AqBanking - Bug #271

Transaction discards all but the first line of the purpose

20.08.2022 00:38 - jralls@ceridwen.us

Status:	Feedback	**Beginn:**	20.08.2022
Priorität:	Normal	**Abgabedatum:**	
Kategorie:	AqBanking		
Betriebssystem:	Linux, MacOSX, Windows	**Anwendung:**	GnuCash
AqBanking-Version:	6.5.0	**Version der Anwendung:**	3.8 and later

Beschreibung

With an MT940 record element
:86:1017?61SAMMELGUTSCHRIFT FÜR KONTO: CH2909000000250094239 VERARBEI
TUNG VOM 17.06.2020 PAKET ID: 9999999999999999

AB_TransactionGetPurposeAsList() returns a GWEN_STRINGLIST of length 1 containing "7?61SAMMELGUTSCHRIFT FÜR KONTO: CH2909000000250094239 VERARBEI". The second line is lost.

I've traced this to AB_Transaction_ReadDb where
/* member "purpose" */
if (p_struct->purpose) {
free(p_struct->purpose); p_struct->purpose=NULL;
p_struct->purpose=NULL;
}   { const char *s; s=GWEN_DB_GetCharValue(p_db, "purpose", 0, NULL); if (s) p_struct->purpose=strdup(s); }
sets the value to only the first stored line in the db and
GWEN_STRINGLIST *AB_Transaction_GetPurposeAsStringList(const AB_TRANSACTION *t) {
const char *s;
assert(t);
s=AB_Transaction_GetPurpose(t);

```
if (s && *s)
{
    GWEN_STRINGLIST *stringList;
    stringList=GWEN_StringList_fromString(s, "\n", 0);
    return stringList;
  }
  return NULL;
}
Where AB_Transaction_GetPurposeAsStringList tries to split it into multiple lines on newlines. The catch is that Swift940.c's _parseTransactionData already did that and stored the results in the database in multiple strings.
```

I'm not able to find where AB_Transaction_ReadDb is implemented in transaction.xml

See https://bugs.gnucash.org/show_bug.cgi?id=797565

Historie

#1 - 20.08.2022 00:57 - martin

- Status wurde von New zu Resolved geändert

Hi,

in function AHB_SWIFT940_Parse_86(https://www.aquamaniac.de/rdm/projects/aqbanking/repository/revisions/master/entry/src/libs/plugins/parsers/swift/swift940_86.c) there are calls to _transformPurposeIntoOneString() which transforms the list of purpose strings into a single string before the transaction reaches AqBanking.

That call was in the code for structured :86: fields but missing for unstructured ones. I added that missing call 19 days ago and therefore it is already in AqBanking 6.5.3. So I consider this bug fixed unless reported otherwise.

BTW: transaction.xml gets translated into source files (transaction.{c,h}) which contain getters, setters as well as functions to read/write to/from GWEN_DB objects, XML objects and others using GWEN's typemaker2 tool. This is done by the packager before creating the release tarball if needed (or on the fly when using gwbuild).

Since transaction.{c,h} are generated files they are not in our GIT repository.

Regards
Martin

#2 - 21.08.2022 03:31 - jralls@ceridwen.us

I wonder if you might have gone too far. The whole transaction record is
:61:2006180618C1500,FMSC6536086135//07582000303F0C7U
2020061800005000303F0C7U000000000
:86:1017?61SAMMELGUTSCHRIFT FÜR KONTO: CH2909000000250094239 VERARBEI
TUNG VOM 17.06.2020 PAKET ID: 9999999999999999

and purpose is
2020061800005000303F0C7U000000000 1017?61SAMMELGUTSCHRIFT FÜR KONTO: CH2909000000250094239 VERARBEI TUNG VOM 17.06.2020 PAKET ID: 9999999999999999

Is it supposed to pick up the extra line from the :61: tag?

#3 - 21.08.2022 11:24 - martin

- Status wurde von Resolved zu Feedback geändert

I believe I asked you about what should be done with the extra data in the :61: field because in the test data you provided me a few weeks ago (I think in a Gnucash bug entry) that line was the only source of information about the transaction while the :86: field was nearly empty.

That's why I introduced the profile setting "readExtraData61" which is set to yes in the profile "SWIFT-MT940" (but not in the profile we use for HBCI in "fints940" because we don't have that extra data here AFAIK).

I wonder about the format of this data here: It seems like they tried to make it structured but never pulled through: They seem to be using a four digit code in the beginning of the :86: field and then use half-baked subfield numbers (introduced by "?"). Where is that file from?

#4 - 21.08.2022 19:16 - jralls@ceridwen.us

Martin,

Sorry, I'm not sure where that example came from save that it's a somewhat random one from a website that I found last month while I was looking for examples to test with. From the header info {1:F01POFICHB0AXXX0000000000}{2:I940XXXXXXXXXXXXN}{3:{108:SWIFT00010366395}}{4:
:20:SWIFT00010366395
:25:CH2909000000250094239
:28C:118/1
:60F:C200618CHF211993,19
it appears to be a swiss bank from 2 years ago.

The time stamp is 29 July so it's from before our email conversation about making a description field and your mention of a Swiss example file for the :NS: field handling.
Yes, they do appear to be using some sort of formatting that no doubt makes sense to their software.

I don't recall discussing the purpose field or extra :61: data with you before or even knowing that :61: might have extra data.

#5 - 21.08.2022 19:20 - martin

About the discussion of extra data in :61: please see https://www.aquamaniac.de/rdm/issues/262#note-6 and your response ;-)

#6 - 22.08.2022 01:34 - jralls@ceridwen.us

Ah, so peripheral I completely dumped it. And the response was "As for the supplemental data, heck if I know." That's still true.

The spec says what the SWIFT folks think it should be for--either to replace the customer reference if that's NONREF, to facilitate reconciliation, for something called ERI, or to provide an exchange rate for multi-currency transactions. The instant case seems to repeat the transaction date and the last 10 characters of subfield 8. That doesn't seem to me to support any of those purposes, but that's just that one instance from a possibly mythical bank. For all I know other banks put useful information in that field.

"If not that info what goes into AB_Transaction_GetTransactionText and AB_Transaction_GetPurposeAsStringList?" was a question, not a suggestion. I now know that TransactionText comes only from :NS:17 tags and I've found at least some of the places in MT940 that get used to create Purpose.

I guess it's one more component of that "what pieces do you want where" dialog box from https://bugs.gnucash.org/show_bug.cgi?id=797877, though it would have to be added to the file import code too, not just HBCI/FinTS online transaction retrieval. This is probably heading down the "more trouble than it's worth" path.