

Electronic Banking in der Finanzbuchhaltung und im E-Commerce

Fortschrittliche Finanzverwaltung mit *HBCI4Java*



von **Stefan Palme, Egbert Schindler und Silke Karaus**

ANCOSO Development GmbH
Sommerfelderstrasse 10, 04299 Leipzig
Tel: 0341/ 861 25 54; <http://www.ancoso-development.de>
Ansprechpartner: Egbert Schindler, egbert.schindler@ancoso-development.de

Infostand: Juli 2004

Immer mehr und mehr Anwendungen setzen heute auf den multibankfähigen E-Banking-Standard HBCI (Homebanking Computer Interface). Durch die Softwareunabhängigkeit von HBCI lassen sich deshalb problemlos alle HBCI-Konten unterschiedlicher Banken mit ein und derselben Software verwalten. Die Anbieter sprechen von Multibankfähigkeit. Die Einsatzmöglichkeiten gehen dabei weit über das klassische E-Banking hinaus. HBCI bietet ungeahnte Möglichkeiten bei der Verwaltung von Geschäftsvorfällen.

HBCI4Java bietet Entwicklern von multibankfähigen Kundenprodukten, Herstellern von Kiosk-Systemen (für Versicherungen, Handel, Banken, Museen, Messen, Städte, Gemeinden, Hotel u.s.w.), Softwarehäusern und Rechenzentren HBCI-Softwarekomponenten für die Entwicklung von HBCI-Client- und -Server-Produkten an. Durch den Einsatz dieser Komponenten wird die Entwicklungszeit für die Nutzung des sicheren HBCI-Standards in Anwendungen erheblich verkürzt.

HBCI (HomeBanking Computer Interface) bietet Ihnen Vertraulichkeit und Sicherheit, die Sie erwarten.

INHALTSVERZEICHNIS

1	ÜBERSICHT.....	3
2	PRODUKTKONZEPT.....	4
2.1	Kurzbeschreibung HBCI4Java.....	4
2.2	Was kann ich alles mit HBCI4Java machen?	5
2.3	Was kostet HBCI4Java?	6
2.4	Unsere Referenz-Anwendungen mit HBCI4JAVA.....	7
3	WARUM HBCI?	7
3.1	Wo besteht der Unterschied zu anderen E-Banking-Verfahren?	7
3.2	Welchen Vorteil bietet mir HBCI?	8
3.3	Was benötige ich, um HBCI verwenden zu können?	8

1 Übersicht

In vielen Finanzbuchhaltungs-Anwendungen wird heute E-Banking (elektronischer Zahlungsverkehr) eingesetzt, um anstehende Aufträge wie ausgehende Zahlungen, Lastschriftaufträge usw. so aufzubereiten, dass diese Aufträge mit Hilfe bestimmter Softwarekomponenten auf elektronischem Weg an die jeweilige Bank übermittelt werden können. Außerdem wird von vielen Anwendungen das automatische Einlesen und Auswerten (Verbuchen) von Kontoauszügen unterstützt.

Aktuelle FiBu-Software erreicht im Bereich des electronic Banking einen der folgenden „Ausprägungsgrade“:

1. Es wird gar kein electronic Banking unterstützt.
2. „Electronic Banking“ in der Art, dass die Software entsprechend ausgefüllte Formulare (Überweisungsaufträge) ausdrucken kann, die anschließend auf herkömmlichem Wege bei der Bank eingereicht werden.
3. Die Software kann eine Datei erzeugen, welche Auftragsdaten in einem standardisierten Format (DTAUS) enthält. Der Anwender überträgt die so erzeugte Datei anschließend mit einer *speziellen* Software an die Bank (oder reicht diese Datei „manuell“ bei der Bank ein, z.B. auf einer Diskette).



Für eine Anwendung, mit der hauptsächlich Finanzen und Finanztransaktionen bearbeitet werden, ist diese Integrationstiefe nicht ausreichend! Wünschenswert wäre ein Verfahren, um Finanztransaktionen direkt realisieren zu können, ohne Umweg über ausgedruckte Formulare oder über zusätzliche Software für den Datenaustausch mit der Bank.

Für diesen Zweck steht **HBCI4Java** zur Verfügung, eine Programmbibliothek, die eine Anwendung um genau diese Funktionalität erweitert.

Mit **HBCI4Java** steht ein Werkzeug mit folgenden Eigenschaften zur Verfügung:

- Kommunikation mit der Bank direkt aus der Anwendung heraus, keine zusätzliche Software für den Datenaustausch mit der Bank mehr erforderlich. Damit stehen sämtliche Bankdaten in Echtzeit zur Verfügung, Aufträge können sofort und ohne Verzögerung ausgelöst werden
- Unterstützung vieler E-Banking-Funktionen, die über die herkömmlichen Funktionen hinausgehen, wie z.B. – Verwalten terminierter Aufträge
 - Dauerauftragsverwaltung
 - Unterstützung für Wertpapiergeschäfte
 - usw.
- Daneben werden natürlich alle gängigen Funktionen wie Einreichen von Zahlungs- und Lastschriftaufträgen, Abholen von Kontoumsatzdaten usw. unterstützt
- Voraussetzung ist lediglich ein HBCI-Zugang bei der eigenen Bank
- Lauffähig auf vielen verschiedenen Plattformen (Java)
- Integration in nahezu jede höhere Programmiersprache möglich

2 Produktkonzept

Entsprechend den differenzierten Anforderungen an HBCI-Software im dezentralen Umfeld werden Banking-Kernels und Server-Komponenten für unterschiedliche Einsatzszenarien bereitgestellt.

HBCI4Java-Clients

HBCI4Java setzt die Anforderungen an HBCI-Software auf Basis der technologisch zeitgemäßen Java-Plattform um. Dies bedeutet, dass die Vorteile dieser Plattform nun auch für HBCI Kundenprodukte verfügbar werden. Besonderer Wert wurde dabei auf einen hohen Abstraktionsgrad der Softwareschnittstellen gelegt. Damit können HBCI-Client-Anwendungen entwickelt werden, ohne dass weitgehende Kenntnisse des HBCI-Standards vorausgesetzt werden. Aufgrund der vielseitigen Möglichkeiten von **HBCI4Java** ist es möglich, neuartige innovative Kundenprodukte auf Basis von **HBCI4Java** zu pflegen und zu entwickeln.



HBCI4Java-Server

Neben den Softwareschnittstellen für die Entwicklung von HBCI-Client-Anwendungen werden auch Komponenten für die Erstellung eigener HBCI-Server-Produkte angeboten. Damit wird es erstmals möglich, unabhängig von Anbietern teurer Bankensoftware selbst HBCI-Server-Anwendungen zu erstellen. Damit haben vor allem kleine und mittelständische Unternehmen erstmals die Möglichkeit, unabhängige HBCI-Server-Software verwenden und entwickeln zu können, ohne auf die wenigen Monopolisten angewiesen zu sein.

2.1 Kurzbeschreibung HBCI4Java



HBCI steht für Homebanking Computer Interface und beschreibt ein Internet-Protokoll, welches die Kommunikation eines Bankkunden mit seiner Bank über das Internet ermöglicht. Über diese Schnittstelle kann der Bankkunde Geschäftsvorfälle auslösen, (Konto-)Informationen abrufen usw. Die meisten Banken in Deutschland stellen diese Schnittstelle nach außen zur Verfügung.

Mit **HBCI4Java** steht eine Klassenbibliothek zur Verfügung, die von beliebigen Java-Anwendungen verwendet werden kann, um das HBCI-Protokoll zu benutzen. Damit wird die Entwicklung von Applikationen wie Homebanking-/Buchhaltungs-Software, automatisierte Abrechnung (z.B. in Händlersystemen, Brokerage usw.) auf Basis von HBCI möglich.

Im Folgenden werden einige Eigenschaften von **HBCI4Java** aufgeführt, die in dieser Kombination nur von **HBCI4Java** geboten werden:

- Implementation vollständig in Java – Damit wird eine 100%ige Verfügbarkeit für alle Plattformen garantiert, für die es eine entsprechende Java-Version gibt.
- Der Quelltext zu den meisten Teilen von **HBCI4Java** ist öffentlich verfügbar.
- **HBCI4Java** unterstützt derzeit alle gängigen HBCI-Versionen (2.01, 2.1 und 2.2) mit den darin vorgegebenen Sicherheitsmechanismen. Zusätzlich wird die HBCI-PIN/TAN-Erweiterung unterstützt, die es erlaubt, das bekannte PIN/TAN-Verfahren für die Authentifikation zusammen mit dem HBCI-Protokoll zu verwenden.

- Modulare Architektur – dadurch ist es leicht möglich, zukünftige Versionen des HBCI-Protokolles, neue Sicherheitsmechanismen zur Authentifizierung/ Verschlüsselung, bankspezifische Anpassungen usw. sehr schnell in **HBCI4Java** zu integrieren.
- Anwendungsorientierte Programmierschnittstelle – Anwendungsentwickler, die **HBCI4Java** verwenden, benötigen keinerlei Kenntnisse des HBCI-Protokolles, sondern können vielmehr direkt typische Geschäftsvorfälle sehr einfach erzeugen und auswerten. **HBCI4Java** übernimmt dabei absolut transparent sämtliche Aufgaben, die nötig sind, um die gewünschten Geschäftsvorfälle erfolgreich bei der jeweiligen Bank einzureichen und um die Ergebnisse auszuwerten.
- Dynamische Anpassungen an bankspezifische Besonderheiten möglich – Module, die Anpassungen an das Verhalten einiger spezieller HBCI-Server realisieren, können zur Laufzeit zu- oder abgeschaltet werden.
- Einsatz in beliebigen Umgebungen möglich – **HBCI4Java** lässt sich nicht nur in "normalen" Java-Anwendungen verwenden, sondern wurde auch bereits in Applets, in Servlets (auf einem Application Server), als "PaymentProvider-Service", in Lasttest-Tools für HBCI-Server und sogar als eigener HBCI-Server eingesetzt.
- Benutzung von Daten "fremder" Software – die modulare Architektur von **HBCI4Java** ermöglicht es, Schlüsseldateien anderer Softwareprodukte verwenden zu können (In den Schlüsseldateien werden die personenbezogenen Daten und Sicherheitsmerkmale eines Bankkunden für seinen HBCI-Zugang abgelegt. Das Format dieser Schlüsseldateien ist nicht vorgeschrieben, so dass jede HBCI-fähige Software hier u.U. ein eigenes Format benutzt). Zurzeit werden die Schlüsseldateien im SIZ-Format (verwendet von StarMoney, GENOlite, u.a.) sowie Schlüsseldateien von OpenHBCI (freie HBCI-Client-Implementation in C++, wird z.B. in GnuCash verwendet) unterstützt.
- Es ist ein Framework für das Aufsetzen eines eigenen HBCI-Servers vorhanden, der ebenfalls die **HBCI4Java**-Schnittstellen benutzt und demzufolge als Java-Anwendung realisiert ist.
- Es werden Tools und Demo-Anwendungen bereitgestellt, welche die Arbeit mit **HBCI4Java** demonstrieren und die als Beispielapplikationen bzw. Skelette für eigene Applikationen verwendet werden können.
- Es existieren Wrapper-Bibliotheken, die den Einsatz von **HBCI4Java** auch aus anderen Programmiersprachen heraus zu ermöglichen (z.B. Delphi).

2.2 Was kann ich alles mit HBCI4Java machen?

- Es stehen Softwarekomponenten für die Entwicklung von sowohl HBCI-Client- wie auch HBCI-Server-Anwendungen zur Verfügung
- Unterstützung von dateibasiertem RDH-Verfahren, PIN/TAN-Verfahren und chipkartenbasierten DDV-Verfahren, dabei Unterstützung von Kartentypen 0 und 1, Chipkartenleser Klassen 1-3 (benötigt CTAPI-Treiber) sowie Biometrieinheit (nur Reiner-SCT)
- Unterstützung von SIZ-RDH-Schlüsseldateien (StarMoney, GENOlite) mit Hilfe einer zusätzlichen Laufzeitbibliothek
- direkte Unterstützung von OpenHBCI-Schlüsseldateien für RDH-Zugänge
- Derzeit Unterstützung der folgenden HBCI-Geschäftsvorfälle:



- Einzelüberweisungen
 - Auslandsüberweisungen
 - Sammelüberweisungen, Sammellastschriften (jeweils HBCI4Java-eigener DTAUS-Generator vorhanden)
 - Verwalten von terminierten Überweisungen
 - Verwalten von Daueraufträgen
 - Einreichen von Lastschriften
 - Verwalten von Festgeldanlagen
 - Abfrage von Kontoinformationen
 - Saldenabfragen
 - Abholen von Umsatzinformationen (Kontoauszug)
 - Abholen von Kontostammdaten
 - Abholen von Informationen zu ausgegebenen Karten
 - Abholen und Senden verschiedener Informationen und Nachrichten
 - Auswertung des bankinternen Statusprotokolls
 - Anzeigen von Informationen zu TAN-Listen (nur bei PIN/TAN-Verfahren)
 - Abholen von Wertpapierdepot-Informationen
 - es werden u.U. einige weitere Geschäftsvorfälle unterstützt, die in dieser Aufzählung noch nicht aufgeführt sind.
- Zuordnung von Bankleitzahlen zu Kreditinstituts-Bezeichnungen
 - diverse Tools zum Auslesen/Bearbeiten der DDV-Chipkarte, zum Erzeugen eines INI-Briefes, zum Bearbeiten der Passports (Sicherheitsmedien), usw.
 - Batch-Tool für automatisierte Abarbeitung von HBCI-Jobs, die in einer Textdatei definiert werden können
 - Unterstützung aller Nachrichten für die Schlüsselverwaltung
 - verfolgen der HBCI-Kernel-Aktivitäten durch Callbacks möglich
 - volle Kontrolle über den HBCI-Kernel durch so genannte Kernel-Parameter, die von einer Anwendung beliebig manipuliert werden können
 - dynamische Anpassungen an HBCI-Server mit "Fehlern" in der Implementation des HBCI-Standards
 - Callback-Mechanismus für die Kommunikation zwischen HBCI-Kernel und Anwendung, um unabhängig von der Laufzeitumgebung (Applikation, Servlet, Applet, mit oder ohne GUI usw.) zu sein
 - Einsatz in multithreaded Anwendungen möglich



2.3 Was kostet HBCI4Java?

Alle **HBCI4Java**-Komponenten (Client- und Server-Suite) sind unter der GPL¹ lizenziert und stehen damit kostenlos für Entwickler von GPL-Software zur Verfügung. Für die Verwendung von **HBCI4Java** in Anwendungen, die nicht unter die GPL gestellt werden (closed source, kommerzieller Vertrieb, etc.), kann eine entsprechende Lizenz individuell erworben werden.

¹ <http://opensource.org/licenses/gpl-license.php>

2.4 Auszug aus unserer Referenz-Liste mit HBCI4JAVA



- Studienprojekt für automatische Massen-Überweisungen mit ausgewählten Daten aus externen Datenbankquellen
- **OpenSource-Projekt Hibiscus** - Integration von **HBCI4Java** in OpenSource HBCI OnlineBanking Client
- **DM – „Digital Money“** - Integration von **HBCI4Java** als Bezahl-Schnittstelle in ein Micropayment-System
- **RST Datentechnik GmbH** - Integration von **HBCI4Java** als Bezahlssystem in bestehende komplexe Payment-Infrastruktur.
- **GAD eG Münster** - Einsatz des **HBCI4Java Demo Servers** als HBCI Server für interne Lasttests
- **BIVG Hannover** - Einsatz des **HBCI4Java Demo Servers** als HBCI Server für Tests eigener HBCI-Client-Applikationen
- **Hard- und Software Consulting GmbH** - Integration von **HBCI4Java** und dem **HBCI4Java Delphi Wrapper** in existierende Software für die Finanzbuchhaltung
- **eVerlage Projekt**- Integration von **HBCI4Java** als Bezahl- und Abrechnungssystem in die digitale Bibliothek *eVerlage*.

Weitere Informationen finden Sie auf unserer Webseite.

3 Warum HBCI?

3.1 Wo besteht der Unterschied zu anderen E-Banking-Verfahren?

HBCI ist ein Standard mit folgenden Vorteilen:

HBCI ist ein Banking-Protokoll, welches von vielen Banken unterstützt wird. Damit lässt sich ein und dasselbe "E-Banking-Programm" (wenn es HBCI kann) mit allen diesen Banken benutzen. Für den Anwender, der nur Homebanking machen will, ist das nicht so kritisch, da er nur selten seine Bank wechselt und auch selten Konten bei mehreren verschiedenen Banken hat. Im Industrie- und Wirtschaftsbereich sieht das schon anders aus. Wollen Hersteller von (z.B.) Finanzbuchhaltungssoftware eine Bankenschnittstelle direkt in ihre Software einbauen, so muss dafür natürlich ein Standard her (HBCI). Gäbe es diesen Standard nicht, so müsste für jede Bank, die evtl. mit dieser Software benutzt werden soll, ein separates Modul eingebaut werden, welches die Banking-Schnittstelle dieser Bank beherrscht. Dank



HBCI muss nur eine einzige solche Schnittstelle implementiert werden. Dann kann diese Software mit jeder Bank, die HBCI kann, benutzt werden, ohne dass diese Software jemals speziell an eine bestimmte Bank angepasst werden muss.

3.2 Welchen Vorteil bietet mir HBCI?

HBCI ist ein reines Netzwerkprotokoll.



Viele heutige "Banking-Schnittstellen" sind auf die speziellen Anwendungen "Homebanking" und "Zugriff via Web" zugeschnitten worden. Das hat meist zur Folge, dass diese "Banking-Schnittstellen" nicht oder nur schlecht in andere Software zu integrieren sind (z.B. eine Software, die automatisch in einem bestimmten Intervall Kontoauszüge abholt und irgendwie auswertet). Bei "Banking-Schnittstellen", die nur auf Homebanking zugeschnitten sind, ist das meist schwierig. Vor allem die Integration in Servlets, nicht-interaktive Anwendungen usw. ist oft mit hohem Aufwand verbunden.

Da es sich bei dem HBCI-Protokoll "nur" um die Beschreibung von Nachrichten und deren mögliche Abfolge beim Austausch zwischen Kunde und Bank handelt, ist eine Integration in die verschiedensten Systeme und Anwendungen möglich.

3.3 Was benötige ich, um HBCI verwenden zu können?

Voraussetzung ist, dass die eigene Bank HBCI-Zugänge anbietet. Eine Liste aller entsprechenden Banken ist im Internet unter <http://www.hbci.de> zu finden. Im Zweifelsfall telefonische Nachfrage bei der Bank.

Sie benötigen HBCI-fähige Software, um den HBCI-Zugang Ihrer Bank nutzen zu können. Jede auf HBCI4Java basierende Software sollte dazu in der Lage sein. Wenn Sie auf spezielle Software oder Software-Komponenten angewiesen sind, helfen wir Ihnen gern weiter.

Abhängig von dem von der Bank unterstützten HBCI-Sicherheitsverfahren wird u.U. ein Chipkartenleser benötigt. Den erhalten Sie im besten Fall von Ihrer Bank zusammen mit Ihren HBCI-Zugangsdaten. Wenn nicht, können Sie einen Leser auch von uns bekommen.

Sprechen Sie uns an, wir helfen Ihnen.

ANCOSO Development GmbH

Sommerfelderstrasse 10, 04299 Leipzig

Tel: 0341/ 861 25 54; <http://www.ancoso-development.de>

Ansprechpartner: Egbert Schindler, egbert.schindler@ancoso-development.de