

ForNAV support for ZUGFeRD, Factur-X , and X-Rechnung

ZUGFeRD and its successor Factur-X, are PDF/A3 files embedded with an XML document, specifying the electronic document. X-Rechnung is the pure XML representation of the electronic document. The ForNAV 7.2 Customizable Report Pack supports in- and out-going ZUGFeRD, Factur-X, and X-Rechnung documents.

The output has been validated using [ZF/FX Validation - ZUGFeRD Community \(zugferd-community.net\)](https://zugferd-community.net) service:

ZF/FX Validation

Das ZUGFeRD-PDF ist valide.
Profile: PDF/A-3B validation profile
Statement: PDF file is compliant with Validation Profile requirements.
Signature: unknown
Passed checks: 4505
Passed rules: 124
Failed checks: 0
Failed rules: 0

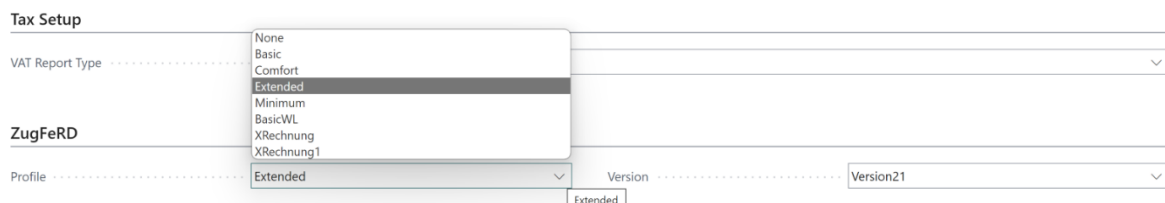
Das ZUGFeRD-xml ist valide.
Profile: urn:cen.eu:en16931:2017#conformant#urn:factur-x.eu:1p0:extended

Installation

To use the ForNAV ZUGFeRD feature, install the ForNAV eDocument extension from AppSource on cloud installations, or the Universal Code version of the Customizable Report Pack, [ForNAV Report Pack \(Universal Code\), for OnPrem.](#)

Setup

To set up ZUGFeRD, on the **ForNAV Setup** page, select the version and the profile that you want (the default is *Version2.1* and *Extended*).



The screenshot shows the 'Tax Setup' section of the ForNAV Setup page. It includes three dropdown menus: 'VAT Report Type' (with options: None, Basic, Comfort, Extended, Minimum, BasicWL, XRechnung, XRechnung1), 'ZugFeRD Profile' (with options: Extended), and 'Version' (with options: Version21). A small 'Extended' button is visible below the Profile dropdown.

The following combinations are supported:

Profile	Version 1	Version 2	Version 2.1
Minimum		X	X
Basic WL		X	X

Basic	X	X	X
Comfort/EN16391	X	X	X
Extended	X	X	X
X-Rechnung 1.0			X
X-Rechnung 2.0			X

When you have chosen a profile, all ForNAV reports using the **Posted Invoice/credit memo** table and the **Business Central Standard Sales invoice/Credit memo** table will automatically be modified into ZUGFeRD compliant documents. This means that the output is converted to the PDF/3a format and the electronic document embedded in the PDF.

If you select *None* as the profile, the ForNAV ZUGFeRD feature is turned off and the output is not modified.

Using ForNAV ZUGFeRD

Out of the box, ForNAV supports the following scenarios:

- Print/Send from a posted document.
- Sending electronic documents using the X-Rechnung or ZUGFeRD formats.
- Importing an electronic document.

Print/Send from a posted document

When you click **Print/Send** on a posted document, a ZUGFeRD PDF file is created in the process:

CRONUS AG | Sales ▾ | Purchasing ▾ | Inventory ▾ | Posted Documents ▾ | Shopify ▾ | ForNAV ▾ | ☰

Posted Sales Invoices: All ▾ | 🔍 Search | 🗑️ Delete | Home ▾ | Print/Send ▾ | Invoice ▾ | ↶ Correct ▾ | Actions ▾ | Fewer

✕ The document has been printed to attachments. Show Attachments

No.	Customer No.	Customer Name	Created On	Amount	Amount Including VAT
103035	01121212	Spotsmeyer's Furnishings	23-02-2025	3.980,50	3.980,50
103034	49633663	Autohaus Mielberg KG		1.800,00	2.142,00
103033	43687129	Designstudio Gmunden	23-02-2025	2.000,00	2.000,00
103032	40000	Graphische Werke Landsberg	23-02-2025	3.200,00	3.808,00

Print...
Send by Email
Send...
Attach as PDF

When the resulting PDF file, in this case the output from the Standard Sales Invoice report, is open in the Adobe PDF reader, the reader automatically detects that the PDF is in a PDF/A format and it has the XML-document attached, as specified in the ZUGFeRD standards.

ports ForNAV Trial Version

Rechnung

103034



Autohaus Mielberg KG
Porschestraße 911
22417 Hamburg 36

CRONUS AG

Hofstraße 12
Benrath

20097 Hamburg

999 / 9 99 99 99

Belegdatum: 23. Januar 2025
Fälligkeitsdatum: 6. Februar 2025
Zahlungsbedingungen: 14 Tage netto

Nr.	Beschreibung	Menge	Einheit	VK-Preis Ohne MwSt.	MwSt. %	Zeilenbetrag Ohne MwSt.
1100	Vorderrad	2	Stück	1.000,00	-10%	1.800,00
Zw.summe						1.800,00
						19% MwSt.
Total €						2.142,00

Using the X-Rechnung or [ZUGFeRD](#) electronic document formats to send electronic documents

When setting up electronic document sending in Business Central (<https://learn.microsoft.com/en-us/dynamics365/business-central/across-how-to-set-up-electronic-document-sending-and-receiving#to-select-the-peppol-electronic-document-format-for-electronic-document-sending>), you can select either the **ForNAV X-RECHNUNG** or **ForNAV ZUGFeRD** format:

Code ↑	Description
→ FORNAV X-RECHNUNG	X-Rechnung
FORNAV ZUGFERD	ZugFeRD
PEPPOL BIS3	PEPPOL BIS3-Format (Pan-European Public Procurement Online)

Importing electronic documents

On the **Purchase Invoices** page, you can import a purchase invoice from a ZUGFeRD PDF or an X-Rechnung XML file:

Invoice ▾	Actions ▾	Fewer options
Invoice >		Import from ZugFeRD pdf
⚡ Request Approval >		Import from X-Rechnung xml
⚡ Posting >		Other >

If the vendor in the invoice, which is identified by the VAT Registration number, does not exist in your Business Central system, you will be asked if you want to create it. After that, the **Purchase Invoice** page opens and you can adjust it.

Customization

Beyond the default functionality from ForNAV, there is a rich set of AL-constructs to modify the existing behavior and add new document types:

- Mapping from and to Business Central.
- Modifying/adding document types.
- ZUGFeRD support for non-ForNAV reports

Mapping from and to Business Central

To support mapping from and to tables in Business Central, ForNAV has defined a set of tables where the root table, *"ForNAV InvoiceDescriptor"*, corresponds to the Header tables in Business Central and *"ForNAV TradeLineItem"* corresponds to the equivalent Lines tables:

```

InvoiceDescriptor: "ForNAV InvoiceDescriptor"
  AdditionalReferencedDocuments: "ForNAV AddRefDoc"
  DeliveryNoteReferencedDocument: "ForNAV DeliveryNoteRefDoc"
  ContractReferencedDocument: "ForNAV ContractRefDoc"
  SpecifiedProcuringProject: "ForNAV SpecProcProj"
  Buyer: "ForNAV Party"
    SpecifiedLegalOrganization: "ForNAV LegalOrganization"
  BuyerContact: "ForNAV Contact"
  BuyerTaxRegistration: "ForNAV TaxRegistration"
  Seller: "ForNAV Party"
  SellerContact: "ForNAV Contact"
  SellerTaxRegistration: "ForNAV TaxRegistration"
  Invoicee: "ForNAV Party"
  ShipTo: "ForNAV Party"
  Payee: "ForNAV Party"
  ShipFrom: "ForNAV Party"
  Notes: "ForNAV Note"
TradeLineItems: "ForNAV TradeLineItem"
  AssociatedDocument: "ForNAV AssociatedDoc"
    Notes: "ForNAV Note"
  BuyerOrderReferencedDocument: "ForNAV BuyerOrderRefDoc"
  DeliveryNoteReferencedDocument: "ForNAV DeliveryNoteRefDoc"
  ContractReferencedDocument: "ForNAV ContractRefDoc"
  AdditionalReferencedDocuments: "ForNAV AddRefDoc"
  TradeAllowanceCharges: "ForNAV TradeAllowanceCharge"
    Tax: "ForNAV Tax"
  ReceivableSpecifiedTradeAccountingAccounts: "ForNAV RecvSpecTradeAcc"
  ApplicableProductCharacteristics: "ForNAV ApplProdChar"
  Taxes: "ForNAV Tax"
  ServiceCharges: "ForNAV ServiceCharge"
    Tax: "ForNAV Tax"
  TradeAllowanceCharges: "ForNAV TradeAllowanceCharge"
  PaymentTerms: "ForNAV PaymentTerms"
  InvoiceReferencedDocument: "ForNAV InvoiceRefDoc"
  ReceivableSpecifiedTradeAccountingAccounts: "ForNAV RecvSpecTradeAcc"
  CreditorBankAccounts: "ForNAV BankAcc"
  DebitorBankAccounts: "ForNAV BankAcc"
  PaymentMeans: "ForNAV PaymentMeans"
    FinancialCard: "ForNAV FinancialCard"
  SellerOrderReferencedDocument: "ForNAV SellerOrderRefDoc"

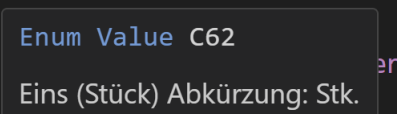
```

When coding against the ForNAV tables , you can use IntelliSense to make the data structure easier to understand:

```

if UnitCode <> '' then
  if UOM.Get(UnitCode) then
    if Evaluate(QC, UOM."Internat
      exit(QC);
    exit("ForNav QuantityCodes"::C62);

```



```

Tax. (field) TypeCode: Enum "ForNav TaxTypes"
Tax. Type of tax.
Tax. Generally, the fixed value is: "VAT"
Tax.TypeCode := Tax.TypeCode::VAT;
Tax.Insert();
end;

```

Mapping from Business Central

Each table has a set of Init methods that need to be called before setting field values on the record. *Insert()* must be called as in the following example:

```

internal procedure AddBuyer(var InvoiceDescriptor: Record "ForNAV
InvoiceDescriptor"; var C: Record Customer; DefaultCountryCode: Enum "ForNav
CountryCodes"): Text;
var
    TaxRegistration: Record "ForNAV TaxRegistration";
    Contact: Record "ForNAV Contact";
begin
    InvoiceDescriptor.InitBuyerTaxRegistration(TaxRegistration);
    TaxRegistration.No := FormatVATRegistrationNo(C."VAT Registration
No.", C."Country/Region Code", DefaultCountryCode);
    TaxRegistration.SchemeID := "ForNav TaxRegistrationSchemeID"::VA;
    TaxRegistration.Insert();

    InvoiceDescriptor.InitBuyerContact(Contact);
    Contact.Name := C.Contact;
    Assign(Contact.FaxNo, C."Fax No.");
    Contact.PhoneNo := C."Phone No.";
    Contact.EmailAddress := C."E-Mail";
    Contact.Insert();
end;

```

Mapping to Business Central

Each table has a set of FindFirst methods that need to be called before getting field values from the first record. After that, *Next()* is called in a loop to get the rest of the records:

```

local procedure AddPurchaseLines(var InvoiceDescriptor: Record "ForNAV
InvoiceDescriptor"; "Document No.": Code[20]; var Lines: Record "Purchase
Line")
var
    TradeLineItem: Record "ForNAV TradeLineItem";
....
begin
    if InvoiceDescriptor.FindFirstTradeLineItems(TradeLineItem) then
begin
    repeat
        Lines.Init();
        Lines."Document No." := "Document No.";
        Lines."Document Type" := Lines."Document Type"::Invoice;
....
        Lines.Insert();

```

```

        until TradeLineItem.Next <> 1;
    end;
end;

```

Modifying/adding document types

You can modify the default mapping by overriding the event in the *"ForNAV eDocument Interface"* codeunit:

```

codeunit 6189213 "ForNAV eDocument Interface"
{
    [IntegrationEvent(false, false)]
    internal procedure OnAfterDocument2InvoiceDescriptor(var
InvoiceDescriptor: Record "ForNAV InvoiceDescriptor")
    begin
    end;

    [IntegrationEvent(false, false)]
    internal procedure OnDocument2InvoiceDescriptor(var InvoiceDescriptor:
Record "ForNAV InvoiceDescriptor"; var eInvoice: Record "ForNAV eInvoice")
    begin
    end;
}

```

The *"OnAfterDocument2InvoiceDescriptor"* event is used to change the ForNAV default mapping and the *"OnDocument2InvoiceDescriptor"* event is used to override the default mapping completely and to map new document types.

[ZUGFeRD support for non-ForNAV reports](#)

To add ZUGFeRD support for non-ForNAV reports, you simply subscribe to an event in the report that is called with the document header record, and then call the AddZugFeRD method in the *"ForNAV eDocument Interface"* codeunit with the report number and the *RecordId*:

```

[EventSubscriber(ObjectType::Report, Report::"Standard Sales - Invoice",
'OnAfterGetSalesHeader', '', true, true)]
    local procedure OnAfterGetSalesInvoiceHeader(SalesInvoiceHeader: Record
"Sales Invoice Header")
    var
        eDocument: Codeunit "ForNAV eDocument Interface";
    begin
        eDocument.AddZugFeRD(Report::"Standard Sales - Invoice",
SalesInvoiceHeader.RecordId);
    end;

```