

AqBanking - Bug #262

SWIFT 940 parsing substitutes wrong entry date if the :61 field contains none.

28.07.2022 23:49 - jralls@ceridwen.us

Status:	Closed	Beginn:	28.07.2022
Priorität:	Normal	Abgabedatum:	
Kategorie:	AqBanking	Anwendung:	GnuCash
Betriebssystem:	Linux, MacOSX, Windows	Version der Anwendung:	4.11
AqBanking-Version:	6.5		
Beschreibung			
The SWIFT 940 format's transaction field provides for two dates, a mandatory value date and an optional entry date. It further specifies (https://www2.swift.com/knowledgecentre/publications/us9m_20220722/1.0?topic=con_sfld_HVXYFeYEEuYlf8G2SgDog_-694543811fld.htm) "Subfield 2, Entry Date, is normally only used when different from the value date.", implying that if it's missing the value date should be used for the entry date. When importing the attached file that has only value dates, <code>AB_Transaction_GetDate()</code> returns the date of the beginning balance (60) field instead of the date returned by <code>AB_Transaction_GetValutaDate()</code>.			

Historie

#1 - 29.07.2022 00:02 - martin

- Kategorie wurde auf AqBanking gesetzt
- Status wurde von New zu In Progress geändert

#2 - 29.07.2022 21:04 - martin

- Status wurde von In Progress zu Feedback geändert

John,

SWIFT formats are extremely complicated to get right across all parts of the information flow from bank server to the customer. Our SWIFT importer has evolved for probably 20 years now and it is now in a state where it works best with most banks here in Germany, that's why I'm reluctant to change the default behaviour.

There are major ways how those SWIFT documents are used here:

- 1) transactions statements for the whole report timespan with only ONE (or no) opening balance and ONE (or no) closing balance
- 2) for every day of the report timespan:
 - ONE (or no) opening balance for the current day of the report timespan
 - multiple transaction statements for the current day of the report timespan
 - ONE (or no) closing balance for the current day of the report

In case 2) the default to use the date of the opening balance is the correct choice because that determines the booking day of the report. That might not be what the specs say but it is what we learned how those documents are used by most bank servers here.

In case 1) you are right, it might be incorrect to use the date of the opening balance because that is the start of the whole report timespan.

But to determine what kind of document we have here (case 1 or 2) is not as simple as one might think. Nothing is really simple with SWIFT documents...

So I don't currently see how this behaviour of the parser can be changed easily without side effects for the current users :-/

Regards
Martin

#3 - 29.07.2022 21:49 - jralls@ceridwen.us

Martin,

Isn't it the case that in type 2 reports the value date and the balance date are the same if there's no entry date?

Have you any experience of banks using 12 30-day months for value dates and thereby producing invalid February dates (see https://bugs.gnucash.org/show_bug.cgi?id=798491)?

BTW, this arose out of investigating https://bugs.gnucash.org/show_bug.cgi?id=798578.

Regards,
John Ralls

#4 - 29.07.2022 22:18 - martin

John,

Isn't it the case that in type 2 reports the value date and the balance date are the same if there's no entry date?

Right, and thats what the parser currently does.

Have you any experience of banks using 12 30-day months for value dates and thereby producing invalid February dates (see https://bugs.gnucash.org/show_bug.cgi?id=798491)?

Yep, we encountered 30.02.xx as a date which is supposed to mean "last day of February", this special date is handled by the parser (so far only for valuta date of a transaction) and translated into the correct last day of February for the given year.

Regards
Martin

#5 - 29.07.2022 22:31 - jralls@ceridwen.us

Martin,

Well, if the value date matches the balance date then there's no harm in AB_Transaction_GetDate() returning the value date instead of the balance date when there's no entry date, is there? Doing so would also correctly handle the no-entry-date case for case 1.

The problem in bug 798491 was 2022-02-29. Since this isn't a leap year that's an invalid date too and the parser passed it through unchanged.

Regards,
John Ralls

#6 - 29.07.2022 23:13 - martin

John,

indeed, that was a bug I just fixed in Gwenhywfar (which handles date stuff). So passing invalid dates (e.g. 29 Feb in non-leap years etc.) should be fixed in GIT.

BTW: To import all the data of your SWIFT file the parser needs to accept the data at the end of the SWIFT elements (:61.; :86:). I will look into that as we currently ignore unstructured code at the end.

However, I'm not sure where to add e.g. "supplementary details" of field :61:.. Any ideas?

Regards,
Martin

#7 - 29.07.2022 23:54 - jralls@ceridwen.us

Martin,

OK, but how about using the value date for the entry date?

As for the supplemental data, heck if I know. It mostly looks out of scope for GnuCash. If not that info what goes into AB_Transaction_GetTransactionText and AB_Transaction_GetPurposeAsStringList?

#8 - 30.07.2022 00:28 - martin

John,

OK, but how about using the value date for the entry date?

That's what I tried to explain above (I'm afraid my rudimentary English skills betrayed me again ;-)

Please note that SWIFT MT940 documents are used by the German HBCI/FinTS banking protocol to report transaction statements for a requested period (well, actually, with most banks you can change that to CAMT which I did with my accounts).

Regarding the returned SWIFT document there is typically one of two situations:

Case 1:

Document contains

- opening balance for the report period (optional)
- a number of transactions
- closing balance for the report period (optional)

In this case you're right, when there is no entry date given, it might be best to use valuta date which aligns best with the specs you cited.

Case 2:

- for every day of the report period:
- opening balance for the day (optional)
- a number of transactions entered for the given day which affects the closing balance
- closing balance for the day (optional)

When there is no entry date then the date of the opening or closing balance is most probably the correct one, because that is the day for which transactions are reported. Using valuta date here might not be correct.

In my experience case 2 applies in most cases, that's why we use the date of the opening balance as fallback.

The 100% way would be of course to determine whether we have case 1 or case 2 and act accordingly. However, this determination isn't easy to make. It would really complicate the code (which already is complicated because of all the quirks for all the bank servers).

That's why I'm reluctant to change the current fallback strategy because that's what seems to be working for most banks here (which is up until now the prevalent use of the parser).

Changing that would mean next time a user requests a report the entry date of a known transaction might also change which interferes with the duplicate recognition of existing apps.

Actually, maybe it's not such a big problem after all, because in all German SWIFT files I've seen the entry date is set, but I don't know how often SWIFT documents of other users lack those entry dates...

Regards,
Martin

#9 - 30.07.2022 01:37 - jralls@ceridwen.us

Your English is fine. It's just that

Right, and thats what the parser currently does.

Gave me hope that you'd fix it, because using the value date is not what the parser currently does.

So I have two bugs (a dupe just appeared a couple of hours ago) from users whose banks send MT940 reports with no entry dates covering multiple days. Since there's no way that I see to distinguish case 1 and case 2 nor whether the transaction has an entry date from the info that AQB exposes I have to assume that I have a case 1 import and that therefore `AB_Transaction_GetDate()` returns an incorrect value and use only `AB_Transaction_GetValutaDate()`.

Regards,
John Ralls

#10 - 30.07.2022 01:45 - martin

John,

Gave me hope that you'd fix it, because using the value date is not what the parser currently does.

The only possibility I currently see is to add an importer profile with a new option which tells the parser what to do in case of a missing entry date. The original bug reporter could then use that new profile instead of the current one called "SWIFT-MT940".

Would that be an option?

As I wrote I currently see no way to reliably determine whether we have case 1 or case 2 (should be easier when there are multiple opening/closing balances for multiple days, but it would also have to work when a case 1 sends data for a single day only).

Regards,
Martin

#11 - 30.07.2022 02:12 - jralls@ceridwen.us

Martin,

Yes, that would work, though it would be applied to all MT940 users. The profiles are hardcoded into the dispatch function that interprets the e.g. `File>Import>MT940` menu item.

BTW, aside from matching the profiles with what Gwenhywfar knows about nearly all of the code that converts AQB information to GnuCash structures is common, regardless of import format. There are a few scattered checks that treat OFX differently from everything else. Is that a

problem?

Another off-topic question: You mentioned CAMT and we got a bug a few weeks ago saying that the Dutch and Belgian banks are dropping MT940 support in favor of CAMT. Can I just pass a CAMT profile to Gwenhywfar and expect reasonable results or will that require special conversion code like OFX does?

For distinguishing, I guess if you have an opening balance followed by a closing balance on the same day it's safe to assume that all of the transactions in between were entered on that day; if you have only an opening or a closing then you're taking a risk, but if you have an opening and followed by a closing on a different day then it's not reasonable to assume that the transactions in between were all entered on the same day.

Regards,
John Ralls

#12 - 30.07.2022 02:43 - martin

BTW, aside from matching the profiles with what Gwenhywfar knows about nearly all of the code that converts AQB information to GnuCash structures is common, >regardless of import format. There are a few scattered checks that treat OFX differently from everything else. Is that a problem?

Actually, the SWIFT importer is a special biest for historical reasons: It consists of two parts. One part parses the SWIFT message into a GWEN_DB containing the data from the file. This is a GWEN_DBIO parser which is an old interface for Gwenhywfar which allows to read arbitrary data into GWEN_DB structs. The other part reads the GWEN_DB and imports it into AqBanking structures. This approach was used for some very old formats like SWIFT and some German format called DTAUS.

Another off-topic question: You mentioned CAMT and we got a bug a few weeks ago saying that the Dutch and Belgian banks are dropping MT940 support in favor of CAMT. Can I just pass a CAMT profile to Gwenhywfar and expect reasonable results or will that require special conversion code like OFX does?

For those new XML/XML-like formats AqBanking uses its XML import/export module. It takes format files to already import/export AqBanking data to/from OFX, CAMT, SEPA/pain messages.

OFX is only in version 1 different and needs special handling because it is not XML but SGML, so to be able to handle those with the same parser we use some tricks which aren't necessary for any other format (like CAMT).

For the file format you might want to look at [https://www.aquamaniac.de/rdm/projects/aqbanking/repository/revisions/master/entry/src/libs/plugins/imexporters/xml/data/camt_052_001_02.xml]]. It's rather simple. Adding support for new SEPA or CAMT messages is quite easy: You only need to create a format file (like the one linked to above).

To make the new format known to applications you also need a simple profile file (like [https://www.aquamaniac.de/rdm/projects/aqbanking/repository/revisions/master/entry/src/libs/plugins/imexporters/xml/profiles/camt_052_001_02.conf]]).

Fortunately, those files already exist for at least those CAMT files used in DE/EUR.

For distinguishing, I guess if you have an opening balance followed by a closing balance on the same day it's safe to assume that all of the transactions in >between were entered on that day; if you have only an opening or a closing then you're taking a risk, but if you have an opening and followed by a closing on >a different day then it's not reasonable to assume that the transactions in between were all entered on the same day.

I understand, but there is no way to know that for sure, that's my point. If I have multiple opening/closing balances for different days then I **know** I look at case 2. But if there is only one day in the file I can't know for sure which case it is. Also, you can't trust that every bank uses the opening/closing balance the same way. A lesson we learned in all those years we had to adapt the parser to yet another variety of MT940 file introduced by some banks...

If I simply change that now the problem might be moved simply from one group of users to another. I'd rather like to have a good solution for all cases, and I'm sure we'll come up with one :-/

Regards,
Martin

#13 - 30.07.2022 05:09 - jralls@ceridwen.us

The question about various SWIFT profiles plus adding CAMT is from the AqBanking API side. Once the import is parsed I need to convert the information that AqBanking/Gwenhywfar have about it into GnuCash transactions. Is it reasonable to expect that functions like AB_Transaction_GetPurpose and AB_Transaction_GetValutaDate will have the basic transaction information regardless of the file format?

I understand, but there is no way to know that for sure, that's my point.

That's actually my point: There's no way to know for sure so you're making assumptions. That's not necessarily unreasonable, but it limits the

program to the one use case that's supported by your assumptions. If you made the underlying information available to client programs then they could make different assumptions based on their (possibly different) user demographic or even ask the user.

#14 - 30.07.2022 11:08 - martin

John,

The question about various SWIFT profiles plus adding CAMT is from the AqBanking API side. Once the import is parsed I need to convert the information that AqBanking/Gwenhywfar have about it into GnuCash transactions. Is it reasonable to expect that functions like `AB_Transaction_GetPurpose` and `AB_Transaction_GetValutaDate` will have the basic transaction information regardless of the file format?

That's the whole purpose of AqBanking's import/export interface... It's used internally by the banking backends to import received SWIFT, CAMT, SEPA and OFX documents and also to export SEPA documents.

I understand, but there is no way to know that for sure, that's my point.

That's actually my point: There's no way to know for sure so you're making assumptions. That's not necessarily unreasonable, but it limits the program to the one use case that's supported by your assumptions. If you made the underlying information available to client programs then they could make different assumptions based on their (possibly different) user demographic or even ask the user.

Before this discussion gets overly philosophical: The SWIFT importer is really ancient code which evolved over a long period of time into a module which learned to handle all the SWIFT documents we encountered so far (including Swiss versions of SWIFT).

BTW: By the same token ("leave it to the user") you could just make AqBanking's generic importer available to the user and let her/him choose the profile to use for import instead of hard-selecting importer and profile for the user via menu entries. The import dialog can also be used with presets (i.e. preset the dialog with previously selected file, importer and profile). I use it every day this way to import stock market info (I just need to select the correct file, the rest doesn't change).

Regards,
Martin

#15 - 30.07.2022 20:00 - jralls@ceridwen.us

Martin,

Martin,

That's the whole purpose of AqBanking's import/export interface...

Good. I hoped so, thanks for confirming it.

But to the original point of this bug: SWIFT MT940 specifies that there must always be a value date while the entry date is optional. I suppose that would be true of all of the SWIFT formats, but OFX flips it: DTPOSTED, the entry date, is mandatory and DTAVAIL, the value date, is optional. Am I correct that for an OFX import `AB_Transaction_GetDate()` will return DTPOSTED and `AB_Transaction_GetValutaDate()` will return DTAVAIL if it's included in the STMTTRN record or NULL if it's not?

BTW: By the same token ("leave it to the user") you could just make AqBanking's generic importer available to the user and let her/him choose the profile to use for import instead of hard-selecting importer and profile for the user via menu entries. The import dialog can also be used with presets (i.e. preset the dialog with previously selected file, importer and profile). I use it every day this way to import stock market info (I just need to select the correct file, the rest doesn't change).

I like that idea. IIUC we could populate a combo from the contents of plugins/imexporters and provide access to all of the importers you support instead of the small subset that's been hardcoded in the menu and largely unchanged for the last 15 years.

Regards,
John Ralls

#16 - 30.07.2022 22:45 - martin

John,

But to the original point of this bug: SWIFT MT940 specifies that there must always be a value date while the entry date is optional. I suppose that would be true of all of the SWIFT formats, but OFX flips it: DTPOSTED, the entry date, is mandatory and DTAVAIL, the value date, is optional. Am I correct that for an >OFX import `AB_Transaction_GetDate()` will return DTPOSTED and `AB_Transaction_GetValutaDate()` will return DTAVAIL if it's included in the STMTTRN record or NULL >if it's not?

I think the OFX importer is not very usefull to GnuCash because 1) you have your own OFX importer which is much better adapted to your app and 2) it most probably does not import all data your own importer uses.

The SWIFT MT940 importer enforces the valuta date because we know that this field always has to be set in MT940 and that it is a vital part of a transaction statement. But most other importer modules don't enforce any date. If a date is missing then the corresponding function (e.g. AB_Transaction_GetDate) will return NULL. That's because not every importer actually needs the date (e.g. transfer jobs in a SEPA file not always contain a date because the bank is supposed to execute the order on the same or next work day, so a date is not really needed).

I like that idea. IIUC we could populate a combo from the contents of plugins/imexporters and provide access to all of the importers you support instead of the small subset that's been hardcoded in the menu and largely unchanged for the last 15 years.

AqBanking provides a gui independent generic import dialog (please see [https://www.aquamaniac.de/rdm/projects/aqbanking/repository/revisions/master/entry/src/libs/aqbanking/banking_dialogs.h]) which can already be used without much work needed on application side. I use it frequently for CSV imports in AqFinance. It is also able to make use of presets if provided by the app.

Regards,
Martin

#17 - 30.07.2022 22:58 - jralls@ceridwen.us

Martin,

I think the OFX importer is not very usefull to GnuCash because 1) you have your own OFX importer which is much better adapted to your app and 2) it most probably does not import all data your own importer uses.

Except that we use AqBanking for OFX DirectConnect and those results go through the same conversion code.

/src/libs/aqbanking/banking_dialogs.h]]) which can already be used without much work needed on application side. I use it frequently for CSV imports in AqFinance. It is also able to make use of presets if provided by the app.

Thanks, I will.

Regards,
John Ralls

#18 - 21.08.2022 18:09 - martin

- Status wurde von Feedback zu Closed geändert

Dateien

mt940testdates.txt	1,61 KB	28.07.2022	jralls@ceridwen.us
--------------------	---------	------------	--------------------