

## AqBanking - Bug #60

### aqbanking + PSD2 + gnucash = geht nicht

20.09.2019 14:53 - Anonym

|                           |           |                               |            |
|---------------------------|-----------|-------------------------------|------------|
| <b>Status:</b>            | Closed    | <b>Beginn:</b>                | 20.09.2019 |
| <b>Priorität:</b>         | Normal    | <b>Abgabedatum:</b>           |            |
| <b>Kategorie:</b>         | AqBanking |                               |            |
| <b>Betriebssystem:</b>    | Linux     | <b>Anwendung:</b>             | GnuCash    |
| <b>AqBanking-Version:</b> | 5.99.36.0 | <b>Version der Anwendung:</b> |            |

**Beschreibung**

1. Ich habe gnucash und aqbanking kompiliert mit Hilfe des Makefiles (siehe ganz unten).
2. Ich die Anleitung <https://www.aquamaniac.de/rdm/projects/aqbanking/wiki/SetupPinTan> befolgt und konnte Kontostand und Umsätze abrufen (auf der Konsole, Bank DKB, <https://banking-dkb.s-fints-pt-dkb.de/fints30>). Ich verwende chipTAN manuell, werden zur TAN-Eingabe aufgefordert (Startcode eingeben, TAN erzeugen), gebe Tan ein und es funktioniert.
3. Ich öffne gnucash (make run) und sehe in den Onlinebankingsettings die auf der Konsole eingerichteten Konten.
4. Ich probiere Transaktionen aus gnucash heraus abzurufen und erhalte ich **ERROR (11)**

Aktivierung des Loggings nach <https://www.aquamaniac.de/rdm/projects/aqbanking/wiki> erzeugt jedoch keinerlei Ausgabe auf der Konsole.

Eine Einrichtung des Onlinebankings aus gnucash heraus ist nicht möglich! Man kommt irgendwann nicht weiter. Es endet mit folgender Meldung. Auf der Konsole hingegen funktioniert es.

```
14:49:49 Retrieving SSL certificate
14:49:49 Preparing connection
14:49:49 Connecting to server...
14:49:49 Using GnuTLS default ciphers.
14:49:49 TLS: SSL-Ciphers negotiated: TLS1.2:ECDHE-RSA-AES-128-GCM:AEAD
14:49:49 Connected.
14:49:49 Disconnected.
14:49:49 Got certificate
14:49:49 Retrieving system id
14:49:49 Executing HBCI jobs
14:49:49 AqHBCI started
14:49:49 Encoding queue
14:50:14 Sending message
14:50:14 Using GnuTLS default ciphers.
14:50:14 TLS: SSL-Ciphers negotiated: TLS1.2:ECDHE-RSA-AES-128-GCM:AEAD
14:50:14 Message sent
14:50:14 Queue sent
14:50:14 Waiting for response
14:50:15 Response received
14:50:15 HBCI: 9050 - Die Nachricht enthält Fehler. (M)
14:50:15 HBCI: 9800 - Dialog abgebrochen (M)
14:50:15 HBCI: 9010 - Initialisierung fehlgeschlagen, Auftrag nicht bearbeitet. (M)
14:50:15 HBCI: 3079 - Bitte wenden Sie sich an den Hersteller des Banking-Programms. (S)
14:50:15 HBCI: 9078 - Dialog abgebrochen - Banking-Programm ist nicht registriert. (S)
14:50:15 HBCI: 9010 - Der Auftrag wurde nicht ausgeführt. (S)
14:50:15 Dialog aborted by bank, assuming bad PIN
14:50:15 Dialog aborted by server
14:50:15 Error processing job JobSync
14:50:15 AqHBCI finished.
14:50:15 Operation finished, you can now close this window.
```

---

```
AQBANKING_TAR=aqbanking-5.99.30beta.tar.gz
AQBANKING_MD5=1806a9d73cc2d9c6878dbec032e59ff2
AQBANKING_DIR=$(AQBANKING_TAR:.tar.gz=)
GWENHYWFAR_TAR=gwenhywfar-4.99.15beta.tar.gz
GWENHYWFAR_MD5=ee5317a6b583458ed9a9451b4f07ef13
GWENHYWFAR_DIR=$(GWENHYWFAR_TAR:.tar.gz=)
```

```

GNUCASH_TAR=gnucash-3.7.tar.gz
GNUCASH_MD5=ff6ab2ac4e5364ee6a52d593c6dfb496
GNUCASH_DIR=$(GNUCASH_TAR:.tar.gz=)
PREFIX=$(shell pwd)
SHELL=bash

all: gnucash

dependencies:
    sudo apt-get install -y build-essential libgcrypt20-dev pkg-config libgnutls28-dev libxmlsec1-
dev libwebkit2gtk-4.0-dev guile-2.2-dev libofx-dev libdbi-dev libdbi1 libdbd-sqlite3 libboost-date
-time-dev libboost-regex-dev libboost-locale-dev libboost-filesystem-dev libboost-system-dev libse
cret-1-dev libgtest-dev libgmock-dev libgtk-3-dev cmake xsftproc

$(GWENHYWFAR_TAR):
    wget https://www.aquamaniac.de/rdm/attachments/download/136/$(GWENHYWFAR_TAR)
    md5sum -b $(GWENHYWFAR_TAR) |grep $(GWENHYWFAR_MD5)

$(GWENHYWFAR_DIR)/configure: $(GWENHYWFAR_TAR)
    tar xf $(GWENHYWFAR_TAR)
    touch $(GWENHYWFAR_DIR)/configure

$(GWENHYWFAR_DIR)/config.status: $(GWENHYWFAR_DIR)/configure
    cd $(GWENHYWFAR_DIR) && ./configure --prefix=$(PREFIX) --with-guis=gtk3

lib/libgwenhywfar.so: $(GWENHYWFAR_DIR)/config.status
    $(MAKE) -C $(GWENHYWFAR_DIR)
    $(MAKE) -C $(GWENHYWFAR_DIR) install

gwenhywfar: lib/libgwenhywfar.so

$(AQBANKING_TAR):
    wget https://www.aquamaniac.de/rdm/attachments/download/134/$(AQBANKING_TAR)
    md5sum -b $(AQBANKING_TAR) |grep $(AQBANKING_MD5)

$(AQBANKING_DIR)/configure: $(AQBANKING_TAR)
    tar xf $(AQBANKING_TAR)
    touch $(AQBANKING_DIR)/configure

$(AQBANKING_DIR)/config.status: $(AQBANKING_DIR)/configure lib/libgwenhywfar.so
    cd $(AQBANKING_DIR) && ./configure --prefix=$(PREFIX)

lib/libaqbanking.so: $(AQBANKING_DIR)/config.status
    $(MAKE) -C $(AQBANKING_DIR)
    $(MAKE) -C $(AQBANKING_DIR) install

aqbanking: lib/libaqbanking.so

$(GNUCASH_TAR):
    wget https://github.com/Gnucash/gnucash/releases/download/3.7/$(GNUCASH_TAR)
    md5sum -b $(GNUCASH_TAR) |grep $(GNUCASH_MD5)

$(GNUCASH_DIR)/CMakeLists.txt: $(GNUCASH_TAR)
    tar xf $(GNUCASH_TAR)
    touch $(GNUCASH_DIR)/CMakeLists.txt

bin/gnucash: $(GNUCASH_DIR)/CMakeLists.txt lib/libaqbanking.so
    mkdir -p $(GNUCASH_DIR)/build
    cmake -S $(GNUCASH_DIR) -B $(GNUCASH_DIR)/build -D CMAKE_INSTALL_PREFIX=$(PREFIX) -D CMAKE_PRE
FIX_PATH=$(PREFIX)
    $(MAKE) -C $(GNUCASH_DIR)/build
    $(MAKE) -C $(GNUCASH_DIR)/build install

gnucash: bin/gnucash

# https://www.aquamaniac.de/rdm/projects/aqbanking/wiki/SetupPinTan

```

```

HBCITOOl=LD_LIBRARY_PATH=$(PREFIX)/lib bin/aqhbci-tool4
AQBANKING=LD_LIBRARY_PATH=$(PREFIX)/lib bin/aqbanking-cli
URL=https://banking-dkb.s-fints-pt-dkb.de/fints30
UID=1
FROM=20190601

adduser:
    $(HBCITOOl) adduser -t pintan --context=1 -b $(BLZ) -u $(USER) -s "$(URL)" -N "$(NAME)" --hbc
version=300
    $(HBCITOOl) listusers

settlsflag:
    $(HBCITOOl) adduserflags -u $(UID) -f tlsIgnPrematureClose

getdata:
    $(HBCITOOl) getbankinfo -u $(UID)
    $(HBCITOOl) getsysid -u $(UID)
    $(HBCITOOl) listitanmodes -u $(UID)

settanmode:
    $(HBCITOOl) setitanmode -u $(UID) -m $(MODE)

settanmediumid:
    $(HBCITOOl) setTanMediumId -u $(UID) -m "$(MID)"

getaccounts:
    $(HBCITOOl) getaccounts -u $(UID)
    $(HBCITOOl) listaccounts -v

getsepainfo:
    $(HBCITOOl) getaccsepa -a $(AID)

gettransactions:
    $(AQBANKING) request --account=$(AID) --fromdate=$(FROM) --transactions

getbalance :
    $(AQBANKING) request --account=$(AID) --balance

getversions:
    $(AQBANKING) versions

run:
    LD_LIBRARY_PATH=$(PREFIX)/lib bin/gnucash

.ONESHELL:
AppRun:
    cat <<- EOF > $@
    #!/bin/sh
    HERE="\${dirname "$(readlink -f "${0}")}"
    export PATH="\${HERE}/bin:\${PATH}"
    export LD_LIBRARY_PATH="\${HERE}/lib:\${LD_LIBRARY_PATH}"
    export XDG_DATA_DIRS="\${HERE}/share:\${XDG_DATA_DIRS}"
    EXEC=$(grep -e '^Exec=.*' "${HERE}/*.desktop" | head -n 1 | cut -d "=" -f 2 | cut -d "
" -f 1)
    exec "\${EXEC}" "\${$@}"
    EOF
    chmod +x $@

gnucash-icon.png:
    convert share/gnucash/pixmaps/gnucash-icon.ico[4] $@

gnucash.desktop:
    cp share/applications/$@ .

CP=cp -vd
LIB=lib/
LIBSRC=/usr/lib/x86_64-linux-gnu

```

```
.PHONY: lib
lib:
    mkdir -p $@
    $(CP) /lib/x86_64-linux-gnu/libgcrypt.so.* $@
    ln -srv $@/libgcrypt.so.* $@/libgcrypt.so
    $(CP) $(LIBSRC)/libgnutls*.so* $@
    $(CP) $(LIBSRC)/libgmp.so* $@
    $(CP) $(LIBSRC)/libgtk-3.so* $@
    $(CP) $(LIBSRC)/libguile*.so* $@
    $(CP) $(LIBSRC)/libxml2.so* $@
    $(CP) $(LIBSRC)/libgettext*.so* $@
    $(CP) $(LIBSRC)/libxslt.so* $@
    $(CP) $(LIBSRC)/libicu*.so* $@
    $(CP) $(LIBSRC)/libboost*.so* $@
    $(CP) $(LIBSRC)/libwebkit2gtk-4.0.so* $@
    $(CP) $(LIBSRC)/libdbi.so* $@
    $(CP) $(LIBSRC)/dbd/libdbdsqlite3.so* $@
    $(CP) $(LIBSRC)/libsqlite3.so* $@
    $(CP) $(LIBSRC)/libgc.so* $@
    ln -srvf $@/gnucash/*.so $@

.PHONY: share
share:
    mkdir -p $@
    cp -vdr /usr/share/guile $@

.PHONY: bin
bin:
    mkdir -p $@
    cp -v `which guile` $@

gnucash.appimage: AppRun gnucash.desktop gnucash-icon.png bin etc lib share
    rm -rf appimage/
    mkdir appimage/
    cp -r $^ appimage/
    appimagetool -v $(OPTS) appimage $@
    rm -rf appimage/

appimage: gnucash.appimage

clean:
    rm -rf aqbanking* gwenhywfar* gnucash* bin etc lib share include appimage AppRun
```

## Historie

### #1 - 20.09.2019 18:12 - cstimming

Hinweis: Die Fehlermeldung "HBCI: 9078 - Dialog abgebrochen - Banking-Programm ist nicht registriert. (S)" ist ein Fehler in gnucash in der releasen Version 3.7. Abhilfe momentan: Die git-Version ("maint", nicht "master") benutzen, bzw. diesen patch <https://github.com/Gnucash/gnucash/commit/ccaaa14187d5aa37e8ed24669fbba6ad055f9df7> ergänzen. Oder auch von hier das zip von "maint" herunterladen: <https://github.com/Gnucash/gnucash>

### #2 - 21.09.2019 10:42 - Anonym

Kompilierung des entpackten ZIPs funktioniert nicht, man muss schon git clone machen.

Der Branch maint (496ca94a989d0001a64c19f5e2104ae02fe1a604, enthält ccaaa1418) liefert bei Abruf der Transaktionen **rejected (5)**.

Nachdem ich in GnuCash im Onlinebankingsetup den Benutzer editiert habe und get bankinfo, systemid, usw. abgerufen habe, können Transaktionen abgerufen werden.

### #3 - 29.09.2019 19:02 - martin

- Status wurde von New zu Feedback geändert

Hast Du es noch mal mit aktuellen Versionen probiert?

### #4 - 30.09.2019 11:28 - Anonym

Ich habe erneut getestet mit

- gwenhyfar 4b92f3b134a55075fa4aa5059fdd810b24e65e7e 4.99.19.0
- aqbanking 5cc6775e3905fa6d7e79be120f4ae8b9625fc8ce 5.99.36.0
- gnuccash 2684e51dbf12d3cf3a9aa4d3c7c9f22f5278e9d2 3.7

Der Abruf von Transaktionen/Salden funktioniert. Die Einrichtung des Bankingbenutzers aus GnuCash heraus funktioniert **nicht**. Nach Eingabe der Benutzerdaten wird der Server kontaktiert und dann geht es nicht weiter im Einrichtungs-Wizzard.

Man muss den Benutzer auf der Konsole anlegen, dann in GnuCash im Einrichtungs-Wizzard den Benutzer editieren und Systeminfo, Kontoliste, usw. abrufen, dann Kontos zuordnen, dann funktioniert es.

(Es wäre übrigens komfortabler und für die Community zugänglicher, wenn das Projekt auf GitHub/GitLab wäre und nicht in diesem Redmine-Dinosaurier hier. Es gibt ja schon <https://github.com/aqbanking/aqbanking>, einfach mal die Issues, Wiki, usw. dahin verschieben. Fände ich gut.)

**#5 - 30.09.2019 17:24 - martin**

Moin,

[...]

(Es wäre übrigens komfortabler und für die Community zugänglicher, wenn das Projekt auf GitHub/GitLab wäre und nicht in diesem Redmine-Dinosaurier hier. Es gibt ja schon <https://github.com/aqbanking/aqbanking>, einfach mal die Issues, Wiki, usw. dahin verschieben. Fände ich gut.)

[...]

Ich mag solche Kommentare - zumal noch im Bugtracker - nicht.

Es muss doch nicht jedes Open-Source-Projekte seine Arbeiten einer zentralen Infrastruktur anzuvertrauen, die inzwischen von Microsoft betrieben wird. Ich finde es ganz gut, dass es Alternativen gibt. Redmine ist eine solche Alternative, und sie tut momentan genau das, was unsere Projekte brauchen. Nicht mehr, aber auch nicht weniger. Und sie wird aktiv entwickelt.

Die Community hat uns in den letzten Wochen eigentlich auch mit dem von Dir so geschmaechten "Redmine-Dinosaurier" und den Mailinglisten ganz gut erreicht. Die Bugtracker haben uns bisher geholfen, den PSD2-Kram zu sortieren.

Auf github gibt es inzwischen unzählige mirrors, da steige ich gar nicht mehr durch, wer da warum welchen mirror betreibt. Aber ich muss mich daran ja nicht auch noch beteiligen.

Also. Genug jetzt mit der Kritik an verwendeten Open-Source-Projekten (wie eben Redmine, wo sicherlich auch einige Leute kostenlos und in ihrer Freizeit daran arbeiten) und lieber ausführliche Bugreports gepostet ;-)

Gruss

Martin

**#6 - 03.10.2019 14:16 - martin**

- Kategorie wurde auf AqBanking gesetzt

**#7 - 17.01.2020 22:19 - martin**

- Status wurde von Feedback zu Closed geändert

- AqBanking-Version wurde auf 5.99.36.0 gesetzt

- Betriebssystem Linux wurde hinzugefügt

- Anwendung GnuCash wurde hinzugefügt