



Installations- und Gebrauchsanweisung für das IBAN-Tool

**Schnittstellenbeschreibung und Einsatzmöglichkeit des
IBAN-Tools**

Spezifikation für Software-Firmen und Finanzinstitute

Allgemeine Hinweise

Die in diesem Dokument enthaltenen Angaben entsprechen dem aktuellen Entwicklungsstand. SIX Interbank Clearing behält sich vor, dieses Dokument bei Bedarf jederzeit ohne vorherige Benachrichtigung zu ändern.

Für dieses Dokument werden alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien sowie der Übersetzung in fremde Sprachen.

Das Dokument ist mit grösster Sorgfalt erstellt worden, doch können Fehler und Ungenauigkeiten nicht vollständig ausgeschlossen werden.

SIX Interbank Clearing kann für Fehler und deren Folgen weder eine juristische Verantwortung noch irgendwelche Haftung übernehmen.

Wenn Sie allfällige Fehler in diesem Dokument feststellen oder wenn Sie Verbesserungsvorschläge dazu haben, so sind wir Ihnen dankbar, wenn Sie dies der SIX Interbank Clearing AG melden:

Per E-Mail an **contact.sic@six-group.com** oder telefonisch an **+41 58 399 4420**.

Über dieses Dokument

Das vorliegende Dokument richtet sich an die am schweizerischen Bankenclearing angeschlossenen Finanzinstitute, an Software-Firmen, welche ihren Kunden Zahlungsverkehrsanwendungen (ZV-Software) zur Verfügung stellen sowie an allfällige «Eigenrealisatoren» von Zahlungsverkehrsanwendungen.

Die Ausführungen betreffen vor allem **Finanzinstitute (FI)**

- die ihre Algorithmen für das Implementieren des IBAN-Tools zur Verfügung gestellt haben und
- die Begünstigten-Stammdaten in ihren ZV-Ausgangsanwendungen (Scanning-Systeme, Daueraufträge, E-Banking usw.) abgespeichert haben.

Bei den **Software-Firmen** (SW-Firmen) sind jene Unternehmen angesprochen

- die ERP- sowie ZV-Software für Zahlungspflichtige (ZP) bzw. Zahlungsempfänger (ZE) anbieten und
- die Standard-Bankenapplikationen entwickeln.

Allfällige «**Eigenrealisatoren**» sind Firmen, die ihre Kundenstammdaten in einer von ihnen selbst entwickelten Applikation verwalten, und die daraus Zahlungsrecords bzw. Dateien mit LSV-Einzugsaufträgen generieren.

Revisionsnachweis

Nachfolgend werden alle bedeutenden durchgeführten Änderungen an diesem Dokument mit Änderungsdatum, kurzer Änderungsbeschreibung und Angabe der betroffenen Ziffern aufgelistet.

Version	Datum	Änderungsbeschreibung	Ziffer
8.0	14.01.2020	Diverse Aktualisierungen	Alle
7.1	01.03.2019	Hinweis bezüglich verketteter IIDs (keine IBAN-Berechnung erlaubt)	4.1.2
7.0	21.01.2019	Allgemeine Aktualisierung – u.a. IID anstelle von BC-Nummer	Alle
		Anpassung der Lauffähigkeit (ab Windows Vista)	5.1
6.0	20.11.2017	Diverse Aktualisierungen	Alle
5.0	16.06.2011	Ablaufdatum und Releases	4.6
		Kapitel ‚Termine‘ wurde aus Aktualitätsgründen gelöscht	10
		Anpassung der Support-Angaben	11
4.0	15.02.2010	Korrektur in „Anwendungsbeispiel“	7.5
		Anpassung der Support-Angaben	11
3.0	17.08.2009	Nochmalige Überarbeitung sowie Aktualisierungen	Alle
2.0	20.11.2006	Komplette Überarbeitung	Alle
1.0	20.09.2006	Erstausgabe	Alle

Darstellung der Änderungen

Die letzten Änderungen gegenüber der Vorgängerversion werden im Dokument durch Änderungsmarkierungen gekennzeichnet, sofern diese Änderungen bedeutsam sind. Layoutänderungen, Korrekturen von Schreibfehlern und geänderte Begriffe, die sich im gesamten Dokument mehrfach wiederholen, werden nicht markiert. Ältere Änderungen von anderen Versionen müssen der Änderungskontrolle entnommen werden.

Inhaltsverzeichnis

1	Ausgangslage	5
1.1	IBAN in der Schweiz.....	5
1.2	Standard der IBAN-Ausprägung für die Schweiz und Liechtenstein.....	5
1.3	IBAN als Mittel zur Erhöhung der STP-Rate im nationalen Zahlungsverkehr.....	5
2	Zielsetzungen des IBAN-Tools.....	6
2.1	IBAN-Tool als Migrationshilfe	6
2.2	Umfang der zu bereinigenden Stammdaten.....	6
2.3	Kurzbeschreibung der Migration	7
3	Mengengerüst	8
3.1	Anzahl Finanzinstitute und Zahlungspflichtige	8
3.2	Anzahl teilnehmende Finanzinstitute	8
4	Konzeptionelle Aspekte des IBAN-Tools	9
4.1	Konversion bankproprietärer Kontonummern	9
4.2	Konversionsumfang	9
4.3	Abgrenzung	10
4.4	Argumente für Haftungsbeschränkung in den Lizenzbestimmungen	10
4.5	Laufzeitbeschränkung und Releases des IBAN-Tools	11
4.6	Konversion der abgespeicherten Stammdaten	11
5	Technische Konzeption des IBAN-Tools	12
5.1	Programmierung in «Java» und als «Windows DLL»	12
5.2	Elemente des IBAN-Tools	12
6	Formate der Input-/Output-Datenfelder	13
6.1	Struktur der Datenfelder	13
6.2	Genereller Record-Aufbau im Falle der XML-/ASCII-Schnittstelle.....	17
7	Technische Komponenten der Java-Version.....	18
7.1	Mögliche Java-Versionen.....	18
7.2	Input-/Output-Schnittstellen der Java-Version	18
7.3	Optimierungsaspekte bei Massenmutationen.....	18
7.4	Startparametrisierung und Kommandozeile.....	19
7.5	Java Interface / Schnittstellenspezifikation IBAN-Tool Java	20
7.6	Benutzeroberfläche (GUI) für Java-Version	25
7.7	Installation des IBAN-Tools	26
8	Technische Komponenten der Windows-Version.....	27
8.1	Voraussetzungen für Windows DLL	27
8.2	Schnittstellenbeschreibung Windows DLL.....	27
8.3	Benutzeroberfläche (GUI) für Windows-DLL-Version	31
9	Einsatzmöglichkeiten des IBAN-Tools.....	32
9.1	Generelle Bemerkungen.....	32
9.2	Minimale Erwartungen an den Aufbau der Datenbanken.....	32
9.3	Bereinigung der Stammdaten der Massenmigration	35
9.4	Bereinigungsstand der Stammdaten nach durchgeführter Massenmigration.....	38
9.5	Einsatzmöglichkeit des IBAN-Tools in der täglichen ZV-Verarbeitung	39
9.6	Künftiges Generieren der Zahlungsrecords im ZV-Ausgang.....	41
10	Feedback und Fragen.....	43

1 Ausgangslage

1.1 IBAN in der Schweiz

Schweizer und liechtensteinische Finanzinstitute entschieden bereits 2000, den IBAN-Standard zu übernehmen. Wegen der Verschiebung des IBAN-Obligatoriums aufgrund der späteren Einführung der QR-Rechnung wurde noch keine konsequente Umstellung auf IBAN im nationalen Zahlungsverkehr vorgenommen.

Ein IBAN-Obligatorium im Schweizer Zahlungsverkehr kann erst nach Einstellung des ES- und ESR-Verfahrens erfolgen.

Es wird jedoch dringend empfohlen, wo möglich nur noch die IBAN zu verwenden.

1.2 Standard der IBAN-Ausprägung für die Schweiz und Liechtenstein

Im Rahmen des ISO 13616 Standards bestehen gewisse Freiheiten für eine landesspezifische Ausprägung der IBAN: Die Länge der IBAN sowie die darin eingebetteten IID und BAN können auf die jeweiligen Bedürfnisse abgestimmt werden. Verbindlich sind hingegen die Position und der Wert des Landcodes sowie die Position der Prüfziffer und deren Berechnungsmodus (Modulo 97-10).

Voraussetzung ist im Weiteren, dass sich die Finanzinstitute eines Landes auf einen einheitlichen Landesstandard einigen.

Die Finanzinstitute der Schweiz und des Fürstentums Liechtenstein einigten sich auf folgende 21-stellige CH-Ausprägung:

CH69 0647 0016 0066 7100 2

Kontonummer (12 Stellen) = BAN

IID (5 Stellen)

Prüfziffer (2 Stellen)

Ländercode (2 Stellen): «CH» für Schweiz und «LI» für Liechtenstein

1.3 IBAN als Mittel zur Erhöhung der STP-Rate im nationalen Zahlungsverkehr

Eine vom Verwaltungsrat der SIX Interbank Clearing in Auftrag gegebene Studie zeigte, dass viele Zahlungen beim Finanzinstitut des Zahlungsempfängers (bzw. beim Finanzinstitut des Zahlungspflichtigen im Lastschriftverfahren) aufgrund falscher oder mangelhafter Begünstigten- bzw. Zahlungspflichtiger-Daten nicht automatisch verarbeitet werden. Für die Ermittlung des effektiven Kontoinhabers ist deshalb eine arbeitsintensive Nachbearbeitung erforderlich.

Der Hauptgrund für die mangelhafte Datenqualität ist die Tatsache, dass Kontonummern durch den Absender nicht validiert werden können. Dies im Gegensatz zur Postkonto-Nummer, deren Standard und Prüfzifferberechnung in den schweizerischen Zahlungsverkehrssystemen implementiert sind. Ein verbindlicher Standard mit Prüfzifferberechnung existiert auch bei der IBAN.

2 Zielsetzungen des IBAN-Tools

2.1 IBAN-Tool als Migrationshilfe

Inzwischen sind insbesondere auch die E-Banking-Applikationen IBAN-fähig und viele Stammdaten in den ZV-Applikationen der Kunden werden mit dem IBAN-Tool periodisch bereinigt. Jedoch sind immer noch in vielen ZV-Systemen proprietäre Bankkontonummern abgespeichert, welche bei jeder Zahlungsauslösung automatisch in den Zahlungsrecords übernommen werden.

Ziel ist es, dass auch jene Zahlungspflichtigen, welche bisher noch gezögert haben, das **IBAN-Berechnungs-Tool** in ihre ZV-Applikationen zu installieren, die in ihren Stammdaten gespeicherten bankproprietären Kontonummern **in korrekte IBANs** umwandeln. Damit lässt sich der sonst notwendige, manuelle Bereinigungsaufwand vermeiden, bzw. bis auf wenige Fälle reduzieren.

Das IBAN-Tool kann die proprietären Kontonummern – wie auch die Konto-Identifikation in der Referenzzeile der roten Einzahlungsscheine – der weitaus meisten Finanzinstitute in der Schweiz und in Lichtenstein in korrekte **CH-/LI-IBANs** umwandeln.

IBANs anderer Länder können im IBAN-Tool nicht berechnet und auch nicht validiert werden.

2.2 Umfang der zu bereinigenden Stammdaten

Durch das IBAN-Tool können die proprietären Kontonummern bereinigt bzw. in IBANs migriert werden, die in den Stammdaten der betroffenen ZV-Applikationen abgespeichert sind. Gleichzeitig können auch die dazu gehörenden IID und Postkontonummern überprüft und allenfalls berichtigt werden.

Nicht bereinigt werden Namen und Adressen der unter diesen Kontonummern abgespeicherten Zahlungsempfänger bzw. Zahlungspflichtigen (bei Lastschriften).

Bei dieser Stammdatenbereinigung geht es nicht um die eigenen Kontonummern, sondern um jene der Empfänger einer Zahlung bzw. Lastschrift-Transaktion.

Konkret werden somit – sofern möglich – folgende Kontonummern durch das IBAN-Tool in IBANs konvertiert:

- **in den Stammdaten der Zahlungspflichtigen** bzw. **Finanzinstitute der Zahlungspflichtigen** die **Kontonummern der abgespeicherten Zahlungsempfänger** bzw.
- **in den Stammdaten des Zahlungsempfängers** – bei Lastschriften – **die Kontonummern der abgespeicherten Zahlungspflichtigen**

In den folgenden Kapiteln wird – der Einfachheit halber – meistens nur von den bei den Zahlungspflichtigen bzw. Finanzinstituten gespeicherten Kontonummern der Zahlungsempfänger gesprochen. Damit sind jedoch stets auch die bei den Zahlungsempfängern gespeicherten Kontonummern der Zahlungspflichtigen bei Lastschriften gemeint.

2.3 Kurzbeschreibung der Migration

Anforderungen an das IBAN-Tool

Das IBAN-Tool enthält eine Input- und eine Output-Schnittstelle, damit die Finanzinstitute der Zahlungspflichtigen bzw. die Zahlungspflichtigen die in ihren Stammdaten gespeicherten Begünstigtendaten ins IBAN-Tool einspeisen und die mit der IBAN ergänzten Records anschliessend wieder in ihre eigene Datenbank zurück übertragen können.

Im Weiteren enthält das IBAN-Tool die bei den Finanzinstituten ermittelten, bankindividuellen Algorithmen.

Anhand dieser Algorithmen entscheidet das IBAN-Tool, ob aufgrund der eingelesenen proprietären Kontonummern korrekte IBANs errechnet werden können oder ob auf eine Umwandlung verzichtet werden muss.

Vor jedem IBAN-Tool-Release gibt es einige Finanzinstitute, die ihre Kontonummern-Strukturen neu im IBAN-Tool hinterlegen lassen und Finanzinstitute, deren Kontonummern-Struktur sich aufgrund des Wechsels der Informatik-Plattform oder aufgrund von Fusionen/Zentralisierung verändert hat.

Für die Benutzer des IBAN-Tools heisst dies, dass die Stammdaten-Bereinigung keine einmalige Aktion darstellt.

3 Mengengerüst

SIX Interbank Clearing hat in Zusammenarbeit mit den zuständigen Interbank-Gremien verschiedene Erhebungen durchgeführt, um das Potenzial eines solchen IBAN-Tools auszuloten.

3.1 Anzahl Finanzinstitute und Zahlungspflichtige

Auf der Seite der Auslöser einer elektronischen Zahlung finden sich

- die rund 800 am schweizerischen Zahlungsverkehr teilnehmenden Finanzinstitute,
- rund 40'000 DTA/EZAG-Kunden,
- rund 2500 Teilnehmer am Lastschriftverfahren der Schweizer Banken,
- rund 110'000 Kunden mit E-Banking-Offline-Tools sowie
- über eine Million E-Banking-/yellownet-Online-Kunden,

die mehr oder weniger viele Stammdaten abgespeichert haben. Insbesondere die vier ersten Kategorien sind für den Einsatz des IBAN-Tools prädestiniert.

3.2 Anzahl teilnehmende Finanzinstitute

Das IBAN-Tool kann nur dann eine echte Migrationsunterstützung leisten, wenn die Algorithmen der betroffenen Banken darin hinterlegt werden.

Über 600 der knapp 800 in der Schweiz domizilierten Finanzinstitute haben sich bereit erklärt, ihre Algorithmen zur IBAN-Konversion der bankproprietären Kontonummern für die Realisierung des IBAN-Tools zur Verfügung zu stellen.

Die Algorithmen der restlichen knapp 200 Finanzinstitute sind im IBAN-Tool somit nicht hinterlegt. Bei diesen Finanzinstituten handelt es sich jedoch grösstenteils um Banken mit sehr wenig Zahlungsverkehr; somit sind deren Kontonummern auch nur in wenigen Datenbanken abgespeichert.

Demgegenüber decken die rund 600 teilnehmenden Finanzinstitute rund 95% des schweizerischen Zahlungsverkehrs ab.

4 Konzeptionelle Aspekte des IBAN-Tools

4.1 Konversion bankproprietärer Kontonummern

4.1.1 Grundsatz

Bei der Einführung der IBAN haben die Finanzinstitute den Grundsatz aufgestellt, dass nur das eigene Institut aus der bisherigen, proprietären Kontonummer eine IBAN berechnen darf.

Im Prinzip wurde der IBAN-Standard für die Schweiz und das Fürstentum Liechtenstein für alle Finanzinstitute allgemein verbindlich. Das bedeutet jedoch nicht, dass anhand der bisherigen IID und der bisherigen proprietären Kontonummer zweifelsfrei eine korrekte IBAN gebildet werden kann. Viele Institute haben nur ihnen bekannte Regeln aufgestellt, wie z.B.

- welche IID zu integrieren ist (z.B. Hauptsitz oder Filiale),
- ob zusätzliche Zeichen in den BAN-Teil einzupflegen oder gewisse Zeichen in der bisherigen proprietären Kontonummer nicht in die IBAN zu übernehmen sind
- ob und allenfalls welche ihrer proprietären Kontonummern gar nicht in IBANs umzuwandeln sind.

Aus diesen Gründen ist es Software-Firmen und Finanzinstituten strikte untersagt, IBANs von Drittkunden mit selbst entwickelten Programmen zu berechnen.

Für das IBAN-Tool haben die zuständigen Gremien eine Ausnahme von diesem Grundsatz beschlossen; dies unter der Auflage, dass bei dessen Entwicklung ganz klare Auflagen eingehalten werden.

4.1.2 Im IBAN-Tool hinterlegte, bankindividuelle Algorithmen

Das IBAN-Tool analysiert die eingelieferten Stammdaten anhand der durch die kontoführende Bank festgelegten Algorithmen und entscheidet in der Folge, ob diese Daten mit genügender Sicherheit in eine IBAN konvertiert werden können.

Je nach Resultat dieser Analyse gibt es via Output-Schnittstelle die mit Algorithmen errechnete IBAN oder eine entsprechende Fehlermeldung aus.

Wichtig:

Für eine IID, die auf eine andere IID verkettet ist, dürfen keine IBAN gerechnet werden (Status 11), da Fusionen in der Regel auch zu neuen Konten führen.

Bei einmaligen Stammdatenbereinigungen dürfen verkettete (bzw. alte) IIDs nicht verändert werden.

4.2 Konversionsumfang

Das IBAN-Tool dient ausschliesslich zur Konversion von proprietären Kontonummern in IBANs. Konvertiert werden dabei

- die in den Stammdaten abgespeicherten proprietären Kontonummern,
- die in der 27-stelligen ES-Codierzeilen des roten Einzahlungsscheins hinterlegten Kontonummern/Konto-Identifikationen der Bankkunden,
- Postkonto-Nummern von PostFinance-Kunden.

4.3 Abgrenzung

ESR-Teilnehmer-Nummern und allenfalls in den 27-stelligen ESR-Referenznummern hinterlegte Kontonummern-Identifikationen von Bankkunden sowie Postkonto-Nummern von Banken werden nicht in IBANs umgewandelt.

4.4 Argumente für Haftungsbeschränkung in den Lizenzbestimmungen

Eine 100% korrekte Umwandlung der proprietären Kontonummern in IBANs kann das IBAN-Tool nicht garantieren. Es gibt Konstellationen, bei welchen es nicht erkennen kann, dass die Basis-Inputdaten zu einer unkorrekten IBAN führen.

Zielsetzung bei der Entwicklung des IBAN-Tools ist es, dieses Risiko von unkorrekt generierten IBANs möglichst gering zu halten. Bei jenen Instituten, welche – zusätzlich zu den Konversions-Algorithmen – auch noch die Berechnungsroutinen ihrer Prüfziffer hinterlegen, wird die Fehlerrate nahe bei Null liegen. Bei jenen Finanzinstituten, die keine Prüfziffer in den proprietären Kontonummern eingebaut haben oder auf diese zusätzliche Analyse im IBAN-Tool verzichten, wird die Fehlerrate leicht höher liegen.

Diese Fehlerrate ist jedoch insofern zu relativieren, als im Falle der proprietären Kontonummern heute täglich zwischen 5% und 10% unvollständig oder fehlerhaft sind.

Um allfällige Missverständnisse auszuschliessen, werden dem IBAN-Tool Lizenzbestimmungen beigefügt, in welchen u.a. darauf hingewiesen wird, dass ein fehlerhafter Input unter Umständen zu einem fehlerhaften Output führen kann und demzufolge kein 100%-iger Anspruch auf eine fehlerfreie Konversion der proprietären Kontonummern geltend gemacht werden kann.

Software-Firmen, welche das IBAN-Tool in ihre Applikationen einbauen, sind verpflichtet, diese Lizenzbestimmungen vor dem erstmaligen Gebrauch des IBAN-Tools einzublenden und dessen Kenntnisnahme durch den Benutzer bestätigen zu lassen (allfällige Textänderungen vorbehalten).

IBAN-Tool-Lizenzbestimmungen von SIX Interbank Clearing

1. Lizenz

- 1.1 *SIX Interbank Clearing (nachstehend Lizenzgeber genannt) gewährt dem Lizenznehmer ein zeitlich unbegrenztes, übertragbares, unterlizenzierbares und nicht ausschliessliches Lizenzrecht an der IBAN-Tool Software und der dazugehörigen Dokumentation. Im Folgenden werden Software und Dokumentation zusammengefasst als Lizenzmaterial bezeichnet.*
- 1.2 *Abgesehen von den in Ziffer 1.1 eingeräumten Nutzungsrechten verbleiben alle Rechte am Lizenzmaterial und allfälligen vom Lizenznehmer hergestellten Kopien davon beim Lizenzgeber. Der Lizenznehmer verpflichtet sich, Schutzvermerke unverändert beizubehalten.*
- 1.3 *Eine vollständige oder teilweise Rückübersetzung des Lizenzmaterials (Revers Engineering durch Revers Assembling, Revers Compilation oder andere Übersetzung) in die Form eines Quellencodes (Sourcecode) ist dem Lizenznehmer nicht gestattet.*
- 1.4 *Die Software ist mit einer Laufzeitbeschränkung versehen. Nach deren Ablauf ist die Software ohne Installation eines Updates nicht mehr lauffähig.*

2. Wartung

Der Lizenzgeber liefert periodisch neue Updates, er erbringt darüber hinaus keine weiteren Wartungsleistungen.

3. Ausschluss der Sachgewährleistung und Beschränkung der Haftung

- 3.1 *Die Sachgewährleistung wird soweit gesetzlich zulässig vom Lizenzgeber ausdrücklich wegbedungen.*
- 3.2 *Die Software errechnet aufgrund der Inputdaten des Lizenznehmers und anhand der vom Lizenzgeber von den Banken erhaltenen Algorithmen die IBAN. Es sind ausnahmsweise Konstellationen denkbar, bei welchen die hinterlegten Algorithmen einen fehlerhaften Input durch den Lizenznehmer nicht feststellen können. Es ist deshalb möglich, dass eine falsche oder ungültige IBAN errechnet wird oder eine IBAN ausgegeben wird, obwohl gar keine Kontobeziehung zum fraglichen Finanzinstitut vorliegt. Der Lizenzgeber schliesst diesbezüglich sowie allgemein die Gewährleistung und Haftung soweit gesetzlich zulässig vollumfänglich aus.*

4.5 Laufzeitbeschränkung und Releases des IBAN-Tools

Ungeachtet der vorgesehenen, umfassenden Tests sind das IBAN-Tool und die daraus berechneten IBANs nur so gut wie die hinterlegten Algorithmen und deren Aktualität.

Änderungen im Banken-Stamm aufgrund von Fusionen, Änderung in der IID-Struktur der IBAN bei Zentralisierungen und Änderungen der Kontonummernstruktur in Zusammenhang mit einem Wechsel der Informatikplattform beim Finanzinstitut können jedoch nicht ausgeschlossen werden. Solche Änderungen könnten unter Umständen dazu führen, dass für die betreffenden IIDs unkorrekte IBANs errechnet werden.

Um dies zu verhindern, schaltet SIX Interbank Clearing über die gesamte Laufzeit des IBAN-Tools 2x jährlich neue Releases mit den entsprechenden Änderungen in den Algorithmen oder Hilfstabellen auf.

Damit die Benutzer nicht mit einer alten Version arbeiten, wird das IBAN-Tool zusätzlich mit einer Laufzeitbeschränkung versehen. Nach Ablauf dieser Zeit setzt sich das IBAN-Tool ausser Betrieb und weist allfällige Input-Records als nicht verarbeitbar zurück. Spätestens am Ende der jeweiligen Laufzeit muss deshalb die neuste IBAN-Tool-Version von der Webseite von SIX Interbank Clearing (www.iban.ch) heruntergeladen werden.

Neue Releases mit aktuell nachgeführten Algorithmen werden alle 6 Monate aufgeschaltet.

4.6 Konversion der abgespeicherten Stammdaten

Im Prinzip wird zwischen der Massenmigration abgespeicherter Stammdaten sowie der Bereinigung/Validierung neu erfasster Stammdaten im Rahmen der täglichen ZV-Verarbeitung unterschieden.

5 Technische Konzeption des IBAN-Tools

Das IBAN-Tool ist in der Form einer «Black Box» mit verschiedenen Input- und Output-Schnittstellen konzipiert.

5.1 Programmierung in «Java» und als «Windows DLL»

Das IBAN-Tool wird in zwei Versionen angeboten, einerseits in der Programmiersprache Java und andererseits als Windows DLL mit C-Schnittstelle.

Die Java-Version wird auf Basis des Java-Releases J2SE 1.7 und unter Beizug der von Apache bereitgestellten und frei verfügbaren Entwicklungsumgebung NetBeans® IDE 8.0 entwickelt. Die Auslieferung des IBAN-Tools wird als Java Archive (jar) erfolgen. Bei der Java-Version handelt es sich um ein vollständig plattformunabhängiges Werkzeug, wobei der Betrieb das Vorhandensein einer geeigneten Version des Java Runtime Environment (JRE) auf der vorgesehenen Informatikplattform voraussetzt. Geeignete Versionen sind 1.4 sowie 5.0 bis 8.

Die Windows DLL mit C-Schnittstelle wird mit Microsoft Visual C++ 6.0 entwickelt und beinhaltet die «Microsoft Foundation Classes» (MFC) als integralen Bestandteil (statisch eingebunden). Diese DLL kann durch jede Windows-Anwendung geladen und aufgerufen werden. Das IBAN-Tool als Windows DLL ist unter allen Windows-Betriebssystemen ab Windows Vista lauffähig.

Hinweis

Der Einfachheit halber wird nachstehend – mit Ausnahme der unterschiedlichen technischen Anforderung in Kapitel 7 – jeweils nur vom IBAN-Tool in Einzahl gesprochen, da Funktionen, Anwendungsmöglichkeiten und Konversionsumfang bei beiden Versionen identisch sind.

5.2 Elemente des IBAN-Tools

Vereinfacht ausgedrückt ist das IBAN-Tool aus drei Hauptkomponenten aufgebaut:

- einer **Input-/Output-Schnittstelle** in Form von verbindlich definierten Record-Strukturen (Kapitel 6), auf deren Basis bei der Java- bzw. Windows-DLL-Version unterschiedliche technische Startparametrisierungen und Methodenaufrufe (Java-Version: → XML-/ASCII-Records oder Direktaufruf aus anderen Applikationen. Windows-DLL-Version: → Direktaufruf aus anderen Windows-Applikationen) zu berücksichtigen sind (Details siehe Kapitel 7 und 8);
- mehreren **Hilfsroutinen und Tabellen** (insbesondere ein erweiterter Banken-Stamm).
- dem eigentlichen **Berechnungsmodul**, in welchem die Algorithmen der bankproprietären Kontonummernstrukturen und deren Konversionsregeln für die Generierung einer IBAN hinterlegt werden.

Für die Anwender relevant sind die verschiedenen Input-/Output-Schnittstellen.

6 Formate der Input-/Output-Datenfelder

Die Input-/Output-Schnittstelle ist als Verbindungsglied zwischen den unterschiedlichen ZV-Applikationen der Finanzinstitute sowie der Zahlungspflichtigen einerseits und dem Kern des IBAN-Tools mit den Berechnungsalgorithmen und Hilfstabellen andererseits zu verstehen.

Die Formate der Input-/Output-Datenfelder sind von Anfang an so ausgelegt, dass eine spätere Anpassung wenn immer möglich vermieden werden kann, da jede Anpassung Implikationen auf die von den Finanzinstituten und Zahlungspflichtigen (bzw. deren Software-Firmen) entwickelten Verarbeitungssoftware haben könnte.

6.1 Struktur der Datenfelder

Nachstehend werden die Input-/Output-Datenfelder auf der Basis eines XML-Records beschrieben (siehe XML-Record-Aufbau unter Ziffer 6.1.1 und 6.1.2).

Formate, Ausprägungen und Dateninhalte sind jedoch auch für die anderen Input-Output-Schnittstellen (ASCII-Records gemäss Ziffer 7.5.1), Direktaufruf Java (gemäss Ziffer **Fehler! Verweisquelle konnte nicht gefunden werden.**) und Direktaufruf Windows-DLL (gemäss Ziffer 8.2) verbindlich. Beim Direktaufruf aus anderen Applikationen (Java und Windows-DLL) entfallen lediglich die beiden Felder 01 (Sequenznummer) und 02 (Individuelle Kundenreferenz), da hier jeweils jeder Record einzeln abgearbeitet wird und diese Felder somit überflüssig wären.

6.1.1 Input-Datenfelder (Beispiel anhand eines XML-Records)

Feld	Kurzbeschreibung	XML-Bezeichnung	Format	Input	Ausprägung	Bemerkungen
01	Sequenznummer	<IBANRECORD SEQNR>	=6n	ma		Fortlaufend nummeriert, beginnend mit 000001
02	Individuelle Kundenreferenz	<INDKREF>	35x	op	frei	Kann von der Applikation des Zahlungspflichtigen – sofern für die Verarbeitung des Output-Records notwendig – mit einem eindeutigen Identifikationsmerkmal des Kundenstammrecords versehen werden. (Bei ASCII-Dateien sind in der Individ. Kundenref. keine «;» erlaubt, da diese als «Feldabgrenzer» der ASCII-Datei eingesetzt werden).
03	IID / Postkonto-Nr. / SWIFT-BIC des Finanzinstituts	<BCPC>	11x	ma	frei	Falls sowohl IID wie Postkonto-Nummer oder der SWIFT-BIC des Finanzinstituts des Zahlungsempfängers abgelegt sind, ist jeweils die IID abzufüllen.
04	Kontonummer	<KOZE>	34x	ma		Falls Kontonummern sowohl in proprietärer Form wie auch aus ES-Codierzeile abgelegt sind, ist jeweils die Kontonummer ES abzufüllen.

Legende zu den Spalten «Format» und «Input» (bei Ziffer 6.1.1) resp. «Output» (bei Ziffer 6.1.2):

ma = obligatorisch (mandatory), co = bedingt (conditional), op = fakultativ (optional)

x = alpha-numerisch (Feld mit variabler Länge), n = numerisch

[=n]= numerisches Feld mit vorgegebener fixer Länge, allenfalls mit vorlaufenden Nullen abgefüllt

6.1.2 Output-Datenfelder (Beispiel anhand eines XML-Records)

Im Output-Datenrecord werden sämtliche im Input-Datenrecord enthaltenen Felder unverändert wieder ausgeliefert. Der Output-Datenrecord enthält ausserdem noch weitere Felder mit den diversen Konversionsergebnissen. Bei den Schnittstellenversionen mit Direktaufruf entfallen somit die beiden Felder 01 und 02 analog zu den Inputdaten.

Feld	Kurzbeschreibung	XML-Bezeichnung	Format	Input	Ausprägung	Bemerkungen
01	Sequenznummer	<IBANRECORD SEQNR>	=6n	ma		Fortlaufend nummeriert, beginnend mit 000001
02	Individuelle Kundenreferenz	<INDKUREF>	35x	co		Gleicher Inhalt wie im Input-Record
03	IID / Postkonto-Nr. / SWIFT-BIC des Finanzinstituts	<BCPC>	11x	ma		Gleicher Inhalt wie im Input-Record
04	Kontonummer	<KOZE>	34x	ma		Gleicher Inhalt wie im Input-Record
05	Validierungsflag	<VFLAG>	=2n	ma	korrekter Input: Codes 01 - 09 fehlerhafter Input: Codes 10 - 29	Bei Code 10 - 29 ist der Zahlungspflichtige oder das Finanzinstitut aufzufordern, die Stammdaten entweder zu löschen oder korrekte Daten beim Kontoinhaber anzufordern. Flag 06/07 kommen nur in standardisiertem Nachbereinigungsrecord vor.
06	IID des Finanzinstituts	<BCZEFI>	5x	co	Korrekte IID des Finanzinstituts, falls Validierungsflag (Feld 05) = 01 - 04	IID bzw. PC-Nr. kann von Feld 03 abweichen, falls das Finanzinstitut neu eine Sammel-IID bzw. eine Sammel-PC-Nr. verwendet und die alten IID bzw. PC-Nr. in den Stammdaten zu ersetzen sind.
07	Postkonto-Nummer (PC-Nr.) des Finanzinstituts	<PCZEFI>	11x	co	Korrekte PC-Nr. des Finanzinstituts, falls Validierungsflag (Feld 05) = 01 - 04 Die PC-Nr. wird im Format: 99-ZZZZZ9-9 ausgeliefert	
08	IBAN	<IBAN>	=21x	co	IBAN wird gemäss Standard des jeweiligen Finanzinstituts generiert, falls Validierungsflag (Feld 05) = 01 - 04	Proprietäre Kontonummer und oder ES-Codierzeile kann in den abgespeicherten Stammdaten durch die IBAN ersetzt werden

Feld	Kurzbeschreibung	XML-Bezeichnung	Format	Input	Ausprägung	Bemerkungen
09	E-Mail des Finanzinstituts des Zahlungsempfängers	<MAILZEFI>	50x	co	E-Mail-Adresse des Finanzinstituts des Zahlungsempfängers	RESERVE Feld 09 wird nicht abgefüllt (war für Phase II vorgesehen)

Die Felder 06 bis 08 enthalten nur dann einen Inhalt, wenn der Validierungsflag in Feld 05 = 01 bis 04 ist!

Bedeutung der Validierungsflags (Feld 05)

korrekter Input	
01	korrekte Kontonummernstruktur in Inputdaten (Prüfziffer in proprietärer Konto-Nr. validiert) → IBAN errechnet
02	korrekte Kontonummernstruktur in Inputdaten (keine Prüfziffer-Validierung in proprietärer Konto-Nr.) → IBAN errechnet
03	CH-/LI-IBAN in Input-Record → IBAN nach Prüfung von Länge, PZ und IID in Output-Record übernommen
04	Postkonto-Nummer des PostFinance-Kunden in Input-Record kann durch IBAN ersetzt werden
05	korrekte Inputdaten aus 27-stelliger ES-Codierzeile (ES-Prüfziffer validiert) → IBAN errechnet
06	Reserve
07	Reserve
08	korrekte CH-/LI-IBAN-Struktur in Input-Record, aber falsche IID → IBAN neu gerechnet
09	korrekte Inputdaten aus Positionen 11-26 der 27-stelligen ES-Codierzeile (ES-Prüfziffer nicht vorhanden) → IBAN errechnet

fehlerhafter Input	
10	ungültige Daten in Feld «IID/PC-Nr. / SWIFT-BIC» → Errechnung IBAN nicht möglich
11	Für diese IID bzw. Postkonto-Nummer kann keine IBAN errechnet werden (Grund: Bank nimmt generell nicht an dieser Dienstleistung teil oder verkettete IID mit neuer Kontonummer nach Fusion)
12	IID unbekannt → Errechnung IBAN nicht möglich
13	Prüfziffer falsch in IID → Errechnung IBAN nicht möglich
14 -19	weitere Fehlercodes IID, nicht definiert, keine Freigabe
20	ungültige Daten in Feld «Proprietäre Kontonummer» → Errechnung IBAN nicht möglich
21	falsche CH-/LI-IBAN Struktur in Input-Record → Validierung IBAN nicht möglich
22	proprietäre Kontonummer oder ES-Codierzeile fehlerhaft (Prüfziffer-Fehler) → Errechnung IBAN nicht möglich
23	Inputdaten gemäss Algorithmus unsicher → keine IBAN gerechnet
24	Reserve
25	Konversion proprietäre Kontonummer in IBAN durch Finanzinstitut des Zahlungsempfängers ausgeschlossen → keine IBAN gerechnet
26	IBAN ist fehlerhaft (Prüfziffer-Fehler) oder ist wegen alter IID nicht mehr gültig → Inputdaten sollten gelöscht werden
27	Daten aus Feld «IID / PC-Nr./ SWIFT-BIC» und IID in eingelesener IBAN gehören nicht zu zusammen → Inputdaten sollten gelöscht werden
28	Reserve, keine Freigabe
29	Formatfehler in Input-Record → Record nicht verarbeitet

Fehlermeldung nach Überschreitung der Laufzeitbeschränkung (wird nur bei direktem Methodenaufruf aus Java- oder Windows-DLL-Version generiert)	
31	IBAN-Tool abgelaufen → keine Konversion mehr möglich / vorgängiger Download neuer IBAN-Tool-Release erforderlich

Die Bedeutung der Validierungsflags ist bei allen Schnittstellen-Versionen identisch. Einzig Flag 31 entfällt bei der Massenverarbeitung auf Dateibasis, da nach Ablauf der Laufzeit des IBAN-Tools gar keine Dateien mehr eingelesen werden.

6.1.3 Output-Totalrecord bei Sammeldateien (Basis XML)

Ein Output-Totalrecord wird nur im Falle von Sammeldateien (ASCII- oder XML-Schnittstelle) generiert und dient zum Abstimmen des ausgelieferten Datenfiles und zur Überprüfung, ob alle Output-Datenrecords verarbeitet wurden.

Ausserdem wird der Output-Totalrecord mit Statistik-Daten ergänzt und kann somit auch zu statistischen Auswertungen weiterverwendet werden.

Feld	Kurzbeschreibung	XML-Bezeichnung	Format	Input	Bemerkungen
01	Sequenznummer	SEQNR	=7n	ma	Höchste Sequenznummer = Sequenznummer des letzten Datenrecords + 1
02	Anzahl Flag 01	<VFlag01>	6n	ma	Anzahl Input-Records, welche aufgrund der Verarbeitung im IBAN-Tool mit dem entsprechenden Validierungsflag versehen wurden
03	Anzahl Flag 02	<VFlag02>	6n	ma	
04	Anzahl Flag 03	<VFlag03>	6n	ma	
05	Anzahl Flag 04	<VFlag04>	6n	ma	
06	Anzahl Flag 05	<VFlag05>	6n	ma	
07	Anzahl Flag 06	<VFlag06>	6n	ma	Reserve
08	Anzahl Flag 07	<VFlag07>	6n	ma	Anzahl Input-Records, welche aufgrund der Verarbeitung im IBAN-Tool mit dem entsprechenden Validierungsflag versehen wurden
09	Anzahl Flag 08	<VFlag08>	6n	ma	
10	Anzahl Flag 09	<VFlag09>	6n	ma	
11	Anzahl Flag 10	<VFlag10>	6n	ma	
12	Anzahl Flag 11	<VFlag11>	6n	ma	
13	Anzahl Flag 12	<VFlag12>	6n	ma	Flag 14 - 19 nicht freigegeben
15 - 20	Anzahl pro Flag 14 - 19	<VFlag14>- <VFlag19>	6n	ma	
21	Anzahl Flag 20	<VFlag20>	6n	ma	
22	Anzahl Flag 21	<VFlag21>	6n	ma	Anzahl Input-Records, welche aufgrund der Verarbeitung im IBAN-Tool mit dem entsprechenden Validierungsflag versehen wurden
23	Anzahl Flag 22	<VFlag22>	6n	ma	
24	Anzahl Flag 23	<VFlag23>	6n	ma	
25	Anzahl Flag 24	<VFlag24>	6n	ma	Reserve
26	Anzahl Flag 25	<VFlag25>	6n	ma	Anzahl Input-Records, welche aufgrund der Verarbeitung im IBAN-Tool mit dem entsprechenden Validierungsflag versehen wurden
27	Anzahl Flag 26	<VFlag26>	6n	ma	
28	Anzahl Flag 27	<VFlag27>	6n	ma	
29	Anzahl Flag 28	<VFlag28>	6n	ma	Reserve
30	Anzahl Flag 29	<VFlag29>	6n	ma	Anzahl Input-Records mit Flag 29
31	Total ausgelieferte Datenrecords	<Recordcounter>	7n	ma	Muss identisch mit der Anzahl Input-Records sein

6.2 Genereller Record-Aufbau im Falle der XML-/ASCII-Schnittstelle

Im Falle der XML-/ASCII-Schnittstelle können Input-Records wahlweise als Einzelrecord oder als Sammeldatei ins IBAN-Tool eingeliefert werden.

Variabel lang definierte Felder sind sowohl beim XML wie auch beim ASCII linksbündig abzufüllen. Sie können – falls gewünscht – mit Blanks auf die maximale Länge aufgefüllt werden (fixe Recordlänge).

Obligatorische Felder müssen angeliefert werden (allenfalls auch mit Blanks abgefüllt).

Spezielle Regeln für die ASCII-Datei

Bei der ASCII-Datei müssen auch die fakultativen Felder erstellt werden (mit oder ohne Inhalt). Als Feldbegrenzer dient ausschliesslich das Semikolon «;».

Die Feld-Nummern und Feld-Bezeichnungen (Header) sind nicht abzufüllen.

6.2.1 Einzelrecord

Einzelrecords bestehen aus jeweils einem einzelnen Datenrecord.

6.2.2 Sammeldateien

Sammeldateien bestehen aus 1 - n Datenrecords.

6.2.3 Output-Daten

Die Output-Daten bestehen – analog zur Einlieferung – aus Einzelrecords oder Sammeldateien, die um die notwendigen Felder ergänzt werden.

Ausserdem wird bei Output-Daten – im Falle von Sammeldateien – zusätzlich ein Totalrecord mitgeliefert. Dieser Totalrecord enthält Statistik-Daten und er kann gleichzeitig zu Abstimmzwecken verwendet werden.

7 Technische Komponenten der Java-Version

7.1 Mögliche Java-Versionen

Das Java-IBAN-Tool kann sowohl auf den verschiedenen, bei den Finanzinstituten betriebenen Informatik-Plattformen als auch im Windows-, MAC- und Unix-Umfeld der bei den Zahlungspflichtigen im Einsatz befindlichen PC/Server eingesetzt werden.

Als Basis zur Ausführung des IBAN-Tools Release wird eine geeignete Version des Java Runtime Environment (JRE) benötigt. Zu beachten ist, dass das JRE neuerer Versionen publiziert sind. Es existieren verschiedene Versionen des JRE für MS Windows, Linux, Solaris SPARC und Solaris x86.

Sollte kein oder ein anderes Java Runtime Environment in Betrieb sein, muss für das entsprechende Betriebssystem die JRE von der Webseite <https://java.com/de/download/> installiert werden.

Um sich zu vergewissern, ob und wenn ja, welches JRE momentan auf Ihrer Informatik-Plattform im Betrieb ist, kann auf der Kommandozeile der folgende Befehl verwendet werden: `java -version`

7.2 Input-/Output-Schnittstellen der Java-Version

Die Input-/Output-Schnittstellen werden einerseits für die Recordformate

- XML (nur Java-Version)
- ASCII / (CSV-Datei)

realisiert und andererseits als

- Direktinput-Schnittstelle (Direkter Methodenaufruf aus anderen Java-Applikationen) angeboten.

Der Output wird stets im gleichen Recordformat/Datenformat wie die Einlieferung ausgeliefert. Beispiele von Input- und Output-Records/-Daten in den verschiedenen Ausprägungen sind unter Ziffern 7.5.1 und 0 aufgeführt.

XML- und ASCII-Records/-Dateien können sowohl für Massenmutationen wie auch für Einzeltransaktionen eingesetzt werden, der Direktaufufruf setzt die Abarbeitung einzelner Transaktionen voraus.

Auf der Basis des Direktaufufrufs arbeitet auch das unter Ziffer 7.6 aufgeführte GUI für Einzelabfragen.

7.3 Optimierungsaspekte bei Massenmutationen

Als Konsequenz dieser weitgehenden «Plattform-Unabhängigkeit» muss das unter Ziffer 5.1 erwähnte «Java Runtime Environment» jeweils aktiviert werden, wenn das IBAN-Tool aufgerufen und eine Input-Datei abgesetzt wird.

Der dafür zu veranschlagende Zeitbedarf (im Sekundenbereich) gilt es bei der Konzeption der von der jeweiligen Applikation zu realisierenden Input-/Output-Schnittstelle durch die Finanzinstitute und Software-Firmen zu berücksichtigen.

Bei Massenmigrationen und grösseren Datenmengen in der täglichen Verarbeitung gemäss Ziffer 9.3 empfiehlt sich deshalb die Generierung von Sammeldateien (siehe Ziffer 6.2.2). Deren Aufbereitung und Abarbeitung müsste im Hintergrund geschehen.

Bei der täglichen Verarbeitung von Zahlungsbelegen und Output-Daten durch den Zahlungsempfänger sind demgegenüber Einzelrecords (siehe Ziffer 6.2.1) bzw. Direktaufufrufe Java (siehe Ziffer **Fehler! V erweisquelle konnte nicht gefunden werden.**) oder Windows DLL (siehe Ziffer 8.2.3) zweckmässig, da die beiden Tools sehr schnell sind und hier im Falle der Java-Version auch der Zeitbedarf für die Aktivierung der «Java Runtime Environment» nicht ins Gewicht fällt.

7.4 Startparametrisierung und Kommandozeile

Bei Windows-Systemen ist die Kommandozeile (MS Eingabeaufforderung) im Startmenu «Programme» oder «Alle Programme», «Zubehör» zu finden. Alternativ kann im Startmenu bei «Ausführen...» der Befehl «cmd» angewandt werden.

Bei der Startparametrisierung ist zwischen Massenverarbeitung (Verarbeitung der Test-Inputdaten oder Verarbeitung eigener Input-Dateien) und der Einzelabfrage (Aufruf des GUI) zu unterscheiden.

Startparameter des IBAN-Tools:

	Parameter	Kurzbeschreibung
Inputformat	-a -x	Dokumentformat des Input-Records liegt in ASCII vor. Dokumentformat des Input-Records liegt in XML vor.
Input Quelle	-i	Pfad und Name der Inputdatei innerhalb von Hochkommata. Bsp: -i "c:/xmlfiles/input.xml"
Output Ziel	-o	Pfad und Name der Outputdatei innerhalb von Hochkommata. Bsp: -o "c:/xmlfiles/output.xml"
Benutzeroberfläche (GUI)	-g	Startet die Benutzeroberfläche für Einzelabfragen und Einsicht ins Logbuch.
Sprache (GUI)	-l	Sprache des GUI bei Einzelabfrage, gültige Werte: D (Deutsch), F (Französisch), I (Italienisch), E (Englisch)
Versionsabfrage	-v	Gibt die Version und das Ablaufdatum des installierten IBAN-Tools aus.
Recordart	-e	Erweiterter Input-Record (nicht freigegeben).

Syntax¹

```
java -jar IBANTool.jar [-a | -x] [-i Inputpfad] [-o Outputpfad] [-l Sprache]
```

Beispiel: Aufruf mit XML-Beispieldateien:

```
java -jar c:/IBAN/IBANTool.jar -x -i "c:/IBAN/In/input.xml" -o "c:/IBAN/Out/output.xml"
```

Beispiel: Aufruf mit ASCII-Beispieldateien:

```
java -jar C:/IBAN/IBANTool.jar -a -i "c:/IBAN/In/input.csv" -o "C:/IBAN/Out/output.csv"
```

Die Benennung der Dateinamen hinter dem Quell- und Zielpfad ist frei wählbar. Bei parallelen Berechnungen soll durch gewählte Namensgebungen durch die Benutzer ungewolltes Überschreiben der Dateien verhindert werden.

Massenverarbeitung

```
java -jar IBANTool.jar [-a | -x] [-i Inputpfad] [-o Outputpfad] [-g] [-v]
```

Beispiel: Aufruf mit XML-Dateien:

```
java -jar c:/IBAN/IBANTool.jar -x -i "c:/IBAN/In/input.xml" -o "c:/IBAN/Out/output.xml"
```

Beispiel: Aufruf mit ASCII-Dateien:

```
java -jar C:/IBAN/IBANTool.jar -a -i "C:/IBAN/In/input.csv" -o "C:/IBAN/Out/output.csv" -g
```

¹ Bemerkung: je nach System müssen Sie Backslash \ statt Slash / im Pfad verwenden.

Einzelabfrage

```
java -jar IBANTool.jar [-g] [-l Sprache]
```

Beispiel:

```
java -jar C:/IBAN/IBANTool.jar -g -l D
```

Versionsangabe

```
java -jar C:/IBAN/IBANTool.jar -v
```

Erläuterung

Die Benennung der Dateinamen hinter dem Quell- und Zielpfad ist frei wählbar. Bei parallelen Berechnungen soll durch gewählte Namensgebungen durch die Benutzer ungewolltes Überschreiben der Dateien verhindert werden.

Das `-g` für *GUI-Oberfläche*, `-v` für *Versionsangabe* und `-l` für *Language* (Sprache) sind optional.

Als beim Start ausgewählte Sprache stehen zur Verfügung: «d» für Deutsch, «e» für Englisch, «f» für Französisch und «i» für Italienisch. Die Sprache ist nur für die grafische Einzelabfrage verfügbar. Ohne Angabe der Sprache wird Deutsch als Standardsprache für die Einzelabfrage angezeigt. Das GUI für die grafische Resultatanzeige der Massenverarbeitung ist nur in Englisch gehalten.

7.5 Java Interface / Schnittstellenspezifikation IBAN-Tool Java

Die Java Version des IBAN-Tools ist als `.jar` Datei erhältlich. Darin sind sämtliche Dateien und Informationen enthalten um Umrechnungen in IBAN durchzuführen.

Dank der offenen Architektur von Java können so Umrechnungen direkt aus einem anderen Java-Programm heraus aufgerufen werden («Direkter Methodenaufruf» aus anderen Java-Programmen),

```
Package ch.sic.ibantool
```

Es werden zwei Klassen für die Umrechnung verwendet: `Class RecordIBAN`

(Enthält die Input und Output Daten eines Records) `Class Main`

(Enthält die Methoden für den Aufruf der Umrechnung)

Die Klassen im Detail

`Class RecordIBAN`

<code>StringBuffer IndKuRef</code>	Individuelle Kundenreferenz	Input
<code>StringBuffer BCPC</code>	IID (oder PC/SWIFT)	Input
<code>StringBuffer KoZe</code>	Kontonummer	Input
<code>StringBuffer VFlag</code>	Validierungsflag	Output
<code>StringBuffer BCZeFi</code>	IID ZE-FI	Output
<code>StringBuffer PCZeFi</code>	PC-Nummer ZE-FI	Output
<code>StringBuffer Iban</code>	IBAN	Output

`Class Main`

```
IBANConvert(RecordIBAN record)
```

```
IBANConvert(StringBuffer BCPC, StringBuffer KoZe)
```

```
IBANConvert(StringBuffer IndKuRef, StringBuffer BCPC, StringBuffer KoZe)
```

Alle drei Varianten der Methode «IBANConvert» geben ein Objekt der Klasse «Record-IBAN» zurück.

Während der *ersten* Verwendung der Methode «IBANConvert» wird der Bankenstamm eingelesen. Wird «IBANConvert» innerhalb einer Schleife verwendet, ist deshalb zu beachten, dass die Instanz der Klasse `Main` im Speicher bleibt, d.h. ausserhalb der Schleife initialisiert wird. Falls dies nicht berücksichtigt wird, kann es zu unerwünschten Performanceeinbrüchen kommen, weil für jede einzelne Umrechnung der Bankenstamm neu eingelesen wird.

Anwendungsbeispiel

```
public static void main(String[] args) {
    ch.sic.ibantool.Main ibanclass = new ch.sic.ibantool.Main();
    ch.sic.ibantool.RecordIban recordiban;

    // Method call with StringBuffers
    recordiban = ibanclass.IBANConvert(new StringBuffer("1234"), new
    StringBuffer("768"), new StringBuffer("250109317507"));
    // or
    recordiban = ibanclass.IBANConvert(new StringBuffer("80-151-4"), new
    StringBuffer("3525-8.888766.2"));

    // Method call with RecordIban class
    recordiban = new ch.sic.ibantool.RecordIban ();
    recordiban.BCPC = new StringBuffer("POFICHBEXXX");
    recordiban.KoZe = new StringBuffer("30-307396-9");
    recordiban = ibanclass.IBANConvert(recordiban);

    // Output Result
    System.out.println("BC: " + recordiban.BCZeFi.toString());
    System.out.println("PC: " + recordiban.PCZeFi.toString());
    System.out.println("IBAN: " + recordiban.Iban.toString());
    System.out.println("Flag: " + recordiban.VFlag.toString());
}
```

7.5.1 Input-/Output-Records/-Dateien im ASCII-Format

Beim ASCII-Format (CSV-Format) sind sämtliche im Input-Record bzw. im Output-Record definierten Felder obligatorisch.

Die einzelnen Felder werden durch einen Feldseparator «;» abgetrennt (Semikolon). Fakultative Felder müssen zumindest den Feldseparator aufweisen. D.h. die Anzahl Feldseparatoren pro Datenzeile ist immer gleich.

In den Datenfeldern selbst (z.B. Kundenreferenz, Kontonummer, usw.) ist der Feldseparator «;» ein nicht erlaubtes Zeichen!

Die Feldlänge der nicht mit =n gekennzeichneten Felder ist variabel. Die Felder können jedoch auch gemäss Felddefinition in fixer Länge eingeliefert werden. In diesem Falle sind die Daten linksbündig abzufüllen, rechtsbündig mit Blanks aufgefüllt.

Beispiel eines Input-Records im ASCII-Format:

SEQNR-ID (=6n) ²	INDUKUREF-ID. (35x)	BCPC-ID (11x)	KOZE-ID (34x)
--------------------------------	------------------------	------------------	------------------

mit variabler Länge

000001;; 8271; 137279.001.11;

mit fixer Länge

001234;	WW-5567934469BBBBBBBBB;	230BB;	K0-096551,4;
---------	-------------------------	--------	--------------

² Die Länge der SeqNr-ID ist fix festgelegt: fehlen die bei kleineren Zahlen die vorlaufenden Nullen (z.B. bei Erstellung des CSV durch Microsoft Excel), so wird kein Output errechnet.

Beispiel eines Output-Records im ASCII-Format (Darstellung mit variabler Länge):

Sequenznummer	Individ. Kundenreferenz	ID/ Postkonto-Nr./ SWIFT-Adresse oder BIC des Finanzinstituts	Kontonummer des Zahlungsempfängers	Validierungsflag	ID des Finanzinstituts (Output)	Postkonto-Nummer des Finanzinstituts (Output)	IBAN	Email ZE-FI
(=6n)	(35x)	(11x)	(34x)	(=2n)	(5x)	(11x)	(=21x)	(50x)

```
000001;3455-2;8271;137279.001.11; 02;8271;80-3119-3;CH3708271013727900111;;
000002;776.aab;755;12.5-6651.0;24;755;45-20-3;;iban@bank.ch;
000003;a22bb1;766;K00965514;01;766;20-136-4;CH8500766000K00965514;;
```

Beispiel eines Output-Totalrecords:

Sequenznummer	Flags 01-05	Flags 06-09	Flags 10-13	Flags 14-19	Flags 20-23	Flags 24-29	Total ausgelieferte Datenrecords
(=7n)	je (6n)	(7n)					

0002427;	803;	0;	3;	0;	235;	0;	2426;
	1013;	0;	12;	0;	2;	0;	
	3;	0;	1;	0;	1;	0;	
	71;	0;	0;	0;	0;	0;	
	281;			0;		0;	
				0;		0;	

7.5.2 Input-/Output-Records-/Dateien im XML-Format

Das Input und Output XML Format besteht aus einem `<INPUT>` respektive `<OUTPUT>` Rotelement.

Dieses enthält ein Element `<IBANRECORDLIST size="6">` mit dem Attribut `size`, das die Anzahl Unterelemente von `<IBANRECORDLIST size="6">` angibt.

Die Unterelemente von `<IBANRECORDLIST size="6">` sind die `<IBANRECORD SEQNR="1">` Elemente.

Der `<IBANRECORD SEQNR="1">` besteht gemäss Spezifikation aus folgenden weiteren Unterelementen:

```
<INDKUREF> 110.3256</INDKUREF>
<BCPC > 80896</BCPC>
<KOZE > 19385.89</KOZE>
```

Das Attribut `SEQNR` innerhalb des `<IBANRECORD SEQNR="1">` entspricht der I/O spezifizierten Sequenznummer.

Input XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<INPUT>
```

```
  <IBANRECORDLIST size="6">
    <IBANRECORD SEQNR="000001">
      <INDKUREF> 1258.365 </INDKUREF>
      <BCPC > 230</BCPC>
      <KOZE > A-10.2350.26-01</KOZE>
    </IBANRECORD>
    <IBANRECORD SEQNR="000002">
      <INDKUREF> 1258.365 </INDKUREF>
      <BCPC > 230</BCPC>
      <KOZE > A-10.2350.26-01</KOZE>
    </IBANRECORD>
```

```

<IBANRECORD SEQNR="000003">
  <INDKUREF> 1258.365 </INDKUREF>
  <BCPC > 230</BCPC>
  <KOZE > A-10.2350.26-01</KOZE>
</IBANRECORD>
<IBANRECORD SEQNR="000004">
  <INDKUREF> 1258.365 </INDKUREF>
  <BCPC > 230</BCPC>
  <KOZE > A-10.2350.26-01</KOZE>
</IBANRECORD>
<IBANRECORD SEQNR="000005">
  <INDKUREF> 1258.365 </INDKUREF>
  <BCPC > 230</BCPC>
  <KOZE > A-10.2350.26-01</KOZE>
</IBANRECORD>
<IBANRECORD SEQNR="000006">
  <INDKUREF> 1258.365 </INDKUREF>
  <BCPC > 230</BCPC>
  <KOZE > A-10.2350.26-01</KOZE>
</IBANRECORD>
</IBANRECORDLIST>
</INPUT>

```

Output XML

```

<?xml version="1.0" encoding="UTF-8" ?>
<OUTPUT>
<CALC_DATE>14h44m30s_11-4-2006</CALC_DATE>
  <IBANRECORDLIST SIZE="6">
    <IBANRECORD SEQNR="000001">
      <INDKUREF> 1258.365</INDKUREF>
      <BCPC>230 </BCPC>
      <KOZE> A-10.2350.26-01</KOZE>
      <VFLAG>01</VFLAG>
      <BCZEFI> 230 </BCZEFI>
      <PCZEFI>80-2-2</PCZEFI>
      <IBAN>CH00002300A1023502601</IBAN>
    </IBANRECORD>
    <IBANRECORD SEQNR="000002">
      <INDKUREF> 1258.365</INDKUREF>
      <BCPC>230 </BCPC>
      <KOZE> A-10.2350.26-01</KOZE>
      <VFLAG>01</VFLAG>
      <BCZEFI> 230 </BCZEFI>
      <PCZEFI>80-2-2</PCZEFI>
      <IBAN>CH00002300A1023502601</IBAN>
    </IBANRECORD>
    <IBANRECORD SEQNR="000003">
      <INDKUREF> 1258.365</INDKUREF>
      <BCPC>230 </BCPC>
      <KOZE> A-10.2350.26-01</KOZE>
      <VFLAG>01</VFLAG>
      <BCZEFI> 230 </BCZEFI>
      <PCZEFI>80-2-2</PCZEFI>
      <IBAN>CH00002300A1023502601</IBAN>
    </IBANRECORD>
  </IBANRECORDLIST>

```

```

</IBANRECORD>
<IBANRECORD SEQNR="000004">
  <INDKUREF> 1258.365</INDKUREF>
  <BCPC>230 </BCPC>
  <KOZE> A-10.2350.26-01</KOZE>
  <VFLAG>01</VFLAG>
  <BCZEFI> 230 </BCZEFI>
  <PCZEFI>80-2-2</PCZEFI>
  <IBAN>CH00002300A1023502601</IBAN>
</IBANRECORD>
<IBANRECORD SEQNR="000005">
  <INDKUREF> 1258.365</INDKUREF>
  <BCPC>230 </BCPC>
  <KOZE> A-10.2350.26-01</KOZE>
  <VFLAG>01</VFLAG>
  <BCZEFI> 230 </BCZEFI>
  <PCZEFI>80-2-2</PCZEFI>
  <IBAN>CH00002300A1023502601</IBAN>
</IBANRECORD>
<IBANRECORD SEQNR="000006">
  <INDKUREF> 1258.365</INDKUREF>
  <BCPC>230 </BCPC>
  <KOZE> A-10.2350.26-01</KOZE>
  <VFLAG>01</VFLAG>
  <BCZEFI> 230 </BCZEFI>
  <PCZEFI>80-2-2</PCZEFI>
  <IBAN>CH00002300A1023502601</IBAN>
</IBANRECORD>
</IBANRECORDLIST>
</OUTPUT>

```

Output Total-Record in XML

```

<?xml version="1.0" encoding="UTF-8" ?>
<TOTALRECORD SEQNR="000115">
  <VFlag01>45</VFlag01>
  <VFlag02>2</VFlag02>
  <VFlag03>3</VFlag03>
  <VFlag04>6</VFlag04>
  <VFlag05>5</VFlag05>
  <VFlag06>0</VFlag06>
  <VFlag07>0</VFlag07>
  <VFlag08>0</VFlag08>
  <VFlag09>0</VFlag09>
  <VFlag10>0</VFlag10>
  <VFlag11>0</VFlag11>
  <VFlag12>0</VFlag12>
  <VFlag13>0</VFlag13>
  <VFlag14>0</VFlag14>
  <VFlag15>0</VFlag15>
  <VFlag16>0</VFlag16>
  <VFlag17>0</VFlag17>
  <VFlag18>0</VFlag18>
  <VFlag19>0</VFlag19>
  <VFlag20>0</VFlag20>

```

```

<VFlag21>0</VFlag21>
<VFlag22>0</VFlag22>
<VFlag23>45</VFlag23>
<VFlag24>5</VFlag24>
<VFlag25>0</VFlag25>
<VFlag26>0</VFlag26>
<VFlag27>0</VFlag27>
<VFlag28>0</VFlag28>
<VFlag29>3</VFlag29>
<Recordcounter>106</Recordcounter>
</TOTALRECORD>

```

7.5.3 Input-/Output-Daten bei Java-Direktaufruf

Das Feld `<IBANRECORD SEQNR>` kommt beim Java-Direktaufruf nicht vor, das Feld `<INDKUREF>` ist fakultativ, die anderen Felder sind obligatorisch. Bezüglich der Feldformate gelten die gleichen Auflagen wie unter Ziffer 6.1 beschrieben.

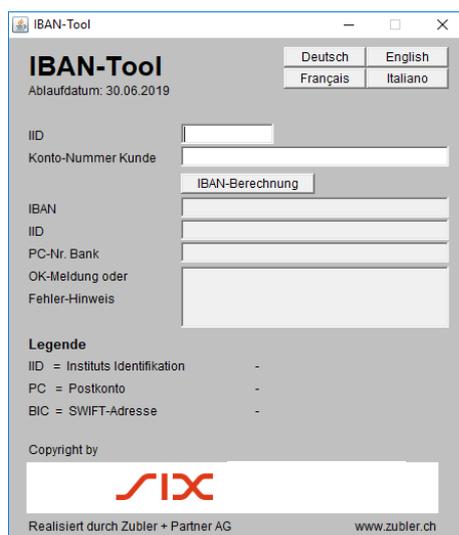
7.6 Benutzeroberfläche (GUI) für Java-Version

7.6.1 Einzelabfrage mit dem IBAN-GUI

Zusätzlich zum Hauptzweck des IBAN-Tools, der Konversion von abgespeicherten proprietären Kontonummern in einen IBAN, die aus den verschiedensten ZV-Applikationen gemäss Beschrieb in Kapitel 9 beschrieben wird, gibt es auch noch eine Funktion «Einzelabfrage», die auf der Basis des nachstehend beschriebenen GUI individuell aufgerufen werden kann.

Dieses GUI dient zur Durchführung von Einzelabfragen (wenn keine Input-/Output-Dateien generiert, sondern eine einzelne IBAN anhand der IID, Postkonto-Nummer, SWIFT-Adresse oder BIC (Bank Identifier Code) und der proprietären Kontonummer mit dem IBAN-Tool validiert werden).

Beispiel des GUI mit Einzelabfrage:



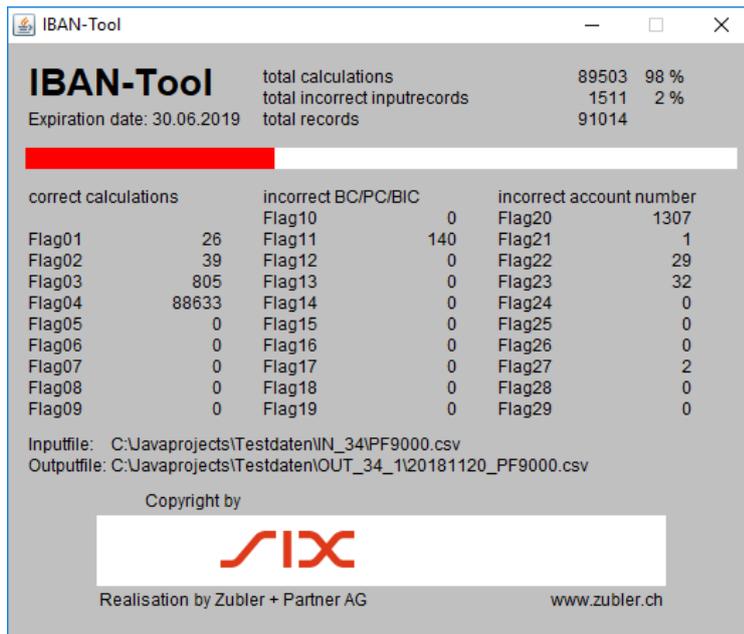
Der Unterschied dieser Einzelabfrage zu anderen, im Web aufgeschalteten, sog. «IBAN-Rechnern» liegt darin, dass – wie bei der Konversion von Inputdaten – die im IBAN-Tool hinterlegten, bankindividuellen Algorithmen beim IBAN-GUI ebenfalls durchlaufen werden und je nach Resultat ein «OK» oder ein entsprechender Fehler-Hinweis ausgegeben wird. Die Konversions-Qualität ist damit wesentlich höher als bei den vorerwähnten «IBAN-Rechnern», denen keine derartigen Algorithmen zugrunde gelegt sind.

7.6.2 Statusmeldungen mit dem IBAN-GUI

Ein zweites GUI dient der Visualisierung des Berechnungsfortschritts und der Umrechnungsstatistik bei Massenverarbeitungen (bei Angabe von Input-/Output-Dateien).

Zudem werden damit statistische Angaben aus der Verarbeitung der Umrechnungsergebnisse in der Benutzeroberfläche ausgegeben.

Beispiel des GUI mit Anzeige des Berechnungsfortschritts und der statistischen Daten bei Massenverarbeitung:



Die Benutzeroberfläche erscheint nur bei Startparameter -g

7.7 Installation des IBAN-Tools

Der Ordner IBAN ist am einfachsten in das Root-Verzeichnis (C:\ bei MS-Betriebssystemen) zu kopieren. Dieser beinhaltet den Bytecode des IBAN-Tools in der Datei IBANTool.jar, sowie einige Inputdaten. Die Inputdaten sind jeweils im ASCII- und XML-Format enthalten.

Natürlich kann ein anderes Verzeichnis als das vorgeschlagene Root-Verzeichnis gewählt werden. Dann ist die Startparametrisierung entsprechend anzupassen (Pfade der Input- und Outputdatei).

8 Technische Komponenten der Windows-Version

8.1 Voraussetzungen für Windows DLL

Das IBAN-Tool als Windows-DLL läuft auf allen Systemen ab Windows Vista. Bei anderen Betriebssystemen muss die Java-Lösung eingesetzt werden.

Im Gegensatz zur Java-Lösung ist keine Installation zusätzlicher «Runtime Environments» notwendig.

Im Gegensatz zur Java-Lösung verzichtet die Windows-DLL-Version auf Input-/Output-Records mit ASCII- oder XML-Dateien, sondern beschränkt sich auf einen Direktaufruf aus anderen Windows-Applikationen.

Konkret handelt es sich somit um eine Abarbeitung von einzelnen Stammdaten. Bei Massenmutationen von grossen Stammdatendateien ist dies entsprechend zu berücksichtigen.

8.2 Schnittstellenbeschreibung Windows DLL

Das IBAN-Tool als Windows DLL exportiert folgende Funktionen:

8.2.1 Funktion «IT_IBANConvert»

Diese Funktion konvertiert eine proprietäre Kontonummer in die entsprechende IBAN und retourniert einen Status an das aufrufende Programm.

Prototyp:

```
int IT_IBANConvert(char *pszKonto,           // [in]           mandatory
                  char *pszBCPC,           // [in]           mandatory
                  char *pszIBAN,           // [in/out]       mandatory
                  int  nIBANLen,           // [in]           mandatory
                  char *pszBC,             // [in/out]       optional
                  int  nBCLen,             // [in]           optional
                  char *pszPC,             // [in/out]       optional
                  int  nPCLen,             // [in]           optional
                  char *pszRESERVED,       // [in/out]       optional
                  int  nRESERVEDLen);      // [in]           optional
```

Parameter:

pszKonto	Pointer auf NULL-terminierten Buffer mit proprietärer, zu konvertierender Kontonummer
pszBCPC	Pointer auf NULL-terminierten Buffer mit der Identifikation des Finanzinstituts (IID, BIC oder PC-Nummer)
pszIBAN	Pointer auf Buffer im Adressraum des Aufrufers, in welchen das IBAN-Tool die umgerechnete IBAN schreiben soll.
nIBANLen	Länge des Buffers im Adressraum des Aufrufers, in welchen das IBAN-Tool die umgerechnete IBAN schreiben soll.
pszBC	Pointer auf Buffer im Adressraum des Aufrufers, in welchen das IBAN-Tool die (eventuell korrigierte) IID des Finanzinstituts schreiben soll.
nBCLen	Länge des Buffers im Adressraum des Aufrufers, in welchen das IBAN-Tool die (eventuell korrigierte) IID des Finanzinstituts schreiben soll.
pszPC	Pointer auf Buffer im Adressraum des Aufrufers, in welchen das IBAN-Tool die (eventuell korrigierte) PC-Nummer des Finanzinstituts schreiben soll.
nPCLen	Länge des Buffers im Adressraum des Aufrufers, in welchen das IBAN-Tool die (eventuell korrigierte) PC-Nummer des Finanzinstituts schreiben soll.
pszRESERVED	Reserve-Parameter für spätere Versionen. Wird derzeit ignoriert.
nRESERVEDLen	Reserve-Parameter für spätere Versionen. Wird derzeit ignoriert.

Return Wert:

int	Resultat-Flag gemäss Ziffer 6.1.2, Feld 05
-----	--

Anwendungsbeispiel: Aufruf aus C#:

```
[DllImport("IBANKernel.dll")]
static extern int IT_IBANConvert(    String sKonto,
    String sBCPC,
    StringBuilder lpszIBAN,
    int nIBANLen,
    StringBuilder lpszBC,
    int nBCLen,
    StringBuilder lpszPC,
    int nPCLen,
    StringBuilder lpszRESERVED,
    int nRESERVEDLen);

int nResult = 0;
StringBuilder sbIBAN = new StringBuilder(64);
StringBuilder sbBC = new StringBuilder(32);
StringBuilder sbPC = new StringBuilder(32);
StringBuilder sbRES = new StringBuilder(32);

nResult = IT_IBANConvert( "12345678.90A", // propr. Kontonummer
    "230", // Clearing-Nummer
    sbIBAN, // Buffer für IBAN
    sbIBAN.Capacity, // Länge des Buffers
    sbBC, // Buffer für BC neu
    sbBC.Capacity, // Länge des Buffers
    sbPC, // Buffer für PC
    sbPC.Capacity, // Länge des Buffers
    sbRES, // for later usage
    sbRES.Capacity); // for later usage
```

Anwendungsbeispiel: Aufruf aus Visual Basic .NET:

```
Imports System.Runtime.InteropServices
Imports System.Text
Module Module1

    <DllImport("IBANKernel.dll ")> _
    Public Function IT_IBANConvert( _
        ByVal sKonto As String, _
        ByVal sBCPC As String, _
        ByVal lpIBAN As StringBuilder, _
        ByVal nIBANLen As Integer, _
        ByVal lpBC As StringBuilder, _
        ByVal nBCLen As Integer, _
        ByVal lpPC As StringBuilder, _
        ByVal nPCLen As Integer, _
        ByVal lpRES As StringBuilder, _
        ByVal nRESLen As Integer) As Integer
    End Function
```

```

Sub Main()
    Dim sbIBAN As StringBuilder = New StringBuilder(255)
    Dim sbBC As StringBuilder = New StringBuilder(32)
    Dim sbPC As StringBuilder = New StringBuilder(32)
    Dim sbRES As StringBuilder = New StringBuilder(32)
    Dim nResult As Integer

    nResult = IT_IBANConvert( "111111.11P", _
                             "230", _
                             sbIBAN, _
                             sbIBAN.Capacity, _
                             sbBC, _
                             sbBC.Capacity, _
                             sbPC, _
                             sbPC.Capacity, _
                             sbRES, _
                             sbRES.Capacity)

    Console.WriteLine(sbIBAN)
    Console.WriteLine(sbPC)
End Sub
End Module

```

8.2.2 Funktion «IT_IBANVersion»

Diese Funktion informiert das aufrufende Programm über die eigene Versionsnummer sowie über das Ablaufdatum der Version. (Ist das Ablaufdatum einmal überschritten, errechnet das IBAN-Tool keine IBANs mehr.)

Prototyp:

```

int IT_IBANVersion( DWORD *pdwMajor,      // [in/out]      mandatory
                   DWORD *pdwMinor,     // [in/out]      mandatory
                   char *pszValidThru,   // [in/out]      mandatory
                   int  nValidThruLen); // [in]         mandatory

```

Parameter:

pdwMajor	Pointer auf DWORD Buffer im Adressraum des Aufrufers, in welchen das IBAN-Tool die Major-Version der DLL schreiben soll.
pdwMinor	Pointer auf DWORD Buffer im Adressraum des Aufrufers, in welchen das IBAN-Tool die Minor-Version der DLL schreiben soll.
pszValidThru	Pointer auf Buffer im Adressraum des Aufrufers, in welchen das IBAN-Tool das Ablaufdatum dieser Version schreiben soll. Das Datum wird in der Form yyyyymmdd geschrieben.
nValidThruLen	Länge des Buffers im Adressraum des Aufrufers, in welchen das IBAN-Tool das Ablaufdatum dieser Version schreiben soll. Die Länge des Buffers muss mindestens 9 Byte sein.

Return Wert:

int	Resultat-Flag: 1= OK, returnierte Daten OK 0 = nicht OK, Version kann nicht bestimmt werden
-----	--

Anwendungsbeispiel: Aufruf aus C#:

```
[DllImport("IBANKernel.dll")]
static extern int IT_IBANVersion(           ref int dwMajor,
                                           ref int dwMinor,
                                           StringBuilder lpzValidThru,
                                           ref int nValidThruLen);

StringBuilder sbValidThru = new StringBuilder(32);
int nMajor = 0;
int nMinor = 0;
int nResult = IT_IBANVersion(           ref nMajor, // Buffer für Major-V.
                               ref nMinor, // Buffer für Minor-V.
                               strDate, // Buffer für Datum
                               strDate.Capacity);
```

Anwendungsbeispiel: Aufruf aus Visual Basic .NET:

```
Imports System.Runtime.InteropServices
Imports System.Text
Module Module1

    <DllImport("IBANKernel.dll ")> _
    Public Function IT_IBANVersion( _
        ByRef nMajor As Integer, _
        ByRef nMinor As Integer, _
        ByVal lpValidThru As StringBuilder, _
        ByVal nValidThruLen As Integer) As Integer
    End Function

    Sub Main()
        Dim sbValidThru As StringBuilder = New StringBuilder(100)
        Dim nMajor As Integer = 0
        Dim nMinor As Integer = 0
        Dim nResult As Integer = IT_IBANVersion(nMajor, _
                                                nMinor, _
                                                sbValidThru, _
                                                sbValidThru.Capacity)

        Console.WriteLine(sbValidThru)
    End Sub

End Module
```

8.2.3 Input-/Output-Daten bei Windows-DLL-Direktaufuf

Die Input-/Output-Daten beim Windows-DLL-Direktaufuf sind unter Ziffer 8.2 beschrieben und mit Beispielen unterlegt.

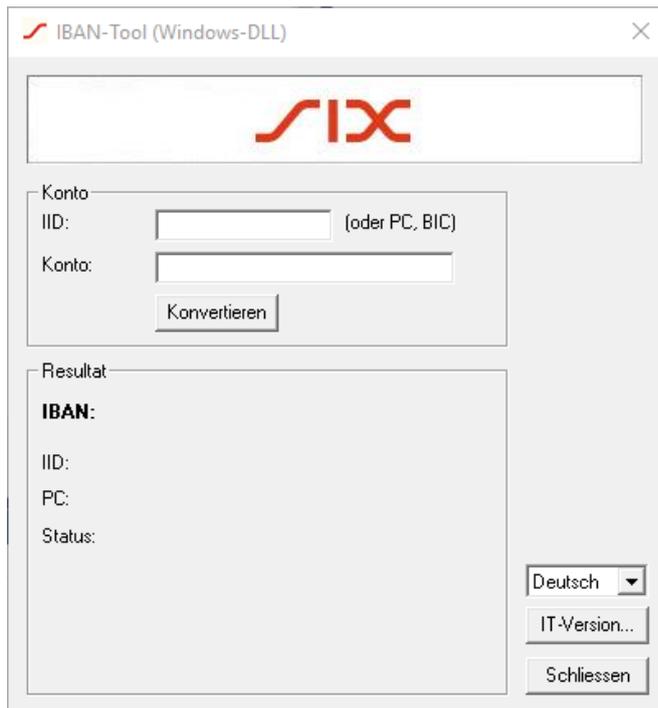
Die Felder **<IBANRECORD SEQNR>** und **<INDKUREF>** kommen beim Windows-DLL-Direktaufuf nicht vor, die anderen Felder sind obligatorisch. Bezüglich der Feldformate gelten die gleichen Auflagen wie unter Ziffer 6.1 beschrieben.

8.3 Benutzeroberfläche (GUI) für Windows-DLL-Version

8.3.1 Einzelabfrage mit dem IBAN-GUI

Mit der Windows-DLL-Version besteht ebenfalls die Möglichkeit, einzelne IBANs anhand der IID, PC-Nummer, SWIFT/BIC-Adresse und der proprietären Kontonummer zu berechnen.

Beispiel des GUI mit Einzelabfrage:



Das GUI der Windows-DLL-Version erfüllt die gleichen Funktionen wie die unter Ziffer 7.6.1 beschriebene Java-Version.

8.3.2 Statusmeldungen

Da bei der Windows-DLL-Lösung keine Massenmutationen auf Dateibasis angeboten werden, entfällt folgerichtig auch ein GUI für Statusmeldungen.

9 Einsatzmöglichkeiten des IBAN-Tools

9.1 Generelle Bemerkungen

Das vorliegende Kapitel hat weitgehend informativen Charakter. Es ist den einzelnen Finanzinstituten, Software-Firmen und Eigenrealisatoren überlassen, wie weit sie die Anregungen bei der individuellen Umsetzung ihrer Schnittstellen-Applikationen übernehmen wollen.

Als Minimalauflage muss jedoch sichergestellt werden, dass – in allen Fällen, in denen eine IBAN errechnet werden konnte – diese anstelle der bankproprietären Kontonummer abgespeichert und bei der Generierung der Zahlungsrecords gemäss den offiziellen SIC/DTA/LSV- bzw. EZAG-Standards weitergeleitet wird.

9.1.1 Haupteinsatzmöglichkeit des IBAN-Tools bei bankproprietären Kontonummern

Hauptzweck des IBAN-Tools ist die Konversion von bankproprietären Kontonummern in IBANs. Nachstehend wird unterschieden zwischen

- einmaliger Massenmigration von bankproprietären Kontonummern in IBANs und
- Konversion bankproprietärer Kontonummern bzw. von Kontoidentifikationen in den Codierzeilen der ES Banken in IBANs im Rahmen der täglichen Zahlungsverkehrs-Verarbeitung.

9.1.2 Zusätzliche Einsatzmöglichkeit des IBAN-Tools im Falle von Postkonto-Nummern

Im Prinzip handelt es sich auch bei der Postkonto-Nummer von Kunden der PostFinance um eine proprietäre Kontonummer. Da es sich dabei jedoch um einen seit langem bekannten Standard der PostFinance handelt, der in den ZV-Applikationen bei der Zahlungserfassung validiert wird, ist die Problematik der Non-STP-Transaktionen wesentlich geringer als bei bankproprietären Kontonummern. Die PostFinance hat sich deshalb im Rahmen des Projektes IBAN entschieden, die Darstellung ihrer Postkonto-Nummer nicht generell auf die IBAN umzustellen.

Sie ist jedoch in der Lage, die Postkonto-Nummern im IBAN-Format zu verarbeiten und gibt sie ihren Kunden – auf entsprechende Anfrage hin – auch ab.

Die PostFinance begrüsst es deshalb, wenn bei der Migration der gespeicherten Stammdaten auch die IBAN von Postkonto-Nummern – unter klaren Auflagen – abgespeichert werden. Diesem Aspekt wird in den folgenden Ziffern jeweils gesondert Beachtung geschenkt.

9.2 Minimale Erwartungen an den Aufbau der Datenbanken

Die nachstehenden Überlegungen gelten sowohl für die einmalige Massenmigration wie auch für die Bereinigung im Rahmen der täglichen Verarbeitung. Grundsätzlich gibt es diverse Möglichkeiten, in welcher Form und in welchen Applikationen die Stammdaten abgespeichert werden. Als Minimalanforderung muss davon ausgegangen werden können, dass die Applikation unterscheiden kann,

- ob es sich um Stammdaten handelt, die aus einem ESR (oranger Einzahlungsschein mit Referenznummer) oder aus einem roten ES stammen bzw. in Form von nicht beleggebundenen Kundendaten (z.B. bei Salärzahlungen oder LSV-Stammdaten) erfasst wurden, und
- ob die gespeicherten Stammdaten zugunsten eines PostFinance-Kunden (1-stufige Kontoidentifikation, d.h. nur Postkonto-Nummer bzw. ESR-TN) oder eines Bankkunden (2-stufige Kontoidentifikation, d.h. IID oder Postkonto-Nummer der Bank + bankproprietäre Kontonummer des Kunden oder Kontoidentifikation aus ES-/ESR-Codierzeile) lauten.

Diese Unterscheidungsmöglichkeiten müssen sich im Aufbau der Datenbank in irgendeiner Struktur niederschlagen. Zum besseren Verständnis wird nachstehend eine solche Minimalstruktur dargelegt.

9.2.1 Minimalstruktur Begünstigten-Datenbank (bei LSV*: Zahlungspflichtigen)

Variante 1 (mit individuellen Feldern)

Diverse Abspeichervarianten der gleichen Belege dargestellt. In der Praxis kommen in der jeweiligen Applikation nicht alle parallel nebeneinander vor.

Erkennungs-kriterium ES/ESR ¹⁾	PC-Nr. / ESR-TN ²⁾	IID / BIC	Kontonummer Kunde ²⁾	Codierzeile ES/ESR ³⁾	Name/ Adresse
	01	02	03	04	05
ES-Bank	800009393	9230	KK 2.345.123-4		Muster AG
ES-Bank	80-939-3	079230045	KK 2.345.123-4	192532685100000000234512348	Muster AG
ES-Bank		079230045	KK 2.345.123-4	192532685100000000234512348	Muster AG
ES-Bank (ohne Bild)	90-5579-3	81084	KK 2.345.123-4		Muster AG
ES mit IBAN	80-939-3	evtl. aus IBAN extrahiert	CH3808888123456789012		Muster AG
ES mit IBAN		070888854	CH3808888123456789012	000000000000000000234512348	Muster AG
ES mit IBAN	80-939-3	070888854	CH3808888123456789012	000000000000000000234512348	Muster AG
ES mit IBAN	800009393		CH3808888123456789012		Muster AG
ES-Post	25-9034-2				Robert Schneider
ES-Post ²⁾			25-9034-2		Robert Schneider
Stammdaten mit BIC ⁴⁾		SELDCHZZ123	KK 2.345.123-4		Muster AG
ESR-Bank	01244875	230		65432100000260189000100002	Muster AG
ESR-Post	01-162-8			010001628	Robert Schneider
→ Basis für Input-Record in IBAN-Tool	Abfüllen in Feld 03 des Input-Records ⁵⁾	Abfüllen in Feld 03 des Input-Records	Abfüllen in Feld 04 des Input-Records ⁵⁾	Abfüllen in Feld 04 des Input-Records	

¹⁾ Anstelle eines eindeutigen Erkennungskriteriums Bank/Post können die Zahlungen zugunsten Banken bzw. PostFinance und die Unterscheidung ES/ESR auch aufgrund des Inhalts von Feld Postkonto-Nummer/ESR-TN bzw. IID auseinander gehalten werden;

²⁾ Bei ES Post ist es auch denkbar, dass die Postkonto-Nummer des Kunden im Feld Kontonummer abgelegt wird;

³⁾ wird nur bei Verwendung eines Codierzeilenlesers abgefüllt;

⁴⁾ bei Kunden mit FW-Konti wird verschiedentlich auch der BIC des Finanzinstituts anstelle der IID abgespeichert

⁵⁾ Falls Spalte 01 und 02 Daten enthalten, ist Inhalt aus Spalte 02 abzufüllen;
falls Spalte 03 und 04 Daten enthalten, ist Inhalt aus Spalte 04 abzufüllen;
falls Postkonto-Nummer eines PostFinance-Kunden in Spalte 03 statt 01 abgespeichert ist, empfiehlt es sich diese Daten in Feld 03 und nicht 04 des Input-Records abzufüllen, damit die Postkonto-Nummer im IBAN-Format zurückgemeldet werden kann (→ allfällige Konsequenzen für den Aufruf der gespeicherten Stammdaten bei der späteren ZV-Verarbeitung müssen durch den Entwickler individuell analysiert werden).

Die farbig gekennzeichneten Feldinhalte wären aufgrund des Output-Records aus dem IBAN-Tool zu verändern.

Variante 2 (mit Multifunktions-Feldern)

Nicht in Datenbank	IID / PC-Nr. / ESR-TN / BIC ¹⁾	Kontonummer Kunde ²⁾	Codierzeile ES/ESR ³⁾	Name/ Adresse
ES-Bank	9230	KK 2.345.123-4		Muster AG
ES-Bank	80-939-3	KK 2.345.123-4		Muster AG
ES-Bank	079230045	KK 2.345.123-4	19253268510000000234512348	Muster AG
ES-Bank	070023535	0230-575013.01D	00000000000260189000100002	Muster AG
ES mit IBAN	80-939-3	CH3808888123456789012		Muster AG
ES mit IBAN	070888854	CH3808888123456789012		Muster AG
ES-Post	25-9034-2			Robert Schneider
ES-Post ²⁾		25-9034-2		Robert Schneider
Stammdaten + BIC	SELDCHZZ123	KK 2.345.123-4		
ESR-Bank	01244875	230	65432100000260189000100002	Muster AG
ESR-Post	01-162-8		010001628	Robert Schneider

¹⁾ ESR-Zahlungen an ersten 2 Stellen = 01 erkennbar; ES Bank enthält IID/IID ES (erste 2 Stellen = 07, aber keine Postkonto-Nummer der Bank), ES Post enthält Postkonto-Nummer (erste 2 Stellen > 07)

²⁾ Bei ES Post ist es auch denkbar, dass die Postkonto-Nummer des Kunden im Feld Kontonummer abgelegt wird

³⁾ wird nur bei Verwendung eines Codierzeilenlesers abgefüllt

9.2.2 Abgespeicherte Stammdaten nach Bereinigung mit IBAN-Tool

Nachstehend wird nur noch der veränderte Datenstand bei der oben stehenden Variante 1 abgebildet (Annahme, dass die Inputdaten zu keinen Fehlermeldungen (Validierungsflag 10 bis 29) führten).

Erkennungs-kriterium ES/ESR ¹⁾	PC-Nr. / ESR-TN ²⁾	IID / BIC ²⁾	Kontonummer Kunde ²⁾	Codierzeile ES/ESR ³⁾	Name/ Adresse
	01	02	03	04	05
ES-Bank	800009393	9230	CH290923000KK23451234		Muster AG
ES-Bank	80-939-3	079230045	CH290923000KK23451234	19253268510000000234512348	Muster AG
ES-Bank		079230045	CH290923000KK23451234	19253268510000000234512348	Muster AG
ES-Bank (ohne Bild)	80-939-3	9230	CH290923000KK23451234		Muster AG
ES mit IBAN	80-939-3	evtl. aus IBAN extrahiert	CH3808888123456789012		Muster AG
ES mit IBAN		070888854	CH3808888123456789012	00000000000000000234512348	Muster AG
ES mit IBAN	80-939-3	070888854	CH3808888123456789012	00000000000000000234512348	Muster AG
ES mit IBAN	800009393		CH3808888123456789012		Muster AG
ES-Post	250090342		CH0309000000250090342		Max Schneider
ES-Post			25-9034-2		Max Schneider
ES-Post ³⁾	25-9034-2		CH0309000000250090342		
Stammdaten + BIC		SELDCHZZ123	CH290923000KK2345123		Muster AG

Erkennungs-kriterium ES/ESR ¹⁾	PC-Nr. / ESR-TN ²⁾	IID / BIC ²⁾	Kontonummer Kunde ²⁾	Codierzeile ES/ESR ³⁾	Name/ Adresse
			4		
ESR-Bank	01244875	230		6543210000026018900010000 2	Muster AG
ESR-Post	01-162-8			010001628	Max Schneider

¹⁾ Nicht veränderte, proprietäre Kontonummern (Beispiel Postkonto) sind schwarz dargestellt; durch IBAN ersetzte Kontonummern sind **rot** markiert; bereits korrekt abgespeicherte IBANs, die unverändert bleiben können, sind **blau**;

²⁾ Allfällige Anpassungen der IID bzw. Postkonto-Nummer aufgrund Rückmeldung in Output-Record (z.B. nach Fusionen, Zentralisierungen usw.) sind **braun** markiert;

³⁾ Allfällige Verschiebung Postkonto-Nummer von Spalte 03 in Spalte 01 (**grün markiert**), falls IBAN zurückgemeldet wird und Applikation damit keine Mühe bekundet.

Bei Variante 2 oder anderen Datenbankvarianten müssten die migrierten Stammdaten analog abgespeichert werden.

9.3 Bereinigung der Stammdaten der Massenmigration

Einerseits sollen die in den Stammdaten abgespeicherten bankproprietären Kontonummern in IBANs konvertiert werden.

Andererseits können auch die Stammdaten zugunsten von Kunden der PostFinance mit IBANs ergänzt werden.

Nicht bereinigt werden Stammdaten mit einer ESR-Teilnehmernummer (9-stellige ESR-TN mit den Ziffern 01 und 03 (EUR) auf den beiden ersten Positionen).

Einsatz bei den Finanzinstituten

Bei den Finanzinstituten steht die Bereinigung folgender Datenbanken im Vordergrund:

- Datenbank, in welchen die Daueraufträge hinterlegt sind;
- Datenbank, auf welche mit Scanning-Systemen/Beleglesern zugegriffen wird;
- generelle Datenbanken, auf welche im Rahmen der ZV-Abwicklung im Backoffice zugegriffen wird

Einsatz bei den Zahlungspflichtigen

Für die bei den Zahlungspflichtigen im Einsatz befindlichen Software-Applikationen gilt für die Massenmigrationen grundsätzlich – mit folgenden Ausnahmen/Ergänzungen – das gleiche Prozedere wie bei den Finanzinstituten.

Zu migrierende Datenbanken finden sich in den ERP-Softwares, wobei hier auch die Stammdaten von Zahlungsempfängern in den LSV-Applikationen betroffen sind sowie in diversen ZV-Applikationen, insbesondere sog. Offline-Tools in Verbindung mit E-Banking/yellownet.

Spezialfall E-Banking-/yellownet-Datenbanken

Hier geht es um jene Zahlungspflichtigen, welche ihre Zahlungen im E-Banking/yellownet nicht unter Beizug eines Offline-Tools abwickeln, sondern online auf den E-Banking/yellownet-Server des Finanzinstituts zugreifen und ihre Stammdaten demzufolge nicht auf ihrem PC, sondern in den Zahlungsvorlagen auf der Datenbank des Servers abspeichern.

Bei einer allfälligen Bereinigung der im Auftrag der Kunden geführten, zentralen E-Banking-Datenbank müsste allenfalls vorgängig abgeklärt werden, ob diese Bereinigung ohne Einholen einer Zustimmung des betreffenden Kunden gemäss institutsinternen Richtlinien zulässig ist. Dies hängt davon ab, ob die Umwandlung der bisherigen, proprietären Kontonummern als Veränderung von Kundendaten interpretiert wird, die nur vom betreffenden Kunden veranlasst werden darf oder ob die IBAN lediglich als andere Darstellungsform der hinterlegten, proprietären Kontonummer verstanden wird.

Im letzteren Fall würde eine Information des Kunden über die vorgenommene Migration vermutlich genügen.

Sicher ist auf jeden Fall, dass der Einbezug der Zahlungsvorlagen in den E-Banking-/ yellownet-Systemen in die Massenmigration die STP-Qualität markant verbessern würde.

Diese Zielgruppe kann auf der Basis des nachstehend beschriebenen Einsatzes bei den Zahlungspflichtigen nur schwer angesprochen werden, da eine individuelle Migration angesichts der grossen Anzahl an betroffenen Zahlungspflichtigen und der gleichzeitig relativ geringen Anzahl gespeicherten Vorlagen pro individuellem Zahlungspflichtigen verhältnismässig aufwändig ist. In der Summe aller Zahlungsvorlagen wären es immerhin rund 5 Mio. gespeicherte proprietäre Kontonummern, so dass sich eine zentrale Migration durch das jeweilige Finanzinstitut des Zahlungspflichtigen auszahlen würde.

9.3.1 Bereinigung der Stammdaten von Zahlungsempfängern mit Bankkontoverbindungen

Die Stammdaten werden auf bankproprietäre Kontonummern abgesucht und in das IBAN-Tool eingespeist. Sofern das IBAN-Tool eine IBAN aus den abgespeicherten Stammdaten errechnen kann, ist die IBAN anstelle der proprietären Kontonummer in den Stammdaten abzuspeichern. Gleichzeitig sind allenfalls nicht mehr aktuelle IID Postkonto-Nummern der Banken in Stammdaten zu ersetzen (falls Input- und Output-Daten nicht identisch sind).

9.3.2 Bereinigung der Stammdaten von Zahlungsempfängern mit einem Postkonto

Falls Stammdaten mit Postkonto-Nummern von PostFinance-Kunden ins IBAN-Tool eingespeist werden, prüft dieses die Korrektheit aufgrund der Prüfziffer (Modulo 10 rekursiv). Im Output liefert es zusätzlich zur 9-stelligen Postkonto-Nummer auch die jeweilige Darstellung im IBAN-Format.

9.3.3 Generieren von Input-Datenrecords

Bei der Bildung von Input-Datenrecords muss nicht zwischen proprietären Kontonummern und Postkonto-Nummern von PostFinance-Kunden unterschieden werden.

Generieren von Input-Datenrecords durch die Finanzinstitute

Jedes Finanzinstitut kann selbst entscheiden, ob es die Input-Records im ASCII- oder XML-Format generieren und ob es für die Massenmigration Einzel- oder Sammelrecords generieren will.

Bei grösseren Datenmengen (spätestens ab 5000 Records) empfiehlt es sich, die **Einlieferung im ASCII-Format** auf der Basis von **Sammelrecords** vorzusehen.

Begründung

Bei Dateigrößen von über 5000 Transaktionen wird die Inputverarbeitung von XML-Records merklich langsamer als das ASCII-Format. Bei der Verwendung von Einzelrecords für die Massenmigration dürfte sich die Tool-Verarbeitungszeit ebenfalls markant erhöhen.

Generieren von Input-Datenrecords durch die Kunden

Angesichts der kleinen Datenmengen empfiehlt sich hier – anders als bei den Finanzinstituten – eher das zukunftsgerichtete XML-Format. Auch bei den Zahlungspflichtigen sollten im Falle der Massenmigration Sammelrecords generiert werden.

9.3.4 Migrationsprozess

Die Migration erfolgt bei Finanzinstituten und Zahlungspflichtigen in gleicher Weise und zwar unabhängig davon, ob es sich um Stammdaten mit einem Bank- oder Postkonto handelt. Hingegen führt die Interpretation der zurückgemeldeten Validierungsflags zu allenfalls unterschiedlichen Stammdatenanpassungen.

Beim Migrationsprozess wird grob folgendes Vorgehen empfohlen:

- Prüfen anhand des Totalrecords, ob für jeden eingelieferten Record auch ein Output-Datenrecord erstellt wurde;
- Analyse der einzelnen Output-Datenrecords anhand des Validierungsflags (Feld 05):
 - Bei Flag 01 und 02: Ersatz der proprietären Kontonummer durch die IBAN aus Feld 08 und evtl. der IID bzw. Postkonto-Nummer durch die in Feld 06/07 gelieferten Nummern;
 - bei Flag 03 ist die IBAN korrekt und kann deshalb zurückgespeichert werden; evtl. Ersatz der IID bzw. Postkonto-Nummer durch die in Feld 06/07 gelieferten Nummern;
 - bei Flag 04 kann die abgespeicherte Postkonto-Nummer durch die IBAN ersetzt werden, falls die «proprietäre Postkonto-Nummer» des PostFinance-Kunden dadurch nicht verloren geht;

- bei Flag 05 und Flag 09 ist die IBAN in jenem Feld abzuspeichern, welches für die Weiterleitung der Begünstigten-Daten hinzugezogen wird; die Daten aus der ES-Codierzeile sind nicht mehr weiterzuleiten, jedoch als Key-Begriff weiterhin abzuspeichern;
- bei Flag 08 ist die falsche IBAN durch die korrekte IBAN zu ersetzen;
- bei Flag 10 bis 29 (fehlerhafter Input) besteht ein genereller Handlungsbedarf. Dabei stehen folgende zwei Massnahmen im Vordergrund:
 - Löschung der Stammdaten; die Neuerfassung erfolgt mit einem nächsten Zahlungsbeleg (Vorgehen eignet sich nur bei Zahlungsauslösung aufgrund eines Zahlungsbeleges, nicht jedoch bei Stammdaten, die ohne Zahlungsbeleg generiert wurden, wie dies z.B. bei Salär- und Rentenzahlungen oder bei LSV⁺-Stammdaten der Fall war);
 - Kontaktaufnahme mit dem Zahlungsempfänger und anfordern der korrekten IBAN (Kontaktaufnahme evtl. via den Zahlungspflichtigen);

Die vierte Möglichkeit, nämlich abwarten und fehlerhafte Kontonummer in den Stammdaten belassen, ist weniger empfehlenswert, da bei der nächsten Zahlungsauslösung mit einer Zahlungsrückleitung und den entsprechenden Umtrieben gerechnet werden muss.

9.4 Bereinigungsstand der Stammdaten nach durchgeführter Massenmigration

Nach Durchführung der Massenmigration dürften die Stammdaten eine stark verbesserte Datenqualität aufweisen, indem ein Grossteil der proprietären Kontonummern durch IBANs ersetzt ist.

Der Erfolgsfaktor hängt wesentlich davon ab, in welchem Ausmass seinerzeit Fehlerfassungen vorkamen und wie fein die Erkennungslogik der hinterlegten, bankindividuellen Algorithmen ist.

9.4.1 Sofortige Bereinigung neuer Stammdaten im Rahmen der täglichen ZV-Verarbeitung

Leider muss man davon ausgehen, dass bereits am Tag nach durchgeführter Massenmigration neue Zahlungsbelege mit proprietären Kontonummern zur Erfassung anstehen. Dies aufgrund der Tatsache, dass rote ES mit proprietären Kontonummern noch lange im Umlauf sein werden. Die Stammdaten-Qualität wird sich somit – ohne gezielte Massnahmen – schrittweise wieder verschlechtern.

Diesem Problem kann auf zwei Arten begegnet werden:

- Durchführung einer Massenmigration in periodischen Abständen. In diesen Fällen wäre es sinnvoll, bei der Aufbereitung der zu migrierenden Stammdaten mit einer IBAN nicht ins IBAN-Tool einzuschleusen, da sonst mit der Zeit unnötig viele korrekte Stammdaten analysiert werden müssten, was sich in einer entsprechend höheren Verarbeitungszeit auswirken würde.
- Analyse der beim Zahlungsausgang zu erfassenden Stammdaten während der täglichen ZV-Verarbeitung.

9.5 Einsatzmöglichkeit des IBAN-Tools in der täglichen ZV-Verarbeitung

Die Einsatzmöglichkeit des IBAN-Tools in der täglichen Verarbeitung unterscheidet sich stark von jener bei der Massenmigration.

Bei der Anpassung der ZV-Applikationen ist dabei insbesondere die Verarbeitung von Einzahlungsscheinen wichtig. Aber auch bei der Erfassung von Zahlungsempfänger-Stammdaten ohne Vorlage eines Einzahlungsscheines kann das IBAN-Tool sinnvoll eingesetzt werden.

Bei der Erfassung von Einzahlungsscheinen gilt es zu berücksichtigen, dass zukünftig – nebst den orangen ESR – drei rote Einzahlungsschein-Varianten im Umlauf sein und die meisten Zahlungspflichtigen mit deren Unterscheidung Mühe bekunden werden. Die Erfahrung im Kundenkontakt zeigt, dass viele den Unterschied zwischen einem roten ES und einem orangen ESR, geschweige denn zwischen einem ES Post und einem ES Bank nicht kennen. Sie führen einfach aus, was ihnen die ZV-Applikationen anzeigen.

ZV- und E-Banking-/yellownet-Applikationen müssen deshalb bei der Generierung der Zahlungen zielgerichtet durch die Erfassungsmasken führen.

Einsatz bei den Finanzinstituten

Der in den folgenden Abschnitten noch näher beschriebene Mehrwert des IBAN-Tools könnte bei den Finanzinstituten sowohl im Backoffice wie auch bei den Hochleistungs-Scanning-Systemen eingebaut werden, um den Nachbearbeitungsaufwand bei der Verarbeitung der verschiedenen ES im ZV-Ausgang zu reduzieren.

Daneben könnte das IBAN-Tool auch durch das Dauerauftrags-System in jenen Fällen beigezogen werden, in denen durch den Kunden noch ein Dauerauftrag mit einer proprietären Kontonummer eingereicht wurde.

Einsatz bei den Zahlungspflichtigen

Der Einsatz des IBAN-Tools im Rahmen der täglichen Verarbeitung beim Zahlungspflichtigen eignet sich insbesondere in Zusammenhang mit der Erfassung bzw. des Scannings der Codierzeile der verschiedenen ES.

Selbstverständlich kann das IBAN-Tool zusätzlich auch bei der Erfassung von Salärzahlungen oder anderweitigen Zahlungen (z.B. Auszahlung von Versicherungs- oder Krankenkassenleistungen), eingesetzt werden, d.h. in jenen Fällen, wo kein Einzahlungsschein vorliegt und nur die proprietäre Kontonummer bekannt ist.

Spezialfall E-Banking/yellownet-Datenbanken

Das IBAN-Tool ist auch im E-Banking optimal einsetzbar. Hier könnte die gleiche Konversions-Logik wie bei der Neuerfassung von ES auch für wiederkehrende Zahlungen verwendet werden, wenn der Zahlungspflichtige eine seiner alten Zahlungsvorlagen aufruft, welche noch eine proprietäre Kontonummer enthält und diese für eine neue Zahlungsauslösung verwendet.

9.5.1 Verarbeitung von orangen Einzahlungsscheinen (ESR)

ESR enthalten keine proprietären Kontonummern und sind deshalb nicht durch das IBAN-Tool zu validieren. Diese Records würden als nicht zugelassen durch das IBAN-Tool zurückgewiesen. Ein entsprechender Input würde somit lediglich die Verarbeitungszeit unnötig erhöhen.

9.5.2 Verarbeitung des roten ES der PostFinance

Wie erwähnt, kann die Post-Finance auch die IBAN des PostFinance-Kunden im ZV-Eingang verarbeiten. Es ist somit grundsätzlich möglich, die im Output-Datenrecord mitgelieferte Postkonto-Nummer im IBAN-Format abzuspeichern und weiterzuleiten.

9.5.3 Verarbeitung des roten ES Bank mit proprietären Konto-Nr. und roter ES Bank mit IBAN

Im Gegensatz zum ES Post handelt es sich beim ES Bank – mit dem Belegartencode 303 – um einen zweistufigen Beleg. Bisher haben die Banken diesen ES Bank mit proprietärer Kontonummer abgegeben. Zweistufig deshalb, weil nebst der Postkonto-Nummer der jeweiligen Banken und deren Bankadresse auf dem Beleg auch die proprietäre Kontonummer des Bankkunden sowie dessen Name/Adresse und – in der Codierzeile – die IID der Bank sowie eine 27-stellige ES-Referenznummer angedruckt wird, in welcher die Kontonummer des Bankkunden (in den Positionen 11-26) enthalten ist.

Gerade bei der korrekten Interpretation der verschiedenen ES-Varianten kann das IBAN-Tool auch in der täglichen ZV-Verarbeitung unterstützend eingesetzt werden.

Prinzipiell besteht sowohl beim Zahlungspflichtigen wie auch bei den Finanzinstituten die Möglichkeit, einen ES aufgrund der Zahlungsdaten auf dem oberen Teil des Beleges oder unter Scanning/Beleglesung der Codierzeile zu erfassen. In beiden Fällen sollte die ZV-Applikation so viele Erfassungshilfen und Kontrollhilfen wie mit vertretbarem Aufwand möglich anbieten. Hier kann das IBAN-Tool einen Mehrwert bieten.

Erfassung ab oberem Teil des roten ES Bank

Beim roten **ES Bank mit proprietärer Kontonummer** fehlt die IBAN. Es wird somit wie bisher die proprietäre Kontonummer sowie die IID erfasst. In diesen Fällen kann das IBAN-Tool beigezogen werden, indem Kontonummer und IID während der Erfassung ins IBAN-Tool eingelesen und die erhaltene IBAN anschliessend in den Stammdaten abgespeichert sowie für die Generierung des Zahlungsrecords verwendet wird. In den meisten Fällen dürften Fehlerfassungen durch das IBAN-Tool erkannt und zurückgemeldet werden. Durch eine geschickte Benutzerführung kann auf die Fehlerfassung hingewiesen und eine Eingabekorrektur veranlasst werden.

Bei der Erfassung eines roten ES Bank mit IBAN ist zwar die IBAN auf dem Beleg abgebildet. Dafür fehlt die bisher ebenfalls angedruckte IID. Für diesen Fall muss die Erfassungslogik der ZV-Applikation angepasst werden.

Das IBAN-Tool bietet in beiden Fällen einen zusätzlichen Mehrwert, indem es nicht nur die IBAN, sondern auch die IID sowie die dazugehörige Postkonto-Nummer der Bank zurückmeldet. Die Erfassung dieser beiden Datenfelder, die bei vielen ZV-Applikationen ebenfalls in den Stammdaten abgelegt und deshalb bisher manuell erfasst werden mussten, lässt sich bei Verwendung des IBAN-Tools in der täglichen Verarbeitung zukünftig vermeiden.

Scanning/Beleglesung der Codierzeile

Beim Scanning der Codierzeile wird die IID ES und die ES-Referenz als Key-Begriff in die Stammdaten abgelegt. Bei der Ersterfassung eines roten Einzahlungsscheins mit einem Belegleser muss in der Regel zusätzlich – nebst der Begünstigten-Adresse, dem Betrag und allfälligen Mitteilungen – auch noch die auf dem Einzahlungsschein befindliche Kontonummer (bisher proprietäre Kontonummer, neu IBAN) sowie vielfach auch die Postkonto-Nummer der Bank erfasst werden.

Hier stellt sich eine analoge Problematik wie bei der manuellen Erfassung ab dem oberen Teil des Einzahlungsscheines sowie ein vergleichbarer Mehrwert.

Sowohl beim roten ES Bank mit proprietärer Kontonummer wie auch beim roten ES Bank mit IBAN kann das IBAN-Tool aufgrund der ES-Referenz die IBAN anzeigen, in die Stammdaten ablegen und für die Generierung des ZV-Records verwenden. Ausserdem können die IID und die Postkonto-Nummer ohne manuelle Erfassung in die Stammdaten abgelegt werden.

Im Falle eines roten ES Bank mit proprietärer Kontonummer wäre allenfalls ein Popup-Fenster mit dem Hinweis angebracht, dass die IBAN errechnet wurde, da auf diesem ES noch die proprietäre Kontonummer ersichtlich ist und der Zahlungspflichtige evtl. verunsichert sein könnte.

9.5.4 Erfassung von nicht beleggebundenen Stammdaten

Damit proprietäre Kontonummern in den Stammdaten schon bei der Ersterfassung durch IBANs ersetzt werden können, sollte das IBAN-Tool auch in jenen Applikationen (z.B. Salär-Applikationen oder Kreditoren-Applikationen von Versicherungen und Krankenkassen), bei welchen Auszahlungen in grösserem Umfang ohne Zahlungsbeleg generiert werden müssen, in der täglichen Verarbeitung eingesetzt werden.

Das gleiche gilt auch im Lastschriftverfahren, wo anstelle der IBAN weiterhin die proprietäre Kontonummer auf der Belastungsermächtigung zurückgemeldet wird.

9.5.5 Generieren der Input-Datenrecords fürs IBAN-Tool

Wie bei der Massenmigration steht auch hier dem Einlieferer frei, in welchem Format er die Input-Datenrecords mit welcher Recordart aufbereiten will.

Da bei der täglichen Verarbeitung im Gegensatz zur Massenmigration jeweils jede Transaktion einzeln ins IBAN-Tool eingegeben wird, drängt sich hier die Abwicklung auf der Basis eines Einzelrecords auf. Als Inputformat wird XML empfohlen.

Diese Empfehlung gilt sowohl für Finanzinstitute der Zahlungspflichtigen wie auch für die von den Software-Firmen entwickelte Schnittstellen-Software für die Zahlungspflichtigen.

9.5.6 Migration im täglichen Einsatz

Hier ist ein analoges Validierungsverfahren wie unter Ziffer **Fehler! Verweisquelle konnte nicht gefunden werden.** beschrieben anzuwenden.

Da es sich um Erstdaten handelt, müssen allerdings keine bestehenden Stammdaten mutiert werden. Die aus dem IBAN-Tool zurückgemeldeten Daten sind erstmalig abzuspeichern.

Gleichzeitig dient das IBAN-Tool als Erfassungshilfe, indem ein beträchtlicher Teil allfälliger Fehlerfassungen in Form einer Fehlermeldung online zurückgemeldet und eine sofortige Korrektur der erfassten Daten vor dem Abspeichern verlangt werden kann.

9.6 Künftiges Generieren der Zahlungsrecords im ZV-Ausgang

Ziel der ganzen Bereinigungsaktion ist, künftig möglichst viele STP-Transaktionen zu generieren. Zu diesem Zweck müssen die Zahlungsdaten korrekt in die richtigen Meldungstypen abgefüllt werden.

Die nachfolgenden Ausführungen gelten sowohl für das Generieren von Zahlungsdaten nach erfolgter Massenmigration wie auch für die tägliche ZV-Verarbeitung.

9.6.1 Generieren von Zahlungsrecords durch die Zahlungspflichtigen

Bei der Bildung der Zahlungsrecords ist zwischen Zahlungsrecords zugunsten von Bankkunden und Zahlungsrecords zugunsten von PostFinance-Kunden zu unterscheiden.

Generieren von Zahlungsrecords zugunsten Bankkunden

Hier ist zwischen Records im DTA-Format (generiert von Zahlungspflichtigen mit Bankverbindung) und Records im EZAG-Format (generiert von Zahlungspflichtigen mit Postkonti) zu unterscheiden:

- **Beim DTA-Format ist in Kombination mit der IBAN – anstelle der proprietären Kontonummer oder der 27-stelligen ES-Referenznummer aus der Codierzeile des roten ES Bank – nicht mehr der TA827, sondern der TA836 zu generieren.**
- **Beim EZAG-Format ist darauf zu achten, dass bei Zahlungen zugunsten von Bankkunden generell die Transaktionsart TA27 (nicht wie bisher oft fälschlicherweise ein TA22) generiert wird und die durch das IBAN-Tool zurückgemeldete IID in das Feld «Clearing-Nummer» und die IBAN in das Feld «Kontonummer Endbegünstigter» abgefüllt wird.**
- **Bei LSV-Records ist die IBAN des Zahlungspflichtigen im TA875, im Feld «KTO-ZP», abzufüllen.**

Bildung von Zahlungsrecords zugunsten PostFinance-Kunden

Zahlungen zugunsten von Postkunden können grundsätzlich in unveränderter Form mit 9-stelliger Postkonto-Nummer generiert werden. Es ist jedoch auch zulässig, anstelle der 9-stelligen Postkonto-Nummer, die aus dem IBAN-Tool gemeldete IBAN der PostFinance-Kunden im Zahlungsrecord abzufüllen. In diesem Fall ist wie folgt vorzugehen:

- **Beim DTA-Format kann wie bisher ein TA827 mit der 9-stelligen Postkonto-Nummer oder ein TA836 mit der IBAN des PostFinance-Kunden generiert werden.**
- **beim EZAG-Format ist – bei Zahlungen zugunsten von PostFinance-Kunden mit Angabe der 9-stelligen Postkonto-Nummer – wie bisher TA22 zu verwenden. Wird eine IBAN anstelle der 9-stelligen Postkonto-Nummer angegeben, ist hingegen die Transaktionsart TA27 zu nutzen.**

9.6.2 Generieren des ZV-Ausgangs im SIC

Bisher wurden in den meisten Fällen die proprietären Kontonummern oder bei roten ES Bank die ES-Codierzeile (bzw. die 16 Stellen ab Position 11 daraus) in der Ausprägung 45A/B in die beiden Meldungstypen abgefüllt. Neu ist zu beachten, dass überall dort, wo im Rahmen der Massenmigration eine IBAN abgespeichert werden konnte, diese in die Feldausprägung 45I abgefüllt wird.

9.6.3 Generieren der Zahlungsrecords im EGA-V

PostFinance muss dabei darauf achten, dass sie die IBAN – sofern in den Input-Datenrecords angeliefert bzw. in den Dauerauftrags-Stammdaten hinterlegt – und nicht mehr die proprietären Kontonummern/ES-Codierzeilen in das Feld Kontonummer des Begünstigten abfüllt.



10 Feedback und Fragen

Allfälliger Feedback oder Fragen in Zusammenhang mit dem Einsatz des IBAN-Tools sind an folgende Adresse zu richten:

SIX Interbank Clearing AG
Technical Support
Hardturmstrasse 201
8021 Zürich

Tel: +41 58 399 4420

E-Mail: contact.sic@six-group.com