



OFCKS.NET

Schnittstellendokumentation

Alle Rechte vorbehalten, auch die der photomechanischen Wiedergabe und der Speicherung in elektronischen Medien.

Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Weder die Firma syska GmbH noch die Autoren können für fehlerhafte Angaben und deren Folgen eine juristische Verantwortung oder sonstige Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind wir dankbar.

© Copyright syska GmbH, Karlsruhe. Alle Rechte vorbehalten.

5201710

Finanzen im grünen Bereich The logo for syska. The word "syska" is written in a lowercase, serif font. The letters "s", "y", and "k" are in a dark green color, while the letters "s", "y", and "a" are in black. A small green square is positioned at the end of the word, after the period.

syska Gesellschaft für betriebliche
Datenverarbeitung mbH
Am Sandfeld 15
D 76149 Karlsruhe
Telefon 0721 / 9 85 93 0
Telefax 0721 / 9 85 93 60
E-Mail info@syska.de
Homepage www.syska.de

Inhalt

OFCKS.NET Schnittstellenübersicht	1
Übersicht zur Schnittstelle OFCKS.NET	1
Verwendungsmöglichkeiten	1
Aufbau der Schnittstelle	3
Objekte.....	3
Objektgenerationen	3
Grundfunktionalitäten	3
Datentypen	4
Die wichtigsten Objekte	4
Namenskonventionen	5
Umgebung	5
Installation	5
Anmeldung.....	5
Protokoll	6
Fibu-Rechte	6
Fehler.....	6
Referenz der Objekte	7
Application	7
Eigenschaften	7
Methoden	8
MandantCollection	8
Eigenschaften	9
Methoden	9
Mandant	9
Eigenschaften	9
Methoden	11
KontoCollection	13
Eigenschaften	13
Methoden	14
Konto	14
Eigenschaften	15
Methoden	18
BankCollection	19
Eigenschaften	19
Methoden	19
Bank.....	19
Eigenschaften	19
Methoden	20
OPSepaMandatCollection	21
Eigenschaften	21
Methoden	21
OPSepaMandat	21
Eigenschaften	21
Methoden	21

Modelle der Zusammenarbeit zwischen ERP-System und Fibu	21
StapelCollection	22
Eigenschaften	22
Methoden	22
Stapel	22
Eigenschaften	23
Methoden	23
StapelBuchungCollection	23
Eigenschaften	23
Methoden	24
Stapelbuchung	24
Eigenschaften	24
Methoden	26
AnlagegutCollection	30
Eigenschaften	31
Methoden	31
Anlagegut	31
Eigenschaften	31
Methoden	31
AnzahlungCollection	31
Eigenschaften	32
Methoden	32
Anzahlung	32
Eigenschaften	32
AufbauZeileCollection	32
Eigenschaften	32
Methoden	32
AufbauZeile	32
Eigenschaften	33
Methoden	33
AuftragCollection	33
Eigenschaften	33
Methoden	33
Auftrag	33
Eigenschaften	34
Methoden	34
AuswertungskreisCollection	34
Eigenschaften	34
Methoden	34
Auswertungskreis	35
Eigenschaften	35
Methoden	35
AZDebitorCollection	35
Eigenschaften	35
Methoden	35
AZDebitor	35
Eigenschaften	35
Methoden	36
BelegzeileCollection	36
Eigenschaften	36
Methoden	36
Belegzeile	37
Eigenschaften	37
Methoden	38
BestellungCollection	38
Eigenschaften	38
Methoden	38

Bestellung	38
Eigenschaften	38
Methoden	39
BuchungskreisCollection	39
Eigenschaften	39
Methoden	39
Buchungskreis	39
Eigenschaften	39
Methoden	40
CifPlanzahlenCollection	40
Eigenschaften	40
Methoden	40
CifPlanzahlen	40
Eigenschaften	40
Methoden	41
FibuAufbauZeileCollection	41
Eigenschaften	41
Methoden	41
FibuAufbauZeile	41
Eigenschaften	41
Methoden	42
FremdWaehrungCollection	42
Eigenschaften	42
Methoden	42
FremdWaehrung	42
Eigenschaften	42
Methoden	42
GruppierungCollection	42
Eigenschaften	43
Methoden	43
Gruppierung	43
Eigenschaften	43
Methoden	43
KontoListeCollection	43
Eigenschaften	43
Methoden	43
KontoListe	44
Eigenschaften	44
Methoden	44
KoreBelegCollection	44
Eigenschaften	44
Methoden	44
KoreBeleg	45
Eigenschaften	45
Methoden	45
KoreBuchungCollection	45
Eigenschaften	46
Methoden	46
KoreBuchung	46
Eigenschaften	46
Methoden	46
KorePlanSchabloneCollection	47
Eigenschaften	47
Methoden	47
KorePlanSchablone	47
Eigenschaften	47
Methoden	47

KorePlanwerteCollection	47
Eigenschaften	47
Methoden	48
KorePlanwerte	48
Eigenschaften	48
Methoden	48
KorePrognosewerteCollection	48
Eigenschaften	49
Methoden	49
KorePrognosewerte	49
Eigenschaften	49
Methoden	49
KoSchabloneCollection	50
Eigenschaften	50
Methoden	50
KoSchablone.....	50
Eigenschaften	50
Methoden	50
KostenartCollection	50
Eigenschaften	50
Methoden	51
Kostenart	51
Eigenschaften	51
Methoden	51
KostenstelleCollection	51
Eigenschaften	51
Methoden	51
Kostenstelle.....	52
Eigenschaften	52
Methoden	52
KostentraegerCollection	52
Eigenschaften	52
Methoden	53
Kostentraeger.....	53
Eigenschaften	53
Methoden	53
KursCollection.....	53
Eigenschaften	54
Methoden	54
Kurs	54
Eigenschaften	54
Methoden	54
OPDebVorgangCollection	54
Eigenschaften	54
Methoden	55
OPDebVorgang	55
Eigenschaften	55
Methoden	55
OPKredVorgangCollection	56
Eigenschaften	56
Methoden	56
OPKredVorgang.....	56
Eigenschaften	56
Methoden	57
OPostenDebCollection	57
Eigenschaften	57
Methoden	57

OPostenDeb	58
Eigenschaften	58
Methoden	59
OPostenKredCollection	59
Eigenschaften	59
Methoden	59
OPostenKred.....	60
Eigenschaften	60
Methoden	60
RechnungCollection	61
Eigenschaften	61
Methoden	61
Rechnung	61
Eigenschaften	61
Methoden	62
SpracheCollection.....	63
Eigenschaften	63
Methoden	63
Sprache	63
Eigenschaften	63
Methoden	63
SteuerCollection.....	63
Eigenschaften	63
Methoden	63
Steuer	64
Eigenschaften	64
Methoden	64
SteuerSatzCollection.....	64
Eigenschaften	64
Methoden	64
SteuerSatz	64
Eigenschaften	65
Methoden	65
ZahlzielCollection	65
Eigenschaften	65
Methoden	65
Zahlziel.....	65
Eigenschaften	66
Methoden	66
Testprogramm OFCKSNET.Test	67
Bedienung des Testprogramms	67
Die XML-Dateien	68
Hinweise zu den Dateien	68
Entwickler-Demoprojekt	71
Visual Studio 2010 Projekt in C#	71
Projektaufbau	71
Bestandteile	71
Index	73

OFCKS.NET Schnittstellenübersicht

Übersicht zur Schnittstelle OFCKS.NET

Das **O**pen **F**inance **C**onnectivity **K**it OFCK ist die Schnittstelle zum Programm Syska ProFI (auch unter anderen Namen bei OEM-Herstellern bekannt) und dessen Daten. Integratoren können damit Daten aus der Buchhaltung abfragen (z.B. Sachkonten, Steuerschlüssel, Zahlungsbedingungen), Stammdaten mit denen in der Finanzbuchhaltung abgleichen (z.B. Kunden – Debitoren, Lieferanten – Kreditoren) oder Daten an die Buchhaltung übergeben (vor allem Buchungen).

Ein Direktzugriff auf die Daten wird nicht empfohlen, da sich die Datenstrukturen jederzeit ändern können. Die Zugriffsfunktionen über OFCKS.NET bleiben dagegen auch bei Änderungen erhalten. Neue Funktionalitäten werden in neuen Generationen der jeweiligen Schnittstellen zur Verfügung gestellt.

Integratoren, die eine Schnittstelle zur bis 2013 gepflegten bisherigen Version des OFCK (COM-Variante) erstellt hatten, müssen OFCKS.NET einmalig neu einbinden. Sie können jedoch ihre bisherige Logik weitestgehend weiter verwenden, da sich OFCKS.NET an der Zugriffsmethodik des OFCK (COM-Variante) orientiert.

Verwendungsmöglichkeiten

OFCKS.NET (OFCKSNET.dll) ist eine Assembly für das .NET Framework 4.5 und spätere (bis zur Version 5.0.7.2 .NET Framework 4.0), realisiert mit C#. Die Datenzugriffe geschehen über ADO.NET auf dieselbe Weise wie auch in den Hauptprogrammen des Syska ProFI. Die Assembly kann auf zweierlei Wegen in Ihr Programm eingebunden werden:

- Direkt als .NET Assembly aus einem .NET Programm.
- Als COM-Server (VTable oder Dispatch-Interface) aus anderen Programmen, die COM-Server verwenden können. Die COM-Schnittstellen werden allerdings nicht weiter entwickelt.

Binden Sie OFCKS.NET über eine Referenz ein. Der primäre Namespace ist OFCKNet. COM-Schnittstellenmethoden, die sich von der .NET Schnittstelle unterscheiden müssen (vor allem wegen der Kompatibilität der Datentypen), tragen ein nachfolgendes "Com" im Namen.

Aufbau der Schnittstelle

Objekte

OFCKS.NET bietet den Zugriff auf die Objekte des Rechnungswesens als Collections an. So greifen Sie z.B. über die KontoCollection auf die Konten eines Mandanten zu (ggf. weiter gefiltert). Über die KontoCollection rufen Sie ein einzelnes Konto ab.

Die Daten von syska ProFI sind (bis auf wenige Ausnahmen) in Mandanten (Firmen) organisiert. So wie der Anwender in der Oberfläche nach der Anmeldung zunächst die Auswahl des Mandanten trifft, muss das auch Ihre Anwendung tun. In der Regel wird in Ihren Daten hinterlegt sein, welcher Fibu-Mandant Ihren Daten zugeordnet ist.

Vom ausgewählten Mandant aus gelangt man zu den anderen Collections, die dann schon nach Mandant eingegrenzt sind.

Objektgenerationen

Das bisherige OFCK hat bei Erweiterungen jeweils eine neue Schnittstellen-Generation angelegt. Die vorigen COM-Interfaces sind so jeweils unverändert erhalten geblieben. Die Zählung wird im COM-Interface fortgeführt, OFCKS.NET bietet jedoch jeweils die neueste Generation als Startnummer an. Beispiel: Die Schnittstelle IKonto startet in OFCKS.NET mit der Generation IKonto10.

Eine künftige Erweiterung des Kontos im OFCKS.NET wird dann die Schnittstelle IKonto11, später IKonto12 anlegen, die vorigen Generationen bleiben erhalten.

Im .NET Interface ist immer die Schnittstelle ohne angehängte Generationsnummer die aktuelle Schnittstelle. Das .NET Framework fordert als typstrenge Umgebung viele Typecasts. Diese Typecasts können daher über die Zeit erhalten bleiben. Neu hinzu gekommene Eigenschaften oder Parameter in Methoden führen nicht zu Fehlern. Im Einzelfall kann es in der späteren Weiterentwicklung zu Umbenennungen oder ähnlichem kommen, die dann dokumentiert werden.

Grundfunktionalitäten

Die **Collections** bieten jeweils die Grundfunktionen:

1. CreateNewItem: Legt ein neues leeres Objekt an. Dieses füllt die verwendende Anwendung dann und speichert es mit Insert ab.
2. Count: Anzahl der Objekte.
3. This[Index] bzw Item[Index]: Der Zugriff auf ein bestimmtes Objekt über den dazugehörigen Schlüsselbegriff. Die Syntax ist je nach Programmierumgebung unterschiedlich, z.B. Konten[1200] oder Konten.Item(1200) für den Zugriff auf Konto 1200.

4. Foreach-Schleifen werden unterstützt. Dazu im Folgenden differenzierte Hinweise.
5. Application (nur .NET): Rückgriff auf das grundlegende Application-Interface.
6. Mandant (nur .NET): Rückgriff auf den Mandant.
7. Type (nur .NET): Gibt den Objekttyp anhand der Enumeration OFCKObjectType zurück.
8. Filter: Filter sind Einschränkungskriterien. Sie müssen vor dem ersten Zugriff auf die Collection-Daten gesetzt werden. Mehrere Filter können in der Regel kombiniert werden. Gibt es eine Funktion RefreshCollData, kann der Filter geändert und die Datenabfrage neu ausgelöst werden.

Weitere Methoden können individuell hinzukommen.

In einer foreach-Schleife in einem .NET Programm ist zu beachten: Die Collection ist ein KeyValuePair, z.B. KeyValuePair<System.String, IKostenstelle>. Die verwendende Anwendung will letztlich auf den .Value zugreifen.

In einer For each Schleife in VBA (Visual Basic for Applications) wird direkt das eigentliche Objekt erreicht. Dies gilt ebenso für andere Programmiersprachen, die dieselbe Konstruktion unterstützen, die auf IEnumVariant beruht.

Die **Einzelobjekte** bieten die folgenden Grundfunktionen (nicht immer sind alle Grundfunktionen verfügbar):

1. Insert: Speichert ein neues Objekt ab.
2. Update: Speichert ein geändertes vorhandenes Objekt ab.
3. Delete: löscht ein vorhandenes Objekt.

Dazu kommen die Eigenschaften und die individuellen Funktionen.

Datentypen

In der .NET Schnittstelle werden CLR-kompatible Datentypen verwendet. Bei Eigenschaften, für die eine Unterscheidung zwischen 0 und NULL (falls numerisch) oder zwischen true / false und NULL Zusatzerkenntnisse liefern kann, werden auch Nullable<Typ> Datentypen verwendet.

In der COM-Schnittstelle werden COM-kompatible Datentypen verwendet. In numerischen oder booleschen Eigenschaften kann der Fall NULL nicht unterschieden werden.

Die wichtigsten Objekte

Bei allen Zugriffen muss zunächst ein **Application**-Objekt angelegt werden und danach ein **Mandant** angesprochen werden. Üblicherweise ist in Ihrer Software hinterlegt, auf welchen Fibu-Mandant Sie zugreifen, oder dies ergibt sich aus Ihrer Firmennummer.

Die wichtigsten Schnittstellen sind üblicherweise:

Stammdatenabgleich der **Personenkonten**. Hier ist zu beachten, dass die **KontoCollection** (Mandant.Konten) alle Kontoarten umfasst: Sachkonten, Debitoren und Kreditoren. Setzen Sie beim Zugriff auf alle Konten den FilterKontoArt, oder greifen Sie gezielt auf eine Kontonummer zu.

Zu den Personenkonten-Stammdaten gehören untergeordnet noch die Bankverbindungen, Konto.Banken. Es handelt sich um die **BankCollection** mit dem Einzelobjekt **Bank**.

Außerdem übergeben die meisten Anwendungen **Buchungen** an die Fibu, z.B. für Ausgangsrechnungen oder Kassenbuchungen. Diese müssen Sie in einen Buchungsstapel

einfügen, den Sie aus einer **StapelBuchungCollection** auswählen oder direkt per Nummer ansprechen. Darin fügen Sie Objekte vom Typ **Stapelbuchung** ein. Einem Buchungsbeleg hängen Sie mit den Hauptfunktionen **AddSoll** und **AddHaben** Buchungszeilen an.

Einzelheiten zu den Objekten und eine Auflistung aller Objekte entnehmen Sie den Referanzinformationen.

Namenskonventionen

In der Schnittstelle werden die folgenden Namenskonventionen verwendet:

- **Collections** enthalten die Objektbezeichnung gefolgt vom Wort **Collection** gefolgt von der Generation. Beispiel: **KontoCollection04**. Diese Konvention wurde im früheren OFCK nicht konsequent angewendet, dadurch haben sich Namen geändert. Die Generation setzt i.A. auf der letzten Generation des früheren OFCK auf.
- **Einzelobjekte** enthalten die Objektbezeichnung gefolgt von der Generation, z.B. **Konto10**.
- **Eigenschaften**, für die in der COM-Schnittstelle ein von der .NET Schnittstelle abweichender Datentyp verwendet wird (da der .NET Datentyp nicht COM-kompatibel ist), tragen die Endung **Com**. Häufig liegt der Unterschied darin, dass die .NET Datentyp ein `Nullable<t>` ist und somit explizit den Null-Wert zulässt.

Umgebung

Installation

Für die Installation steht ein MSI-Setup zur Verfügung.

Die typische Installation installiert die OFCKS.NET-Schnittstelle ohne darauf aufsetzende Schnittstellen. Im Unterverzeichnis *Doku* wird die Dokumentation abgelegt.

Die Setup-Komponente *Richtlinie* (ab 5.0.7.1) wird im Standard mitinstalliert. Sie legt eine sogenannte *Herausgeberrichtlinie* (Policy DLL) in den Global Assembly Cache GAC. Diese sorgt dafür, dass auch ältere verwendende Software mit der neuesten Version des OFCKS.NET funktioniert (sofern weiterhin Schnittstellen-Kompatibilität gegeben ist, worauf nach Möglichkeit immer geachtet wird). Sollen auf einem Rechner mehrere Versionen des OFCKS.NET parallel genutzt werden, muss die *Richtlinie* u.U. deinstalliert werden.

Anmeldung

Zur Anmeldung müssen eine Application erzeugt werden und die folgenden Eigenschaften gesetzt werden. Die Grundangaben zur Datenbankverbindung werden der Konfigurationsdatei `sql.config` entnommen, wie im Handbuch *DB_Installation* beschrieben. Dieselben Angaben werden im Anmeldebildschirm der Dialogprogramme angegeben.

- **DBDatabase**: Name der Datenbank laut `<database>` Tag der `sql.config`
- **DBUser**: Datenbank-Benutzername
- **DBPassword**: Datenbank-Passwort

Anschließend kann die `DBConnect`-Funktion der Application aufgerufen werden.

Alternativ kann ab Version 5.0.5.0 auch auf die Datei `sql.config` verzichtet werden und stattdessen die notwendigen Konfigurationszeilen (einzeln) mit der Funktion `SetSqlConfigLine` übergeben werden (siehe dieser Methode).

Protokoll

OFCKS.NET schreibt ein Protokoll, sofern im Dialogprogramm DB_Inst unter Installation -> Einstellungen & Werte der folgende Eintrag gesetzt wird (in exakt dieser Schreibweise):

OFCK-Fehlerprotokoll

Als Wert muss der Name der Protokolldatei mit Pfad angegeben werden. Es muss sich um einen Pfad handeln, auf den alle OFCK-Nutzer schreibenden Zugriff haben (wichtig bei einer Server-Installation).

Falls die Angaben unrichtig sind oder kein Schreibrecht besteht, funktioniert OFCKS.NET trotzdem und protokolliert nicht.

Fehler werden auf jeden Fall ins Protokoll geschrieben. Mit dem Eintrag

OFCK-Loglevel

mit den Werten 1 und 2 (oder Trace und Verbose) enthält das Protokoll mehr Inhalte je nach Funktion.

Fibu-Rechte

Die in Syska ProFI vergebenen Fibu-Rechte werden nicht abgeprüft. Eine Rechteverwaltung muss vom nutzenden Programm implementiert werden falls erforderlich. Andernfalls wäre es sonst nötig Benutzer in der Fibu anzulegen welche eigentlich keinen Fibuzugang brauchen würden. Damit wird verhindert, dass ein User automatisch Zugang zu der Fibu bekommt, falls dieser nicht für die Buchhaltung zuständig ist.

Hinweis systemseitige Rechte:

Wenn ein Anwender mit systemseitigen Rechten für OFCK ausgestattet werden soll, ohne in den Online-Programmen Zugriffsrechte zu bekommen, können Sie das ebenfalls über die db_inst.exe verwalten. Führen Sie in der db_inst.exe das Skript „-5menu.wts“ aus. Danach können Tabellenrechte für das Modul OFCK (-5) in der db_inst.exe vergeben werden.

Fehler

Fehler des OFCK sind in der .NET Umgebung von OFCKException abgeleitet. In der COM-Schnittstelle haben sie ein HRESULT beginnen mit 0x8004A.

Referenz der Objekte

Die Referenz der Objekte beginnt mit den notwendigen Objekten für die beiden wichtigsten Kopplungen aus Vorsystemen: den Abgleich von Personenkonto und der Übergabe von Buchungen. Danach folgen die weiteren Objekte in alphabetischer Reihenfolge.

Application

Repräsentiert eine Datenbankverbindung als Basis der Anmeldung.

Dieses Objekt muss immer (neu) angelegt werden und eine Connection zu einer Datenbank hergestellt werden. Alle weiteren Objekte lässt man sich jeweils von einem übergeordneten Objekt geben.

Die Konfiguration der DB-Verbindung wird aus der sql.config Datei gelesen, wie es für die Dialogprogramme des Syska ProFI auch gilt. Diese muss im Aufrufverzeichnis der Anwendung vorhanden sein, das die OFCK-DLL einbindet.

Aktuelle Generation: Application01.

Eigenschaften

- **DBDatabase:** Name der Datenbankverbindung laut dem Eintrag <database name="..."> der sql.config.
- **DBUser:** Username mit dem die Verbindung hergestellt werden soll.
- **DBPassword:** Das zum User gehörige Passwort.
- **IntegratedSecurity:** true setzen, wenn die integrierte Sicherheit von Windows für die Anmeldung verwendet werden soll. In diesem Falle werden DBUser und DBPassword ignoriert. Default: false
- **ShowMessages:** Unterdrücken oder Aktivieren der Fehleranzeigen. Default: false
- **DLLInfo:** gibt Versionsinfo zur DLL zurück.
- **FibuTyp:** Typ der Fibuinstallation oder OEMVersion.
Mögliche Werte:
 - **ftSyska:** Standalone bzw. Standardinstalltion
 - **ftOlymp:** Olymp Version
 - **ftApertum:** Apertum Version
 - **ftStepAhead:** StepAhead Version
- **IsPremium:** Handelt es sich um eine Fibu Premium Version? Die Mastermandanten einer Premium-Version sollten nur bedingt über das OFCK angesprochen werden.

Denn sämtliche Master-Funktionalität (wie Konten in zugeordnete Mandanten kopieren) ist nicht implementiert.

- PersonenGlobal : Gibt an, ob die Personenkonten global für alle Mandanten gelten. Werte:
 - nicht global: 0 oder 2 (Standard ist Personenkonten nicht global)
 - global: 1 oder 3
- UserName (Optional): Anwendername im VORSYSTEM, falls dieser vom DBUser (= interner Name) abweicht. Wird z.B. bei Änderungen eines Personenkontos als Name des Ändernden verwendet.
- Mandanten: Gibt die Collection der Mandanten zurück.
- AnbuInstalliert: Gibt an, ob die Anlagenbuchhaltung für den Mandanten aktiviert ist.
- KoreInstalliert: Gibt an, ob die Kostenrechnung für den Mandanten aktiviert ist.
- MitStapelsperre: Gibt an, ob Fibu-Stapel durch OFCKS.NET gesperrt werden. Dies kann durch den Konfigurationseintrag OFCK-MitStapelsperre = 1 im DB_Inst verlangt werden. Andernfalls sperrt OFCKS.NET einen Stapel nicht, in den es Buchungen einstellt.

Methoden

- DBConnect: Nachdem die nötigen Eigenschaften gesetzt wurden kann damit die Datenbankverbindung aufgebaut werden. Einmal verbunden kann keine Änderungen an den Verbindungseigenschaften mehr vorgenommen werden.
- DBDisconnect: Um vor Beendigung des Programms globale DB-Verbindungen abzubauen, ist DBDisconnect aufzurufen.
- SetSqlConfigLine: Normalerweise werden die Parameter zum Datenbank-Zugang in der Datei sql.config hinterlegt, wie für die Programme des Rechnungswesens auch. Es ist jedoch auch möglich, ohne diese Datei zu arbeiten. Dazu müssen neben den Eigenschaften DBDatabase, DBUser und DBPassword alle notwendigen Zeilen des Eintrags in der Syntax der sql.config mit dieser Funktion übergeben werden. Mindestumfang sind die Einträge provider und connection_string. Es sind jedoch nicht alle Einträge implementiert.

MandantCollection

Collection der Mandanten oder Firmen.

Direktzugriff: Index ist die Mandantenummer. Zurückgegeben wird ein Objekt Mandant.

In einer Premiumversion sind auch Mastermandanten zwar zugänglich, um Eigenschaften abfragen zu können. Bitte beachten Sie jedoch: in einem Mastermandanten können in der Oberfläche einige Stammdaten so eingesehen und gepflegt werden, dass sie aus allen zugeordneten Mandanten abgefragt werden oder in diese kopiert werden. Diese Funktionalität ist in OFCKS.NET nicht vorhanden.

In der Oberfläche sind buchende Funktionen in Mastermandanten nicht zugänglich. Dies wäre theoretisch mit OFCKS.NET möglich. Aber die Ergebnisse könnten nicht genutzt werden.

Aktuelle Generation: 01.

Eigenschaften

Keine außer dem Direktzugriff über Item / this.

Methoden

Keine.

Mandant

Die Daten eines einzelnen Mandanten.

Über dieses Objekt erhält man außerdem alle untergeordneten Collections, jeweils schon gefiltert auf die Daten dieses Mandanten.

Hat ein Benutzer keine Rechte auf Tabellen von Nebenanwendungen (Anlagenbuchhaltung, Kostenrechnung), sieht er nur die Fibu-Mandantendaten.

Anfängliche Generation: 09.

Aktuelle Generation: 10.

Eigenschaften

Mandant stellt diese Collections zur Verfügung:

- Anlagegueter: Abfrage der Anlagegüter bei installierter Anlagebuchhaltung.
- Anzahlungen: Anzahlungsvorfälle
- AufbauZeilen: Collection für Informationen zum Bilanz und GuV Aufbau
- Auftraege: In die Liquiditätsplanung importierte Auftragsvorgänge.
- Auswertungskreise: Auswertungskreise, falls Buchungskreise verwendet werden.
- Banken: Bankverbindungen.
- Belegzeilen: Alle Belegzeilen der Buchungen dieses Mandanten. Überladen mit Einschränkung nach Von-Datum und Konto.
- BelegzeilenByRefID: über die Beleg-RefID können Belegzeilen einer bestimmten Buchung geholt werden. Die Collection enthält dann nur diese Zeilen.
- Bestellungen: In die Liquiditätsplanung importierte Bestellvorgänge.
- Buchungskreise: Buchungskreise, falls benutzt.
- CifPlanzahlen: Planzahlen des Moduls Chefinfo.
- FibuAufbauZeilen: Collection für Informationen zum Bilanz und GuV Aufbau
- FremdWaehrungen: Alle Fremdwährungen. Bei EWS Mandanten ohne die EWS Währungen außer diese wurden bereits vor Einführung des Euro als echte Fremdwährung benutzt.
- Gruppierungen: Kontengruppen.
- Konten: Sachkonten und Personenkonten.
- KoreBelege: Für bereits bestehende Sekundärkostenbuchungen.
- KoreBuchungen: Für neue Sekundärkostenbuchungen, die Buchungsschnittstelle der Kostenrechnung. Als Parameter ist die Stapelnummer anzugeben.

- KorePlanschablonen: Planschablonen der Kostenrechnung (saisonale Verteilung).
- KorePlanwerte: Planwerte der Kostenrechnung.
- KorePrognosewerte: Prognosewerte der Kostenrechnung.
- KoSchablonen: Kostenschablonen.
- Kostenarten: Kostenarten.
- Kostenstellen: Kostenstellen.
- Kostentraeger: Kostenträger.
- Kurse: Kurse einer bestimmten Währung. Nicht für die Hauswährung und nicht für den Euro in einem EWS-Mitgliedsland.
- OPostenDebs: Offene Posten eines Debitoren.
- OPostenKreds: Offene Posten eines Kreditoren.
- Rechnungen: Im Rechnungseingangsbuch erfasste Rechnungen.
- Sprachen: Sprachen.
- Stapel: Collection der Buchungstapel.
- StapelBuchungen: Buchungen in einem Buchungstapel. Dies ist auch die Buchungsschnittstelle der Fibu. Als Parameter ist die Stapelnummer anzugeben.
- Steuern: entspricht den Steuerarten der Fibu.
- SteuerSaetze: Parameter: Steuerart, mit Default: -1 (alle Steuerarten). Steuersätze einer bestimmten Steuerart. Bei (-1) werden alle Steuersätze unabhängig von der Steuerart aufgezählt.
- Zahlziele: Zahlungsziele

Mandant hat die folgenden (Einzel-)Eigenschaften. Die meisten Eigenschaften können nur gelesen werden. Einige wenige Eigenschaften können auch gesetzt werden, diese sind mit "(R/W)" gekennzeichnet.

- AlteHauswaehrung: Bei EWS Teilnehmerländern die Währung vor der Umstellung auf den Euro.
- AnbuStart: Startdatum der Anbu falls diese installiert ist.
- FibuStart: Startdatum der Fibu.
- GeJahrErsteAbsPeriode: Erste absolute Periode eines Geschäftsjahres. Bsp: Fibustart ist der 1.8.2001. GeJahrErsteAbsPeriode = 8
- IsAnbuEingerichtet: TRUE falls Anbu für diesen Mandanten aktiv ist. Parameter (optional):
 - dtDate: Datum zu dem abgefragt werden soll ob die Anbu eingerichtet war.
- IsKoreEingerichtet: siehe IsAnbuEingerichtet, aber für die Kore.
- IsKostenstellePflicht: Bei installierter Kore Angabe ob die Kostenstelle eine Pflichtangabe ist bei der Bebuchung von Kostenkonten.
- Kontenrahmen: Name des installierten Kontenrahmens
- KoreStart: Startdatum der Kore
- Land: Landkürzel des Mandanten (der Firma)
- LetztesGeJahr: Das neueste Geschäftsjahr das im Mandanten eröffnet wurde
- MahnBetrag (R/W): Betrag ab dem Mahnungen geschrieben werden

- MandantenNr: Eindeutige Mandantenummer (das ist auch der Zugriffs-Index).
- MandantenWaehrung: Die (aktuelle) Hauswährung des Mandanten. Die Möglichkeit, nach Datum abzufragen, ist in der Funktion GetMandantenWaehrung gegeben.
- Name (R/W): Name des Mandanten.
- Periodenbestimmt: Periodenfremder oder periodenbestimmter Buchungsmandanten (Enum).
- SteuerNr (R/W): Steuernummer des Mandanten
- Uci: Die Gläubiger-ID des Mandanten für SEPA-Lastschriften (ab Mandant10).

Methoden

- BetragsUmrechnung: Umrechnen eines Betrages einer fremden Währung in die zum Zeitpunkt Date gültige Hauswährung. Rückgabewert ist der umgerechnete Hauswährungsbetrag. Parameter:
 - date: Datum zu dem die Umrechnung erfolgen soll. Falls kein Datum angegeben wird das aktuelle Computerdatum genommen. Für Buchungsberechnungen muss hier das Buchungsdatum übergeben werden.
 - betrag: Betrag in Fremdwährung
 - waehrung: Währungskürzel der Fremdwährung
 - periode (optional): Bei periodenbestimmten Buchungen kann hier eine (Kalender-)Periode übergeben werden. Falls 0 wird die Periode aus date ermittelt.
 - absJahr (optional): Bei periodenbestimmten Buchungen kann hier ein Jahr übergeben werden. Falls 0 wird das Jahr aus Date ermittelt. Es ist das Kalenderjahr anzugeben, nicht das Geschäftsjahr.
- BookTransactionAbort: Bricht eine Transaktion welche mit BookTransactionStart begonnen wurde ab.
- BookTransactionCommit: Schreibt eine Transaktion welche mit BookTransactionStart begonnen wurde in der Datenbank fest.
- BookTransactionStart: Startet eine Transaktion für mehrere Buchungen. Diese Funktionalität bezieht sich nur auf StapelBuchung - Objekte. Alle Buchungen welche von diesem Mandanten erstellt wurden und bei denen die Funktion Insert danach aufgerufen wird werden erst in der Datenbank festgeschrieben wenn BookTransactionCommit aufgerufen wird.
- GetAdressDaten: Funktion zum Holen der Adressdaten des Mandanten. Alle diese Informationen können auch über die Funktion SetAdressDaten geändert werden. Parameter:
 - titel
 - name: Auch über eigene Eigenschaft abfragbar.
 - branche
 - uStIDNr
 - strasse
 - plz
 - ort
 - postfachPLZ
 - postfach

- telefon
- fax
- finanzamt
- faStrasse
- faPLZ
- faOrt
- faBundesland
- referat (Österreich)
- GetKoreEinstellungen: Liefert diverse Systemeinstellungen der Kostenrechnung
 - planungBis: Datum, bis zu dem Planwerte noch geändert werden können.
 - isOnlyFixKosten: True, falls keine Unterscheidung zwischen fixen/variablen Kosten in der Kostenrechnung eingestellt.
 - isMengenErfassung: True, falls die Kostenrechnung für Mengenerfassung konfiguriert ist. Hinweis: zusätzlich spielen noch die Einstellungen der jeweiligen Kostenart eine Rolle.
 - letzterKoreAbschluss: Datum des letzten Kore-Abschlusses; Buchungen vor dieses Datum sind nicht möglich.
 - naechstesAbzuschliessendesGeschaeftsJahr: liefert das nächste abzuschließende Geschäftsjahr zurück (muss nicht identisch mit dem aktuellen Jahr sein, wenn mehrere GJ geöffnet sind).
 - isPlanungBisGesetzt: Kennung, ob ein Planungsende-Datum gesetzt ist.
- GetKoTrDefinition: Kostenträger Definitionen der Kostenrechnung. Rückgabewert: Existiert für diesen Index eine Kostenträgerdefinition. Falls FALSE sind die Inhalte der Felder ungültig (nicht definiert) und sollten nicht abgefragt werden oder für Vergleiche benutzt werden. Parameter:
 - index: 0 bis 4. Für welche der maximal 5 Kostenträger-Definitionen wird abgefragt?
 - bezeichnung: Bezeichnung dieses Kostenträgerfeldes
 - koTrSpalte: Spaltenname aus der Tabelle welcher die Kostenträger-Nummer enthält
 - tabellenName: Name der Tabelle in der die Kostenträger geführt werden.
 - firmRefIDSpalte: Spaltenname der Spalte welche die Mandantenummer enthält
 - koTrBezeichnungSpalte: Spaltenname der Spalte welche die Bezeichnung des Kostenträgers enthält
 - isNumerisch: Information ob es sich um ein numerisches Feld bei koTrSpalte handelt (ob die "Kostenträgernummer" numerisch ist).
 - nrVon: Falls Nummernbereiche angegeben sind die kleinste zulässige Kostenträgernummer welche für diesen Kostenträgerkreis benutzt werden darf
 - nrBis: Falls Nummernbereiche angegeben sind die größte zulässige Kostenträgernummer welche für diesen Kostenträgerkreis benutzt werden darf
 - isPflicht: Muss dieser Kostenträger bei einer Kostenbuchung angegeben werden oder nicht?
- GetStandardKonto: Abfrage eines Standardkontos. Rückgabewerte Kontonummer. Parameter:

- index: Enum der verfügbaren Standardkonten.
- GueltigesBuchDatum: Ermittelt ob es sich bei dem übergebenen Datum für das angegebene Modul um ein gültiges (bebuchbares) Buchungsdatum handelt. Parameter:
 - modul: Modul für das die Prüfung gemacht werden soll. Von dem Enum sind Fibu, Anbu oder Kore hier verwendbar.
 - bookDate: Buchungsdatum
- RealDateToRelativeDate: Umrechnung von absoluter Periode und Kalenderjahr in relative Periode und Geschäftsjahr. Parameter:
 - absPeriode (in)
 - kalenderjahr (in)
 - relPeriode: relative Periode wird zurückgegeben
 - geschaeftsjahr: Geschäftsjahr wird zurückgegeben
- RelativeDateToRealDate: wie RealDateToRelativeDate nur dass aus der relativen Periode und dem Geschäftsjahr die absolute Periode und das Kalenderjahr ermittelt werden. Parameter:
 - relPeriode (in)
 - geschaeftsjahr (in)
 - absPeriode: absolute Periode wird zurückgegeben
 - kalenderjahr: Kalenderjahr wird zurückgegeben
- SetAdressDaten: Siehe GetAdressDaten. Mit dieser Funktion werden die Werte geändert

Nicht mehr unterstützte Methoden:

- ErzeugeBuchung: stattdessen ist das Stapelbuchung-Objekt zu verwenden).
- SetStandardKonten: Die Standardkonten müssen in der Oberfläche der Fibu gesetzt werden.

KontoCollection

Collection der Konten eines Mandanten.

Direktzugriff: Index ist die Kontonummer. Zurückgegeben wird ein Objekt Konto.

Konten sind grundsätzlich alle Kontoarten: die Personenkonten Debitoren und Kreditoren sowie die Sachkonten (außerdem Einmalkonten und Einmaladressen). Eine Kontonummer kann nur als eine Kontoart vorhanden sein. Die Sachkonten sind in der Fibu zu pflegen, stehen der Voranwendung jedoch zur Abfrage zur Verfügung. Die Personenkonten können über OFCKS.NET aus dem operativen System heraus erstellt und abgeglichen werden – eine der beiden Hauptkopplungen zur Buchhaltung neben den Buchungen.

Aktuelle Generation: 05.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- **ExistsKontoKuerzel:** Überprüft das übergebene Kontokürzel ob es bereits verwendet wird. Rückgabe true falls schon verwendet.
- **ExistsKontoNummer:** Überprüft die übergebene Kontonummer ob sie bereits verwendet wird. Rückgabe true falls schon verwendet.

Für alle Filter gilt: Der Filter muss vor dem ersten Zugriff auf die Collection-Daten aufgerufen werden oder RefreshCollData muss aufgerufen werden für ein erneutes Laden der Collection Daten. Kombinationen von Filtern sind möglich. Ob die Kombination sinnvoll ist, wird nicht kontrolliert.

- **Filter:** Einschränkung von kontoVon bis kontoBis (optional).
- **FilterEinmalkonto:** Einschränkung auf die Einmaladressen, die zu einmalkonto gehören. Voraussetzung ist ein Filter auf eine Kontoart von Einmaladressen.
- **FilterKontoArt:** Einschränkung nach bestimmten Kontenarten. Die möglichen Kontenarten zeigt ein Enum auf. Einige der Kontenarten sind Zusammenfassungen mehrerer tatsächlicher Kontenarten.
- **FilterKontoKennung:** Einschränkung nach kontoKennung. Zur Verfügung stehen die Kennungen der Sachkonten sowie der Personenkonten, Einmalkonten und Einmaladressen.
- **FilterKontoKuerzel:** Einschränkung nach kontoKuerzel. SQL Wildcards sind möglich
- **FilterKontoName:** Einschränkung nach kontoName. SQL Wildcards sind möglich
- **FilterKontoStatus:** Einschränkung nach dem kontostatus. Normalerweise werden inaktive Konten nicht gelesen. Mit diesem Filter können die inaktiven Konten getrennt eingelesen werden.
- **FilterKundenRefID:** Einschränkung nach einem Bereich der Kunden-RefID (nur Debitoren und Debitor-Einmaladressen). Diese ist dafür vorgesehen, dass das VORSYSTEM darin seine eigene identifizierende Nummer hinterlegen kann.
- **FilterLiefRefID:** Einschränkung nach der Lieferanten RefID (nur Kreditoren und Kreditor-Einmaladressen). Diese ist dafür vorgesehen, dass das VORSYSTEM darin seine eigene identifizierende Nummer hinterlegen kann.
- **FilterWaWiKennung:** Einschränkung nach der kontoWaWiKennung. Mit dieser kann in Sachkonten hinterlegt werden, ob es sich um ein Aufwands- oder Erlöskonto der Warenwirtschaft handelt.
- **GetNextFreeNo:** Nur für Debitoren und Kreditoren zulässig. Holt die nächste freie Kontonummer. Parameter "Von" und "Bis" sind optional. Mit ihnen kann der Nummernkreis ggf. eingeschränkt werden.
Anmerkung: Werden Von/Bis nicht angegeben ist die Kontonummer um eines größer als das letzte Personenkonto der Kontoart. Sind bisher keine Konten angelegt bei der Kontoart wird die erste freie Nummer zurückgegeben welche gefunden wird (in den meisten Fällen wäre das die Nummer 1, was unerwünscht ist!!!)
- **SortOrder:** Hinweise zu Benutzung siehe Methode "Filter". Ruft die Ergebnismenge sortiert nach einem Kriterium ab (Enum, Standard: Kontonummer).

Konto

Die Daten eines einzelnen Kontos. Das Konto kann ein Sachkonto, ein Personenkonto (Debitor oder Kreditor) oder eine Einmaladresse (eines Debitoren-Einmalkontos oder eines Kreditoren-Einmalkontos) sein. Einmaladressen bekommen vom System eine fortlaufende negative "Kontonummer" zugewiesen.

Über ein Personenkonto sind seine Bankverbindungen (als Collection) zugänglich.

Aktuelle Generation: 11.

Eigenschaften

- AdressFlags: Steuerflags für die Adressbildung für Mahnungen und andere Briefreports (Enum). Der Enum-Wert EinstellungKontoAdressflags kann im Konto selbst nicht verwendet werden.
- aHWSaldo: Saldo des Kontos in alter Hauswährung
- Ansprechpartner: Name des Ansprechpartners
- Bank: die einzige Bank der Kontos (Bank-Objekt) oder die präferierte Bank, falls das Konto mehrere Bankverbindungen hat.
- Banken: Collection aller Bankverbindungen des Kontos. Personenkonten können mehrere Banken hinterlegt haben.
- Belegzeilen: Collection. Alle Belege welche auf dieses Konto gebucht wurden.
- BelegzeilenAbRefID: Über die belegID können alle Belegzeilen des Kontos ab dieser internen Belegnummer geholt werden. Die Collection enthält dann nur diese Zeilen.
- Clearing: Nummer der präferierten Bankverbindung des Kontos.
- ClearingFlag: Zahlungsart (Enum). Mögliche Zahlungsarten sind je nach Kontoart:
 - Debitoren:
 - cfDEBI_Abbuchung
 - cfDEBI_Einzug
 - cfDEBI_Rueckzahlung
 - cfDEBI_SEPALastschrift
 - Kreditoren:
 - cfKRED_ESR (Schweiz)
 - cfKRED_Gutschrift
 - cfKRED_Scheck
 - beide:
 - cfKein (gleichbedeutend mit Rechnung)
 - cfUseDefault ist im Konto selbst nicht zulässig sondern in Buchungen ggf. zu verwenden.
- Datev: Datevautomatik: bei Sachkonten: true = Automatikkonto, false = sonst, bei Personenkonten: true = exportiert, false = nicht exportiert
- Einmal: Personenkonto ist ein Einmalkonto.
- eMail
- eMail2
- eMailAnsprechpartner: Der Email-Ansprechpartner
- EurSaldo: Saldo des Kontos in Euro
- Fax
- Forderungskonto: (Debitoren) falls Leer gilt das Standardforderungskonto welches beim Mandanten abgefragt werden kann.

- GruppeRefID: Kontengruppe (Sachkonten)
- Homepage
- KontoArt: Art des Kontos. Möglichkeiten:
 - kaSachkonto
 - kaDebitorkonto
 - kaKreditorkonto
 - kaDebitoradresse
 - kaKreditoradresse
- KontoKennung: Bei Sachkonten die darin hinterlegte Kontokennung. Als Kennungsnummer und Kennungsbezeichnung des Kontos sind die Werte zu verwenden, die in der syska-ProFI--Tabelle fibu_kontenkennung hinterlegt sind und als Enum zur Verfügung stehen.
Die Typen von Personenkonten tragen ebenfalls spezifische Kennungen. Diese werden beim Anlegen aus der Kontoart automatisch gesetzt.
- KontoKlasse: Nur Sachkonten (Enum). Aktiv-, Passiv- oder GuV-Konto.
- KontoKuerzel: Kurzbezeichnung des Kontos. Nur Grossbuchstaben zugelassen. Übergebene Strings werden automatisch umgewandelt.
- KontoName
- KontoNummer: Eindeutige Kontonummer (das ist auch der Zugriffs-Index). Einmaladressen haben eine negative, vom System vergebene Kontonummer.
- KontoStatus: Konten ohne Inhalt sind aktiv. In einem inaktiven Konto steht "I", andere Status sind (derzeit) nicht zulässig. Die Eigenschaft kann nur gelesen werden, zur Änderung steht die Methode SetKontoStatus zur Verfügung.
- Kostenart: Nummer der Kostenart falls Kostenkonto
- KundenID: (Debitoren) Kundennummer. Wird bei Kontoneuanlage gleich Kontonummer gesetzt, sofern nicht explizit mitgegeben.
- KundenRefID: (Debitoren) interne Referenznummer bei Kunden. In diesem Feld kann das ERP-System seine Nummer hinterlegen. Das Feld ist in der Fibu-Oberfläche nicht zugänglich.
- LandKennung: Landes Kennung bei Personenkonten
- LetztesBuchDatum: Das Datum der letzten Buchung auf das Konto ist abrufbar.
- LieferantenID: (Kreditoren) Lieferantenummer. Wird bei Kontoneuanlage gleich Kontonummer gesetzt, sofern nicht explizit mitgegeben.
- LieferantenRefID: (Kreditoren) interne Referenznummer. In diesem Feld kann das ERP-System seine Nummer hinterlegen. Das Feld ist in der Fibu-Oberfläche nicht zugänglich.
- LieferantenKundenNr: Die Kundennummer welche der Lieferant dem Mandanten vergeben hat. Also die eigene Kunden Nummer, welche man beim Lieferanten hat.
- Mahnbar: (Debitoren) Ist Kunde mahnbar oder nicht?
- MaxMahnstufe: (Debitoren) maximale Mahnstufe, die die OPs des Kunden erreicht haben
- Name2: Felder für zusätzliche Namen
- Name3
- OffenePosten: (Personenkonten) gibt eine Collection der Offenen Posten dieses Kontos zurück.

- Bei Debitoren vom Typ: OPostenDebCollection
- Bei Kreditoren vom Typ: OPostenKredCollection
- Option: Frei verwendbare Ja/Nein-Zusatzfelder für Personenkonten. Parameter: Index im Bereich 0-2.
- Ort
- Ort2: Postfach-Ort
- PLZ: Postleitzahl
- PLZ2: Postfach-Postleitzahl
- Saldo: Kontosaldo. In Mandanten, die vor der Euro-Umstellung begannen, ist das der Saldo in der früheren Währung.
- Sprache: (Debitoren) Mahnsprache
- SteuerartNr: Steuerart die dem Konto zugeordnet ist
- SteuerSatz: Steuersatz der sich aus der Steuerart für dieses Konto ergibt. Parameter: Date. Datum zu dem der Steuersatz ermittelt werden soll. Falls nicht angegeben wird das aktuelle PC-Datum genommen.
- Strasse
- Strasse2: Feld für das Postfach.
- TelefonNr
- Text: Frei verwendbare Text-Zusatzfelder für Personenkonten. Parameter: Index im Bereich 0-2.
- Titel
- UStIDNr: Umsatzsteuer ID Nummer
- UStVAKz: Umsatzsteuer Voranmeldungs-Kennziffer. Nur Sachkonten
- VerbandID: -1 = Verbandsmutter, positive Zahl = Verbandsmitglied bei diesem Konto. Nur für Debitoren und Kreditoren.
- VerbindlichkeitenKonto: Falls leer gilt das Standardverbindlichkeitenkonto welches beim Mandanten hinterlegt ist.
- Waehrung: Kontowährung
- Wert: Frei verwendbare Wert-Zusatzfelder für Personenkonten. Parameter: Index im Bereich 0-2.
- WWSKennung: Kennung, ob das Konto ein Aufwands- oder Erlöskonto der Warenwirtschaft ist.
- ZahlzielName. Name des zum Konto gehörenden Zahlungsziels. Siehe Objekt Zahlziele bzw. Zahlziel.
- ZahlzielRefID: Nummer des zum Konto gehörenden Zahlungsziels. Siehe Objekt Zahlziele bzw. Zahlziel.
- ZahlungsAvis: 0 = gem. Zahlungslauf (Standard), 1 = kein Avis, 2 = immer Avis
- Zessionsvermerk
- ZusatzText: Textfeld für Zusatztext bzw. alternativer Kontoname bei Sachkonten.
- PruefstatusUstid (ab Version 5.0.3.0): Falls die Prüfung der Umsatzsteuer-Identnummern im Vorsystem vorgenommen wird, kann hier der Geprüft-Status gesetzt werden (Enum). Zum Zurücksetzen auf Ungeprüft diesen Status setzen und evtl. das Prüfdatum mit NULL übergeben, um dieses zu löschen. Alternativ liest das Vorsystem den Prüfstatus aus den Fibu-Daten aus.

- PruefungUstid (ab Version 5.0.3.0): Datum der Prüfung der Umsatzsteuer-Identnummer. Wird eingetragen, falls auch ein Prüfstatus übergeben wird. Die Übergabe von NULL löscht das Datum.
- SaldoMitStapel (ab Version 5.0.6.0, nur Lesen): Liefert den aktuellen Kontosaldo inklusive Stapelbuchungen (bis zur aktuellen Kalender-Periode). Dieser Saldo kann für eine Kreditlimitprüfung einen genaueren Wert liefern als Saldo, wenn Schnittstellen-Buchungen noch im Buchungstapel stehen.

Methoden

- CreateNewBank: Anlegen einer neuen Bankverbindung des Kontos, um deren Eigenschaften zu setzen und sie dann mit Insert abzuspeichern.
- GetPeriodenSaldo2: Holt Salden des Kontos zu dem übergebenen Zeitraum.
Parameter:
 - buchungskreisNr: Buchungskreis, wenn ohne Buchungskreise gebucht wird 1.
 - geschaeftsJahr: Geschäftsjahr zu dem der Saldo geholt werden soll
 - periode: relative Periode. bzw. relative Perioden (Enum). Bei periodenfremen Buchern ist die Periode 13 gleich der Periode 12. Es können die Werte der einzelnen Perioden (per01 bis per13) oder Quartale (perQ1 bis perQ4) mit den Enum-Werten perInclAbschluss, perInclEroeffnung und perKumuliert kombiniert werden.
 - soll, haben, saldo: out-Parameter mit den Salden in der aktuellen Mandantenwährung
 - hwSoll, hwHaben, hwSaldo: out-Parameter mit den Salden in der früheren Mandantenwährung vor Euro-Umstellung
- GetZahlverhalten (ab Version 5.0.4.0): Liefert einige Kennziffern zum Zahlungsverhalten des Personenkontos (i.a. Debitoren). Parameter:
 - datumsArt (Enum): Die Datumsabgrenzung kann nach Zahlungsdatum oder Valutadatum geschehen. Im ersteren Fall sind nur OPs in der Auswertung, die mindestens teilbezahlt sind.
 - datumVon: Ab welchem Datum gemäß Datumsart sollen OPs in die Auswertung einbezogen werden? Falls leer, dann mit unbegrenztem Alter.
 - datumBis: Bis zu welchem Datum gemäß Datumsart sollen OPs in die Auswertung einbezogen werden? Falls leer, dann auch neueste OPs.
 - schnellsteZahlung: Ausgabe der schnellsten Zahlung in Tagen.
 - langsamsteZahlung: Ausgabe der langsamsten Zahlung in Tagen.
 - durchschnZahlung: Ausgabe der durchschnittlichen Zahlung in Tagen.
 - durchschnVerzug: Ausgabe der durchschnittlichen Zahlung ab Fälligkeit in Tagen.
 - anzahlPosten: Ausgabe der Anzahl Posten, die in die Auswertung eingegangen sind.
- SetOption: Setzen der Zusatzfelder Option 0 bis 2.
- SetText: Setzen der Zusatzfelder Text 0 bis 2.
- SetWert: Setzen der Zusatzfelder Wert 0 bis 2.
- SetKontoStatus (ab Version 5.0.6.5): Ein Personenkonto kann inaktiv oder auch wieder aktiv gesetzt werden (Enum). Der Enum-Wert Invalid ist hier unzulässig. Darf auf Sachkonten nicht angewendet werden.

Nicht mehr unterstützte Methoden:

- GetPeriodenSaldo: stattdessen GetPeriodenSaldo2 verwenden.
- GetAdresse2, SetAdresse2: verwenden Sie stattdessen die Eigenschaften der Felder der Postfachadresse.

BankCollection

Collection von Bankverbindungen.

Direktzugriff: Index ist die Bankverbindungsnummer (vom System vergeben).

Zurückgegeben wird ein Objekt Bank.

Über den Mandant sind alle Bankverbindungen zugänglich, die Hausbanken zu den Sachkonten sowie die Bankverbindungen der Debitoren und Kreditoren.

Eine Fremdanwendung wird sich in der Regel nur für die Bankverbindungen der Personenkonten interessieren. Diese sind über die Eigenschaft Banken des Kontos zu erreichen.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterKontoArt: Schränkt die Bankverbindungen nach Kontoart (Enum) ein. Nur sinnvoll bei Zugriff über den Mandant. Der Filter muss vor dem ersten Zugriff auf die Daten gesetzt werden.

Bank

Die Daten einer einzelnen Bankverbindung.

Ein Personenkonto kann mehrere Bankverbindungen haben. Unter diesen gibt es die präferierte Bankverbindung. Diese ist vom Konto aus in der Eigenschaft Bank zugänglich. Ihre Banknummer kann im Feld Clearing ausgelesen werden.

Aktuelle Generation: 05.

Eigenschaften

- BankFlag: Enum. Derzeit nur in der Schweiz verwendet zur Unterscheidung von Postkonten und Bankkonten.
- BankKontoNr: Bank-Kontonummer der Bankverbindung
- BankName
- BankRefID: Eindeutige interne Banknummer
- BLZ: Bankleitzahl

- **DoppelteEintraegerVermeiden:** Steuert, ob beim Insert einer neuen Bankverbindung kontrolliert wird, ob die Bankverbindung schon vorhanden ist. Und diese ggf. überschrieben wird, anstatt eine neue anzulegen.
- **IBAN:** International Bank Account Number. Es sollte zusätzlich über die Eigenschaft "SwiftCode" die Swift-Adresse/BIC angegeben werden.
- **Inhaber:** Inhaber des Bankkontos
- **Land:** Land der Bank. Nicht nötig für Inlandsbankverbindungen.
- **Name1:** optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- **Name2:** optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- **Ort:** optional, nur für Auslandsbankverbindungen ohne SWIFT-Code
- **PLZ:** Postleitzahl der Bank. Nur für manche Auslandsbankverbindungen notwendig
- **SepaMandate:** Collection der SEPA-Mandate zur Bankverbindung
- **Strasse:** optional, nur für Auslandsbankverbindungen ohne BIC/IBAN
- **SwiftCode:** Swift-Code oder BIC der Bank
- **Zahlungskennung:** Zahlungskennungen(0=, 1=, bei CH)

Methoden

- **SetFibuKonto:** Mit dieser Funktion muss das zugehörige Konto gesetzt werden. Ausnahme: wenn das Objekt bereits über ein Konto neu angelegt wurde (CreateNewBank in Konto) ist diese Information bereits gesetzt. Einfachster Aufruf: SetFibuKonto(Fibukontonummer, 0, FALSE), dann übernimmt die Funktion das Ermitteln der richtigen Werte.

Für größere Performance können Sie die Werte gleich richtig angeben, wenn sie sowieso schon bekannt sind. In diesem Fall ist direct auf true zu setzen. Die beiden anderen Parameter:

- **kontoNr:** bei bDirect = TRUE muss hier folgendes übergeben werden abhängig von der Kontoart:
für Sachkonten die Kontonummer,
für Debitorenkonten die KundenRefID,
für Kreditorenkonten die LieferantenRefID.
- **kontoArt:** Die entsprechende Kontoart des Fibukontos.
- **SetPreferred:** Bank kann beim zugehörigen Konto als präferierte Bankverbindung eingesetzt werden.
- **InsertSepaMandat:** Füge der Bankverbindung ein SEPA-Mandat an. Das Bank-Objekt muss durch ein Konto-Objekt erzeugt worden sein. Parameter:
 - **bankSepaMandat:** Sepa-Mandatsnummer.
 - **unterschriftsdatum:** für ein neu ausgestelltes Mandant 0 bzw. NULL, für ein gültig vereinbartes Mandat konkretes Unterschriftsdatum.
 - **typ:** Enum: Einzel- oder Dauermandat (Basislastschrift) zur Neuanlage, Widerruf oder Inaktiv, Firmenlastschrift als Einzel- oder Dauermandat.
- **GetNewSepaMandatName:** Liefert die nächste Mandatsnummer (Mandatsname). Voraussetzung ist, dass die automatische Vergabe von SEPA-Mandatsnummern konfiguriert ist.

Zur Verwaltung von SEPA-Mandaten siehe auch Abschnitt "[Modelle der Zusammenarbeit zwischen ERP-System und Fibu](#)".

OPSepaMandatCollection

Collection der SEPA-Mandate zu einer Bankverbindung.

Direktzugriff: MandatRefID, interne laufende Mandatsnummer.

Die SEPA-Mandate gehören zu einer konkreten Bankverbindung. Eine Mandatsnummer darf aktiv nur zu einer Bankverbindung vorhanden sein. Zum einfachen Abgleich von Mandaten kann die Methode `Bank.InsertSepaMandat` verwendet werden. Die Collection erlaubt vor allem ein Auslesen vorhandener Mandate sowie des letzten Nutzungsdatums.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

`FilterAktivDauer`: filtert so, dass nur aktive Dauermandate ausgegeben werden.

OPSepaMandat

Die Daten eines einzelnen SEPA-Mandats.

Aktuelle Generation: 01.

Eigenschaften

- `BankRefID`: Nummer der Bankverbindung, zu der das SEPA-Mandat gehört.
- `Konto`: Personenkonto, zu dem das SEPA-Mandat gehört.
- `MandatRefID`: Laufende Mandatsnummer, vom System vergeben.
- `MandatName`: Externe Mandatsnummer
- `MandatVersionen`: `OPSepaMandatVersionCollection`, die Zugriff auf die historischen Versionen des Mandats erlaubt. Diese wird nicht dokumentiert, da ein Zugriff von außen nicht notwendig sein sollte.
- `UCIReferenz`: Numerischer Teil der Gläubiger-ID der Firma, die zum Zeitpunkt der Ausstellung des Mandats gültig war.
- `GlaebigerID`: Vollständige Gläubiger-ID der Firma, die zum Zeitpunkt der Ausstellung des Mandats gültig war.
- `Typ`: Art des Mandats (Enum): Einzel- oder Dauermandat als Basis- oder Firmenlastschrift.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

Modelle der Zusammenarbeit zwischen ERP-System und Fibu

Modell 1: Mandatsverwaltung nur in der Fibu

SEPA-Mandate werden nur in der Fibu gepflegt. Das ERP-System ruft zu der / den Bankverbindung(en) aus der Fibu die verfügbaren Mandate ab, z.B. bei Auftragseingang. Falls kein passendes vorhanden ist, veranlasst der ERP-Anwender beim Fibu-Anwender die Erstellung eines Mandats.

Notwendig für dieses und alle Modelle: lesender Zugriff auf die SEPA-Mandate über Bank5.SepaMandatColl.

Modell 2: ERP stößt Mandatsausstellung an, weitere Verwaltung in der Fibu

Ausgangspunkt wie Modell 1. Wenn kein passendes Mandat vorhanden ist, fordert das ERP-System eine neue Mandatsnummer an (mit Bank5.GetNewSepaMandatName), merkt sich diese und legt das Mandat im Status ausgestellt an (mit Bank5.InsertSepaMandat). Die weitere Abwicklung geschieht in der Fibu. Aus dem ERP-System wird z.B. zur Erstellung der Rechnung wieder nachgeschaut, ob es inzwischen ein gültiges Mandat gibt.

Hierzu notwendig GetNewSepaMandatName und InsertSepaMandat, letzteres nur einmalig pro Mandat.

Modell 3: ERP ist das führende System für die SEPA-Mandate

Das ERP-System übernimmt die komplette Mandatsverwaltung und hält über die Funktion InsertSepaMandat die Fibu aktuell. Über InsertSepaMandat werden auch Statusübergänge Ausgestellt -> Aktiv (mit Unterschriftsdatum) -> ggf. Inaktiv oder Widerruf mitgeteilt. Zulässige Status(übergänge) sind in der Dokumentation zur Fibu beschrieben. Es sollte immer nur 1 aktives Dauermandat an einer Bankverbindung geben.

StapelCollection

Collection der Buchungstapel eines Mandanten.

Direktzugriff: Index ist die Stapelnummer (vom System vergeben). Zurückgegeben wird ein Objekt Stapel.

Buchungstapel gibt es in der Fibu und in der Kore, wobei die Nummerierung Arten-übergreifend ist.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterStapelArt: Filtern nach der Art des Stapels (Enum). Z.B. Fibustapel oder Korestapel und ausschließen von Privatstapeln.

Stapel

Die Daten eines Stapel – Fibu- oder Korestapel.

Etliche Eigenschaften können nur gelesen werden. Die vollständige Verwaltung der Stapel muss in der Fibu geschehen.

Aktuelle Generation: 03.

Eigenschaften

- FaelligkeitsDatum: Ende-Datum für diesen Stapel
- IsLocked: Abfrage ob Stapel aktuell gesperrt ist. Die Eigenschaft wird bei Erstellung des Objekts einmalig gesetzt.
- IsWiederkehrend: Ist es ein wiederkehrender Stapel?
- StapelArt: Art des Stapels (Enum). Mögliche Werte:
 - staBrutto: Bruttostapel
 - staNettoInclSteuer: Netto Stapel
 - staKoreStapel: Kostenrechnungsstapel. Nur für Kostenrechnung.
- StapelName: Bezeichnung
- StapelNr: Eindeutige Nummer des Stapels
- WiederkehrArt: Art der Wiederkehr (Enum)
- WiederkehrendIn: Anzahl Tage oder Monate, in denen der Stapel wiederkehren soll

Methoden

Besonderheit bei der Standardmethode Delete: nur leere Stapel ohne Buchungen können gelöscht werden.

- Lock: Sperrt den Stapel für die Verwendung aus der Fibu bzw. Kore. Der Stapel kann in der Fibu in keiner Weise bearbeitet werden.
- Unlock: Entsperrt den Stapel für die Verwendung aus der Fibu bzw. Kore. Der Stapel kann wieder bearbeitet werden und z.B. überprüft und verbucht werden.

StapelBuchungCollection

Collection der Stapelbuchungen eines Buchungsstapels.

Direktzugriff: Index ist die Stapelnummer (vom System vergeben). Zurückgegeben wird ein Objekt StapelBuchung.

Die Finanzbuchhaltung kennt die Dialogbuchung (jede Buchung wird sofort verbucht) und die Stapelbuchung (Buchungen werden erfasst, geprüft und anschließend gesammelt verbucht). Vorkontrollsysteme müssen Buchungen in einen Buchungsstapel einstellen. Dazu dient das Objekt StapelBuchung zusammen mit der StapelBuchungCollection.

StapelBuchungCollection und StapelBuchung sind nicht zum Auslesen von Stapelbuchungen gedacht, sondern nur zur Übergabe von Buchungen in einen Stapel. Im Regelfall ist in der Voranwendung hinterlegt, in welchen Buchungsstapel sie (in einem Mandanten) übergibt. Falls eine Auswahl zu treffen ist, ermöglicht die StapelCollection das Auslesen der Eigenschaften der Buchungsstapel.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

Stapelbuchung

Ein Container, um die Daten für eine Buchung aufzunehmen, die in den Buchungsstapel der Fibu eingefügt werden soll.

Das Stapelbuchen sperrt einen Stapel nicht gegen andere Benutzer. Ein Benutzer kann dadurch in die Buchungen eines Stapels schauen, auch wenn OFCKS.NET in den Stapel bucht. Das erlaubt eine Kontrolle, birgt aber auch die Gefahr, dass sich die Buchung des OFCKS.NET und Aktionen des Benutzers gegenseitig stören. Setzt man im DB_Inst unter Einstellungen & Werte den Eintrag OFCK-MitStapelsperre = 1, sperrt OFCKS.NET gegen andere Benutzer.

Aktuelle Generation: 12.

Eigenschaften

Bei den Eigenschaften handelt es sich um die Kopfdaten der Buchung. Diese müssen vor den Detaildaten der Buchung gesetzt werden. Die Detaildaten werden über die Methoden gesetzt, wie z.B. AddHaben, AddSoll für die Buchungszeilen.

- **AnzahlungName:** Names des Anzahlungsvorfalles, wird nur beachtet bei Rechnungsbuchungen im Anzahlungsverfahren (Abschlags-, Schlussrechnungen). Der Name ist nur notwendig, wenn Anzahlungsvorfälle im Zuge des Buchens über ihren Namen automatisch angelegt werden sollen: Falls der Name nicht existiert, wird ein neuer Anzahlungsvorfall angelegt.
- **AnzahlungRefID:** Nummer des Anzahlungsvorfalles, wird nur beachtet bei Rechnungsbuchungen im Anzahlungsverfahren (Abschlags-, Schlussrechnungen).
- **AuftragID:** Ein freies Feld in dem das VORSYSTEM eine eigene Auftragsnummer mitgeben kann. Syska ProFI hat keine auf der Auftragsnummer basierende Funktionalität.
- **BelegDatum (Pflicht):** Das Belegdatum muss als einer der ersten Werte gesetzt werden zusammen mit der Belegwährung. Auf jeden Fall bevor irgendeine der Buchungszeilen-Methoden AddHaben oder AddSoll aufgerufen werden. Bestimmt in einem periodenfreien Mandant zugleich über die Buchungsperiode.
- **BelegNr: (Pflicht).** Die externe Belegnummer. Intern bekommt jede Buchung außerdem eine Buchungsnummer, die erst beim Verbuchen vergeben wird.
- **Buchungsart (Enum):** Art der Buchung. Für übliche Buchungen des täglichen Geschäftsverkehrs ist baLfdBuchung zu wählen. Erläuterungen zu den meisten Buchungsarten enthält die Dokumentation zum Syska ProFI. Zusätzliche, an der Oberfläche nicht vorhandene Möglichkeiten:
 - **baAZSchlussKomplettOhneAZ:** Anzahlungsschlussrechnung über den Gesamtbetrag (ohne Anzahlung). Die Standardmethode baAZSchlussrechnung dagegen verwendet nur die noch nicht berechneten Differenzbeträge zum Gesamtvolumen.
 - **baAZSchlussKomplettMitAZ:** Anzahlungsschlussrechnung über den Gesamtbetrag mit Ausbuchungszeilen der Anzahlungen (negativ).
 - **baAZTeilschlussrechnung:** Teilschlussrechnung im Anzahlungsverfahren, Sonder-Erweiterung.

- baAZLetzteTeilschlussrechnung: Letzte Teilschlussrechnung, die einen Anzahlungsvorfall mit Teilschlussrechnungen abschließt.
- baAZTSRStornoOffen: Storno ggf. noch offener Anzahlungsrechnungen im Zuge einer Teilschlussrechnung.
- baAZStorno.....: Über die Schnittstelle können alle Rechnungen im Anzahlungsverfahren storniert werden – sofern der Zustand das noch zulässt. Ggf. muss die schon gebuchte Zahlung darauf zuerst in der Oberfläche storniert werden.
- BuchungskreisNr: Buchungskreisnummer. Falls nicht angegeben wird die Standardbuchungskreisnummer 1 gesetzt.
- DMSDocumentID: ID des zugehörigen Dokumentes im Dokumentenmanagementsystem (DMS).
- Einheit: Einheit zum Fremdwährungskurs, falls der Kurs der einfacheren Lesbarkeit halber mit abweichender Zehnerpotenz angegeben ist.
- GeschaeftsJahr: (bei Mandanteneinstellung „Periodenbestimmt buchen“) Geschäftsjahr (!nicht Kalenderjahr! Wichtig bei Mandanten mit vom Kalenderjahr abweichendem Geschäftsjahr) in das gebucht werden soll. Falls nicht angegeben wird das Geschäftsjahr automatisch aus dem Belegdatum ermittelt. Wird diese Eigenschaft angegeben, so muss auch die Eigenschaft „Periode“ gesetzt werden.
Hinweis: Zur Ermittlung des Geschäftsjahres zu einem Kalenderdatum kann die Methode "RealDateToRelativeDate" (aus Objekt Mandant) verwendet werden. Alternativ kann zum Setzen des Geschäftsjahres und der Periode die Methode „SetAbweichendePeriodeAbs“ verwendet werden.
- Kategorie (optional): Wenn mit Kategorien gebucht wird kann diese hier angegeben werden. Es findet keine Prüfung statt, ob die Kategorie in den Stammdaten angelegt ist. Beim Ändern von Buchungen im Stapel kann die Kategorie jedoch ggf. verloren gehen.
- Kurs (optional): Angabe eines Umrechnungskurses. Dies hat rein informativen Charakter. Es werden keine Umrechnungen über diesen Kurs vorgenommen. Ab OFCKS.NET 5.0.6.1: Mit der Sondereinstellung FI-FW-KursAusBeleg=1 wird (bei fehlendem Eigenwährungsbetrag) in Fremdwährungsbelegen mit dem Kurs des Belegs umgerechnet. Dann muss aber konsequent in allen Fremdwährungsbelegen ein Kurs mitgegeben werden!
Bei EWS Mandanten darf jedoch für EWS Währungen kein Kurs angegeben werden!
- Periode: (bei Mandanteneinstellung „Periodenbestimmt buchen“) absolute (kalendarische) Periode in die gebucht werden soll. (Januar = 1, Februar = 2, ...). Falls nicht angegeben wird die Periode automatisch aus dem Belegdatum ermittelt. Wird diese Eigenschaft angegeben, so muss auch GeschaeftsJahr gesetzt werden. Hinweis: Zur Ermittlung des Geschäftsjahres zu einem Kalenderdatum kann die Methode "RealDateToRelativeDate" (aus Objekt Mandant) verwendet werden. Alternativ kann zum Setzen des Geschäftsjahres und der Periode die Methode „SetAbweichendePeriodeAbs“ verwendet werden.
- StapellfdBuchNr (nur Lesen): Erst nach dem Insert verfügbar. Gibt die laufende Stapelbuchungsnummer zurück.
- StapelRefID: Eigenschaft zum Auslesen der Stapelnummer für die Buchungen. Die Stapelnummer ergibt sich daraus, als welchem Stapel-Objekt heraus man die Stapelbuchung erzeugt hat.
- Verdichten: Bool. Steuerung ob die Fibubuchung und Kostenkontierungszeilen verdichtet werden soll. Kriterien der Verdichtung siehe unten "Hinweise zur Verdichtung" (nach den Methoden).

- VerdichtenNachVZ: Bool. Steuerung ob die Buchungszeilen und Kostenkontierungen verdichtet werden sollen, wobei nur Buchungen gleichen Vorzeichens zusammengefasst werden.
- Waehrung (Pflicht): Wahrung der Buchung. Diese Eigenschaft muss gesetzt werden bevor AddHaben oder AddSoll aufgerufen werden (ebenso wie das BelegDatum). Ggf. muss die Hauswahrung mitgegeben werden.
- Zahlungssperre: fur OP-Anlage mit Zahlungssperre: Datum, bis zu dem der OP gesperrt sein soll.
- Quelle: In diesem Feld kann und sollte die verwendende Anwendung ihren Namen mitgeben. Die Buchungsherkunft kann dann in der Finanzbuchhaltung ruckverfolgt werden. Es stehen zwar 50 Zeichen zur Verfugung, Sie sollten jedoch eine in bis zu 15 Zeichen lesbare Bezeichnung verwenden. Wird das Feld nicht gefullt, tragt OFCKS.NET sich als Buchungsherkunft ein.

Methoden

- AddAnbuHaben (optional, aber Pflicht falls ein Anlagenkonto in Haben bebucht wird mit installierter Anbu): Identifizierende Zusatzangaben fur die Anlagenbuchhaltung. Es werden nur die Bewegungsarten Gutschrift (Teilgutschrift) und Gutschrift Ruckgabe (Vollgutschrift) gebucht. Stornobuchungen sind nicht moglich. Parameter:
 - inventarNr: Inventarnummer des gutzuschreibenden Anlagegutes.
 - unterNr: Unternummer des gutzuschreibenden Anlagegutes.
- AddAnbuSoll (optional, aber Pflicht falls ein Anlagenkonto in Soll bebucht wird mit installierter Anbu): Identifizierende Zusatzangaben fur die Anlagenbuchhaltung. Es werden nur die Bewegungsarten Zugang aus Kauf und Teilzugang aus Kauf gebucht. Die restlichen Angaben mussen in der Anbu manuell erfasst werden. Stornobuchungen sind nicht moglich. Parameter:
 - inventarNr: Inventarnummer des zu bebuchenden Anlagegutes fur einen Teilzugang. Bei einem Neuzugang ohne geplantem Anlagegut muss hier ein Leerstring ubergeben werden.
 - unterNr: Unternummer des zu bebuchenden Anlagegutes fur einen Teilzugang. Bei einem Neuzugang ohne geplantem Anlagegut muss hier eine 0 ubergeben werden.
 - koSt: Kostenstelle fur das Anlagegut (auf die spater die kalk. AfA der Anlage gebucht wird). Angabe ist optional.
- AddFreieWerte: Ubergabe der bis zu 15 freien Werte fur einen Beleg. In der Fibu werden diese als Belegzusatze bezeichnet. Deren Bezeichnungen konnen definiert werden.
 - wert1: 1. freier Wert
 - wert2: 2. freier Wert
 - ...
 - wert15: 15. freier Wert
- AddHaben5: Buchungszeile der Habenseite hinzufugen, aktuelle Generation 5. Mit gleichem Aufbau auch AddSoll5. Parameter:
 - konto: Habenkonto
 - bruttobetrag: Bruttobetrag der auf dieses Konto gebucht werden soll in Buchungswahrung.

- steuerbetrag: Steuerbetrag in Buchungswährung, falls auf dieses Konto mit Steuer gebucht wird.
- eigenBetrag: (muss/optional) Bruttobetrag in Hauswährung des Mandanten (immer in Hauswährung zum Zeitpunkt der Buchung, NICHT frühere Hauswährung vor Euro-Umstellung). Muss nur angegeben werden bei Fremdwährungsbuchungen (nicht nötig bei Buchungen in Hauswährung oder alter Hauswährung). Kann mit Hilfe der Funktion "EigenBetrag" des Mandant-Objekts errechnet werden.
Wird 0 übergeben werden die Eigenbeträge intern über den hinterlegten Kurs berechnet und falls nicht NULL übergeben wurde zurückgegeben. Ein evtl. mitgegebener Kurs wird nicht zur Umrechnung benutzt (Ausnahme siehe oben zum Kurs). Empfehlung: Betrag in Belegwährung und Eigenwährung mitgeben, dann besteht definitiv Betragsgleichheit zum VORSYSTEM.
- eigenSteuer: (muss/optional) Steuerbetrag in Hauswährung. Es gelten die Regeln für den EigenBetrag.
- koreEigenBetrag: (Rückgabe) gibt den in Eigenwährung erwarteten (Netto-)Betrag für eine eventuelle Kostenkontierung zurück. Der Betrag wird immer zurückgegeben auch bei Mandanten welche nicht mit Kore arbeiten. Er berechnet sich aus FremdBruttobetrag abzüglich Fremdsteuerbetrag und das Ergebnis umgerechnet in Eigenwährung.
- zahlziel: (optional) Bezeichnung (nicht ZahlzielRefID) des Zahlzieles, auf der Personenkonto-Seite der Buchung zu setzen.
Hierbei kann das Kürzel eines in Syska ProFI definierten Zahlziels oder ein individueller Zahlzielstring nach folgendem Muster verwendet werden:
NnSsPp,pTtQq,qMmZz
n=Nettotage, s=Skontotage, p=Skontoprozent, t=Skontotage 2,
q=Skontoprozent 2, m=Zielmonat (0=laufender, 1=nächster), z=Zieltag im Monat.
Beispiel:
N30S10P2,5 = netto 30 Tage, 10 Tage Skonto mit 2,5%
N60S10P2,5T20Q1,0 = netto 60 Tage, 10 Tage Skonto mit 2,5%, 20 Tage Skonto mit 1,0% (zweistufiges Skonto)
- buchungstext: (optional) Der Buchungstext der 1. Zeile wird zum Belegkopf-Buchungstext.
- valutaDatum: (optional) falls nicht angegeben wird das Buchungsdatum als Valutadatum verwendet.
- opBelegNr: (optional) Rechnungsnummer des OPs. Wenn nicht gefüllt, wird er gleich der Fibu-Belegnummer gesetzt.
- kostenschablone: (optional) Angabe einer Schablone zur Kostenaufteilung, nur mit installierter Kore erlaubt.
- kostenart: (optional) Falls abweichend kann hier bei installierter Kore die Kostenart angegeben werden.
- steuerartRefID: Es kann eine vom Konto abweichende Steuerart angegeben werden. Soll der Default des Kontos verwendet werden so muss hier -1 übergeben werden.
Es wird empfohlen diesen Parameter zu übergeben, insbesondere wenn ein Konto mit wechselnden Vorsteuersätzen bebucht wird.
Achtung: vom Bebuchen von Erlöskonten mit wechselnden Umsatzsteuersätzen wird dringend abgeraten, da dies zu Abweichungen bei der Umsatzsteuervoranmeldung führen kann!
- einmalAdressRefID: RefID der Einmal-Adresse (des Einmal-Kreditors oder -Debitors) für diese Buchung. Einmalkonten können auch ohne Einmaladresse

- bebuht werden. Dies sollte jedoch aus GOB-Grundsätzen nur in Ausnahmefällen genutzt werden.
- esr: (optional) Angabe einer ESR-Nummer für Zahlungsverkehr in der Schweiz.
 - skontierfaehig: (Enum) Die Skontierfähigkeit ist an den Sachbuchungszeilen anzugeben und kann gemischt angegeben werden. Mit skfEntsprechendStamm wird die Skontierfähigkeit aus dem Sachkonto verwendet. Standard ist skfNicht, deshalb sollte bei Rechnungsbuchungen eine Skontierfähigkeit gesetzt werden.
 - rechnungID: Übergabe der RechnungID für den Sachkonten-OP pro Stapelzeile.
 - abzugsfaehig: Prozentsatz einer nur teilweisen Vorsteuer-Abzugsfähigkeit (sonst NULL oder nicht vorhanden). Eine Abzugsfähigkeit von 0 % kann über die Schnittstelle nicht gesetzt werden, wohl aber Prozentsätze > 0, auch 100 %.
 - AddKoreHaben: (nur mit angeschlossener Kore) Zur Übergabe der Kostenkontierung(en) zur Buchungszeile. Die "AddKoreHaben"-Methode muss im Anschluss an die zugehörige "AddHaben"-Methode aufgerufen werden, bevor eine weitere Fibu-Buchungszeile übergeben wird. Kore-Splitting ist möglich, d.h. zu einer Buchungszeile können mehrere Kostenzeilen übergeben werden. Deren Summe muss dem Nettobetrag des Fibu-Buchungsbetrags entsprechen. Parameter:
 - nettoEigenBetrag: Nettobetrag in Eigenwährung. Es kann 0 übergeben werden, dann muss nettoBetrag gefüllt sein und wird ggf. anhand des Kurses der Währungsverwaltung umgerechnet. Wenn keine Kostenaufteilung vorliegt, kann der Rückgabewert aus AddHaben5.koreEigenBetrag verwendet werden. Wird ein Wert übergeben gilt: Der angegebene Kore-Betrag in Eigenwährung muss dem Fibu-Nettobetrag in Eigenwährung(!), der auf der Fibu-Habenseite gebucht wurde (Brutto – Steuer aus "AddHaben" falls nur eine Zeile übergeben wird), entsprechen. Dies muss vom Aufrufer sichergestellt werden ansonsten gibt es einen Fehler.
Das Betragsvorzeichen muss mit dem des Fibu-Betrags identisch sein (i.a. also positiv). OFCKS.NET gibt Kostenbeträgen zu Habenbuchungen dann das umgekehrte Vorzeichen.
 - nettoBetrag: Nettobetrag in Belegwährung.
 - koSt: Kostenstelle falls erforderlich. Ansonsten ein Leerstring, falls in Kore nicht als Pflicht eingestellt.
 - koTr1: (optional). Angabe des Merkmals, das in der Kostenrechnung als Kostenträger 1 definiert wurde. Es müssen die Einstellungen der Kore berücksichtigt werden, z.B. wenn Kostenträger als Pflichtfeld deklariert wurden.
 - koTr2 – koTr5: siehe koTr1 für die bis zu 5 Kostenträger-Dimensionen, die definiert werden können.
 - fixKosten (Enum): Fixkosten (= fvFix, default) oder Variabel (= fvVariabel). Nur notwendig, wenn in dem Mandant die Unterscheidung getroffen wird.
 - menge: optionale Angabe der Menge, falls mit Mengen gebucht wird.
 - einheit: optionale Angabe der Mengeneinheit, falls mit Mengen gebucht wird.
 - buchungstext (optional)
 - AddKoreSoll: (nur mit angeschlossener Kore) Zur Übergabe der Kostenkontierung(en) zur aktuellen Soll-Buchungszeile. Beschreibung siehe "AddKoreHaben".
 - AddOpExHaben4 (optional): Zur aktuellen Haben-Buchungszeile (auf ein Personenkonto) können zusätzliche OP Daten übergeben werden. Die Bankdaten sollten nur angegeben werden, wenn sie von der hinterlegten präferierten Bankverbindung abweichen. Aktuelle Generation 4. Parameter:

- clearingFlag: Zahlungsart. Falls nicht wie im Personenkonto hinterlegt bezahlt werden soll kann hier eine abweichende Clearingmethode angegeben werden. Bei Übergabe von cfUseDefault wird die Zahlungsart aus dem Konto verwendet.
- bankKontoNr: Kontonummer der Bank
- bankBLZ: Bankleitzahl der Bank
- bankName: Name der Bank
- sprache: Sprache, in der der OP ggf. gemahnt werden soll.
- bankLand: Land der Bank für Zahlungswege, bei denen die Adresse der Bank notwendig ist.
- bankIban: IBAN der Bankverbindung.
- bankSwift: Swiftcode oder BIC der Bankverbindung.
- bankText1 bis bankText5: Weitere Angaben zur Adresse für Zahlungswege, bei denen die Adresse der Bank notwendig ist. Die Felder sind belegt Name1 / Name2 / Ort / PLZ / Strasse.
- opUStIDNr: Individuelle Umsatzsteuer-Identnummer zur Buchung
- opSepaMandat: SEPA-Einzelmandat zum OP oder das zu verwendende Dauermandat.
- opSepaTermin: Gewünschter Ausführungstermin einer SEPA-Lastschrift.
- AddOpExSoll4: Zur aktuellen Soll-Buchungszeile (auf ein Personenkonto) können zusätzliche OP Daten übergeben werden. Beschreibung siehe " AddOpExHaben4".
- AddSoll5: Buchungszeile der Sollseite der hinzufügen. Parameter siehe "AddHaben5" (textuelle Referenzen auf Haben-Buchungen sind durch "Soll" zu ersetzen). Die Reihenfolge von "AddHaben" und "AddSoll" sind nicht von Bedeutung.
- Insert: Nachdem der komplette Buchungsbeleg mit den Eigenschaften und den Add-Methoden aufgebaut wurde, wird er mit dieser Methode in den Stapel eingefügt.
- SetAbweichendePeriodeAbs: (bei Mandanteneinstellung "Periodenbestimmt buchen"): Alternativ zum Setzen der Eigenschaften GeschaeftsJahr und Periode, falls die Buchung in eine vom Belegdatum abweichende Buchungsperiode gehen soll. An die Methode ist der Kalendermonat und das Kalenderjahr der gewünschten Buchungsperiode zu übergeben, daraus wird die Fibu-Periode und –Geschäftsjahr gerechnet und gesetzt.
- SetAbweichendePeriodeRel: (bei Mandanteneinstellung "Periodenbestimmt buchen"): Umgekehrte Funktion zu SetAbweichendePeriodeAbs: Buchungsperiode wird aus dem relativen Monat (d.h. gerechnet ab Geschäftsjahresbeginn) und dem Geschäftsjahr abgeleitet.
- SetKoreBetragAutoKorrektur: Erlaubt die Vorgabe einer Kleindifferenzgrenze, unter der Abweichungen zwischen Fibu und Kore automatisch ausgeglichen werden sollen. Parameter:
 - maxAbweichung: Betrag der maximalen Abweichung
 - autoKorrektur (Enum): Korrektur pro Zeile oder pro Beleg
 - value: Soll korrigiert werden?

Hinweis: Im OFCK (COM-Variante) gab es eine versteckte Eigenschaft KoreBetragAutoKorrektur, die man jedoch wegen notwendigen 3 Parametern nur über ihre put-Methode aufrufen konnte.
- SQLEx: Optionaler zusätzlicher SQL Befehl welcher durch das Objekt innerhalb der Transaktion der Stapelbuchung mit ausgeführt wird. Bsp. kann hiermit ein eigener

Übergeben-Status gesetzt werden in Abhängigkeit ob die Buchung funktioniert oder nicht.

- **Validate:** Validiert den aufgebauten Buchungsbeleg und führt gegebenenfalls die Verdichtung durch. Danach können keine Werte mehr gesetzt werden. Es werden fehlende Werte gegebenenfalls ermittelt bzw. berechnet. Schlägt das Validieren fehl kann das Objekt nicht korrigiert werden aufgrund der berechneten Werte sondern es muss ein neues Objekt erstellt und mit korrekten Werten gefüllt werden.

Hinweise zur Verdichtung:

Falls verdichtet wird werden alle Fibuzeilen und Korezeilen zusammenaddiert welche die im folgenden aufgeführten Kriterien erfüllen. Es werden keine Betragsänderungen durchgeführt falls durch Rundungsdifferenzen nicht übereinstimmende Fibu- und Korebeträge entstehen.

Folgende Werte müssen übereinstimmen damit die Fibuzeilen verdichtet werden:

- Anbu Inventarnummer (falls Anlagenbuchung)
- Anbu Unternummer (falls Anlagenbuchung)
- Kostenart (falls inklusive Kostenrechnung)
- Fibu Kontonummer
- Bank BLZ
- Bank Kontonummer
- Bank Name
- ESR-Nummer (Schweizer Buchung)
- Kategorie
- OPBeleg Nummer
- Zahlziel
- Clearing Flag (Zahlungsart)
- Einheit der Kostenrechnung (falls mit Kostenrechnung und übergebenen Einheiten)
- Skontierfähigkeit

Folgende Werte müssen übereinstimmen damit die Korezeilen verdichtet werden:

- Fixkostenkennung
- Kostenstelle
- Kostenträger 1 bis 5

AnlagegutCollection

Collection der Anlagegüter der Anlagenbuchhaltung.

Direktzugriff: Kombination aus Inventarnummer und Unternummer (3stellig)
alphanumerisch.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- **FilterStatus:** Filter auf einen bestimmten Anlagenstatus (Enum), falls als 2. Parameter true mitgegeben wird. Wird als 2. Parameter false mitgegeben, werden alle Anlagen ohne diesen Status aufgelistet.

Anlagegut

Daten eines Anlageguts in der Anlagenbuchhaltung.

Die Anlagegüter können nur gelesen werden.

Aktuelle Generation: 02.

Eigenschaften

- **AfAEnde:** Datum an dem die AfA zu Ende war
- **AfARestwert:** Restwert des Anlagegutes zur Abschreibung
- **AHK:** Anschaffungs- und Herstellungskosten
- **AnbuKonto:** Anlagenkonto auf dem das Anlagegut gebucht wurde.
- **Anschaffung:** Datum der Anschaffung des Anlagegutes
- **Bezeichnung:** Bezeichnung des Anlagegutes
- **InventarNr:** Inventarnummer
- **IsAltbestand:** Ob das Anlagegut bereits vor dem Start der Anbu vorhanden war und als Altbestand übernommen wurde
- **Kostenstelle**
- **LieferantenKonto:** Fibu-Konto des Lieferanten von dem das Anlagegut gekauft wurde (oder Finanzkonto)
- **Status:** Status des Anlagegutes, Enum: AIB (Anlage im Bau), Aktiv, Deaktiv, Geplant
- **UnterNr:** Unternummer zur Verwaltung komplexer Anlagegüter. Bildet zusammen mit der Inventarnummer den eindeutigen Schlüssel.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

AnzahlungCollection

Collection der Anzahlungsvorfälle, über die Buchungen im Anzahlungsverfahren organisiert werden.

Direktzugriff: Numerische ID des Anzahlungsvorfalles.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- **GetBezByKey**: Holt die Bezeichnung des Anzehlungsvorfalls anhand der Nummer. Alternativ kann auch die Eigenschaft ausgelesen werden.

Anzahlung

Daten eines Anzehlungsvorfalls.

Anzehlungsvorfälle organisieren Buchungen im Anzehlungsverfahren von der Abschlagsrechnung bis zur Bezahlung der Schlussrechnung. Anzehlungsvorfälle können auch automatisch mit der ersten Buchung auf einen Vorfall angelegt werden.

Aktuelle Generation: 03.

Eigenschaften

- **Anmerkung**: Ausführliche Anmerkung zum Anzehlungsvorfall.
- **AnzahlungNr**: Vom System vergebene numerische ID des Anzehlungsvorfalls.
- **AZDebitorZahlungen**: Collection der Anzehlungen in dem Vorfall.
- **Bezeichnung**: Bezeichnung des Anzehlungsvorfalls, der vom Vorsystem vergebene Schlüssel.
- **BuchungskreisNr**: Buchungskreisnummer

AufbauZeileCollection

Collection von Zeilen der Bilanz bzw. GuV.

Direktzugriff: Nicht vorgesehen.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- **FilterAufbau**: Art des Aufbaus der abgefragt werden soll (Enum). Für die Bilanz sind Passiva und Aktive getrennte Aufbauten, außerdem der GuV-Aufbau.

AufbauZeile

Daten einer Zeile in Bilanz bzw. GuV.

Ein Bilanzaufbau Passiva bzw. Aktiva oder ein GuV-Aufbau besteht aus Überschrifts- und Kontenzeilen, durchnummeriert mit dem Index. Noch nicht eingehängte neue Konten haben

einen negativen Index. Der Index ist nicht konstant, sondern kann sich mit der nächsten Änderung am Aufbau ändern.

Aktuelle Generation: 01.

Eigenschaften

- AufbauArt: Art des Aufbaus (Enum). Für die Bilanz sind Passiva und Aktive getrennte Aufbauten, außerdem der GuV-Aufbau.
- Index: Index der Zeile, bestimmt die Reihenfolge. Ein negativer Index bedeutet es ist ein nicht in den Aufbau eingehängtes Konto.
- KontoNr: Falls ungleich 0 handelt es sich um die Kontonummer. Überschriften geben hier eine 0 zurück.
- Text: Bei Überschriften: Überschriftentext. Bei Konten: Kontobezeichnung
- ZeileArt: Art der Zeile (Enum): Eine vom max. 3 Überschriften-Ebenen, Kontozeile oder Kontozeile im Ausweichaufbau (auf der anderen Bilanzseite).

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

AuftragCollection

Collection der Zahlungsdaten eines Warenwirtschaft-Auftrags für die Liquiditätsplanung.

Direktzugriff: Die vom System vergebene ID der Bestellung.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

Auftrag

Zahlungsdaten eines Warenwirtschaft-Auftrags für die Liquiditätsplanung.

Zur Übergabe von Auftragsdaten an die Liquiditätsplanung der Chefinfo (falls nicht ein View in das ERP-System genutzt wird). Die Verwendung dieser Funktion setzt voraus, dass eine Tabelle „cif_li_auftrag“ vorhanden ist, vgl. Schnittstellendokumentation zur Liquiditätsplanung. Informationen zur Übergabe der einzelnen Felder siehe ebenfalls Schnittstellendokumentation zur Liquiditätsplanung.

Vor der Übergabe der Auftragsdaten sollten vorhandene Daten aus einer früheren Übergabe gelöscht werden.

Aktuelle Generation: 01.

Eigenschaften

- AuftragID: Eindeutige AuftragsID, wird vom System als fortlaufende Nummer vergeben.
- BelegDatum: Belegdatum, voraussichtliches Rechnungsdatum
- Bemerkung: Buchungstext, Vorgangstext
- EigenBrutto: Bruttobetrag in Eigenwährung
- EigenSteuer: Steuerbetrag in Eigenwährung
- Einheit: Einheit zum Kurs, falls nicht 1
- FremdBrutto: Bruttobetrag in Fremdwährung
- FremdID: Schlüssel im Fremdsystem, z.B. Bestellnummer, evtl. einschließlich Positionsnummer.
- FremdSteuer: Steuerbetrag in Fremdwährung
- Herkunft: Herkunfts-Kennung Ihrer Anwendung
- Konto: Personenkonto
- Kurs: Umrechnungskurs
- LinkInfo: Pfad zum PDF mit dem Auftrag
- NettoFaellig: Fälligkeitsdatum
- SteuerFaellig: Fälligkeit der Umsatzsteuer
- SteuerProzent: Umsatzsteuer-Prozentsatz
- Waehrung: Fremdwährung
- Zahlziel: Zahlungsziel

Methoden

Es sind nur die Funktionen Insert und Delete vorgesehen, kein Update.

AuswertungskreisCollection

Collection der Auswertungskreise in einem Mandant mit mehreren Buchungskreisen.

Direktzugriff: Nummer des Auswertungskreises.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- GetBezByKey: Holt die Bezeichnung über die Nummer. Stattdessen kann auch der Auswertungskreis gelesen und auf die Eigenschaft Bezeichnung zugegriffen werden.

Auswertungskreis

Daten eines Auswertungskreises in einem Mandant mit mehreren Buchungskreisen.

Die Auswertungskreise müssen in der Fibu gepflegt und mit Buchungskreiszuordnungen und Benutzerrechten versehen werden. Sie können gelesen werden, um Daten für einen Auswertungskreis abrufen zu können.

Aktuelle Generation: 01.

Eigenschaften

- AuswertungskreisNr: Auswertungskreisnummer
- Bezeichnung

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

AZDebitorCollection

Collection der geleisteten Anzahlungen für einen Anzahlungsvorfall eines Debitoren.

Direktzugriff: Interne Belegnummer BelegRefID

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

AZDebitor

Daten einer von einem Debitor erhaltenen Anzahlung in einem Anzahlungsvorfall.

Buchungen im Zuge von Anzahlungsvorfällen bestehen aus Rechnungen (Abschlagsrechnungen und Schlussrechnungen), die im ERP-System entstehen, und Zahlungen, die in der Fibu entstehen. Zur Erstellung der Schlussrechnung möchte das ERP-System ggf. die geleisteten Anzahlungen aus der Fibu abfragen. Diese werden nur zum Lesen zur Verfügung gestellt.

Aktuelle Generation: 01.

Eigenschaften

- aHWBetrag: Betrag der Buchung in alter Hauswährung (Währung vor Umstellung auf den Euro). Bei nicht-EWS Mandanten wird hier der Betrag in Mandantenwährung zurückgegeben.
- aHWSteuer: siehe aHWBetrag nur der Steuerbetrag der Buchung.

- BelegDatum
- BelegNr
- BelegRefID: (Interne fortlaufende) Buchungsnummer
- Betrag: Betrag in Belegwährung
- EigenBetrag: Betrag in Mandantenwährung
- EigenSteuer: Steuerbetrag in Mandantenwährung
- EuroBetrag: Betrag in Euro
- EuroSteuer: Steuerbetrag in Euro
- Kennung: Kennung der Buchungsart im Anzahlungsverfahren (Enum)
- OPBelegNr: Rechnungsnummer des OPs
- Steuerbetrag: Umsatzsteuer-Betrag
- SteuerSatz: Steuersatz eines evtl. Umsatzsteuer-Betrags
- VerweisRefID: Nummer des Anzehlungsvorfalls
- Zusatz: immer 1

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

BelegzeileCollection

Collection von Belegzeilen, den Zeilen von verbuchten Buchungen.

Direktzugriff: Kombiniertes Schlüssel aus Buchungsnummer und Zeilennummer (BelegzeileCollection03.RefID).

Aktuelle Generation: 03.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- GetBelegzeile: Holen einer einzelnen Belegzeile anhand Buchungsnummer und Zeilennummer.
- FilterAuftragID: Filtern nach einer AuftragsID.
- FilterAuftragIDAbBis: Einschränken der Auswahl nach einem Bereich von Auftragsnummern.
- FilterBelegDatum: Einschränken der Auswahl nach Belegdatum von und bis. Beide Werte sind optional.
- FilterPeriodeJahr: Einschränken nach einer bestimmten Periode und Jahr. Beide Werte sind optional. Wird das Jahr jedoch weggelassen werden alle Buchungen einer Periode über mehrerer Jahre geholt.
- FilterBelegRefIDVomBis: Einschränken der Auswahl nach einem Bereich von Buchungsnummern BelegRefID. Beide Werte sind optional.

- FilterBuchungskreisNr: Einschränken der Auswahl nach der Buchungskreisnummer.
- FilterPeriodeJahr: Einschränken auf die Belegzeilen einer Buchungsperiode (relative Periode).

Die Filter können war kombiniert werden, nach dem Abruf von Daten aber nicht mehr zurückgesetzt werden.

Belegzeile

Daten einer Belegzeile, der Teilbuchung auf ein Konto innerhalb eines verbuchten Buchungsbelegs.

Die Belegzeilen sind zum Lesen der verbuchten Belege gedacht. Neue Buchungen werden mit dem Objekt Stapelbuchung in den Buchungsstapel gestellt.

Aktuelle Generation: 03.

Eigenschaften

- aHWBetrag: Betrag der Buchung in alter Hauswährung (Währung vor Umstellung auf den Euro). Bei nicht-EWS Mandanten wird hier der Betrag in Mandantenwährung zurückgegeben.
- aHWSteuer: siehe aHWBetrag nur der Steuerbetrag der Buchung.
- AuftragID: AuftragID der Belegzeile (aus dem Belegkopf)
- BelegDatum
- BelegNummer
- BelegRefID: (Interne fortlaufende) Buchungsnummer
- Betrag: Betrag in Belegwährung
- Buchungsart: Enum der Belegart, Standard ist OFCKBuchungsart. baLfdBuchung
- BuchungsDatum: Erfassungdatum der Buchung
- BuchungskreisNr: Buchungskreisnummer
- BuchungsText
- EigenBetrag: Betrag in Mandantenwährung
- EigenSteuer: Steuerbetrag in Mandantenwährung
- EuroBetrag: Betrag in Euro
- EuroSteuer: Steuerbetrag in Euro
- Gegenkonto: Gegenkonto (informativ)
- IsSoll: auf welcher Seite wurde das Konto bebucht. Falls TRUE auf Sollseite
- Jahr: absolutes Jahr in das gebucht wurde.
- KontoArt: Art des Kontos in dieser Zeile (Enum)
- KontoNr
- KoreBelege: Falls Kore eingerichtet ist wird ein KoreBelege-Collection zurückgegeben mit den Kostenbuchungen. Falls Kore nicht eingerichtet ist kommt der Fehler ITF_E_NOKORE.
- Periode: Absolute Periode in die gebucht wurde (oder 13. Periode)
- RelPeriode: Relative Periode in die gebucht wurde (oder 13. Periode)

- SteuerSatz
- Waehrung
- ZeileNr: Belegzeilennummer in der Buchung

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

BestellungCollection

Collection der Zahlungsdaten einer Warenwirtschaft-Bestellung für die Liquiditätsplanung.

Direktzugriff: Die vom System vergebene ID der Bestellung.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

Bestellung

Zahlungsdaten einer Warenwirtschaft-Bestellung für die Liquiditätsplanung.

Zur Übergabe von Bestelldaten an die Liquiditätsplanung der Chefinfo (falls nicht ein View in das ERP-System genutzt wird). Die Verwendung dieser Funktion setzt voraus, dass eine Tabelle „cif_li_bestellung“ vorhanden ist, vgl. Schnittstellendokumentation zur Liquiditätsplanung. Informationen zur Übergabe der einzelnen Felder siehe ebenfalls Schnittstellendokumentation zur Liquiditätsplanung.

Vor der Übergabe der Bestelldaten sollten vorhandene Daten aus einer früheren Übergabe gelöscht werden.

Aktuelle Generation: 01.

Eigenschaften

- BestellungID: Eindeutige BestellungsID, wird vom System als fortlaufende Nummer vergeben.
- BelegDatum: Belegdatum, voraussichtliches Rechnungsdatum
- Bemerkung: Buchungstext, Vorgangstext
- EigenBrutto: Bruttobetrag in Eigenwährung
- EigenSteuer: Steuerbetrag in Eigenwährung
- Einheit: Einheit zum Kurs, falls nicht 1
- FremdBrutto: Bruttobetrag in Fremdwährung

- FremdID: Schlüssel im Fremdsystem, z.B. Bestellnummer, evtl. einschließlich Positionsnummer.
- FremdSteuer: Steuerbetrag in Fremdwährung
- Herkunft: Herkunfts-Kennung Ihrer Anwendung
- Konto: Personenkonto
- Kurs: Umrechnungskurs
- LinkInfo: Pfad zum PDF mit der Bestellung
- NettoFaellig: Fälligkeitsdatum
- SteuerFaellig: Fälligkeit der Umsatzsteuer
- SteuerProzent: Umsatzsteuer-Prozentsatz
- Waehrung: Fremdwährung
- Zahlziel: Zahlungsziel

Methoden

Es sind nur die Funktionen Insert und Delete vorgesehen, kein Update.

BuchungskreisCollection

Collection von Buchungskreisen, mit denen mehrere Rechnungslegungen organisiert werden.

Direktzugriff: Nummer des Buchungskreises.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- GetBezByKey: Holt die Bezeichnung über die Buchungskreis-Nummer

Buchungskreis

Daten eines Buchungskreises, mit dem mehrere Rechnungslegungen organisiert werden.

Buchungskreise müssen in der Fibu angelegt werden. Ihre Bezeichnung kann über die Schnittstelle geändert werden.

Aktuelle Generation: 01.

Eigenschaften

- BuchungskreisNr: Buchungskreisnummer
- Bezeichnung

- Nebenbuchungskreis: Handelt es sich um einen nicht führenden (Neben-)Buchungskreis?
- Gewinnkonto: Steuerliches Konto für den Gewinnvortrag.
- Saldovortragkonto: Gegenkonto für die Saldovorträge.
- BuchungssperreBis: Datum, bis zu dem ggf. eine Buchungssperre gesetzt ist.
- GesperrtFuerAnbu: Der Buchungskreis ist für die Anlagenbuchhaltung gesperrt.
- AnbuAbschlussAb: Der Buchungskreis ist für die Anlagenbuchhaltung offen ab (Datum).

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

CifPlanzahlenCollection

Collection der Planzahlen für benutzerdefinierte Fibu-Auswertungen im Modul Chefinfo.

Die Planzahlen sind in Szenarien organisiert. Pro Jahr, Konto, Szenario und Buchungskreis kann es einen Datensatz geben mit den Planzahlen für die Perioden des Jahres, in Mandantenwährung und Euro.

Direktzugriff: Der Schlüssel besteht aus mehreren Feldern. Dafür wird das Interface ICifPlanzahlenKey zur Verfügung gestellt.

Aktuelle Generation: 01

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterBuchungskreis: Filtern nach einem anderen als dem Standardbuchungskreis 1.
- FilterJahr: Filtern nach dem Jahr der Planung (Kalenderjahr).
- FilterKonto: Filtern nach einem Konto
- FilterSzenario: Filtern nach den Werten in einem Szenario.

CifPlanzahlen

Daten zu Planzahlen für benutzerdefinierte Fibu-Auswertungen im Modul Chefinfo.

Planzahlen können angelegt und geändert, nicht jedoch gelöscht werden. Setzen Sie die Planzahlen ggf. auf 0 zurück.

Aktuelle Generation: 01.

Eigenschaften

- Szenario: Nummer des Szenarios.
- Buchungskreis: Nummer des Buchungskreises (Standard ist 1).

- Jahr
- Konto

Methoden

- **GetPlanzahl:** Abfragen von Planzahlen für eine bestimmte Periode. Gültige Perioden sind 1 (Januar) bis 12 (Dezember). Es werden die Werte in Mandantenwährung und in Euro zurück gegeben.
- **SetPlanzahl:** Setzen von Planzahlen in Mandantenwährung und in Euro.

FibuAufbauZeileCollection

Collection der Zeilen verschiedener Aufbauten:

- Zeilen von benutzerdefinierten Reports der Kore
- Zeilen von Kostenstellen- oder Kostenträger-Bereichsaufbauten
- Zeilen von benutzerdefinierten Reports der Chefinfo

Die Aufbauten können nur gelesen werden.

Direktzugriff: Schlüssel bestehend aus Aufbau-ID und Index (Zeilennummer im Aufbau).

Aktuelle Generation: 01

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- **FilterAufbauID(int aufbauID):** Filtert auf die Zeilen eines Aufbaus. Die Aufbau-IDs sind: negativ = Kore-Bereichsaufbau, positiv = Kore-Reportaufbau, positiv > 10000 = Chefinfo-Reportaufbau.

FibuAufbauZeile

Daten einer Zeile in einem Aufbau wie unter FibuAufbauZeileCollection genannt.

Aktuelle Generation: 01.

Eigenschaften

- **AufbauID:** Nummer des Aufbaus. Negativ = Kore-Bereichsaufbau, positiv = Kore-Reportaufbau, positiv > 10000 = Chefinfo-Reportaufbau.
- **Ebene:** Ebene im Aufbau.
- **Index:** Laufende Nummer im Aufbau. Negative Indexnummern sind noch nicht zugeordnete Zeilen / Konten.
- **Kennung**
- **KontoNr:** Nummer des Kontos bzw. der Kostenstelle, bei Überschriften leer

- Text: Zeilentext. Entweder der manuell eingegebene Überschriftentext oder die Kontobezeichnung oder die Formel.
- ZeileArt: Art der Zeile je nach Reportaufbau.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

FremdWaehrungCollection

Collection von Fremdwährungen mit Zugriff auf ihre Kurse.

Direktzugriff: Währungskürzel. Zurückgegeben wird ein Objekt FremdWaehrung.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

FremdWaehrung

Daten einer Fremdwährung, das ist nur der Währungsschlüssel selbst. Mit Zugriff auf die Kurse.

Die Währung muss in der Fibu angelegt werden. Sie kann hier gelesen werden, aber nicht angelegt oder geändert werden. Es ist jedoch ein Zugriff auf die Kurse möglich.

Aktuelle Generation: 01.

Eigenschaften

- Kurse: Collection der Kurse für die Währung, siehe KursCollection.
- Waehrungskuerzel: Währungskürzel der Währung (nur Lesen).

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

GruppierungCollection

Collection der Kontogruppen bzw. Kostenartengruppen.

Direktzugriff: Kontonummer.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterGruppeKennung: Einschränkung auf die Kennung der Gruppierung (Enum). Zugriff auf Kontogruppen oder Kostenartengruppen.

Gruppierung

Daten einer Kontogruppe bzw. Kostenartengruppe.

Aktuelle Generation: 01.

Eigenschaften

- GruppeRefID: Interne Nummer der Gruppierung
- GruppeID: Kontonummer bzw. Kostenartennummer
- Bezeichnung: Bezeichnung der Gruppierung
- Kennung: Kennung der Gruppierung

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KontoListeCollection

Collection der Konten eines Mandanten in einer schlanken Listenform.

Direktzugriff: Index ist die Kontonummer. Zurückgegeben wird ein Objekt KontoListe.

Konten sind grundsätzlich alle Kontoarten: die Personenkonten Debitoren und Kreditoren sowie die Sachkonten (außerdem Einmalkonten und Einmaladressen).

Aktuelle Generation: 01. Ab Version 5.0.6.0.

Eigenschaften

- TopAnzahl: Die Liste kann auf eine gewünschte Anzahl Konten beschränkt werden (optional).

Methoden

Für alle Filter gilt: Der Filter muss vor dem ersten Zugriff auf die Collection-Daten aufgerufen werden oder RefreshCollData muss aufgerufen werden für ein erneutes Laden der Collection Daten. Kombinationen von Filtern sind möglich. Ob die Kombination sinnvoll ist, wird nicht kontrolliert.

- FilterKontoArt: Einschränkung nach bestimmten Kontenarten. Die möglichen Kontenarten zeigt ein Enum auf. Einige der Kontenarten sind Zusammenfassungen mehrerer tatsächlicher Kontenarten, diese dürfen hier nicht verwendet werden.

- **FilterName:** Einschränkung nach dem Kontoname. Der Inhalt wird für eine Like-Suche verwendet und kann Wildcard-Zeichen enthalten.
- **FilterKuerzel:** Einschränkung nach dem Kontokürzel. Der Inhalt wird für eine Like-Suche verwendet und kann Wildcard-Zeichen enthalten.
- **SortOrder:** Ruft die Ergebnismenge sortiert nach einem Kriterium ab (Enum, Standard: Kontonummer).

KontoListe

Die Daten eines einzelnen Konto-Eintrags in einer Kontenliste. Enthält nur die Überblicks-Informationen zu einem Konto.

Aktuelle Generation: 01. Ab Version 5.0.6.0.

Eigenschaften

- **KontoKuerzel:** Kurzbezeichnung des Kontos. Nur Grossbuchstaben zugelassen. Übergebene Strings werden automatisch umgewandelt.
- **KontoName**
- **KontoNummer:** Eindeutige Kontonummer (das ist auch der Zugriffs-Index). Einmaladressen haben eine negative, vom System vergebene Kontonummer.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KoreBelegCollection

Die Collection, um verbuchte Kore-Buchungen auszulesen. Enthält sowohl Primärkostenbuchungen aus der Fibu als auch Sekundärkostenbuchungen aus der Kore.

Direktzugriff: Der Schlüssel besteht aus mehreren Feldern, wie im Interface IKoreBelegKey zusammen gefasst.

Aktuelle Generation: 01

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- **FilterBelegDatum:** Einschränken nach Belegdatum von und bis.
- **FilterBewegArt:** Filterung auf eine Bewegungsart (Enum mit Kosten, Leistungen, Umlagen usw.) möglich.
- **FilterKoreBeleg:** Filterung auf eine Belegzeile möglich, bestehend aus Beleg-Refid und Zeilennummer. Durch Kore-Splitting können das mehrere Zeilen (Laufende Nummer innerhalb der Zeile) sein.
- **FilterPeriodeJahr:** Filter nach Periode und Jahr möglich.

KoreBeleg

Daten einer verbuchten Kore-Buchungszeile.

Kann sowohl eine Primärkostenbuchung aus der Fibu als auch eine Sekundärkostenbuchung aus der Kore sein oder Umlagebuchung sein. Eine Zeile ist identifiziert durch die Beleg-Refid, Zeilennummer und laufende Nummer.

Aktuelle Generation: 01.

Eigenschaften

- BelegDatum:
- BelegNummer:
- BelegRefID:
- Betrag:
- BuchungsText:
- Einheit:
- FixKosten: Sind es Fixkosten (im Gegensatz zu variablen Kosten)?
- Jahr:
- Kostenart:
- Kostenstelle:
- Menge:
- Periode:
- Waehrung:
- ZeileLfdNr:
- ZeilenRefID:

Methoden

- Kostentraeger: Über den Index kann auf die Kostenträger 1 bis 5 zugegriffen werden, jedoch 0-basiert (Kostenträger 1 = 0).

KoreBuchungCollection

Die Collection, um Sekundärkostenbuchungen in einen Stapel der Kostenrechnung zu übergeben.

Sekundärkostenstapel müssen in der Kore angelegt werden, Fibu-Stapel sind dafür nicht vorgesehen. Primärkostenbuchungen dagegen müssen zusammen mit den Fibu-Buchungen in einen Fibu-Stapel gestellt werden.

Direktzugriff: Nicht vorgesehen. Beim Zugriff auf die Eigenschaft KoreBuchungen des Mandanten ist die Stapelnummer anzugeben.

Aktuelle Generation: 01

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KoreBuchung

Buchungsdaten zum Einfügen von Sekundärkostenbuchungen in einen Stapel der Kostenrechnung.

Aktuelle Generation: 01.

Eigenschaften

- BelegDatum
- BelegNr
- Buchungskreis
- Einheit: Bezugseinheit bei Buchung mit Mengen (Optional bzw. Pflicht je nach Systemeinstellung)
- GeschaeftsJahr (optional): Bei periodenbestimmten Buchen das Geschäftsjahr falls abweichend zum Belegdatum.
- Kostenart: keine Primärkostenarten zulässig, nur Sekundärkostenarten.
- Periode (optional): Bei periodenbestimmten Buchen die Periode in die gebucht wird.
- StapelNr: Stapel Nummer auf den gebucht werden soll. Normalerweise schon gesetzt. Kann jedoch geändert werden.

Methoden

Von den Standardmethoden steht nur die Methode Insert zur Verfügung. Zuvor müssen die Eigenschaften gesetzt sein und mit AddInfo die Buchungszeilen angefügt sein.

- AddInfo: Zum Hinzufügen einzelner Korebuchungszeilen. Parameter:
 - EigenBetrag: Betrag in Mandantenwährung
 - Kostenstelle: Kostenstelle die bebucht wird
 - fvFix: Kennung (Enum) ob Fixkosten (=Default) oder variable Kosten. Nur relevant, wenn die Kostenrechnung mit Unterscheidung zwischen fixen/variablen Kosten konfiguriert ist.
 - Menge: falls die Buchung neben dem Betrag auch mit Menge gebucht werden soll (Einheit muss dann gesetzt sein, siehe Eigenschaften). Falls die Kostenart mit Mengenpflicht in der Kostenrechnung definiert ist, dann Pflichtangabe.
 - KoTr1: Kostenträger 1. Je nach Einrichtung der Kostenrechnung
 - KoTr2: Kostenträger 2
 - KoTr3
 - KoTr4

- KoTr5
- Buchungstext

KorePlanSchabloneCollection

Collection von Schablonen für die Verteilung von Planzahlen in der Kostenrechnung.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KorePlanSchablone

Daten einer Schablone für die Verteilung von Planzahlen in der Kostenrechnung.

Die Daten können nur gelesen werden.

Index: Schablonen-Nummer.

Eigenschaften

- isGlobal: Handelt es sich um eine globale Schablone?
- Name: Bezeichnung der Kostenschablone
- Nummer: Planschablonen-Nummer

Methoden

- GetPeriodenProzente: Abfragen der Prozentverteilung für eine Periode. Parameter:
 - perAbsolut: Absolute Periode für die die Prozentverteilung abgefragt werden soll. Gültig sind nur Periode 1 für Januar bis Periode 12 für Dezember.

KorePlanwerteCollection

Collection der Planwerte in der Kostenrechnung.

Planwerte können bis zu einem Plananpassungszeitpunkt gesetzt und geändert werden, danach nur noch Prognosewerte.

Direktzugriff: Der Schlüssel besteht aus mehreren Feldern. Dafür wird das Interface IKorePlanwertKey zur Verfügung gestellt.

Aktuelle Generation: 01

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterArt: Planstellen-Art nach der gefiltert werden soll. Enum mit diesen Möglichkeiten:
 - 0 = filtern nach Kostenstelle
 - 1-5 = filtern nach einem der Kostenträger 1 bis 5
- FilterJahr: Filtern nach dem Jahr der Planung (Kalenderjahr).
- FilterKostenart: Filtern nach einer Kostenart
- FilterKostenstelleKostentraeger: ID für eine gesuchte Kostenstelle oder Träger
- GueltigePlanungsPeriode: Hilfsfunktion mit der überprüft werden kann ob es zulässig ist eine bestimmte Periode/Jahr Kombination zu planen.

KorePlanwerte

Planwerte in der Kostenrechnung.

Die Planwerte für alle Perioden eines Jahres für eine Kombination Kostenstelle / Kostenart oder Kostenträger / Kostenart stellen einen Planwert-Satz dar.

Die Planwerte können eingefügt und geändert, jedoch nicht gelöscht werden.

Aktuelle Generation: 01

Eigenschaften

- Bemerkung: Bemerkungstext für einen Planwerte-Satz.
- PlanstellenArt: Art des Kostenbegriffs Kostenstelle / Kostenträger 1 bis 5.
- KostenArt
- KoStID_KoTrID: Kostenstelle oder Kostenträger je nach PlanstellenArt.
- Jahr

Methoden

- GetPlanwert: Abfragen von Planwerten für eine bestimmte Periode. Gültige Perioden sind 1 (Januar) bis 12 (Dezember). Es werden die Werte für Fixkosten und variable Kosten zurück gegeben.
- SetPlanwert: Setzen von Planwerten. Diese Funktion ist nur gültig für Perioden im nächsten abzuschließenden Geschäftsjahr falls das Planungsende-Datum noch nicht überschritten ist. Ebenfalls nicht geändert werden dürfen in der Kore bereits abgeschlossene Monate.

KorePrognosewerteCollection

Collection der Prognosewerte in der Kostenrechnung.

Nach dem Plananpassungszeitpunkt können keine Planwerte mehr gesetzt und geändert werden, stattdessen aber Prognosewerte. Die Prognosewerte werden mit Planwerten automatisch eingefügt, sie können nur geändert werden.

Direktzugriff: Der Schlüssel besteht aus mehreren Feldern. Dafür wird das Interface IKorePrognosewerteKey zur Verfügung gestellt.

Aktuelle Generation: 01

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterArt: Planstellen-Art nach der gefiltert werden soll. Enum mit diesen Möglichkeiten:
 - 0 = filtern nach Kostenstelle
 - 1-5 = filtern nach einem der Kostenträger 1 bis 5
- FilterJahr: Filtern nach dem Jahr der Planung (Kalenderjahr).
- FilterKostenart: Filtern nach einer Kostenart
- FilterKostenstelleKostentraeger: ID für eine gesuchte Kostenstelle oder Träger
- GueltigePrognosePeriode: Hilfsfunktion mit der überprüft werden kann ob es zulässig ist die Prognosewert in einer bestimmte Periode/Jahr Kombination zu bearbeiten.

KorePrognosewerte

Prognosewerte in der Kostenrechnung.

Die Prognosewerte für alle Perioden eines Jahres für eine Kombination Kostenstelle / Kostenart oder Kostenträger / Kostenart stellen einen Prognosewert-Satz dar.

Die Prognosewerte können nur geändert, jedoch nicht gelöscht werden. Sie werden automatisch mit den Planwerten eingefügt.

Aktuelle Generation: 01

Eigenschaften

- Bemerkung: Bemerkungstext für einen Planwerte-Satz.
- PlanstellenArt: Art des Kostenbegriffs Kostenstelle / Kostenträger 1 bis 5.
- KostenArt
- KoStID_KoTrID: Kostenstelle oder Kostenträger je nach PlanstellenArt.
- Jahr

Methoden

- GetPrognosewert: Abfragen von Prognosewerten für eine bestimmte Periode. Gültige Perioden sind 1 (Januar) bis 12 (Dezember). Es werden die Werte für Fixkosten und variable Kosten zurück gegeben.
- SetPrognosewert: Setzen von Prognosewerten. Diese Funktion ist nur gültig für Perioden im nächsten abzuschließenden Geschäftsjahr nach dem Planungsende-Datum. Ebenfalls nicht geändert werden dürfen in der Kore bereits abgeschlossene Monate.

KoSchabloneCollection

Collection von Schablonen (Kopfdaten) für die Verteilung in der Kostenrechnung.

Direktzugriff: Kostenschablonen-Nummer.

Von den Kostenschablonen können die Kopfdaten gelesen werden. Die Pflege muss in der Kore erfolgen.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KoSchablone

Kopfdaten einer Kostenschablone für die Verteilung in der Kostenrechnung.

Die Verteilungsdaten einer Kostenschablone können nicht ausgelesen werden.

Aktuelle Generation: 02.

Eigenschaften

- Einheit: Bezugseinheit für die Verteilung.
- GesamtBetrag: Gesamtbetrag der Verteilung.
- GueltigVon: Gültigkeitszeitraum von/bis
- GueltigBis: Gültigkeitszeitraum von/bis
- Kuerzel: Kürzel der Kostenschablone
- Nummer: Alphanumerische eindeutige Kennung der Kostenschablone
- Name: Bezeichnung der Kostenschablone

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KostenartCollection

Collection der Kostenarten.

Direktzugriff: Kostenart-Nummer.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterKostenart: Es kann nach Primär- oder Sekundärkostenart eingeschränkt werden (Enum).

Kostenart

Daten einer Kostenart.

Kostenarten können Primärkostenarten sein (verbunden mit Fibu-Konten), Sekundärkostenarten (nur Kore), Leistungsarten (nur Kore) oder Umlagekostenarten (nur Kore für die Umlage).

Aktuelle Generation: 02.

Eigenschaften

- Einheit: Bezugseinheit. Sie wird auch dann gespeichert, wenn sie in den Kore-Einstellungen nicht angelegt ist.
- Erloesart: Handelt es sich um eine Erlös-Kostenart ? Nur für Primärkostenarten.
- Fixkostenart: Handelt es sich um eine Fixkosten-Kostenart ?
- GruppeRefID: ID (nicht Nummer) der Kostenarten-Gruppe.
- KoArtArt: Art der Kostenart (Enum).
- KoArtKuerzel: Kürzel für die Kostenart.
- KoArtName: Name der Kostenart.
- KoArtNummer: Kostenarten-Nummer.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KostenstelleCollection

Collection der Kostenstellen.

Direktzugriff: Kostenstellen-"Nummer", die alphanumerisch ist bzw. sein kann.

Zurückgegeben wird ein Objekt Kostenstelle.

In der Kore kann festgelegt werden, ob die Kostenstellen rein numerisch geführt werden sollen. OFCKS.NET behandelt die Kostenstelle jedoch immer alphanumerisch.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- ExistsKoStKuerzel: Überprüft ob das übergebene Kürzel (nicht Nummer!) bereits existiert

- Filter: Aufruf der Methode muss vor dem ersten Zugriff auf die Daten der Collection erfolgen oder RefreshCollData muss aufgerufen werden. Als Parameter werden KoStNummer als Angabe der Bereiche Von und (optional) Bis erwartet.
- SortOrder: Es kann eine Sortierung angegeben werden (Standard: nach KoStNummer). Enum.

Kostenstelle

Daten einer Kostenstelle.

Aktuelle Generation: 02.

Eigenschaften

- Budget:
- GueltigBis: Gültigkeitszeitraum von/bis
- GueltigVon: Gültigkeitszeitraum von/bis
- KoStArt: Kostenstellenart (Enum). Hauptkostenstellen werden von Hilfskostenstellen und allgemeinen Kostenstellen unterschieden.
- KoStKuerzel: Kurzbezeichnung
- KoStName: Bezeichnung der Kostenstelle
- KoStNummer: alphanumerische eindeutige Kennung der Kostenstelle
- Leiter: Name des Leiters der Kostenstelle
- Saldo: Saldo der Kostenstelle (nur Lesen)
- SaldoReset: Kennung ob der Saldo beim Jahreswechsel zurückgesetzt wird. 1 = zurücksetzen, 0 = beibehalten

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KostentraegerCollection

Collection der Kostenträger in der Tabelle kr_kostentraeger.

Direktzugriff: Kostenträger-"Nummer", die alphanumerisch ist bzw. sein kann. Zurückgegeben wird ein Objekt Kostenstelle.

In der Kore kann für bis zu 5 Kostenträger-Dimensionen festgelegt werden, in welcher Tabelle die Daten dafür gehalten werden. OFCKS.NET erlaubt die Stammdatenpflege jedoch nur auf der Tabelle kr_kostentraeger unabhängig von den Kostenträger-Definitionen. Andere Tabellen liegen in der Datenhoheit anderer Anwendungen.

Aktuelle Generation: 02.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- ExistsKoTrKuerzel: Überprüft ob das übergeben Kürzel (nicht Nummer!) bereits existiert.
- Filter: Aufruf der Methode muss vor dem ersten Zugriff auf die Daten der Collection erfolgen oder RefreshCollData muss aufgerufen werden. Als Parameter werden KoTrNummer als Angabe der Bereiche Von und (optional) Bis erwartet.
- FilterLike: Aufruf der Methode muss vor dem ersten Zugriff auf die Daten der Collection erfolgen oder RefreshCollData muss aufgerufen werden. Als Parameter wird eine KoTrNummer als Angabe für eine Like-Einschränkung (mit Wildcards) erwartet.
- SortOrder: Es kann eine Sortierung angegeben werden (Enum. Standard: nach KoTrNummer).

Kostentraeger

Daten eines Kostenträgers in der Tabelle kr_kostentraeger. Anders definierte Kostenträgerkeise sind nicht zugänglich.

Aktuelle Generation: 02.

Eigenschaften

- KoTrNummer: alphanumerische eindeutige Kennung des Kostenträgers
- KoTrName: Bezeichnung
- KoStKuerzel: Kurzbezeichnung
- Budget:
- GueltigVon: Gültigkeitszeitraum von/bis
- GueltigBis: Gültigkeitszeitraum von/bis
- Leiter: Zuständiger für den Kostenträger
- Saldo: Saldo des Kostenträgers (nur Lesen)
- SaldoReset: Kennung ob der Saldo beim Jahreswechsel zurückgesetzt wird. 1 = zurücksetzen, 0 = beibehalten

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

KursCollection

Collection der Kurse einer Fremdwährung.

Direktzugriff: verwenden Sie die Methode GetKurs.

Neue Kurse können mit der Methode NewKurs angelegt werden. Vorhandene Kurse können nicht geändert werden. Die Währung selbst muss in der Fibu angelegt werden.

Die Kurse (einer Währung) können direkt vom Mandant aus angesprochen werden oder über die Objekte FremdWaehrungCollection -> FremdWaehrung (siehe dort).

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- **GetKurs:** holt zu einem bestimmten Datum den gültigen Kurs. Anzugeben sind Währung und Kurstag, ggf. erfolgt ein Rückgriff auf den Monats-Durchschnittskurs. Zurückgegeben wird ein Objekt des Typs Kurs.

Diese Methode war im früheren OFCK anders definiert. Sie lieferte den Kurs als Rückgabewert und die weiteren Eigenschaften als out-Parameter.

- **NewKurs:** Anlage eines neuen Kurses. Nicht möglich bei einer eEvolution Version, die Pflege erfolgt in diesem Fall im ERP-System. Die Währung ergibt sich durch die Collection. Zu den Parametern siehe die Erklärungen beim Kurs-Objekt.

Kurs

Daten eines Fremdwährungs-Tageskurses oder Monatskurses.

Aktuelle Generation: 01.

Eigenschaften

- **BezugsWaehrung:** Frühere Hauswährung, falls Währung eines Euro-Landes.
- **Einheit:** ggf. 10er Potenz, um große Kurszahlen zu vermeiden.
- **GuelteigDatum:** der Tag, an dem der Kurs gilt.
- **IsMengennotiz:** Handelt es sich bei der Kursangabe um Mengen- oder Preisnotiz?
- **IsMonatsDurchschnitt:** Handelt es sich um einen Monats-Durchschnittskurs?
- **Kuerzel:** Währungskürzel.
- **Kurs:** Der Umrechnungskurs.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

OPDebVorgangCollection

Collection der Vorgänge zu einem OP auf ein Personenkonto Debitor.

Direktzugriff: keiner, kein eindeutiger Schlüssel.

Die Vorgänge zu Offenen Posten können nur gelesen werden.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterVorgangsart: Filter nach der Art der Vorgänge.
- FilterVorgangDatum: Einschränken nach dem Datum des Vorgangs. Datum Von und Bis.

OPDebVorgang

Daten eines Vorgangs zu einem OP auf ein Personenkonto Debitor.

Vorgänge sind Rechnungsbuchungszeilen, Zahlungen, aber auch Mahnungen, Verrechnungen usw. In der Anzeige der Vorgänge in Opos werden die Vorgangs-Kürzel auf Doppelklick dargestellt.

Aktuelle Generation: 03.

Eigenschaften

- AuftragID: Auftragsnummer des ERP-Systems.
- Belegdatum: (Fibu-)Belegdatum
- BelegRefID: RefID des zugehörigen Buchungsbeleges.
- BelegZeileRefID: Zeilennummer der Zeile des zugehörigen Buchungsbeleges.
- Bemerkung: Buchungstext
- EigenNettobetrag: Nettobetrag des Vorgangs in Eigenwährung
- EigenSteuerbetrag: Steuerbetrag des Vorgangs in Eigenwährung
- ERHAZKonto: Konto für erhaltene Anzahlungen
- Gegenkonto: Gegenkonto laut der zugehörigen Buchungszeile.
- Geschaeftsjahr: Geschäftsjahr in dem der Vorgang erfolgte
- NNEAZKonto: Konto der noch nicht erhaltenen Anzahlungen (Abschlagsrechnung).
- OPRefID: RefID des OPs
- Personenkonto: Personenkonto zu dem der OP gehört.
- RelativePeriode: relative Periode in welche der OP gehört.
- Skontierfaehig: Ob der OP skontierfähig ist (ungleich 0) oder nicht (0)
- Skontokonto: Konto, auf das ggf. Skonto zu buchen ist.
- Steuerkonto: Konto für die Umsatzsteuer bzw. Vorsteuer.
- SteuerSatz: Prozentsatz der Umsatzsteuer.
- User: Benutzer der den Vorgang gebucht oder als letzter geändert hat.
- Vorgangsdatum: (Kalender)Datum an welchem der Vorgang erfolgte.
- Vorgangsart: Art des OP Vorgangs. Hier wird das Vorgangskürzel zurückgegeben

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

OPKredVorgangCollection

Collection der Vorgänge zu einem OP auf ein Personenkonto Kreditor.

Direktzugriff: keiner, kein eindeutiger Schlüssel.

Die Vorgänge zu Offenen Posten können nur gelesen werden.

Aktuelle Generation: 02.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterAuftragID: Filter auf eine Auftragsnummer.
- FilterVorgangsart: Filter nach der Art der Vorgänge.
- FilterVorgangDatum: Einschränken nach dem Datum des Vorgangs. Datum Von und Bis.

OPKredVorgang

Daten eines Vorgangs zu einem OP auf ein Personenkonto Kreditor.

Vorgänge sind Rechnungsbuchungszeilen, Zahlungen, aber auch z.B. Verrechnungen usw. In der Anzeige der Vorgänge in Opos werden die Vorgangs-Kürzel auf Doppelklick dargestellt.

Aktuelle Generation: 03.

Eigenschaften

- AuftragID: Auftragsnummer des ERP-Systems.
- Belegdatum: (Fibu-)Belegdatum
- BelegRefID: RefID des zugehörigen Buchungsbeleges.
- BelegZeileRefID: Zeilennummer der Zeile des zugehörigen Buchungsbeleges.
- Bemerkung: Buchungstext
- EGKonto: Das Steuer-Gegenkonto bei Vorgängen mit VSt-/USt-Automatik.
- EigenNettobetrag: Nettobetrag des Vorgangs in Eigenwährung
- EigenSteuerbetrag: Steuerbetrag des Vorgangs in Eigenwährung
- GegenKonto: Gegenkonto laut der zugehörigen Buchungszeile.
- Geschaefstsjahr: Geschäftsjahr in dem der Vorgang erfolgte
- GTGAZKonto: Konto für geleistete Anzahlungen
- InventarNr: Inventarnummer des Anlagegutes falls eines beteiligt ist
- NNGAZKonto: Konto der noch nicht geleisteten Anzahlungen (Abschlagsrechnung).
- OPRefID: RefID des OPs
- Personenkonto: Personenkonto zu dem der OP gehört.
- RelativePeriode: relative Periode in welche der OP gehört.

- Skontierfaehig: Ob der OP skontierfähig ist (ungleich 0) oder nicht (0)
- Skontokonto: Konto, auf das ggf. Skonto zu buchen ist.
- Steuerkonto: Konto für die Umsatzsteuer bzw. Vorsteuer.
- SteuerSatz: Prozentsatz der Umsatzsteuer.
- UnterNr: Unternummer des Anlagegutes falls eines beteiligt ist
- User: Benutzer der den Vorgang gebucht oder als letzter geändert hat.
- Vorgangsdatum: (Kalender)Datum an welchem der Vorgang erfolgte.
- Vorgangsart: Art des OP Vorgangs. Hier wird das Vorgangskürzel zurückgegeben

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

OPostenDebCollection

Collection der Offenen Posten (aktuell noch offen oder bereits historisiert) eines Personenkontos Debitor.

Direktzugriff: OP-Nummer, i.d.R. die Buchungsnummer (OPRefID).

Die Offenen Posten entstehen beim Buchen und können gelesen werden.

Aktuelle Generation: 03.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterAuftragID: Filter nach einer AuftragID.
- FilterBelegDatum: Einschränken nach Belegdatum. Datum Von und Bis. Parameter sind optional.
- FilterBelegNr: Einschränken nach einer Belegnummer.
- FilterOPBelegNr: Einschränken nach einer Rechnungsnummer, auch OP-Belegnummer genannt (ab 5.0.7.3).
- FilterFaelligkeit: Einschränken nach Fälligkeitsdatum. Fälligkeitsdatum Von und Bis. Parameter sind optional.
- FilterGeloeschtDatum: Einschränken nach Löschdatum (OP-Ausgleich) von / bis.
- FilterKonto: Einschränken auf ein bestimmtes Konto.
- FilterKontenBereich: Einschränken auf einen Kontenbereich von / bis.
- InclusiveHistorie: Wenn gesetzt werden auch die bereits bezahlten OPs mit zurückgegeben.
- SortOrder: Sortierung möglich nach Belegdatum, Belegnummer, Kontonummer oder Fälligkeit (Enum).

OPostenDeb

Daten eines Offenen Posten (aktuell noch offen oder bereits historisiert) eines Personenkontos Debitor.

Die Offenen Posten sind grundsätzlich zum Lesen gedacht. Einige Felder können jedoch auch geändert werden.

Aktuelle Generation: 05.

Eigenschaften

- Anmerkung: Anmerkung zum Offenen Posten.
- Ausstand: offener Betrag in Belegwährung
- BelegDatum
- BelegNr: Belegnummer der Fibu
- BruttoBetrag: Gesamtbetrag des OP in OP-Währung
- BuchungskreisNr: Buchungskreis
- EigenAusstand: Ausstand des OPs in Mandantenwährung
- EigenBetrag: Bruttobetrag in Mandantenwährung
- HistorischSeit: Datum seitdem der OP bezahlt ist
- IsHistorisch: Handelt es sich um einen historischen oder aktiven OP?
- Kategorie
- KontoNr: Debitorenkonto
- MahnDatum: Datum der letzten Mahnung
- Mahnflags: Enum mit Mahnstop und anderen temporären Mahnzuständen.
- Mahnstufe: aktuelle Mahnstufe des OPs
- NettoDatum: Frist bis zu dem OP netto fällig ist, ab dann wird gemahnt
- OPBelegNr: Rechnungsnummer
- OPBemerkung: Buchungstext des OPs
- OPRefID: interne OP-Nummer
- OPTyp: Typ des OPs (Rechnung, Gutschrift, etc., Enum)
- OPVorgaenge: Zugriff auf die Vorgänge des OPs (OPDebVorgangCollecion).
- SkontoDatum1 / SkontoDatum2: Datum der ersten und zweiten Skontofrist. Im früheren OFCK indexierte Eigenschaft.
- SkontoProz1 / SkontoProz2: Skontoprozentsatz der ersten und zweiten Skontofälligkeit. Im früheren OFCK indexierte Eigenschaft.
- Sprache: Mahnsprache
- ValutaDatum
- Waehrung: Belegwährung
- WiedervorlageDatum: Datum der Wiedervorlage
- Zahlungssperre: Datum, vor dem der OP nicht bezahlt bzw. eingezogen werden darf.
- ZahlzielNr: Nummer des Zahlungszieles

Diese Eigenschaften können geändert werden: Mahnflags, Mahndatum, Mahnstufe, Nettodatum, Skontodatum 1 / 2, Sprache, Valutadatum, Wiedervorlagedatum, Zahlungssperre.

Das frühere OFCK kannte auch die Eigenschaft AuftragID. Diese wurde jedoch bereits seit SQL REWE Version 6 (Vorgänger von syska ProFI; der Wechsel von syska SQL REWE zu syska ProFI fand mit Version 10 statt) nicht mehr im OP bedient. Sie befindet sich jetzt im OP-Vorgang, die Filterung nach Auftragsnummer ist möglich.

Methoden

- GegenKonto: Gegenkonto je Steuersatz indexiert
- SkontoKonto: Konto auf das das Skonto gebucht wird je Steuersatz indexiert
- SteuerKonto: Konto auf das die Steuer gebucht wurde je Steuersatz indexiert
- SteuerSatz: Prozentsatz der Steuer je Steuersatz indexiert
- Update: Aktualisieren der oben genannten Eigenschaften.

Die Methoden SetOPMahndaten und SetOPZahlzielDatum des früheren OFCK gibt es nicht mehr. Setzen Sie die entsprechenden Eigenschaften und verwenden Sie die Methode Update.

OPostenKredCollection

Collection der Offenen Posten (aktuell noch offen oder bereits historisiert) eines Personenkontos Kreditor.

Direktzugriff: OP-Nummer, i.d.R. die Buchungsnummer (OPRefID).

Die Offenen Posten entstehen beim Buchen und können gelesen werden.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- FilterBelegDatum: Einschränken nach Belegdatum. Datum Von und Bis. Parameter sind optional.
- FilterBelegNr: Einschränken nach einer Belegnummer.
- FilterOPBelegNr: Einschränken nach einer Rechnungsnummer, auch OP-Belegnummer genannt (ab 5.0.7.3).
- FilterFaelligkeit: Einschränken nach Fälligkeitsdatum. Fälligkeitsdatum Von und Bis. Parameter sind optional.
- FilterKonto: Einschränken auf ein bestimmtes Konto.
- InclusiveHistorie: Wenn gesetzt werden auch die bereits bezahlten OPs mit zurückgegeben.
- SortOrder: Sortierung möglich nach Belegdatum, Belegnummer, Kontonummer oder Fälligkeit (Enum).

OPostenKred

Daten eines Offenen Posten (aktuell noch offen oder bereits historisiert) eines Personenkontos Kreditor.

Die Offenen Posten sind grundsätzlich zum Lesen gedacht. Einige Felder können jedoch auch geändert werden.

Aktuelle Generation: 04.

Eigenschaften

- Anmerkung: Anmerkung zum Offenen Posten.
- Ausstand: offener Betrag in Belegwährung
- BelegDatum
- BelegNr: Belegnummer der Fibu
- BruttoBetrag: Gesamtbetrag des OP in OP-Währung
- BuchungskreisNr: Buchungskreis
- EigenAusstand: Ausstand des OPs in Mandantenwährung
- EigenBetrag: Bruttobetrag in Mandantenwährung
- HistorischSeit: Datum seitdem der OP bezahlt ist
- IsHistorisch: Handelt es sich um einen historischen oder aktiven OP?
- Kategorie
- KontoNr: Debitorenkonto
- NettoDatum: Frist bis zu dem OP netto fällig ist, ab dann wird gemahnt
- OPBelegNr: Rechnungsnummer
- OPBemerkung: Buchungstext des OPs
- OPRefID: interne OP-Nummer
- OPTyp: Typ des OPs (Rechnung, Gutschrift, etc., Enum)
- OPVorgaenge: Zugriff auf die Vorgänge des OPs (OPKredVorgangCollecion).
- SkontoDatum1 / SkontoDatum2: Datum der ersten und zweiten Skontofrist. Im früheren OFCK indexierte Eigenschaft.
- SkontoProz1 / SkontoProz2: Skontoprozentsatz der ersten und zweiten Skontofälligkeit. Im früheren OFCK indexierte Eigenschaft.
- ValutaDatum
- Waehrung: Belegwährung
- ZahlungsSperre: Datum, vor dem der OP nicht bezahlt bzw. eingezogen werden darf.
- ZahlzielNr: Nummer des Zahlungszieles

Diese Eigenschaften können geändert werden: Nettodatum, Skontodatum 1 / 2, Valutadatum, Zahlungssperre.

Methoden

- Gegenkonto: Gegenkonto je Vorgang indexiert

- SkontoKonto: Konto auf das das Skonto gebucht wird je Vorgang indexiert
- SteuerKonto: Konto auf das die Steuer gebucht wurde je Vorgang indexiert
- SteuerSatz: Prozentsatz der Steuer je Vorgang indexiert
- Update: Aktualisieren der oben genannten Eigenschaften.

Die Methode SetOPZahlzielDatum des früheren OFCK gibt es nicht mehr. Setzen Sie die entsprechenden Eigenschaften und verwenden Sie die Methode Update.

RechnungCollection

Collection der erfassten Rechnungen im Rechnungseingangsbuch.

Direktzugriff: Index ist die interne laufende Nummer der Rechnung.

Das Zusammenspiel mit dem ERP-System ist so vorgesehen: Eingangsrechnungen werden im syska-ProFI-Modul REBU erfasst. Die Rechnungsprüfung findet im ERP statt, erstellt den Buchungssatz, übergibt diesen an die Fibu und löscht dabei die Rechnung im Rechnungseingangsbuch (logische Löschung).

Die Rechnungen können deshalb nur gelesen und "gelöscht" werden.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- GetBezByKey: Holt die externe Rechnungsnummer über die interne Nummer der Rechnung.
- FilterKonto (ab 5.0.7.0): Setzt einen Filter auf das angegebene Kreditorkonto, um nur die Rechnungen für einen Kreditur abzurufen.
- FilterOffene (ab 5.0.7.0): Setzt einen Filter so, dass nur offene Rechnungen (die nicht in die Buchhaltung übernommen wurden) abgerufen werden.
- FilterBelegnr (ab 5.0.7.0): Setzt einen Filter auf eine Fibu-Belegnummer, um nur die Rechnungen mit dieser Belegnummer abzurufen. Das sollte i.d.R. nur eine Rechnung sein, mehrere Ergebnisse sind aber möglich.

Rechnung

Daten einer Rechnung im Rechnungseingangsbuch.

Aktuelle Generation: 04.

Eigenschaften

- AdressNr: Adressnummer einer Einmaladresse
- Anmerkung: Anmerkungstext (Langtext)
- AuftragID: Auftragsnummer
- BankRefID: Bank

- BelegDatum: Belegdatum
- BelegNr: Fibu-Belegnummer
- BelegRefID: Interne Buchungsnummer
- Bestellnummer: Bestellnummer
- Betrag: Betrag
- BetragNetto: Nettobetrag
- BuchungskreisNr: Nummer des Buchungskreises
- Buchungstext: Buchungstext
- ClearingFlag: Clearingflag, Zahlungsart (Enum)
- DmsDocumentID: DokumentenID
- EingangsDatum: Eingangsdatum
- ESR: ESR-Nummer des Schweizer Zahlungsverkehrs
- GeloeschtDatum: Datum der (logischen) Löschung
- Geschaeftsjahr: Geschäftsjahr
- Kategorie: Interne Nummer der Kategorie
- KategorieName: Bezeichnung der Kategorie
- Kennung: Kennung
- KontoNr: Personenkonto
- OpBelegNr: OP-Belegnummer/Rechnungsnummer
- Periode: Periode
- RechnungNr: interne Nummer der Rechnung
- SkfBetrag: skontierfähiger Betrag
- StapellfdBuchNr: Laufende Nummer im Buchungsstapel (falls übergeben)
- StapelNr: Nummer des Buchungsstapel (falls übergeben)
- Status: Status
- SteuerartNr: Steuerart
- SteuersatzNr: Nummer des Steuersatzes
- ValutaDatum: Valutadatum
- Waehrung: Währung
- WiedervorlageDatum: Wiedervorlagedatum
- Zahlungssperre: Zahlungssperre bis
- Zahlzielname: Zahlungsziel-Kürzel

Methoden

- SetGeloescht: Setzt das Gelöscht-Flag für eine Rechnung.

SpracheCollection

Collection von Sprachen zur Sprachsteuerung in Briefreports wie Mahnungen.

Direktzugriff: Index ist die Bezeichnung der Sprache.

Die Sprachen können nur gelesen werden, die Pflege muss in der Fibu erfolgen.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- GetBezByKey: Holt eine Sprache über die Sprachen Nummer und nicht über das Kürzel.

Sprache

Daten einer Sprache zur Sprachsteuerung in Briefreports wie Mahnungen.

In den Konten wird die Sprache im Klartext gespeichert, nicht die Nummer der Sprache!

Aktuelle Generation: 01.

Eigenschaften

- Bezeichnung: Bezeichnung der Sprache
- SpracheNr: interne Nummer der Sprache

Methoden

Keine über die Standardmethoden hinausgehenden Eigenschaften.

SteuerCollection

Collection der (Umsatz-)Steuerschlüssel (oder Steuerarten) in einem Mandant.

Direktzugriff: Index ist die Nummer der Steuerart.

Steuerarten können nur gelesen werden, die Pflege muss in der Fibu erfolgen.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine besonderen Methoden. Die Methode CreateNewItem steht nicht zur Verfügung.

Steuer

Daten zu einem (Umsatz-)Steuerschlüssel (oder Steuerart) in einem Mandant.

Die Daten können nur gelesen werden.

Aktuelle Generation: 03.

Eigenschaften

- Beschreibung
- Steuerart: Bezeichnung der Steuerart
- SteuerartNr: interne Nummer der Steuerart (SteuerartRefID)
- DatevSteuer: Datev-Steuernummer (falls eingetragen)
- IsEGErwerb: Kennzeichen ob es sich um eine EG-Erwerbssteuer (genauer: Steuerart mit VSt-/USt-Buchungen) handelt. (0 = kein EG Erwerb) (sonst = EG Erwerb)
- Steuersaetze: Collection aller Steuersätze dieser Steuer

Methoden

Die Standardmethoden Insert, Update, Delete stehen nicht zur Verfügung.

- GetSteuerSatz: Liefert das zu dem angegebenen Datum gültige Objekt vom Typ Steuersatz.
- GetSteuerSatzProzent: Liefert den zu dem angegebenen Datum gültigen Steuer-Prozentsatz.

SteuerSatzCollection

Collection der Prozentsätze zu einem Steuerschlüssel (zu einem Objekt Steuer).

Direktzugriff: Index ist die Nummer des Steuersatzes.

Steuersätze können nur gelesen werden, die Pflege muss in der Fibu erfolgen.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

Keine besonderen Methoden. Die Methode CreateNewItem steht nicht zur Verfügung.

SteuerSatz

Daten zu einem Prozentsatz eines Steuerschlüssels (oder Steuerart).

Die Daten können nur gelesen werden.

Aktuelle Generation: 01.

Eigenschaften

- EGGeegenkonto: Umsatzsteuerkonto, falls Steuerschlüssel mit VSt-/USt-Buchung.
- GueltigAb: Datum ab dem der Steuersatz gültig ist
- NNFSkontokonto: Skontokonto für noch nicht fällige Steuer (Anzahlungsverfahren).
- NNFSteuerkonto: Steuerkonto für noch nicht fällige Steuer (Anzahlungsverfahren).
- SkontoKonto: Konto auf das standardmäßig Skonto gebucht wird
- SteuerartNr: Nummer der Steuerart zu dem der Steuersatz gehört
- SteuerKonto: Konto auf das standardmäßig die Steuer gebucht wird
- Steuersatz: Prozentsatz
- SteuersatzNr: interne Nummer des Steuersatzes (SteuerRefID)

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

ZahlzielCollection

Collection der Zahlungsziele.

Direktzugriff: Index ist die Nummer des Zahlungsziels.

Für die üblichen Zahlungsziele in Abstandstagen gilt die Regel, dass das Kürzel eindeutig sein muss. Es gibt jedoch außerdem die Zahlungsziele mit Gültigkeit für Bereiche eines Monats. Hier gilt, dass es unter einem Kürzel mehrere Einträge gibt, die mit ihrem Gültig-Ab-Datum den ganzen Monat abdecken müssen.

Aktuelle Generation: 01.

Eigenschaften

Keine über die Standardeigenschaften hinausgehenden Eigenschaften.

Methoden

- SearchZZ: Nach einem Zahlziel mit den angegebenen Werten suchen. Es wird das erste Zahlziel (als Zahlziel-Objekt) zurückgegeben bei dem alle Werte übereinstimmen, falls vorhanden. Anzugeben sind die Suchkriterien Skontotage1 / Skontoprozent1 / Skontotage2 / Skontoprozent2 / Nettotage / Zieltag / MonateZieltag. Nicht benötigte Werte sind auf 0 zu setzen.

Zahlziel

Daten eines Zahlungsziels, das die Fälligkeiten eines Offenen Posten wiedergibt. Zur eindeutigen Identifizierung siehe ZahlzielCollection.

Aktuelle Generation: 01.

Eigenschaften

- RefID: Nummer des Zahlungsziels (nur Lesen).
- Zahltag: Beginn des Gültigkeitsbereichs bei Zahlungszielen für Gültigkeitsbereiche. In der Fibu "Gültig ab (Tag im Monat)".
- Bezeichnung: eindeutige Bezeichnung des Zahlziels, in der Fibu "Kürzel". Diese Eigenschaft kann nur bei neuen Zahlzielen eingegeben werden. Es ist nicht möglich die Bezeichnung eines bestehenden Zahlzieles zu ändern.
- Text: Text zum Zahlungsziel, in der Fibu "Beschreibung".
- NurNetto: Skontierfähiger Betrag aus Netto.
- Skontosatz / Skontosatz2: Skonto-Prozentsätze für die erste und zweite Skontofrist. Im früheren OFCK indexierte Eigenschaft.

Für Zahlungsziele in Abstandstagen:

- Skontotage / Skontotage2: Anzahl Tage für die erste bzw. zweite Skontofrist. Im früheren OFCK indexierte Eigenschaft.
- Nettotage: Anzahl an Tagen bis zu dem der Betrag in Netto fällig ist.

Für Zahlungsziele mit Gültigkeitsbereichen:

- Zieltag: Zieltag im Monat für die Nettofälligkeit.
- MonatePlus: Monatsversatz für die Nettofälligkeit. 0, wenn der Nettofälligkeit-Zieltag im Monat der Entstehung des OPs liegt.
- MonatsSkontoZieltag / MonatsSkontoZieltag2: Zieltag und Monatsversatz für die erste und zweite Skontofälligkeit. Achtung: Monate mit 100 multipliziert addieren! Im früheren OFCK indexierte Eigenschaft.

Für beide Arten von Zahlungszielen:

- Toleranztage / Toleranztage2: tolerierte Zusatz-Abstandstage zur ersten und zweiten Skontofälligkeit.
- Skontowarnung / Skontowarnung2: Warngrenze für die erste und zweite Skontofälligkeit.

Methoden

Keine über die Standardmethoden hinausgehenden Methoden.

Testprogramm OFCKSNET.Test

Mit dem Testprogramm können Sie die Übergabe von Personenkonten und Stapelbuchungen testen. Die Testdaten müssen in einem XML-Format für einen SOAP-Envelope (für Webservices) vorliegen.

Das Testprogramm kann nach der Installation des OFCKS.NET auch genutzt werden, um mit der Anmeldung die korrekte Installation zu testen.

Bedienung des Testprogramms

Kopieren Sie die Konfigurationsdatei sql.config für den Datenbankzugang aus dem Programmverzeichnis des Syska ProFI in das Installationsverzeichnis des OFCKS.NET, sofern noch nicht gemacht. Starten Sie die OFCKSNET.Test.exe aus diesem Verzeichnis. Die Bedienung erfolgt über diese Maske:

The screenshot shows the 'Test OFCKS.NET' application window. It features a standard Windows-style title bar with minimize, maximize, and close buttons. The main area is divided into several sections:

- Database Information:** Fields for 'Datenbankname' (wr9100), 'Benutzer' (wr), 'Passwort' (masked with **), and 'Mandant' (50). An 'Anmelden' button is located below the Mandant field.
- Transaction Control:** A section titled 'Transaktion' containing 'Start', 'Commit', and 'Stop' buttons, and a 'Keine Transaktion' label.
- File Selection (Buchungs-Umschlag):** A 'Wie' button, a text field containing the path 'C:\Program Files (x86)\syska GmbH\syska SOAP Service 2.0\SOAP-Envelopes-9100\SE_StapelBuchung', and 'Suchen' and 'Buchen' buttons. Below the text field is another line: 'SE_StapelBuchungSave-OFCKSNET-Debitorenrechnung.xml'.
- File Selection (Konto-Umschlag):** A 'Wie' button, a text field containing the path 'C:\Program Files (x86)\syska GmbH\syska SOAP Service 2.0\SOAP-Envelopes-9100\SE_KontoSave_OF', and 'Suchen' and 'Speichern' buttons. Below the text field is another line: 'SE_KontoSave_OFCKSNET_DebitormitBanken.xml'.
- Close Button:** A 'Schließen' button is located at the bottom right of the window.

In den Feldern können folgende Eingaben gemacht werden:

- **Datenbankname:** Name der Datenbank, wie er in der sql.config im Tag <database name="..."> angegeben ist.
- **Benutzer:** Name des DB-Benutzers.
- **Passwort:** Passwort des DB-Benutzers.
- **Mandant:** Nummer des Mandanten, in dem gearbeitet werden soll.
- **Buchungs-Umschlag:** XML-Datei für eine Stapelbuchung.
- **Konto-Umschlag:** XML-Datei für ein Personenkonto.

Mit den Schaltflächen werden folgende Aktionen ausgelöst:

- **Anmelden:** Verbindung zur Datenbank aufbauen mit den angegebenen Anmeldedaten. Die Schaltfläche wird grün, wenn die Anmeldung erfolgreich war.
- **Wie:** Anzeige des Aufbaus der XML-Datei (ohne Namespace). Die Felder werden mit Datentyp genannt.
- **Suchen:** Auswählen einer XML-Datei.
- **Buchen:** Buchen = Einstellen der Stapelbuchung der XML-Datei in den Stapel.
- **Speichern:** Speichern des Kontos der XML-Datei.
- **Start / Stop / Commit:** Um mehrere Stapelbuchungen nacheinander in einer einzigen Transaktion zu buchen, verwenden Sie die Schaltfläche Start, wählen nacheinander die Dateien aus, buchen diese und verwenden abschließend die Schaltfläche Commit. Stop ist für den Abbruch einer begonnenen Transaktion vorgesehen.
- **Schließen:** Maske und Programm verlassen.

Die XML-Dateien

Im Unterverzeichnis Doku finden sich ausführliche Beispieldateien für die Aktionen.

Der Aufbau orientiert sich an dem notwendigen Aufbau für das UI-Testtool für SOAP-Envelopes im .NET Framework. Übernehmen Sie den Rahmen und ändern Sie nur Felder. Verwenden Sie zu den Feldern jeweils das Namensraum-Präfix.

Hinweise zu den Dateien

Es wird nur jeweils 1 Top-Element (Stapelbuchung, Konto) in einer Datei verarbeitet. Untergeordnete Elemente (Buchungszeilen, Bankverbindungen, Sepa-Mandate) können mehrfach vorhanden sein.

Beachten Sie Tags wie isKontoNew, in dem (zur Sicherheit) angegeben werden muss, ob man das Objekt neu einfügen oder aktualisieren möchte.

An vielen Stellen können Bool-Angaben als 1 / 0 oder als true / false angegeben werden.

Bei Bankverbindungen zu Konten ist die <bankID> wichtig. Zur Neuanlage darf es die Angabe nicht geben. Zum Update stellt es die eindeutige Zuordnung her. Aber auch ohne bankID kann man die Eigenschaft VorhandeneBankUpdates setzen. Dann versucht OFCK, anhand BLZ / Kontonummer bzw. IBAN vorhandene Bankverbindungen zum Update zu identifizieren.

Deutsche Umlaute sind – auch in Kommentaren – nur zulässig, wenn sie im passenden Zeichensatz zum encoding laut XML-Header vorhanden sind.

Beträge mit Dezimalpunkt, Datumsangaben gemäß Windows-Ländereinstellung (also i.A. tt.mm.jjjj).

Entwickler-Demoprojekt

Um die grundsätzliche Vorgehensweise zu verdeutlichen, wird ein Entwicklungsprojekt mit ausgeliefert. Zur Installation ist die angepasste Installation zu wählen.

Visual Studio 2010 Projekt in C#

Das Projekt OFCKSNET.Demo verdeutlicht mögliche Wege, um

- eine Stapelbuchung (eine Ausgangsrechnung) in einen Buchungstapel einzufügen.
- ein Personenkonto anzulegen oder zu aktualisieren.

Es führt in stark vereinfachter Form dieselben Aktionen aus, wie das Testprogramm OFCKSNET.Test. Diese beiden Schnittstellen zu einer Finanzbuchhaltung hin werden am häufigsten von Voranwendungen benutzt.

Projektaufbau

Ausgeliefert werden die Source-Bestandteile für ein lauffähiges Projekt. Angepasst werden muss der Pfad zur OFCKSNET.dll, auf die verwiesen wird.

In der Oberfläche sind möglich:

- Sich in einer Datenbank und einem Mandant anzumelden. Dazu ist eine Konfigurationsdatei sql.config zusätzlich notwendig, wie sie auch für die Hauptprogramme und OFCKS.NET verwendet wird.
- Eine Stapelbuchung, deren Daten im Programm hinterlegt sind, mit der entsprechenden Schaltfläche in einen Buchungstapel einzufügen.
- Ein Debitorenkonto, dessen Daten im Programm hinterlegt sind, mitsamt Bankverbindungen mit der entsprechenden Schaltfläche zu speichern – zur Neuanlage oder zur Änderung.

Bestandteile

Die wesentlichen Source-Einheiten sind:

- DemoMain.cs enthält die Oberfläche und die zugehörige Verarbeitung.
- DemoOFCK.cs enthält die eigentliche Verarbeitung, die nachfolgend genauer beschrieben ist.
- DatenStapelbuchung.cs enthält Daten einer Stapelbuchung mit 2 Buchungszeilen Debitoren an Erlöskonto, letztere Zeile mit Kostenkontierung.
- DatenKonto.cs enthält Daten eines Debitorenkontos mit 2 Bankverbindungen. Das Konto ist im Standard auf Neuanlage eingestellt mit dem Schalter istNeu.

Die wesentlichen verarbeitenden Funktionen in DemoOFCK.cs sind:

- Anmelden führt die Anmeldung an der Datenbank durch und lädt den gewünschten Mandant.
- StapelbuchungBuchen ist die Ereignisverarbeitung hinter der Schaltfläche Stapelbuchen.
- InsertStapelBuchung lädt den gewünschten Buchungsstapel, erzeugt darin eine neue leere Buchung und füllt sie mit Daten. Neben den Daten des Belegkopfs sind dies (in Unterfunktionen) 2 Buchungszeilen, eine davon mit Kostenrechnung. Anschließend wird die Insert-Methode der Stapelbuchung aufgerufen.
- KontoSpeichern ist die Ereignisverarbeitung hinter der Schaltfläche Konto speichern.
- KontoSave lädt entweder ein vorhandenes Personenkonto oder legt ein neues leeres Konto an. Anschließend werden seine Felder gesetzt, danach das Konto eingefügt oder aktualisiert. Mit Unterfunktionen werden 2 Bankverbindungen angehängt.

Index

Keine Indexeinträge gefunden.