Accord Migration technical Workshop

Jan Dings

22 and 24 March 2016

Agenda

1. Understanding the Accord specificities
   - Internal structure, theory, concepts, terminology
   - GUI aspects to consider w.r.t. migration
2. Understanding the migration approach
3. The steps to take in an Accord migration
4. The tools and documentation we'll make available

Accord internal structure, theory, concepts, terminology

Accord essentials

- **Central matching –** messages sent and received by a customer are copied to it
- Counterparty does not need to be on Accord
- Customers can **include/exclude** parts of their traffic
- Collective set of **default rules;** different **styles** available (e.g. regular versus CLS style, for MT 300)
- Personalised rules, “equivalences/synonyms”, known as **MRI**
- **GUI** and **API**
- Optional **Long Term Archive (→ not for today)**
- Items kept in live database till 7 days (14 for commodities) after value date/expiry date

Message types Matched in Accord

Foreign Exchange & Money Market	
MT 300	Foreign Exchange
MT 320	Fixed Loan/Deposit
MT 330	Call/Notice Loan/Deposit

Derivatives	
MT 305	Foreign currency option
MT 306	Exotic foreign currency option
MT 340	Forward rate agreement
MT 341	Forward rate agreement settlement
MT 360	Single currency interest derivatives
MT 361	Cross currency interest rate swap
MT 362	Interest rate swap reset / advice of payment

Commodities	
MT 600	Commodity trade confirmation
MT 601	Commodity Option Confirmation

Confirmation processing steps

Steps applied to every confirmation processed by Accord

Validation → fails? → status = “rejected” (e.g. duplicates, cancel of non-existing previous msg…)

Chaining (Amend, Cancel…)

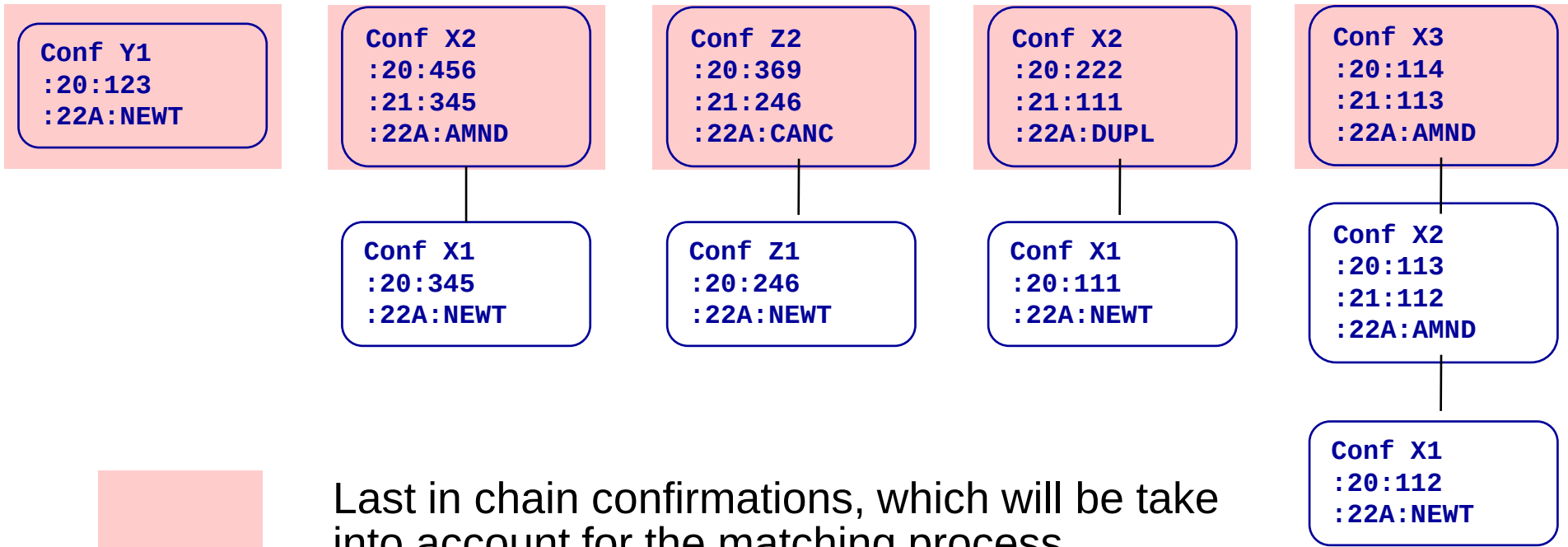
Matching

…

Archival (or deletion)

Chaining – Last in chain confirmations

- Used to initiate, amend or cancel a confirmation
- Rules will be provided in separate doc/course
- Important for this workshop: only last in chain msgs are exported



Last in chain confirmations, which will be take into account for the matching process.

Matching

Each MT has a list of unmatch fields, a list of mismatch fields, and some fields are ignored

MT300 **Un-match fields**
Sender Receiver 94A – Scope of operation 82a – Party A 87a – Party B 30V – Value date 32B – Currency, amount bought 33B – Currency, amount sold

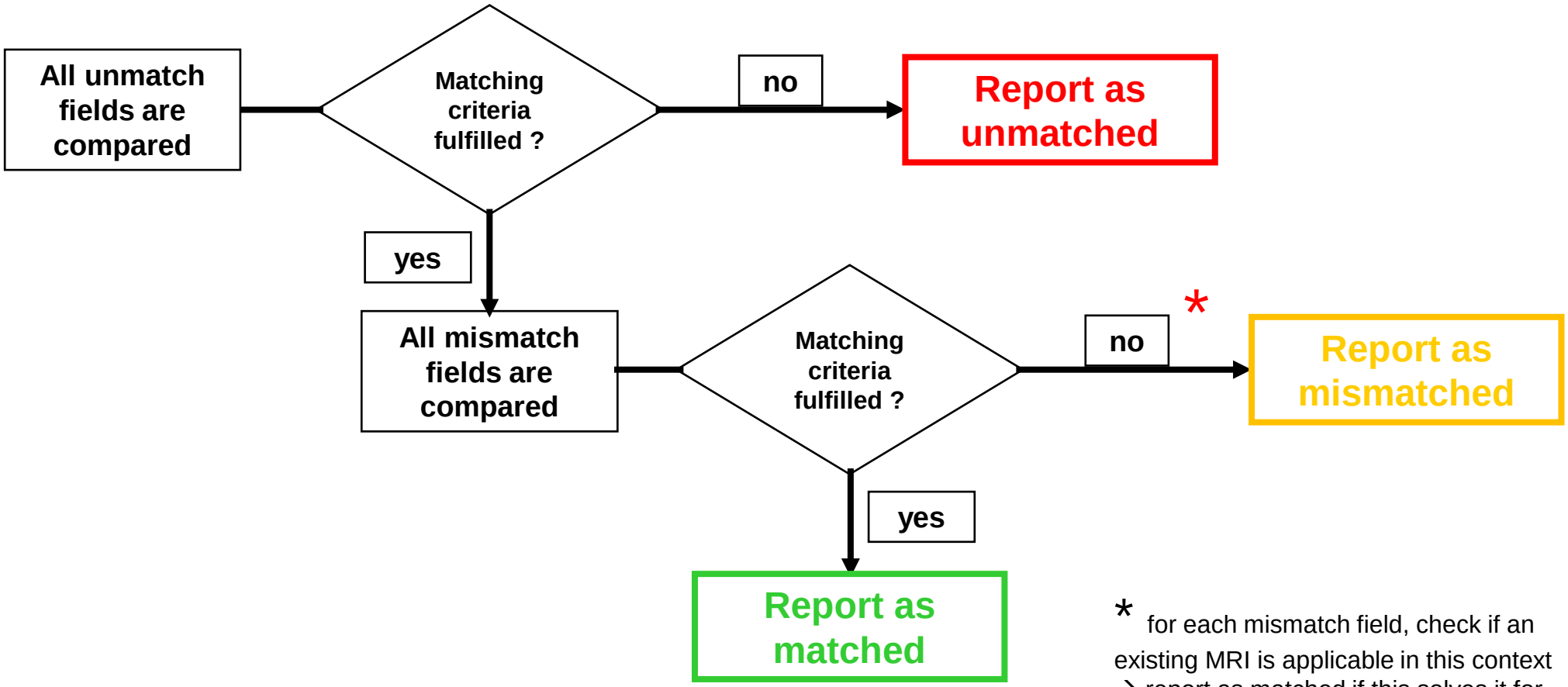
MT300 **Mis-match fields**
77D – Terms and Conditions 77H – Type, Date, Version of the Agreement 14C – Year of Definitions 83a – Fund or Beneficiary Customer 30T – Trade Date Seq B1 56a – Intermediary Seq B1 57a – Receiving Agent Seq B2 56a – Intermediary Seq B2 57a – Receiving Agent

Full matching rules to be supplied in separate doc /training

Two types of matching:

- Standard matching
- ‘Customised’ matching, using matching rule instructions (MRI)

Matching process flow



* for each mismatch field, check if an existing MRI is applicable in this context → report as matched if this solves it for all mismatching fields.

Customized Matching Rules (MRI)

- **Customized matching rules can be created :**
  - For **unmatched** confirmations if differences only in sender/receiver or 82a/87a
  - For **mismatched** confirmations if differences are on BICs or free text (example: 77D, 83a, 56a+57a, 31G, …) – e.g. BIC versus Full Name.
- Their application requires **specific context** to be present (e.g. Counterparty, MT, payment direction, CCY, for 56/57 MRI in MT 300)

- If not used for three months → deleted automatically. → **only relevant ones in the database**
- MRI **spanning several fields**: fields considered as one single space.(56a/57a; 77H/77D/14C in MT300)
- Cooperates with some flexible matching routines: e.g. MRI stating <*BANKBEBB*> = <*Banque Belge, Brussels*>, also applies when BANKBEBBXXX is used.

MRI Types – two special cases

- **30G MT 306** (generalised “presence with absence”), versus “presence_of_exactly_X with absence”
- **Un-matches** (iso mismatches), for combination of Sender/Receiver and fields 82a/87a (party A/party B). If difference is only in sender/receiver OR 82a/87a fields=**FPARTY type**

Perspective of msg sent			Perspective of msg Recvd
Sender	BANKBEBB	BANKBEBB	Receiver
Receiver	CPYTCCLL	CPTYCCLA	Sender
82	82A:BANKBEBB	82A:BANKBEBB	87
87	87D: counterparty Bank CCLL	82D:Counterparty Bank CCLL	82

Perspective of msg sent			Perspective of msg Recvd
Sender	BANKBEBB	BANKBEBB	Receiver
Receiver	CPYTCCLL	CPTYCCLL	Sender
82	82A:BANKBEBB	87D:BB Bank, Brussels	87
87	87D: counterparty Bank CCLL	82D:Counterparty Bank, LL	82

Confirmation – Deal status (=predefined set)

Matched: all **primary** (or unmatch) **and secondary** (or mismatch) fields **match** as per Accord rules, or a mismatch was **forced** into matched, or an **MRI** was applied to it

Mismatched: a pair is found with matching primary fields, but there are **differences in the secondary fields** → user can force match
!! A mismatch is always **THE** mismatch, not the best from a list

Unmatched: either a confirmation is **sent** or a confirmation is **received**, with no valid matching candidate → "**Pairing**" algorithm applied

Cancelled: the confirmation is no longer processed by Accord

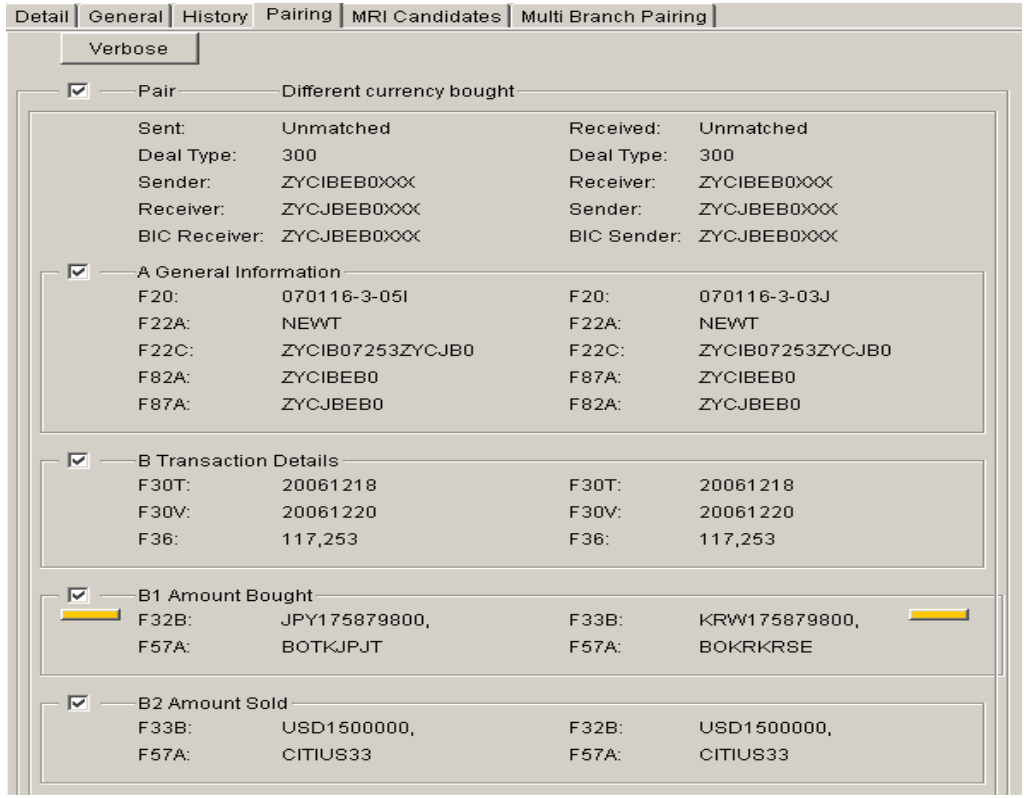
Rejected : the confirmation is not processed by Accord **(failed validation)**

Paired unmatched confirmations

- Accord will pair two unmatched items in opposite direction, with **specific limited, human errors** (often 1 field wrong only, or 1 + the fields calculated from it)
- Pairing details available under “Pairing” tab
- Not for MT 361
- If 1 candidate → most often the one intended reply, with 1 error in major field or set of related fields

→ **Export for migration: only pairs with ONE candidate are provided**

→ **Unmatched with 1 pair can correspond to specific status in your application**



Concept of Matching Entity

Logical grouping of SWIFT BICs (mostly BIC-11), these BICs are considered as "the same", when it comes to matching. E.g. CORPCAMMXXX containing 2 BICs (**CORPCAMM**TSY and **CORPCAMM**XXX

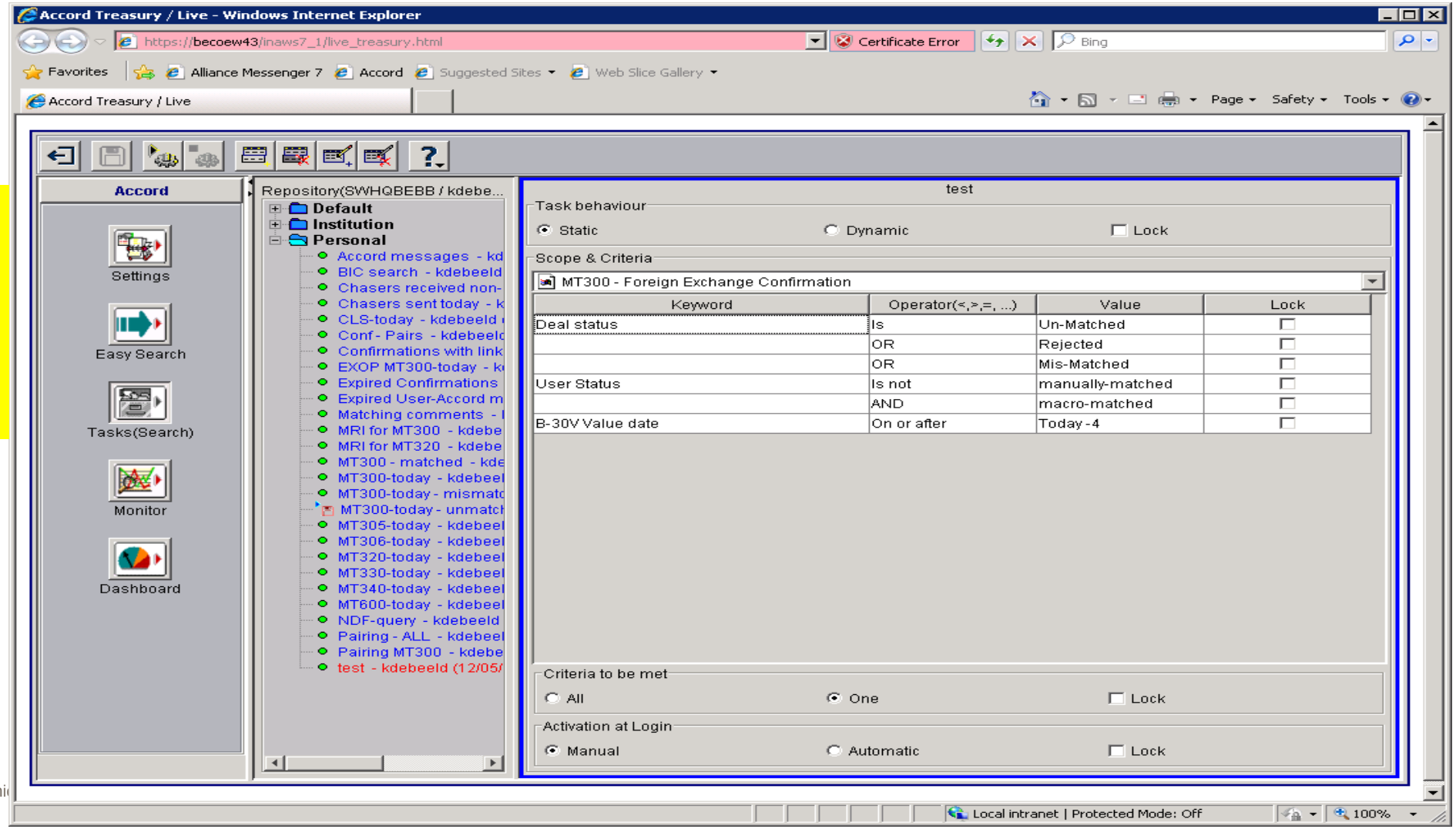
- **Every confirmation sent by one BIC of the Matching entity can potentially match with a confirmation received by another BIC of the entity**
- **One BIC can belong to only one ME; most MEs contain 1 BIC, same as their name.**
- **Everything is done at the ME level: billing, matching, provisioning, archiving, including/excluding…**

→ make sure you get those definitions! (we have them…)

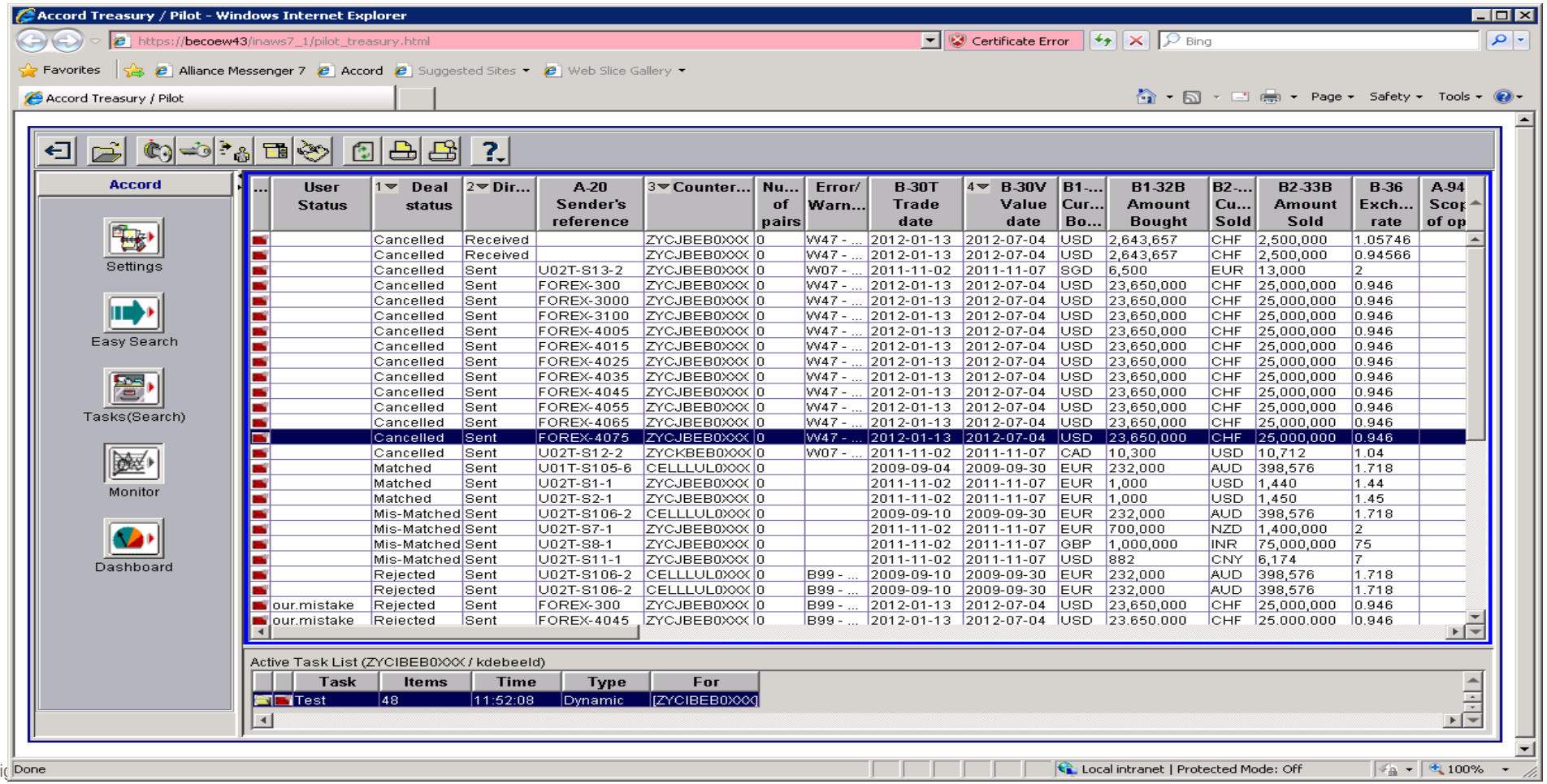
Accord GUI aspects to consider w.r.t. migration

Task (=query) – Definition

Export will be based on tasks too; most often the pre-defined tasks will be sufficient.



Summary – Selection result



Matching comments - Principle

- Indicates **how** Accord has processed the confirmation
- Often **part of the matching result !!** – e.g. force-matched versus automatched; contains list of mis-matching fields
- Column in the task output, also included in the Accord export

Deal status	Matching comments	B-30T Trade date	B-30V Premium Paymen...	Number of pairs
Matched	/CSU	2009-10-26	2009-10-28	0
Matched	/CSU	2009-10-12	2009-10-13	0
Un-Matched		2009-07-28	2009-07-30	0

- All applicable values are concatenated (X/Y/Z…) → use “contains” to find them

Matching comments – Processing info

- **CSU**: confirmation was “force-matched”
- **MRI**: an “equivalence” was applied when processing this confirmation
- **MTOL**: for match/mismatch, subject to an acceptable **rounding difference**
- **Mismatch reasons:** all mismatching fields are listed

  Example: A-77D/B1-57A

Settings Module – User Status

- **User-defined** statuses, defined at **Matching Entity** level
- Can be assigned to Confirmations, MRIs, and Chasers
- **Used to :**
  - Classify confirmations → can be **combined with pre-defined status**, to trigger settlement or not. (“OK for us”; “manually matched”; “settle”…)
  - Indicate next workflow step (“send chaser”; “call customer”…)

Sending a chaser from a confirmation

Free format message, sent:

- as **MT 399**, delivered by ACCOBEB3
- as free text, **directly attached** to the shared problem (between Accord subscribers)

Chasers can be sent for other objects too!

Chasers are **archived** at the same time as the related confirmation

Reporting Inclusion/Exclusion

- Relevant at moment **LTA** data is **exported**: important to prove that the archive is actually complete. (as opposed to: MT 305 excluded between dates X and Y).

- By default confirmations sent & received by a Matching Entity are either all reported or all ignored in Accord

This can afterwards be modified & configured per :

  - Message Type
  - Counterparty
  - Currency

Migration approach, steps to take and tools to use

Migration Approach

- **Responsibilities**
- **When and how to use the tools**
  - Benchmarking (multiple runs/compare your output with Accord's)
  - Backloading (seamless transition between Friday evening on Accord, Monday morning on Alternative Solution
- **Timeline**
- **Predefined tasks (=queries) and task creation (with SWIFT's help)**

Migration Approach - Responsibilities

- **450 entities, 240 master BICs, so about 250 decision makers**
- **Contractually: CUSTOMERS are responsible to migrate in time.**
- **In practice: Most banks will have a VENDOR, +/- 20 different ones in total**
- **Migration approaches:**
  - **Simplistic**: install new system, and use. No data migration, no parallel usage → smaller parties, or spot FX only (no accumulating data)
  - **Medium**: work in parallel, possibly feed some FIN retrievals in new system (traffic of last 124 days max), and/or reconfirm trades (of any age), and declare them matched manually

  → Problem, even for small parties: the “MRI” or equivalences/synonyms. To be recreated manually.

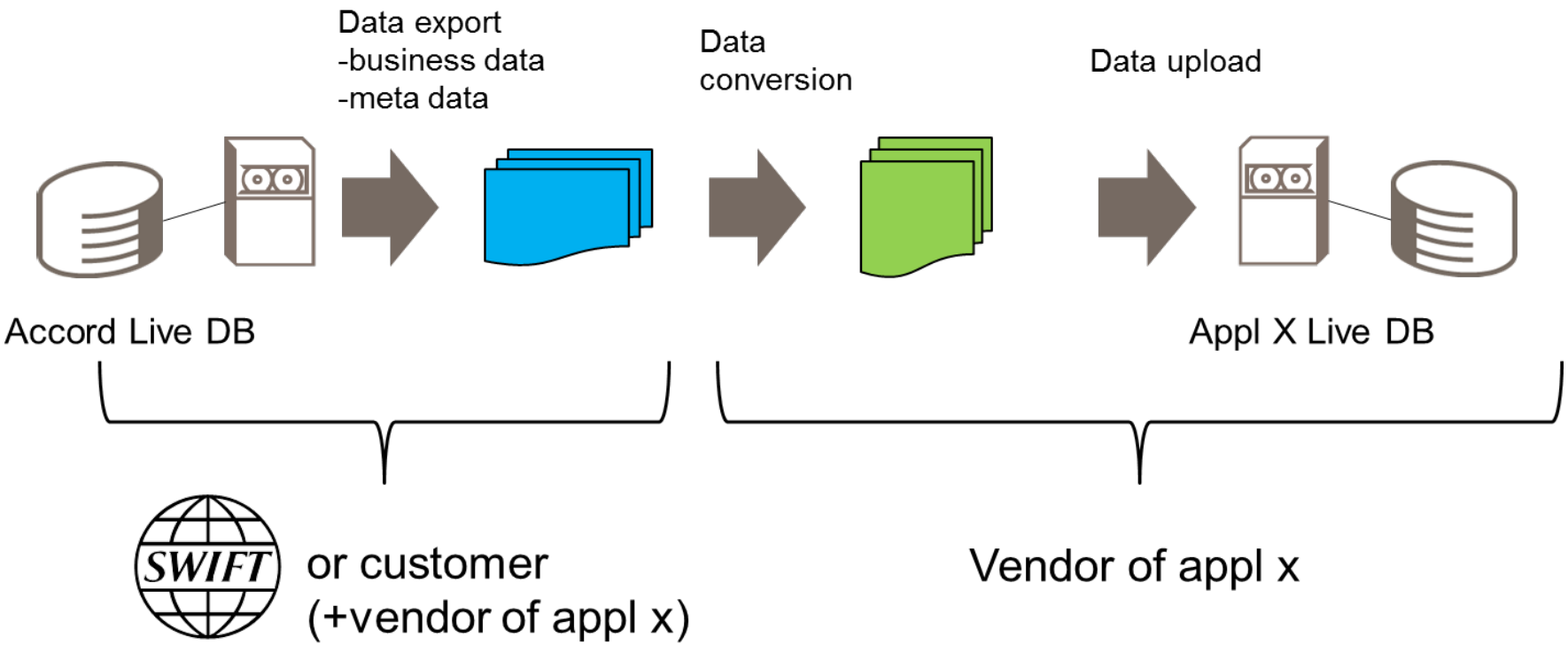
  - **Larger /punctual customers: FULL MIGRATION of all live data, MRIs, etc, after benchmarking with Accord.**

  → Significant project for specialist matching vendors

  → More projects, in more countries than SWIFT can deal with

  → Chosen approach: DIY migration tools, with full documentation and training for all interested vendors.

Migration Approach - Responsibilities



Accord migration: when and how to use the tools

- **Benchmarking/Tuning/Training**
  - Can be done several times
  - Can be done with full traffic, or relevant sample
  - Can be done per Entity, per MT…
  - No disruption of live Accord usage
- **Backloading of Accord replacement**
  - After successful benchmarking/try-outs

  → normally once, but can be repeated if necessary

  → even after replacement became prime system: still fall-back possible
  - Can be per Entity, per MT, any combination
- **Concerns LIVE DATA only: last in chain, last information, but about ALL Live transactions (i.e. less than 7 days beyond value date)**
- **LTA: collective export end 2017; plus delivery of Relational Database Tool for queries.**

Accord migration: when and how to use the tools

- **small/mid-size:** export in one single step (1 entity, using Mt 300/305/320; limited accumulated traffic) during day or evening hours.
- **Large bank:**
  - Many different **entities**, combination of centralised/local management, regional hubs
  - Desire to migrate **MT by MT**, or subset of MTs
  - Need to replace API-based integration with alternative
  - More steps required to deal with massive accumulated traffic, and **during weekends**:
  - Per MT/Entity or combination: export/import series of traffic slices
    1. Weekend-1: all data >1week old
    2. Daily evening exports of more recent data
    3. (possibly weekend-2) export of last updates, just before cut-over

Migration Timeline – Preparatory steps

- **Vendor**
  - Develop tools to analyse/transform Accord export
    - Extracting the FIN confirmations, metadata,
    - Converting “MRIs”
    - Automated results comparison

- **Customer (and/or Vendor)**
  - Get familiar with include/exclude mechanism on Accord (to avoid excessive billing) → document has been circulated to all customers.
  - Get familiar with facility on SWIFT FIN interface to start/stop feeding of cat 3 and 6 messages to New Matching server.
  - Study the next steps to take
  - Organise possibility to conduct TESTS, with … LIVE (i.e. “production”) data! → normally in production environment.

Migration – step 1 (export)

1. Export **MRIs** to file, with standard **GUI function**
2. **Manually** export ***relevant*** **semi-static data** from Accord: entity definition, User Statuses, user-defined lists, end-of-shift tasks, tasks (in order of importance)
3. **Automatically** export the “Last in chain data items” to a file – the “Open Items File”
   - You can take sample only, or a slice if needed or preferred.
   - ONLY IF you take this step for (potentially) the last time: take ALL data, do not take any actions in Accord, and ensure the alternative system is fully fed with FIN, and go to step 3B

Migration – step 2a (CONVERT)

- Transform all **static data** (MRIs file, matent definition etc.) into equivalents in the target solution, and upload the result
- Transform all "**last in chain data items**" and **import** the result (i.e. the **MT 3xx/6xx confirmations**) in the target application

---

Migration Step 2B – Validate, tune, retry…. looping

! Important for your first customers, to tune your system

- **reprocess the live items** using YOUR rules, and transformed MRIs.
- **Compare** matching results with results as in the export file.
- **Train operators** of new system
- **Not OK? ➔ Tune/improve** (matching rules, conversion tool…) → DB reset, convert and upload again… = retrying till satisfied
- **OK ? ➔** DB reset, redo step 1(in pieces, if necessary) **➔** // run (FIN feeding both)

    - // run ACCORD (alternative) = 3A → training, testing of responsiveness, etc.; go to 3B when satisfied.
    - // run (accord) ALTERNATIVE = 3B → Accord only for emergency fall back

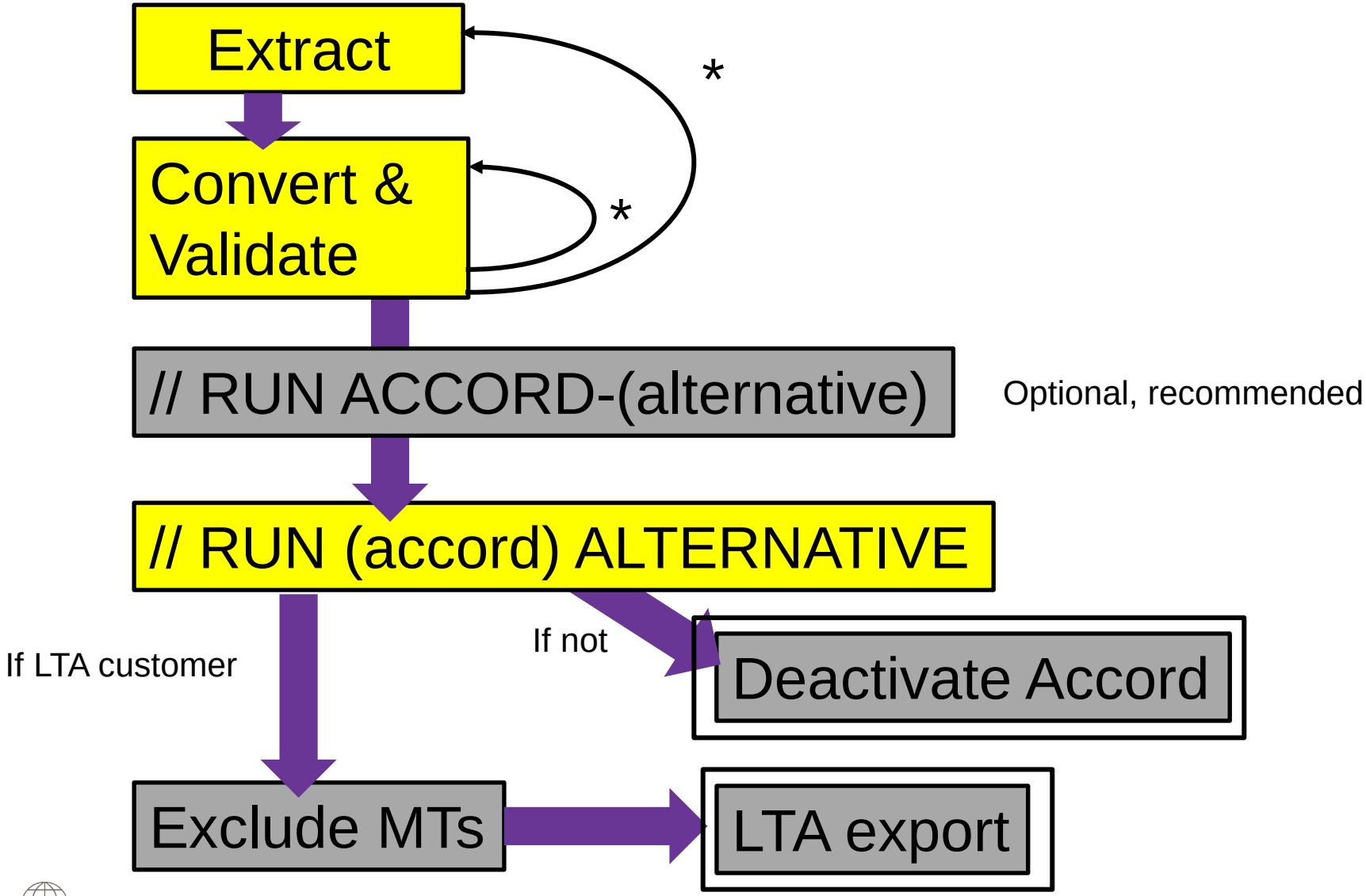
(!! If before step 3a, you do NOT redo the synchronisation… → Chaining will not work, as incoming AMENDs will not find the preceding NEW message)

Migration steps 3A, 3B – two types of // run

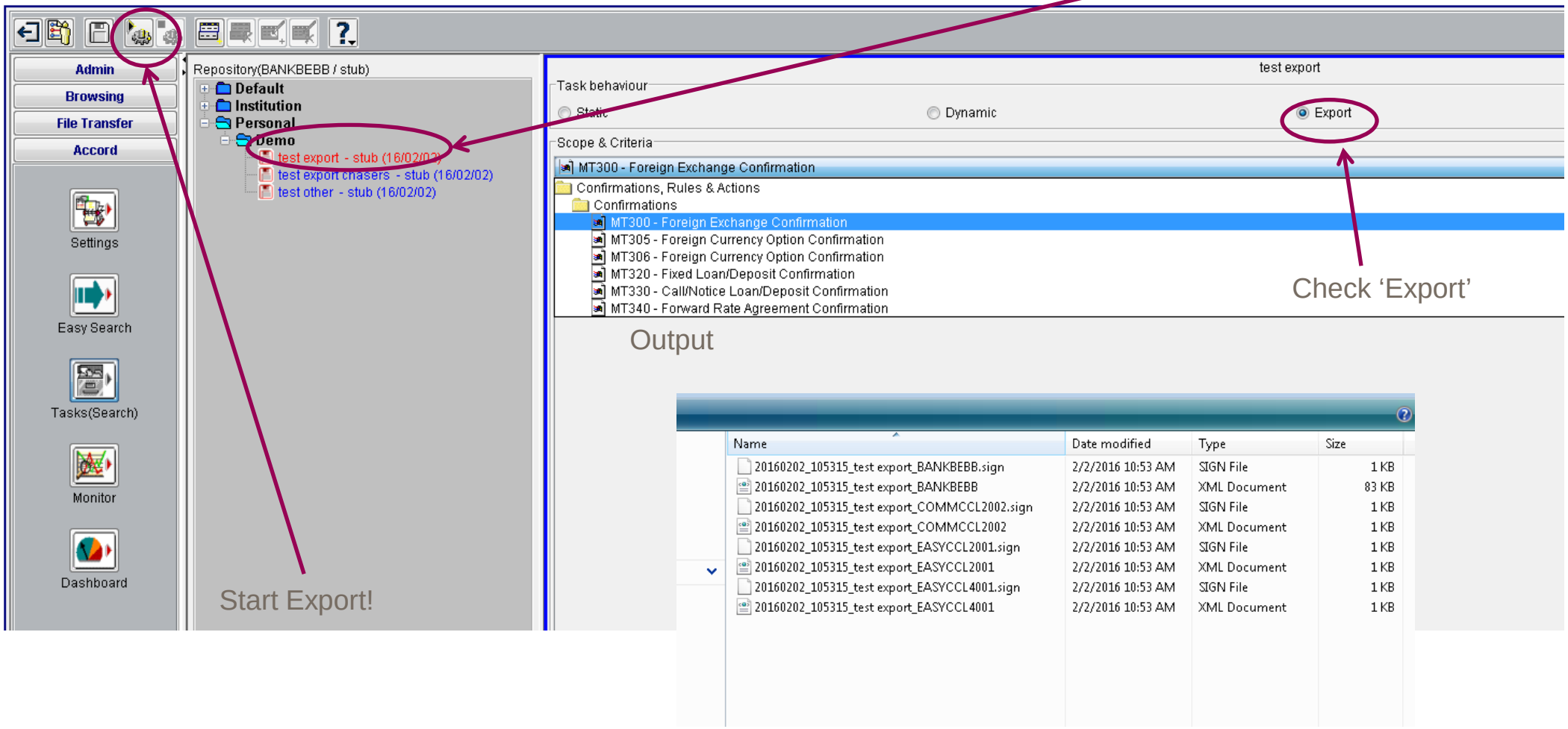
- **3A: ACCORD + (alternative); systems synchronised, and both fed by FIN**
  - Accord is still PRIME (feeding settlement etc.)
  - Training
  - Validation of responsiveness
  - Testing of All procedures around matching
- **3B: (accord) + ALTERNATIVE; systems synchronised, both fed by FIN**
  - Accord has become emergency fall-back
  - To fall-back: purge Alternative, and start from step 1; lost are the actions taken on Alternative
  - During // run: history of live items is in Accord (GUI), archived items are in LTA.
  - Do NOT modify anything in Accord, if you want LTA to reflect what went into Alternative.

Migration Steps 4, 5, 6 – deactivation, and LTA export

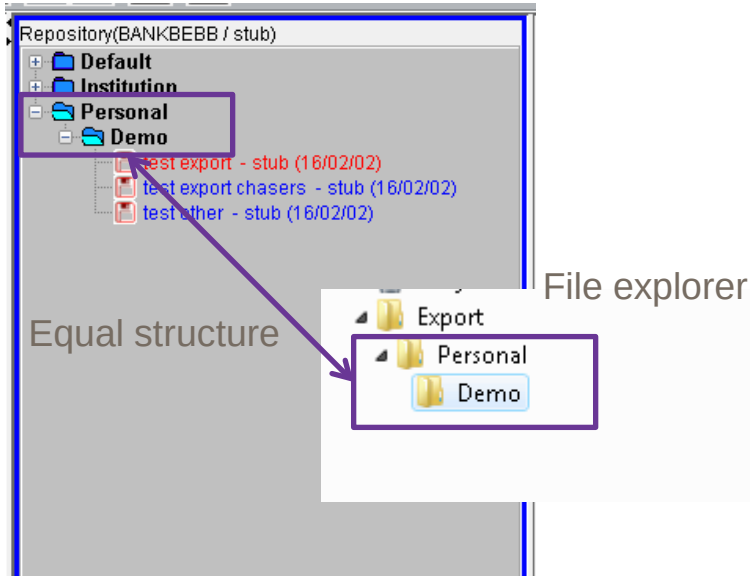
- **Step 4:** gradually **exclude MTs** from Accord (to prevent billing)
- **Step 5:** IF no LTA customer: **de-activate Accord**. Else: do full exclusion
- **Step 6**: (end 2017) - **LTA export**


Extract
*
Convert & Validate
*
// RUN ACCORD-(alternative)
Optional, recommended
// RUN (accord) ALTERNATIVE
If LTA customer
If not
Deactivate Accord
Exclude MTs
LTA export

GUI Export


Selection of task
Admin
Browsing
File Transfer
Accord
Settings
Easy Search
Tasks(Search)
Monitor
Dashboard
Repository(BANKBEBB / stub)
Default
Institution
Personal
Demo
test export - stub (16/02/02)
test export chasers - stub (16/02/02)
test other - stub (16/02/02)
Start Export!
test export
Task behaviour
Static
Dynamic
Export
Scope & Criteria
MT300 - Foreign Exchange Confirmation
Confirmations, Rules & Actions
Confirmations
MT300 - Foreign Exchange Confirmation
MT305 - Foreign Currency Option Confirmation
MT306 - Foreign Currency Option Confirmation
MT320 - Fixed Loan/Deposit Confirmation
MT330 - Call/Notice Loan/Deposit Confirmation
MT340 - Forward Rate Agreement Confirmation
Check 'Export'
Output
Name
Date modified
Type
Size
20160202_105315_test export_BANKBEBB.sign 2/2/2016 10:53 AM SIGN File 1 KB
20160202_105315_test export_BANKBEBB 2/2/2016 10:53 AM XML Document 83 KB
20160202_105315_test export_COMMCCL2002.sign 2/2/2016 10:53 AM SIGN File 1 KB
20160202_105315_test export_COMMCCL2002 2/2/2016 10:53 AM XML Document 1 KB
20160202_105315_test export_EASYCCL2001.sign 2/2/2016 10:53 AM SIGN File 1 KB
20160202_105315_test export_EASYCCL2001 2/2/2016 10:53 AM XML Document 1 KB
20160202_105315_test export_EASYCCL4001.sign 2/2/2016 10:53 AM SIGN File 1 KB
20160202_105315_test export_EASYCCL4001 2/2/2016 10:53 AM XML Document 1 KB

GUI Export – file structure

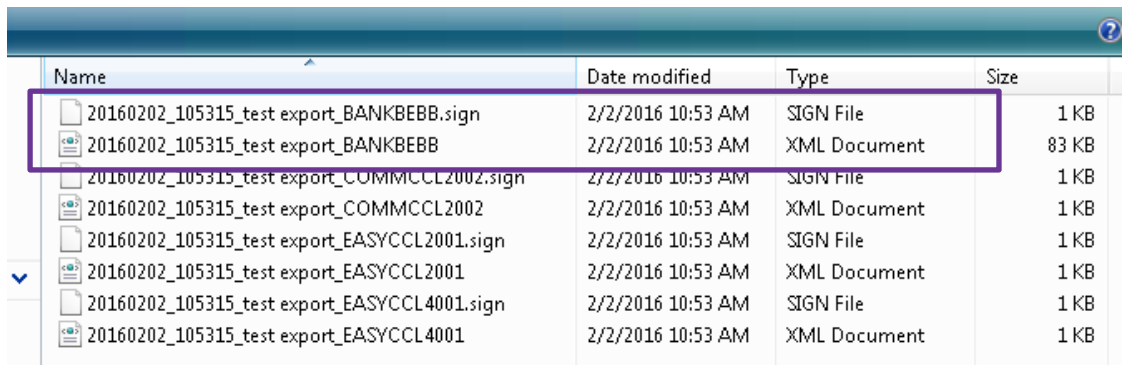


Two files: xml & sign file

- Xml file contains the exported data
- Sign file is a HMAC SHA-256 generated file with the given passphrase

  It guarantees
  - Completeness of data
  - Correct file transfer
  - File integrity

\Personal\Demo



Name	Date modified	Type	Size
20160202_105315_test export_BANKBEBB.sign	2/2/2016 10:53 AM	SIGN File	1 KB
20160202_105315_test export_BANKBEBB	2/2/2016 10:53 AM	XML Document	83 KB
20160202_105315_test export_COMMCCL2002.sign	2/2/2016 10:53 AM	SIGN File	1 KB
20160202_105315_test export_COMMCCL2002	2/2/2016 10:53 AM	XML Document	1 KB
20160202_105315_test export_EASYCCL2001.sign	2/2/2016 10:53 AM	SIGN File	1 KB
20160202_105315_test export_EASYCCL2001	2/2/2016 10:53 AM	XML Document	1 KB
20160202_105315_test export_EASYCCL4001.sign	2/2/2016 10:53 AM	SIGN File	1 KB
20160202_105315_test export_EASYCCL4001	2/2/2016 10:53 AM	XML Document	1 KB

Folder where the files will be stored:

C:/Users/%USER%/SWIFT/SNAccord_Export/<RepositoryFolder>/<RepositorySubFolder>/…

File naming convention:

<YYYYMMDD_hhmmss>_<TaskName>_<MatchingEntity>_<SeqNumber>

File integrity check: the Checking Tool

Purpose:

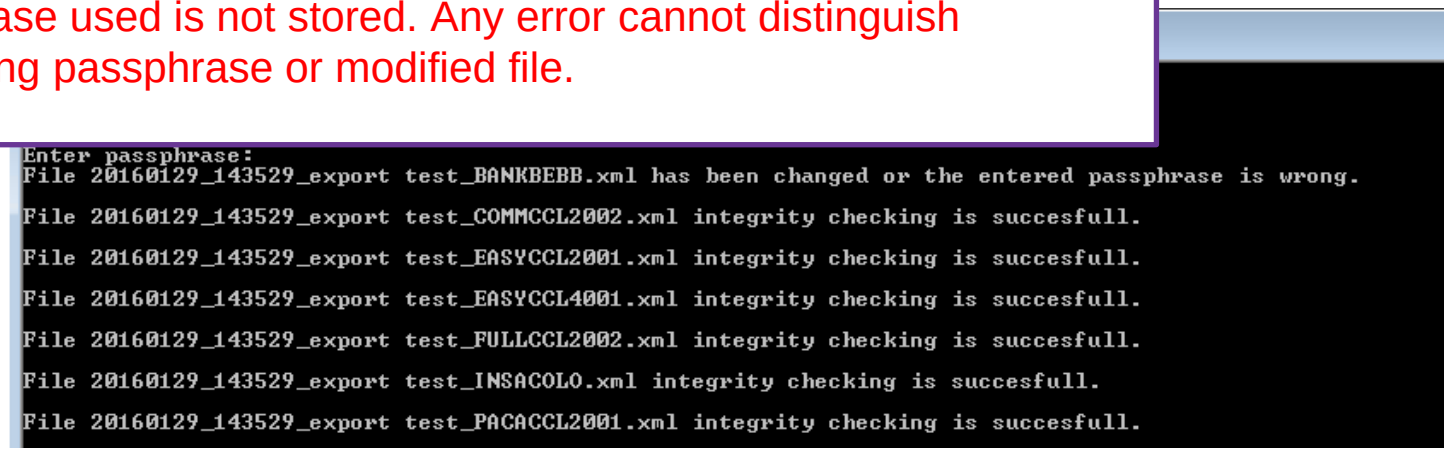
- Checking the integrity of the file with a given passphrase
- Proper error messages will be given

How?

- C
  p
- P

Warning!!

- File is not password protected, it is just a signature given to the file to check the integrity of the file
- The passphrase used is not stored. Any error cannot distinguish between wrong passphrase or modified file.



Three types of export files

1. Confirmations → **requires GUI 7.5 tool**
   - Last status of confirmation chain histories
   - Files are limited to 5,000 confirmation pairs

2. Chasers → **requires GUI 7.5 tool**
   - Chaser messages sent or received
   - Files are limited to 5,000 chasers

3. Equivalences (MRI) → **export possible NOW**
   - Customised matching rules
   - Csv type of data
   - One different per equivalence type

```
<Export>
        <Confirmations>
        </Confirmations>
        ...
        <Confirmations>
        </Confirmations>
</Export>
```

```
<Export>
        <ChaserMessage>
        </ChaserMessage >
        ...
        < ChaserMessage >
        </ChaserMessage >
</Export>
```

Confirmation Export file - content

For each confirmation:

Sent

<Sent>
<Payload>
<Key>
<SequenceNr>
<Time>
</Sent>

Received

<Received>
<Payload>
<Key>
<SequenceNr>
<Time>
</Received>

StatusBlock

<Sent>
<Payload>
{1:F01PACACCL2A0011112141610}{2:I300PACACCL2X002N}
{3:{108:.}}{4:
15A:
:20:3009981435316001
:22A:NEWT
:22C:PACAL20001PACAL2
:17I:N
:82A:PACACCL2001
:87A:BANKBEBB
:83J:/ACCT/123456
/NAME/NAME1
:15B:
:30T:20151112
:30V:20151112
:36:1,000
:32B:EUR111214351,
:57A:AAAAVVL2BRN
:33B:USD000000002000,
:57A:AAAAVVL2BRN}
{5:{MAC:ACBDEF12}{CHK:1A5B65BF545A}}
</Payload>
<Key>1131806164,145441,6</Key>
<SequenceNr>1</SequenceNr>
<Time>2015/11/11 12:24:38 GMT</Time>
</Sent>

<Received> ….. ***(Same format as Sent, for the received confirmation)***
</Received>

Confirmation Export file - content

Confirmation

Sent

Received

StatusBlock

```
<StatusBlock>
        <Status>
        <Time>
        <MComments>
        <ManualMatch>
        <UserStatus>
        <UserStatusTime>
        <UserComments>
        <RejectReason> *
        <RejectText> *
</StatusBlock>
```

```
<StatusBlock>
  <Status>S</Status>
  <Time>2015/11/12 14:36:05 GMT</Time>
  <MComments>/A-83/RATE</MComments>
  <ManualMatch>FALSE</ManualMatch>
  <UserStatus></UserStatus>
  <UserStatusTime></UserStatusTime>
  <UserComments>This is a mismatch</UserComments>
</StatusBlock>
```

* Can be absent, when not relevant

Chaser Export file content

ChaserMessage

- **<SenderBic>**
- **<ReceiverBic>**
- **<Time>**
- **<DealType>**
- **<MessageBlock>**
  - **<Direction>**
  - **<Text>**
  - **<UserComments>**
- **<RelatedConfirmationKey>**
- **<RelatedConfirmationSequence>**

```
<Export>
   <ChaserMessage>
      <SenderBic>BNPAAEAAXXX</SenderBic>
      <ReceiverBic>DEUTGB2LXXX</ReceiverBic>
      <Time>2015/12/07 14:26:03 GMT</Time>
      <DealType>395</DealType>
      <MessageBlock>
         <Direction>Received</Direction>
         <Text>:20:1425CNF002
:21:1425CNF001
:75:QUERY TEXT
:77A:
:11R:300
151207
9876543210
:79:text of my chaser message on fx confo
         </Text>
         <UserComments>my comment on the chaser as a user</UserComments>
      </MessageBlock>
      <RelatedConfirmationKey>1133619962,305111,6</RelatedConfirmationKey>
      <RelatedConfirmationSequence>1</RelatedConfirmationSequence>
   </ChaserMessage>
</Export>
```

Export file content – key points

- Data reported is based on GUI search criteria
- Both sent and received confirmations are reported
  - Unmatch cases – sent or received confirmation only
- A chaser may refer to a confirmation not retrieved in a confirmation file
  - Check RelatedConfirmationSequence field
- XML Encoding for
  - UserComments
  - ChaserText

```
<UserComments>
        <![CDATA[<Test1>É<Test2>Ü<Test3>]]>
</UserComments>
```

- Detailed character sets and field length to be documented in the
  - Documentation
  - Export file schemas

Equivalences (“MRI”) – Fields 56/57, MT 3xx

Results generated by standard GUI export include:

- The message type and counterparty it applies to

Entity	Deal type	Counterparty
BANKBEBBXXX	MT 300 - Foreign Exchange Confirmation	BARCSGSGXXX

- A *common* set of columns, mostly not relevant for this context

Status	User Status	Locked by	Tagged to	Rule status	Action	Action status	Action created by	Note	Item status	Archival date	Status date	Is locked	Is tagged
Detail seen or Action taken				Active	Extend matching Rule	Complete	John	No	Item Active	never	2008-05-15 13:28:08 GMT	No	No

- The *specific* fields for the type of equivalence, in specific columns:

A Currency	A Payment direction	A56 Tag	A56 Account line	A56 Address	A57 Tag	A57 Account line	A57 Address	B Currency	B Payment direction	B56 Tag	B56 Account line	B56 Address	B57 Tag	B57 Account line	B57 Address
USD	Pay	A		BARCUS33XXX	A		BARCGB5GXXX	USD	Receive	A	/12345	BARCUS33XXX	A		BARCGB5GXXX

Equivalences – 17 types of equivalences in total

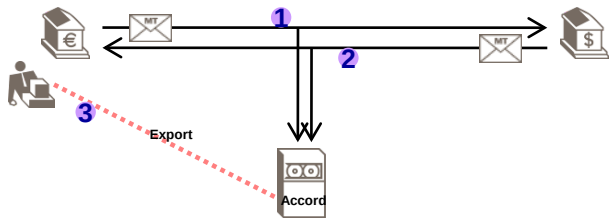
- Specific subset per message type
- Some apply across multiple message types
- Not always same fields used from my and counterparty's confirmation
- Equivalences use Accord matching rules (Not a bitmap comparison)

message type	*Equivalence fields (my confirmation)*	*Type*	*Equivalence fields counterparty's view)*
MT300	seqB1 F56/57	F56F57	seqB2 F56/57
	seqB2 F56/57	F56F57	seqB1 F56/F57
	seqA F83a	F83	seqA F83a
	seqA F77H, seqA F77D, seqA F14C	F77	seqA F77H, seqA F77D, seqA F14C
	seqA F82/87	FPARTY	seqA F87/82
	Seq A F17I	F17I	Seq A F17I

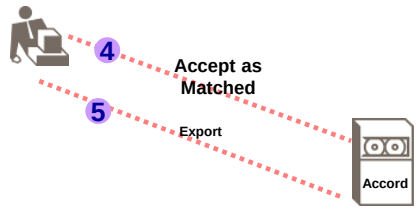
message type	*Equivalence fields (my confirmation)*	*Type*	*Equivalence fields counterparty's view)*
MT306	seqA F82/87	FPARTY	seqA F87/82
	SeqA-83a	F83	SeqA-83a
	seqA F77D	F77	seqA F77D
	SeqA-22K (second subfield)	F22K	SeqA-22K (second subfield)
	SeqB-29E	F29E	SeqB-29E
	SeqC-56a	F56F57	SeqC-56a
	SeqC-57a		SeqC-57a
	SeqE-56a	F56F57	SeqE-56a
	SeqE-57a		SeqE-57a
	SeqF1-29J	F29J	SeqF1-29J
	SeqF1-29K	F29K	SeqF1-29K
	seqA-77H	F77H	seqA-77H
	seqF1-30G	F30G	seqF1-30G

Exported data: examples of a few advanced considerations

Split runs & message duplicates – Example 1



A mis-matched confirmation



becomes matched at a subsequent export

```
<Confirmations>
   <Sent>
      <Payload>{1:...}…{5:…}</Payload>
      <Key>1138805474,148054,2</Key>
      <SequenceNr>1</SequenceNr>
      <Time>2016/02/01 14:51:16 GMT</Time>
   </Sent>
   <Received>
      <Payload>{1:...}…{5:…}</Payload>
      <Key>1138805477,113260,6</Key>
      <SequenceNr>1</SequenceNr>
      <Time>2016/02/01 14:51:17 GMT</Time>
   </Received>
   <StatusBlock>
      <Status>S</Status>
      <Time>2016/02/01 14:51:17 GMT</Time>
      <MComments>/B2-56</MComments>
      <ManualMatch>FALSE</ManualMatch>
      …
   </StatusBlock>
</Confirmations>
<Confirmations>
   …
```

```
<Confirmations>
   <Sent>
      <Payload>{1:...}…{5:…}</Payload>
      <Key>1138805474,148054,2</Key>
      <SequenceNr>1</SequenceNr>
      <Time>2016/02/01 14:51:16 GMT</Time>
   </Sent>
   <Received>
      <Payload>{1:...}…{5:…}</Payload>
      <Key>1138805477,113260,6</Key>
      <SequenceNr>1</SequenceNr>
      <Time>2016/02/01 14:51:17 GMT</Time>
   </Received>
   <StatusBlock>
      <Status>M</Status>
      <Time>2016/02/01 14:54:09 GMT</Time>
      <MComments>/CSU</MComments>
      <ManualMatch>TRUE</ManualMatch>
      …
   </StatusBlock>
</Confirmations>
<Confirmations>
   …
```

Two records in subsequent export files: same **key**, different **status**.

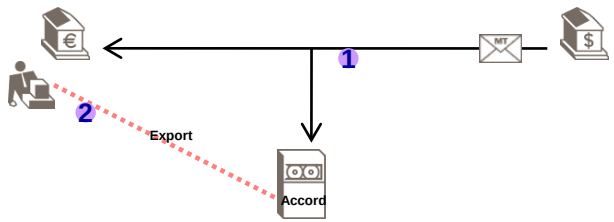
These are not duplicate confirmations, but two confirmations in the same chain.

When importing, be aware that only the most recent status is correct.

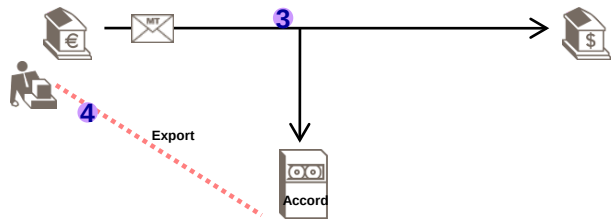
Reminder:

"amend" or "cancel" confirmations share the same key as the original confirmation that they chain with.

Split runs and message duplicates – Example 2



An un-matched received confirmation



becomes matched at a subsequent export

```
<Confirmations>
   <Sent/>
   <Received>
   <Payload> {1:...}... {5:...} </Payload>
   <Key>1138811591,126449,6</Key>
   <SequenceNr>1</SequenceNr>
   <Time>2016/02/01 16:33:11 GMT</Time>
   </Received>
   <StatusBlock>
   <Status>U</Status>
   <Time>2016/02/01 16:33:11 GMT</Time>
   <MComments/>
   <ManualMatch>FALSE</ManualMatch>
   …
   </StatusBlock>
</Confirmations>
<Confirmations>
```

```
<Confirmations>
   <Sent>
      <Payload>{1:..}…{5…}</Payload>
      <Key>1138811831,128379,2</Key>
      <SequenceNr>1</SequenceNr>
      <Time>2016/02/01 16:37:11 GMT</Time>
   </Sent>
   <Received>
      <Payload>{1:...}... {5:...} </Payload>
      <Key>1138811591,126449,6</Key>
      <SequenceNr>1</SequenceNr>
      <Time>2016/02/01 16:33:11 GMT</Time>
   </Received>
   <StatusBlock>
      <Status>M</Status>
      <Time>2016/02/01 16:37:11 GMT</Time>
      <MComments/>
      <ManualMatch>FALSE</ManualMatch>
      …
   </StatusBlock>
</Confirmations>
<Confirmations>
   …
```

Two records in subsequent export files share the same confirmation key.

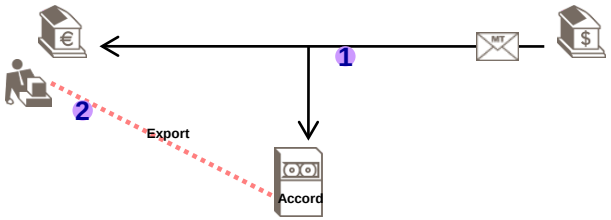
These are not duplicate confirmations, but two confirmations in the same chain.

When importing, be aware that only the most recent status is correct.

Reminder:

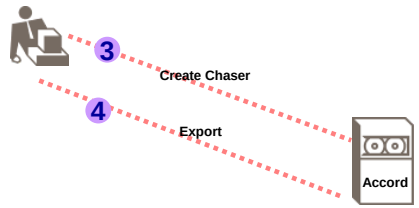
"amend" or "cancel" confirmations share the same key as the original confirmation that they chain with.

Cross-referencing between items – Example



A confirmation

```
<Confirmations>
   <Sent>
      <Payload>{1:…}… {5:…}</Payload>
      <Key>1138287363,215699,2</Key>
      <SequenceNr>1</SequenceNr>
      <Time>2016/01/26 14:56:03 GMT</Time>
   </Sent>
   <Received/>
   <StatusBlock>
      <Status>U</Status>
      <Time>2016/01/26 14:56:03 GMT</Time>
      <MComments/>
      <ManualMatch>FALSE</ManualMatch>
      …
   </StatusBlock>
</Confirmations>
<Confirmations>
…
```



A chaser is created

```
<ChaserMessage>
   <SenderBic>MAFOBEBBXXX</SenderBic>
   <ReceiverBic>PACACCL2001</ReceiverBic>
   <Time>2016/02/02 09:38:51 GMT</Time>
   <DealType>300</DealType>
   <MessageBlock>
      <Direction>Sent</Direction>
      <Text><![CDATA[Chaser text, escaped]]></Text>
      <UserComments/>
   </MessageBlock>
   <RelatedConfirmationKey>1138287363,215699,2</RelatedConfirmationKey>
   <SequenceNr>1</SequenceNr>
</ChaserMessage>  …
```

Related records in export files share the same key.

Tools, documentation and training
The help available from SWIFT

1. **Documents**
   - MRI guide: you should have already
   - Full Accord matching rules Guide, from developers' perspective
   - Full migration GUIDE
   - Format examples
2. **Tools**
   - Export tool; Accord GUI 7.5, June 2016
3. **Courses (to be confirmed)**
   - Accord matching rules; from a developer's perspective
   - Migration, in detail
   - → let us know level of interest, preferred location/duration

Thank you !