d.velop

d.velop eInvoice converter: Administrator

Table of Contents

1. d.velop eInvoice converter

1.1. Basic information about the application

This chapter contains product information and general information.

1.1.1. About d.velop eInvoice converter

d.velop eInvoice converter converts different e-invoice formats into PDF files so that all the information can be displayed. The application is also used in the context of invoice processing in d.velop document reader to prepare the invoice information from the e-invoice for recognition.

1.2. Installation and uninstallation

This chapter contains information about installing the application.

1.2.1. System requirements

Please refer to the central system requirements for d.velop products (on-premises). You can find deviating or more extensive system requirements in the documentation.

1.2.2. Installing d.velop eInvoice converter

You install the software exclusively using d.velop software manager. If an application is required for different products, the corresponding software packages are also installed automatically.

For further information on installing the software, see the d.velop software manager manual.

1.2.3. Installing updates for d.velop eInvoice converter

You can only update the software using d.velop software manager.

For further information about updating the software, see the d.velop software manager manual.

1.2.4. Rolling back an installation of d.velop eInvoice converter

You can restore an earlier version of the software that you installed with d.velop software manager. During this process, the software is only reset to a previous version.

For further information on rolling back to an earlier version, see the d.velop software manager manual.

1.2.5. Uninstalling d.velop eInvoice converter

The software you installed using d.velop software manager can only be uninstalled with d.velop software manager. If the software to be uninstalled has dependencies with other software packages, you must resolve these conflicts accordingly.

For further information on uninstallation, see the d.velop software manager manual.

1.2.6. Enabling the default ports for d.velop eInvoice converter

By default, the port for d.velop eInvoice converter is determined dynamically in the range from 7071 to 8070. However, you can specify your own port range in the installation directory in the **config.json** file. The range must contain at least two free ports.

1.3. Configuration

This chapter contains information about configuring d.velop eInvoice converter. Open the configuration of the conversion tool via the Config app. You can find the settings in the **Invoice processing** area under **einvoice converter**.

You can use the local directory monitoring configuration only with on-premises deployment.

1.3.1. Converting e-invoices into a universal format

You can view the supported e-invoice formats in the configuration. Each format is identified by a unique XML namespace. The different XML formats are converted into a universal XML format using an XML style sheet. A PDF file for viewing the e-invoice is generated from the universal format. This allows you to generate a uniform PDF view from different e-invoice formats. You can find more information about the process for conversion to a PDF file in the following chapter.

Downloading a style sheet

In the **Options** column, you can download a style sheet for an e-invoice format using the download icon (**Download default or custom style sheet**). For each format, there is a standard style sheet, which you can download as an XSLT file. If you have already uploaded a custom style sheet, you can also download this style sheet. In this case, a dialog opens first in which you can select between the options **Default** or **Custom** to download.

Uploading and deleting a custom style sheet

You can select and upload an XSLT file using the upload icon (**Upload custom style sheet**). The style sheet is not validated during the upload process. The uploaded style sheet is activated automatically. In the **Custom** column, you can control whether the default style sheet or a custom style sheet is used. You can use the trash can icon (**Delete custom style sheet**) to delete the custom style sheet completely and activate the default style sheet for future conversions to the universal format.

In the **Default** column, you can see whether there is a standard style sheet for an e-invoice format.

In the following example, the invoice number is converted into the uniform format for two different formats:

UN/CEFACT Cross Industry Invoice:

```
<xr:InvoiceNumber>
    <xsl:value-of select="/rsm:CrossIndustryInvoice/rsm:ExchangedDocument/
ram:ID"/>
</xr:InvoiceNumber>
```

UBL 2.1 Invoice:

```
<xr:InvoiceNumber>
    <xsl:value-of select="/ubl:Invoice/cbc:ID"/>
</xr:InvoiceNumber>
```

Adding an additional namespace – this is how it works

1. Click **Add stylesheet for additional namespace**.
2. Define a unique name and enter the namespace of your e-invoice.
3. Click **Add**.

A new entry with the name and namespace is added to the list. You can now upload a style sheet for the namespace you created manually. If you delete the style sheet, the namespace that was added will be removed again.

1.3.2. Converting the universal format into a PDF file

Once the different e-invoice formats have been converted into a universal format, you can generate PDF files from the invoices. By default, a German and English version is offered for display. To convert to a PDF format, an HTML file is first created for display from the universal format using a style sheet. The HTML file is then converted into a PDF file.

Adding an additional language – this is how it works

1. Click **Add style sheet for another language**.
2. Select an additional language for a style sheet.
3. Click **Add.**

You can now upload a style sheet for this language. Similar to converting to a standardized format, you can upload and download style sheets for each language. The style sheets are transferred as XSLT files.

You can select the language used by default in the **Default language** column. You can now select a language as the default. In the **Default** column, you can see whether there is a default style sheet for a language. This is not the case only for languages that were added manually.

Below is an example from the style sheet for generating the HTML/PDF file:

```
<tr>
    <xsl:choose>
        <xsl:when test="xr:Invoice/xr:InvoiceTypeCode='381'">
            <th style="text-align:left;">Gutschrift</th>
        </xsl:when>
        <xsl:otherwise>
            <th style="text-align:left;">Rechnung</th>
        </xsl:otherwise>
    </xsl:choose>
    <td>
        <xsl:value-of select="xr:Invoice/xr:InvoiceNumber"/>
    </td>
</tr>
```

A value from the standardized XML format is accessed by specifying the XML path, e.g. **xr:Invoice/xr:InvoiceNumber**. In the example, the invoice number is determined by the path. In addition, a conditional statement is used to determine whether "Credit note" or "Invoice" is to be used in the display. The default style sheets are available for further guidance.

1.3.3. Configuring local directory monitoring

Local directory monitoring is intended to replace the functionality of the d.velop document reader eInvoice converter service. You can monitor a directory for new e-invoices. The incoming e-invoices are converted into a PDF file and saved in an output directory. The e-invoice is saved in addition to the PDF file.

You perform the configuration in JSON files, which are located in the **ConvertConfiguration** subdirectory of the application.

Configuring the task structure

You can configure the output structure in the file **converterconfiguration.json**. This chapter contains further information about the parameters.

- **HtmlOutput**: If you set this parameter to **true**, the HTML file created for generating the PDF file is also saved in the output directory. (default value: **true**)
- **JsonOutput**: The metadata in the XML file is used to generate a standardized JSON file with the invoice information. An example JSON file is contained in chapter Use in the Process app (default value: **false)**
- **XmlOutputSetting**: You can use this parameter to set which files are integrated in the PDF file. You can define the following values:
  - **0** (default value): No further variants for displaying the information are attached.
  - **1**: A simple representation of the information from the XML file is generated from the standardized format. The corresponding HTML and PDF files are integrated into the PDF file. The names of the attachments are as follows: **<Originalname>.simple_cml.pdf/<Originalname>.simple_cml.html**

- **2**: The original e-invoice is used to generate a simple representation of the information from the XML file. The corresponding HTML and PDF files are integrated into the PDF file. The names of the attachments are as follows: **<Originalname>.simple_original.pdf/<Originalname>.simple_original.html**
- **3**: Both variants from options **1** and **2** are attached.

- **AddXmlAsPdfAttachment**: If you set this parameter to **true**, the original e-invoice is integrated in the PDF file as an XML file. The name of the attachment is: **<Original name>.original.xml** (default value: **false**)

Alongside the PDF file that is generated, the original XML file is always saved in the output directory under the original name. A CML file is also created. The CML file is the standardized format and is used for processing invoice information.

Coded base64 attachments from the original e-invoice XML file are also integrated in the PDF file. The names are copied identically.

Configuring the input directories

The file **folderconfiguration.json** contains a list of the directories being monitored. The example monitoring of a subdirectory of the application is preconfigured by default. This chapter contains further information about the parameters.

- **InputFolder**: Here, you enter the directory that is to be monitored for e-invoices.
  Example: **EInvoiceTestFolder/Input** or **\\\\Servername\\TestEInvoice** (for monitoring a server directory). Backslashes must be entered twice in JSON.
- **OutputFolder**: Enter the output directory here.
- **ErrorFolder**: If an error occurs when processing the XML file, the XML file is saved in the configured error directory.
- **ProvidedPdfOutput**: In addition to the XML file in the input directory, a PDF file with an identical name can be provided. Activate the parameter with **true** to use a PDF file for the output.
- **WaitForOk**: The XML file in the input directory is only processed if a file of the same name with the extension **ok** exists in the input directory. Activate the parameter with **true** to set the **FilterExtension** setting to the value **ok**.
- **WatchSubdirectory**: This is where you can activate the monitoring of subdirectories in the input directory. All directories are then monitored. The directory structure is not taken into consideration in the output.
- **FilterExtension**: By default, files with the extension **xml** in the input directory are filtered. You can specify additional file extensions with this parameter.

Settings for the data transfer to d.velop scripting engine

You can transfer monitored files to d.velop scripting engine. This allows the imported content to be adapted before it is actually processed as an e-invoice. The script must return a valid e-invoice format. The following settings are required in order to use this function:

- **ScriptingEngineConfiguration.UseScriptingEngine: true**
- **ScriptingEngineConfiguration.ScriptingEngineProfile: eInvoiceConverterScript** (pre-configured by default)
- **ScriptingEngineConfiguration.TenantBaseUrl**: Enter the d.velop tenant in which d.velop scripting engine is opened.
- **ScriptingEngineConfiguration.AuthorizationBearer**: Enter an API key of a linked user who has sufficient permissions to open d.velop scripting engine.

In the following example script, the original file content is read in as XML. The value is changed using a tag in the XML file. The change is then sent back to d.velop eInvoice converter as XML. At this point, the actual processing takes place.

```
XmlDocument doc = new XmlDocument();
doc.LoadXml(document.Content);

var invoiceNumberElement = doc.GetElementsByTagName("cbc:ID");
invoiceNumberElement[0].InnerText = "RE_" +
invoiceNumberElement[0].InnerText;

document.Content = doc.OuterXml;
return document;
```

1.3.4. Use in the Process app

You can use the API of d.velop eInvoice converter to read the information from the e-invoice. You must start a separate process for each e-invoice that is to be classified.

The Process app requires the following information to start an asynchronous process for an e-invoice:

- **documentName**: Name of the document with extension, e.g. "Invoice.xml". Only file names with the extension **.xml** are valid.
- **documentFileLink**: Link to the e-invoice.

The following example process shows how you can use d.velop eInvoice converter to read the invoice number and transfer it to the Process app.

```
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:camunda="http://camunda.org/schema/1.0/bpmn" targetNamespace="">
  <process id="EInvoiceAPI" name="E-Invoice API Demo" isExecutable="true">
    <extensionElements>
      <camunda:properties>
        <camunda:property name="service:/classcon-einvoice/api/v1/
ProcessApp:in:documentName" value="String" />
        <camunda:property name="service:/classcon-einvoice/api/v1/
ProcessApp:in:documentFileLink" value="URL" />
        <camunda:property name="service:/classcon-einvoice/api/v1/
ProcessApp:out:number" value="String" />
        <camunda:property name="variable:documentName" value="String" />
        <camunda:property name="variable:documentFileLink" value="URL" />
        <camunda:property name="variable:number" value="String" />
      </camunda:properties>
    </extensionElements>
    <startEvent id="start"/>
    <sendTask id="call_einvoiceAPI" name="Call 'E-Invoice API'"
camunda:delegateExpression="${asyncService}" camunda:asyncBefore="true"
camunda:exclusive="true">
      <extensionElements>
            <camunda:inputOutput>
                <camunda:inputParameter name="service.uri">/classcon-
einvoice/api/v1/ProcessApp</camunda:inputParameter>
                <camunda:inputParameter name="documentName">$
{variables.get("documentName")}</camunda:inputParameter>
                <camunda:inputParameter name="documentFileLink">$
{variables.get("documentFileLink")}</camunda:inputParameter>
            </camunda:inputOutput>
        </extensionElements>
    </sendTask>
    <receiveTask id="receive_einvoiceAPI" name="Wait for 'E-Invoice API'"
camunda:asyncAfter="true" camunda:exclusive="true">
```

```
      <extensionElements>
            <camunda:inputOutput>
                <camunda:outputParameter name="number">$
{variables.get("number")}</camunda:outputParameter>
            </camunda:inputOutput>
        </extensionElements>
    </receiveTask>
    <endEvent id="end"/>
    <sequenceFlow id="s1" sourceRef="start" targetRef="call_einvoiceAPI" />
    <sequenceFlow id="s2" sourceRef="call_einvoiceAPI"
targetRef="receive_einvoiceAPI" />
    <sequenceFlow id="s3" sourceRef="receive_einvoiceAPI" targetRef="end" />
  </process>
</definitions>
```

Further information can also be returned. The JSON file has the following structure:

```
{
  "$contentType": "invoice",
  "$contentTypeVersion": "1.0.1",
  "number": "R123456789",
  "issueDate": "2021-09-27T01:00:00+02:00",
  "performancePeriod": {
    "start": "0001-01-01T00:00:00",
    "end": "0001-01-01T00:00:00"
  },
  "type": "invoice",
  "vatBreakdowns": [
    {
      "taxableAmount": 277.05,
      "taxAmount": 52.64,
      "categoryCode": "S",
      "rate": 19
    }
  ],
  "currencyCode": "EUR",
  "documentTotals": {
    "sumOfLineItemsNetAmounts": 277.05,
    "amountWithoutVat": 277.05,
    "amountWithVat": 329.69,
    "vatAmount": 52.64,
    "paidAmount": 0,
    "roundingAmount": 0,
    "amountDueForPayment": 329.69
  },
  "references": {
    "ref": "1002019005",
    "contractRef": "V876543210",
    "purchaseOrderRef": "1002019005",
    "salesOrderRef": "A123456789"
  },
  "lineItems": [
    {
      "id": "0010",
      "performancePeriod": {
        "start": "0001-01-01T00:00:00",
```

```
      "end": "0001-01-01T00:00:00"
    },
    "item": {
      "buyerItemId": "",
      "manufacturerItemId": "",
      "name": "",
      "sellerItemId": "10225483",
      "standardItemId": "",
      "description": "Adidas Deutschland Trikot WM 2014"
    },
    "quantity": 10,
    "quantityUnit": "XPP",
    "netAmount": 242.2,
    "unitPrice": 24.22,
    "priceBaseQuantityUnit": "",
    "priceBaseQuantity": 0,
    "discounts": [],
    "charges": [],
    "vatBreakdown": {
      "taxableAmount": 0,
      "taxAmount": 0,
      "categoryCode": "S",
      "rate": 19
    },
    "references": {}
  },
  {
    "id": "0020",
    "performancePeriod": {
      "start": "0001-01-01T00:00:00",
      "end": "0001-01-01T00:00:00"
    },
    "item": {
      "buyerItemId": "",
      "manufacturerItemId": "",
      "name": "",
      "sellerItemId": "10225488",
      "standardItemId": "",
      "description": "Adidas Deutschland Torwart Trikot WM 2014"
    },
    "quantity": 1,
    "quantityUnit": "XPP",
    "netAmount": 34.85,
    "unitPrice": 34.85,
    "priceBaseQuantityUnit": "",
    "priceBaseQuantity": 0,
    "discounts": [],
    "charges": [],
    "vatBreakdown": {
      "taxableAmount": 0,
      "taxAmount": 0,
      "categoryCode": "S",
      "rate": 19
    },
    "references": {}
```

```
    }
  ],
  "seller": {
    "id": "3",
    "name": "Sport24 GmbH",
    "vatId": "DE811166979",
    "contact": {
      "name": "Ihr Customer Service Team",
      "phone": "\u002B49 (0)231617170555",
      "email": "info@sport24.com"
    },
    "address": {
      "addressLine1": "Rheinlanddamm 207-209",
      "zipCode": "44137",
      "city": "Dortmund",
      "countryCode": "DE"
    },
    "bankAccounts": [],
    "legalRegistration": "HRB 123"
  },
  "buyer": {
    "id": "Solbau GmbH",
    "name": "Solbau GmbH",
    "address": {
      "addressLine1": "Baustra\u00DFe 40",
      "zipCode": "46395",
      "city": "Bocholt",
      "countryCode": "DE"
    },
    "vatId": "",
    "bankAccounts": [],
    "legalRegistration": "",
    "contact": {
      "name": "",
      "phone": "",
      "email": ""
    }
  },
  "deliveryInformation": {
    "id": "",
    "name": "",
    "date": "0001-01-01T00:00:00",
    "address": {
      "addressLine1": "",
      "zipCode": "",
      "city": "",
      "countryCode": ""
    }
  },
  "payment": {
    "terms": {
      "text": "The payment is made monthly as a partial payment and is due
monthly on the last working day. \u2026",
      "dueDate": "2022-02-01T00:00:00+01:00",
      "discounts": []
```

```
    },
    "instructions": {
      "typeCode": "30",
      "text": "",
      "remittanceInformation": "",
      "bankAccounts": [
        {
          "accountId": "DE25500700100092018100",
          "accountName": "",
          "serviceProviderId": "DE25500700100092018100"
        }
      ],
      "creditCard": {
        "cardNumber": "",
        "cardHolder": ""
      }
    }
  }
}
```


1.4. Tips and tricks

This section explores further options offered by the application to help you achieve your goal faster.

1.4.1. Integration in d.velop document reader

The d.velop document reader app uses d.velop eInvoice converter to generate a PDF file and evaluate the metadata from the XML file. For this purpose, the attachments of incoming documents are also searched in the d.velop document reader app. If the e-invoice is an attachment of an existing PDF file and it is a valid e-invoice, only the metadata of the XML file is used. In this case, a PDF file is not generated as the original PDF file is used for display and further processing. Only the metadata from the XML file is used to recognize an e-invoice.

1.4.2. Integration in the Inbound app

e-invoices can be made available as XML files in the Inbound app. d.velop eInvoice converter is used to create a PDF file to display the e-invoice. If it is a valid e-invoice, a PDF file is returned. The original XML file for the e-invoice is integrated in the PDF file.

1.5. Frequently asked questions

You can find answers to frequently asked questions in this section.

1.5.1. Will additional e-invoice formats be added?

If you wish to process e-invoices in a format that is not yet supported, get in touch with your d.velop contact person. New formats can be added to d.velop eInvoice converter at any time.

1.6. Additional information sources and imprint

If you want to deepen your knowledge of d.velop software, visit the d.velop academy digital learning platform at https://dvelopacademy.keelearning.de/.

Our E-learning modules let you develop a more in-depth knowledge and specialist expertise at your own speed. A huge number of E-learning modules are free for you to access without registering beforehand.

Visit our Knowledge Base on the d.velop service portal. In the Knowledge Base, you can find all our latest solutions, answers to frequently asked questions and how-to topics for specific tasks. You can find the Knowledge Base at the following address: https://kb.d-velop.de/

Find the central imprint at https://www.d-velop.com/imprint.