's side and contains static and dynamic validations that control what values can be set for specific fields in various e-documentation.

For example: A business firewall rule might prevent an invoice if the PO number referenced on the invoice does not contain a 9 digit alpha-numeric number. In certain cases a list of values can be added to dynamically validate against that list. Business firewall is not exposed directly to the seller. Instead, when they go to create a document to send to that Buyer, they will get validation errors and specific language forbidding them from sending the document with values that do not adhere to the business firewalls.

## *Security and compliance*

### Auditing standards

Tradeshift is ISAE 3402 certified (the successor to SAS70). The controls of this report cover the following security-related aspects:

- Integrity of business messages
- Authenticity of origin
- Archiving
- Access restrictions to data and functions

Tradeshift builds on secure cloud platform with data centers that conform to the following security, infrastructure and related standards:

- SOC 1/SSAE 16/ISAE 3402
- SOC 2 + SOC 3
- ISO 27001 certified
- FedRAMP Compliant Cloud Service Provider (CSP)
- PCI DSS level 1 compliant
- DoD 5220.22-M / NIST 800-88 for decommissioning of hardware

## General account security properties

The platform features strong tenant isolation, meaning that per-user storage is considered isolated units. For example, when sending an invoice from one account to another, a copy of that invoice is created and sent to the receiving account, rather than this document being shared across the two account. This allows for scenarios where documents are physically stored in different geographical locations with different privacy or other regulatory requirements.

## Communication security

### Web security

For users of the web application, all communication is protected with SSL using EV certificates ('extended validation' certificates). Tradeshift supports a number of authentication options ranging from username/password with email verification to SSO solutions including SAML 2.0. The use of SSO allows users to take full control over authentication policies and rely on in-house identity providers and local administrator policies.

### Integration services security

Integration services include API-based integrations (see below for security properties of the API) as well as SFTP and FTPS, both with forwarding.

### API-level security

At the API level, 2-legged OAuth together with SSL. OAuth signs messages to ensure integrity, and SSL provides privacy.

Multiple access keys may be issued to businesses or integrators on the platform, and each tenant may grant or revoke access to individual key holders. This allows the same scheme to be used for both enterprise integrations (where a large business will need to access its own account), and SaaS integration scenarios, where a single SaaS provider needs large number of tenants to grant access to tenant data.

## Document level security

Business documents such as orders and invoices are signed when they are persisted to ensure the integrity of the documents. Tradeshift keeps redundant, encrypted backups of all data. Tradeshift maintains a dedicated security layer in the platform for ensuring tenant data isolation, as reflected in our ISAE 3402 controls.

In applicable jurisdictions or whenever required by customers, signatures provided by certified 3rd party signature services are available in addition to the basic integrity signature.

## Tradeshift infrastructure

### Scalability

Tradeshift has been developed as a massively scalable SaaS cloud platform. The underlying cloud platform gives us the elastic scalability required to keep operational costs to a minimum and ensures the availability of resources at all times. The platform architecture gives us the ability to react to changes in traffic patterns with just-in-time resource provisioning, and to support global scalability through a multi-datacenter strategy.

Scalability is ensured by three main properties of the platform:

- The Cloud Platform: The underlying cloud platform ensures that the platform is able to scale elastically both up and down with the demand, both regarding storage and service throughput
- The stateless, REST-based architecture is the enabler of on-demand driven up- and down-scaling, caching and failover
- A dynamic “continuous deployment” framework developed by Tradeshift that allows rapid deployment of new services across regions
- Real-time and historical monitoring of traffic patterns

### Quality of service

Tradeshift is committed to provide high quality service through its operations. Tradeshift's ISAE 3402 certification certifies the operational maturity of the Tradeshift platform and infrastructure with respect to monitoring, uptime, availability, failover, storage, integrity and general operational processes.

Tradeshift is committed to give integration partners and customers real-time insight into the operational status of our infrastructure and operations.

### Global reach

The RESTful platform architecture with its stateless services and strong tenant isolation is the foundation for its scalability. Together with the deployment framework developed by Tradeshift, the architecture allows Tradeshift to utilize processing power and storage in different regions, close to the users of the service, ensuring excellent response times and low latency.



## *cXML, an XML Implementation*

cXML (commerce eXtensible Markup Language) is a meta-markup language used to create syntaxes for languages, created by Ariba in 1999, intended for communication of business documents between procurement applications, e-commerce hubs and sellers. cXML is based on XML and provides formal XML schemas for standard business transactions, allowing programs to modify and validate documents without prior knowledge of their form.

XML has a structure similar to HTML (HyperText Markup Language), which is an implementation of SGML, XML's parent metalanguage. Applications can extract and use data from XML documents more easily than from HTML documents, however, because XML data is tagged according to its purpose. XML contains only data, while HTML contains both data and presentation information.

Each cXML document is constructed based on XML Document Type Definitions (DTDs). Acting as templates, DTDs define the content model of a cXML document, for example, the valid order and nesting of elements, and the data types of attributes.

## *cXML Capabilities*

cXML allows buying organizations, sellers, service providers, and intermediaries to communicate using a single, standard, open language. It's a well-defined, robust language designed specifically for B2B e-commerce, and represents the choice of high volume buying organizations and sellers. cXML transactions consist of documents, which are simple text files containing values enclosed by predefined tags. Most types of cXML documents are analogous to hardcopy documents traditionally used in business.

## *cXML Sample Files Library*

Along with this user guide, a package with sample files is also available for download from <https://tradeshift.com/integrate/>. Below, our users will be able to learn what type of files can be found in these zip archives and how to use them.

⚠ All documents uploaded to Tradeshift platform are transformed in TS UBL format and the original file will also be sent as an attachment. Therefore all formats supported by Tradeshift for direct integration with the platform will provide specification spreadsheets describing transformation the conversion from the respective foreign format to natively supported UBL.

Each document type supported by Tradeshift in cXML format has examples, a specification file, and a guidance document explaining the transformation from cXML to TS UBL. These files are:

- Master\_INV\_Specification, Master\_CRN\_Specification and Master\_ORD\_Specification spreadsheets define all the elements present in the respective document type along with their canonical TSUBL names. These master spreadsheets are also providing a business description of the field indicating the format required for each element with examples.
- cXML\_2\_TSUBL\_INV, cXML\_2\_TSUBL\_CRN, cXML\_2\_TSUBL\_ORD are structural documents used to identify the hierarchy of the cXML fields and their XPath correspondence.
- TSUBL\_2\_cXML\_INV, TSUBL\_2\_cXML\_CRN, TSUBL\_2\_cXML\_ORD are structural specification files for the relevant type of document. The structure required for the relevant document type is listed within these documents, along with mapping indications between the canonical fields and cXML fields.
- Various sample files for each document type which can be used for guidance.

## on your mark, get set, go

Master specification spreadsheets are providing an overview of the various fields, their use in specific scenarios and, where applicable, defines limitations. This is the starting point for the integration, here are described the canonicals expected formats and much more.

### *Master specification spreadsheets*

Master specification spreadsheet includes a variety of tabs which are referenced as described further below in this section.

#### Overview

This tab provides an overview of the specific fields and, is unique for the document type reviewed by user (Invoice, Credit Note etc). Following tabs within the spreadsheet are general within respective package no matter which document type is reviewed.

Here's a description of what each column represents:

- **TSUBL use:**
  - "1" marks the field as mandatory for the Tradeshift platform for creating a valid document.
  - "0..n" marks the field as optional, and repeatable for n number of times. The field should be present if is filled with information.

- "1..n" marks the field as mandatory and repeatable for n number of times if needed.
- "0..1" marks the field as optional and not repeatable.
- ⚠ Even if a specific field is not mandated by the format specification, it might be marked as required by the Buyer business firewall rules. Please review [Business Firewall](#) section for more details.
- **Canonical Name:** Represents the variable name where Tradeshift platform will store the value provided in the TSUBL field.
- **Business description:** Provides a short description of its use.
- **Format:** If applicable, shows the field format, accepted by the Tradeshift platform
- **Example:** This shows an example of the field can be used.
- **Comments:** Any additional comments, if applicable.

## Mandatory fields

All documents transacted on Tradeshift platform, regardless of the type, hold the following fields with mandatory information:

- **Number:** An identifier for the Invoice assigned by the Creditor
- **Date:** The date assigned by the Creditor on which the Invoice was issued
- **CurrencyCode:** The currency in which the Document is presented
- **SenderPartyID:** The ID of the invoice sender (seller)
- **SenderPartyIDScheme:** The ID of the invoice sender (seller)
- **SenderPartyName:** The Name of the invoice sender (seller)
- **SenderCountryCode:** An identifier for the Country (seller)
- **ReceiverPartyID:** The ID of the invoice receiver (buyer)
- **ReceiverPartyIDScheme:** The scheme for the ID of the invoice receiver (buyer)
- **ReceiverPartyName:** The Name of the invoice receiver (buyer)
- **ReceiverCountryCode:** An identifier for the Country (buyer)
- **TotalTaxAmount:** The total tax amount for a tax type; the sum of each of the tax subtotals for each tax category within the tax type
- **SubTotalTaxableAmount:** The net amount to which the tax percent (rate) is applied to calculate the tax amount
- **SubTotalTaxAmount:** The amount of tax stated explicitly
- **SubTotalTaxCategoryID:** Identifies the tax category
- **SubTotalTaxCategoryPercent:** The tax rate for the category, expressed as a percentage
- **SubTotalTaxSchemeID:** Identifies the tax scheme
- **TotalLineAmount:** The total of line amounts net of tax and settlement discounts
- **TaxTotal:** The total tax amount of the invoice
- **Total:** The total amount to be paid
- **PayableAmount:** The total amount to be paid
- **LineID:** Identifies the line within the document

- **LineQuantity:** The quantity (of Items) on the line
- **LineUnitCode:** The unitcode related to the quantity
- **LineExtensionAmount:** The total amount for the Invoice Line, including Allowance and Charges but net of taxes
- **LineTotalTaxAmount:** The total tax amount for a tax type; the sum of each of the tax subtotals for each tax category within the tax type
- **LineSubTotalTaxableAmount:** The net amount to which the tax percent (rate) is applied to calculate the tax amount
- **LineSubTotalTaxAmount:** The amount of tax stated explicitly
- **LineSubTotalTaxCategoryID:** Identifies the tax category
- **LineSubTotalTaxCategoryPercent:** The tax rate for the category, expressed as a percentage
- **LineSubTotalTaxSchemeID:** Identifies the tax scheme
- **LineItemDescription:** Free-form field that can be used to give a text description of the item
- **LineUnitPrice:** The unit price of the item, including Allowance and Charges but net of taxes
- **LineBaseQuantity:** The actual quantity to which the price applies, defaults to 1
- **LineBaseQuantityUnitCode:** The unitcode related to the quantity
- **LineOrderableUnitFactorRate:** The factor by which the base price unit can be converted to the orderable unit, defaults to 1

If a document is uploaded without any of these fields, it will be rejected by platform until respective file is uploaded with all mandatory information.

## Currency Code

All currency codes accepted by Tradeshift platform are listed here in [ISO 4217](#) standard.

## Country Identification Code

All country codes accepted by Tradeshift platform are listed here in [ISO 3166](#) standard.

## Tax Category ID

All tax categories accepted by Tradeshift platform are listed here in in UN/ECE 5305 standard.

## Tax Scheme ID

This tab contains all tax categories accepted by Tradeshift platform in UN/ECE 5153 standard.

## Unit Of Measure Code

This tab contains a list of tax categories accepted by Tradeshift platform in UN/ECE rec 20 standard.

## TS PartyID schemes

All party identification schemes accepted by Tradeshift Platform are listed here.

## TS PartyID schemes names

This tab contains all the accepted values for parameter TS\_cac:PartyIdentification/cbc:ID/@schemeName. Note that only one of the values presented here are accepted by the Tradeshift Platform.

## Binary Object Mime Code

This tab contains all the mime types standardised by IANA MIME Media Types. Note that only one of the values presented here are accepted by the Tradeshift Platform

## TS Document Types

This tab contains all the Tradeshift Extensions implemented over the UBL for DocumentReference field for storing specific industry information. Note that only one of the values presented here is accepted by the Tradeshift Platform.

## TS Districts

This tab contains the list with districts that are accepted by the Tradeshift Platform. The list is composed out of the district code, the description and the country code for which the district is implemented. Note that only one of the values presented here are accepted by the Tradeshift Platform

## cXML Mapping for direct integration

The format specific spreadsheets, listed below, define the mapping from the specific format to the canonical business elements. These can be used as a reference in order to understand mapping from, and to, cXML format in order to meet Tradeshift requirements.

- cXML\_2\_TSUBL\_INV.xls
- cXML\_2\_TSUBL\_CRN.xls
- TSUBL\_2\_cXML\_INV.xls
- TSUBL\_2\_cXML\_CRN.xls

## *cXML to TSUBL spreadsheet*

This document contains the cXML fields along with the associated TSUBL canonical names. Its purpose is to identify the supported cXML field names and associated XPath of a particular field to determine the exact location which needs to be populated.

While **SourceXPath** provides details to the correct fields or subfields to use, **Mapping rule (formula)** on the other hand, explains the expected formatting and the logic/functions that has been applied to the field/subfields.

Similarly to [Master Specification Spreadsheet](#), **TSUBL use** column marked with:

- "1" indicates a mandatory field.
- "0..n" indicates optional fields which could be repeated for n number of times.
- "1..n" indicates mandatory fields which could be repeated for n number of times as required.
- "0..1" indicates optional fields which cannot be repeated.

## *TSUBL to cXML spreadsheet*

This document contains the cXML structure along with the associated TSUBL canonical names. Its purpose is to provide a visual representation of the cXML structure along with the specific location of cXML fields.

While **cXML structure** provides a visual representation of the cXML structure along with the specific location of cXML fields, **Mapping rule (formula)** on the other hand, explains the expected formatting and the logic/functions that has been applied to the field/subfields.

Similarly to [Master Specification Spreadsheet](#), **TSUBL use** column marked with:

- "1" indicates a mandatory field.
- "0..n" indicates optional fields which could be repeated for n number of times.
- "1..n" indicates mandatory fields which could be repeated for n number of times as required.
- "0..1" indicates optional fields which cannot be repeated.

## cXML Structure & Syntax

The cXML structure is divided into two main components, Header and Request.

**Header** - Contains authentication information and addressing.

**Request** - Contains specific information to be passed.

```
<cXML>
  <Header>
    Header information
  </Header>
  <Request>
    Request information
  </Request>
</cXML>
```

Though both components are mandatory for the cXML to be valid, most of the data for TSUBL will be mapped from the Request segment of the cXML. To identify that the file is a cXML, the <cXML> tag will be present at the beginning of the message after the xml version. The cXML structure consists of two properties:

1. **Element** (Names begin with an uppercase)
  - Used to describe properties of traditional business documents (ie. Header, InvoiceDetailRequest)
  - Used to describe subdivisions and relationships of those subdivisions (ie. PostalAddress that comprises of street name, city, postal code and country)
2. **Attribute** (Names begin with a lowercase)
  - Used to modify elements or provide context (ie. invoiceID, invoiceDate)

⚠ Further below is described the content of sample file cXML\_Invoice\_01.xml. In order to review another document type in a similar format (ie cXML\_CreditNote\_01.xml), look for the first sample file of respective type.

### Header

The following example shows the Header element:

```
<Header>
  <From>
    <!-- From element is mandatory. For an Invoice it identifies the supplying organisation.
    <Credential domain="DUNS">
      <!-- A Credential element with domain="DUNS" is mandatory.
      <Identity>123456711</Identity>
      <!--Identity is the DUNS number and is mandatory.
    </Credential>
    <Credential domain="Zanzibar">
      <Identity>bb02520f-7546-4329-b0c4-0011259c9b8a</Identity>
    </Credential>
```

```

</From>

<To>
<!-- To element is mandatory. For an Invoice it identifies the buying organisation.
      <Credential domain="DUNS">
<!-- A Credential element with domain="DUNS" is mandatory.
      <Identity>111110501</Identity>
<!-- Identity is the DUNS number and is mandatory.
      </Credential>
      <Credential domain="Zanzibar">
<!-- Identity is the Zanzibar number and is mandatory.
      <Identity>ff66e12b-db53-4589-958e-0011259c9bb8</Identity>
      </Credential>
</To>

<Sender>
<!-- Sender element is mandatory.
<!-- It identifies the organisation/system that has sent the document.
<!-- It is often the same organisation/system as identified in the From credentials.
      <Credential domain="NetworkUserID">
<!-- A Credential element is mandatory.
      <Identity>ProcServe Supplier Portal</Identity>
      </Credential>
      <UserAgent>ProcServe Supplier Portal</UserAgent>
<!-- UserAgent element is mandatory.
</Sender>

</Header>

```

The From and To elements are synonymous with From and To in SMTP mail messages; they are the logical source and destination of the messages. Sender is the party that opens the HTTP connection and sends the cXML document.

Sender contains the Credential element, which allows the receiving party to authenticate the sending party. This credential allows strong authentication without requiring a public-key end-to-end digital certificate infrastructure. Only a username and password need to be issued by the receiving party to allow the sending party to perform Requests. When the document is initially sent, Sender and From are the same, However, if the cXML document travels through e-commerce network hubs, the Sender element changes to indicate current sending party.

## Request

Only one Request element is allowed for each cXML envelope element. The two main Request elements supported by Tradeshift are:

- OrderRequest (Purchase Orders)
- InvoiceDetailRequest (Invoice & Credit Notes)

The main attribute of the Request element is deploymentMode which indicates if it is a test request or a production request. Only "production" (default) or "test" can be populated in this attribute.



<Request>

<InvoiceDetailRequest>

<InvoiceDetailRequestHeader invoiceID="1280" purpose="standard" operation="new" invoiceDate="">

<InvoiceDetailHeaderIndicator isVatRecoverable="yes"/>

<InvoiceDetailLineIndicator isTaxInLine="yes"/>

<!-- InvoiceDetailLineIndicator element is mandatory.

<!-- isTaxInLine is optional. Presence or absence indicates whether tax details are present or absent

<!-- within each InvoiceDetailItem element. Always present and populated as "yes" by Procsolve.

<InvoicePartner>

<!-- InvoicePartner elements are optional.

<!-- If present each must have a Contact with a unique role attribute value.

<!-- 4 standard role values are recognised: "issuerOfInvoice", "soldTo", "remitTo" and "billTo".

<!-- The deprecated "from" role is also recognised by ProcServe in place of "issuerOfInvoice".

<!-- All 5 InvoicePartner elements are populated by ProcServe. "from" is identical to "issuerOfInvoice"

<Contact role="issuerOfInvoice" addressID="35">

<!-- Contact element is mandatory if InvoicePartner element is present.

<!-- role is mandatory.

<!-- "issuerOfInvoice" indicates this is for the party which is issuing the invoice.

<!-- addressID is optional.

<Name xml:lang="en-GB">Standard Garments Ltd.</Name>

<!-- Name element is mandatory if Contact element is present.

<!-- It contains the name of the organisation.

<!-- xml:lang is optional and ignored by ProcServe. Populated as "en-GB" by Procsolve.

<PostalAddress name="Accounts">

<!-- PostalAddress element and all elements within it are optional.

<!-- However, complete lack of content is not advisable.

<!-- name is optional. It is an identifier for the address, not the name of a person.

<DeliverTo>Tellula Stevens</DeliverTo>

<!-- DeliverTo is optional.

<!-- A maximum of 2 DeliverTo elements per PostalAddress are recognised.-->

<!-- If present, the first should contain the name of the contact.

<!-- The second may be another identifier e.g. department

<DeliverTo>Accounts Department</DeliverTo>

<Street>Standard House</Street>

<!-- Maximum of 3 Street elements per PostalAddress are recognised.-->

<Street>Oak Tree Close</Street>

<City>Guildford</City>

<State>Surrey</State>

<PostalCode>GU1 4TX</PostalCode>

<Country isoCountryCode="GB">United Kingdom</Country>

<!-- isoCountryCode is optional. If present, must be the 2 character ISO 3166 code.

</PostalAddress>

```

    <Email name="default">tellula.stevens@standardgarments.com</Email>
    <Phone name="direct">
    <TelephoneNumber>
      <CountryCode isoCountryCode="GB">44</CountryCode>
      <AreaOrCityCode>01483</AreaOrCityCode>
      <Number>703025</Number>
    </TelephoneNumber>
    </Phone>
  </Contact>
  <IdReference domain="vatID" identifier="GB223344556"/>
  <!-- IdReference element with domain="vatID" must be provided for
  role="issuerOfInvoice" (or
  <!-- "from") if the supplier is VAT registered in the UK, to comply with HMRC
  regulations.
  <!-- It must also be provided for role="billTo" when trading between EU countries.
  <!-- Otherwise it is optional.
  <!-- domain is mandatory. Populated as "vatID" by Proserve.
  <!-- identifier is mandatory, and when domain="vatID", is the VAT Registration
  Number of the
  <!-- supplier for role="issuerOfInvoice" (or from) and of the buyer for role="billTo".
</InvoicePartner>

<InvoicePartner>
<!-- Refer also to comments within the first InvoicePartner element

  <Contact role="soldTo" addressID="874">
  <!-- "soldTo" indicates this is for the party which the goods or services were sold to.

    <Name xml:lang="en-GB">Great SouthEastern plc</Name>
    <PostalAddress name="Purchasing">
      <DeliverTo>Bob Turner</DeliverTo>
      <DeliverTo>Floor 30 (Purchasing Dept.)</DeliverTo>
      <Street>Zanzibar House</Street>
      <Street>35 High Street</Street>
      <City>Gravesend</City>
      <State>Kent</State>
      <PostalCode>GV3 6FT</PostalCode>
      <Country isoCountryCode="GB">United Kingdom</Country>
    </PostalAddress>
    <Email>bob.turner@gse.co.uk</Email>
    <Phone name="work">
      <TelephoneNumber>
        <CountryCode
        isoCountryCode="GB">44</CountryCode>
        <AreaOrCityCode>0123</AreaOrCityCode>
        <Number>123458</Number>
      </TelephoneNumber>
    </Phone>
  </Contact>
</InvoicePartner>

<InvoicePartner>
<!-- Refer also to comments within the first InvoicePartner element

  <Contact role="remitTo" addressID="35">
  <!-- "remitTo" indicates this is for the party to which payment should be made.

    <Name xml:lang="en-GB">Standard Garments Ltd.</Name>

    <PostalAddress name="Accounts">
      <DeliverTo>Tellula Stevens</DeliverTo>
      <DeliverTo>Accounts Department</DeliverTo>

```

```

        <Street>Standard House</Street>
        <Street>Oak Tree Close</Street>
        <City>Guildford</City>
        <State>Surrey</State>
        <PostalCode>GU1 4TX</PostalCode>
        <Country isoCountryCode="GB">United Kingdom</Country>
    </PostalAddress>

    <Email name="default">tellula.stevens@standardgarments.com</Email>

    <Phone name="direct">
        <TelephoneNumber>
            <CountryCode
                isoCountryCode="GB">44</CountryCode>
            <AreaOrCityCode>01483</AreaOrCityCode>
            <Number>703025</Number>
        </TelephoneNumber>
    </Phone>

</Contact>

</InvoicePartner>

<InvoicePartner>
<!-- Refer also to comments within the first InvoicePartner element

    <Contact role="billTo" addressID="874">
<!-- "soldTo" indicates this is for the party which is being invoiced.

        <Name xml:lang="en-GB">Great SouthEastern plc</Name>

        <PostalAddress name="Purchasing">
            <DeliverTo>Bob Turner</DeliverTo>
            <DeliverTo>Floor 30 (Purchasing Dept.)</DeliverTo>
            <Street>Zanzibar House</Street>
            <Street>35 High Street</Street>
            <City>Gravesend</City>
            <State>Kent</State>
            <PostalCode>GV3 6FT</PostalCode>
            <Country isoCountryCode="GB">United Kingdom</Country>
        </PostalAddress>

        <Email>bob.turner@gse.co.uk</Email>

        <Phone name="work">

            <TelephoneNumber>
                <CountryCode
                    isoCountryCode="GB">44</CountryCode>
                <AreaOrCityCode>0123</AreaOrCityCode>
                <Number>123458</Number>
            </TelephoneNumber>

        </Phone>

    </Contact>

    <IdReference domain="vatID" identifier="GB998877665"/>

</InvoicePartner>

```

```

<PaymentTerm payInNumberOfDays="30"/>
<!-- PaymentTerm elements are optional.
<!-- payInNumberOfDays is mandatory.

<PaymentTerm payInNumberOfDays="7">
    <Discount>
    <!-- Discount element with in a PaymentTerm element is optional.
    <!-- It indicates an early settlement discount.

        <DiscountPercent percent="2.50"/>
        <!-- DiscountPercent is the only element recognised within Discount.
        <!-- percent is mandatory and must be numeric.

    </Discount>
</PaymentTerm>

<Comments>We appreciate prompt payment.</Comments>
<!-- Comments element is optional.

<Extrinsic name="BuyersCodeForSupplier">STA501</Extrinsic>
<!-- Extrinsic elements are optional. If present, each must have a name attribute.

<Extrinsic name="OrderPayloadID">XML1138973517001</Extrinsic>

</InvoiceDetailRequestHeader>

<InvoiceDetailOrder>
<!-- InvoiceDetailOrder element is mandatory.

    <InvoiceDetailOrderInfo>
    <!-- InvoiceDetailOrderInfo element is mandatory.

        <OrderReference orderID="HA5102305">
        <!-- OrderReference element is optional for cXML. But omitting it is not avisable.
        <!-- orderID is mandatory. This is the buying organisation's order reference.

            <DocumentReference payloadID="XML1138973517001"/>
            <!-- DocumentReference element is mandatory.
            <!-- payloadID is mandatory for cXML. It is the payloadID of the order.
            <!-- A blank payloadID is also acceptable to ProcServe systems

        </OrderReference>

        <SupplierOrderInfo orderID=""/>
        <!-- SupplierOrderInfo element is optional.
        <!-- orderID is mandatory. This is the supplying organisation's order reference.

    </InvoiceDetailOrderInfo>

    <InvoiceDetailItem invoiceLineNumber="1" quantity="1">

        <UnitOfMeasure>PK</UnitOfMeasure>
        <UnitPrice>
            <Money currency="GBP">34.84</Money>
    
```

```

</UnitPrice>

<InvoiceDetailItemReference lineNumber="1">
<!-- InvoiceDetailItemReference element is mandatory.
<!-- lineNumber is mandatory and must be numeric or empty. It is the
order line number.

    <ItemID>
    <!-- ItemID element is optional.

        <SupplierPartID>670800</SupplierPartID>
        <!-- SupplierPartID element is mandatory.
        <!-- It is the supplying organisation's reference to the
        goods or service.

    </ItemID>

    <Description xml:lang="en-GB">Mepore 6cm x 7cm Dressing
(Multipack)</Description>
    <!-- Description element is optional. It is a description of the
    goods or service.

</InvoiceDetailItemReference>

<SubtotalAmount>
<!-- SubtotalAmount element is optional.

    <Money currency="GBP">34.84</Money>
    <!-- Money element is mandatory.
    <!-- It is the value of the item line (excluding tax) i.e. quantity x
    unit price.
    <!-- It must be numeric, maximum 2 decimals.
    <!-- currency is mandatory and must be the 3 character ISO 4217
    code e.g. "GBP" or "USD".

</SubtotalAmount>

<Tax>
<!-- Tax element here is optional but is recommended for VAT.

    <Money currency="GBP">5.94</Money>
    <!-- Money element is mandatory and is the total value of tax on
    this item line.
    <!-- It must be numeric, maximum 2 decimals.
    <!-- currency is mandatory and must be the 3 character ISO 4217
    code e.g. "GBP" or "USD".

    <Description xml:lang="en-GB">Total Line Item
Tax</Description>
    <!-- Description element is mandatory for cXML but ignored by
    ProcServe.
    <!-- Populated as "Total Line Item Tax" by ProcServe.

    <TaxDetail purpose="tax" category="vat" percentageRate="17.5"
taxPointDate="2009-07-25">
    <!-- TaxDetail element is optional.
    <!-- TaxDetail elements other than the first within a Tax element
    are ignored by ProcServe.
    <!-- purpose is optional. Populated as "tax" by Procserve.
    <!-- category is mandatory for cXML and should have the value

```

"vat" (EU) or "sales" (USA).  
 <!-- percentageRate is optional but, for VAT, omitting it is not advisable.  
 <!-- taxPointDate here is optional and is ignored by ProcServe.  
 <!-- If present, it should be in the format YYYY-MM-DDThh:mm:ss or YYYY-MM-DD.

<TaxableAmount>

<!-- TaxableAmount element is optional.

<Money currency="GBP">34.84</Money>

<!-- Money element is mandatory.

<!-- It is the value (for this item line) which is liable to this type/rate of tax.

<!-- It must be numeric, maximum 2 decimals.

<!-- currency is mandatory and must be the 3 character ISO 4217 code.

</TaxableAmount>

<TaxAmount>

<!-- TaxAmount element is mandatory.

<Money currency="GBP">5.94</Money>

<!-- Money element is mandatory.

<!-- It is the value of this type/rate of tax on this item line.

<!-- It must be numeric, maximum 2 decimals.

<!-- currency is mandatory and must be the 3 character ISO 4217 code.

</TaxAmount>

<TaxLocation xml:lang="en-GB">GB</TaxLocation>

<!-- TaxLocation element is optional. Populated as the 2 char. ISO 3166 code by ProcServe

<!-- xml:lang is optional and ignored by ProcServe. Populated as "en-GB" by Procserve.

<Description

xml:lang="en-GB">Standard</Description>

<!-- Description element is optional.

<!-- Preferred values for VAT are "Standard", "Reduced", "Zero", "Exempt".

<!-- xml:lang is optional and ignored by ProcServe. Populated as "en-GB" by Procserve.

</TaxDetail>

</Tax>

<GrossAmount>

<!-- GrossAmount element is optional.

<Money currency="GBP">40.78</Money>

<!-- Money element is mandatory.

<!-- It is the value of the item line (including tax).

<!-- It must be numeric, maximum 2 decimals.

<!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or "USD".

</GrossAmount>

<NetAmount>

<!-- NetAmount element is optional.

<Money currency="GBP">40.78</Money>

<!-- Money element is mandatory.

<!-- It is the value of the item line (including tax) after any discount.

<!-- Note that this is net of discount, NOT net of tax

<!-- It must be numeric, maximum 2 decimals.

<!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or "USD".

</NetAmount>

<Comments>Use by September 2010</Comments>

<!-- Comments element is optional.

<Extrinsic name="UNSPSC">42311005</Extrinsic>

<!-- Extrinsic elements are optional. If present, each must have a name attribute.

<!-- Certain name attributes are recognised and interpreted as follows:

<!-- name="UNSPSC" is interpreted as the UNSPSC classification code for the invoice item.

<!-- name="ContractReference" identifies the contract under which this purchase was made

<!-- Other Extrinsics are noted but not interpreted as having a particular significance by ProcServe.

<Extrinsic name="ContractReference">N45-302</Extrinsic>

</InvoiceDetailItem>

</InvoiceDetailOrder>

<InvoiceDetailSummary>

<!-- InvoiceDetailSummary element is mandatory.

<SubtotalAmount>

<!-- SubtotalAmount element is mandatory.

<Money currency="GBP">34.84</Money>

<!-- Money element is mandatory.

<!-- It is the value of all the item lines (excluding tax) i.e. sum of (quantity x unit price).

<!-- It must be numeric, maximum 2 decimals.

<!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or "USD".

</SubtotalAmount>

<Tax>

<!-- Tax element here is mandatory.

```
<Money currency="GBP">5.94</Money>
<!-- Money element is mandatory.
<!-- It is the total value of all tax on this invoice.
<!-- It must be numeric, maximum 2 decimals.
<!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or "USD".
```

```
<Description xml:lang="en-GB">Total Tax</Description>
<!-- Description element is mandatory for cXML but ignored by ProcServe.
<!-- Populated as "Total Tax" by ProcServe.
```

```
<TaxDetail purpose="tax" category="vat" percentageRate="17.5"
taxPointDate="2009-07-25">
<!-- TaxDetail element is optional for cXML but at least one must be present for UK
invoices.
<!-- Multiple TaxDetail elements may be present.
<!-- purpose is optional. Populated as "tax" by Procserve.
<!-- category is mandatory for cXML and should have the value "vat" (EU) or "sales"
(USA).
<!-- percentageRate is optional but, for VAT, omitting it is not advisable.
<!-- taxPointDate here is mandatory for UK invoices.
<!-- It should be in the format YYYY-MM-DDThh:mm:ss or YYYY-MM-DD.
<!-- Other formats may or may not be be interpreted correctly.
```

```
<TaxableAmount>
<!-- TaxableAmount element is optional for cXML but must be present for
UK invoices.
```

```
<Money currency="GBP">34.84</Money>
<!-- Money element is mandatory.
<!-- It is the value of items in this invoice liable to this type/rate
of tax.
<!-- It must be numeric, maximum 2 decimals.
<!-- currency is mandatory and must be the 3 character ISO 4217
code e.g. "GBP" or "USD".
```

```
</TaxableAmount>
```

```
<TaxAmount>
<!-- TaxAmount element is mandatory.
```

```
<Money currency="GBP">5.94</Money>
<!-- Money element is mandatory.
<!-- It is the value of this type/rate of tax.
<!-- It must be numeric, maximum 2 decimals.
<!-- currency is mandatory and must be the 3 character ISO 4217
code e.g. "GBP" or "USD".
```

```
</TaxAmount>
```

```
<TaxLocation xml:lang="en-GB">GB</TaxLocation>
<!-- TaxLocation element is optional. Populated as the 2 character ISO
3166 code by ProcServe
<!-- xml:lang is optional and ignored by ProcServe. Populated as "en-GB"
by Procserve.
```

```
<Description xml:lang="en-GB">Standard</Description>
```



<!-- Description element is optional.  
 <!-- Preferred values for VAT are "Standard", "Reduced", "Zero", "Exempt".  
 <!-- xml:lang is optional and ignored by ProcServe. Populated as "en-GB"  
 by Procserve.

</TaxDetail>

</Tax>

<SpecialHandlingAmount>

<!-- SpecialHandlingAmount element is optional.  
 <!-- It is advisable to use an InvoiceDetailItem element rather than this when tax is applicable.

<Money currency="GBP">0.00</Money>

<!-- Money element is mandatory.  
 <!-- It is the value of any charges other than shipping (e.g. express processing).  
 <!-- It must be numeric, maximum 2 decimals.  
 <!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or  
 "USD".

</SpecialHandlingAmount>

<ShippingAmount>

<!-- ShippingAmount element is optional.  
 <!-- It is advisable to use an InvoiceDetailItem element rather than this when tax is applicable.

<Money currency="GBP">0.00</Money>

<!-- Money element is mandatory.  
 <!-- It is the value of any freight/delivery/shipping charges.  
 <!-- It must be numeric, maximum 2 decimals.  
 <!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or  
 "USD".

</ShippingAmount>

<GrossAmount>

<!-- GrossAmount element is optional.

<Money currency="GBP">0.00</Money>

<!-- Money element is mandatory.  
 <!-- It is the value of the invoice (including tax and charges).  
 <!-- It must be numeric, maximum 2 decimals.  
 <!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or  
 "USD".

</GrossAmount>

<NetAmount>

<!-- NetAmount element is mandatory.

<Money currency="GBP">45.78</Money>

<!-- Money element is mandatory.  
 <!-- It is the value of the invoice (including tax and charges) after any discount.  
 <!-- Note that this is net of discount, NOT net of tax  
 <!-- It must be numeric, maximum 2 decimals.  
 <!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or  
 "USD".

</NetAmount>

```

    <DueAmount>
      <!-- DueAmount element is optional and ignored by ProcServe.

              <Money currency="GBP">45.78</Money>
              <!-- Money element is mandatory.
              <!-- It is the amount due for payment (e.g. after taking a deposit into account).
              <!-- It must be numeric, maximum 2 decimals.
              <!-- currency is mandatory and must be the 3 character ISO 4217 code e.g. "GBP" or
              "USD".

    </DueAmount>

  </InvoiceDetailSummary>

</InvoiceDetailRequest>

</Request>

```

## Segment mapping examples

Below are few examples describing how to map respective information in a cXML document.

### InvoiceNumber

- Canonical Name: InvoiceNumber
- SourceXPath:  
cXML/Request/InvoiceDetailRequest/InvoiceDetailRequestHeader/@invoiceID
- Example File: cXML\_Invoice\_01.xml

```

<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailRequestHeader invoiceID="12345">
        </InvoiceDetailRequestHeader>
      </InvoiceDetailRequest>
    </Request>
  </cXML>

```

Based on the SourceXPath, for each node separated by a '/', the node (Element node) is nested until the last node (Attribute node), is reached and populated. In the screenshot above, the path is denoted by the red arrow before finally reaching the attribute node, invoiceID. Therefore, to populate the Invoice Number, the tag is nested as follows, cXML > Request > InvoiceDetailRequest > InvoiceDetailRequestHeader > invoiceID as seen in the screenshot above.

### InvoiceSenderPartyID

- Canonical Name: InvoiceSenderPartyID

- SourceXPath:  
cXML/Request/InvoiceDetailRequest/InvoiceDetailRequestHeader/InvoicePartner[Contact/@role = 'issuerOfInvoice']/IdReference[@domain='vatID']/@identifier
- Example File: cXML\_Invoice\_01.xml

```
<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailRequestHeader>
        <InvoicePartner>
          <Contact role="issuerOfInvoice">
          </Contact>
          <IdReference domain="vatID" identifier="1234567890">
          </IdReference>
        </InvoicePartner>
      </InvoiceDetailRequestHeader>
    </InvoiceDetailRequest>
  </Request>
</cXML>
```

For this example, the SourceXPath consists of predicates (denoted by '['). Nodes contained in these predicates are XPath conditions that must be met before proceeding. In the screenshot above, the XPath conditions are coloured in red. Therefore, based on the SourceXPath, the path is as follows cXML > Request > InvoiceDetailRequest > InvoiceDetailRequestHeader > InvoicePartner > [Contact > @role = 'issuerOfInvoice'] > IdReference > [@domain='vatID'] > identifier.

## Sender Assigned (as identifier)

- Canonical Name: InvoiceSenderPartyID
- SourceXPath:  
cXML/Request/InvoiceDetailRequest/InvoiceDetailRequestHeader/InvoicePartner[Contact/@role='issuerOfInvoice']/IdReference[@domain='SenderAssigned']/@identifier

```
<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailRequestHeader>
        <InvoicePartner>
          <Contact addressID="" role="issuerOfInvoice">
          </Contact>
          <IdReference domain="vatID" identifier="1234567890"/>
          <IdReference domain="legalID" identifier=""/>
          <IdReference domain="SenderAssigned" identifier="abc"/>
        </InvoicePartner>
      </InvoiceDetailRequestHeader>
    </InvoiceDetailRequest>
  </Request>
</cXML>
```

SenderAssigned is extra identifier, which can be used when supplier has multiple accounts under their Master Account structure and where two or more branches have Shared VAT. In this case system will need additional identifier to make sure that invoice routing works

fine and invoice will land on the correct supplier account.

## ReceiverPartyID (TSGLI, TSLEID and VAT)

- Canonical Name: InvoiceReceiverPartyID
- SourceXPath:  
 cXML/Request/InvoiceDetailRequest/InvoiceDetailRequestHeader/InvoicePartner[Contact/@role = 'billTo']/IdReference[@domain='tsgliID']/@identifier  
 cXML/Request/InvoiceDetailRequest/InvoiceDetailRequestHeader/InvoicePartner[Contact/@role = 'billTo']/IdReference[@domain='tsleID']/@identifier  
 cXML/Request/InvoiceDetailRequest/InvoiceDetailRequestHeader/InvoicePartner[Contact/@role = 'billTo']/IdReference[@domain='vatID']/@identifier
- Example File: cXML\_Invoice\_01.xml

```
<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailRequestHeader>
        <InvoicePartner>
          <Contact role="billTo">
          </Contact>
          <IdReference domain="tsgliID" identifier="ABCDEF"/>
          <IdReference domain="tsleID" identifier="12345"/>
          <IdReference domain="vatID" identifier="1234567890"/>
        </InvoicePartner>
      </InvoiceDetailRequestHeader>
    </InvoiceDetailRequest>
  </Request>
</cXML>
```

The SourceXPath is similar to mapping InvoiceSenderPartyID, however the XPath conditions are different, specifically with Contact > @Role="billTo" as opposed to "issueroofInvoice". Therefore, based on the SourceXPath, the path is as follows cXML > Request > InvoiceDetailRequest > InvoiceDetailRequestHeader > InvoicePartner > [Contact > @role = 'billTo'] > IdReference > [@domain='vatID'] > identifier. The value in @domain will be based on the identifier type used.

## Invoice line Order reference

- Canonical Name: InvoiceLineOrderReference
- SourceXPath:cXML/Request/InvoiceDetailRequest/InvoiceDetailOrder/InvoiceDetailOrderInfo[orderID = ""]
- Example File: cXML\_Invoice\_03

```
<cXML>
  <Request>
    <InvoiceDetailRequest>
```

```

        <InvoiceDetailOrder>
          <InvoiceDetailOrderInfo>
            <OrderReference orderID="12345">
              <DocumentReference payloadID=""/>
            </OrderReference>
          </InvoiceDetailOrderInfo>
        </InvoiceDetailOrder>
      </InvoiceDetailRequest>
    </Request>
  </cXML>

```

## InvoicePaymentDueDate

- Mapping Sheet: cXML\_2\_TSUBL\_INV.xls
- Canonical Name: InvoicePaymentDueDate
- SourceXPath:
- cXML/Request/InvoiceDetailRequest/InvoiceDetailSummary/Tax/TaxDetail[Description = 'Standard']/@paymentDate

```

<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailSummary>
        <Tax>
          <TaxDetail paymentDate="2019-01-01">
            <Description>Standard</Description>
          </TaxDetail>
        </Tax>
      </InvoiceDetailRequestHeader>
    </InvoiceDetailRequest>
  </Request>
</cXML>

```

## InvoiceAllowanceAmount

- Mapping Sheet: cXML\_2\_TSUBL\_INV.xls
- Canonical Name: InvoiceAllowanceAmount
- SourceXPath:
- cXML/Request/InvoiceDetailRequest/InvoiceDetailSummary/InvoiceDetailDiscount/Money

```

<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailSummary>
        <InvoiceDetailDiscount>
          <Money>100.00</Money>
        </InvoiceDetailDiscount>
      </InvoiceDetailSummary>
    </InvoiceDetailRequest>
  </Request>
</cXML>

```

## InvoiceChargeAmount

- Mapping Sheet: cXML\_2\_TSUBL\_INV.xls
- Canonical Name: InvoiceChargeAmount
- SourceXPath:
  - cXML/Request/InvoiceDetailRequest/InvoiceDetailSummary/ShippingAmount/Money
  - cXML/Request/InvoiceDetailRequest/InvoiceDetailSummary/SpecialHandlingAmount/Money

### With shipping charges

```
<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailSummary>
        <ShippingAmount>
          <Money>50.00</Money>
        </ShippingAmount>
      </InvoiceDetailSummary>
    </InvoiceDetailRequest>
  </Request>
</cXML>
```

### With other charges

```
<cXML>
  <Request>
    <InvoiceDetailRequest>
      <InvoiceDetailSummary>
        <SpecialHandlingAmount>
          <Money>50.00</Money>
        </SpecialHandlingAmount>
      </InvoiceDetailSummary>
    </InvoiceDetailRequest>
  </Request>
</cXML>
```