



Kreditoridentifikation durch IBAN

Mit Document Capture

15.06.2022

Inhaltsverzeichnis

1	Versionsübersicht.....	1
2	Einleitung	2
3	Lösungskonzept.....	3
4	Suche mit der Identifikationsvorlage	5
5	Suchen mit der IBAN.....	11

1 Versionsübersicht

Version	Datum	Autor	Kommentar
1.0	15.06.2022	Christian Böteführ/Bennet Balk	

2 Einleitung

Im Rahmen eines Kundenprojekts kam die Anforderung auf, Kreditoren nicht über die USt.-ID zu identifizieren, sondern über die IBAN. Hintergrund der Anforderung war, dass die IBANs deutlich besser im System gepflegt waren und der Kunde viele Kreditoren ohne USt.-ID hat.

Die Schwierigkeit hierbei liegt darin, dass in Business Central die Kreditorenbankkonto Informationen inklusive der IBAN in einer eigenen Tabelle gespeichert sind, da ein Kreditor natürlich auch mehrere Bankkonten haben kann. Die Standardidentifikationsmechanismen von Document Capture greifen daher auf diese Informationen nicht zu.

Document Capture nutzt zur Identifikation der Kreditoren drei verschiedene Ansätze in der genannten Reihenfolge:

1. Die Identifikation über die Suchtexte: Hierbei wird auf den Dokumenten nach an der Vorlage fix hinterlegten Suchtexten gesucht. Diese Variante greift nicht auf die Stammdaten zu.
2. Die Identifikation über das Identifikationstemplate. Hier wird im Standard auf dem Dokument nach Wörtern gesucht, die den Regeln einer USt.-ID entsprechen. Dann wird mit dem Kreditorenstamm abgeglichen, ob gefunden Treffer einer hinterlegten USt.-ID entsprechen.
3. Die Identifikation über die Kreditor-Identifikationsfelder. Hier werden in der Einrichtung hinterlegte Felder aus den Kreditorenstammdaten per Volltextsuche auf dem Dokument gesucht und für jeden Treffer gibt es eine definierte Anzahl Punkte. Der Kreditor, der die meisten Trefferpunkte erzielt, wird als identifizierter Kreditor ausgegeben.

3 Lösungskonzept

Die Identifikation über die Suchtexte bieten keine sinnvollen Erweiterungsmöglichkeiten, wir haben uns daher auf das Identifikationstemplate und die Identifikationsfelder konzentriert.

Zunächst wird das Identifikationstemplate dahingehend abgeändert, dass es statt nach einer USt.-ID nach einer IBAN sucht. Außerdem muss der Abgleich der gefundenen Werte gegen das IBAN-Feld aus den Kreditorenbankkonten erfolgen.

Darüber hinaus soll auch der Ansatz der Identifikationsfelder genutzt werden. Hier wurde zunächst überlegt eine analoge einrichtbare Lösung für Identifikationsfelder aus den Kreditor-Bankkonten zu schaffen. Diese Idee wurde aber verworfen, da nur die Kontonummer hier ein weiteres sinnvolles Kriterium wäre, diese wiederum ist aber Teil der IBAN.

Daher wurde die Suchlogik ausschließlich auf die IBAN erweitert.

Um einen Fallback zu haben, sollte außerdem die Standard-Identifikationsfelder weiter genutzt werden.

Voraussetzung für die Umsetzung war das Event „OnBeforeFindDocumentSource“ in der Codeunit 6085575 „CDC Capture Engine“, welches den angepassten Identifikationscode triggert.

```

no commands
procedure FindDocumentSource(var Document: Record "CDC Document")
var
    IsHandled: Boolean;
begin
    // *****
    // FIND THE SOURCE ID OF A DOCUMENT
    // *****
    OnBeforeFindDocumentSource(Document, IsHandled);
    IF IsHandled THEN
        EXIT;

    IF Document."Temp Page No." = 0 THEN
        Document."Temp Page No." := 1;
    Document."Source Record ID Tree ID" := 0;

    // STEP 1: FIND SOURCE WITH SEARCH TEXTS
    IF FindSourceWithSearchTexts(Document) THEN
        EXIT;

    // STEP 2: FIND SOURCE WITH IDENTIFICATION TEMPLATE
    IF FindSourceWithIdentTemplate(Document) THEN
        EXIT;

    // STEP 3: FIND SOURCE WITH IDENTIFICATION FIELDS
    FindSourceWithIdentFields(Document);
    OnAfterFindDocumentSource(Document);
end;

```

Für unsere Lösung haben wir außerdem den Ansatz verfolgt, das Handling komplett zu übernehmen. Aus diesem Grund wird das IsHandled immer auf TRUE gesetzt und der Standardcode von Continia nicht ausgeführt. Wir greifen aber auf die Continia Standard-Funktionen zu, um z.B. die Suche mit Suchtexten oder den Identifikationsfeldern wie gehabt durchzuführen. Dadurch werden doppelte Suchen ausgeschlossen:

```

[EventSubscriber(ObjectType::Codeunit, Codeunit::"CDC Capture Engine", 'OnBeforeFindDocumentSource', '', false, false)]
0 references
local procedure OnBeforeFindDocumentSource(var Document: Record "CDC Document"; var IsHandled: Boolean)
begin
    FindDocumentSourceByIBAN(Document);
    IsHandled := TRUE;
end;

1 reference
local procedure FindDocumentSourceByIBAN(var CDCDocument: Record "CDC Document")
begin
    if CDCCaptureEngine.FindSourceWithSearchTexts(CDCDocument) then
        EXIT;

    IF CDCCaptureEngine.FindSourceWithIdentTemplate(CDCDocument) THEN
        EXIT;

    IF FindSourceWithIBAN(CDCDocument) THEN
        EXIT;

    CDCCaptureEngine.FindSourceWithIdentFields(CDCDocument);
end;

```

4 Suche mit der Identifikationsvorlage

Zunächst wurde die Identifikationsvorlage EINKAUF-ID angepasst. Das Feld USt.-ID wurde entfernt und durch ein neues Feld IBAN ersetzt:

←

Vorlage Feld Karte

✎

+

🗑️

✓ Gespeichert

🔗

EINKAUF-ID - Kopf - IBAN - IBAN

🗨 Suchtexte

📋 Regeln

📄 Codeunits

Weitere Optionen

Allgemein

Mehr anzeigen

Feldname

IBAN

Wert übertragen nach...

Feld in Einkaufskopf

Feld in Einkaufszeile

Suchtext

IBAN (4 Suchtexte gesamt)

Regel

(EG)[0-9]{27} (93 Regeln gesamt)

Formel

Feldübersetzungen

Verwalten

🔗

🔗

→	Übersetze von	Übersetze nach	Groß-/Kle... beachte
-			<input type="checkbox"/>
.			<input type="checkbox"/>
•			<input type="checkbox"/>
•			<input type="checkbox"/>
•			<input type="checkbox"/>
•			<input type="checkbox"/>
•			<input type="checkbox"/>
°			<input type="checkbox"/>
:			<input type="checkbox"/>

Die Suchtexte wurden auf die IBAN angepasst und die Regeln so eingerichtet, dass es Regular Expressions für IBANs aus diversen Ländern gibt:

(EG)[0-9]{27}	Ägypten
(AL)[0-9]{26}	Albanien
(DZ)[0-9]{22}	Algerien

(AD)[0-9]{22}	Andorra
(AO)[0-9]{23}	Angola
(AZ)[0-9]{26}	Aserbaidshan
(BH)[0-9]{20}	Bahrain
(BE)[0-9]{14}	Belgien
(BJ)[0-9]{26}	Benin
(BA)[0-9]{18}	Bosnien und Herzegowina
(BR)[0-9]{27}	Brasilien
(VG)[0-9]{22}	Britische Jungferninseln
(BG)[0-9]{20}	Bulgarien
(BF)[0-9]{25}	Burkina Faso
(BI)[0-9]{14}	Burundi
(CR)[0-9]{20}	Costa Rica [20][21]
(CI)[0-9]{26}	Côte d’ Ivoire (Elfenbeinküste)
(DK)[0-9]{16}	Dänemark
(DE)[0-9]{20}	Deutschland
(DO)[0-9]{26}	Dominikanische Republik
(SV)[0-9]{26}	El Salvador
(EE)[0-9]{18}	Estland
(FO)[0-9]{16}	Färöer
(FI)[0-9]{16}	Finnland
(FR)[0-9]{25}	Frankreich [Anm. 1]
(GA)[0-9]{25}	Gabun
(GE)[0-9]{20}	Georgien

(GI)[0-9]{21}	Gibraltar
(GR)[0-9]{25}	Griechenland
(GL)[0-9]{16}	Grönland
(GT)[0-9]{26}	Guatemala
(IQ)[0-9]{21}	Irak
(IR)[0-9]{24}	Iran
(IE)[0-9]{20}	Irland
(IS)[0-9]{24}	Island
(IL)[0-9]{21}	Israel
(IT)[0-9]{25}	Italien
(JO)[0-9]{28}	Jordanien
(CM)[0-9]{25}	Kamerun
(CV)[0-9]{23}	Kap Verde
(KZ)[0-9]{18}	Kasachstan
(QA)[0-9]{27}	Katar
(CG)[0-9]{25}	Kongo (Brazzaville)
(XK)[0-9]{18}	Kosovo
(HR)[0-9]{19}	Kroatien
(KW)[0-9]{28}	Kuwait
(LV)[0-9]{19}	Lettland
(LB)[0-9]{26}	Libanon
(LI)[0-9]{19}	Liechtenstein
(LT)[0-9]{18}	Litauen
(LU)[0-9]{18}	Luxemburg

(MG)[0-9]{25}	Madagaskar
(ML)[0-9]{26}	Mali
(MT)[0-9]{29}	Malta
(MR)[0-9]{25}	Mauretanien
(MU)[0-9]{28}	Mauritius
(MD)[0-9]{22}	Moldau
(MC)[0-9]{25}	Monaco
(ME)[0-9]{20}	Montenegro
(MZ)[0-9]{23}	Mosambik
(NL)[0-9]{16}	Niederlande
(MK)[0-9]{17}	Nordmazedonien
(NO)[0-9]{13}	Norwegen
(AT)[0-9]{18}	Österreich
(TL)[0-9]{21}	Osttimor
(PK)[0-9]{22}	Pakistan
(PS)[0-9]{27}	Palästinensische Autonomiegebiete
(PL)[0-9]{26}	Polen
(PT)[0-9]{23}	Portugal
(RO)[0-9]{22}	Rumänien
(SM)[0-9]{25}	San Marino
(ST)[0-9]{23}	São Tomé und Príncipe
(SA)[0-9]{22}	Saudi-Arabien
(SE)[0-9]{22}	Schweden
(CH)[0-9]{19}	Schweiz

(SN)[0-9]{26}	Senegal
(RS)[0-9]{20}	Serbien
(SC)[0-9]{29}	Seychellen
(SK)[0-9]{22}	Slowakei
(SI)[0-9]{17}	Slowenien
(ES)[0-9]{22}	Spanien
(LC)[0-9]{30}	St. Lucia
(CZ)[0-9]{22}	Tschechien
(TN)[0-9]{22}	Tunesien
(TR)[0-9]{24}	Türkei
(UA)[0-9]{27}	Ukraine
(HU)[0-9]{26}	Ungarn
(VA)[0-9]{20}	Vatikanstadt
(AE)[0-9]{21}	Vereinigte Arabische Emirate
(GB)[0-9]{20}	Vereinigtes Königreich
(BY)[0-9]{26}	Weißrussland
(CY)[0-9]{26}	Zypern
(CF)[0-9]{25}	Zentralafrikanische Republik

Außerdem wurde in den Codeunits für die Codeunit „Nach Capture“ eine eigene Codeunit „IBAN Purch. Doc. Identificat.“ hinterlegt, die den Standardcode für die Suche nach IBANs abwandelt.

Die Codeunit ist eine Kopie der Codeunit 6085702 „CDC Purch. Doc. – Identificat.“.

Die Variablen wurden um die Tabelle „Vendor Bank Account“ ergänzt. Der Schritt „Find the Identification Template“ bleibt dabei identisch. Der darauffolgende Code wurde aber so abgewandelt, dass die IBAN gefunden wird:

```
// CAPTURE THE IBAN FROM THE DOCUMENT VALUES
// *****
IBAN := COPYSTR(CDCCaptureEngine.CaptureField2(Rec, Rec."Temp Page No.", CDCTemplateField, FALSE, CDCTemplateFieldCaption), 1, MaxStrLen(VendorBankAccount.IBAN))
// *****
```

Der Abgleich findet nun über die Kreditorenbankkonten-Tabelle statt und ermittelt darüber den Kreditorenbankkonto, der ermittelt wird:

```
5 IBAN := CopyStr(CDCCaptureManagement.ReplaceIllegalFilterCharacters(IBAN, FALSE, MAXSTRLEN(IBAN)), 1, MaxStrLen(IBAN));
6 if IBAN <> '' then begin
7   VendorBankAccount.SetRange(IBAN, IBAN);
8   if VendorBankAccount.FindSet() then
9     repeat
10      Vendor.Get(VendorBankAccount."Vendor No.");
11      RecIDTreeID := CDCRecordIDMgt.GetRecIDTreeID2(
12        CDCDocumentCategory."Source Table No.", Vendor.FIELDNO("No."), CDCDocumentCategory."Document Category GUID", Vendor."No.");
13
14      IF RecIDTreeID <> 0 THEN
15        Found := NOT CDCDocumentCatSourceExcl.GET("Document Category Code", RecIDTreeID);
16
17        Found := Vendor.Blocked <> Vendor.Blocked::" ";
18
19        IF Found THEN
20          FoundVendor := Vendor;
21      UNTIL Found OR (VendorBankAccount.Next() = 0);
22
23  IF NOT Found THEN BEGIN
24    IBAN := CopyStr(RemoveLeadingLetters(IBAN), 1, MaxStrLen(IBAN));
25    IF (STRLEN(IBAN) > 14) THEN BEGIN
26      VendorBankAccount.SetRange(IBAN, IBAN);
27      IF VendorBankAccount.FindSet() THEN
28        REPEAT
29          Vendor.Get(VendorBankAccount."Vendor No.");
30          RecIDTreeID := CDCRecordIDMgt.GetRecIDTreeID2(
31            CDCDocumentCategory."Source Table No.", Vendor.FIELDNO("No."), CDCDocumentCategory."Document Category GUID", Vendor."No.");
32          IF RecIDTreeID <> 0 THEN
33            Found := NOT CDCDocumentCatSourceExcl.GET("Document Category Code", RecIDTreeID);
34            Found := Vendor.Blocked <> Vendor.Blocked::" ";
35            IF Found THEN
36              FoundVendor := Vendor;
37          UNTIL Found OR (VendorBankAccount.Next() = 0);
38
39      IF NOT Found THEN BEGIN
40        VendorBankAccount.SetFilter(IBAN, '%1', '*' + IBAN + '*');
41        IF VendorBankAccount.FindSet() THEN
42          REPEAT
43            Vendor.Get(VendorBankAccount."Vendor No.");
44            RecIDTreeID := CDCRecordIDMgt.GetRecIDTreeID2(
45              CDCDocumentCategory."Source Table No.", Vendor.FIELDNO("No."), CDCDocumentCategory."Document Category GUID", Vendor."No.");
46            IF RecIDTreeID <> 0 THEN
47              Found := NOT CDCDocumentCatSourceExcl.GET("Document Category Code", RecIDTreeID);
48              Found := Vendor.Blocked <> Vendor.Blocked::" ";
49              IF Found THEN
50                FoundVendor := Vendor;
51            UNTIL Found OR (VendorBankAccount.Next() = 0);
52
53      END;
54
```

Der restliche Quellcode ist identisch zum Original.

Als Ergebnis des Kundenprojekts hat sich gezeigt, dass es hier noch Verbesserungsbedarf gibt, da über das Identifikationstemplate bisher kaum Treffer erzeugt wurden.

5 Suchen mit der IBAN

Als zusätzlicher Suchschritt wurde die Funktion „FindSourceWithIBAN“ ergänzt. Diese orientiert sich an der Funktion FindSourceWithIdentFields aus dem Document Capture Standard, der eine geänderte Version der Funktion „GetRecFromIdentField“ aufruft.

Die geänderten Stellen sind in der folgenden Abbildung dargestellt:

```
begin
DocCat.GET(CDCDocument."Document Category Code");
VendorBankAccount.SETFILTER("Vendor No.", '<>%1', '');
VendorBankAccount.SETFILTER(IBAN, '<>%1', '');
if NOT VendorBankAccount.FindSet() then
    exit;
    You, vor 8 Monaten • Init App ...

IF CDCDocument."File Type" = CDCDocument."File Type"::XML THEN BEGIN
CDCDocument.BuildXmlBuffer(XmlBuffer);
XmlBuffer.SETFILTER(Type, '%1|%2', XmlBuffer.Type::Element, XmlBuffer.Type::Attribute);
IF NOT XmlBuffer.FINDSET(FALSE, FALSE) THEN
    EXIT;

REPEAT
    BigString.Append(UPPERCASE(DELCHR(XmlBuffer.Value, '=', ' ', '.,-:/\*+')));
UNTIL XmlBuffer.Next() = 0;
END ELSE BEGIN
//Change Key to Top,Left
DocWord.SETCURRENTKEY("Document No.", "Page No.", Top, Left);
DocWord.SETRANGE("Document No.", CDCDocument."No.");
DocWord.SETRANGE("Page No.", PageNo);
IF NOT DocWord.FINDSET(FALSE, FALSE) THEN
    EXIT;

REPEAT
    BigString.Append(UPPERCASE(DELCHR(DocWord.Word, '=', ' ', '.,-:/\*+')));
UNTIL DocWord.NEXT() = 0;
END;

REPEAT
    Vendor.GET(VendorBankAccount."Vendor No.");
RecRef.GETTABLE(Vendor);
RecIDTreeID := RecIDMgt.GetRecIDTreeID(RecRef, FALSE);
IF NOT SourceExcl.GET(CDCDocument."Document Category Code", RecIDTreeID) OR (Vendor.Blocked <> Vendor.Blocked::" ") THEN BEGIN
    RecMatchPoint := 0;
    IdentificationFields := '';
    VendorRecRef.GETTABLE(VendorBankAccount);
    FieldRef := VendorRecRef.FIELD(VendorBankAccount.FIELDNO(IBAN));
    RecPoints := GetPoints(FORMAT(FieldRef.VALUE), BigString, STRLEN(FORMAT(FieldRef.VALUE))) * IdentifierField.Rating;
    IF RecPoints > 0 THEN BEGIN
        RecMatchPoint += RecPoints;
        IdentFieldNameAndValue := VendorBankAccount.FIELDCAPION(IBAN) + ': ' + FORMAT(FieldRef.VALUE);
    END;
    IF IdentificationFields <> '' THEN
        IdentificationFields := IdentificationFields + ' ' + IdentFieldNameAndValue;
    END;
END;
END;
```

Wichtig ist, dass die Suche mit den IBANs aus den Kreditorenbankkonten erfolgt. Außerdem wurde die Sortierung der CDC Document Words geändert. Diese Tabelle enthält alle Wörter, die auf dem Dokument durch die OCR Erkennung gefunden wurden. Standardmäßig ist diese nach der „Entry No.“ Sortiert. Die Entry No. beschreibt aber nicht zwangsläufig die Positionierung der Wörter von oben links nach unten rechts. Da IBANs häufig mit Leerzeichen dargestellt werden, sollte dies die Trefferquote erhöhen.

Das Sucherverfahren nimmt an dieser Stelle alle gefundenen Wörter und fügt Sie in einem zusammenhängenden BigString ohne Leer- und Sonderzeichen zusammen. Bei IBANs mit Leerzeichen

können wir so beeinflussen, dass diese im BigString auch möglichst als zusammenhängende IBAN auftaucht.

Die IBANs aus den Kreditorenbankkonten (ohne Leer- und Sonderzeichen) werden nun in diesem BigString gesucht. Bei jedem Treffer gibt es hier analog zu den Identifikationsfeldern einen Punkt. Da nur über ein Feld (die IBAN) gesucht wird, ist der erste Treffer hier auch immer der Beste.

Eine mögliche Optimierung an dieser Stelle wäre also entweder die Trefferanzahl zu zählen, da aktuell nur geguckt wird, ob die IBAN überhaupt auftaucht. Alternativ könnte die Suche, sobald ein Treffer existiert abgebrochen werden, da es nie einen besseren Treffer geben wird.