



syska EURO FIBU – Finanzsoftware für alle, die ihren Weg machen.

syska EURO FIBU 2019

OFCK für Windows (Open Finance Connectivity Kit)

Schnittstelle zum dynamischen Datenaustausch via API / DLL

1. INHALT

1.	<i>INHALT</i>	2
2.	<i>Einleitung</i>	6
3.	<i>Hinweise für OFCK-Updater</i>	7
	Änderungen von Version 7.2d auf Version 2007	7
	Änderungen von Version 2007 auf Version 2008.....	7
	Änderungen von Version 2008 auf Version 2008a.....	8
	Änderungen von Version 2008 auf Version 2009.....	8
	Änderungen von Version 2009 auf Version 2010.....	8
	Änderungen von Version 2010 auf Version 2011.....	9
	Änderungen von Version 2011 auf Version 2011a.....	9
	Änderungen von Version 2011a auf Version 2012.....	10
	Änderungen von Version 2012 auf Version 2013.....	10
	Änderungen von Version 2013 auf Version 2014.....	10
	Änderungen von Version 2014 auf Version 2015.....	11
	Änderungen von Version 2015 auf Version 2016.....	11
	Änderungen von Version 2016 auf Version 2016b	11
	Änderungen von Version 2016b auf Version 2017	11
	Änderungen von Version 2017 auf Version 2017d	12
	Änderungen von Version 2018 auf Version 2018a.....	12
	Änderungen von Version 2018a auf Version 2019.....	12
	Änderungen von Version 2019a auf Version 2019b	12
4.	<i>Wichtige Tipps zur Anbindung</i>	13
5.	<i>Technische Hintergründe</i>	14
	Dateien des OFCK	14
	Installation des OFCK.....	14
	Datenbank.....	14
6.	<i>Grundfunktionen</i>	16
	OFCK_Status	16
	OFCK_HoleMandantenNr	16
	OFCK_LoginDialog.....	17
	OFCK_Login.....	17
	OFCK_HoleMandantenInfo.....	17
	OFCK_HoleMandantenInfoDirekt.....	18
7.	<i>Konten</i>	19

	OFCK_HoleKonto	19
	OFCK_Peko	19
	OFCK_PekoDirekt	22
	OFCK_HoleFreieKontonr	23
	OFCK_ExistiertKonto	23
	OFCK_HoleSaldo	23
	OFCK_HoleUmsatz.....	24
	OFCK_ExistiertKuerzel.....	25
	OFCK_Sako	26
8.	<i>Buchungen</i>	28
	OFCK_HoleStapelNr	28
	OFCK_BuchungEinfuegen	28
	OFCK_BuchungEinfuegenDirekt	30
	Grundsätzliches zu Splitbuchungen.....	30
	OFCK_SplitBuchungEinfuegen	30
	OFCK_SplitStart	32
	OFCK_SplitEinfuegen	32
	OFCK_SplitEinfuegenEx.....	32
	OFCK_SplitKobuEinfuegen.....	32
	OFCK_SplitKobuEinfuegenEx	33
	OFCK_SplitEnde.....	34
9.	<i>Holen von Datensätzen</i>	35
	Grundsätzliches zu Hole-Funktionen	35
	OFCK_BeendeHolen	35
	OFCK_InitHoleBube.....	36
	OFCK_HoleBube	36
	OFCK_HoleKobu	38
	OFCK_InitHoleKonto.....	39
	OFCK_HoleSako	39
	OFCK_HolePeko	40
	OFCK_HoleKost.....	43
	OFCK_InitHoleSollwerte	44
	OFCK_HoleSollwerte.....	44
	OFCK_InitHoleOP	45
	OFCK_HoleOP	46
	OFCK_InitHoleHistOP.....	46
	OFCK_HoleHistOP	47
	OFCK_InitHoleSteusa	49

	OFCK_HoleSteusa.....	49
	OFCK_InitHoleStapel.....	50
	OFCK_HoleStapel	50
	OFCK_InitHoleMandant.....	51
	OFCK_HoleMandant.....	51
	OFCK_InitHoleSprache	52
	OFCK_HoleSprachName.....	52
	OFCK_InitHoleBatch	52
	OFCK_HoleBatchHeader	53
	OFCK_HoleBatchSplit.....	54
	OFCK_HoleBatchKobu	55
10.	Kostenrechnung.....	56
	OFCK_Kost.....	56
	OFCK_KostDirekt.....	57
	OFCK_HoleKostenSaldo	57
	OFCK_HoleKostenschablone.....	58
	OFCK_InitHoleKoscha.....	58
	OFCK_HoleKoscha	58
	OFCK_HoleKoschaZeile.....	59
	OFCK_BeendeHoleKoscha.....	59
	OFCK_HoleEKROptionen	59
11.	Aufbauten.....	61
	OFCK_InitHoleAufbau.....	61
	OFCK_HoleAufbau.....	61
	OFCK_HoleAufbauID	62
	OFCK_InitHoleAufbauSaldo	63
	OFCK_HoleAufbauSaldo	64
	OFCK_BeendeHoleAufbauSaldo.....	64
12.	sonstige Funktionen.....	66
	OFCK_HoleZahlungsziel	66
	OFCK_Zahlungsziel	67
	OFCK_HoleZZnr.....	67
	OFCK_HoleAktMahnstufe	67
	OFCK_Mandant	68
	OFCK_MandantDirekt.....	69
	OFCK_HoleKurs	70
	OFCK_KursEinfuegen.....	70
	OFCK_HoleSprache.....	71

	OFCK_Stapel	71
	OFCK_LoescheStapel	72
	OFCK_PruefeRecht	72
	OFCK_GetUsedEngine	73
	OFCK_SQLInit.....	73
	OFCK_SQLDeInit.....	74
	OFCK_HoleMandantenVersion.....	74
	OFCK_HoleFehlertext	75
	OFCK_HoleStartJahr	75
	OFCK_IsProfiversion	75
	OFCK_ErmittleBonitaet.....	76
13.	Protokoll	77
	Grundsätzliches zum Protokoll	77
	OFCK_Protokoll.....	78
	OFCK_ProtokollStart	79
	OFCK_ProtokollEnde.....	79
14.	Beispiele in C	80
	Anlage eines Kunden (Debitors).....	80
	Übergabe einer Ausgangsrechnung	80
	Schrittweises Holen von Buchungsbelegen	81
15.	Hinweise und Beispiele für Visual Basic	82
	Wichtige Hinweise für die Nutzer von Visual Basic.....	82
	Beispiele	83
16.	OFCKOUT	85
	Übersicht.....	85
	Die Funktion OFCK_OUT().....	86

2. Einleitung

Das OFCK (Open Finance Connectivity Kit) für Windows der syska GmbH dient der Anbindung von beliebigen Softwarepaketen (z. B. Warenwirtschaftssystemen, Lohnbuchhaltungen, Branchenlösungen, usw.) an die syska EURO FIBU unter Windows. Es richtet sich primär an den Entwickler von Anwendungssoftware aller Bereiche.

Das OFCK ermöglicht die direkte Übergabe von Buchungen in einen Stapel der Finanzbuchhaltung via DLL-Aufrufen von einem Rechner aus, oder gleichzeitig von mehreren Plätzen im Netz. Der Umweg über Austauschdateien wird somit überflüssig, die Datenübergabe im Speicher gewährleistet Datenbankunabhängigkeit. Der Aufruf der Funktionen ist über jede Programmiersprache möglich, ebenfalls über Makrosprachen von Standardprogrammen, die DLL-Funktionen zulassen (z. B. Excel, WinWord, StarOffice etc.).

Durch die Beschränkung auf einige wenige, einfach handhabbare Funktionsaufrufe (die Grundfunktionen) ist die Anbindung an die Finanzbuchhaltung in kürzester Zeit erstellt. Dabei sind Detailkenntnisse in der Finanzbuchhaltung nicht notwendig, man spart auf diesem Weg jedoch die aufwendige Eigenentwicklung einer Buchhaltungssoftware unter Windows. Darüber hinaus bietet das OFCK aber auch zahlreiche weitere Funktionen, um eine nahtlose Integration der syska EURO FIBU in Ihre Anwendung zu ermöglichen.

Das OFCK liegt jeder Versionsvariante der syska EURO FIBU kostenlos mit gleichem Funktionsumfang bei und kann ebenfalls kostenfrei Ihrem Softwarepaket beigelegt werden. Voraussetzung für den Einsatz beim Anwender ist dann lediglich eine Standard- oder Plusversion der syska EURO FIBU. Dadurch ist dann auch der Support für die Finanzbuchhaltung von der syska GmbH abgedeckt.

Bei weiteren Fragen können Sie sich gerne an uns wenden:

syska GmbH
Am Sandfeld 15
76149 Karlsruhe
Tel.: ++49 - (0)7 21 / 9 85 93 - 0
Fax: ++49 - (0)7 21 / 9 85 93 - 60
E-Mail: efsupport@syska.de
Homepage: www.syska.de

3. Hinweise für OFCK-Updater

Dieser Abschnitt ist für Benutzer/Integratoren früherer OFCK-Versionen gedacht und enthält Hinweise auf Erweiterungen und Änderungen zu den einzelnen Versionsschritten in der OFCK-Schnittstelle, die aufgrund von Erweiterungen in der syska EURO FIBU möglich und nötig wurden.

Je nachdem, welche Version des OFCK Sie aktuell nutzen, lesen Sie daher bitte die folgenden Abschnitte bis zur aktuellen Version durch. Die Änderungen zwischen Version 6.0 und 7.2d wurden der Übersicht halber entfernt. Diese sind in der OFCK-Beschreibung zur Version 9.0 (2008) noch enthalten. Bei einem Update von solch alten Versionen empfehlen wir allerdings, die Änderungshistorie bei Seite zu legen und Ihre verwendeten Funktionen im Referenzteil anzusehen.

Änderungen von Version 7.2d auf Version 2007

Die syska EURO FIBU Version 2007 entspricht der internen Versionskennung 8.0, von welcher auch im OFCK die Rede ist.

Es sind die folgenden neuen Funktionen hinzugekommen:

Zum Holen von historischen OPs:

- OFCK_InitHoleHistOP
- OFCK_HoleHistOP

Zum Holen von Sollwerten:

- OFCK_InitHoleSollwerte
- OFCK_HoleSollwerte

Folgende Funktionen und Strukturen wurden erweitert:

- OFCK_KOSTENSTELLE: Hier sind die Angaben zur Laufzeitbegrenzung hinzugekommen
- OFCK_KostDirekt: Die Parameter zur Laufzeitbegrenzung sind hinzugekommen
- OFCK_HoleKost: Die Parameter zur Laufzeitbegrenzung sind hinzugekommen
- OFCK_Kost: Der Parameter vom Typ OFCK_KOSTENSTELLE ist erweitert worden (s.o.)

Um ein Update zu erleichtern, wird die Datei OFCK72.DLL zusätzlich zur OFCK-Kaskade ausgeliefert.

Ab Version 2007a wird die Anbindung des OFCK auch unter **.NET** und **Java** unterstützt. Für **.NET** werden hierzu die Deklarationen des OFCK für C# und C++ mitgeliefert. Für Java wird neben den Deklarationen zusätzlich eine Wrapper-DLL mitgeliefert, welche die OFCK98.DLL kapselt und von Java aus aufgerufen werden kann. Näheres zu diesen Dateien und deren Nutzung finden Sie in den folgenden Kapiteln 4 und 5.

Änderungen von Version 2007 auf Version 2008

Die syska EURO FIBU Version 2008 entspricht der internen Versionskennung 9.0, von welcher auch im OFCK die Rede ist.

Es sind die folgenden neuen Funktionen hinzugekommen:

Zum Holen von Kostenschablonen und Kostenschablonenzeilen ohne Dialog:

- OFCK_InitHoleKoscha
- OFCK_HoleKoscha
- OFCK_HoleKoschaZeile
- OFCK_BeendeHoleKoscha

Zum Überprüfen von Benutzerdaten:

- OFCK_LoginDialog
- OFCK_Login
- OFCK_IsProfiversion

Zum Holen von Benutzerrechten:

- OFCK_PruefeRecht

Zum Abfragen der EURO KORE Optionen:

- OFCK_HoleEKROptionen

Änderungen von Version 2008 auf Version 2008a

Beim Holen des Bilanz-Aufbaus (Aktiv und Passiv) kann über die Funktion OFCK_HoleAufbau nun auch zusätzlich der Ausweichaufbau geholt werden.

Änderungen von Version 2008 auf Version 2009

Beim Einfügen von Splitbuchungen können gleiche Kontonummern im Soll und Haben übergeben werden. Es wird dabei ein Habensplit angenommen.

Bei der Übergabe von Personenkonten mit Länderkürzel und einer Umsatzsteuer-ID werden diese auf ihre Übereinstimmung geprüft.

Durch die zusätzliche Anbindungsmöglichkeit an eine MySQL-Datenbank haben sich die bisher nicht verwendeten Funktionen OFCK_SQLInit() und OFCK_SQLDelnit() geändert. Die Funktionen können zum Aufbauen einer Verbindung zu einem MySQL-Server und zum Trennen der Verbindung verwendet werden. Wenn auf die Datenbank zugegriffen wird ohne zuvor eine Verbindung hergestellt zu haben, wird der Verbindungsdialog der EURO FIBU angezeigt. Wenn die Option „Beim nächsten Start automatisch Verbinden.“ zuvor aktiviert wurde, wird der Verbindungsaufbau sofort versucht und der Verbindungsdialog wird nur bei einem fehlgeschlagenen Verbindungsversuch angezeigt. Um den Dialog trotz aktiviertem „Autoconnect“ anzuzeigen kann die Steuerungstaste (Strg) während dem ersten Datenbankzugriff gedrückt werden.

Mit der Funktion OFCK_UseSQL() kann geprüft werden, ob MySQL verwendet wird.

Mit den Funktionen OFCK_InitHoleAufbau() und OFCK_HoleAufbau() können zusätzlich die Hierarchien für Kostenstellen, Kostenträger und Kostenarten geholt werden.

Mit der neuen Funktion OFCK_HoleKostenSaldo() kann der Saldo einer Kostenstelle/eines Kostenträgers pro Kostenart ermittelt werden.

Das Intervall für das Aufrufen der OFCK_OUT() Funktion kann über einen INI-Eintrag in Millisekunden eingestellt werden.

Bei der Übergabe von OFCK-Strukturen (z.B. OFCK_PEKO) werden die darin enthaltenen Felder auf ihre Länge überprüft. Wenn ein Feld zu lang ist, liefert die entsprechende Funktion OFCKERR_Ueberlauf zurück.

Änderungen von Version 2009 auf Version 2010

Geänderte Funktionen:

- OFCK_HolePeko, OFCK_Peko, OFCK_PekoDirekt: Bei diesen Funktionen zu Personenkonten wurden die Parameter um die neuen Felder erweitert, insbesondere zu den Zusatzfeldern. Zudem wurden die Angaben von IBAN, SWIFT und Land der Bank für die erste Bankverbindung aufgespalten, so dass hier je ein eigener Parameter zur Verfügung steht.

Die Angabe des Startjahres bei der Anlage wurde aus der Struktur herausgenommen und in die Parameterliste aufgenommen.

Das Feld „Bankname“ hat jetzt bis zu 40 Zeichen, beim Holen sind 41 Zeichen zu reservieren.

- OFCK_Sako, OFCK_HoleSako: Hier sind neue Parameter hinzugekommen, insbesondere zur Buchungssperre. Bei Angabe der Steuersätze sind auch die drei neuen Werte möglich.

Entfallene Funktionen:

- OFCK_HolePekoEx, OFCK_PekoEx, OFCK_PekoDirektEx: Die Funktionalität dieser drei Funktionen wurde in die jeweilige Funktion ohne das Suffix „Ex“ integriert.

Konstanten:

- Durch die Erweiterung der Steuersätze sind drei neue Konstanten hinzugekommen (OFCK_MISCSTEUERSATZ2, OFCK_VERRECHNUNGSSATZ3, OFCK_VERRECHNUNGSSATZ4), die beim Holen von Sachkonten und beim Holen der Steuersätze als Rückgabewerte auftreten können.

Änderungen von Version 2010 auf Version 2011

- Die Länge der Belegnummer wurde von 8 auf 16 Zeichen verlängert und ein 2. Buchungstext aufgenommen. Dies betrifft alle Funktionen, die Buchungen (auch Split und Kosten) sowie OPs übergeben oder holen:

- OFCK_BuchungEinfuegen()
- OFCK_SplitBuchungEinfuegen()
- OFCK_BuchungEinfuegenDirekt()
- OFCK_SplitEinfuegen()
- OFCK_SplitEinfuegenEx()
- OFCK_HoleBube()
- OFCK_HoleKobu()
- OFCK_HoleBatchSplit()
- OFCK_HoleBatchKobu()
- OFCK_HoleOP()
- OFCK_HoleHistOP()

Achten Sie insbesondere bei den Hole-Funktionen auf die neue Feldlänge und fügen Sie ein Feld für den 2. Buchungstext ein.

- Die INI-Dateien EFIBU.INI und OFCK.INI wurden durch Registry-Einträge abgelöst. Bei Installation der EURO FIBU werden diese automatisch übernommen. Beachten Sie daher bitte, dass auf allen Arbeitsplätzen, an denen das OFCK oder OFCK_Out genutzt wird, die Installation oder Netzinstitution der EURO FIBU ausgeführt wird.
- Die Funktion OFCK_HoleSaldo wurde um einen Parameter erweitert, so dass ein Saldo auch zu einem Stichtag ermittelt werden kann.

Änderungen von Version 2011 auf Version 2011a

- Über die neue Funktion OFCK_SplitKobuEinfuegenEx() können nun auch Kostenbuchungen mit Mengenangaben übergeben werden.

Änderungen von Version 2011a auf Version 2012

- Bei den folgenden Funktionen können mit einem zusätzlichen Parameter DMS-IDs geholt oder übergeben werden:
 - OFCK_BuchungEinfuegen()
 - OFCK_SplitBuchungEinfuegen()
 - OFCK_BuchungEinfuegenDirekt()
 - OFCK_SplitStart()
 - OFCK_HoleBube()
 - OFCK_HoleBatchHeader()
 - OFCK_HoleOP()
 - OFCK_HoleHistOP()
- Hinweise zur Übergabe der DMS-ID finden Sie am Ende der Funktion OFCK_BuchungEinfuegen().

- Für das Holen von Aufbauten und Aufbausalden wurden die neuen Aufbau-IDs für die zweite und dritte GuV sowie für die zweite und dritte Bilanz (aktiv und passiv) hinzugefügt: OFCK_AUFGUV2; OFCK_AUFGUV3; OFCK_AUFBILAKT2; OFCK_AUFBILAKT3; OFCK_AUFBILPAS2; OFCK_AUFBILPAS3;

Änderungen von Version 2012 auf Version 2013

- Bei den folgenden Funktionen wurden Parameter für die Einschränkung oder Übergabe der Kostenstellendimensionen 1-6 hinzugefügt:
 - OFCK_Kost()
 - OFCK_HoleKonto()
 - OFCK_SplitKobuEinfuegen()
 - OFCK_SplitKobuEinfuegenEx()
 - OFCK_HoleKobu()
 - OFCK_HoleKost()
 - OFCK_ExistiertKonto()
 - OFCK_HoleSaldo()
 - OFCK_HoleUmsatz()
 - OFCK_KostDirekt()
 - OFCK_HoleBatchKobu()
 - OFCK_InitHoleSollwerte()
 - OFCK_HoleKoschaZeile()
 - OFCK_HoleKostenSaldo()

Diese werden mit 0-5 übergeben. Dabei steht 0 für die Dimension 1 in der Fibu usw.

Änderungen von Version 2013 auf Version 2014

- Die Funktionen für die Interaktion mit Personenkonten OFCK_Peko(), OFCK_PekoDirekt() und OFCK_HolePeko() wurden um zusätzliche Parameter für die SEPA-Mandatsangaben ergänzt.
Damit in der Fibu bereits eingetragene oder konvertierte Bankverbindungsdaten wie die IBAN nicht ungewollt durch leer übergebene Einträge gelöscht werden (z.B. da diese in einer alten Version der einbindenden Software noch nicht vorhanden sind), werden diese im Gegensatz zu anderen Feldern nicht gelöscht.
- Die EURO FIBU und auch das OFCK unterstützt nun wahlweise auch den Microsoft SQL Server. Dazu wurde die Funktion OFCK_UseSQL() durch die Funktion OFCK_GetUsedEngine() ersetzt. Über diese kann der verwendete Datenbanktyp ermittelt werden (OFCK_ENG_BTREIVE, OFCK_ENG_MYSQL oder OFCK_ENG_MSSQL). Auch die Funktion für den spezifischen Verbindungsaufbau OFCK_SQLInit() hat sich aufgrund der Unterstützung von MSSQL geändert.

- Die neue Funktion OFCK_HoleZZnr() holt zu einem Zahlungszielkürzel dessen Nummer.
- Bei den Zahlungsdaten entfällt OFCK_IPI als Konstante. Es kamen Konstanten für die SEPA-Überweisung und die Lastschrift sowie die Mandatstypen hinzu, die bei OFCK_HoleHistOP() zurückgegeben werden.

Änderungen von Version 2014 auf Version 2015

- Die Funktionen für die Interaktion mit Personenkonten OFCK_Peko(), OFCK_PekoDirekt() und OFCK_HolePeko() wurden um zusätzliche Parameter für Inkasso, Bonität und SEPA/CAMT ergänzt.
- Die Funktionen zum Holen von OPs und historischen OPs wurden um Parameter für das Inkasso ergänzt.
- Die Funktion OFCK_ErmittleBonitaet() zum Ermitteln der Bonität ist hinzugekommen.

Achtung! Aufgrund einer Vereinheitlichung des OFCKs für die verschiedenen Programmiersprachen, haben sich kleinere Änderungen ergeben, die einer einmaligen Anpassung nach Einbindung des OFCK97 bedürfen:

- Bei der C++-Schnittstelle wurden einige Parameter und Strukturmember vom Typ long in BOOL geändert. Beide Typen entsprechen letztlich einem Integer (32Bit), der intern als boolescher Wert (1/0) behandelt wird. Dies ändert also nichts an der Logik oder dem Aufbau der Funktionen.

Bei Fehlern während des Kompilierens kann entweder der Typ geändert werden oder bei den entsprechenden Parametern in BOOL gecastet werden.
- In den anderen Sprachen sind Änderungen bei den Variablennamen möglich. Diese werden durch den Compiler aufgezeigt, da die entsprechenden Strukturmember nicht mehr gefunden werden. Die Änderungen betreffen überwiegend den typspezifischen Prefix (z.B.: name = sName; szName = sName; wert = iWert...) und die Groß-/Kleinschreibung einzelner Buchstaben.
- Bei der Java-Schnittstelle wird auf die Verwendung der StringBuffer-Klasse verzichtet. Alle Receiveparameter werden in Objekten gesetzt, welche der Funktion als zusätzlicher Parameter übergeben werden. Der jeweilige Klassentyp des zu übergebenden Objekts ist in der Datei Funktionen.java als Kommentar hinterlegt.

Änderungen von Version 2015 auf Version 2016

- Die Funktionen für die Interaktion mit Personenkonten OFCK_Peko(), OFCK_PekoDirekt() und OFCK_HolePeko() wurden um die Steuernummer ergänzt.

Änderungen von Version 2016 auf Version 2016b

- Dokumentation wurde ergänzt. Bitte beachten Sie die Funktion OFCK_SQLDelnit(), auch wenn Sie keine der unterstützten SQL-Datenbanken bei der Entwicklung verwenden.

Änderungen von Version 2016b auf Version 2017

- Die Funktionen für die Interaktion mit Personenkonten OFCK_Peko() und OFCK_PekoDirekt() wurden um zwei unsigned 64-Bit UpdateFlags Parameter ergänzt, mit deren Hilfe einzelne Felder der Personenkonten aktualisiert werden können. Sofern Sie diese Funktionen nutzen und das OFCK100 oder eine künftige Version einbinden, müssen Sie diese Parameter aufnehmen. Die OFCK-DLL ist aber unverändert eine 32-Bit Bibliothek.

Änderungen von Version 2017 auf Version 2017d

- Es können nun über das OFCK Anzahlungsvorfälle angelegt werden, sofern das Konto der Habenseite (Beispielsweise: 1390 SKR-03, 1490 SKR-04) als solches erkannt wird.

Änderungen von Version 2018 auf Version 2018a

- Es können nun über das OFCK Schlussrechnungen bei Anzahlungsvorfällen übergeben werden.

Beachten Sie hierzu die Voraussetzungen und Buchungsabläufe, welche im Handbuch bzw. der F1-Hilfe beschrieben sind. Achten Sie unbedingt darauf, dass in den Einstellungen der syska EURO FIBU unter Buchen / Anzahlungsvorfälle die Option: "Schlussrechnungsautomatik bei Anzahlungen" aktiv ist.

Im Zusammenhang mit Anzahlungsvorfällen kann es zu den folgenden Rückgabewerten kommen:

OFCKERR_AZV_SR_Pending (-107)
OFCKERR_AZV_SR_Generierung_Fail (-108)
OFCKERR_AZV_SR_Generierung_Abgeschlossen (-109)
OFCKERR_AZV_SR_Generierung_Stapel (-110)
OFCKERR_AZV_SR_Abschluss_Kleiner_Anzahlung (-111)
OFCKERR_AZV_SR_Konsistenz_Steuersaetze (-112)
OFCKERR_AZV_SR_Konsistenz_Opbetrag (-113)
OFCKERR_AZV_SR_Abschluss_Vor_Abschluss (-114)
OFCKERR_AZV_SR_Fail (-115)
OFCKERR_AZV_SR_Belnr_Sako (-116)
OFCKERR_AZV_SR_Schlussbuchungen (-117)

Änderungen von Version 2018a auf Version 2019

- OFCK_Sako und OFCK_HoleSako unterstützen die beiden neuen Felder (long lUst2aKz, long lFormularKennung) für die Umsatzsteuererklärung. Sofern Sie diese Funktionen nutzen und das OFCK120 oder eine künftige Version einbinden, müssen Sie diese Parameter aufnehmen.
- OFCK_Sako() wurde zudem um ein UpdateFlag ergänzt, mit dessen Hilfe einzelne Felder aktualisiert werden können. Sofern Sie diese Funktionen nutzen und das OFCK120 oder eine künftige Version einbinden, müssen Sie diesen Parameter aufnehmen.

Änderungen von Version 2019a auf Version 2019b

- OFCK_HoleHistOP() liefert im Parameter plZahlungsart auch eine Information darüber, ob der OP im Zuge der Anzahlungsvorfall-Schlussrechnung ausgeglichen wurde.

4. Wichtige Tipps zur Anbindung

In diesem Abschnitt werden zunächst einige **wichtige Informationen** und Tipps zur Einbindung des OFCK aufgeführt, bevor auf die Funktionen im Einzelnen eingegangen wird:

Die Funktionsprototypen, Datenstrukturen und Konstantendefinitionen sind in der Datei **OFCK.H** (für C bzw. C++), **OFCK.BAS** (für Visual Basic, Access, Word, VBA, ...), **OFCK.CS** (für C# sowie C++ unter .NET), **OFCKJ.JAR** (Java) sowie teilweise bzw. exemplarisch in der **OFCK.PAS** (für Delphi) enthalten. Zur Einbindung in andere Programmiersprachen kann eine der Dateien syntaktisch angepasst werden.

Die entsprechende Datei kann einfach in Ihren Quelltext eingebunden werden. Dabei ist zu beachten, dass Variablen unter Windows Vista/7/8/10 vom Typ 'long' vier Byte lang sind.

Für Java ist zusätzlich die **OFCKJ120.DLL** nötig, welche die OFCK120.DLL kapselt.

Achtung: Unter **MS-Access** und **Visual Basic** sind Integer- und BOOL-Werte lediglich zwei Byte lang, so dass hier zur Übergabe ein Long verwendet werden muss! Beachten Sie bei Strings, dass diese mit einer abschließenden/terminierenden Null versehen werden. Beachten Sie zudem auf alle Fälle das Kapitel 14 mit weiteren Hinweisen sowie Beispielen.

Achtung: Bei Verwendung von Strukturen ist beim Compiler das **'1-Byte-Alignment'** einzustellen.

Initialisieren Sie Strukturen vor deren Benutzung komplett **mit Nullwerten** (ASCII 0 bzw. \0, nicht „0“). Dadurch sind alle nicht benötigten (optionalen) Felder richtig initialisiert und Sie müssen diese nicht separat leeren.

Die häufigste Ursache für Abstürze der OFCK-DLL sind **zu kurze** oder falsch übergebene **Rückgabevariablen**, so dass hier äußerst sorgfältig vorgegangen werden sollte. Insbesondere bei MS-Access oder Visual Basic ist darauf zu achten, dass ein Zeiger auf die Variable - und nicht der Variableninhalt selbst - übergeben wird und das außerdem genügend Speicher für den Rückgabewert zuvor reserviert wird. Um Fehler zu vermeiden und besser eingrenzen zu können, ist es sehr hilfreich, einige Funktionsaufrufe in einem kleinem Beispielprogramm zuerst auszuprobieren, bevor diese in die eigentliche Anwendung integriert werden.

Beim Start der OFCK-DLL werden die Registry-Einträge der syska EURO FIBU ausgewertet, um den Datenbankpfad der syska EURO FIBU - Dateien zu ermitteln. Liegt diese Datei nicht vor oder wird dort kein gültiger Eintrag gefunden, so wird noch in der Registry unter \Software\syska\settings\EFIBU\ nach dem Pfad geschaut. Die OFCK-DLL zur syska EURO FIBU 2019 heißt OFCK120.DLL. Die OFCK-DLL ist außerdem multithread-fähig, d. h. sie kann von verschiedenen Threads innerhalb eines Prozesses gleichzeitig aufgerufen werden.

Achten Sie beim Einbinden in Ihr Projekt darauf, dass es sich bei der OFCK-DLL um eine 32-Bit Bibliothek handelt.

Beachten Sie bitte insbesondere auch das folgende Kapitel mit den technischen Hintergründen und Installationshinweisen!

5. Technische Hintergründe

Dateien des OFCK

- OFCK120.DLL Kern des OFCK mit den im Folgenden beschriebenen Funktionen
- OFCK.PDF Dieses Dokument im PDF-Format
- OFCK.BAS Includedatei mit Funktionsprototypen in VB - Notation
- OFCK.PAS Includedatei mit Funktionsprototypen in Delphi / Pascal – Notation (unvollständig)
- OFCK.H Includedatei mit Funktionsprototypen in C/C++ - Notation
- OFCK120.LIB Linker-Library
- OFCK.CS Includedatei mit Funktionsprototypen in C++ sowie C# für .NET
- OFCKJ.JAR Definitionsdateien für Java
- OFCKJ120.DLL Wrapper-DLL für Aufrufe aus Java

Installation des OFCK

Zur Implementierung der OFCK-Anbindung installieren Sie zunächst die syska EURO FIBU. Danach kopieren Sie aus Ihrem EUROFIBU-Verzeichnis die Dateien OFCK120DLL, OFCK.H (bzw. diejenigen für Ihre Entwicklungsumgebung) und OFCK120.LIB in die entsprechenden Verzeichnisse Ihrer Entwicklungsumgebung. Die OFCK120.LIB oder OFCK120.DLL ist dann noch dem Linker Ihrer Entwicklungsumgebung als zu verwendende Library bekannt zu geben. Für Java kopieren Sie hierzu zusätzlich die OFCKJ120.DLL.

Zur Installation beim Anwender sollte zunächst die syska EURO FIBU installiert werden. Bei der Installation im Netz muss zuerst eine komplette Version im Netz installiert werden, erst dann können die Netzinstallationen (s.u.) auf den weiteren Rechnern erfolgen.

Auf Plätzen, auf denen Sie keine vollständige Kopie der syska EURO FIBU installieren möchten, sollten Sie stattdessen die Netzinstallation (install.exe mit Parameter ‚n‘ aufrufen) der syska EURO FIBU durchführen. Hierbei werden nur die absolut notwendigen Dateien kopiert (die OFCK-DLL in den jeweils verfügbaren Versionen, die OCR-Fonts für die Reports sowie die OFCKJ120.DLL). Die Registrierung aller benötigten COM-Komponenten, Einträge ins Start-Menü und Registry etc. erfolgt ebenfalls.

Das manuelle Kopieren einzelner Dateien entfällt. Auf diese Weise kann der Anwender selbst die Zwischenversionen der syska EURO FIBU aktualisieren, ohne Dateien selbst kopieren/ersetzen zu müssen

Wichtig: Bei einer Installation im Netz muss die syska EURO FIBU komplett auf einem Netzwerklaufwerk installiert werden, da das OFCK Programm- und Datenbankdateien der syska EURO FIBU verwendet, auf die es ansonsten nicht zugreifen könnte. Außerdem benötigen alle Anwender Ihrer Anwendung, die über das OFCK auf die syska EURO FIBU zugreifen, Schreib/Lese-Rechte und Rechte um Dateien anzulegen und zu löschen.

Datenbank

Die syska EURO FIBU nutzt die indexsequentielle Datenbank Btrieve. Diese wird als lokale Engine mitgeliefert, welche die Datenbankaufrufe abhandelt und die Datendateien pflegt. Die Datenbankdateien können sich dabei wahlweise auf der Arbeitsstation oder auf dem Server befinden. Btrieve kann auch in der (allerdings lizenzpflichtigen) Client/Server-Variante eingesetzt werden, bei welcher die Anfragen auf dem Server verarbeitet und die Daten an den Client bzw. Arbeitsplatz zurückgesendet werden. Für die syska EURO FIBU und OFCK-DLL ist es dabei gleichgültig auf welcher konkreten Btrieve-Installation sie aufsetzt.

Alternativ kann die syska EURO FIBU Plus in einer SQL-Variante auch MySQL / MS SQL als Datenbank nutzen. Diese kann lokal oder auf einem Server installiert werden. Näheres finden Sie in der MySQL-Installationsanleitung zur EURO FIBU.

Beachten Sie hierzu die Funktion OFCK_SQLDeInit(), welche die Datenbankverbindung abbaut. Diese sollten Sie beim Beenden Ihrer Anbindung bzw. beim Entladen der OFCK-DLL aufrufen. Auch dann, wenn bei der Entwicklung keine SQL-Datenbank zur Anwendung kommt.

6. Grundfunktionen

OFCK_Status

long WINAPI **OFCK_Status**()

Diese Funktion gibt den Status der OFCK-DLL zurück. Damit kann getestet werden, ob die DLL richtig initialisiert wurde.

Rückgabewert:

long lErrno OFCK_OK, falls die DLL korrekt initialisiert wurde, OFCKERR_Datenbank, falls die Datenbank nicht korrekt initialisiert wurde (meist ist dann der Pfad in der Registry falsch gesetzt); OFCKERR_Ini, falls die Einträge in der Registry nicht existieren oder die Datenbankpfade nicht gesetzt sind.

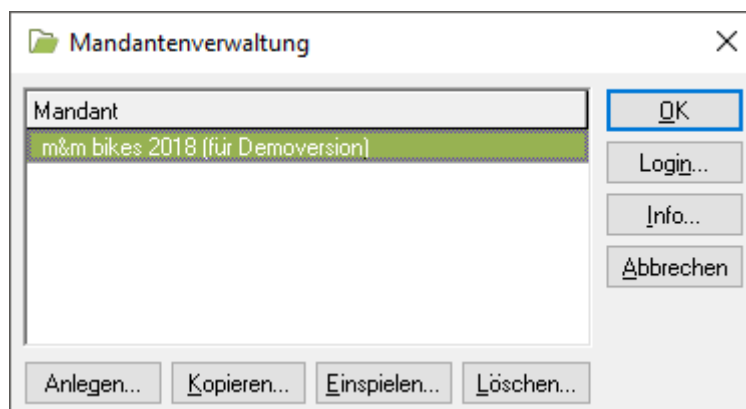
Sinn dieser Funktion ist, einem die OFCK-DLL benutzenden Programm zu ermöglichen, andere Funktionsaufrufe von vornherein zu unterdrücken, falls die DLL sowieso keine korrekten Werte oder Fehlermeldungen liefert.

OFCK_HoleMandantenNr

long WINAPI **OFCK_HoleMandantenNr**(HWND hParent, LPSTR lpszMandantenName)

Dient der Ermittlung der Nummer des zu bebuchenden Mandanten. Diese wird bei allen folgenden Aufrufen benötigt. Sie sollte im aufrufenden Programm gespeichert werden. Ist sichergestellt, dass immer nur der gleiche Mandant bebucht werden soll, so kann die Mandantenummer fest vorgegeben werden (der zuerst angelegte Fibu-Mandant hat immer die Nummer 1). Für den Rückgabewert des Namens müssen die in {}-Klammern angegebene Anzahl an Bytes reserviert werden.

Es erscheint der folgende Dialog zur Mandantenauswahl:



Parameter:

HWND hParent Handle des Parent-Fensters unter dem der Dialog zur Mandantenauswahl erscheinen soll.

LPSTR
lpszMandantenName {41} Falls ungleich NULL: Enthält nach erfolgreichem Aufruf den ermittelten Mandantennamen. Es müssen mindestens 41 Byte für diesen Rückgabewert reserviert werden.

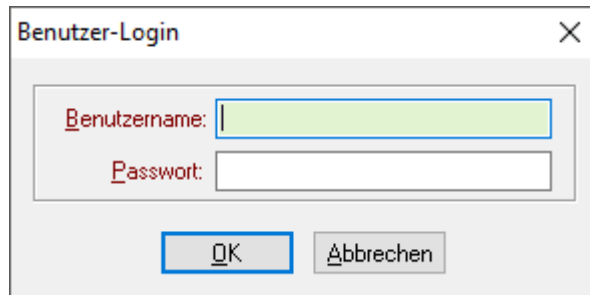
Rückgabewert:

long lMandnr Mandantenummer > 0; 0 bei Dialogabbruch; Fehlernummer < 0.

OFCK_LoginDialog

long WINAPI **OFCK_LoginDialog**(HWND hParent, long IMandnr)

Dient zur Überprüfung von Benutzerdaten zu einem Mandanten anhand eines Abfragedialogs:



Parameter:

HWND hParent Handle des Parent-Fensters unter dem der Dialog zur Mandantenauswahl erscheinen soll.

long IMandnr Die Mandantenummer für welchen ein Login stattfinden soll.

Rückgabewert:

long lErrno OFCK_OK bei erfolgreichem Login; 0 bei Dialogabbruch; Fehlernummer < 0.

OFCK_Login

long WINAPI **OFCK_Login**(long IMandnr, LPSTR pszBenutzer, LPSTR pszPasswort, BOOL bShowErr)

Dient der Überprüfung von Benutzerdaten zu einem Mandanten.

Parameter:

long IMandnr Die Mandantenummer für welchen ein Login stattfinden soll.

LPSTR pszBenutzer {31} Der Benutzername für welchen ein Login stattfinden soll.

LPSTR pszPasswort {31} Das zu überprüfende Passwort

BOOL bShowErr TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long IMandnr OFCK_OK bei erfolgreichem Login; Fehlernummer <= 0.

OFCK_HoleMandantenInfo

long WINAPI **OFCK_HoleMandantenInfo**(long IMandnr, LPOFCKMI lpMandInfo, BOOL bShowErr)

Holt nähere Informationen zu einem Mandanten. Aufbau der OFCK_MANDANTENINFO - Struktur s.u.

Parameter:

long IMandnr Mandantenummer.

LPOFCKMI lpMandInfo Zeiger auf die OFCK_MANDANTENINFO – Struktur. Ist der Zeiger auf die Struktur der NULL-Zeiger, so wird OFCKERR_nicht_moeglich zurückgegeben.

BOOL bShowErr TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno OFCK_OK, falls die Mandanteninformationen ermittelt werden konnten, bei Fehlern: Fehlernummer < 0.

```

typedef struct
{
    long   aktjahr;           // aktuelles Geschäftsjahr (z.B. 1998)
    long   gejabeginn;       // Geschäftsjahresbeginn; Monat e {1, .., 12}
    long   kontenrahmen;    // e {0, OFCK_SKR03, OFCK_SKR04, OFCK_GKR,
                            //     OFCK_RLG, OFCK_KMU}
    BOOL   isistversteuerer; // ungleich 0 <=> der Mandant ist Istversteuerer
    BOOL   iseinueber;      // ungleich 0 <=> der Mandant ist
                            // Einnahmen/Überschussrechner
    BOOL   steuerfrei;      // ungleich 0 <=> keine Steuern (z.B.
                            // gemeinnützige Einrichtungen)
    BOOL   euro;            // ungleich 0 <=> der Euro ist die Grundwährung
                            // des Mandanten
    char   idnr[16];        // USt.ID-Nummer
    char   land[4];         // Landeskürzel des Mandanten (z.B. 'D' / 'A')
    char   waehrung[4];     // Währungskürzel der Landeswährung (z.B. 'DEM'
                            // oder 'ATS')
    char   eurokurs[12];    // Umrechnungskurs Euro in Landeswährung
                            // (z.B. 1 Eur = '1,95583' DM)
    char   svsako[8];       // Saldovortragskonto für Eröffnungsbuchungen
                            // auf Sachkonten
    char   svkred[8];       // Saldovortragskonto für Eröffnungsbuchungen
                            // auf Kreditoren
    char   svdebi[8];      // Saldovortragskonto für Eröffnungsbuchungen
                            // auf Debitoren
    char   name[41];        // Name des Mandanten
    char   zusatz[41];      // Zusatz aus dem Firmenstamm
} OFCK_MANDANTENINFO;

```

OFCK_HoleMandantenInfoDirekt

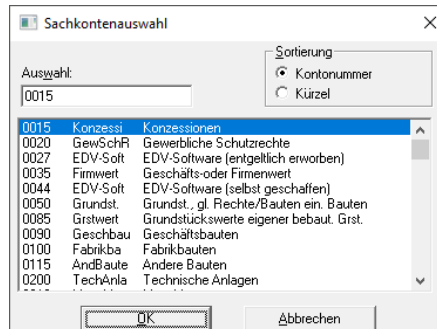
long WINAPI **OFCK_HoleMandantenInfoDirekt**(long lMandnr, long* plAktjahr, long* plGejabeginn, long* plKontenrahmen, BOOL* plIstversteuerer, BOOL* plIseinueber, BOOL* plSteuerfrei, BOOL* plEuro, LPSTR pszIdnr, LPSTR pszLand, LPSTR pszWaehrung, LPSTR pszEurokurs, LPSTR pszSvsako, LPSTR pszSvkred, LPSTR pszSvdebi, LPSTR pszName, LPSTR pszZusatz, BOOL bShowErr)

Diese Funktion gleicht der Funktion OFCK_HoleMandantenInfo. Die Parameter entsprechen der OFCK_MANDANTENINFO-Struktur. Zusätzlich kann für alle Zeiger (long* und LPSTR) der NULL-Zeiger übergeben werden.

7. Konten

OFCK_HoleKonto

long WINAPI **OFCK_HoleKonto**(HWND hParent, long lMandnr, long lKontoart, LPSTR lpszVon, LPSTR lpszBis, LPSTR lpszKnrOut, LPSTR lpszNameOut, LPSTR lpszKuerzelOut, long lDimension, LPSTR lpszKostref)



Dient der Ermittlung einer gültigen Kontonummer oder Kostenstelle/träger. Darüber können z.B. Erlös-konten ermittelt und festgehalten werden, um sie bei Ausgangsrechnungen angeben zu können. Es er-scheint der folgende Dialog zur Kontenauswahl:

Parameter:

- HWND hParent Handle des Parent-Fensters unter dem der Dialog zur Kontenauswahl erscheinen soll.
- long lMandnr Mandantenummer.
- long lKontoart Kontoart (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_KOST für Kostenstellen/träger).
- LPSTR lpszVon Kontonummer (mit führenden Nullen) ab welcher die Kontenliste aufzubauen ist; bei NULL wird die Liste vom ersten Konto aus aufgebaut.
- LPSTR lpszBis Kontonummer (mit führenden Nullen) bis zu welcher die Kontenliste aufzubauen ist; bei NULL wird die Liste bis zum letzten Konto aufgebaut.
- LPSTR lpszKnrOut {8} Ausgewählte Kontonummer (mit führenden Nullen). Es müssen mindestens 8 Byte für diesen Rückgabewert reserviert werden.
- LPSTR lpszNameOut {41} Kontoname, sofern der Zeiger ungleich NULL ist. Es müssen mindestens 41 Byte für diesen Rückgabewert reserviert werden.
- LPSTR lpszKuerzelOut {9} Kontokürzel, sofern der Zeiger ungleich NULL ist. Es müssen mindestens 9 Byte für diesen Rückgabewert reserviert werden.
- long lDimension Dimensionseinschränkung bei Kostenstellen. Bei -1 werden alle Kostenstellen aufgelistet.
- LPSTR lpszKostref {8} Kostenstellenreferenz. Bei aktivierter Kostenstellenreferenzierung und der 4. Dimension kann hier zusätzlich auf die Kostenreferenz eingeschränkt werden.

Rückgabewert:

- long lErrno OFCK_OK, falls in lpszKnrOut eine ausgewählte Kontonummer zurückgegeben wird; 0 bei Dialogabbruch; Fehlernummer < 0.

OFCK_Peko

long WINAPI **OFCK_Peko**(long lAktion, long lArt, LPSTR lpszAbJahr, OFCK_PEKO* lpPeko, BOOL bShowErr)

Dient der Neuanlage, Änderung oder Löschung eines Personenkontos (Debitor = Kunde; Kreditor = Lieferant). Soll der Datev-Export in der Fibu genutzt werden, muss die Debitorennummer im Bereich von 10000 bis 69999 und die Kreditorennummer im Bereich von 70000 bis 99999 liegen. Ansonsten kann die Kontonummer frei vergeben werden, muss aber eindeutig sein.

Die Umsatzsteuer-ID wird mit dem übergebenen Länderkürzel verglichen und bei Nichtübereinstimmung ein Fehler zurückgegeben.

Damit in der Fibu bereits eingetragene oder konvertierte Bankverbindungsdaten wie die IBAN nicht ungewollt durch leer übergebene Einträge gelöscht werden (z.B. da diese in einer alten Version der einbindenden Software noch nicht vorhanden sind), werden diese im Gegensatz zu anderen Feldern nicht gelöscht. Das Feld für die SEPA-Lastschriftenart und die SEPA-Flags werden dabei nur übernommen, wenn ein Mandat für die zugehörige Bankverbindung angegeben wurde.

Mittels zweier Updateflags, welche in der OFCK.H und den Include-Dateien der anderen unterstützten Programmiersprachen definiert sind, wird bei einem OFCK_UPDATE bestimmt, welche Felder eines Personenkontos aktualisiert werden sollen. Wird ein Flag gesetzt, so wird das entsprechende Feld mit dem übergebenen Wert aktualisiert. Auf diese Weise können Felder, die nur in der Fibu gepflegt werden, stehen gelassen werden.

Ebenfalls können neben den einzelnen veroderten Updateflags mit Hilfe zweier vordefinierter Konstanten OFCK_PEKO_UPDATE_FLAG1_ALLE und OFCK_PEKO_UPDATE_FLAG2_ALLE, alle Flags des jeweiligen Updateflags gesetzt werden.

Parameter:

long IAktion Aktion (OFCK_INSERT, OFCK_UPDATE, OFCK_DELETE).
long IArt Kontoart (OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren).
LPSTR lpszAbjahr Erstes Jahr. Format: 2- oder 4-stellig. Wird NULL oder ein Leerstring angegeben, so wird das aktuelle Jahr verwendet. Die Aktion wird ab diesem Jahr bis zum aktuellen Buchungsjahr durchgeführt. Werden die gültigen Werte überschritten, so wird das erste Jahr (Jahr vor erstem Jahr) oder das letzte Jahr (Jahr nach aktuellem Jahr) gesetzt.
OFCK_PEKO* lpPeko Zeiger auf die Struktur mit den Kontoangaben.
BOOL bShowErr TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno OFCK_OK oder Fehlernummer < 0.

Die Personenkontenstruktur ist wie folgt aufgebaut (siehe OFCK.H, OFCK.BAS bzw. OFCK.PAS):

```
typedef struct
{
    ULONG64 ulUpdateFlags1; // UpdateFlag 1 (64Bit)
    ULONG64 ulUpdateFlags2; // UpdateFlag 2 (64Bit)
    long mandnr; // Mandantennummer > 0
    long mahnkennziffer; // nur bei Kunden; (-1 <=> nicht mahnen,
                        // >=0 <=> mahnbar)
    long zahlungsziel; // optional: Zahlungszielschlüssel
    long skontotage; // Skontofrist in Tagen (>= 0)
    long zahltage; // Zahlungsfrist in Tagen (>= 0)
    BOOL einmalkonto; // e {OFCK_NEIN, OFCK_JA}
    BOOL lohnveredeler; // e {OFCK_NEIN, OFCK_JA}
    long sprachkey; // optional: Sprachkey für Mahnungen
                  // (mit OFCK_HoleSprache zu erfahren)
    BOOL lSperre; // Kred: Zahlungssperre, Debi: Liefersperre
    BOOL lBuchungssperre; // Buchungssperre
    BOOL lOption1; // optional: Zusatzfeld: Option1
    BOOL lOption2; // optional: Zusatzfeld: Option1 (nur PV)
    BOOL lOption3; // optional: Zusatzfeld: Option1 (nur PV)
    char dbkonto[8]; // optional: Durchbuchungskonto
    char kontonr[8]; // max. 7-stellige eindeutige Kontonummer
    char kuerzel[9]; // max. 8-stelliges eindeutiges Kürzel
    char name[41]; // Name
    char anred_firma[16]; // optional: Anrede Firma
}
```

```

char zusatz[41]; // optional: Zusatz
char ustidnr[16]; // optional: Umsatzsteuer ID-Nummer
char land[4]; // optional: Landeskürzel
char waehrung[5]; // optional: Landeswährung, leer lassen wenn
// identisch mit Grundwährung des Mandanten
char plz[7]; // optional: Postleitzahl
char ort[31]; // optional: Ortsbezeichnung
char strasse[31]; // optional: Strasse
char gegenkonto[8]; // optional: Kontonummer eines Sachkontos
char skontoproz[6]; // optional: Skontoprozentsatz im Format "00,00"
char gegenkonto2[8]; // optional: Kontonummer eines Sachkontos
char ansprech[41]; // optional: Ansprechpartner
char anred_anspr[16]; // optional: Anrede Ansprechpartner
char gefue[41]; // optional: Geschäftsführer
char anred_gefue[16]; // optional: Anrede Geschäftsführer
char tel[21]; // optional: Telefon
char fax[21]; // optional: Fax
char email[41]; // optional: E-Mail
char bemerkung[41]; // optional: Bemerkung (z.B. Zessionsvermerk)
char verwendungszweck[28]; // optional: Verwendungszweck
// (Vorschlag für die Überweisung)
char szZessionstext[41]; // optional: Zessionstext
char szText1[51]; // optional: Zusatzfeld: Text1
char szText2[51]; // optional: Zusatzfeld: Text2 (nur PV)
char szText3[51]; // optional: Zusatzfeld: Text3 (nur PV)
char szWert1[16]; // optional: Zusatzfeld: Wert1: -000000000,00
// oder leer
char szWert2[16]; // optional: Zusatzfeld: Wert1: -000000000,00
// oder leer (nur PV)
char szWert3[16]; // optional: Zusatzfeld: Wert1: -000000000,00
// oder leer (nur PV)
char szDatum1[11]; // optional: Zusatzfeld: Datum1: TT.MM.JJJJ
// oder leer
char szDatum2[11]; // optional: Zusatzfeld: Datum1: TT.MM.JJJJ
// oder leer (nur PV)
char szDatum3[11]; // optional: Zusatzfeld: Datum1: TT.MM.JJJJ
// oder leer (nur PV)
char szPfadInfo[MAX_PATH]; // optional: Infoverzeichnis
// MAX_PATH = 260
// erste Bankverbindung
char bank[41]; // optional: Bankname
char blz[9]; // optional: Bankleitzahl
char bankknr[17]; // optional: Bankkontonummer
char swiftcode[12]; // optional: SWIFT-Code der Bank
char szIban1[35]; // optional:
char szBankland1[3]; // optional:
char szKontoinhaber1[28]; // optional: abweichender Kontoinhaber
// zweite Bankverbindung
char szBank2[41]; // optional: Bankname
// (40 Stellen, wird ggf. abgeschnitten)
char szBlz2[9]; // optional: 8-stellige Blz
char szBankknr2[17]; // optional: 10- bzw. 11(für Österreich)-
// stellige Bankknr (Schweiz: 16 Stellen)
char szIban2[35]; // optional: IBAN
char szSwiftcode2[12]; // optional: SWIFT-Code (=BIC)
char szBankland2[3]; // optional: ISO-Land der Bank
char szKontoinhaber2[28]; // optional: abweichender Kontoinhaber
//40 Stellen, wird ggf. auf 27 abgeschnitten
// SEPA (erste Bankverbindung)
char szSepaMandat[36]; // optional: SEPA-Mandat
char szSepaMandatDatum[11]; // optional: Datum zum SEPA-Mandat.
char szSepaMandatNutzung1[11]; // optional: Datum zur SEPA-Nutzung
BOOL bSepaMandatFirmenLS1; // optional: Mandat ist Firmenlastschrift
long lSepaMandatFlags1; // ungenutzt: Flags zur SEPA-Nutzung.
char szSepaMandatDMSID1[51]; // ungenutzt: DMS-ID zum Mandat.
// SEPA (zweite Bankverbindung)
char szSepaMandat2[36]; // optional: SEPA-Mandat
char szSepaMandatDatum2[11]; // optional: Datum zum SEPA-Mandat.

```

```

char  szSepaMandatNutzung2[11]; // optional: Datum zur SEPA-Nutzung
BOOL  bSepaMandatFirmenLS2; // optional: Mandat ist Firmenlastschrift
long  lSepaMandatFlags2; // ungenutzt: Flags zur SEPA-Nutzung.
char  szSepaMandatDMSID2[51]; // ungenutzt: DMS-ID zum Mandat.
// SEPA (Allgemein)
char  szCreditorIdentifizier[36]; // opt.: Gläubiger-ID des Pekos.
char  szEnd2EndId[36]; // opt.: Ende-zu-Ende-ID.
long  lEnd2EndIdFlags; // opt.: Einstell. zur Ende-Zu-Ende-Id
// u.a. für Inkasso
char  szRegisternummer[21]; // opt.: Registernummer, z.B: HRB 123456
char  szRegistergericht[41]; // optional: Registergericht, z.B.:
// Amtsgericht Karlsruhe
char  szRechtsform[31]; // opt.: Rechtsform, z.B. GmbH & Co KG
char  szGeburtsdatum[11]; // optional: TT.MM.JJJJ
long  lInkassoGegner; // optional: Gegnernummer beim Inkasso.
long  lInkassoAkte; // optional: Aktive Akte beim Inkasso.
long  lBonitaet; // Bonität: 0 = ungeprüft,
// 1 = grün, 2 = gelb, 3 = rot.
char  szBonitaetPruefdatum[11]; // TT.MM.JJJJ oder leer
long  lBonitaetSuchTyp; // Typ. 0 = Unbekannt, 1 = Person,
// 2 = Einzelunternehmer/in, 3 = Firma
char  szVorname[31]; // optional: Vorname
char  szSteuernummer[16]; // optional: Steuernummer
} OFCK_PECO;

```

Die char-Felder sind grundsätzlich mit einer terminierenden Null (\0) abzuschließen. Leere Felder sind mit einer Null (\0) zu belegen. Es empfiehlt sich, die Struktur vor dem Zusammenstellen mit Null zu initialisieren.

OFCK_PekoDirekt

long WINAPI **OFCK_PekoDirekt**(long lMandnr, long lAktion, long lArt, LPSTR lpszAbjahr, BOOL bShowErr, ULONG64 ulUpdateFlags1, ULONG64 ulUpdateFlags2, LPSTR szKontonr, LPSTR szKuerzel, LPSTR szName, LPSTR szZusatz, LPSTR szUstidnr, LPSTR szLand, LPSTR szWaehrung, LPSTR szPlz, LPSTR szOrt, LPSTR szStrasse, LPSTR szBlz, LPSTR szBankknr, LPSTR szBank, LPSTR szGegenkonto, long lSkontotage, long lZahltag, LPSTR szSkontoproz, long lMahnkennziffer, long lZahlungsziel, LPSTR szGegenkonto2, LPSTR szAnsprech, LPSTR szGefue, LPSTR szTel, LPSTR szFax, LPSTR szEmail, LPSTR szBemerkung, BOOL lEinmalkonto, BOOL lLohnveredeler, LPSTR szVerwendungszweck, LPSTR szDBKonto, long lSprachkey, LPSTR szSwiftcode, LPSTR szAnredeFirma, LPSTR szAnredeAnsprech, LPSTR szAnredeGefue, LPSTR szZessionstext, LPSTR szText1, LPSTR szText2, LPSTR szText3, LPSTR szWert1, LPSTR szWert2, LPSTR szWert3, LPSTR szDatum1, LPSTR szDatum2, LPSTR szDatum3, BOOL lOption1, BOOL lOption2, BOOL lOption3, BOOL lSperre, LPSTR szPfadInfo, BOOL lBuchungssperre, LPSTR szlban1, LPSTR szBankland1, LPSTR szKontoinhaber1, LPSTR szBlz2, LPSTR szBankknr2, LPSTR szBank2, LPSTR szlban2, LPSTR szSwiftcode2, LPSTR szBankland2, LPSTR szKontoinhaber2, LPSTR szSepaMandat, LPSTR szSepaMandatDatum, LPSTR szSepaMandatNutzung1, BOOL bSepaMandatFirmenLS1, long lSepaMandatFlags1, LPCSTR szSepaMandatDMSID1, LPSTR szSepaMandat2, LPSTR szSepaMandatDatum2, LPSTR szSepaMandatNutzung2, BOOL bSepaMandatFirmenLS2, long lSepaMandatFlags2, LPCSTR szSepaMandatDMSID2, LPCSTR szCreditorIdentifizier, LPCSTR szEnd2EndId, long lEnd2EndIdFlags, LPCSTR szRegisternummer, LPCSTR szRegistergericht, LPCSTR szRechtsform, LPCSTR szGeburtsdatum, long lInkassoGegner, long lInkassoAkte, long lBonitaet, LPCSTR szBonitaetPruefdatum, long lBonitaetTyp, LPCSTR szVorname, LPCSTR szSteuernummer)

Diese Funktion entspricht der OFCK_Peko-Funktion, nur werden die Personenkonten-Parameter direkt übergeben. Die einzelnen Parameter entsprechen denen in der OFCK_Peko-Struktur.

Die bisherige Übergabe von Swiftcode (BIC) und IBAN ab Version 7.1 entfällt. Für IBAN, BIC und das Länderkürzel stehen ab Version 9.2 je ein eigenes Feld zur Verfügung.

long lErrno OFCK_OK oder Fehlernummer < 0.

OFCK_HoleFreieKontonr

long WINAPI **OFCK_HoleFreieKontonr**(long IMandnr, long IKontoart, LPSTR IpszKnrVon, LPSTR IpszKnrBis, LPSTR IpszKnrOut)

Holt die nächste freie Kreditoren- oder Debitorenkontonummer. Falls von dem gewählten Kontentyp noch kein Konto existiert, wird für Debitoren die Kontonummer 10000, für Kreditoren die Kontonummer 70000 zurückgegeben. Über die optionalen Parameter IpszKnrVon und IpszKnrBis kann ein Nummernkreis angegeben werden, aus welchem die nächste freie Nummer bis ausschließlich der IpszKnrBis ermittelt werden soll, was insbes. bei Sachkonten genutzt werden sollte.

Parameter:

long IMandnr	Mandantenummer.
long IKontoart	Kontoart (OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_SAKO für Sachkonten).
LPSTR IpszKnrVon	Optional: falls ungleich NULL kann hier die Anfangsnummer eines Nummernkreises übergeben werden, IpszKnrBis muss dann auch mit übergeben werden.
LPSTR IpszKnrBis	Optional: falls ungleich NULL kann hier die Endnummer eines Nummernkreises übergeben werden, IpszKnrVon muss dann auch mit übergeben werden.
LPSTR IpszKnrOut	Ermittelte (neue) Kontonummer (mit führenden Nullen). Es müssen mindestens 8 Byte für diesen Rückgabewert reserviert werden.

Rückgabe:

long IErrno	OFCK_OK, falls eine neue Kontonummer ermittelt werden konnte, OFCKERR_Ueberlauf, falls die neue Kontonummer länger als 7 Stellen wäre und OFCKERR_nicht_moeglich, falls die letzte Kontonummer in der Sortierreihenfolge keine Zahl ist und daher nicht erhöht werden kann, bei anderen Fehlern: Fehlernummer < 0.
-------------	--

OFCK_ExistiertKonto

long WINAPI **OFCK_ExistiertKonto**(long IMandnr, LPSTR IpszKontonr, long IKontoart, long IDimension, LPSTR IpszKostref)

Prüft, ob ein gegebenes Konto existiert.

Parameter:

long IMandnr	Mandantenummer.
LPSTR IpszKontonr	Kontonummer (mit führenden Nullen), die geprüft werden soll.
long IKontoart	Kontoart (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_KOST für Kostenstellen/Kostenträger).
long IDimension	Einschränkung auf die Dimension bei Kostenstellen.
LPSTR IpszKostref	Einschränkung auf die Kostenstellenreferenz bei Dimension=4 und aktivierter Kostenstellenreferenz für den Mandant.

Rückgabe:

long IErrno	OFCK_OK, falls das Konto existiert; falls es nicht existiert: OFCK_keinKonto, bei anderen Fehlern: Fehlernummer < 0.
-------------	--

OFCK_HoleSaldo

long WINAPI **OFCK_HoleSaldo**(long IMandnr, LPSTR IpszKontonr, long IKontoart, BOOL bShowErr, long IAusgabeart, LPSTR IpszSaldo, LPSTR IpszEuroSaldo, LPSTR pszPerDatum, long IDimension, LPSTR IpszKostref)

Dient der Ermittlung des Saldos eines gültigen Kontos, ein positiver Saldo bedeutet Sollsaldo, ein negativer Habensaldo.

Parameter:

long IMandnr	Mandantenummer.
LPSTR IpszKontonr	Gültige Kontonummer (mit führenden Nullen), sollte vorher über OFCK_HoleKonto ermittelt sein.
long IKontoart	Kontoart (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_KOST für Kostenstellen/Kostenträger).
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen.
long IAusgabeart	Ausgabeart des Saldos (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, Beschreibung der Ausgabearten s. u.).
LPSTR IpszSaldo {16}	Enthält bei Erfolg den Saldo (in Landeswährung) im gewünschten Format. Es müssen mindestens 16 Byte für diesen Rückgabewert reserviert werden.
LPSTR IpszEuroSaldo {16}	Enthält bei Erfolg den Saldo in Euro im gewünschten Format. Es müssen mindestens 16 Byte für diesen Rückgabewert reserviert werden.
LPSTR IpszperDatum {16}	Wird in diesem optionalen Eingabeparameter ein Datum im Format TTMMJJ, TT.MM.JJ oder TT.MM.JJJJ übergeben, so wird der Saldo zu diesem Stichtag ermittelt. Ausnahme bei Kostenstellen (OFCK_KOST): hier wird bei Übergabe eines Datums der Saldo bis zum Monatsletzten des übergebenen Monats ermittelt (z.B. bei 24.08.2010 wird bis Ende August summiert)
long IDimension	Einschränkung auf die Dimension bei Kostenstellen.
LPSTR IpszKostref	Einschränkung auf die Kostenstellenreferenz bei Dimension=4 und aktivierter Kostenstellenreferenzierung für den Mandant.

Rückgabewert:

long lErrno OFCK_OK, falls in IpszKnrOut eine ausgewählte Kontonummer zurückgegeben wird; 0 bei Dialogabbruch; Fehlernummer < 0.

Ausgabearten:

OFCK_VARLEN	Der Saldo wird mit einer variablen Länge ausgegeben, hat folgendes Format: mN...N,NN, wobei m ein Minus bei negativem Betrag enthält. Ein Saldo von -1935,73 (ein Habensaldo) erzeugt die Ausgabe "-1935,73", ein Saldo von +145 (Sollsaldo) würde "145,00" erzeugen.
OFCK_FIXLEN	Der Saldo wird mit einer festen Länge von 13 Zeichen ausgegeben und hat folgendes Format: vNNNNNNNNNN,NN wobei v das Vorzeichen der Zahl enthält ('+', '-'). Falls der Saldo den Betrag 0 hat, enthält v ebenfalls eine '0', in IpszSaldo steht dann "000000000,00". Der Saldo 125,60 würde als "+000000125,60" ausgegeben.
OFCK_AUSGABE	Der Saldo wird im syska EURO FIBU - Ausgabeformat ausgegeben, das sich vom OFCK_VARLEN-Format nur durch die Tausendertrennung mit Punkten ('.') unterscheidet. Ein Saldo von -1935,73 erzeugt die Ausgabe "-1.935,73", ein Saldo von 0 erzeugt "0,00".

OFCK_HoleUmsatz

long WINAPI **OFCK_HoleUmsatz**(long IMandnr, LPSTR IpszKontonr, long IKontoart, long IZeitraum, long IJahr, BOOL bShowErr, long IAusgabeart, LPSTR IpszSollumsatz, LPSTR IpszHabenumsatz, LPSTR IpszUmsatzsaldo, LPSTR IpszEuroSoll, LPSTR IpszEuroHaben, LPSTR IpszEuroSaldo, long IDimension, LPSTR IpszKostref)

Dient der Ermittlung der Umsätze eines gültigen Kontos in einem oder wahlweise bis zu einem bestimmten Zeitraum. Für Kostenstellen/träger kann nur der Umsatzsaldo ermittelt werden, für die anderen Kontoarten auch der Soll- und Habenumsatz einzeln.

Parameter:

long IMandnr	Mandantenummer.
LPSTR IpszKontonr	Gültige Kontonummer (mit führenden Nullen), sollte vorher über OFCK_HoleKonto ermittelt sein.
long IKontoart	Kontoart (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren,

	OFCK_DEBI für Debitoren, OFCK_KOST für Kostenstellen/träger)
long IZeitraum	Zeitraum, für den der Umsatz berechnet werden soll. Der Zeitraum bezieht sich stets auf den Geschäftsjahresbeginn des gewählten Mandanten. Umsätze können monatsweise ermittelt werden: IZeitraum = 1...12 (1 = erster Monat des Geschäftsjahres, 2 = zweiter Monat usw.), IZeitraum = 13...16 (13 = erstes Quartal des Geschäftsjahres, 14 = zweites Quartal usw.), IZeitraum = 17 (Eröffnungsumsätze), IZeitraum = 18 (Umsatz aller Monate des Geschäftsjahres), IZeitraum = 19 (Abschlussumsätze), IZeitraum = 20 (Gesamtjahresumsätze mit Eröffnungs- und Abschlussumsätzen). Bei einem abweichenden Geschäftsjahr, das z.B. im April 1998 beginnt, wird bei einer Zeitraumangabe von 2 der Umsatz des Monats Mai zurückgegeben. Durch Addition von 32 zu IZeitraum werden - sofern sinnvoll - die Eröffnungswerte mit einbezogen. Bei Addition von 64 werden die Abschlusswerte mit einbezogen. Bei Addition von 128 wird der Umsatz vom Geschäftsjahresbeginn bis einschließlich des gewählten Zeitraumes (kumuliert) ermittelt. Die additiven Parameter können kombiniert werden.
long IJahr	Falls > 0: gewünschtes Geschäftsjahr (z. B. 1997), falls -1: aktuelles Geschäftsjahr des Mandanten. Bei einem abweichenden Geschäftsjahr ist als Jahreszahl immer das Jahr des Geschäftsjahresbeginns anzugeben.
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen.
long IAusgabeart	Ausgabeart der Beträge (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, siehe Beschreibung der Ausgabearten bei Funktion OFCK_HoleSaldo).
LPSTR IpszSollumsatz {16}	Falls ungleich NULL: enthält bei Erfolg den Sollumsatz (in Landeswährung) im gewünschten Format. Es müssen mindestens 16 Byte für diesen Rückgabewert reserviert werden. Dieser Wert kann nicht für Kostenstellen/träger ermittelt werden.
LPSTR IpszHabenumsatz {16}	Falls ungleich NULL: enthält bei Erfolg den Habenumsatz im gewünschten Format. Es müssen mindestens 16 Byte für diesen Rückgabewert reserviert werden. Dieser Wert kann nicht für Kostenstellen/träger ermittelt werden.
LPSTR IpszUmsatzsaldo {16}	Falls ungleich NULL: enthält bei Erfolg den Saldo im gewünschten Format. Es müssen mindestens 16 Byte für diesen Rückgabewert reserviert werden.
LPSTR IpszEuroSoll {16}	Falls ungleich NULL werden hier die Beträge in Euro zurückgegeben; es müssen jeweils mind. 16 Byte reserviert werden.
LPSTR IpszEuroHaben {16}	
LPSTR IpszEuroSaldo {16} {16} {16}	
long IDimension	Einschränkung auf die Dimension bei Kostenstellen.
LPSTR IpszKostref	Einschränkung auf die Kostenstellenreferenz bei Dimension=4 und aktivierter Kostenstellenreferenzierung für den Mandant.

Rückgabewert:

long IErrno OFCK_OK, falls der Umsatz des Konto ermittelt werden konnte; sonst Fehlernummer <= 0.

OFCK_ExistiertKuerzel

long WINAPI **OFCK_ExistiertKuerzel**(long IMandnr, LPSTR IpszKuerzel, long IKontoart)

Prüft, ob ein gegebenes Kürzel existiert.

Parameter:

long IMandnr Mandantenummer.
 LPCTSTR IpszKuerzel Kürzel, das geprüft werden soll.
 long IKontoart Kontoart (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_KOST für Kostenstellen/träger).

Rückgabewert:

long IErrno OFCK_OK, falls das Konto existiert; falls es nicht existiert:

OFCK_Sako

long WINAPI **OFCK_Sako**(long ISakoUpdateFlag, long IAktion, BOOL bShowErr, long IMandnr, LPCSTR szKontonr, LPCSTR szKuerzel, LPCSTR szName, LPCSTR szEroeffdatum, long IZuordnung, BOOL IDBKonto, BOOL IUmsatzerloese, BOOL IKostenart, LPCSTR szWaehrung, long ISteuerart, long ISteuerkennung, LPCSTR szSteuersatz, long IVorsteuerkennung, long IUstvaKz, BOOL IDatevAutomatik, long IKostenschablone, BOOL IBuchungssperre, long IEuerKz), long IUst2aKz, long IFormularKennung)

Mit dieser Funktion lassen sich Sachkonten anlegen oder ändern. Eine Neuanlage ist ab dem Eröffnungsdatum möglich, eine Änderung nur im aktuellen (neuesten) Jahr. Das zu ändernde Konto wird anhand der Kontonummer identifiziert. Das Eröffnungsdatum lässt sich nicht ändern. Die Kontowährung wird nur bei schweizerischen Mandanten beachtet, bei allen anderen Mandanten muss sie leer sein. Ist das Konto nicht löscherbar, so lässt sich die Währung nicht mehr ändern. Die Steuerkennung „Teilsteuersatz“ ist nur bei schweizerischen Mandanten zulässig. Auch die Vorsteuerkennung ist nur in der Schweiz von Belang. Die Kennung für die DATEV-Automatik ist nur in Deutschland interessant.

Es sind die Zusammenhänge aus der Fibu zu beachten, es werden nicht alle möglichen Kombinationen akzeptiert. Parameter, deren Werte irrelevant sind (z.B. DATEV-Automatik außerhalb Deutschland, Eröffnungsdatum bei einem Update) müssen dennoch angegeben werden, allerdings darf der angegebene Wert beliebig sein.

Parameter:

long ISakoUpdateFlag	Mittels des UpdateFlags wird gesteuert, welche Spalten/Daten des Sachkontos bei einem Update überschrieben werden sollen. Auf diese Weise können Felder, die nur in der Fibu gepflegt werden, stehen gelassen werden. Die einzelnen Flags, die verodert übergeben werden, sind in der OFCK.H und in den Include-Dateien zu anderen Programmiersprachen definiert. Mittels der Konstanten OFCK_SKU_UPDATE_FLAG_ALLE können alle Spalten aktualisiert werden.
long IAktion	Die Art der Aktion. Mögliche Werte: OFCK_INSERT oder OFCK_UPDATE.
BOOL bShowErr	Mit dieser Option werden Fehler auch per Dialog angezeigt, statt nur über den Rückgabewert. Sehr empfehlenswert. Mögliche Werte: 0 (FALSE), ungleich 0 (TRUE).
long IMandnr	Die Mandantenummer. Gültige Werte lassen sich per OFCK_HoleMandantNr() ermitteln.
LPCSTR szKontonr	Kontonummer. Sie darf maximal 7-stellig sein und darf nur aus Ziffern bestehen. Sie muss pro Mandant eindeutig sein.
LPCSTR szKuerzel	Kürzel: Es darf höchstens 8-stellig sein. Die Fibu-Option „Doppelte Kürzel zulassen“ wird beachtet.
LPCSTR szName	Name: Es werden maximal 40 Stellen herangezogen. Es muss ein Name angegeben werden.
LPCSTR szEroeffdatum	Nur Anlage: Eröffnungsdatum. Das Konto wird im Geschäftsjahr, welches das Eröffnungsdatum enthält, angelegt und in allen weiteren. Akzeptierte Formate: TTMMJJ, TT.MM.JJ, TT.MM.JJJJ.
long IZuordnung	Die Zuordnung zu den Klassen Bilanz-Aktiva, Bilanz-Passiva, GUV, Ohne Zuordnung. Mögliche Werte: OFCK_BILAKTIV, OFCK_BILPASSIV, OFCK_GUNDEV, OFCK_OHNE.
BOOL IDBKonto	Markiert ein Konto als Durchbuchungskonto: Mögliche Werte: 0 (kein DB-Konto), ungleich 0 (DB-Konto).
BOOL IUmsatzerloese	Markiert ein Konto als Umsatzerlöskonto: Mögliche Werte: 0 (kein Umsatzerlöskonto), ungleich 0 (Umsatzerlöskonto).
BOOL IKostenart	Markiert ein Konto als Kostenart: Mögliche Werte: 0 (keine Kostenart), ungleich 0 (Kostenart).

LPCSTR szWaehrung	Nur Schweiz: Kontowährung. Maximal 4 Stellen. Bei nicht-löschbaren Konten kann die Währung nicht geändert werden.
long ISteuerart	Die Steuerart. Möglichkeiten: Vorsteuer, Umsatzsteuer, keine Steuerpflicht. Mögliche Werte: OFCK_VORSTEUER, OFCK_UMSATZSTEUER, OFCK_KEINESTEUERPFLICHT.
long ISteuerkennung	Die Steuerkennung: Es können verschieden Steuersätze gewählt werden: Regelsatz, ermässiger Satz, Nullsteuersatz, sonstiger Satz, EG-Steuersatz und (nur Schweiz) Teilsteuersatz. Die dazu passenden Konstanten sind: OFCK_REGELSTEUERSATZ, OFCK_ERMSTEUERSATZ, OFCK_NULLSTEUERSATZ, OFCK_MISCSTEUERSATZ, OFCK_EGSTEUERSATZ, OFCK_TEILSTEUERSATZ (nur Schweiz).
LPCSTR szSteuersatz	Der Steuersatz. Er muss angegeben werden, wenn die Steuerkennung OFCK_MISCSTEUERSATZ, OFCK_EGSTEUERSATZ oder OFCK_TEILSTEUERSATZ gewählt wird. In den anderen Fällen wird die Angabe ignoriert. Es sind bis zu 2 Vorkommastellen und bis zu zwei Nachkommastellen zulässig. Der Dezimaltrenner richtet sich nach den Windows-Einstellungen.
long IVorsteuerkennung	Nur Schweiz: Markiert ein vorsteuerpflichtiges Konto als Ware/Dienstleistung oder als Investition. Mögliche Werte: OFCK_INVESTITION, OFCK_WAREDIENST.
long IUstvaKz	Ordnet ein Konto einer UStVA/UVA-Kennziffer zu. Es können nur einige grundlegende Konsistenzprüfungen vorgenommen werden. Nur Schweiz: Da manche Konten zu mehreren Ziffern beitragen, können zwei Kennziffern angegeben werden, im Format Ziffer1 + 100 * Ziffer2. So trägt z.B. die Ziffer 1070 im Kontenstamm in der UVA zu Ziffern 10 und 70 bei. In diesem Beispiel gleichwertig wäre die 7010.
BOOL IDatevAutomatik	Nur Deutschland: Kennzeichnet ein Konto als DATEV-Automatikkonto. Mögliche Werte: 0 (keine DATEV-Automatik), ungleich 0 (DATEV-Automatik).
long IKostenschablone	Die Nummer einer Kostenschablone. Die Existenz der angegebenen Schablone wird überprüft. Soll keine Kostenschablone verwendet werden, so muss eine 0 angegeben werden. Gültige Werte können mit OFCK_HoleKostenschablone() in Erfahrung gebracht werden.
BOOL IBuchungssperre	Gibt eine Buchungssperre (1) an. Sonst 0.
long IEuerKz	DE: Gibt die Kennzahl für das EÜR-Formular an. AT: ESt.Kennziffer. CH: 2. MWSt-Kennziffer.
long IUst2aKz	DE: Gibt die Kennziffer für die Umsatzsteuererklärung an.
long IFormularKennung)	DE: Gibt die Formularkennung für die Umsatzsteuererklärung an.
<u>Rückgabewert:</u>	
long IErrno	OFCK_OK: Operation erfolgreich <0: Fehlernummer

8. Buchungen

OFCK_HoleStapelNr

long WINAPI **OFCK_HoleStapelNr**(HWND hParent, long IMandnr, LPSTR lpszStapelName)

Dient der Ermittlung der Nummer des zu bebuchenden Stapels. Sie sollte im aufrufenden Programm gespeichert werden. Es erscheint ein Dialog zur Stapelauswahl mit allen auswählbaren Stapeln. Dieser gleicht der Mandantenauswahl. Für den Rückgabewert des Stapelnamens müssen die in {}-Klammern angegebene Anzahl an Bytes reserviert werden.

Parameter:

HWND hParent Handle des Parent-Fensters unter dem der Dialog zur Stapelauswahl erscheinen soll.

long IMandnr Mandantenummer.

LPSTR Falls ungleich NULL: Enthält nach erfolgreichem Aufruf den ermittelten Stapelnamen. Es müssen mindestens 41 Byte für diesen Rückgabewert reserviert werden.

Rückgabewert:

long lStapelNr Stapelnummer > 0; 0 bei Dialogabbruch; Fehlernummer < 0.

OFCK_BuchungEinfuegen

long WINAPI **OFCK_BuchungEinfuegen**(LPOFCKBUC lpBuc, BOOL bShowErr)

Fügt die in lpBuc beschriebene Buchung in den darin referenzierten Stapel ein. Das eigentliche Verbuchen findet erst beim Verbuchen des Stapels in der Fibu statt, so dass dort noch die Möglichkeit besteht die Buchung zu korrigieren oder zu löschen.

Parameter:

LPOFCKBUC lpBuc Zeiger auf die Struktur mit den Buchungsangaben.

BOOL bShowErr TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno OFCK_OK oder Fehlernummer < 0.

Die Buchungsstruktur ist wie folgt aufgebaut (siehe OFCK.H, OFCK.BAS bzw. OFCK.PAS):

```
typedef struct
{
    long   mandnr;           // Mandantenummer > 0
    long   stapelnr;        // Stapelnummer > 0
    long   kostenschablone; // ungleich 0 <=> Kostenschablonenkennung
    long   zahlungsziel;   // ungleich 0 <=> Zahlungszielkennung
    long   isteb;           // optional; != 0 bei Eröffnungsbuchung
    long   istab;           // optional; != 0 bei Abschlussbuchung
    long   sollart;         // Kontoart des Sollkontos
    long   habenart;        // Kontoart des Habenkontos
    long   ustdirekt;       // ungleich 0 <=> wenn bei einem Ist-Versteuerer
                          // die Umsatzsteuer doch direkt gebucht werden
                          // soll
    long   istub;           // optional; != 0 bei einer Umbuchung
    char   belegdatum[7];  // "TTMMJJ"
    char   belegnr[17];    // beliebiger String
    char   text[36];       // Buchungstext
    char   sollkonto[8];   // max. 7-stelliges Sollkonto (z.B. 0210)
                          // aus (OFCK_SAKO, OFCK_KRED, OFCK_DEBI)
    char   habenkonto[8];  // max. 7-stelliges Habenkonto
                          // aus (OFCK_SAKO, OFCK_KRED, OFCK_DEBI)
}
```

```

char bruttobetrag[13]; // 000000000,00 // mit Dezimaltrenner gem.
                        // Ländereinstellung von Windows
char steusa[6];        // optional: falls abweichend oder sonstiger
                        // Steuersatz
char steuerbetrag[13]; // optional: falls vom errechneten abweichend
char opbelegnr[17];    // OP- bzw. Ursprungsbelegnr.
char valutadatum[7];  // "TTMMJJ"
char waehrung[4];     // 3-stelliges Kürzel
char esr[50];         // komplette ESR-Nummer (nur in Schweiz wichtig,
sonst leer)
char text2[36];       // 2. Buchungstext
char szDmsID[51];     // DMS-ID
} OFCK_BUCHUNG;

```

Zum Füllen der Struktur gelten die gleichen Grundsätze wie beim Personenkonto.

Wird in der Kennung 'isteb' ein Wert ungleich 0 übergeben, so wird die übergebene Buchung als Eröffnungsbuchung gebucht. In allen anderen Fällen ist diese auf 0 zu setzen.

Der Euro wird im gesamten OFCK nicht als Fremdwährung betrachtet.

Die Reihenfolge der Währungserkennung (Feld 'waehrung'):

1. Wird die Währung mit übergeben (z.B. „EUR“ oder „USD“), so wird immer in dieser Währung gebucht. Eine evtl. in einem beteiligten Personenkonto hinterlegte Währung wird dann ignoriert.
2. Wird keine Währung übergeben, aber in einem der beteiligten Personenkonten ist eine Währung hinterlegt, so wird in der Währung des Personenkontos gebucht.
3. Wird keine Währung übergeben und ist keine Fremdwährung in den beteiligten Konten hinterlegt, so wird in Grundwährung gemäß der Grundwährungseinstellung im Firmenstamm gebucht (also in der jeweiligen Landeswährung oder in EUR).

Bemerkungen:

- Bei reinen Sachkontenbuchungen können keine Fremdwährungen übergeben werden. Deshalb ist die Buchungswährung entweder die angegebene Währung oder sie wird aus dem Firmenstamm bestimmt.
- Soll definitiv in EURO gebucht werden, so ist das Währungskürzel „EUR“ anzugeben.
- Werden zwei Personenkonten mit unterschiedlich hinterlegter Fremdwährung bebucht und es ist keine Währung angegeben, so kommt eine Fehlermeldung und die Buchung wird abgebrochen.

Eine Eröffnungsbuchung ist immer steuerfrei und muss ein leeres Kontonummernfeld (0) in der Buchungsstruktur enthalten - dieses wird vom OFCK automatisch auf das Saldovortragskonto (i.d.R. Kontonr. 9000) gesetzt. Wird z.B. eine Eröffnungsbuchung für das Kassenkonto übergeben, so sollte diese im Feld 'sollkonto' stehen und das Feld 'habenkonto' leer sein. Das Belegdatum muss bei Eröffnungsbuchungen auf den Geschäftsjahresbeginn gestellt werden.

Mit dem Feld 'istub' kann eine Buchung als Umbuchung gekennzeichnet werden. Die Überprüfung erfolgt allerdings erst beim Verbuchen in der syska EURO FIBU.

Das Feld 'esr' ist nur für die Schweizer Version wichtig. Hier kann eine ESR-Nummer übergeben werden, die so in den OP übernommen wird und später zur Zuordnung von Zahlungsbuchungen verwendet werden kann. Es muss die komplette ESR-Nummer übergeben werden, bereinigt um Leer- und Sonderzeichen (also nur Ziffern). Es werden alle ESR- und ESR+-Nummern unterstützt.

Bei der Übergabe von Buchungen wird nun für die OP-Belegnummer, sofern keine angegeben wurde, die Belegnummer verwendet.

Die DMS-ID wird nicht auf Gültigkeit überprüft.

Das Feld zur Übergabe der DMS-ID enthält bei Einsatz der DMS-Schnittstelle die Beleg-ID des DMS-Systems.

Bei Einsatz des Mini-DMS der EURO FIBU ist hier der Dateiname des Belegs anzugeben (z.B. „AR400.tif“). Dieser muss sich im Ablageverzeichnis befinden, welches in den Grundeinstellungen zum DMS in der Fibu zu hinterlegen ist. Dieses sollte in der Fibu auf das Dokumentenverzeichnis der integrierten Anwendung eingestellt werden, so dass die Datei nicht umkopiert werden muss. Falls ein Jahresunterverzeichnis oder eine andere Verzeichnisstruktur genutzt wird, muss bei OFCK-Übergabe der Dateiname um den relativen Pfad unterhalb des Ablageverzeichnisses ergänzt werden (z.B. "2011\AR500.pdf").

OFCK_BuchungEinfuegenDirekt

long WINAPI **OFCK_BuchungEinfuegenDirekt**(BOOL bShowErr, long IMandnr, long IStapelnr, LPSTR IpszBelegdatum, LPSTR IpszBelegnr, LPSTR IpszText, LPSTR IpszSollkonto, long ISollart, LPSTR IpszHabenkonto, long IHabenart, LPSTR IpszBruttobetrag, LPSTR IpszSteusa, LPSTR IpszSteuerbetrag, BOOL IUstdirekt, long IKostenschablone, long IZahlungsziel, BOOL IInsteb, BOOL IInstab, LPSTR IpszOpbelegnr, LPSTR IpszValutadatum, LPSTR IpszWaehrung, BOOL IInstub, LPSTR szEsr, LPSTR IpszText2, LPSTR pszDmsID)

Diese Funktion entspricht der OFCK_BuchungEinfuegen-Funktion, nur werden die Buchungs-Parameter direkt übergeben. Die einzelnen Parameter entsprechen denen in der OFCK_Buchung-Struktur. Das Belegdatum kann jedoch auch in der Form TT.MM.JJJJ oder TT.MM.JJ übergeben werden.

Grundsätzliches zu Splitbuchungen

Eine Splitbuchung enthält mindestens eine Teilbuchung. Die Anzahl der übergebenen Teilbuchungen muss in der Strukturvariablen 'teilmuchungsanz' übergeben werden. Das Feld 'teilmuchung' muss dann genau diese Anzahl an sinnvollen Teilbuchungen enthalten. Falls die Splitbuchung nur eine Teilbuchung enthält, wird sie als normale Buchung gewertet, wodurch eine einheitliche Übergabe von Buchungen und Splitbuchungen möglich ist.

Jede Teilbuchung muss ein Soll- und Habenkonto enthalten. Die Art der Splitbuchung wird automatisch ermittelt. Bei einer Habensplitbuchung (Split im Haben = verschiedene Habenkonto) beispielsweise muss das Sollkonto in jeder Teilbuchung gleich sein und auf der Habenseite müssen verschiedene Konten stehen. Es dürfen sowohl alle Sollkonten als auch alle Habenkonto untereinander gleich sein. Da die Art der Splitbuchung (ob Soll- oder Habensplit) in diesem Fall nicht ermittelt werden kann, wird ein Habensplit angenommen. Soll eine solche Buchung mit einem Sollsplit übergeben werden, so kann für die nicht gesplittete Seite ab der 2. Teilbuchung auf die Angabe der Kontonummer verzichtet werden (d.h. die Kontonummer wird leer übergeben).

Eine Splitbuchung kann immer nur in einer Währung übergeben werden. Die Reihenfolge der Währungserkennung ist die gleiche wie bei der Funktion OFCK_BuchungEinfuegen (s. vorheriges Kapitel).

Achtung! Da die Umsatzsteuer in der Fibu zur korrekten Erzeugung der Umsatzsteuervoranmeldung für jede Teilbuchung separat ermittelt werden muss, kann es hierbei zu Rundungsdifferenzen kommen, sofern in Ihrer Anwendung die Steuer aus der Gesamtnettosumme errechnet wird. Dies ist z.B. bei Ausgangsrechnungen auf verschiedene Erlöskonto mit gleichem Steuersatz der Fall. Vor der Übergabe einer solchen Buchung sollten Sie daher prüfen, ob die, in Ihrer Anwendung aus der Gesamtsumme ermittelte, Umsatzsteuer von der Summe der einzelnen Steuerbeträge abweicht. Ist dies der Fall, so sollten Sie eine separate Teilbuchung auf ein steuerfreies Erlöskonto über die Differenz mit übergeben, damit der Offene Posten in der Fibu mit der Rechnungssumme übereinstimmt.

Mit der Funktion OFCK_SplitBuchungEinfuegen ist es möglich, eine Splitbuchung auf einmal zu übergeben. Allerdings ist diese Variante auf Splitbuchungen mit max. 10 Teilbuchungen beschränkt. Die Übergabe einer Splitbuchung kann aber auch in Form mehrerer einzelner Funktionsaufrufe erfolgen, was bei Splitbuchungen mit mehr als 10 Teilbuchungen dann auch gemacht werden muss. Dazu verwenden Sie die drei Funktionen (OFCK_SplitStart, OFCK_SplitEinfuegen, OFCK_SplitEnde), mit denen ein sukzessives Füllen einer OFCK_SPLITBUCHUNGS-Struktur möglich ist. Nur hier ist es außerdem möglich, mit Hilfe von OFCK_KobuEinfuegen einzelne Kostenbuchungen zu jeder Splitteilbuchung nach OFCK_SplitEinfuegen einzufügen.

Bei der Übergabe von Buchungen wird nun für die OP-Belegnummer, sofern keine angegeben wurde, die Belegnummer verwendet.

Eine Buchung die in jedem Split dieselben Konten enthält ist zulässig. In diesem Fall wird ein Habensplit angenommen.

OFCK_SplitBuchungEinfuegen

long WINAPI **OFCK_SplitBuchungEinfuegen**(LPOFCKSPLITBUC lpSplitBuchung, BOOL bShowErr)

Fügt die in IpSplitBuc beschriebene Splitbuchung in den darin referenzierten Stapel ein. Das eigentliche Verbuchen findet erst beim Verbuchen des Stapels in der Fibu statt, so dass dort noch die Möglichkeit besteht, die Buchung zu korrigieren oder zu löschen. Wenn die Splitbuchung nur eine Teilbuchung enthält, wird sie als normale Buchung gewertet.

Parameter:

LPOFCKSPLITBUC IpSplitBuc Zeiger auf die Struktur mit den Splitbuchungsangaben.
 BOOL bShowErr TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno OFCK_OK oder Fehlernummer < 0.

Die Splitbuchungsstruktur ist wie folgt aufgebaut (siehe OFCK.H, OFCK.BAS bzw. OFCK.PAS):

```
typedef struct
{
    long mandnr;           // Mandantenummer > 0
    long stapelnr;        // Stapelnummer > 0
    long teilbuchungsanz; // Anzahl der Teilbuchungen in einer
                        // Splitbuchung (max. 10)
    long ustdirekt;       // ungleich 0 <=> wenn bei einem Ist-Versteuierer
                        // die Umsatzsteuer doch direkt gebucht werden
                        // soll
    long zahlungsziel;    // ungleich 0 <=> Zahlungszielkennung
    long isteb;           // optional; != 0 bei Eröffnungsbuchung
    long istab;           // optional; != 0 bei Abschlussbuchung
    long istub;           // optional; != 0 bei einer Umbuchung
    char belegdatum[7];   // "TTMMJJ"
    char belegnr[17];     // beliebiger String
    char opbelegnr[17];   // OP- bzw. Ursprungsbelegnr.
    char valutadatum[7]; // "TTMMJJ"
    char waehrung[4];     // 3-stelliges Kürzel
    char esr[50];         // komplette ESR-Nummer
                        // (nur in Schweiz wichtig, sonst leer)
    char szDmsID[51];     // DMS-ID
    OFCK_TEILBUCHUNG teilbuchung[10]; // einzelne Teilbuchungen
} OFCK_SPLITBUCHUNG;
```

Die einzelnen Teilbuchungen (mind. eine, höchstens zehn) werden vom kleinsten Index an fortlaufend im Feld 'teilbuchung' innerhalb der Splitbuchungsstruktur übergeben. Die Währung gilt für die gesamte Splitbuchung inkl. der eventuellen Kostenbuchungen.

Die Teilbuchungsstruktur ist wie folgt aufgebaut (siehe OFCK.H, OFCK.BAS bzw. OFCK.PAS):

```
typedef struct
{
    long sollart;         // Kontoart des Sollkontos
                        // aus (OFCK_SAKO, OFCK_KRED, OFCK_DEBI)
    long habenart;       // Kontoart des Habenkontos
                        // aus (OFCK_SAKO, OFCK_KRED, OFCK_DEBI)
    long kostenschablone; // ungleich 0 <=> Kostenschablonenkennung
    char sollkonto[8];    // max. 7-stelliges Sollkonto (z.B. 0210)
    char habenkonto[8];   // max. 7-stelliges Habenkonto
    char bruttobetrag[13]; // 00000000,00 // mit Dezimaltrenner gem.
                        // Ländereinstellung von Windows
    char dummy1;         //
    char steusa[6];      // optional: falls abweichend oder sonstiger
                        // Steuersatz
    char steuerbetrag[13]; // optional: falls vom errechneten abweichend
    char dummy2;         //
    char text[36];       // Buchungstext
    char text2[36];      // 2. Buchungstext
} OFCK_TEILBUCHUNG;
```

Zum Füllen der Strukturen gelten die gleichen Grundsätze wie beim Personenkonto.

OFCK_SplitStart

long WINAPI **OFCK_SplitStart**(long IMandnr, long IStapelnr, LPSTR IpszBelegdatum, LPSTR IpszBelegnr, BOOL IUstdirekt, long IZahlungsziel, LPSTR IpszOpbelegnr, LPSTR IpszValutadatum, LPSTR IpszWaehring, BOOL Ilstub, LPSTR szEsr, LPSTR szDmsID)

Mit dieser Funktion wird das Füllen einer vom OFCK geführten OFCK_SPLITBUCHUNG-Struktur eingeleitet. Die einzelnen Parameter entsprechen denen in der OFCK_SPLITBUCHUNG-Struktur. Das Belegdatum und Valutadatum kann auch in der Form TT.MM.JJJJ oder TT.MM.JJ übergeben werden. Da mit dieser Funktion nur eine Struktur gefüllt wird, werden praktisch keine Parameter außer dem Belegdatum überprüft. Die Parameterprüfung geschieht erst beim Aufruf der OFCK_SplitEnde-Funktion. Falls diese Funktion einen Fehler zurückgibt, muss die gestartete Splitbuchung nicht mit OFCK_SplitEnde abgebrochen werden.

Rückgabewert:

long IErrno OFCK_OK oder Fehlernummer < 0.

OFCK_SplitEinfuegen

long WINAPI **OFCK_SplitEinfuegen**(LPSTR IpszSollkonto, long ISollart, LPSTR IpszHabenkonto, long IHabenart, LPSTR IpszBruttobetrag, LPSTR IpszSteusa, LPSTR IpszSteuerbetrag, LPSTR IpszText, long IKostenschablone, LPSTR IpszText2)

Diese Funktion ermöglicht die Übergabe einer von mehreren Teilbuchungen nach erfolgreichem Start einer sukzessiven Splitbuchungsübergabe mittels OFCK_SplitStart. Hier können beliebig viele Teilbuchungen zu einer Splitbuchung übergeben werden. Dabei wird die Teilbuchung als nächste in die intern geführte OFCK_SPLITBUCHUNG-Struktur eingefügt und die Anzahl der Teilbuchungen erhöht. Auch hier findet praktisch keine Parameterprüfung statt, da nur Felder gefüllt werden. Endet diese Funktion trotzdem mit einem Fehler, so muss die angefangene Splitbuchung nicht mit OFCK_SplitEnde abgebrochen werden.

Rückgabewert:

long IErrno OFCK_OK oder Fehlernummer < 0.

OFCK_SplitEinfuegenEx

long WINAPI **OFCK_SplitEinfuegenEx**(LPSTR IpszSollkonto, long ISollart, LPSTR IpszHabenkonto, long IHabenart, LPSTR IpszBruttobetrag, LPSTR IpszBruttobetragEigen, LPSTR IpszSteusa, LPSTR IpszSteuerbetrag, LPSTR IpszSteuerbetragEigen, LPSTR IpszText, long IKostenschablone, LPSTR IpszText2)

Diese Funktion ermöglicht bei schweizerischen Mandanten die Übergabe von Splitbuchungen mit Fremdwährung und Eigenwährung. Diese Funktion ist alternativ zur Funktion OFCK_SplitEinfuegen (ohne Ex) zu verwenden. Die Parameter gleichen denjenigen der Funktion OFCK_SplitEinfuegen(), lediglich die Parameter IpszBruttoEigen und IpszSteuerEigen sind dazugekommen und müssen mit dem Eigenwährungsbetrag belegt werden (der Steuerbetrag nur, wenn auch der Steuerbetrag in Fremdwährung angegeben ist). Sollen Kostenbuchungen erfasst werden, so wird die Funktion OFCK_SplitKobuEinfuegen ganz normal verwendet, zu beachten ist, dass auf Grund der Angabe der Eigenwährung eine genauere Prüfung der Kostenteilbeträge mit dem Nettobetrag des Splits vorgenommen wird. Die Funktion gibt bei einem nicht-schweizerischen Mandanten den Fehler OFCKERR_Swiss_only zurück.

Rückgabewert:

long IErrno OFCK_OK oder Fehlernummer < 0.

OFCK_SplitKobuEinfuegen

long WINAPI **OFCK_SplitKobuEinfuegen**(LPSTR IpszKostenstelle1, LPSTR IpszKostenstelle2, long IKostenkennung, LPSTR IpszBetrag, LPSTR IpszBemerkung, LPSTR IpszBemerkung2, LPSTR IpszKostenstelle3, LPSTR IpszKostenstelle4, LPSTR IpszKostenstelle5, LPSTR IpszKostenstelle6)

Diese Funktion ermöglicht die Übergabe von Kostenbuchungen bei der sukzessiven Splitbuchungsübergabe. Zu jeder Splitteilbuchung können beliebig viele Kostenbuchungen übergeben werden, die im Summe exakt dem Nettobetrag der Splitteilbuchung entsprechen müssen. Die zu einer Teilbuchung gehörenden Kostenbuchungen müssen direkt nach dem Einfügen der jeweiligen Splitteilbuchung (über OFCK_SplitEinfuegen) durch evtl. mehrmaligen Aufruf dieser Funktion (OFCK_SplitKobuEinfuegen) übergeben werden.

Parameter:

LPSTR IpszKostenstelle1	Zeiger auf gültige Kostenstelle1
LPSTR IpszKostenstelle2	Zeiger auf gültige Kostenstelle2
long IKostenkennung	Kennung der Kostenbuchung (OFCK_FIX für fixe Kosten, OFCK_VARIABEL für variable Kosten).
LPSTR IpszBetrag	Betrag der Kostenteilbuchung.
LPSTR IpszBemerkung	Bemerkungstext für die Kostenbuchung. Es werden maximal 40 Zeichen berücksichtigt.
LPSTR IpszBemerkung2	2. Bemerkungstext für die Kostenbuchung. Es werden maximal 40 Zeichen berücksichtigt.
LPSTR IpszKostenstelle3	Zeiger auf gültige Kostenstelle3
LPSTR IpszKostenstelle4	Zeiger auf gültige Kostenstelle4
LPSTR IpszKostenstelle5	Zeiger auf gültige Kostenstelle5
LPSTR IpszKostenstelle6	Zeiger auf gültige Kostenstelle6

Rückgabewert:

long lErrno OFCK_OK oder Fehlernummer < 0.

OFCK_SplitKobuEinfuegenEx

long WINAPI **OFCK_SplitKobuEinfuegen**(LPSTR IpszKostenstelle1, LPSTR IpszKostenstelle2, long IKostenkennung, LPSTR IpszBetrag, LPSTR IpszMenge, LPSTR IpszEinheit, LPSTR IpszBemerkung, LPSTR IpszBemerkung2, LPSTR IpszKostenstelle3, LPSTR IpszKostenstelle4, LPSTR IpszKostenstelle5, LPSTR IpszKostenstelle6)

Diese Funktion ermöglicht die Übergabe von Kostenbuchungen bei der sukzessiven Splitbuchungsübergabe. Zu jeder Splitteilbuchung können beliebig viele Kostenbuchungen übergeben werden, die im Summe exakt dem Nettobetrag der Splitteilbuchung entsprechen müssen. Die zu einer Teilbuchung gehörenden Kostenbuchungen müssen direkt nach dem Einfügen der jeweiligen Splitteilbuchung (über OFCK_SplitEinfuegen) durch evtl. mehrmaligen Aufruf dieser Funktion (OFCK_SplitKobuEinfuegenEx) übergeben werden.

Parameter:

LPSTR IpszKostenstelle1	Zeiger auf gültige Kostenstelle1 oder Kostenstelle.
LPSTR IpszKostenstelle2	Zeiger auf gültige Kostenstelle2 oder Kostenträger.
Long IKostenkennung	Kennung der Kostenbuchung (OFCK_FIX für fixe Kosten, OFCK_VARIABEL für variable Kosten).
LPSTR IpszBetrag	Betrag der Kostenteilbuchung.
LPSTR IpszMenge	Die Menge entsprechend des Betrages.
	ACHTUNG: Das Verhältnis zwischen Betrag und Menge bei einzelnen Kostenbuchungen wird nicht überprüft; auch nicht, ob die Gesamtmenge

	richtig aufgeteilt wird. Prüfen Sie dies vor der Übergabe bitte genau.
LPSTR IpszEinheit	Die Einheit der Menge. Die Einheit muss bei allen Kostenbuchungen, welche zu einer Splitbuchung gehören, gleich sein.
LPSTR IpszBemerkung	Bemerkungstext für die Kostenbuchung. Es werden maximal 40 Zeichen berücksichtigt.
LPSTR IpszBemerkung2	2. Bemerkungstext für die Kostenbuchung. Es werden maximal 40 Zeichen berücksichtigt.
LPSTR IpszKostenstelle3	Zeiger auf gültige Kostenstelle3
LPSTR IpszKostenstelle4	Zeiger auf gültige Kostenstelle4
LPSTR IpszKostenstelle5	Zeiger auf gültige Kostenstelle5
LPSTR IpszKostenstelle6	Zeiger auf gültige Kostenstelle6
<u>Rückgabewert:</u>	
long lErrno	OFCK_OK oder Fehlernummer < 0.

OFCK_SplitEnde

long WINAPI **OFCK_SplitEnde**(BOOL bEinfuegen, BOOL bShowErr)

Mit dieser Funktion wird eine mittels OFCK_SplitStart begonnene und mit OFCK_SplitEinfuegen (und evtl. OFCK_SplitKobuEinfuegen) sukzessive übergebene Splitbuchung abgeschlossen und abhängig vom Wert des Parameters bEinfuegen versucht, in den Stapel zu schreiben oder gelöscht. Falls bEinfuegen = TRUE ist, so wird versucht, die intern geführte OFCK_SPLITBUCHUNG-Struktur mit der OFCK_SplitBuchungEinfuegen-Funktion in den Stapel zu schreiben. In diesem Falle gibt die Funktion den Rückgabewert der eigentlichen Buchungsfunktion zurück. Ist jedoch bEinfuegen = FALSE, so wird die angefangene Splitbuchung beendet und gelöscht.

Parameter:

BOOL bEinfuegen TRUE, falls die aktuelle Splitbuchung in den Stapel eingefügt werden soll, FALSE, falls die Splitbuchung abgebrochen werden soll.

BOOL bShowErr TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno OFCK_OK oder Fehlernummer < 0.

9. Holen von Datensätzen

Grundsätzliches zu Hole-Funktionen

Die Hole-Funktionen ermöglichen dem OFCK-Benutzer ein schrittweises Durchgehen durch Buchungsbelege, Sachkonten, Personenkonten, Kostenstellen/trägern und Offene Posten. Dazu muss zur Initialisierung eine der OFCK_InitHole-Funktionen aufgerufen werden (z.B. OFCK_InitHoleBube oder OFCK_InitHoleKonto). Diese Funktion sucht einen ersten Eintrag in der Datenbank, wobei zusätzlich ein Suchkriterium eingegeben werden kann (erster Tag bei Buchungsbelegen bzw. erste Kontonummer bei Konten), wobei der Schlüssel des ersten zurückgegebenen Datensatzes auch größer als dieses Kriterium sein kann (also Datum nach dem gesuchten Datum bzw. Kontonummer größer als die gesuchte Kontonummer). Die Init-Funktion gibt OFCKweitererDatensatz zurück, falls überhaupt ein dem Kriterium genügender Datensatz existiert, ansonsten wird OFCKkeinDatensatz bzw. eine Fehlermeldung zurückgegeben.

Die eigentliche Abfrage geschieht mittels der OFCK_Hole-Funktionen (OFCK_HoleBube, OFCK_HoleSako, OFCK_HolePeko, OFCK_HoleKost bzw. OFCK_HoleOP). Dies ist nur dann möglich, wenn die korrespondierende OFCK_InitHole-Funktion den Wert OFCKweitererDatensatz zurückgeliefert hat. Es ist dabei zu beachten, dass zwischen OFCK_InitHole... und OFCK_Hole... keine andere OFCK-Funktion aufgerufen werden darf. Die OFCK_Hole-Funktion gibt OFCKweitererDatensatz zurück, falls der aktuelle Datensatz mit Erfolg geholt werden konnte und ein nächster Datensatz existiert. Falls der aktuelle Datensatz der letzte war, wird OFCKletzterDatensatz zurückgegeben. Ansonsten endet der Aufruf mit Rückgabe einer Fehlermeldung. Die einzelnen OFCK_Hole-Funktionen ermöglichen dem Benutzer der Funktion die Eingabe von Zeigern auf Felder, die die Informationen aufnehmen. Falls einige der möglichen Rückgabewerte für die Anwendung irrelevant sind, können hier NULL-Zeiger übergeben werden. Alle anderen Zeiger müssen auf Puffer zeigen, die groß genug sind, um die Rückgabedaten aufzunehmen. Die benötigte Größe der Felder ist den genauen Funktionsbeschreibungen zu entnehmen. Falls die Zeiger vom Typ LPSTR sind, wird stets ein mit 0 endender String zurückgegeben.

Soll die schrittweise Abfrage von Datensätzen beendet werden, obwohl noch weitere Datensätze existieren, so muss auf jeden Fall die Funktion OFCK_BeendeHolen aufgerufen werden, da ansonsten keine andere OFCK-Funktion als OFCK_Hole... aufgerufen werden kann. Der Aufruf der Funktion OFCK_BeendeHolen ist in allen anderen Fällen (Fehler bei OFCK_InitHole..., Fehler bei OFCK_Hole..., Rückgabe von OFCKletzterDatensatz oder OFCKkeinDatensatz usw.) nicht nötig.

Bei Parametern bei denen dem Variablentyp ein Stern ,*' folgt (z.B. long *), ist ein Zeiger (bzw. die Referenz) auf die Variable zu übergeben, damit das OFCK an dieser Stelle den Wert zurückschreiben kann.

Ein Beispiel für die Anwendung der Funktionen finden Sie in Kapitel 14.

OFCK_BeendeHolen

long WINAPI OFCK_BeendeHolen()

Diese Funktion dient dem vorzeitigen Beenden (Ende der Abfrage, obwohl noch weitere Datensätze existieren) einer schrittweisen Abfrage von Buchungsbelegen, Fibu-Konten, Kostenstellen/trägern oder OPs. Sie muss nur aufgerufen werden, wenn eine OFCK_InitHole-Funktion oder eine OFCK_Hole-Funktion den Wert OFCKweitererDatensatz zurückgeliefert hat. Bei Fehlermeldungen bzw. Rückgabe von OFCKkeinDatensatz bzw. OFCKletzterDatensatz ist der Aufruf dieser Funktion nicht notwendig.

Rückgabewert:

long lErrno OFCK_OK, bei erfolgreichem Abbruch der Abfrage, < 0 sonst.

OFCK_InitHoleBube

long WINAPI **OFCK_InitHoleBube**(long lMandnr, LPSTR pszAbDatum, LPSTR pszKontonr, LPSTR pszBelegnummer, LPSTR pszBuchungsdatum, BOOL bShowErr)

Mit dieser Funktion ist es möglich die Abfrage einzelner Buchungsbelege einzuleiten. Das bedeutet, dass das OFCK darauf vorbereitet wird, bei folgenden Aufrufen der Funktion OFCK_HoleBube Buchungsbelege zu liefern. Zwischen dem Aufruf von OFCK_InitHoleBube und OFCK_HoleBube dürfen keine anderen OFCK-Funktionen außer OFCK_BeendeHolen (vorzeitiges Beenden des schrittweisen Durchgehens von Datensätzen) aufgerufen werden! Über den optionalen Parameter pszKontonr können die Buchungen zu einem bestimmten Konto geholt werden, also quasi das Kontenblatt zu einem Konto.

Mit Hilfe des Parameters pszBelegnummer werden alle Buchungsbelege zu einer Belegnummer geholt. Dabei werden alle bebuchbaren Jahre durchsucht, falls pszAbDatum leer ist. Sollen aus Performancegründen nicht alle Jahre durchsucht werden, so kann in pszAbDatum zusätzlich ein beliebiges Datum aus dem Startjahr zur Suche übergeben werden (dabei wird nur das Jahr des Datums berücksichtigt).

Mit Hilfe des Parameters pszBuchungsdatum werden alle Buchungen ab einem Buchungsdatum (Erfassungsdatum) geholt. Zuerst kommen die Buchungen ab dem Buchungsdatum aus dem ersten Geschäftsjahr, dann die Buchungen ab dem Buchungsdatum aus dem folgenden Geschäftsjahr usw. Sollen nicht alle Jahre durchsucht werden, so kann, wie bei der Suche nach Belegnummer, in pszAbDatum ein Datum aus dem Startjahr übergeben werden.

Wird zu einer Belegnummer oder ab einem Buchungsdatum geholt und ist ein Belegdatum angegeben, so wird die Jahresangabe als Startjahr der Suche gewertet. Auf diese Weise werden nicht alle vorhandenen Jahre durchsucht (Performance).

Parameter:

long lMandnr	Mandantenummer.
LPSTR pszAbDatum	Belegdatum des ersten gewünschten Buchungsbeleges. Es wird der erste Buchungsbeleg gesucht, dessen Belegdatum größer oder gleich lpszAbDatum ist. Falls der erste Buchungsbeleg des aktuellen Buchungsjahres gesucht wird, kann lpszAbDatum = NULL sein oder auf einen Leerstring zeigen.
LPSTR pszKontonr	Optional: falls ungleich NULL, muss hier eine Kontonummer stehen, zu welcher die Buchungen geholt werden sollen. Nur in Kombination zum Belegdatum.
LPSTR pszBelegnummer	Optional: falls ungleich NULL: Es werden Buchungen zu dieser Belegnummer herausgesucht. Es werden alle bebuchbaren Jahre durchgegangen, sofern nicht in pszAbDatum ein Startjahr angegeben ist.
LPSTR pszBuchungsdatum	Optional, falls ungleich NULL: Es werden Buchungen ab diesem Buchungsdatum (Erfassungsdatum) herausgesucht. Es werden alle bebuchbaren Jahre durchgegangen, sofern nicht in pszAbDatum ein Startjahr angegeben ist.
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno	OFCKweitererDatensatz, falls es einen dem Kriterium lpszAbDatum genügenden Buchungsbeleg gibt, OFCKkeinDatensatz, falls ein solcher Buchungsbeleg nicht existiert; OFCKERR_nicht_moeglich: Das Holen nach den Kriterien „Konto und Buchungsdatum“ oder nach „Konto und Belegnummer“ (jeweils gemeinsam) ist nicht möglich; < 0 bei sonstigen Fehlern.
-------------	---

OFCK_HoleBube

long WINAPI **OFCK_HoleBube**(long* plSplitnr, LPSTR lpszBelegdatum, LPSTR lpszBelegnr, LPSTR lpszText, LPSTR lpszSollkonto, long* plSollart, LPSTR lpszHabenkonto, long* plHabenart, long lAusgabeart, LPSTR lpszBetrag, LPSTR lpszSteuersatz, LPSTR lpszSteuerbetrag, LPSTR

lpzEurobetrag, LPSTR lpzEurosteuerbetrag, long* plKobus, LPSTR lpzBuchungsdatum, long* plBuchungsnummer, LPSTR pszOPBelegnummer, LPSTR pszValutadatum, LPSTR pszBenutzer, LPSTR pszZahlziel, LPSTR lpzText2, LPSTR pszDmsID)

Diese Funktion dient der schrittweisen Abfrage der relevanten Daten von Buchungsbelegen. Sie darf nur nach erfolgreichem Aufruf der Funktion OFCK_InitHoleBube (also Rückgabewert OFCKweitererDatensatz) benutzt werden. Sämtliche Parameter können mit NULL belegt werden, falls einzelne Daten für die Anwendung irrelevant sind. Nur der Parameter lAusgabeart ist zwingend, falls lpzBetrag, lpzSteuersatz, lpzSteuerbetrag lpzEurobetrag oder lpzEurosteuerbetrag ungleich NULL sind. Die Zeiger ungleich NULL müssen auf Puffer zeigen, die groß genug sind, um die Daten aufzunehmen. Die benötigte Größe eines Puffers steht in geschweiften Klammern direkt hinter dem Parameternamen. Long-Variablen benötigen 4 Bytes.

Parameter:

long* plSplitnr	Zeiger (Referenz) auf ein Long: Falls die Buchung Teil einer Splitbuchung ist, wird dieser Parameter mit einer Zahl >= 1 belegt. Diese Zahl sagt aus, die wievielte Teilbuchung diese Buchung in der Splitbuchung ist. Ansonsten wird 0 zurückgegeben.
LPSTR lpzBelegdatum{11}	Gibt das Belegdatum im Format TT.MM.JJJJ zurück.
LPSTR lpzBelegnr{17}	Belegnummer.
LPSTR lpzText{36}	Buchungstext.
LPSTR lpzSollkonto{8}	Kontonummer des Sollkontos.
long* plSollart	Zeiger auf die Kontoart des Sollkontos (OFCK_SAKO für Sachkonto, OFCK_KRED für Kreditor, OFCK_DEBI für Debitor)
LPSTR lpzHabenkonto{8}	Kontonummer des Habenkontos.
long* plHabenart	Zeiger auf die Kontoart des Habenkontos (OFCK_SAKO für Sachkonto, OFCK_KRED für Kreditor, OFCK_DEBI für Debitor)
long lAusgabeart	Ausgabeart aller zurückzugebenden Beträge (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, Beschreibung der Ausgabearten unter OFCK_HoleSaldo). Dieser Parameter ist nur notwendig, falls eine der Betrags-Variablen ungleich NULL ist.
LPSTR lpzBetrag{16}	Bruttobetrag der Buchung (in Landeswährung) im mit lAusgabeart festgelegten Format.
LPSTR lpzSteuersatz{7}	Steuersatz der Buchung im mit lAusgabeart festgelegten Format (Bei lAusgabeart = OFCK_FIXLEN ist zu beachten, dass die Länge nur 6 Zeichen ist).
LPSTR lpzSteuerbetrag{16}	Steuerbetrag der Buchung (in Landeswährung) im mit lAusgabeart festgelegten Format.
LPSTR lpzEurobetrag{16}	Bruttobetrag der Buchung in Euro im mit lAusgabeart festgelegten Format.
LPSTR lpzEurosteuerbetrag{16}	Steuerbetrag der Buchung in Euro im mit lAusgabeart festgelegten Format.
long* plKobus	Zeiger auf Anzeiger, ob zu dieser Buchung Kostenbuchungen existieren (>0).
LPSTR lpzBuchungsdatum{11}	Datum, an dem diese Buchung erfasst wurde.
long* plBuchungsnummer	Fortlaufende Buchungsnummer, eindeutig pro Mandant und Geschäftsjahr.
LPSTR pszOPBelegnummer {17}	OP-Belegnummer der Buchung. Kann leer sein bei Buchungserfassung mit einer syska EURO FIBU Version vor 7.0
LPSTR pszValutadatum {11}	Valutadatum der Buchung. Kann leer sein bei Buchungserfassung mit einer syska EURO FIBU Version vor 7.0
LPSTR pszBenutzer {31}	Anwender, der die Buchung erfasst hat. Nur in Plus-Version.
LPSTR pszZahlziel {11}	Das Zahlziel-Kürzel zur Buchung, falls gegeben
LPSTR lpzText2{36}	2. Buchungstext.
LPSTR pszDmsID{51}	DMS-ID

Rückgabewert:

long lErrno OFCK_weitererDatensatz, falls es einen weitere Buchungsbeleg gibt, OFCK_letzterDatensatz, falls dieser Buchungsbeleg der letzte war, < 0 bei Fehlern.

OFCK_HoleKobu

long WINAPI **OFCK_HoleKobu**(long lAusgabeart, LPSTR lpszBelegdatum, LPSTR lpszKost1, LPSTR lpszKost2, LPSTR lpszKostenart, LPSTR lpszBetrag, LPSTR lpszEurobetrag, long* plKostenkennung, LPSTR lpszBemerkung, LPSTR lpszBemerkung2, LPSTR lpszKost3, LPSTR lpszKost4, LPSTR lpszKost5, LPSTR lpszKost6)

Holt die Kostenbuchung(en) zu einem Buchungsbeleg (sofern vorhanden).

Für diese Funktion ist kein gesondertes Initialisieren oder Beenden des Hole-Vorganges notwendig, der Aufruf ist jedoch nur sinnvoll, wenn OFCK_HoleBube die Existenz einer Kostenbuchung mit *plKobus ungleich 0 signalisiert hat.

Alle Zeiger können auch NULL-Zeiger sein, wenn der entsprechende Wert uninteressant ist. Für Rückgabewerte müssen die in {}-Klammern angegebene Anzahl an Bytes reserviert werden.

Parameter:

long lAusgabeart Die gewünschte Ausgabeart von lpszBetrag und lpszEurobetrag.
Mögliche Werte:
- OFCK_VARLEN
- OFCK_FIXLEN
- OFCK_AUSGABE
Sind lpszBetrag und lpszEurobetrag beide NULL-Zeiger, so ist dieser Wert irrelevant.

LPSTR lpszBelegdatum {11} Belegdatum zur Kostenbuchung.

LPSTR lpszKost1 {8} Kostenstelle 1.

LPSTR lpszKost2 {8} Kostenstelle 2.

LPSTR lpszKostenart {8} Kostenart (Sachkonto-Kontonummer)

LPSTR lpszBetrag {16} Betrag in Landeswährung. Ausgabe im mit lAusgabeart gewünschten Format.

LPSTR lpszEurobetrag {16} Betrag in EURO. Ausgabe im mit lAusgabeart gewünschten Format.

long* plKostenkennung Kostenkennung (OFCK_FIX, OFCK_VARIABEL).

LPSTR lpszBemerkung {41} Bemerkung zur Kostenbuchung.

LPSTR lpszBemerkung2 {41} 2. Bemerkung zur Kostenbuchung.

LPSTR lpszKost3 {8} Kostenstelle 3.

LPSTR lpszKost4 {8} Kostenstelle 4.

LPSTR lpszKost5 {8} Kostenstelle 5.

LPSTR lpszKost6 {8} Kostenstelle 6.

Rückgabewert:

long lErrno OFCK_keinDatensatz - Es gibt keine Kostenbuchungen.
OFCKERR_uzl_Format - Kein gültiger Wert für lAusgabeart.
OFCK_weitererDatensatz - Es gibt weitere Kostenbuchungen.
OFCK_letzterDatensatz - Dies ist die letzte Kostenbuchung. Achtung! Es bedeutet nicht, dass dies auch der letzte Buchungsbeleg ist. Dies kann nur aus dem Rückgabewert von OFCK_HoleBube geschlossen werden.

OFCK_InitHoleKonto

long WINAPI **OFCK_InitHoleKonto**(long IMandnr, long IKontoart, long ISortierung, LPSTR IpszAbKonto, BOOL bShowErr)

Mit dieser Funktion ist es möglich den Vorgang einer Einzelabfrage von Konten und Kostenstellen/trägern einzuleiten. Das bedeutet, dass das OFCK darauf eingestellt wird, bei folgenden Aufrufen der Funktion OFCK_HoleSako, OFCK_HolePeko bzw. OFCK_HoleKost relevante Kontendaten zu liefern. Zwischen den Aufrufen von OFCK_InitHoleKonto und OFCK_HoleSako, OFCK_HolePeko bzw. OFCK_HoleKost dürfen keine anderen OFCK-Funktionen außer OFCK_BeendeHolen (vorzeitiges Beenden des schrittweisen Durchgehens von Datensätzen) aufgerufen werden!

Parameter:

long IMandnr	Mandantenummer.
long IKontoart	Kontoart der gewünschten Konten (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_KOST für Kostenstellen/träger). Von diesem Parameter hängt es ab, welche Hole-Funktion im folgenden benutzt werden muss: Bei OFCK_SAKO wird die Funktion OFCK_HoleSako benutzt, bei OFCK_PEKO OFCK_HolePeko und bei OFCK_KOST OFCK_HoleKost.
long ISortierung	Sortierung der Konten: OFCK_NUMMER nach Kontonummer, OFCK_KUERZEL nach Kürzel, OFCK_NAME nach Kontoname.
LPSTR IpszAbKonto	Kontonummer des ersten gewünschten Kontos/der ersten gewünschten Kostenstelle/träger. Es wird das erste Konto/die erste Kostenstelle/träger gesucht, dessen Nummer größer oder gleich IpszAbKonto ist. Falls das erste Konto/die erste Kostenstelle/träger seiner Art überhaupt gesucht wird, kann IpszAbKonto = NULL sein oder auf einen Leerstring zeigen.
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long IErrno	OFCKweitererDatensatz, falls es ein dem Kriterium IpszAbKonto genügendes Konto/Kostenstelle/Kostenträger der Art IKontoart gibt, OFCKkeinDatensatz, falls ein solches Konto/eine solche Kostenstelle/Kostenträger nicht existiert, < 0 bei Fehlern.
-------------	---

OFCK_HoleSako

long WINAPI **OFCK_HoleSako**(LPSTR IpszKontonr, LPSTR IpszKuerzel, LPSTR IpszName, long* plKlasse, BOOL* pllsumsaterloes, BOOL* pllkostenart, long* plSteuerkennung, long* pllsvorsteuer, long* plUstvakz, BOOL* plDatevAutomatik, LPSTR IpszSteuersatz, BOOL* plDBKonto, long* plKostenschablone, LPSTR pszWaehrung, BOOL* pllInvestition, BOOL* plBuchungssperre, long* plEuerKz, long IUst2aKz, long IFormularKennung)

Diese Funktion dient der schrittweisen Abfrage der relevanten Daten von Sachkonten. Sie darf nur nach erfolgreichem Aufruf der Funktion OFCK_InitHoleKonto (also Rückgabewert OFCKweitererDatensatz) mit iKontoart = OFCK_SAKO benutzt werden. Sämtliche Parameter können mit NULL belegt werden, falls die Daten für die Anwendung irrelevant sind. Die Zeiger ungleich NULL müssen auf Puffer zeigen, die groß genug sind, die Daten aufzunehmen. Die benötigte Größe eines Puffers steht in geschweiften Klammern direkt hinter dem Parameternamen. long-Variablen benötigen 4 Bytes.

Parameter:

LPSTR IpszKontonr{8}	Kontonummer.
LPSTR IpszKuerzel{10}	Kürzel des Kontos.
LPSTR IpszName{41}	Name des Kontos.
long* plKlasse	Zeiger auf die Kontenklasse (OFCK_OHNE falls das Konto keiner Klasse angehört, OFCK_BILAKTIV, falls das Konto ein Bilanz-Aktiv-Konto ist, OFCK_BILPASSIV, falls das Konto ein Bilanz-Passiv-Konto ist, OFCK_GUNDV, falls das Konto ein GuV-Konto ist).
BOOL* pllsumsaterloes	Zeiger (Referenz) auf eine Long-Variable: TRUE (1), falls das

	Konto ein Umsatzerlöskonto ist, FALSE (0) sonst.
BOOL* plIskostenart	Zeiger (Referenz) auf eine Long-Variable: TRUE (1), falls das Konto eine Kostenart ist, FALSE (0) sonst.
long* plSteuerkennung	Zeiger (Referenz) auf eine Long-Variable, die die Steuerkennung des Kontos aus {OFCK_NOSTEUER, OFCK_REGELSTEUERSATZ, OFCK_ERMSTEUERSATZ, OFCK_MISCSTEUERSATZ, OFCK_EGSTEUERSATZ, OFCK_NULLSTEUERSATZ} aufnimmt. Die Steuersätze (mit Ausnahme des Nullsteuersatzes) können über OFCK_HoleSteusa und OFCK_InitHoleSteusa ermittelt werden.
BOOL* plIsvorsteuer	Zeiger (Referenz) auf eine Long-Variable: TRUE (1), falls das Konto vorsteuerpflichtig ist, FALSE (0) sonst.
long* plUstvakz	Zeiger (Referenz) auf eine Long-Variable, in die Umsatzsteuervoranmeldungskennziffer eingetragen wird.
LPSTR IpszEroeffnungsdatum {11}	Eröffnungsdatum im Format TT.MM.JJJJ
BOOL* plDatevAutomatik	Zeiger auf eine Long-Variable: TRUE (1), falls das Konto ein DATEV-Automatikkonto ist, FALSE (0) sonst
LPSTR IpszSteuersatz {6}	Steuersatz. Wenn *plSteuerkennung OFCK_EGSTEUERSATZ oder OFCK_MISCSTEUERSATZ ist, dann enthält IpszSteuersatz einen Wert (Format: nn,nn), andernfalls wird eine leere Zeichenkette zurückgegeben.
BOOL* plDBKonto	Rückgabe: Konto ist ein Durchbuchungskonto
long* plKostenschablone	Rückgabe: Die Nummer der verwendeten Kostenschablone oder 0.
LPSTR pszWaehrung {5}	Rückgabe: Kontowährung (nur SwissFibu)
BOOL* plInvestition	Rückgabe: Konto ist ein Investitionskonto (nur SwissFibu)
BOOL* plBuchungssperre	Rückgabe: Konto hat eine Buchungssperre
long* plEuerKz	Rückgabe: DE: EÜR-Kennziffer. AT: ESt-Kennziffer. CH: 2. MWSt-Kennziffer.
long lUst2aKz	Rückgabe: DE: Kennziffer für die Umsatzsteuererklärung
long lFormularKennung	Rückgabe: DE: Formularkennung für die Umsatzsteuererklärung.
<u>Rückgabewert:</u>	
long lErrno	OFCKweitererDatensatz, falls es ein weiteres Sachkonto gibt, OFCKletzterDatensatz, falls dieses Sachkonto das letzte in der Reihe war, < 0 bei Fehlern.

OFCK_HolePeko

long WINAPI **OFCK_HolePeko**(LPSTR IpszKontonr, LPSTR IpszKuerzel, LPSTR IpszName, LPSTR IpszZusatz, LPSTR IpszUstidnr, LPSTR IpszLand, LPSTR IpszWaehrung, LPSTR IpszPlz, LPSTR IpszOrt, LPSTR IpszStrasse, LPSTR IpszBlz, LPSTR IpszBankknr, LPSTR IpszBank, LPSTR IpszGegenkonto, long* plSkontotage, long* plZahltag, long lAusgabeart, LPSTR IpszSkontoproz, long* plMahnkennziffer, long* plZahlungsziel, LPSTR IpszGegenkonto2, LPSTR IpszAnsprech, LPSTR IpszGefue, LPSTR IpszTel, LPSTR IpszFax, LPSTR IpszEmail, LPSTR IpszBemerkung, BOOL* plEinmalkonto, BOOL* plLohnveredeler, LPSTR IpszVerwendungszweck, LPSTR IpszDBKonto, long* plSprachkey, LPSTR IpszSwiftcode, LPSTR IpszAnredeFirma, LPSTR IpszAnredeAnsprech, LPSTR IpszAnredeGefue, LPSTR IpszEroeffnungsdatum, LPSTR IpszZessionstext, LPSTR IpszText1, LPSTR IpszText2, LPSTR IpszText3, LPSTR IpszWert1, LPSTR IpszWert2, LPSTR IpszWert3, LPSTR IpszDatum1, LPSTR IpszDatum2, LPSTR IpszDatum3, BOOL* plOption1, BOOL* plOption2, BOOL* plOption3, BOOL* plSperre, LPSTR IpszPfadInfo, BOOL* plBuchungssperre, LPSTR IpszIban1, LPSTR IpszBankland1, LPSTR IpszKontoinhaber1, LPSTR IpszBlz2, LPSTR IpszBankknr2, LPSTR IpszBank2, LPSTR IpszIban2, LPSTR IpszSwiftcode2, LPSTR IpszBankland2, LPSTR IpszKontoinhaber2, LPSTR IpszSepaMandat, LPSTR IpszSepaMandatDatum, LPSTR IpszSepaMandatNutzung1, BOOL* bSepaMandatFirmenLS1, long* plSepaMandatFlags1, LPSTR IpszSepaMandatDMSID1, LPSTR IpszSepaMandat2, LPSTR IpszSepaMandatDatum2, LPSTR

IpszSepaMandatNutzung2, BOOL* pbSepaMandatFirmenLS2, long* plSepaMandatFlags2, LPSTR IpszSepaMandatDMSID2, LPSTR szCreditorIdentifier, LPSTR szEnd2Endld, long* lEnd2EndldFlags, LPSTR IpszRegisternummer, LPSTR IpszRegistergericht, LPSTR IpszRechtsform, LPSTR IpszGeburtsdatum, long* plInkassoGegner, long* plInkassoAkte, long* plBonitaet, LPSTR IpszBonitaetPruefdatum, long* plBonitaetTyp, LPSTR IpszVorname, LPSTR IpszSteuernummer)

Diese Funktion dient der schrittweisen Abfrage der relevanten Daten von Personenkonten. Sie darf nur nach erfolgreichem Aufruf der Funktion OFCK_InitHoleKonto (also Rückgabewert OFCK_weitererDatensatz) mit IKontoart = OFCK_DEBI oder IKontoart = OFCK_KRED benutzt werden. Sämtliche Parameter können mit NULL belegt werden, falls die Daten für die Anwendung irrelevant sind. Nur der Parameter lAusgabeart ist zwingend, falls IpszSkontoproz ungleich NULL ist. Die Zeiger ungleich NULL müssen auf Puffer zeigen, die groß genug sind, um die Daten aufzunehmen. Die benötigte Größe eines Puffers steht in geschweiften Klammern direkt hinter dem Parameternamen (long-Variablen benötigen 4 Bytes). Die Art des geholten Personenkontos hängt von IKontoart im initialisierenden Aufruf OFCK_InitHoleKonto ab. Einige der Rückgabewerte können leer sein (das sind genau die als optional bezeichneten Werte in der OFCK_PEKO-Struktur).

Das Holen von Swiftcode (BIC), IBAN und des Länderkürzels der Bank ab Version 9.2 erfolgt mit je einem eigenen Parameter

Parameter:

LPSTR IpszKontonr{8}	Kontonummer.
LPSTR IpszKuerzel{10}	Kürzel des Kontos.
LPSTR IpszName{41}	Name des Kontos.
LPSTR IpszZusatz{41}	Zusatz zum Kontenname.
LPSTR IpszUstidnr{16}	Umsatzsteuer-ID-Nummer.
LPSTR IpszLand{4}	Landeskürzel.
LPSTR IpszWaehrung{5}	Währung in welcher das Personenkonto bebucht wird.
LPSTR IpszPlz{7}	Postleitzahl.
LPSTR IpszOrt{31}	Ortsbezeichnung.
LPSTR IpszStrasse{31}	Straße.
LPSTR IpszBlz{9}	Bankleitzahl des Bankkontos. (1. Bank)
LPSTR IpszBankknr{17}	Bankkontonummer. (1. Bank)
LPSTR IpszBank{41}	Name der Bank. (1. Bank)
LPSTR IpszGegenkonto{8}	Kontonummer des als Gegenkonto vermerkten Sachkontos.
long *plSkontotage	Zeiger (Referenz) auf die Skontofrist in Tagen (>= 0).
long *plZahltag	Zeiger (Referenz) auf die Zahlungsfrist in Tagen (>= 0).
long lAusgabeart	Ausgabeart des zurückzugebenden Betrages in skontoproz (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, Beschreibung der Ausgabearten unter OFCK_HoleSaldo). Dieser Parameter ist nur notwendig, falls skontoproz ungleich NULL ist.
LPSTR IpszSkontoproz{7}	Skontoprozentsatz im mit lAusgabeart festgelegten Format. Bei lAusgabeart = OFCK_FIXLEN ist zu beachten, dass die Länge nur 6 Zeichen ist.
long *plMahnkennziffer	Nur bei Kunden (Debitoren) sinnvoll (-1 ⇔ nicht mahnen, >=0 mahnbar). Bei Kreditoren wird nichts zurückgegeben.
long *plZahlungsziel	Zeiger (Referenz) auf den Zahlungszielschlüssel.
LPSTR IpszGegenkonto2{8}	Kontonummer des als Gegenkonto2 vermerkten Sachkontos.
LPSTR IpszAnsprech{41}	Ansprechpartner.
LPSTR IpszGefue{41}	Geschäftsführer.
LPSTR IpszTel{21}	Telefon.
LPSTR IpszFax{21}	Fax.
LPSTR IpszEmail{41}	E-Mail.
LPSTR IpszBemerkung{41}	Bemerkung.
BOOL * plEinmalkonto	Einmalkonto.

BOOL* plLohnveredeler	Lohnveredeler.
LPSTR IpszVerwendungszweck {28}	Verwendungszweck (Überweisungen).
LPSTR IpszDBKonto {8}	Durchbuchungskonto.
long* plSprachkey	Die hinterlegte Sprache für Mahnungen.
LPSTR IpszSwiftcode {12}	SWIFT-Code für den Auslandszahlungsverkehr. (1. Bank)
LPSTR IpszAnredeFirma {16}	Anrede des Mandanten / der Firma.
LPSTR IpszAnredeAnsprech {16}	Anrede des Ansprechpartners.
LPSTR IpszAnredeGefue {16}	Anrede des Geschäftsführers.
LPSTR IpszEroeffnungsdatum {11}	Eröffnungsdatum des Kontos im Format TT.MM.JJJJ
LPSTR IpszZessionstext {41}	Zessionstext (AT), zusätzlicher Bemerkungstext (DE=
LPSTR IpszText1 {51}	Zusatzfeld: Text 1
LPSTR IpszText2 {51}	Zusatzfeld: Text 2
LPSTR IpszText3 {51}	Zusatzfeld: Text 3
LPSTR IpszWert1 {16}	Zusatzfeld: Wert 1
LPSTR IpszWert2 {16}	Zusatzfeld: Wert 2
LPSTR IpszWert3 {16}	Zusatzfeld: Wert 3
LPSTR IpszDatum1 {11}	Zusatzfeld: Datum 1
LPSTR IpszDatum2 {11}	Zusatzfeld: Datum 2
LPSTR IpszDatum3 {11}	Zusatzfeld: Datum 3
BOOL* plOption1	Zusatzfeld: Option 1
BOOL* plOption2	Zusatzfeld: Option 2
BOOL* plOption3	Zusatzfeld: Option3
BOOL* plSperre	Liefersperre/Mahnsperre
LPSTR IpszPfadInfo {260}	Infoverzeichnis
BOOL* plBuchungssperre	Buchungssperre
LPSTR IpszIban1 {35}	IBAN (1. Bank)
LPSTR IpszBankland1 {5}	Bankland (1.Bank)
LPSTR IpszKontoinhaber1 {28}	Kontoinhaber (1. Bank)
LPSTR IpszBlz2 {9}	BLZ (2. Bank)
LPSTR IpszBankknr2 {17}	Bankknr. (2. Bank)
LPSTR IpszBank2 {41}	Bankname (2. Bank)
LPSTR IpszIban2 {35}	IBAN (2. Bank)
LPSTR IpszSwiftcode2 {12}	SWIFT/BIC (2. Bank)
LPSTR IpszBankland2 {5}	Bankland (2. Bank)
LPSTR IpszKontoinhaber2 {28}	Kontoinhaber (2. Bank)
LPSTR IpszSepaMandat {36}	SEPA-Mandat (1. Bank)
LPSTR IpszSepaMandatDatum {11}	Datum für SEPA-Mandat (1. Bank)
LPSTR IpszSepaMandatNutzung1 {11}	SEPA-Nutzungsdatum (1. Bank)
BOOL* bSepaMandatFirmenLS1	SEPA-Firmenlastschrift (1. Bank)
long* plSepaMandatFlags1	SEPA-Flags (1. Bank)
LPSTR IpszSepaMandatDMSID1 {51}	SEPA-DMS-ID (1. Bank)
LPSTR IpszSepaMandat2 {36}	SEPA-Mandat (2. Bank)
LPSTR IpszSepaMandatDatum2 {11}	Datum für SEPA-Mandat (2. Bank)
LPSTR IpszSepaMandatNutzung2	SEPA-Nutzungsdatum (2. Bank)

{11}	
BOOL* bSepaMandatFirmenLS2	SEPA-Firmenlastschrift (2. Bank)
long* plSepaMandatFlags2	SEPA-Flags (2. Bank)
LPSTR lpszSepaMandatDMSID2	SEPA-DMS-ID (2. Bank)
{51}	
LPSTR szCreditorIdentifizier {36}	Gläubiger-ID des Pekos
LPSTR szEnd2EndId {36}	Ende-zu-Ende-ID
long* lEnd2EndIdFlags	Einstellungen zur Ende-zu-Ende-ID
LPSTR lpszRegisternummer {21}	Registernummer z.B.: HRB 123456
LPSTR lpszRegistergericht {41}	Registergericht z.B.: Amtsgericht Karlsruhe
LPSTR lpszRechtsform {31}	Rechtsform z.B.: GmbH & Co KG
LPSTR lpszGeburtsdatum {11}	Geburtsdatum
long* plInkassoGegner	Gegnernummer beim Inkasso (Default = 0)
long* plInkassoAkte	Aktive Akte beim Inkasso (Default = 0)
long* plBonitaet	0 = ungeprüft; 1 = grün; 2 = gelb; 3 = rot
LPSTR lpszBonitaetPruefdatum {11}	Prüfdatum der Bonität
long* plBonitaetTyp	0 = unbekannt; 1 = person; 2 = Einzelunternehmer; 3 = Firma
LPSTR lpszVorname {31}	Vorname
LPSTR lpszSteuernummer	Steuernummer

Rückgabewert:

long lErrno OFCK_weitererDatensatz, falls es ein weiteres Personenkonto gibt, OFCK_letzterDatensatz, falls dieses Personenkonto das letzte in der Reihe war, < 0 bei Fehlern.

OFCK_HoleKost

long WINAPI **OFCK_HoleKost**(long* plSteuerkennung, LPSTR lpszKontonr, LPSTR lpszKuerzel, LPSTR lpszName, LPSTR lpszBemerkung, long* plDimension, long* plSaldovortrag, LPSTR pszStartdatum LPSTR pszEnddatum, LPSTR lpszKostref)

Schrittweises holen von Kostenstellen und Kostenträgern. Vor dem Aufruf dieser Funktion muss zuerst OFCK_InitHoleKonto mit der Kontenart OFCK_KOST aufgerufen werden, liefert diese Funktion den Rückgabewert OFCK_weitererDatensatz, so kann mit dem schrittweisen Abrufen der Kostenstellen und Kostenträger begonnen werden. Ist ein Parameter für die Anwendung uninteressant, so kann statt dessen der NULL-Zeiger übergeben werden. In Klammern steht bei Zeichenketten die minimal benötigte Länge für diesen Rückgabewert. Soll das Holen vorzeitig abgebrochen werden, ist OFCK_BeendeHolen aufzurufen.

Parameter:

long* plSteuerkennung	Nur bei Vereinsbuchführung relevant: Zeiger auf eine long-Variable (4 Bytes), mögliche Werte sind: - OFCK_AUFTEIL (Aufzuteilende VSt.) - OFCK_NOSTEUER (Steuerfrei) - OFCK_ABZUG (Abzugsfähige VSt.)
LPSTR lpszKontonr {8}	Erhält die Kostenstellen/trägernummer.
LPSTR lpszKuerzel {9}	Erhält das Kürzel der Kostenstelle/träger.
LPSTR lpszName {31}	Erhält den Namen der Kostenstelle/träger.
LPSTR lpszBemerkung {41}	Erhält die Bemerkung zur Kostenstelle/träger.
long* plDimension	Erhält die Dimension (0-5).
long* plSaldovortrag	Erhält die Kennung, ob der Saldovortrag dieser Kostenstelle am Jahresende beibehalten wird. Auswirkungen nur bei installierter EURO

KORE.

- LPSTR pszStartdatum {11} Erhält das Startdatum der Laufzeitbegrenzung. Format: TT.MM.JJJJ. Dieser Wert ist nur bei aktivierter EURO KORE interessant.
- LPSTR pszEnddatum {11} Erhält das Enddatum der Laufzeitbegrenzung. Format: TT.MM.JJJJ. Dieser Wert ist nur bei aktivierter EURO KORE interessant.
- LPSTR pszKostref {8} Erhält die Kostenstellenreferenz falls die Dimension 4 ist und die Kostenstellenreferenz für den Mandant aktiviert wurde.

Rückgabewert:

- long lErrno
- OFCKweitererDatensatz. Es liegt ein weiter Datensatz vor, dessen Daten mit dieser Funktion abgerufen werden können.
 - OFCKletzterDatensatz. Dies ist der letzte Datensatz. Ein explizites Beenden der Hole-Funktion ist nicht nötig.
 - OFCKERRkeinInit. Das Holen von Kostenstellen/trägern wurde nicht initialisiert.

OFCK_InitHoleSollwerte

long WINAPI **OFCK_InitHoleSollwerte**(long lMandnr, long lKontoart, long lJahr, long lSortierung, LPSTR pszAb, BOOL bShowErr, long lDimension, LPSTR lpszKostref)

Mit dieser Funktion ist es möglich den Vorgang einer Einzelabfrage von Sollwerten zu Konten und Kostenstellen/trägern einzuleiten. Das bedeutet, dass das OFCK darauf eingestellt wird, bei folgenden Aufrufen der Funktion OFCK_HoleSollwert die Sollwerte zu liefern. Zwischen den Aufrufen von OFCK_InitHoleSollwerte und OFCK_HoleSollwerte dürfen keine anderen OFCK-Funktionen außer OFCK_BeendeHolen (vorzeitiges Beenden des schrittweisen Durchgehens von Datensätzen) aufgerufen werden!

Parameter:

- long lMandnr Mandantenummer.
- long lKontoart Kontoart der gewünschten Konten (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_KOST für Kostenstellen/träger).
- long lJahr Das Jahr (Geschäftsjahr), aus dem die Sollwerte geholt werden sollen.
- long lSortierung Sortierung der Konten: OFCK_NUMMER nach Kontonummer, OFCK_KUERZEL nach Kürzel, OFCK_NAME nach Kontoname.
- LPSTR lpszAb Kontonummer der ersten gewünschten Sollwerte zu Kontos bzw. der ersten gewünschten Kostenstelle/träger. Es wird das erste Konto gesucht, dessen Nummer größer oder gleich lpszAb ist. Falls das erste Konto seiner Art überhaupt gesucht wird, kann lpszAbKonto = NULL sein oder auf einen Leerstring zeigen.
- BOOL bShowErr TRUE, falls Fehlermeldungen angezeigt werden sollen.
- long lDimension Einschränkung auf eine Dimension bei Kostenstellen.
- LPSTR lpszKostref Einschränkung auf die Kostenstellenreferenz bei lDimension=4 und aktivierter Kostenstellenreferenzierung für den Mandant.

Rückgabewert:

- long lErrno OFCKweitererDatensatz, falls es ein dem Kriterium lpszAbKonto genügendes Konto der Art lKontoart gibt, OFCKkeinDatensatz, falls ein solches Konto bzw. eine solche Kostenstelle/träger nicht existiert, < 0 bei Fehlern.

OFCK_HoleSollwerte

long WINAPI **OFCK_HoleSollwerte**(LPSTR pszKontonr, LPSTR pszKuerzel, LPSTR pszKontoname, long lAusgabeart, LPSTR pszMon1, LPSTR pszMon2, LPSTR pszMon3, LPSTR pszMon4, LPSTR pszMon5, LPSTR pszMon6, LPSTR pszMon7, LPSTR pszMon8, LPSTR pszMon9, LPSTR pszMon10, LPSTR pszMon11, LPSTR pszMon12, LPSTR pszGesamt, BOOL* plSWinEuro);

Diese Funktion dient der schrittweisen Abfrage der Sollwerte von Konten bzw. Kostenstellen/trägern. Sie darf nur nach erfolgreichem Aufruf der Funktion OFCK_InitHoleSollwerte (also Rückgabewert OFCK_weitererDatensatz) benutzt werden. Sämtliche Parameter können mit NULL belegt werden, falls die Daten für die Anwendung irrelevant sind. Die Zeiger ungleich NULL müssen auf Puffer zeigen, die groß genug sind, die Daten aufzunehmen. Die benötigte Größe eines Puffers steht in geschweiften Klammern direkt hinter dem Parameternamen. long-Variablen benötigen 4 Bytes.

Parameter:

LPSTR lpszKontonr{8}	Kontonummer.
LPSTR lpszKuerzel{10}	Kürzel des Kontos.
LPSTR lpszName{41}	Name des Kontos.
long lAusgabeart	Ausgabeart der zurückzugebenden Beträge (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, Beschreibung der Ausgabearten unter OFCK_HoleSaldo).
LPSTR pszMon1 {16}	Erhält den Sollwert zum jeweiligen Monat bzw. die Summe für das Gesamtjahr. Beträge im Soll sind dabei positiv, Beträge im Haben negativ. Bei Kostenstellen und –trägern sind der fixe und der variable Anteil jeweils kumuliert.
LPSTR pszMon2 {16}	
LPSTR pszMon3 {16}	
LPSTR pszMon4 {16}	
LPSTR pszMon5 {16}	
LPSTR pszMon6 {16}	
LPSTR pszMon7 {16}	
LPSTR pszMon8 {16}	
LPSTR pszMon9 {16}	
LPSTR pszMon10 {16}	
LPSTR pszMon11 {16}	
LPSTR pszMon12 {16}	
LPSTR pszGes {16}	
BOOL* pISWinEuro	Gibt an, ob die Sollwerte in (alter) Landeswährung oder in EUR angegeben sind.

Rückgabewert:

long lErrno	OFCK_weitererDatensatz, falls es ein weiteres Konto gibt, OFCK_letzterDatensatz, falls dieses Konto das letzte in der Reihe war, < 0 bei Fehlern.
-------------	---

OFCK_InitHoleOP

long WINAPI **OFCK_InitHoleOP**(long lMandnr, LPSTR lpszKonto, long lKontoart, BOOL bShowErr)

Mit dieser Funktion ist es möglich den Vorgang einer Einzelabfrage von Offenen Posten einzuleiten. Das bedeutet, dass das OFCK darauf eingestellt wird, bei folgenden Aufrufen der Funktion OFCK_HoleOP relevante Daten der Offenen Posten zu liefern. Zwischen den Aufrufen von OFCK_InitHoleOP und OFCK_HoleOP dürfen keine anderen OFCK-Funktionen außer OFCK_BeendeHolen (vorzeitiges Beenden des schrittweisen Durchgehens von Datensätzen) aufgerufen werden!

Parameter:

long lMandnr	Mandantenummer
LPSTR lpszKonto	Kontonummer des Kontos, dessen OPs betrachtet werden sollen. Falls lpszKonto = NULL oder leer ist, werden sämtliche OPs der gegebenen Kontenart durchgegangen.
long lKontoart	Kontenart der Konten, zu denen die OPs betrachtet werden sollen (OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren).
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen

Rückgabewert:

long lErrno OFCK_weitererDatensatz, falls es zu einem dem Kriterium lpszKonto genügenden OP gibt, OFCK_keinDatensatz, falls kein Offener Posten existiert, < 0 bei Fehlern.

OFCK_HoleOP

long WINAPI **OFCK_HoleOP**(LPSTR lpszKontonr, LPSTR lpszKuerzel, LPSTR lpszKontoname, LPSTR lpszBelegnr, LPSTR lpszOpbelegnr, LPSTR lpszBelegdatum, LPSTR lpszValutadatum, LPSTR lpszSkontodatum1, LPSTR lpszSkontoproz1, LPSTR lpszSkontodatum2, LPSTR lpszSkontoproz2, LPSTR lpszFaelldatum, LPSTR lpszText, long iAusgabeart, LPSTR lpszBruttobetrag, LPSTR lpszAusstand, LPSTR lpszWaehrung, long* plMahnstufe, LPSTR lpszMahndatum, LPSTR lpszBemerkung, LPSTR lpszGegenkonto, long* plZahlziel, long* plZahlungssperre, LPSTR lpszSkontokonto1, LPSTR lpszSkontokonto2, LPSTR lpszSteuersatz1, LPSTR lpszSteuersatz2, LPSTR lpszAusstand2, LPSTR lpszGWBetrag, long* plMahnsperre, LPSTR lpszText2, LPSTR pszDmsID, long* plInkasso, long* plInkassoAkte, long* plInkassoForderung, long* plInkassoMahnkosten, long* plMahnsperre, LPSTR pszText2, LPSTR pszDmsID, long* plInkasso, long* plInkassoAkte, long* plInkassoForderung, long* plInkassoMahnkosten)

Diese Funktion dient der schrittweisen Abfrage der relevanten Daten von OPs zu Personenkonten. Sie darf nur nach erfolgreichem Aufruf der Funktion OFCK_InitHoleOP (also Rückgabewert OFCK_weitererDatensatz) benutzt werden. Sämtliche Parameter können mit NULL belegt werden, falls die Daten für die Anwendung irrelevant sind. Nur der Parameter iAusgabeart ist zwingend. Die Zeiger ungleich NULL müssen auf Puffer zeigen, die groß genug sind, um die Daten aufzunehmen. Die benötigte Größe eines Puffers steht in geschweiften Klammern direkt hinter dem Parameternamen. Die Art des geholten OPs hängt von iKontoart im initialisierenden Aufruf OFCK_InitHoleOP ab.

OFCK_InitHoleHistOP

long WINAPI **OFCK_InitHoleHistOP**(long lMandnr, LPSTR pszKonto, long lKontoart, pszBuchungsdatum, LPSTR pszBelegnr, LPSTR pszOPBelegnr, BOOL bShowErr)

Mit dieser Funktion ist es möglich den Vorgang einer Einzelabfrage von historischen Offenen Posten einzuleiten. Das bedeutet, dass das OFCK darauf eingestellt wird, bei folgenden Aufrufen der Funktion OFCK_HoleHistOP relevante Daten der historischen Offenen Posten zu liefern. Zwischen den Aufrufen von OFCK_InitHoleHistOP und OFCK_HoleHistOP dürfen keine anderen OFCK-Funktionen außer OFCK_BeendeHolen (vorzeitiges Beenden des schrittweisen Durchgehens von Datensätzen) aufgerufen werden!

Es kann nach einem der Parameter Kontonummer, Buchungsdatum, Belegnummer und OP-Belegnummer sortiert werden. Die höchste Priorität hat dabei das Buchungsdatum, dann folgen Belegnummer, OP-Belegnummer und Kontonummer. Soll eines oder mehrere der Kriterien nicht verwendet werden, so ist ein NULL-Zeiger oder die leere Zeichenkette zu übergeben. Wird gar keines der Kriterien angegeben, so ist die Reihenfolge die der Erfassung als historischer OP. Als Buchungsdatum gilt das Zahlungsdatum des OPs.

Parameter:

long lMandnr	Mandantenummer
LPSTR pszKonto	Kontonummer des Kontos, dessen OPs betrachtet werden sollen. Bitte beachten Sie obige Hinweise
long lKontoart	Kontenart der Konten, zu denen die OPs betrachtet werden sollen (OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren).
LPSTR pszBuchungsdatum	Das Buchungsdatum, ab dem die historischen OPs zurückgegeben werden. Bitte beachten Sie obige Hinweise. Als Buchungsdatum gilt das Zahlungsdatum des OPs.
LPSTR pszBelegnr	Die Belegnummer, zu der die historischen OPs zurückgegeben werden sollen. Bitte beachten Sie obige Hinweise.
LPSTR pszOPBelegnr	Die OP-Belegnummer, zu der die historischen OPs zurückgegeben werden sollen. Bitte beachten Sie obige Hinweise.
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen

Rückgabewert:

long lErrno OFCK_weitererDatensatz, falls es zu einem dem Kriterium lpszKonto genügenden OP gibt, OFCK_keinDatensatz, falls kein Offener Posten existiert, < 0 bei Fehlern.

OFCK_HoleHistOP

long WINAPI **OFCK_HoleHistOP**(LPSTR lpszKontonr, LPSTR lpszKuerzel, LPSTR lpszKontoname, LPSTR lpszBelegnr, LPSTR lpszOpbelegnr, LPSTR lpszBelegdatum, LPSTR lpszValutadatum, LPSTR lpszSkontodatum1, LPSTR lpszSkontoproz1, LPSTR lpszSkontodatum2, LPSTR lpszSkontoproz2, LPSTR lpszFaelldatum, LPSTR lpszText, long iAusgabeart, LPSTR lpszBruttobetrag, LPSTR lpszAusstand, LPSTR lpszWaehrung, long* plMahnstufe, LPSTR lpszMahndatum, LPSTR lpszBemerkung, LPSTR lpszGegenkonto, long* plZahlziel, long* plZahlungssperre, LPSTR lpszSkontokonto1, LPSTR lpszSkontokonto2, LPSTR lpszSteuersatz1, LPSTR lpszSteuersatz2, LPSTR lpszAusstand2, LPSTR lpszGWBetrag, long* plMahnsperre, LPSTR pszZahlung, LPSTR pszZahldatum, LPSTR lpszBenutzer, LPSTR lpszSkontobetrag, LPSTR lpszZahlbelegnr, LPSTR lpszZahltext, long* plZahlungsart, LPSTR lpszBenutzer, LPSTR lpszSkontobetrag, LPSTR lpszZahlbelegnr, LPSTR lpszZahltext, long* plZahlungsart, LPSTR pszText2, LPSTR pszZahltext2, LPSTR pszDmsID, long* plInkasso, long* plInkassoAkte, long* plInkassoForderung, long* plInkassoMahnkosten)

Diese Funktion dient der schrittweisen Abfrage der relevanten Daten von historischen OPs zu Personenkonten. Sie darf nur nach erfolgreichem Aufruf der Funktion OFCK_InitHoleHistOP (also Rückgabewert OFCK_weitererDatensatz) benutzt werden. Sämtliche Parameter können mit NULL belegt werden, falls die Daten für die Anwendung irrelevant sind. Nur der Parameter iAusgabeart ist zwingend. Die Zeiger ungleich NULL müssen auf Puffer zeigen, die groß genug sind, um die Daten aufzunehmen. Die benötigte Größe eines Puffers steht in geschweiften Klammern direkt hinter dem Parameternamen. Die Art des geholten OPs hängt von iKontoart im initialisierenden Aufruf OFCK_InitHoleOP ab.

Parameter:

LPSTR lpszKontonr{8}	Kontonummer des Debitoren- bzw. Kreditoren-Kontos.
LPSTR lpszKuerzel{10}	Kürzel des Kontos.
LPSTR lpszKontoname {41}	Name des Kontos.
LPSTR lpszBelegnr{17}	Belegnummer der Buchung.
LPSTR lpszOpbelegnr {17}	OP-Belegnummer.
LPSTR lpszBelegdatum{11}	Belegdatum im Format TT.MM.JJJJ.
LPSTR lpszValutadatum {11}	Valutadatum im Format TT.MM.JJJJ.
LPSTR lpszSkontodatum1{11}	1. Skontodatum des OPs (TT.MM.JJJJ).
LPSTR lpszSkontoproz1 {6}	1. Skontosatz.
LPSTR lpszSkontodatum2{11}	2. Skontodatum des OPs (TT.MM.JJJJ).
LPSTR lpszSkontoproz2{6}	2. Skontosatz.
LPSTR lpszFaelldatum{11}	Fälligkeitsdatum des OPs (TT.MM.JJJJ).
LPSTR lpszText{36}	OP-Text bzw. –Bezeichnung.
long iAusgabeart	Ausgabeart der zurückzugebenden Beträge (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, Beschreibung der Ausgabearten unter OFCK_HoleSaldo). Dieser Parameter ist nur notwendig, falls bruttobetrag oder ausstand ungleich NULL ist.
LPSTR lpszBruttobetrag{16}	Ursprünglicher Brutto- bzw. Rechnungsbetrag des OPs im mit iAusgabeart festgelegten Format.

LPSTR lpszAusstand{16}	Ausstand des OPs im mit iAusgabeart festgelegten Format
LPSTR lpszWaehrung{4}	Währung der im OP angegebenen Beträge (z.B. 'EUR' falls der OP in Euro gebucht wurde).
long *plMahnstufe	Die aktuelle Mahnstufe des OPs (-1 ⇔ der OP ist nicht mahnbar).
LPSTR lpszMahndatum{11}	Datum der letzten Mahnung (TT.MM.JJJJ). Dieses kann leer sein, obwohl der OP eine Mahnstufe ungleich 0 hat (dies passiert falls der letzte Mahnlauf rückgängig gemacht wurde).
LPSTR lpszBemerkung{41}	Bemerkung zum OP
LPSTR lpszGegenkonto {8}	Das Gegenkonto das den OP erzeugenden Buchung.
long* plZahlziel	Die Zahlzielnummer des zugrundeliegenden Zahlzieles. Wurden abweichende Angaben gemacht, so sind diese zu verwenden (z.B. lpszSkontodatum1)
long* plZahlungssperre	Zahlungssperre (>0, falls gesetzt)
LPSTR lpszSkontokonto1 {8}	Erstes Skontokonto.
LPSTR lpszSkontokonto2 {8}	Zweites Skontokonto
LPSTR lpszSteuersatz1 {6}	Buchungssteuersatz
LPSTR lpszSteuersatz2 {6}	2. Steuersatz, falls OP aus Splitbuchung mit 2 Steuersätzen erzeugt wurde.
LPSTR lpszAusstand2 {16}	Ausstand zum 2. Steuersatz
LPSTR lpszGWBetrag {16}	Nur bei Fremdwährungs-OPs: der Ausstand in Grundwährung (DM/ATS oder EUR)
long* plMahnsperre	Dieser OP ist mit einer Mahnsperre versehen.
LPSTR pszZahlung {16}	Zahlbetrag am Zahldatum.
LPSTR pszZahldatum {11}	Datum der Zahlung.
LPSTR lpszBenutzer{31}	Benutzer
LPSTR lpszSkontobetrag{16}	Skontobetrag
LPSTR lpszZahlbelegnr{9}	Zahlungsbelegnummer
LPSTR lpszZahltext{36}	Zahlungstext
long* plZahlungsart	Zahlungsart (OFCK_SONSTIGE, OFCK_UEBERWEISUNG, OFCK_SCHECK, OFCK_EINZUG, OFCK_ABBUCHUNG, OFCK_POS, OFCK_ESR, OFCK_SEPA_UEBERWEISUNG, OFCK_SEPA_LASTSCHRIFT_CORE, OFCK_SEPA_LASTSCHRIFT_B2B, Zu den beiden Sepa-Lastschriftarten ist eines der folgenden Flags hinzuerodert: OFCK_SEPA_LS_ONCE, OFCK_SEPA_LS_FIRST, OFCK_SEPA_LS_RECUR, OFCK_SEPA_LS_FINAL) OPs, welche im Rahmen der Anzahlungsschlussrechnung ausgeglichen wurden, haben OFCK_ANZRECH_SCHLUSS hinzuerodert.
LPSTR pszText2{36}	2. OP-Text bzw. -Bezeichnung.
LPSTR pszZahltext2{36}	2. Zahlungstext
LPSTR pszDmsID{51}	DMS-ID
long* plInkasso	1 wenn sich der OP im Inkasso befindet, 0 falls nicht
long* plInkassoAkte	Aktennr. eines im Inkasso befindlichen OPs
long* plInkassoForderung	Forderungsnr.
long* plInkassoMahnkosten	Forderungsnr. der Mahnkosten, sofern bei der Inkasso-Übergabe welche ermittelt wurden.

Rückgabewert:

long lErrno OFCKweitererDatensatz, falls es einen weiteren OP gibt,

OFCK_letzterDatensatz, falls dieser Offene Posten der letzte in der in OFCK_InitHoleOP festgelegten Selektion war, < 0 bei Fehlern.

OFCK_InitHoleSteusa

long WINAPI **OFCK_InitHoleSteusa**(long IMandnr, BOOL bShowErr)
Leitet das Holen von Steuersätzen ein.

Parameter

long IMandnr Mandantenummer, zu der die Steuersätze geholt werden sollen.
BOOL bShowErr Fehler anzeigen.

Rückgabewert:

long IErrno OFCK_weitererDatensatz: Es sind weitere Datensätze vorhanden.
 OFCK_keinDatensatz: Es sind keine Datensätze vorhanden.
 < 0: Fehlernummer.

OFCK_HoleSteusa

long WINAPI **OFCK_HoleSteusa**(long* pISteuerkennung, LPSTR pszDatum, LPSTR pszSteuersatz, LPSTR pszUstkonto, LPSTR pszVstkonto, LPSTR pszVstkonto2, LPSTR pszGewSkonti, LPSTR pszErhSkonti, LPSTR pszAbzugsAnteil)

Holt die Daten eines Steuersatzes. Vor dem ersten Aufruf muss die Funktion OFCK_InitHoleSteusa aufgerufen werden. Für alle Parameter kann alternativ auch der NULL-Zeiger übergeben, es erfolgt dann keine Rückgabe. Die Zahl in Klammern hinter einem Parameter gibt die Anzahl der zu reservierenden Bytes an.

Parameter

long* pISteuerkennung Rückgabe: Die Art des Steuersatzes. Mögliche Werte:
 OFCK_NULLSTEUERSATZ für den Nullsteuersatz,
 OFCK_ERMSTEUERSATZ für den ermäßigten Steuersatz,
 OFCK_REGELSTEUERSATZ für den Regelsteuersatz,
 OFCK_MISCSTEUERSATZ für den ersten sonstigen Steuersatz,
 OFCK_MISCSTEUERSATZ 2 für den zweiten sonstigen Steuersatz,
 OFCK_EGSTEUERSATZ für einen EG-Steuersatz
 OFCK_TEILSTEUERSATZ, OFCK_TEILSTEUERSATZ2
 OFCK_VERRECHNUNGSSATZ, OFCK_VERRECHNUNGSSATZ2,
 OFCK_VERRECHNUNGSSATZ3, OFCK_VERRECHNUNGSSATZ4

LPSTR pszDatum {11} Rückgabe: Das Datum, ab dem der Steuersatz gültig ist. Format::
 TT.MM.JJJJ

LPSTR pszSteuersatz {6} Rückgabe: Der Steuersatz. Bei Regelsatz und ermäßigtem Satz wird
 hier der Steuersatz im Format „nn,nn“ zurückgegeben werden. Bei EG-
 Steuersätzen und sonstigen Steuersätzen wird eine leere Zeichenkette
 zurückgegeben.

LPSTR pszUstkonto {8} Rückgabe: Das Konto für die Buchung der Umsatzsteuer

LPSTR pszVstkonto {8} Rückgabe: Das Konto für die Buchung der Vorsteuer

LPSTR pszVstkonto2 {8} Rückgabe: Das 2. Vorsteuerkonto (nur in SwissFibu relevant)

LPSTR pszGewSkonti {8} Rückgabe: Das Konto für gewährte Skonti

LPSTR pszErhSkonti {8} Rückgabe: Das Konto für erhaltene Skonti

LPSTR pszAbzugsfAnteil {6} Rückgabe Der abzugsfähige Anteil. Format: „nn,nn“

Rückgabewert:

long lErrno OFCK_weitererDatensatz: Es sind weitere Datensätze vorhanden.
 OFCK_letzterDatensatz: Es sind keine weiteren Datensätze vorhanden.
 OFCKERR_keinInit: Das Holen von Datensätzen wurde nicht initialisiert.
 <0: Fehlernummer

OFCK_InitHoleStapel

long WINAPI OFCK_InitHoleStapel(long lMandnr, BOOL bShowErr);

Diese Funktion leitet das Holen von Stapeln ohne Dialog ein.

Parameter

long lMandnr Mandantenummer

BOOL bShowErr Fehler anzeigen. Gilt auch für die folgenden Aufrufe von OFCK_HoleStapel.

Rückgabewert:

long lErrno OFCK_weitererDatensatz: Es gibt noch weitere Stapel.
 OFCK_letzterDatensatz: Dies ist der letzte Stapel
 <=0: Fehlernummer

OFCK_HoleStapel

long WINAPI OFCK_HoleStapel(long* plStapelNr, LPSTR pszStapelname, long* plWiederkehr, long* plTransitorisch, long* plSperr, long* plBebucht, long* plAbstimm0)

Diese Funktion holt die Daten eines Stapels ab.

Parameter

long* plStapelnummer Rückgabe: Stapelnummer

LPSTR pszStapelname Rückgabe: Stapelname.
{33}

long* plWiederkehr Rückgabe: Das Wiederkehr-Intervall in Monaten. Bei 0 ist der Stapel nicht wiederkehrend.

long* plTransitorisch Rückgabe: ungleich 0: Der Stapel ist transitorisch.

long* plSperr Rückgabe: ungleich 0: Der Stapel ist gesperrt (diese Kennung ist nach einem durchgeführten Clearinglauf gesetzt).

long* plBebucht Rückgabe: ungleich 0: Der Stapel ist bebucht.

long* plAbstimm0 Rückgabe der Option: „Nach Verbuchen Abstimmsumme auf 0,00 setzen“.

Rückgabewert:

long lErrno OFCK_weitererDatensatz: Es gibt noch weitere Stapel.
 OFCK_letzterDatensatz: Dies ist der letzte Stapel
 <=0: Fehlernummer

OFCK_InitHoleMandant

long WINAPI OFCK_InitHoleMandant()

Mit dieser Funktion wird das Holen von Mandanten ohne Dialog eingeleitet.

Parameter Keine.

Rückgabewert:

long lErrno OFCK_weitererDatensatz: Es gibt noch weitere Mandanten.
 OFCKERR_keine_Mandanten: Es sind keine Mandanten vorhanden
 <=0: Fehlernummer

OFCK_HoleMandant

long WINAPI OFCK_HoleMandant(long* pIMandant, LPSTR pszName, LPSTR pszZusatz)

Mit dieser Funktion werden die installierten Mandanten der Reihe nach abgefragt. Für die Rückgabe stehen die Mandantenummer, der Name sowie der Zusatz zur Verfügung. Für weitergehende Informationen ist die Funktion OFCK_HoleMandanteninfo verfügbar.

Parameter

long* pIMandant Rückgabe: Hier wird die Nummer des Mandanten zurückgegeben. Dieser Zeiger kann NULL sein.

LPSTR pszName {41} Rückgabe: Hier wird der Name des Mandaten zurückgegeben. Für diesen Parameter sind mind. 41 Bytes zu reservieren. Dieser Parameter kann NULL sein.

LPSTR pszZusatz{41} Rückgabe: Hier wird der Zusatzdes Mandaten zurückgegeben. Für diesen Parameter sind mind. 41 Bytes zu reservieren. Dieser Parameter kann NULL sein.

Rückgabewert:

long lErrno OFCK_weitererDatensatz: Es gibt noch weitere Mandanten.
 OFCK_letzter_Datensatz: Dies ist der letzte Mandant.
 OFCKERR_keine_Mandanten: Es sind keine Mandanten vorhanden
 <=0: Fehlernummer

OFCK_InitHoleSprache

long WINAPI OFCK_InitHoleSprache(long IMandant)

Mit dieser Funktion wird das Holen von Sprachen zu einem Mandanten eingeleitet.

Parameter

long IMandant Die Nummer des Mandanten, zu dem die Sprachen abgefragt werden sollen.

Rückgabewert:

long IErrno
OFCK_weitererDatensatz: Es gibt noch weitere Sprachen.
OFCK_letzter_Datensatz: Dies ist die letzte Sprache.
OFCKERR_keine_Sprachen: Es sind keine Sprachen vorhanden
<=0: Fehlernummer

OFCK_HoleSprachName

long WINAPI OFCK_HoleSprachName(long* pINummer, LPSTR pszName)

Mit dieser Funktion werden die vorhandenen Sprachen zu einem Mandaten abgefragt

Parameter

long* pINummer Rückgabe: Hier wird die Nummer der Sprache zurückgegeben. Dieser Zeiger kann NULL sein.

LPSTR pszName {41} Rückgabe: Hier wird der Name der Sprache zurückgegeben. Für diesen Parameter sind mind. 41 Bytes zu reservieren. Dieser Parameter kann NULL sein.

Rückgabewert:

long IErrno
OFCK_weitererDatensatz: Es gibt noch weitere Sprachen.
OFCK_letzter_Datensatz: Dies ist die letzte Sprache
<=0: Fehlernummer

OFCK_InitHoleBatch

long WINAPI OFCK_InitHoleBatch(long IMandnr, long IStapelnr, BOOL bShowErr)

Mit dieser Funktion wird das Holen von Stapelbuchungen (Batchbuchungen) eingeleitet. Zum Beenden sollte unbedingt OFCK_BeendeHolen aufgerufen werden.

Die geholten Stapelbuchungen sind sortiert nach Stapel und innerhalb eines Stapels nach Erfassung. Buchungen in wiederkehrenden Stapeln werden nur einmal geholt. Unvollständige Buchungen können nur im Stapel „Unvollständige Buchungen“ auftreten, in welchen dann einige Parameter fehlen können.

Parameter

long IMandnr Mandantenummer

long IStapelnr Stapelnummer. Diese kann mit OFCK_HoleStapelNr in Erfahrung gebracht werden. Wird eine 0 übergeben, so werden alle Buchungen zu allen Stapeln geholt.

BOOL bShowErr Fehler anzeigen. Gilt auch für die folgenden Aufrufe von OFCK_HoleBatchHeader, OFCK_HoleBatchSplit und OFCK_HoleBatchKobu.

Rückgabewert:

long lErrno OFCKweitererDatensatz: Es gibt noch weitere Batchbuchungen.
OFCKletzterDatensatz: Dies ist die letzte Batchbuchung
OFCKERRkeinDatensatz: Es gibt keine Batchbuchungen zu diesem Stapel
<=0: Fehlernummer

OFCK_HoleBatchHeader

long WINAPI OFCK_HoleBatchHeader(long* pStapel, LPSTR pszBelegnr, LPSTR pszOpBelegnr, LPSTR pszBelegdatum, LPSTR pszValutadatum, long* pTeilbuanz, LPSTR pszWaehrung, BOOL* pUstdirekt, BOOL* pKeinOP, LPSTR pszZahlziel, BOOL* pUnvollst, BOOL* pIstEB, BOOL* pIstAB, BOOL* pIstUB, BOOL* pUeberweisung, LPSTR pszDmsID)

Mit dieser Funktion werden übergreifende Daten für alle Teilbuchungen einer Batchbuchung ermittelt. Für alle Parameter kann der NULL-Zeiger übergeben werden. Einzeilige Buchungen enthalten genau eine Splitteilbuchung, welche dann mit OFCK_HoleBatchSplit geholt werden kann.

Parameter

long* pStapel	Rückgabe: Stapelnummer
LPSTR pszBelegnr {17}	Rückgabe: Belegnummer
LPSTR pszOpBelegnr {17}	Rückgabe: OP-Belegnummer
LPSTR pszBelegdatum {11}	Rückgabe: Belegdatum
LPSTR pszValutadatum {11}	Rückgabe: Valutadatum
long* pTeilbuanz	Rückgabe: Anzahl der Teilbuchungen
LPSTR pszWaehrung {5}	Rückgabe: Währung bei einer Fremdwährungsbuchung
BOOL* pUstdirekt	Rückgabe: ungleich 0 (Null): Die Umsatzsteuer wird direkt gebucht (nur bei Istversteuerung interessant)
BOOL* pKeinOP	Rückgabe: ungleich 0 (Null): Zu dieser Buchung wird kein OP erzeugt
LPSTR pszZahlziel {11}	Rückgabe: Das zu Grunde liegende Zahlziel
BOOL* pUnvollst	Rückgabe: ungleich 0 (Null): Diese Buchung ist unvollständig. Nur im Stapel „Unvollständige Buchungen“
BOOL* pIstEB	Rückgabe: ungleich 0 (Null): Diese Buchung ist eine Eröffnungsbuchung
BOOL* pIstAB	Rückgabe: ungleich 0 (Null): Diese Buchung ist eine Abschlussbuchung
BOOL* pIstUB	Rückgabe: ungleich 0 (Null): Diese Buchung ist eine Umbuchung
BOOL* pUeberweisung	Rückgabe: ungleich 0 (Null): Zu dieser Buchung gibt es eine Überweisung
LPSTR pszDmsID{51}	Rückgabe: DMS-ID

Rückgabewert:

long lErrno OFCKweitererDatensatz: Es gibt noch weitere Batchbuchungen.

OFCK_letzter_Datensatz: Dies ist die letzte Batchbuchung

<=0: Fehlernummer

OFCK_HoleBatchSplit

long WINAPI OFCK_HoleBatchSplit(long* pISplitnr, LPSTR pszText, LPSTR pszSollkonto, long* pISollart, LPSTR pszHabenkonto, long* pIHabenart, LPSTR pszBetrag, LPSTR pszEurobetrag, LPSTR pszFWBetrag, LPSTR pszSteuersatz, LPSTR pszSteuerbetrag, LPSTR pszEurosteuerbetrag, long* pIKobuanz, long lAusgabeart, LPSTR pszText2)

Mit dieser Funktion werden der Reihe nach alle Splitteilbuchungen zu einer Splitbuchung in Erfahrung gebracht. Für alle Rückgabewerte kann der NULL-Zeiger übergeben werden. Es werden keine Durchbuchungen und keine Rundungsdifferenz-Buchungen zurückgegeben, da diese ggf. erst bei endgültiger Verbuchung entstehen.

Parameter

long* lSplitnr	Rückgabe: Nummer des Splits
LPSTR pszText {36}	Rückgabe: Buchungstext
LPSTR pszSollkonto {8}	Rückgabe: Sollkonto
long* pISollart	Rückgabe: Art des Sollkontos (OFCK_SAKO, OFCK_DEBI, OFCK_KRED)
LPSTR pszHabenkonto {8}	Rückgabe: Habenkonto
long* pIHabenart	Rückgabe: Art des Habenkontos (OFCK_SAKO, OFCK_DEBI, OFCK_KRED)
LPSTR pszBetrag {16}	Rückgabe: Bruttobetrag in Landeswährung
LPSTR pszEurobetrag {16}	Rückgabe: Bruttobetrag in EUR
LPSTR pszFWBetrag {16}	Rückgabe: Bruttobetrag in Fremdwährung oder leere Zeichenkette, falls keine Fremdwährungsbuchung (erkennbar am Währungskürzel)
LPSTR pszSteuersatz {5}	Rückgabe: Steuersatz der Buchung
LPSTR pszSteuerbetrag {16}	Rückgabe: Steuerbetrag in Landeswährung
LPSTR pszEurosteuerbetrag {16}	Rückgabe: Steuerbetrag in EUR
long* pIKobuanz	Rückgabe: Anzahl der Kostenbuchungen
long lAusgabeart	Ausgabeart der Beträge (OFCK_FIXLEN, OFCK_VARLEN, OFCK_AUSGABE)
LPSTR pszText2 {36}	Rückgabe: 2. Buchungstext

Rückgabewert:

long lErrno	OFCKweitererDatensatz: Es gibt noch weitere Splitteilbuchungen. OFCKletzterDatensatz: Dies ist die letzte Splitteilbuchung. Ob es noch weitere Batchbuchungen gibt, kann nur aus dem Rückgabewert von OFCK_HoleBatchHeader herausgelesen werden. <=0: Fehlernummer
-------------	--

OFCK_HoleBatchKobu

long WINAPI OFCK_HoleBatchKobu(LPSTR pszKost1, LPSTR pszKost2, LPSTR pszKostenart, LPSTR pszBetrag, LPSTR pszEurobetrag, long* plKostenkennung, LPSTR pszBemerkung, long lAusgabeart, LPSTR pszBemerkung2, LPSTR pszKost3, LPSTR pszKost4, LPSTR pszKost5, LPSTR pszKost6)

Mit dieser Funktion lassen sich die Kostenbuchungen zu einer Splitteilm Buchung holen, sofern Kostenbuchungen vorhanden sind (OFCK_HoleBatchSplit lieferte zuvor im Parameter plKobuanz einen Wert > 0 zurück). Für alle Rückgabewerte kann der NULL-Zeiger übergeben werden. Die Summe der Kostenbuchungen zu einer Splitteilm Buchung entspricht deren Nettobetrag.

Parameter

LPSTR pszKost1 {8}	Rückgabe: Kostenstelle 1
LPSTR pszKost2 {8}	Rückgabe: Kostenstelle 2
LPSTR pszKostenart {8}	Rückgabe: Kostenart
LPSTR pszBetrag {16}	Rückgabe: Betrag der Kostenbuchung in Landeswährung
LPSTR pszEurobetrag {16}	Rückgabe: Betrag der Kostenbuchung in EUR
long* plKostenkennung	Rückgabe: Kostenkennung (OFCK_FIX, OFCK_VARIABEL)
LPSTR pszBemerkung {41}	Rückgabe: Bemerkungstext
long lAusgabeart	Ausgabeart der Beträge (OFCK_FIXLEN, OFCK_VARLEN, OFCK_AUSGABE)
LPSTR pszBemerkung2 {41}	Rückgabe: 2. Bemerkungstext
LPSTR pszKost3 {8}	Rückgabe: Kostenstelle 3
LPSTR pszKost4 {8}	Rückgabe: Kostenstelle 4
LPSTR pszKost5 {8}	Rückgabe: Kostenstelle 5
LPSTR pszKost6 {8}	Rückgabe: Kostenstelle 6

Rückgabewert:

long lErrno	OFCK_weitererDatensatz: Es gibt noch weitere Kostenbuchungen. OFCK_letzterDatensatz: Dies ist die letzte Kostenbuchung. Ob es noch weitere Batchbuchungen gibt, kann nur aus dem Rückgabewert von OFCK_HoleBatchHeader herausgelesen werden. Sinngemäß für weitere Splitteilm Buchungen. <=0: Fehlernummer
-------------	--

10. Kostenrechnung

OFCK_Kost

long WINAPI **OFCK_Kost**(long IMandnr, long IAktion, OFCK_KOSTENSTELLE zKost, LPSTR lpszAbjahr, BOOL bShowErr)

Mit dieser Funktion können Kostenstellen und Kostenträger angelegt, geändert oder gelöscht werden.

Parameter:

long IMandnr Der Mandant, für den die Kostenstelle/träger eingefügt werden soll.
long IAktion Mögliche Aktionen:
- OFCK_INSERT: Fügt eine Kostenstelle/träger ein.
- OFCK_UPDATE: Ändert die Daten der Kostenstelle/träger anhand der Kostenstellen/trägernummer.
- OFCK_DELETE: Löscht die Kostenstelle/träger anhand der Kontonummer. Dies geht nur, wenn die Kostenstelle/träger löschar (unbebuch) ist.

OFCK_KOSTENSTELLE Daten der Kostenstelle/ träger. Definition der Struktur siehe unten.

zKost

LPSTR lpszAbjahr Erstes Anlagejahr. Format: 2- oder 4-stellig. Wird NULL oder ein Leerstring angegeben, so wird das aktuelle Jahr verwendet. Es werden Kostenstellen/träger ab diesem Jahr bis zum aktuellen Buchungsjahr angelegt. Werden die gültigen Werte überschritten, so wird das erste Jahr (Jahr vor erstem Jahr) oder das letzte Jahr (Jahr nach aktuellem Jahr) gesetzt.

BOOL bShowErr Flag, ob Fehlermeldungen als Dialoge angezeigt werden sollen.

Rückgabewert:

long lErrno OFCK_OK oder Fehlernummer <= 0:
- OFCKERR_uzl_Aktion: es wurde weder OFCK_INSERT noch OFCK_UPDATE noch OFCK_DELETE als Aktion angegeben.

Bemerkungen:

- Die Steuerkennung wird nur bei Mandanten mit Vereinsbuchführung ausgewertet.
- Die Kostenstellen/trägernummer muss über alle Kostenträger und Kostenstellen eindeutig sein.

```
typedef struct
{
    long steuerkennung;        // Steuersatz (Verein)
                             // e { OFCK_NOSTEUER = Steuerfrei
                             //       OFCK_ABZUG     = Abzugsfähige VSt.
                             //       OFCK_AUFTEIL   = Aufzuteilende VSt. }
    long dimension;            // Dimension 0-5
    long saldovortrag;        // 1 = falls Saldo am Jahresende beibehalten
                             // werden soll, 0 sonst
    char kontonr[8];           // Nummer, eindeutig
    char kuerzel[9];           // Kuerzel, eindeutig
    char name[31];            // Bezeichnung / Name
    char bemerkung[41];       // Bemerkung
    char startdatum[7];       // "TTMMJJ" oder leer: Start der
                             // Laufzeitbegrenzung
    char enddatum[7];         // "TTMMJJ" oder leer: Ende der Laufzeitbegrenzung
                             // Laufzeitbegrenzung nur bei
                             // aktivierter EURO KORE
                             // Beide Daten oder keine von beiden sind
                             // anzugeben.
    char kostref[8];          // Kostenstellenreferenz (wird verwendet bei
                             // aktivierter Referenzierung und dimension=4)
} OFCK_KOSTENSTELLE;
```

OFCK_KostDirekt

long WINAPI **OFCK_KostDirekt**(long IMandnr, long IAction, long steuererkennung, long dimension, BOOL ISaldovortrag, LPSTR kontonr, LPSTR kuerzel, LPSTR name, LPSTR bemerkung, LPSTR IpszAbjahr, BOOL bShowErr, LPSTR startdatum, LPSTR enddatum, LPSTR IpszKostref)

Diese Funktion entspricht der OFCK_Kost-Funktion, nur werden die Kostenstellen/träger-Parameter direkt übergeben. Die einzelnen Parameter entsprechen denen in der OFCK_KOSTENSTELLE-Struktur.

Rückgabewert: zusätzlich zu OFCK_Kost
long IErrno - OFCKERR_uzl_Format. Kostenstellen/träger-
nummer/kürzel/name/bemerkung ist zu lang

OFCK_HoleKostenSaldo

long WINAPI OFCK_HoleKostenSaldo(long IMandnr, LPCSTR IpszNummer, LPCSTR IpszKostenart, long IAufbau, long IZeitraum, long IJahr, BOOL bShowErr, long IAusgabeart, LPSTR IpszEuroSaldo, long IDimension, LPCSTR IpszKostref)

Ermittelt den Saldo einer Kostenstelle oder eines Kostenträgers pro Kostenart.

Parameter:

long IMandnr	Mandantenummer
LPCSTR IpszNummer	Die Nummer der Kostenstelle/des Kostenträgers für welchen der Saldo geholt werden soll.
LPCSTR IpszKostenart	Die Nummer der Kostenart für welche der Saldo geholt werden soll.
long IAufbau	Typ des Aufbaus, zu dem die Werte ermittelt werden. Mögliche Werte: OFCK_AUFGUV für den Aufbau der GuV, OFCK_AUFGUV2 für den Aufbau der 2. GuV, OFCK_AUFGUV3 für den Aufbau der 3. GuV, OFCK_AUFBILAKT für den Aufbau der Bilanz Aktiva, OFCK_AUFBILAKT2 für den Aufbau der 2. Bilanz Aktiva, OFCK_AUFBILAKT3 für den Aufbau der 3. Bilanz Aktiva, OFCK_AUFBILPAS für den Aufbau der Bilanz Passiva, OFCK_AUFBILPAS2 für den Aufbau der 2. Bilanz Passiva, OFCK_AUFBILPAS3 für den Aufbau der 3. Bilanz Passiva, OFCK_AUFBWA1, OFCK_AUFBWA2, OFCK_AUFBWA3, OFCK_AUFBWA4, OFCK_AUFBWA5, OFCK_AUFBWA6 für den 1. – 6. benutzerdefinierten Aufbau.
long IZeitraum	Zeitraum, für den der Saldo berechnet werden soll. Der Zeitraum bezieht sich stets auf den Geschäftsjahresbeginn des gewählten Mandanten. Umsätze können monatsweise ermittelt werden: IZeitraum = 1...12 (1 = erster Monat des Geschäftsjahres, 2 = zweiter Monat usw.), IZeitraum = 13...16 (13 = erstes Quartal des Geschäftsjahres, 14 = zweites Quartal usw.), IZeitraum = 17 (Umsatz aller Monate des Geschäftsjahres),
long IJahr	Das Jahr für welches der Saldo ermittelt werden soll.
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen.
long IAusgabeart	Ausgabeart des Saldos (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, Beschreibung der Ausgabearten unter OFCK_HoleSaldo).
LPSTR IpszEuroSaldo{16}	Der Saldo (in Euro). Für diesen Wert sind 16 Byte zu reservieren.
long IDimension	Einschränkung auf die Dimension bei Kostenstellen.
LPCSTR IpszKostref	Einschränkung auf die Kostenstellenreferenz bei Dimension=4 und

aktivierter Kostenstellenreferenzierung für den Mandant.

Rückgabewert:

long lErrno OFCK_OK. Der Saldo wurde erfolgreich ermittelt.
OFCKERR_uzl_Mandnr: Der Mandant ist unzulässig.
OFCKERR_uzl_Kontonr: Die Kostenstelle/der Kostenträger oder die
Kostenart ist unzulässig.
OFCKERR_uzl_Zeitraum: Der Zeitraum ist ungültig.

OFCK_HoleKostenschablone

long WINAPI **OFCK_HoleKostenschablone**(HWND hParent, long lMandnr, LPSTR lpszSchablonenName)

Dient der Auswahl der Nummer einer Kostenschablone. Diese kann bei einer Buchung mit übergeben werden, um eine andere Kostenaufteilung zu veranlassen, als diese im Sachkonto hinterlegt ist. Es erscheint ein Dialog zur Kostenschabloneauswahl mit allen angelegten Kostenschablonen. Dieser ähnelt der Mandantenauswahl.

Parameter:

HWND hParent Handle des Parent-Fensters unter dem der Dialog zur
Kostenschabloneauswahl erscheinen soll.

long lMandnr Mandantenummer.
LPSTR
lpszSchablonenName Falls ungleich NULL: Enthält nach erfolgreichem Aufruf den ermittelten
Kostenschablonennamen. Es müssen mindestens 41 Byte für diesen
Rückgabewert reserviert werden.

Rückgabewert:

long lKoschanr Kostenschablonennummer > 0; 0 bei Dialogabbruch; Fehlernummer < 0.

OFCK_InitHoleKoscha

long WINAPI **OFCK_InitHoleKoscha**(long lMandnr)

Leitet das Holen von Kostenschablonen ein.

Parameter:

long lMandnr Mandantenummer.

Rückgabewert:

long lErrno OFCK_OK, falls das holen der Kostenschablonen erfolgreich initialisiert
wurde; sonst Fehlernummer <= 0.

OFCK_HoleKoscha

long WINAPI **OFCK_HoleKoscha**(long* plSchablonennr, LPSTR pszSchablonenName, BOOL* plAutomatik, long* plZeilenanzahl, LPSTR pszBezGroesse, LPSTR pszBezEinheit, long lAusgabeart)

Dient zum Holen der ersten und den darauf folgenden Kostenschablonen.

Parameter:

long* plSchablonennr Falls ungleich NULL: Enthält nach erfolgreichem Aufruf die
Schablonennummer.

LPSTR
pszSchablonenName Falls ungleich NULL: Enthält nach erfolgreichem Aufruf den ermittelten
Kostenschablonennamen. Es müssen mindestens 41 Byte für diesen
Rückgabewert reserviert werden.

BOOL* plAutomatik	Falls ungleich NULL: Wenn ungleich 0 ist die Automatik der Kostenschablone aktiviert.
long* plZeilenanzahl	Falls ungleich NULL: Enthält die Anzahl der zugehörigen Zeilen.
LPSTR pszBezGroesse	Falls ungleich NULL: Enthält die Bezugsgröße. Es müssen mindestens 13 Byte für diesen Rückgabewert reserviert werden.
LPSTR pszBezEinheit	Falls ungleich NULL: Enthält die Einheit der Bezugsgröße.
long lAusgabeart	Bestimmt die Ausgabeart der Bezugsgröße.

Rückgabewert:

long lErrno	OFCK_OK, falls die Kostenschablone erfolgreich geholt wurde; sonst Fehlernummer <= 0.
-------------	---

OFCK_HoleKoschaZeile

long WINAPI **OFCK_HoleKoschaZeile**(LPSTR pszKost1, LPSTR pszKost2, long* plKostenkennung, LPSTR pszAnteil, LPSTR pszBemerkung, LPSTR pszBezAnteil, long lAusgabeart, LPSTR szKostenstelle3, LPSTR szKostenstelle4, LPSTR szKostenstelle5, LPSTR szKostenstelle6)

Dient zum Holen der Kostenschablonenzeilen welche der aktuell geholten Kostenschablone angehören.

Parameter:

LPSTR pszKost1	Falls ungleich NULL: Enthält die Kostenstelle 1.
LPSTR pszKost2	Falls ungleich NULL: Enthält die Kostenstelle 2.
LPSTR lpszSchablonenName	Falls ungleich NULL: Enthält nach erfolgreichem Aufruf den ermittelten Kostenschablonenname. Es müssen mindestens 41 Byte für diesen Rückgabewert reserviert werden.
LPSTR pszKost3	Falls ungleich NULL: Enthält die Kostenstelle 3.
LPSTR pszKost4	Falls ungleich NULL: Enthält die Kostenstelle 4.
LPSTR pszKost5	Falls ungleich NULL: Enthält die Kostenstelle 5.
LPSTR pszKost6	Falls ungleich NULL: Enthält die Kostenstelle 6.

Rückgabewert:

long lErrno	OFCK_OK, falls die Kostenschablonenzeile erfolgreich geholt wurde; sonst Fehlernummer <= 0.
-------------	---

OFCK_BeendeHoleKoscha

long WINAPI **OFCK_BeendeHoleKoscha**()

Diese Funktion muss aufgerufen werden, wenn das Holen der Kostenschablonen mit HoleKoscha vor dem Holen des letzten Datensatzes beendet wird.

Rückgabewert:

long lErrno	OFCK_OK, falls das holen der Datensätze erfolgreich beendet wurde; sonst Fehlernummer <= 0.
-------------	---

OFCK_HoleEKROptionen

long WINAPI **OFCK_HoleEKROptionen**(long lMandnr, BOOL* plKleineKostenrechnung, BOOL* plKostenstellenpflicht, BOOL* plEinfuehrung, BOOL* plEKRDemo, LPSTR pszEKRStartdatum)

Diese Funktion holt die Optionen für die EURO KORE.

Parameter

long lMandnr	Mandantenummer.
BOOL* pIKleineKostenrechnung	Falls ungleich NULL: Pointer auf die Option kleine Kostenrechnung. Die Option ist aktiviert wenn pIKleineKostenrechnung != 0.
BOOL* pIKostenstellenpflicht	Falls ungleich NULL: Pointer auf die Option Kostenstellenpflicht. Die Option ist aktiviert wenn pIKostenstellenpflicht != 0.
BOOL* pIEinfuehrung	Falls ungleich NULL: Pointer auf die Option EURO KORE Einführung. Die Option ist aktiviert wenn pIEinfuehrung != 0.
BOOL* pIEKRDemo	Falls ungleich NULL: Pointer auf die Option EURO KORE Demo. Die Option ist aktiviert wenn pIEKRDemo != 0.
LPSTR pszEKRStartdatum	Falls ungleich NULL: Pointer auf das EURO KORE Einführungsdatum. Wenn die EURO KORE nicht eingeführt wurde wird dieses Feld nicht belegt.

Rückgabewert:

long lErrno	OFCK_OK bei erfolgreichem holen der EURO KORE Optionen; Fehlernummer <= 0.
-------------	--

11. Aufbauten

OFCK_InitHoleAufbau

long WINAPI **OFCK_InitHoleAufbau**(long lMandnr, long lKlasse, BOOL bShowErr)

Diese Funktion leitet das Holen von Aufbaueinträgen ein. Sie muss vor dem Aufruf von OFCK_HoleAufbau aufgerufen werden.

Parameter

long lMandnr	Mandantenummer, zu dem die Aufbauten geholt werden sollen.
long lKlasse	Typ des Aufbaus, der gelesen werden soll. Mögliche Werte: OFCK_AUFGUV für den Aufbau der GuV, OFCK_AUFGUV2 für den Aufbau der 2. GuV, OFCK_AUFGUV3 für den Aufbau der 3. GuV, OFCK_AUFBILAKT für den Aufbau der Bilanz Aktiva, OFCK_AUFBILAKT2 für den Aufbau der 2. Bilanz Aktiva, OFCK_AUFBILAKT3 für den Aufbau der 3. Bilanz Aktiva, OFCK_AUFBILPAS für den Aufbau der Bilanz Passiva, OFCK_AUFBILPAS2 für den Aufbau der 2. Bilanz Passiva, OFCK_AUFBILPAS3 für den Aufbau der 3. Bilanz Passiva, OFCK_AUFBWA1, OFCK_AUFBWA2, OFCK_AUFBWA3, OFCK_AUFBWA4, OFCK_AUFBWA5, OFCK_AUFBWA6 für den 1. - 6. benutzerdefinierten Aufbau. OFCK_AUFGKOST1, OFCK_AUFGKOST2, OFCK_AUFGKOST3, OFCK_AUFGKOST4, OFCK_AUFGKOST5, OFCK_AUFGKOST6 für die 1. – 6. Kostenstellenhierarchie. OFCK_AUFGKOART1, OFCK_AUFGKOART2, OFCK_AUFGKOART3, OFCK_AUFGKOART4, OFCK_AUFGKOART5, OFCK_AUFGKOART6 für die 1. – 6. Kostenartenhierarchie OFCK_AUFGKOTR1, OFCK_AUFGKOTR2, OFCK_AUFGKOTR3, OFCK_AUFGKOTR4, OFCK_AUFGKOTR5, OFCK_AUFGKOTR6 für die 1. – 6. Kostenträgerhierarchie Zusätzlich kann die Option OFCK_MIT AUSWEICHAUFBAU übergeben werden. Die Option bewirkt, dass zusätzlich der Ausweichsaufbau geholt wird.
BOOL bShowErr	Fehler anzeigen.

Rückgabewert:

long lErrno	OFCK_weitererDatensatz: Es sind weitere Datensätze vorhanden. OFCK_keinDatensatz: Es sind keine Datensätze vorhanden. <0: Fehlernummer
-------------	--

OFCK_HoleAufbau

long WINAPI **OFCK_HoleAufbau**(long* plArt, LPSTR pszName, long* plEbene)

Diese Funktion dient der schrittweisen Abfrage der Aufbaueinträge. Vor dem ersten Aufruf muss die Funktion OFCK_InitHoleAufbau aufgerufen werden.

Die Einträge werden der Reihe nach ausgegeben. Untereinträge können an der tieferen Ebene (größere Nummer in *plEbene erkannt werden. Konten ohne Zuordnung kommen am Schluss und sind auf Ebene 0.

Parameter

long* plArt	Rückgabe: Dieser Wert beschreibt die Art des Eintrages. Mögliche Werte: OFCK_GRUPPE: Der Eintrag ist ein Gruppeneintrag. In pszName steht dann der Beschreibungstext der Gruppe OFCK_KONTO: Der Eintrag ist ein Konto. In pszName steht die Kontonummer. Wenn beim Aufruf der Funktion OFCK_InitHoleAufbau zusätzlich die Option OFCK_MIT AUSWEICHAUFBAU übergeben wurde, wird bei einem Konto das dem Ausweichsaufbau zugeordnet ist, zusätzlich der Wert OFCK_AUSWEICHAUFBAU zurückgegeben. (OFCK_KONTO OFCK_AUSWEICHAUFBAU)
LPSTR pszName {51}	Rückgabe: Der Name des Eintrages. Ist der Eintrag ein Konto steht hier eine Kontonummer. Es müssen mindestens 51 Bytes für diesem Rückgabewert reserviert werden.
long* plEbene	Rückgabe: Die Ebene des Eintrages. Gruppeneinträge und zugeordnete Konten haben eine Ebene >= 1, nicht zugeordnete Konten haben die Ebene 0.

Rückgabewert:

long lErrno	OFCKweitererDatensatz: Es sind weitere Datensätze vorhanden. OFCKletzterDatensatz: Es sind keine weiteren Datensätze vorhanden. OFCKERR_keinInit: Es wurde vergessen die Funktion OFCK_InitHoleAufbau aufzurufen. <0: Fehlernummer.
-------------	--

OFCK_HoleAufbauID

long WINAPI **OFCK_HoleAufbauID**(long* plArt, LPSTR pszName, long* plEbene, LPSTR pszID)

Diese Funktion dient der schrittweisen Abfrage der Aufbaueinträge. Vor dem ersten Aufruf muss die Funktion OFCK_InitHoleAufbau aufgerufen werden.

Die Einträge werden der Reihe nach ausgegeben. Untereinträge können an der tieferen Ebene (größere Nummer in *plEbene erkannt werden. Konten ohne Zuordnung kommen am Schluss und sind auf Ebene 0.

Parameter

long* plArt	Rückgabe: Dieser Wert beschreibt die Art des Eintrages. Mögliche Werte: OFCK_GRUPPE: Der Eintrag ist ein Gruppeneintrag. In pszName steht dann der Beschreibungstext der Gruppe OFCK_KONTO: Der Eintrag ist ein Konto. In pszName steht die Kontonummer.
LPSTR pszName {51}	Rückgabe: Der Name des Eintrages. Ist der Eintrag ein Konto steht hier eine Kontonummer. Es müssen mindestens 51 Bytes für diesem Rückgabewert reserviert werden.
long* plEbene	Rückgabe: Die Ebene des Eintrages. Gruppeneinträge und zugeordnete Konten haben eine Ebene >= 1, nicht zugeordnete Konten haben die Ebene 0.
LPSTR pszID {11}	Rückgabe: Die ID der Aufbaugruppe, sofern vorhanden. Bei Konten wird die Kontonummer zurückgegeben. Es müssen mind. 11 Bytes für diesem Rückgabewert reserviert werden.

Rückgabewert:

long lErrno OFCK_weitererDatensatz: Es sind weitere Datensätze vorhanden.
 OFCK_letzterDatensatz: Es sind keine weiteren Datensätze vorhanden.
 OFCKERR_keinInit: Es wurde vergessen die Funktion OFCK_InitHoleAufbau aufzurufen.
 <0: Fehlernummer.

OFCK_InitHoleAufbauSaldo

long WINAPI OFCK_InitHoleAufbauSaldo(long lMandnr, long lAufbau, long lZeitraum, long lJahr, BOOL bEuro, BOOL bEröffnungsbilanz, BOOL bVorschau)

Diese Funktion leitet das Holen von Werten aus den einzelnen Aufbauten ein. Die Werte werden mit diesem Aufruf bestimmt und werden bis zum Aufruf von OFCK_BeendeHoleAufbauSaldo() intern vorgehalten.

Die Werte selbst werden mit der Funktion OFCK_HoleAufbauSaldo() geholt. Sind alle Werte geholt, sollte die Funktion OFCK_BeendeHoleAufbauSaldo() aufgerufen werden, um den intern benötigten Speicher wieder freizugeben.

Parameter:

long lMandnr	Mandantenummer
long lAufbau	Typ des Aufbaus, zu dem die Werte ermittelt werden. Mögliche Werte: OFCK_AUFGUV für den Aufbau der GuV, OFCK_AUFGUV2 für den Aufbau der 2. GuV, OFCK_AUFGUV3 für den Aufbau der 3. GuV, OFCK_AUFBILAKT für den Aufbau der Bilanz Aktiva, OFCK_AUFBILAKT2 für den Aufbau der 2. Bilanz Aktiva, OFCK_AUFBILAKT3 für den Aufbau der 3. Bilanz Aktiva, OFCK_AUFBILPAS für den Aufbau der Bilanz Passiva, OFCK_AUFBILPAS2 für den Aufbau der 2. Bilanz Passiva, OFCK_AUFBILPAS3 für den Aufbau der 3. Bilanz Passiva, OFCK_AUFBWA1, OFCK_AUFBWA2, OFCK_AUFBWA3, OFCK_AUFBWA4, OFCK_AUFBWA5, OFCK_AUFBWA6 für den 1. – 6. benutzerdefinierten Aufbau.
long lZeitraum	Zeitraum, für den der Saldo berechnet werden soll. Der Zeitraum bezieht sich bei Quartalen und beim Gesamtjahr auf den Geschäftsjahresbeginn des gewählten Mandanten, bei den Monaten auf das Kalenderjahr. <ul style="list-style-type: none"> • Umsätze können monatsweise ermittelt werden: lZeitraum = 1...12 (1 = Januar, 2 = Februar usw.), • Quartalsweise: lZeitraum = 13...16 (13 = erstes Quartal des Geschäftsjahres, 14 = zweites Quartal usw.), • Jahresweise: lZeitraum = 17.
long lJahr	Das Jahr, zu dem die Werte geholt werden. Bei einem abw. Geschäftsjahr das erste Jahr bzw. das Jahr des Geschäftsjahresbeginns.
BOOL bEuro	Legt fest, ob die Werte in EUR oder in alter Landeswährung geholt werden. SwissFibu: Ist dieser Parameter 1, so wird der Fremdwährungssaldo des Kontos geholt, sofern vorhanden. Zu Gruppen und Zwischensummen existiert kein Fremdwährungssaldo.
BOOL bEröffnungsbilanz	Holt die Werte der Eröffnungsbilanz.
BOOL bVorschau	Berücksichtigt Stapelbuchungen, d.h. diese werden mit eingerechnet.

Rückgabewert:

long lErrno OFCK_OK: Die Werte wurden ermittelt

OFCKERR_uzl_Zeitraum: Der Parameter IZeitraum ist kleiner als 0 oder größer als 17

OFCKERR_uzl_Datum: Der verlangte Zeitraum liegt nicht in einem vorhandenen Jahr

Sonst: <=0: Fehlernummer

OFCK_HoleAufbauSaldo

long WINAPI OFCK_HoleAufbauSaldo(LPCSTR szKonto, LPCSTR szKuerzel, long IAusgabeart, LPSTR pszWert, BOOL* pbZwischensumme, long* pEbene)

Holt den Wert bzw. Saldo eines Kontos, eines Aufbaueintrages oder den Gesamtsaldo des Aufbaus. Für Konten ist die Kontonummer anzugeben. Bei Aufbaueinträgen wie Gruppen und Zwischensummen ist szKonto leer zu lassen und stattdessen in szKuerzel die ID des Eintrages anzugeben (diese ID wird innerhalb der Finanzbuchhaltung einer Aufbaugruppe oder Zwischensumme zugeordnet). Für den Gesamtsaldo ist sowohl das Konto als auch das Kürzel leer zu lassen.

Parameter:

LPCSTR szKonto	Kontonummer, zu welcher der Saldo ermittelt werden soll.
LPCSTR szKuerzel	Kürzel der Gruppe oder Zwischensumme, zu welcher ein Saldo ermittelt werden soll. Es wird zwischen Groß- und Kleinschreibung unterschieden. Ist sowohl eine Kontonummer als auch ein Kürzel angegeben, so wird das Konto genommen.
long IAusgabeart	Ausgabeart des Betrages (OFCK_VARLEN, OFCK_FIXLEN oder OFCK_AUSGABE, Beschreibung der Ausgabearten unter OFCK_HoleSaldo.
LPSTR pszWert {16}	Betrag. Hierfür sind 16 Zeichen zu reservieren.
BOOL* pbZwischensumme	Gibt an, ob es sich bei dem Eintrag um eine Zwischensumme (ungleich 0) oder um eine Gruppe (gleich 0) handelt. Dieser Wert wird nur bei Gruppen oder Zwischensummen gesetzt, wenn also szKonto nichts enthält.
long* pEbene	Gibt die Ebene des Eintrages zurück (1-10). Dieser Wert wird nur bei Gruppen oder Zwischensummen gesetzt, wenn also szKonto nichts enthält.

Rückgabewert:

long lErrno	<ul style="list-style-type: none">• OFCK_OK: alles OK.• OFCK_keinDatensatz: Es gibt keinen Eintrag mit dieser Kontonummer oder diesem Kürzel. Bilanz: Möglicherweise befindet sich dieser Wert im dazugehörigen Ausweichaufbau• OFCKERR_keinInit: Die Funktion OFCK_InitHoleAufbauSaldo wurde nicht aufgerufen.• OFCKERR_uzl_Kontonr: Es wurde weder eine Kontonummer noch ein Kürzel angegeben.• OFCKERR_uzl_Format: Es wurde kein gültiger Wert für die Ausgabeart angegeben.
-------------	---

OFCK_BeendeHoleAufbauSaldo

long WINAPI OFCK_BeendeHoleAufbauSaldo()

Beendet das Holen von Werten. Der Speicher für die intern vorgehaltenen Werte wird dabei freigegeben.

Parameter: keine

Rückgabewert:
long lErrno OFCK_OK. Der Speicher wurde freigegeben.

12. sonstige Funktionen

OFCK_HoleZahlungsziel

long WINAPI **OFCK_HoleZahlungsziel**(HWND hParent, long lMandnr, LPSTR lpszZahlungszielName, long* plSkontotage, LPSTR lpszSkonto, long* plSkontotage2, LPSTR lpszSkonto2, long* plZahltag, long* plTagInMonat, long* plMonatePlus, LPSTR lpszBemerkung, long* plZahlungsart, long lZZNummer, long* plNaechsteZZNummer)

Dient der Auswahl der Nummer eines Zahlungszieles. Diese kann bei der Anlage oder Änderung eines Personenkontos (siehe OFCK_Peko und OFCK_PekoDirekt) mit übergeben werden, so dass der Anwender hierüber die Zahlungsbedingung bei einem Personenkonto hinterlegen kann. Es erscheint ein Dialog zur Zahlungszielauswahl mit allen angelegten Zahlungszielen. Dieser ähnelt der Mandantenauswahl.

Werden die Daten zu einem speziellen Zahlziel gewünscht, so kann dessen Nummer in lZZNummer (ungleich 0) übergeben werden. In diesem Fall erscheint kein Dialog. Zahlungsziele werden fortlaufend, bei 1 beginnend nummeriert. Die Nummer des nächsten gültigen Zahlzieles steht in plNaechsteZZNummer (>0) (sind keine weiteren Zahlziele vorhanden, so wird hier die Nummer 0 zurückgegeben.). Da sich Zahlziele in der Finanzbuchhaltung löschen lassen, können Lücken in der fortlaufenden Nummerierung vorhanden sein, auch am Anfang. Ob ein Zahlziel mit der übergebenen Nummer existiert, kann anhand des Rückgabewertes (<0) überprüft werden.

Die Parameter ab lpszZahlungszielName sind optional. Hier werden nähere Angaben zum gewählten Zahlungsziel zurückgegeben.

Parameter:

HWND hParent	Handle des Parent-Fensters unter dem der Dialog zur Zahlungszielauswahl erscheinen soll.
long lMandnr	Mandantenummer.
LPSTR lpszName {11}	Falls ungleich NULL: Enthält nach erfolgreichem Aufruf den ermittelten Zahlungszielnamen. Es müssen mindestens 11 Byte für diesen Rückgabewert reserviert werden.
long* plSkontotage	Falls ungleich NULL: Zeiger (Referenz) auf die 1. Skontofrist in Tagen (>= 0).
LPSTR lpszSkonto {6}	Falls ungleich NULL: 1. Skontosatz in Prozent (0,00 bis 99,99).
long* plSkontotage2	Falls ungleich NULL: Zeiger (Referenz) auf die 2. Skontofrist in Tagen (>= 0).
LPSTR lpszSkonto2 {6}	Falls ungleich NULL: 2. Skontosatz in Prozent (0,00 bis 99,99).
long* plZahltag	Falls ungleich NULL: Zeiger (Referenz) auf die Zahlungsfrist in Tagen (>= 0).
long* plTagInMonat	Falls ungleich NULL: zahlbar zu einem bestimmten Tag im Monat (1 bis 31).
long* plMonatePlus	Falls ungleich NULL: falls lTagInMonat gesetzt, dann wird hier die Anzahl der Folgemonate übergeben, in welchem der Zahlungstag liegen soll (0 bis 99).
LPSTR lpszBemerkung {41}	Falls ungleich NULL: die Bemerkung wird zurückgegeben.
Long* plZahlungsart	Falls ungleich NULL: Die Zahlungsart wird zurückgegeben. Mögliche Werte: OFCK_SONSTIGE, OFCK_UEBERWEISUNG, OFCK_ABBUCHUNG, OFCK_EINZUG, OFCK_SCHECK, OFCK_POS
long lZZNummer	Falls ungleich 0: Es werden direkt die Daten zu diesem Zahlziel geholt, es erscheint kein Dialog.
long* plNaechsteZZNummer	Falls ungleich NULL: Enthält die Nummer des nächsten Zahlzieles (>0) oder 0.

Rückgabewert:

long lZahlzielnr

Zahlungszielnummer > 0; 0 bei Dialogabbruch; Fehlernummer < 0.

OFCK_Zahlungsziel

long WINAPI **OFCK_Zahlungsziel**(long lMandnr, long lAktion, long lZahlungszielnummer, LPSTR lpszName, long lSkontotage, LPSTR lpszSkonto, long lSkontotage2, LPSTR lpszSkonto2, long lZahltag, long lTagInMonat, long lMonatePlus, LPSTR lpszBemerkung, long lZahlungsart, BOOL bShowErr)

Dient der Neuanlage oder Änderung eines Zahlungszieles. Die Zahlungs- bzw. Fälligkeitsfrist kann entweder in Tagen (z.B. zahlbar in 14 Tagen) oder zu einem bestimmten Tag in einem der Folgemonate übergeben werden (z.B. zahlbar am 15. des Folgemonats).

Parameter:

long lMandnr	Mandantenummer.
long lAktion	Aktion (OFCK_INSERT, OFCK_UPDATE).
long lZahlungszielnummer	Hier wird bei einem Update die Nummer des Zahlungszieles übergeben, welches verändert werden soll. Bei einem Insert ist dieser Parameter ohne Bedeutung.
LPSTR lpszName {11}	Eindeutiger Name des Zahlungszieles (max. 10 Zeichen).
long lSkontotage	1. Skontofrist in Tagen (0 bis 999).
LPSTR lpszSkonto {6}	1. Skontosatz in Prozent (0,00 bis 99,99).
long lSkontotage2	2. Skontofrist in Tagen (0 bis 999).
LPSTR lpszSkonto2	2. Skontosatz in Prozent (0,00 bis 99,99).
long lZahltag	Zahlungsfrist in Tagen (0 bis 999).
long lTagInMonat	Zahlbar zu einem bestimmten Tag im Monat (1 bis 31), 0 falls die lZahltag genutzt werden sollen.
long lMonatePlus	Falls lTagInMonat gesetzt, dann wird hier die Anzahl der Folgemonate übergeben, in welchem der Zahlungstag liegen soll (0 bis 99).
LPSTR lpszBemerkung {41}	Bemerkung.
long lZahlungsart	Zahlungsart: Mögliche Werte: OFCK_SONSTIGE, OFCK_UEBERWEISUNG, OFCK_ABBUCHUNG, OFCK_EINZUG, OFCK_SCHECK, OFCK_POS
BOOL bShowErr	TRUE, falls Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno Zahlungszielnummer > 0 oder Fehlernummer < 0.

OFCK_HoleZZnr

long WINAPI **OFCK_HoleZZnr**(long lMandnr, LPSTR lpszKuerzel)

Holt zu einem Zahlungszielkürzel bzw. -namen dessen Nummer. Diese kann dann bei einer anschließend aufgerufenen Buchungsfunktion mit übergeben werden.

Parameter:

long lMandnr	Mandantenummer.
LPSTR lpszKuerzel	Zahlungszielkürzel bzw. -name.

Rückgabewert:

long lErrno Zahlungszielnummer > 0 oder Fehlernummer < 0.

OFCK_HoleAktMahnstufe

long WINAPI **OFCK_HoleAktMahnstufe**(long lMandnr, LPSTR lpszDebitorenKontonr)

Gibt zu einem Debitorenkonto die höchste aktuelle Mahnstufe zurück, die zum Zeitpunkt des Aufrufes in den vorhandenen OPs vorliegt. Dadurch kann z.B. die Warenwirtschaft eine Liefersperre zu einem Kunden vergeben, falls dieser eine gewisse Mahnstufe erreicht hat.

Parameter:

long lMandnr Mandantenummer.
LPSTR lpszDebitorenKontonr Kontonummer des Debitoren.

Rückgabewert:

long lMahnstufe Höchste aktuelle Mahnstufe; Fehlernummer < 0.

OFCK_Mandant

long **OFCK_Mandant**(HWND hParent, long lAktion, BOOL bDialog, OFCK_MANDANT* pzMand, BOOL bShowErr)

Mit dieser Funktion kann ein neuer Mandant angelegt werden oder die Firmenstammdaten eines existierenden Mandanten geändert werden.

Parameter:

HWND hParent Der Handle des Fensters, unter dem im Dialogbetrieb der Dialog erscheint.

long lAktion Mögliche Aktionen:
- OFCK_INSERT: Fügt einen Mandanten ein.
- OFCK_UPDATE: Ändert die Firmenstammdaten eines Mandanten.

BOOL bDialog Flag, ob die Aktion im Dialogbetrieb ausgeführt wird (TRUE) oder nicht (FALSE).

OFCK_MANDANT
* pzMand Zeiger auf die OFCK_MANDANT-Struktur mit den benötigten Daten. Beschreibung der Felder s.u. Im Dialogbetrieb kann dieser Parameter bei der Mandantenanlage auch NULL sein. Für einige Werte werden dann Standardwerte vorgeschlagen.

BOOL bShowErr Flag, ob Fehlermeldungen angezeigt werden sollen.

Rückgabewert:

long lErrno Mandantenummer > 0 oder Fehlernummer <= 0:
- OFCKERR_uzl_Data: Ein Parameter in der OFCK_MANDANT-Struktur ist mit einem ungültigen Wert belegt (z.B. Geschäftsjahrbeginn im Monat 0)
- OFCKERR_uzl_Format: Mind. ein Parameter in der OFCK_MANDANT-Struktur ist fehlerhaft belegt (nicht initialisiert)
- OFCKERR_uzl_Aktion: es wurde weder OFCK_INSERT noch OFCK_UPDATE als Aktion angegeben.
- OFCKERR_Abbruch: Abbruch eines Dialoges.
- OFCKERR_uzl_Ustid: Die Umsatzsteuer-ID stimmt nicht mit dem Format des angegebenen Landes überein.

```

// es können alle optionalen Daten sowie IpszMandantenname geändert werden
typedef struct
{
    long vereinsbuchfuehrung; // optional e { OFCK_JA | OFCK_NEIN }
    long geschaeftsjaehrbeginnMonat; // Pflicht e { 1..12 }
    long versteuerungsart; // Pflicht e { OFCK_VSOLL | OFCK_VIST | OFCK_VEUR }
    long kontenrahmen; // Pflicht e { OFCK_SKR03 | OFCK_SKR04 | OFCK_RLG |
    // OFCK_GKR | OFCK_KRIND | OFCK_KMU }
    long oesterreich; // optional e { 0=Deutschland, 1=Österreich }
    long swiss; // optional e { 0=Deutschland, 1=Schweiz }
    long bundesland; // optional e { 1..16, alphabetisch } (nur Deutschland)
    long grundwaehrung; // Pflicht e { OFCK_EURO | OFCK_LAND }
    long steuerfrei; // Pflicht e { OFCK_JA | OFCK_NEIN }
    // (nicht bei Ist-Versteuerung)
    long eurokore; // optional e { OFCK_JA | OFCK_NEIN } Freischaltung der
    // EURO KORE (nur bei entsprechender Lizenz)
    char geschaeftsjaehrbeginnJahr[5]; // Pflicht 2 oder 4 Stellen
    char mandantenname[41]; // Pflicht
    char zusatz[41]; // optional
    char strasse[31]; // optional
    char ort[31]; // optional
    char land[4]; // optional
    char telefon[21]; // optional
    char ustidnr[16]; // optional
    char steuernummer[16]; // optional
    char referatsnummer[11]; // optional (Oesterreich)
    char finanzamt[39]; // optional
    char fastrasse[31]; // optional
    char faort[31]; // optional
    char waehrung[4]; // Pflicht z.B. "DEM"
    char stbname[31]; // Steuerberater Name (optional)
    char stbzusatz[31]; // Steuerberater Zusatz (optional)
    char stbstrasse[31]; // Steuerberater Strasse (optional)
    char stbort[31]; // Steuerberater Ort (optional)
    char stbtel[21]; // Steuerberater Telefon (optional)
} OFCK_MANDANT;

```

Bemerkungen:

- Bei der Mandantenneuanlage werden alle obigen Daten berücksichtigt, bei einer Änderung der Firmenstammdaten können nur die optionalen Daten und der Mandantenname geändert werden.
- Bei einer Änderung der Daten werden in leeren Felder die bisherigen Daten vorgeschlagen (im Dialogbetrieb) bzw. übernommen (ohne Dialog).
- Bei Änderungen wird der Mandant anhand des Namens identifiziert, der dazu im entsprechenden Feld der Struktur übergeben werden muss. Eine Änderung des Mandantennamens ist daher nur im Dialogbetrieb möglich.
- Ein individueller Kontenrahmen ist nur im Dialogbetrieb möglich.
- Die EURO KORE wird nur bei Vorhandensein einer entsprechenden Lizenz aktiviert, nur Neuanlage.

OFCK_MandantDirekt

```

long WINAPI OFCK_MandantDirekt( HWND hParent, long lAktion, BOOL bDialog, long*
pIVereinsbuchfuehrung, long* pIGeschaeftsjaehrbeginnMonat, long* pIVersteuerungsart, long*
pIKontenrahmen, long* pIOesterreich, long* pIBundesland, long* pIGrundwaehrung, long* pISteuerfrei,
long* pIEurokore, LPSTR pszGeschaeftsjaehrbeginnJahr, LPSTR pszMandantenname, LPSTR
pszZusatz, LPSTR pszStrasse, LPSTR pszOrt, LPSTR pszLand, LPSTR pszTelefon, LPSTR
pszUstidnr, LPSTR pszSteuernummer, LPSTR pszReferatsnummer, LPSTR pszFinanzamt, LPSTR
pszFastrasse, LPSTR pszFaort, LPSTR pszWaehrung, LPSTR pszStbname, LPSTR pszStbzusatz,
LPSTR pszStbstrasse, LPSTR pszStbort, LPSTR pszStbtel, BOOL bShowErr )

```

Diese Funktion entspricht der OFCK_Mandant-Funktion, nur werden die Mandanten-Parameter direkt übergeben. Die einzelnen Parameter entsprechen denen in der OFCK_MANDANT-Struktur. Zusätzlich können für die Zeiger (long* und LPSTR) der NULL-Zeiger übergeben werden, dies wird behandelt als würde eine 0 bzw. ein Leerstring übergeben, mit Ausnahme der Pflichtfelder.

Rückgabewert: zusätzlich zu OFCK_Mandant

long lErrno Mandantenummer > 0 oder Fehlernummer <= 0:

- OFCKERR_uzl_Format: Für mind. einen der Pflichtparameter ist der NULL-Zeiger übergeben worden.

OFCK_HoleKurs

long WINAPI **OFCK_HoleKurs**(long IMandnr, LPSTR lpszAbDatum, LPSTR lpszWaehrung, long* pIDurchschnittskurs, LPSTR lpszKurs, long* pEinheitenanzahl, long* pIMengennotiz, BOOL bShowErr)
 Holt in einem Mandanten zu einem Datum den Umrechnungskurs einer Währung. Bezugskurs ist immer der EURO.

Die Änderung für den Durchschnittskurs ist kompatibel mit früheren Versionen, da ein Jahresdurchschnittskurs immer auch den Kurs eines Monats in diesem Jahr bestimmt.

Parameter:

long IMandnr	Die Mandantenummer.
LPSTR lpszAbDatum	Das Datum im Format TTMMJJ, TTMMJJJJ oder TT.MM.JJJJ.
LPSTR lpszWaehrung	Währung. max. 3 Stellen. Es wird zwischen Groß- und Kleinschreibung unterschieden.
long*	Rückgabe:
pIDurchschnittskurs	OFCK_KURS_TAG für einen tagesgenauen Kurs OFCK_KURS_MONAT für einen Monatsdurchschnittskurs OFCK_KURS_JAHR für einen Jahresdurchschnittskurs
LPSTR lpszKurs {13}	Rückgabe: Kurs (Zähler).
long* pEinheitenanzahl	Rückgabe: Kurs: Nenner. Mögliche Werte: 1, 10, 100, 1000, 10000.
long* pIMengennotiz	Rückgabe: Inverser Kurs (z.B.: 1 EUR = 6,65 FFR) falls ungleich 0, sonst 0.
BOOL bShowErr	Fehler anzeigen.

Rückgabewert:

long lErrno	- OFCK_OK - kein Fehler. - OFCKERR_uzl_Datum - kein korrektes Datum angegeben. - OFCKERR_kein_Kurs - kein Kurs zu den angegebenen Daten vorhanden. - sonst < 0.
-------------	--

OFCK_KursEinfuegen

long WINAPI **OFCK_KursEinfuegen**(long IMandnr, LPSTR lpszWaehrung, long IDurchschnittskurs, LPSTR lpszAbDatum, LPSTR lpszKurs, long lEinheitenanzahl, long lMengennotiz, BOOL bShowErr)

Fügt einen Kurs mit den angegebenen Daten ein. Der Bezugskurs ist immer der EURO.

Die Änderungen für den Durchschnittskurs betreffen einen neuen Wert zur Kennzeichnung eines Jahresdurchschnittskurses.

Parameter:

long IMandnr	Die Mandantenummer.
LPSTR lpszWaehrung	Währung. max. 3 Stellen. Es wird zwischen Groß- und Kleinschreibung unterschieden.
long	Folgende Werte sind gültig:
IDurchschnittskurs	OFCK_KURS_TAG für einen tagesgenauen Kurs OFCK_KURS_MONAT für einen Monatsdurchschnittskurs OFCK_KURS_JAHR für einen Jahresdurchschnittskurs
LPSTR lpszAbDatum	Das Datum im Format TTMMJJ, TTMMJJJJ, TT.MM.JJJJ.
LPSTR lpszKurs	Kurs: maximal 9 Vorkommastellen, maximal 2 Nachkommastellen (Zähler).
long lEinheitenanzahl	Kurs: Nenner. Mögliche Werte: 1, 10, 100, 1000.
long lMengennotiz	Inverser Kurs: Zähler und Nenner vertauschen (z.B.: 1 EUR = 6,65 FFR) falls ungleich 0, sonst 0.
BOOL bShowErr	Fehler anzeigen.

Rückgabewert:

long lErrno

- OFCK_OK - kein Fehler.
- OFCKERR_uzl_Datum - kein korrektes Datum angegeben.
- OFCKERR_kein_Kurs - kein Kurs zu den angegebenen (falschen) Daten möglich.
- OFCKERR_Kurs_vorhanden - Kurs ist bereits vorhanden
- OFCKERR_uzl_KursDurchschnitt - unzulässiger Parameter für IDurchschnittskurs
- sonst < 0.

OFCK_HoleSprache

long WINAPI **OFCK_HoleSprache**(HWND hParent, long lMandnr, LPSTR lpszName)

Besorgt den Sprach-Key zu einer Sprache.

Parameter

HWND hParent Handle des Fensters, unter dem der Dialog erscheinen soll.
long lMandnr Mandantenummer, zu dem eine Sprache geholt werden soll.
LPSTR lpszName {41} Zeiger auf den Namen der gewählten Sprache. Es sind mindestens 41 Byte für diesen Parameter zu reservieren. Ist der Name uninteressant, so kann auch ein NULL-Zeiger übergeben werden.

Rückgabewert:

long lErrno

- > 0: Sprach-Key.
- <= 0: Fehlernummer.

OFCK_Stapel

long WINAPI **OFCK_Stapel**(long lAktion, long lMandnr, long lStapelnr, LPCSTR szStapelname, long lWiederkehr, long lTransitorisch, long lAbstimm0, BOOL bShowErr)

Diese Funktion legt Stapel an oder ändert diese. Passwortgeschützte Stapel können nicht geändert werden.

Parameter

long lAktion OFCK_INSERT oder OFCK_UPDATE zum Einfügen oder ändern von Stapeln. Soll ein Stapel geändert werden, so muss dessen Stapelnummer im Parameter „lStapelnummer“ angegeben werden.
long lMandnr Mandantenummer, bei dem ein Stapel angelegt oder geändert werden soll
long lStapelnummer Nut bei Änderung eines Stapels: Die Stapelnummer des zu ändernden Stapels. Wird bei Anlage ignoriert.
LPCSTR szStapelname Der Name des Stapels.
long lWiederkehr Das Wiederkehrintervall in Monaten bei wiederkehrenden Stapeln. Ist es 0, so ist der Stapel nicht wiederkehrend. Gültige Werte: 1-99 (wiederkehrend), 0 (nicht wiederkehrend).
long lTransitorisch Legt einen transitorischen Stapel an (Wert <> 0), sonst 0.
long lAbstimm0 Setzt die Einstellung: „Nach Verbuchen Abstimmsumme auf 0,00 setzen“.
BOOL bShowErr Fehler anzeigen.

Rückgabewert:

long lErrno >0: Stapelnummer
 <=0: Fehlernummer

OFCK_LoescheStapel

long WINAPI OFCK_LoescheStapel(long lMandnr, long lStapelnr, long lMitStapel, BOOL bShowErr)

Mit dieser Funktion kann ein Stapel mit den zugehörigen Buchungen und ggf. Kostenbuchungen gelöscht werden. Es lassen sich nur Stapel löschen, die per OFCK angelegt wurden, nicht nach einem Clearinglauf gesperrt sind und die nicht durch ein Passwort geschützt sind.

Parameter

long lMandnr Der Mandant, für welchen der Stapel gelöscht werden soll.
long lStapelnr Die Stapelnummer des Stapels, der gelöscht werden soll.
long lMitStapel Gibt an, ob der Stapel selbst auch mit gelöscht werden soll.
 Ungleich 0: Stapelbuchungen und Stapel werden gelöscht
 Gleich 0: Nur Stapelbuchungen werden gelöscht.
BOOL bShowErr Fehler anzeigen.

Rückgabewert:

long lErrno OFCK_OK
 OFCKERR_uzl_Stapelnr: Stapel existiert nicht oder ist aus den o.g. Gründen nicht lösbar.

OFCK_PruefeRecht

long WINAPI OFCK_PruefeRecht(long lMandnr, LPSTR pszBenutzer, long* lRecht, long lUnused, BOOL bShowErr)

Dient zum Holen und dem darauffolgenden Prüfen der internen Rechtekennung eines Benutzers.

Parameter

long lMandnr Mandantenummer.
LPSTR pszBenutzer Name des Benutzers für welchen die Rechtekennung geholt werden soll.
long* lRecht Falls ungleich NULL: Pointer auf die interne Rechtekennung. Folgende Rechte können enthalten sein:
 OFCK_RECHT_ADMIN
 OFCK_RECHT_VERBUCHEN
 OFCK_RECHT_BUCHUNGENERFASSEN
 OFCK_RECHT_JAHRESWECHSEL
 OFCK_RECHT_KONTEN
 OFCK_RECHT_AUSWERT_SALDEN
 OFCK_RECHT_OPSMAHNEN
 OFCK_RECHT_STEUERSAETZE
 OFCK_RECHT_KRAUSWERTEN
 OFCK_RECHT_DATENEXPORT
 OFCK_RECHT_SICHERN
 OFCK_RECHT_HOLEN
 OFCK_RECHT_KONSOLIDIERUNG
 OFCK_RECHT_AUSWERT_STAMMDATEN
 OFCK_RECHT_ZV
 OFCK_RECHT_GRUNDEINSTELL


```

OFCK_RECHT_AUSWERT_BUCHUNGEN
OFCK_RECHT_AUSWERT_OP
OFCK_RECHT_AUSWERT_HIERARCHIE
OFCK_RECHT_AUSWERT_FINANZ
OFCK_RECHT_KONTOSALDO
OFCK_RECHT_KONTEN_ANSEHEN
OFCK_RECHT_AUSWERT_SPEICHERN
OFCK_RECHT_ALLE
OFCK_RECHT_ALLE_AUSWERTUNGEN

```

Die Rechtekennungen sind in der Datei ofck.h aufgeführt.

long lUnused Wird derzeit nicht verwendet.
 BOOL bShowErr TRUE: Fehlermeldungen werden als Dialog angezeigt.
Rückgabewert:
 long lErrno OFCK_OK = Die Rechtekennung wurde geholt; Fehlernummer <= 0

OFCK_GetUsedEngine

long WINAPI OFCK_GetUsedEngine()

Diese Funktion ermittelt, welche Datenbank verwendet wird.

Parameter:

keine

Rückgabewert:

long lErrno OFCK_ENG_BTRIEVE: Btrieve wird verwendet.
 OFCK_ENG_MYSQL: MySQL wird verwendet
 OFCK_ENG_MSSQL: MS SQL wird nicht verwendet.

OFCK_SQLInit

long WINAPI OFCK_SQLInit(long lEngine, BOOL bDialog, LPCSTR szHost, LPCSTR szUser, LPCSTR szPassword, long lPort, LPCSTR szZusatz, LPCSTR szDatabase)

Diese Funktion stellt eine Verbindung mit einer Datenbank her. Vor dem Aufruf dieser Funktion dürfen keine anderen OFCK-Funktionen aufgerufen werden, da sonst bereits ein Verbindungsaufbau mit den zuletzt verwendeten Einstellungen stattgefunden hat. Wenn die zuletzt verwendeten Einstellungen verwendet werden sollen muss die Funktion nicht aufgerufen werden.

Parameter

long lEngine Datenbanktyp (OFCK_ENG_BTRIEVE, OFCK_ENG_MYSQL oder OFCK_ENG_MSSQL)
 BOOL bDialog TRUE: Der Verbindungsdialog wird angezeigt mit den übergebenen Einstellungen.
 LPCSTR szHost Host/IP eines SQL-Servers (bei MSSQL ggf. gefolgt von einem Instanzname)
 LPCSTR szUser Benutzername für die Anmeldung an der Datenbank (bei MSSQL leer lassen bzw. NULL übergeben falls die Anmeldung über das Windows-Benutzerkonto erfolgen soll)
 LPCSTR szPassword Passwort für die Anmeldung an der Datenbank (bei MSSQL leer lassen bzw. NULL übergeben falls die Anmeldung über das Windows-Benutzerkonto

	erfolgen soll)
long lPort	Der Port welcher bei der MYSQL-Installation eingestellt wurde (Standard: 3306). Für MSSQL hat dieser Parameter keine Bedeutung.
LPCSTR szZusatz	Hier kann ein Zusatz für einen MSSQL-Verbindungsaufbau angegeben werden. Der Zusatz wird dem für die Verbindung generierten ODBC-Connectionstring angehängt. Bisher ist für dieses Feld keine Verwendung vorgesehen.
LPCSTR szDatabase	Der Name der Datenbank. (Standard: eurofibu)

Rückgabewert:

long lErrno	OFCK_OK: Die Verbindung wurde erfolgreich aufgebaut. OFCKERR_Abbruch: Der Dialog wurde abgebrochen. OFCKERR_Verbindungsfehler: Mit den Verbindungsdaten konnte keine Verbindung hergestellt werden.
-------------	---

OFCK_SQLDeInit

long WINAPI OFCK_SQLDeInit()

Diese Funktion beendet die Verbindung zur Datenbank.

Diese Funktion sollten Sie beim Beenden Ihrer Anbindung bzw. beim Entladen der OFCK-DLL immer aufrufen. Auch dann, wenn bei der Entwicklung keine SQL-Datenbank zur Anwendung kommt.

Parameter

keine

Rückgabewert:

long lErrno	OFCK_OK: Die Verbindung wurde getrennt.
-------------	---

OFCK_HoleMandantenVersion

long WINAPI OFCK_HoleMandantenVersion(long lMandnr)

Mit dieser Funktion lässt sich die Version der syska EURO FIBU ermitteln, mit welcher der angegebene Mandant zuletzt geöffnet (und ggf. konvertiert) wurde. In einer Datenbank der syska EURO FIBU können sich mehrere Mandanten mit unterschiedlicher Version befinden, sofern noch nicht alle Mandanten mit der neuen Version geöffnet wurden.

Eine Unterscheidung nach Zwischenversionen (z.B. 6.1a oder 6.1b) ist nicht möglich, da diese Angabe im Mandanten nicht mitgeführt wird - sie ist aber auch nicht notwendig, da in den Zwischenversionen die Datenbank seitens der syska EURO FIBU nicht konvertiert wird.

Parameter

long lMandnr	Die Mandantenummer des zu überprüfenden Mandanten.
--------------	--

Rückgabewert:

long lErrno	> 0: Version der Mandantendaten, z.B. 61, 62 für die Versionen 6.1 bzw. 6.2 OFCKERR_uzl_Mandnr: es wurde keine gültige Mandantenummer übergeben
-------------	--

OFCK_HoleFehlertext

long WINAPI OFCK_HoleFehlertext(long IFehlernummer, LPSTR pszKurz, LPSTR pszLang)

Mit dieser Funktion kann ein Fehlertext zu einer Fehlernummer in Erfahrung gebracht werden. Dabei gibt es einen Kurztext und einen Langtext. Die Texte sind für die Entwickler gedacht, nicht für die Endanwender.

Parameter

long IFehlernummer Die Fehlernummer, zu der ein Text geholt werden soll.
pszKurz {50} Kurztext. Dies ist das Symbol der Zeichenkette.
PszLang {250} Langtext. Hier sind z.T. Fehlerursachen aufgeführt.

Rückgabewert:

long IErrno OFCK_OK
 OFCKERR_keine_Fehlernummer: die angegebenen Nummer ist keine Fehlernummer.

OFCK_HoleStartJahr

long WINAPI OFCK_HoleStartJahr(long IMandnr)

Liefert das erste Jahr der Buchhaltung. Dies kann ein Archivjahr sein.

Parameter

long IMandnr Der Mandant, für welchen das Startjahr in Erfahrung gebracht werden soll.

Rückgabewert:

long IErrno > 0 : Startjahr
 <= 0: Fehlermeldung.

OFCK_IsProfiversion

long WINAPI OFCK_IsProfiversion()

Diese Funktion liefert zurück, ob die Plusversion oder die Standardversion verwendet wird

Parameter

keine

Rückgabewert:

long IErrno OFCK_OK: Plus-Version
 0: Standard-Version

OFCK_ErmittleBonitaet

long WINAPI OFCK_ErmittleBonitaet(HWND hParent, long lMandnr, long lKontoart, LPCSTR szKnr, long lSuchTyp, LPCSTR szName, LPCSTR szZusatz, LPCSTR szStr, LPCSTR szPLZ, LPCSTR szOrt, LPCSTR szVorname, LPCSTR szGeburtstag, LPCSTR szRegisternummer)

Mit dieser Funktion kann die Bonität eines in der EURO FIBU existierenden Personenkontos oder mittels Adressdaten einer Person oder Firma geprüft werden. Für ein existierendes Konto muss die Kontenart und die Kontonummer angegeben werden. I.d.F wird das Prüfergebnis auch im Personenkonto gespeichert.

Nach Aufruf der Funktion öffnet sich ein Dialog in welchem der Anwender unvollständige Daten ergänzen und die Ermittlung durchführen kann. Hierzu benötigt der Anwender einen Zugang zum per Webservice angebotenen Dienstleister bei welchem Kosten pro Abruf anfallen. Daher sollte innerhalb der anbindenden Anwendung ein Recht zur Bonitätsermittlung vergeben werden.

Parameter

HWND hParent	Fensterhandle als Parent für den anzuzeigenden Dialog
long lMandnr	Der Mandant, bei welchem auch die Zugangsdaten hinterlegt sind
long lKontoart	Kontoart des existierenden Kontos (OFCK_DEBI, OFCK_KRED), 0 sonst
LPCSTR szKnr	Kontonummer des existierenden Kontos, leer oder NULL sonst <i>Die nachfolgenden Daten werden nur benötigt, falls keine Kontonummer übergeben wird:</i>
long lSuchtyp	Suchtyp (OFCK_PERSON, OFCK_EINZELUNTERNEHMEN, OFCK_FIRMA)
LPCSTR szName	Nachname bei einer Person oder Einzelunternehmen. Firmenname bei einer Firma
LPCSTR szZusatz	Zusatz, wird nicht geprüft
LPCSTR szStr	Strasse und Hausnummer, optional aber empfohlen
LPCSTR szPLZ	Postleitzahl, optional aber empfohlen
LPCSTR szOrt	Ort, optional aber empfohlen
LPCSTR szVorname	Vorname, nur bei Personen und Einzelunternehmern, i.d.F. Pflicht (kann aber im Dialog ergänzt werden)
LPCSTR szGeburtstag	Geburtstag, nur bei Personen und Einzelunternehmern, TT.MM.JJJJ oder leer.
LPCSTR szRegisternummer	Registernummer, nur bei Firmen, i.d.F. Pflicht (kann aber im Dialog ergänzt werden)

Rückgabewert:

long lErrno	0: Bonität Unbestimmt, z.B. fehlende Angaben. 1: OFCK_BONITAET_GRUEN 2: OFCK_BONITAET_GELB 3: OFCK_BONITAET_ROT < 0: Fehlermeldung.
-------------	---

13. Protokoll

Grundsätzliches zum Protokoll

Für Entwicklungszwecke ist das OFCK mit einer Protokollfunktion ausgestattet. Ist sie aktiv, so werden alle OFCK-Funktionsaufrufe protokolliert und in eine Datei geschrieben.

Die Syntax der Protokolldatei ist dabei:

<Funktionsname>:

<Parameter>;[.....;]<letzter Parameter>.

In der ersten Zeile folgt auf den Funktionsnamen ein „.“. Die zweite Zeile beginnt mit einem TAB-Zeichen (Tabulator), gefolgt von der Parameterliste. Die einzelnen Parameter sind durch ein Semikolon getrennt, auf den letzten Parameter folgt ein Punkt. Für <Parameter> kann stehen:

- Der Wert des Parameters. Dies ist der Fall bei Übergabeparametern.

Bei Parametern, die Zeiger sind (also auch bei Verwendung von Text), kann außerdem folgendes stehen:

- „NULL“. Der Parameter ist NULL und dies ist ein möglicher (erlaubter) Wert ist.
- „NULL!“ (mit Ausrufezeichen) Der Parameter ist NULL, dies ist jedoch i. d. F. kein gültiger Wert.
- „nicht NULL“. Der Parameter ist kein Nullzeiger, der Inhalt ist jedoch (zu diesem Zeitpunkt) nicht interessant bzw. nicht bestimmbar. Dies tritt bei Rückgabewerten auf, deren Wert zum Protokollzeitpunkt noch nicht feststeht, (z.B. bei „OFCK_HoleMandantenNr“ der Parameter „IpszMandantenName“).

Ausnahmen:

- Bei Funktionen ohne Parameter steht statt der Parameterliste: „Funktion hat keine Parameter“.
- Die Funktionen „OFCK_ProtokollStart“ und OFCK_ProtokollEnde“ haben keine Parameter und geben statt dessen Systeminformationen (u.a. Aufrufzeit) aus.
- Bei der Funktion „OFCK_SplitBuchungEinfuegen“ folgen auf die obige Parameterliste noch die einzelnen (durchnummerierten) Teilbuchungen in je zwei eigenen Zeilen.
- Der Parameter „bShowErr“ wird als TRUE bzw. FALSE aufgelöst.

Bemerkungen:

- Generell wird unmittelbar nach dem Funktionsaufruf protokolliert, vor jeglicher Parameterprüfung.
- Da die OFCK-Funktionen z.T. auch weitere OFCK-Funktionen aufrufen, werden diese ggf. ebenfalls protokolliert. Dies tritt z.B. bei OFCK_Peko und OFCK_PekoDirekt oder OFCK_SplitStart und OFCK_SplitEnde sowie weiteren auf.
- Ist ein Text ohne Inhalt (Leerstring), so erscheint sofort der Parametertrenner (Semikolon), es können so u. U. mehrere Semikola aufeinander folgen.
- Strukturen werden unmittelbar in der Reihenfolge protokolliert, wie sie in der Datei „ofck.h“ stehen, Funktionsparameter in deren Reihenfolge beim Aufruf.

Hinweise:

- Die Protokollierung kann auch über den Eintrag „Protokoll=J“ in der Registry unter \Software\syska\settings\EFIBU\ gestartet werden, dabei wird im Programmverzeichnis der syska EURO FIBU die Datei OFCKPROT.TXT angelegt.
- Mit folgenden Einträgen in der Registry unter \Software\syska\settings\OFCK\ wird das Protokoll der jeweiligen Forward-DLL aktiviert. Die Protokolldatei heißt dabei wie angegeben.

Protokoll5051=<dateiname50.txt>
Protokoll5160=<dateiname51.txt>
Protokoll6061=<dateiname60.txt>
Protokoll6162=<dateiname61.txt>
Protokoll6270=<dateiname62.txt>
Protokoll7071=<dateiname70.txt>
Protokoll7172=<dateiname71.txt>
Protokoll7280=<dateiname72.txt>

Protokoll8090=<dateiname80.txt>
 Protokoll9091=<dateiname90.txt>
 Protokoll9192=<dateiname91.txt>
 Protokoll9293=<dateiname92.txt>
 Protokoll9394=<dateiname93.txt>
 Protokoll9495=<dateiname94.txt>
 Protokoll9596=<dateiname95.txt>
 Protokoll9697=<dateiname96.txt>
 Protokoll9798=<dateiname97.txt>

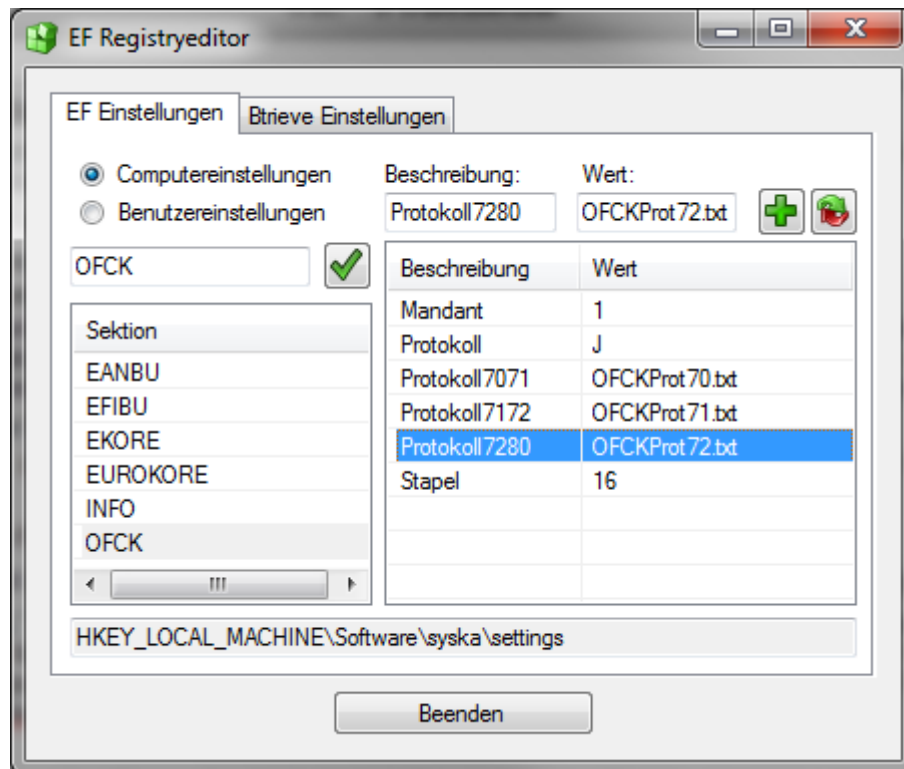


Abb. "EFRegEdit.exe mit Protokoll-Einträgen"

- Alle Programme, welche die Protokollfunktion benutzen, schreiben in die gleiche Protokolldatei, sowohl die syska EURO FIBU, das OFCK-Testprogramm als auch eigene Anwendungen.
- Der erste Schreiber fängt die Protokolldatei von vorne an, alle weiteren Einträge werden angehängt. Schreibt keiner mehr, so wird beim nächsten Start die Datei von vorne angefangen. Die Feststellung, welcher Schreiber der erste Schreiber ist, erfolgt anhand des Registry-Key HKEY_LOCAL_MACHINE\software\syska\OFCK, Schlüssel Protokolldatei. Dies ist ein DWORD-Wert, der flüchtig gespeichert wird (beim nächsten Neustart existiert dieser Schlüssel nicht, er wird jedes mal neu angelegt und bei System-Ende automatisch gelöscht). Der Zugriff auf diesen Schlüssel ist im OFCK über Mutual-Exclusion-Objekte abgesichert.

OFCK_Protokoll

long WINAPI **OFCK_Protokoll**(HWND hOwner, long lAktion, LPTSTR lpszDatei)

Dient zum Setzen bzw. Holen einer Protokolldatei. Die gewählte Datei wird in der Registry gespeichert, bzw. die Daten aus der Registry gelesen. Es wird weder die alte Datei geschlossen noch eine neue Datei geöffnet. Dies geschieht nur über OFCK_ProtokollStart() bzw. OFCK_ProtokollEnde().

Schlüssel: „HKEY_CURRENT_USER\Software\syska\OFCK“, Wert „Prokolldatei“

Parameter:

HWND hOwner	Handle des Fensters, unter dem der „Speichern unter“ Dialog erscheinen soll.
long lAktion	Aktion, folgende Werte sind möglich: <ul style="list-style-type: none">- OFCK_P_INTERAKTIV: legt die Protokolldatei über einen „Speicher unter...“-Dialog fest. Der Pfad der Datei muss existieren, kann jedoch auch in diesem Dialog angelegt werden. Ist der Parameter lpszDatei ungleich NULL, enthält er die gewählte Datei. Hierzu müssen für lpszDatei 260 (_MAX_PATH) Zeichen reserviert werden. Die Dateiendung ist immer „.txt“.- OFCK_P_SETZEN: Legt die Protokolldatei ohne Dialog fest. Es findet keine Prüfung des Dateipfades auf Existenz statt.- OFCK_P_HOLEN: Holt die eingestellte Protokolldatei.
LPTSTR lpszDatei	Enthält den zu schreibenden/zu lesenden Dateinamen (inkl. LW, Verzeichnis, Name und Extension). Wird der Dateiname gelesen, so müssen für diesen Rückgabewert 260 (_MAX_PATH) Zeichen reserviert werden.

Rückgabewert:

long lErrno	OFCK_OK falls eine Datei geschrieben oder gelesen werden konnte OFCKERR_Abbruch bei Dialog-Abbruch, oder OFCKERR_keine_Protokolldatei falls bei den Aktion OFCK_P_SETZEN und OFCK_P_HOLEN lpszDatei ein NULL-Pointer ist.
-------------	---

OFCK_ProtokollStart

long WINAPI **OFCK_ProtokollStart**()

Startet das Protokoll. Kann die Protokolldatei nicht erstellt werden (z.B. ungültiger Pfad, keine Schreibrechte, Datei wird von einem anderen Prozess benutzt, ...), wird nicht protokolliert. Das Verhalten kann aus dem Rückgabewert geschlossen werden.

Parameter:

keine Statt einer Parameterliste wird das Systemdatum und die Systemzeit ausgegeben.

Rückgabewert:

long lErrno	<ul style="list-style-type: none">- OFCK_OK: es wird protokolliert- OFCKERR_keine_Protokolldatei: es wird nicht protokolliert (Datei konnte nicht zum Schreiben geöffnet werden).- OFCKERR_bereits_Protokolldatei: es wurde bereits protokolliert
-------------	---

OFCK_ProtokollEnde

long WINAPI **OFCK_ProtokollEnde**()

Beendet das Protokoll. Diese Funktion kann auch aufgerufen werden, wenn nicht protokolliert wurde.

Parameter:

keine Statt einer Parameterliste wird das Systemdatum und die Systemzeit ausgegeben.

Rückgabewert:

long lErrno	<ul style="list-style-type: none">- OFCK_OK: es wurde protokolliert- OFCKERR_keine_Protokolldatei: es wurde nicht protokolliert (z.B. Datei konnte nicht zum Schreiben geöffnet werden).
-------------	---

14. Beispiele in C

Anlage eines Kunden (Debitors)

```
OFCK_PEKO pk;

if (Kunde im WWS)
{
    // Personenkonto-Struktur zusammenstellen:
    memset(&pk, 0, sizeof(OFCK_PEKO)); // mit \0 initialisieren
    pk.mandnr = 1;
    strcpy(pk.kontonr, "10000");
    strcpy(pk.kuerzel, "syska");
    strcpy(pk.name, "syska GmbH");
    strcpy(pk.zusatz, "Ges. für betriebliche DV");
    strcpy(pk.ustidnr, "DE143604909");
    strcpy(pk.land, "D");
    strcpy(pk.plz, "76149");
    strcpy(pk.ort, "Karlsruhe");
    strcpy(pk.strasse, "Am Sandfeld 9");
    strcpy(pk.blz, "66190000");
    strcpy(pk.bankknr, "55011303");
    strcpy(pk.bank, "VoBa Karlsruhe");
    strcpy(pk.gegenkonto, "8400");
    pk.mahnkennziffer = 0;
    pk.einmalkonto = OFCK_NEIN;
    // Konto anlegen:
    if (OFCK_Peko(OFCK_INSERT, OFCK_DEBI, &pk, TRUE) < 0)
        Fehler_behandeln();
}
```

Übergabe einer Ausgangsrechnung

```
OFCK_BUCHUNG bu;

if (Ausgangsrechnung erstellt)
{
    // Buchungs-Struktur zusammenstellen:
    memset(&bu, 0, sizeof(OFCK_BUCHUNG)); // mit \0 initialisieren
    bu.mandnr = 1;
    bu.stapelnr = 99; // Nummer aus Stapelauswahl ermitteln
    strcpy(bu.belegdatum, "160403");
    strcpy(bu.belegnr, "AR0815");
    strcpy(bu.text, "Software");
    strcpy(bu.sollkonto, "10000"); // Debitorenkonto
    bu.sollart = OFCK_DEBI;
    strcpy(bu.habenkonto, "8400"); // Erlöskonto im SKR-03
    bu.habenart = OFCK_SAKO;
    strcpy(bu.bruttobetrag, "116,00");
    strcpy(bu.waehrung, "EUR"); // Buchung in EURO
    // Buchung übergeben:
    if (OFCK_BuchungEinfuegen(&bu, TRUE) < 0)
        Fehler_behandeln();
}
```

Schrittweises Holen von Buchungsbelegen

```
char szBelegnr[9], szBelegdatum[11], szSollkonto[8], szHabenkonto[8], szBetrag[16],
    szBuchungsdatum[11];
long lSollArt, lHabenArt, lErg;

lErg = OFCK_InitHoleBube(1, "01.04.03", 0, 0, 0, 0);
if (lErg == OFCK_keinDatensatz)
    Kein_Datensatz_behandeln();
else
    {
        while (lErg == OFCK_weitererDatensatz)
            {
                lErg = OFCK_HoleBube(NULL, szBelegdatum, szBelegnr,
                    NULL, szSollkonto, &lSollArt, szHabenkonto,
                    &lHabenArt, OFCK_FIXLEN, szBetrag, NULL, NULL, NULL, NULL,
                    szBuchungsdatum, NULL, NULL, NULL, NULL, NULL, NULL);
                if (lErg > 0)
                    Behandle_Datensatz();
            }
        if (lErg != OFCK_letzterDatensatz)
            Fehler_behandeln();
    }
```

15. Hinweise und Beispiele für Visual Basic

Wichtige Hinweise für die Nutzer von Visual Basic

- Die C-Datei „OFCK.H“ enthält stets die aktuellen Deklarationen und ist die Referenz.
- Da sich unter Visual Basic kein Alignment für Strukturen einstellen lässt und die feste Voreinstellung nicht mit der Einstellung der OFCK-DLL verwendbar ist, muss auf die Verwendung von Strukturen zur Übergabe an das OFCK verzichtet werden. Dies stellt jedoch keine funktionale Einschränkung dar, da alle entsprechenden Aktionen auch über Funktionen ohne Strukturen ausgeführt werden können. Aus diesem Grund sind die entsprechenden Funktionen aus der Datei OFCK.BAS auskommentiert.
- Übergabeparameter können auch Rückgabewerte erhalten. Alle entsprechenden Parameter des OFCK sind unter VB mit ByRef deklariert. In dieser Dokumentation sind bei der Funktionsbeschreibung diese Parameter gesondert erwähnt.
- Texte werden unter Visual Basic und Visual C++ unterschiedlich gehandhabt. Bei der Übergabe von Texten kann auf die automatische Konversion von VB vertraut werden, jedoch nicht bei der Rückgabe von Texten. Hier ist es ratsam, Variablen zur Aufnahme von Texten den Erwartungen des C-Interfaces der OFCK-DLL entsprechend zu deklarieren und zu übergeben. Eine Variable zur Aufnahme des Mandantennamens würde also folgendermaßen deklariert werden: „Dim mandant(41) As Byte“. Bei der Verwendung dieser Variable wird dann ein Zeiger auf das erste Byte übergeben, also im Beispiel: „OFCK_HoleMandantenNr(0, mandant(0))“. Die Interpretation als Zeiger ist in der Deklaration der Funktion enthalten (ByRef).
- Um einen solchermaßen erhaltenen Text in VB lesen zu können, muss dieser in die in VB verwendete Darstellung ungewandelt werden. Hierzu verwenden Sie bitte die mitgelieferten Funktionen „ByteToBSTR“ (Byte-Array → VB-String) und „BSTRToByte“ (VB-String → Byte-Array) aus der Datei OFCK.BAS.
- Für reine Übergabewerte wird die Konvertierung automatisch durch VB durchgeführt. Es muss nur darauf geachtet werden, dass diese Variablen auch etwas enthalten, da ansonsten zu einem NULL-Zeiger konvertiert wird, welcher bei Übergabewerten i.d.R. nicht akzeptiert wird. Als sichere Lösung können Sie ein 0-Byte anhängen (& chr(0)). Dies ist insbesondere bei Edit-Feldern (TextBox) ratsam.

```
Dim konto As String
dim ergebnis as Long
ergebnis = OFCK_InitHoleOP( 1, konto & chr(0), OFCK_DEBI, 1 )
```

- NULL-Zeiger wie unter C oder C++ sind für die relevanten Parameter nur schwer zu realisieren. Als Alternative bietet sich bei Zeichenketten an, ein Byte-Array von der größten benötigten Größe anzulegen und dies bei allen uninteressanten Parametern zu verwenden. Für Long-Werte kann analog ebenfalls eine einzige Variable definiert werden.
- Datumswerte aus Date-Time-Pickern (DTPControl) sollten zuerst in einen VB-String konvertiert werden und dieser sollte übergeben werden, da die direkte automatische Konvertierung nicht richtig konvertiert.
- Gültige Werte für den Datentyp BOOL sind 0 (FALSE) und 1 (TRUE), bitte verwenden Sie OFCK_JA für TRUE und OFCK_NEIN für FALSE beim Aufruf der OFCK-Funktionen (und nicht true bzw. false, diese werden falsch konvertiert), vergleichen Sie hierzu bitte die C-Deklarationen.

Beispiele

Holen eines Mandanten:

```
Option Explicit
Dim Mandant As Long
Dim Fehler As Long

Private Sub HoleMandant_Click()
'HoleMandantenNr
Dim Name(41) As Byte

Mandant = OFCK_HoleMandantenNr(0, Name(0))
Fehler = Mandant

MandantNr = Mandant
MandantName = ByteToBSTR(Name, 41)

End Sub
```

Holen eines Stapels, hierbei muss in ‚Mandant‘ ein gültiger Wert sein (etwa mit vorigem Beispiel):

```
Option Explicit
Dim Mandant As Long
Dim Stapel As Long
Dim Fehler As Long

Private Sub HoleStapel_Click()
'Hole StapelNr
Dim Name(41) As Byte

Stapel = OFCK_HoleStapelNr(0, Mandant, Name(0))
Fehler = Stapel

StapelNr = Stapel
StapelName = ByteToBSTR(Name, 41)

End Sub
```

Holen eines Kontos, hierbei muss in ‚Mandant‘ ein gültiger Wert sein (etwa aus dem 1. Beispiel):

```
Option Explicit
Dim Mandant As Long
Dim Fehler As Long

Private Sub HoleDebitor_Click()
'Hole Debitor
Dim Kuerzel(9) As Byte
Dim Name(41) As Byte
Dim Debikonto(8) As Byte

Dim DebiNummer As String
Dim DebiKuerzel As String
Dim DebiName As String

Fehler = OFCK_HoleKonto(0, Mandant, OFCK_DEBI, "10000", "99999", Debikonto(0), _
Name(0), Kuerzel(0))

DebiNummer = ByteToBSTR(Debikonto, 8)
DebiKuerzel = ByteToBSTR(Kuerzel, 9)
DebiName = ByteToBSTR(Name, 41)

End Sub
```

Übergabe einer Ausgangsrechnung, hierbei muss in ‚Mandant‘ und in ‚Stapel‘ ein gültiger Wert sein (etwa aus den vorigen Beispielen):

```
Option Explicit
Dim Mandant As Long
Dim Stapel As Long
Dim Fehler As Long

Private Sub BuchungEinfuegen_Click()
'Buchung einfügen

Dim Belegdatum As String
Dim Belegnr As String
Dim Buchungstext As String
Dim DebiNummer As String
Dim SakoNummer As String
Dim Betrag As String
Dim Waehrung As String

Belegdatum = "01.01.2003"
Belegnr = "Test001"
Buchungstext = "Buchung über OFCK"
DebiNummer = "10000"
SakoNummer = "8400"
Betrag = "1160,00"
Waehrung = "EUR"

Fehler = OFCK_BuchungEinfuegenDirekt(1, Mandant, Stapel, Belegdatum, _
Belegnr, Buchungstext, DebiNummer, OFCK_DEBI, SakoNummer, OFCK_SAKO, _
Betrag, "", "", 0, 0, 0, 0, 0, 0, Belegdatum, Waehrung, 0, "", "")

If (Fehler = OFCK_OK) Then _
    Call MsgBox("Buchung eingefügt.", vbInformation) _
Else _
    Call MsgBox("Fehler!", vbCritical)

End Sub
```

Hierbei wird vorausgesetzt, dass der Debitor mit der Nummer „10000“ und das Erlöskonto „8400“ existieren sowie dass das Jahr 2003 angelegt und bebuchbar ist.

16. OFCKOUT

Übersicht

Die syska EURO FIBU bietet zusätzlich die Möglichkeit bei der Erfassung einer Buchung in der Finanzbuchhaltung, diese Buchung zeitnah an Ihre Anwendung zu übergeben. Dadurch bekommt Ihre Anwendung jede Buchung gleich mit, und kann z.B. die Bezahlung einer Rechnung gemäß den eigenen Bedürfnissen weiterverarbeiten. Entsprechendes gilt bei Veränderungen im Kontenstamm der syska EURO FIBU.

Um diese zeitnahe Form der Rückkopplung zu realisieren, benötigen Sie eine eigene DLL, welche die Funktionalität zur Übernahme der Daten aus der syska EURO FIBU bereitstellt. Dabei spielt es in der syska EURO FIBU keine Rolle, wie und ob die Daten durch Ihren eigenen Code verarbeitet werden. Die Veränderung im Datenbestand der syska EURO FIBU wird auf alle Fälle ausgeführt! Dieses Vorgehen hat im Vergleich zu den zuvor beschriebenen Hole-Funktionen den Vorteil, dass wirklich nur die geänderten bzw. neuen Daten betrachtet werden müssen.

Sie finden die gezippten Quelltexte einer Beispiel-DLL für MS-Visual C/C++ in der Datei SYSKAOUT.ZIP in Ihrem syska EURO FIBU - Verzeichnis. Diese DLL schreibt die übergebenen Daten in eine ASCII-Datei und kann von Ihnen gemäß Ihren eigenen Anforderungen modifiziert werden. Damit Ihre DLL vom OFCKOUT bzw. der syska EURO FIBU aufgerufen werden kann, müssen Sie diese bei der Installation Ihrer Anwendung im *Windows-Registry* eintragen. Dazu muss ein neuer Unterschlüssel zum Schlüssel **HKEY_LOCAL_MACHINE** angelegt werden, der folgenden Namen hat: **SOFTWARE\SyskaOFCK_Out\Application**, wobei *Application* ein Schlüssel sein sollte, der die hinter der DLL stehenden Anwendung eindeutig charakterisiert.

Bei einem 64-Bit System ist der Schlüssel unter:

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\syska\OFCK_Out\syskaout

Unter diesem Schlüssel müssen zwei Werte abgelegt werden: Unter dem Wertnamen **Application** ein String, der den Namen der Anwendung trägt, sowie unter dem Wertnamen **DLLPath** der vollständige Pfad der eigenen DLL (siehe Abbildung "Registrierungs-Editor").

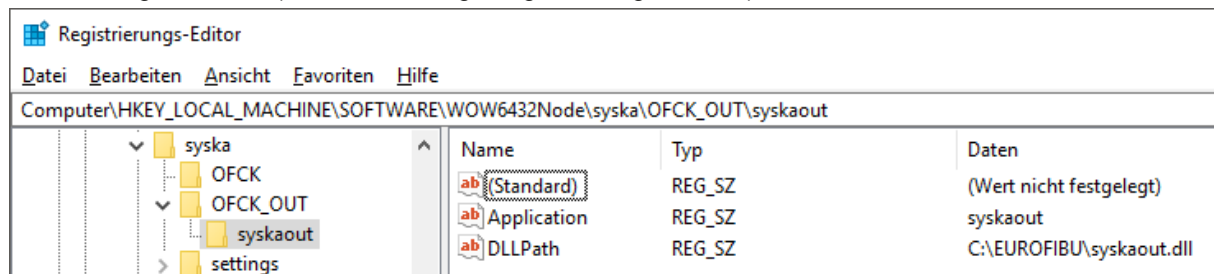


Abb. "Registrierungs-Editor"

Sollte Ihr Entwicklungssystem nicht die Möglichkeit bieten, eine DLL zu erstellen, so können Sie auch einfach die SYSKAOUT.DLL umbenennen und die dadurch erzeugte ASCII-Datei beim Start Ihrer Anwendung auswerten und anschließend löschen.

Das OFCKOUT selbst ist ein eigener Prozess, der von der syska EURO FIBU bereitgestellt und mit den zu übergebenden Daten aufgerufen wird. Nach dem Start der syska EURO FIBU wird dieser Prozess bei Bedarf automatisch gestartet. Daraufhin wertet dieser die Registry-Einträge aus und baut sich eine Liste der eingetragenen DLLs auf. Im laufenden Betrieb der syska EURO FIBU wird dann bei jeder Buchung und Veränderung im Kontenstamm die Funktion OFCK_OUT in jeder registrierten DLL aufgerufen. Die Auslagerung des OFCKOUT in einen eigenständigen Prozess hat den Vorteil, dass bei einem Absturz einer der registrierten DLLs nur dieser Prozess mit abstürzt und die Daten in der syska EURO FIBU nicht durch einen Absturz der Fibu selbst verloren gehen.

Um einen periodischen oder durch den Anwender manuell anzustoßenden Abgleich der Daten in der syska EURO FIBU und Ihrer Anwendung zu ermöglichen, sollten Sie auf alle Fälle zusätzlich die Hole-Funktionen des OFCK nutzen.

Die Funktion OFCK_OUT()

Ihre DLL, die vom OFCKOUT aufgerufen wird, muss lediglich die folgende Funktion enthalten:

extern "C" long **OFCK_OUT**(long IAktion, long IArt, LPVOID lpData)

Parameter:

long IAktion	durchgeführte Aktion (OFCK_INSERT, OFCK_UPDATE, OFCK_DELETE); bei Buchungen kann hier nur OFCK_INSERT erscheinen, da Buchungen laut Gesetzgeber nicht nachträglich verändert oder gelöscht werden dürfen
long IArt	Art der übergebenen Daten (OFCK_SAKO für Sachkonten, OFCK_KRED für Kreditoren, OFCK_DEBI für Debitoren, OFCK_BUBE für Buchungsbelege)
LPVOID lpData	Zeiger (bzw. Referenz) auf die Datenstruktur; diese ist abhängig von der Datenart und eine der aus dem OFCK bekannten Strukturen OFCK_PEKO, OFCK_SACHKONTO, OFCK_SPLITBUCHUNG (siehe vorne in diesem Dokument oder in der Datei OFCK.H)

Rückgabe:

long Der Rückgabewert wird nicht ausgewertet.

Wie bei den OFCK-Funktionen kann über die Mandantenummer in den Datenstrukturen der betroffene Mandant identifiziert werden. Wird hier eine unbekannte Nummer übergeben, so ist davon auszugehen, dass es sich hierbei um einen Testmandanten des Anwenders handelt. In diesem Fall sollten die Daten natürlich ignoriert werden.

Es wird grundsätzlich jede Buchung in der Splitbuchungsstruktur übergeben - demzufolge hat eine einfache Buchung eben die Splitbuchungsanzahl 1. Durchbuchungen (auf das Forderungs- und Verbindlichkeitenkonto) sowie bei Mandanten mit Ist-Versteuerung die Ist-Umbuchungen (von nicht fällige USt. auf fällige USt.) werden nicht übergeben.

Zur Erstellung der DLL ist es hilfreich, die Deklaration der OFCK_OUT-Funktion aus dem mitgelieferten Quelltext zur Beispiel-DLL SYSKAOUT.DLL zu übernehmen, welche die übergebenen Daten exemplarisch in die Datei "LOG_OFCK.TXT" protokolliert. Die Verarbeitung der Daten sollte möglichst schnell erfolgen oder rein notierender Art sein (z.B. Erstellung einer Protokolldatei, die jeden Aufruf erfasst und die bei Beginn der eigenen Anwendung ausgewertet wird) und nur wenige Datenbankaufrufe benutzen, da sonst aufgrund von zu langen Verarbeitungszeiten der Timeout-Mechanismus (dieser ist standardmäßig auf 3 Sekunden gesetzt) in der syska EURO FIBU zukünftige Übertragungen an die DLLs unterbindet. Dies berührt zwar das Verhalten der Fibu in keinsten Weise, allerdings ist dann die Übertragung aller während einer Fibu-Sitzung geänderten Daten nicht sichergestellt. Sollte es ihnen nicht möglich sein die Daten in dem voreingestellten Zeitraum von 3 Sekunden zu verarbeiten, können Sie den Timeout durch den Eintrag „OfckOutTimeout“ in der Registry unter \Software\syska\settings\OFCK\ verlängern. Die Angabe des Timeouts erfolgt in Millisekunden.

Hinweis: Wenn Sie eine DLL unter Delphi erstellen, so muss die OFCK_OUT Funktion in der Form *Function OFCK_OUT(IAktion ,IArt: LongInt; lpData: Pointer): LongInt; export; cdecl;* deklariert werden.



Gesellschaft für betriebliche Datenverarbeitung mbH

Am Sandfeld 15

76149 Karlsruhe

Tel.: +49 (0)721 985 93-0

Fax: +49 (0)721 985 93-60

info@syska.de

www.syska.de