**Maintenance Change Request**

**for the update of ISO 20022 financial repository items**

1. **Origin of the request:**

*A.1 Submitter*: nexo A.I.S.B.L

*A.2 Contact person:*

* Andrew Hamilton ([andrewr.hamilton@ncratleos.com](mailto:andrewr.hamilton@ncratleos.com))
* Philippe Cece (philippe.cece@nexo-standards.org)

*A.3 Sponsors*: nexo A.I.S.B.L.

1. **Related messages:**

* ATMExceptionAdviceV01 (Caam.011.001.01),
* ATMWithdrawalRequestV02 (catp.001.001.02),
* ATMWithdrawalResponseV02 (catp.002.001.02),
* ATMWithdrawalCompletionV02 (catp.003.001.02),
* ATMInquiryRequestV02 (catp.006.001.02),
* ATMInquiryResponseV02 (catp.007.001.02),
* ATMCompletionAdviceV02 (catp.008.001.02),
* ATMPinManagementRequestV02 (catp.010.001.02),
* ATMPinManagementResponseV02 (catp.011.001.02),
* ATMDepositRequestV01 (catp.012.001.01),
* ATMDepositResponseV01 (catp.013.001.01),
* ATMDepositCompletionAdviceV01 (catp.014.001.01),
* ATMTransferRequestV01 (catp.016.001.01),
* ATMTransferResponseV01 (catp.017.001.01).

1. **Description of the change request:**

The following lists the changes requested. Changes are highlighted with a yellow highlighter.

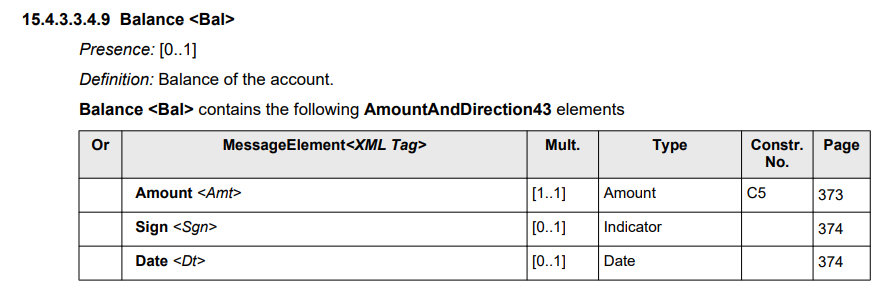
1. ExpiryDate <XpryDt> within PlainCardData <PlainCardData>

Enhance the description of the ExpirytDate <XpryDt> element. The description at present states: “Expiry date of the card expressed either in the YYYY-MM format, or in the YYYY-MM-DD format”. The proposal is to add the format “YY-MM” to the description. The standard ISO card (ISO/IEC 7813) uses the format YYMM. The client should not be guessing what the century is; it is up to the card issuer to determine this. The definition would then become:

*Definition*: Expiry date of the card expressed either in the YY-MM format, YYYY-MM format, or in the YYYY-MM-DD format.

1. Support Multiple Balances within AccountInformation <AcctInf> within ATMInquiryResponse <ATMNqryRspn>

The Account Balance returned in an ATMInquiryResponse is defined as:



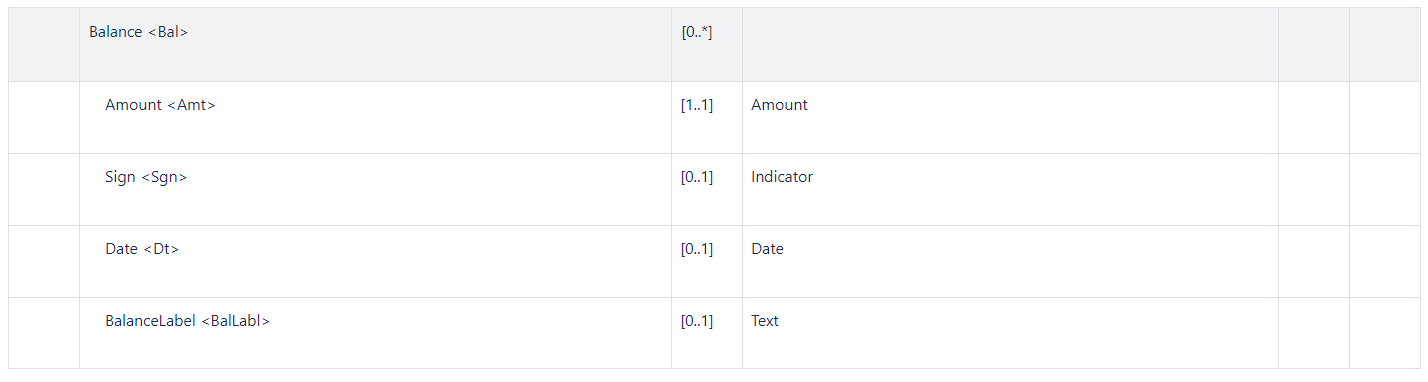
This limits this response to return a single Balance per account. There is a requirement to return multiple balances. An example is where we wish to return a ‘ledger’ and an ‘available’ balance. To enable this we would propose making the cardinality of Balance <Bal> [0..\*], and adding an additional sub-element Label which defines the type of balance being returned.

The new structure would then be:

Balance <Bal>

Presence: [0..\*]

Definition: Balance of the account.



where BalanceLabel is defined:

Label <BalLabl>

Presence: [0..1]

Definition: Description of the balance amount that may be provided to the customer (for example ‘Available’, ‘Ledger’). If this element is not included in a Balance <Bal> it is assumed the balance is a ‘ledger’ balance.

1. Authorisation Code <AuthstnCd>

The AuthorisationCode <AuthstnCd> within the AuthorisationResult **<**AuthstnRslt**>** a ATMWithdrawalCompletionAdvice <ATMWdrwlCmpltnAdvc> is defined as:

Presence: [0..1]

Definition: Value assigned by the authorising party.

Datatype: "Min6Max8Text" on page 1299

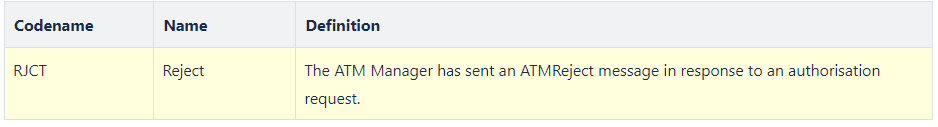
This type seems to be restricted to the value element 38 in an ISO 8583 response. The requirements for this data type should be extended to include a string whose minimal length is 1 and maximal length is 8 and hence not be restricted to the ISO 8583 value (i.e. Min1Max8Text). Hence the change would be:

Datatype: " Min1Max8Text "

1. Reject Incident Code

When we get an ATMReject message in a response to an authorisation request, there is value in sending a Completion Advice message to indicate this has occurred even if no other incidents have occurred. In this case bank staff would be able to answer any customer queries regarding this failed transaction.

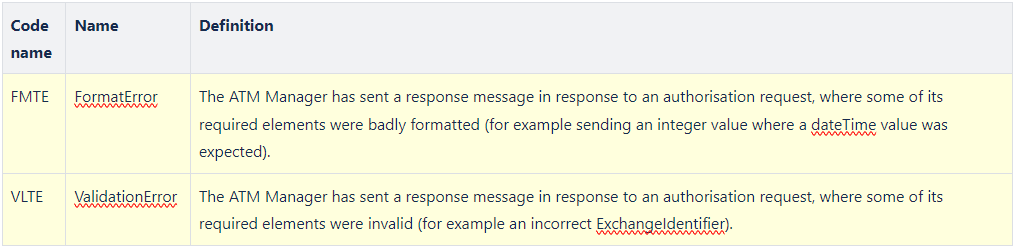
The proposal is to add to <Incdnt> the following value (i.e. a change to FailureReason7Code):



1. FormatError & ValidationError Incident Code

When we get an authorisation response which fails some validation error (for example the ExchangeIdentifier does not match the Request’s one) or format (for example sending an integer where a dateTime was expected), instead of sending an ATMReject message to the server some will prefer to send a Completion Advice message. In this case it would be necessary to send an Incident with information of the reason for a transaction failure. The following suggests we send an incident to inform the server a validation or format error which occurred with the authorisation response.

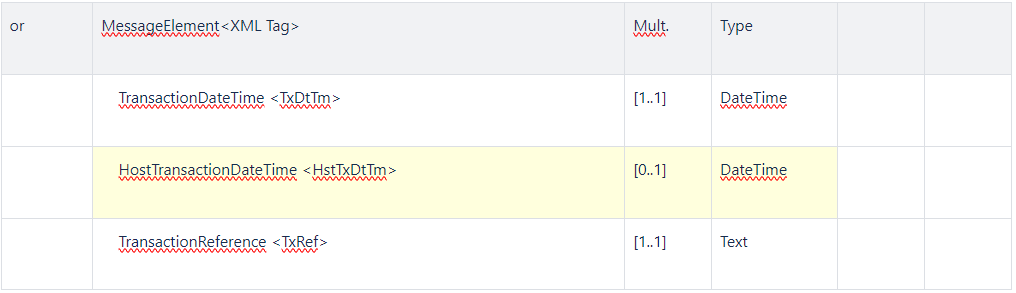
The proposal is to add to <Incdnt> the following values (i.e. a change to FailureReason7Code):



1. Add Support for HostTransactionDateTime

The Transaction Identification <TxId> contains a TransactionDateTime <TxDtTm> element which identifies the local time for the transaction assigned by the ATM. The terminal clock may not be reliable in some circumstances, and some prefer to use a date time assigned by the server, which is more reliable. The proposal is to add a HostTransactionrDateTime <TxSrvDtTm> element to the TransactionIdentifier <TxId> element as an optional element. This element would optionally be populated in a Response message, and subsequently could be populated within all CompletionAdvice message components and CompletionAcknowledgement message components.

The proposal then is to add a new element to <TxId> (i.e. change the TransactionIdentifier1):



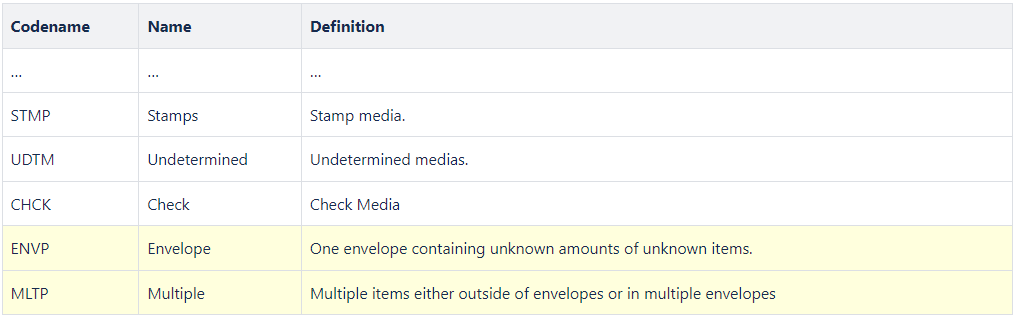
1. Add Support for Cheques, Envelope & Multiple Deposits in Deposit and Reconciliation Messages

The ATMDepositRequest has a MediaType which includes a code list value for cheques. However, the MediaType does not support envelope deposit or multiple deposit (i.e. multiple items outside an envelope or in multiple envelopes). Additionally, the ATMReconciliationAdvice message does not support cheques, envelopes or multiple deposits. The following details suggestions how to support these.

## 7.1 Add Support for Envelope & Multiple Deposits in the ATMDepositRequest Message

In the ATMDepositRequest, ATMDepositResponse & ATMDepositCompletionAdvice (both within DepositedMedia <DpstdMdia> and ToBeReconciledMediaCounters <ToBeRcncldMdiaCntrs>) the MediaType <MdiaTp> can be extended to include the missing code set values.

ATMMediaType2Code should become ATMMediaType3Code:



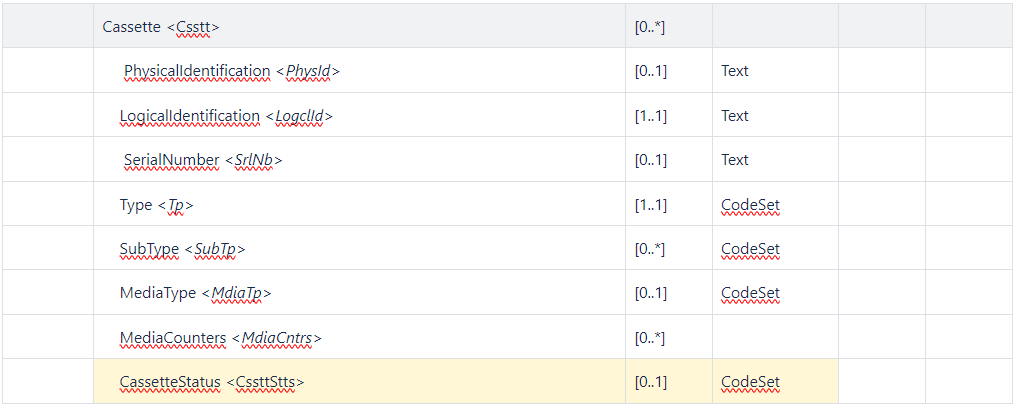
## 7.2 Add Support for Cheque, Envelope & Multiple Deposits in the ATMReconciliationAdvice Message

In the ATMReconciliationAdvice message the MediaType within the Cassette element should be changed to use the new MediaType defined (i.e ATMMediaType1Code should become ATMMediaType3Code).

1. Add Support for Cash Dispenser/Deposit Cassette Status

There are three places where cassette information is provided to a host system in the existing messages. These are in the: Cash Withdrawal Completion Advice/Acknowledgement, the Deposit Completion Advice/Acknowledgement and the Reconciliation Advice/Acknowledgement. The proposal is to add a new simple element <CssttStts> to the Cassette <Csstt> element contained within each of these messages.

Cassette <Csstt> to contain the following ATMCassette3:



Where Status is defined:

CassetteStatus < CssttStts >

Presence: [0..1]

Definition: The status of the cassette is as reported by CEN XFS, which can be one of: OK, Full, High, Low, Empty, Inoperative, Missing, NotAvailable, NoReference or Manipulated. This element may not be provided if the status has not changed from the previous time the cassette information has been provided.

Datatype: CassetteStatus1Code

|  |  |  |
| --- | --- | --- |
| CodeName | Name | Definition |
| CUOK | OK | The cash unit is in a good state. |
| CUFL | Full | The cash unit is full. |
| CUHG | High | The cash unit is high, that is it has reached or exceeded the threshold defined for this cassette. |
| CULW | Low | The cash unit is low, that is it has reached or is below the threshold defined for this cassette. |
| CUMT | Empty | The cash unit is empty. |
| CUNP | Inoperative | The cash unit is inoperative. |
| CUMS | Missing | The cash unit is missing. |
| CUNA | NotAvailable | The values of this cash unit are not available. |
| CUNR | NoReference | There are no reference values available for this cash unit. |
| CUMP | Manipulated | The device has been inserted when the device was not in an exchange state. |

1. Add Support for Replenishment Swap Operation

Probably the most common replenishment operation at an ATM is the cash swap. This will involve removing existing cassettes, and replacing these with a new set of cassettes. The existing ATMReconciliationAdvice message does not allow us to create a single message for this operation; it would allow us to create a message to Unload a cassette, and another to Load a cassette (see the TypeOfOperation <TpOfOpr> element within Transaction <Tx>). It would be preferable to have a single message, and to enable this the proposal is to add an Operation <Opr> element which allows one to have multiple operations under this.

Transaction <Tx> to contain the following ATMTransactionXX:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr.  No. | Page |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | ReconciliationOperation <ReconOpr> | [0..\*] |  |  |  |

where ReconciliationOperation<ReconOpr> contains the following ReconOpr1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ReconciliationOperation <ReconOpr> | [0..\*] |  |  |  |
|  | TypeOfOperation <TpOfOpr> | [0..1] | CodeSet |  |  |
|  | TransactionIdentification <TxId> | [1..1] |  |  |  |
|  | ReconciliationIdentification <RcncltnId> | [1..1] | Text |  |  |
|  | ATMTotals <ATMTtls> | [0..\*] |  |  |  |
|  | Cassette <Csstt> | [0..\*] |  |  |  |

The following shows an example message where a single logical type of cassette with 12 notes is replaced with a new logical cassette with 1000 notes.

##### **Example ATMRcncltnAdv**

1. <?xml version="1.0" encoding="UTF-8"?>
2. <n1:Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:n1="urn:iso:std:iso:20022:tech:xsd:caam.009.001.02" xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:caam.009.001.02 caam.009.001.02.xsd">
3. <n1:ATMRcncltnAdvc>
4. <n1:Hdr>
5. <n1:MsgFctn>
6. <n1:Fctn>TMOP</n1:Fctn>
7. </n1:MsgFctn>
8. <n1:PrtcolVrsn>1</n1:PrtcolVrsn>
9. <n1:XchgId>1</n1:XchgId>
10. <n1:CreDtTm>2001-12-17T09:30:47Z</n1:CreDtTm>
11. <n1:InitgPty>a</n1:InitgPty>
12. </n1:Hdr>
13. <n1:ATMRcncltnAdvc>
14. <n1:Envt>
15. <n1:ATM>
16. <n1:Id>123</n1:Id>
17. <n1:BaseCcy>USD</n1:BaseCcy>
18. </n1:ATM>
19. </n1:Envt>
20. <n1:Tx>
21. <n1:ReconOpr>
22. <n1:TpOfOpr>UNLD</n1:TpOfOpr>
23. <n1:TxId>
24. <n1:TxDtTm>2001-12-17T09:30:47Z</n1:TxDtTm>
25. <n1:TxRef>1</n1:TxRef>
26. </n1:TxId>
27. <n1:RcncltnId>1</n1:RcncltnId>
28. <n1:ATMTtls>
29. <n1:MdiaTp>NOTE</n1:MdiaTp>
30. <n1:ATMBal>0</n1:ATMBal>
31. <n1:ATMCur>0</n1:ATMCur>
32. </n1:ATMTtls>
33. <n1:Csstt>
34. <n1:LogclId>A</n1:LogclId>
35. <n1:Tp>DISP</n1:Tp>
36. <n1:MdiaTp>NOTE</n1:MdiaTp>
37. <n1:MdiaCntrs>
38. <n1:UnitVal>20</n1:UnitVal>
39. <n1:MdiaCtgy>FITN</n1:MdiaCtgy>
40. <n1:CurNb>0</n1:CurNb>
41. <n1:CurAmt>0</n1:CurAmt>
42. <n1:FlowTtls>
43. <n1:Tp>PRTN</n1:Tp>
44. <n1:RmvdNb>12</n1:RmvdNb>
45. </n1:FlowTtls>
46. </n1:MdiaCntrs>
47. </n1:Csstt>
48. </n1:ReconOpr>
49. <n1:ReconOpr>
50. <n1:TpOfOpr>LOAD</n1:TpOfOpr>
51. <n1:TxId>
52. <n1:TxDtTm>2001-12-17T09:30:47Z</n1:TxDtTm>
53. <n1:TxRef>1</n1:TxRef>
54. </n1:TxId>
55. <n1:RcncltnId>1</n1:RcncltnId>
56. <n1:ATMTtls>
57. <n1:MdiaTp>NOTE</n1:MdiaTp>
58. <n1:ATMBal>20000</n1:ATMBal>
59. <n1:ATMCur>20000</n1:ATMCur>
60. </n1:ATMTtls>
61. <n1:Csstt>
62. <n1:LogclId>A</n1:LogclId>
63. <n1:Tp>DISP</n1:Tp>
64. <n1:MdiaTp>NOTE</n1:MdiaTp>
65. <n1:MdiaCntrs>
66. <n1:UnitVal>20</n1:UnitVal>
67. <n1:CurNb>1000</n1:CurNb>
68. <n1:CurAmt>20000</n1:CurAmt>
69. <n1:FlowTtls>
70. <n1:Tp>OPER</n1:Tp>
71. <n1:AddedNb>1000</n1:AddedNb>
72. </n1:FlowTtls>
73. </n1:MdiaCntrs>
74. </n1:Csstt>
75. </n1:ReconOpr>
76. </n1:Tx>
77. </n1:ATMRcncltnAdvc>
78. </n1:ATMRcncltnAdvc>
79. </n1:Document>
80. Add a new Type of FlowTotals Operation & Improve Descriptions

The types <Tp> of FlowTotals <FlowTtls> has the values defined below:

|  |  |  |
| --- | --- | --- |
| **CodeName** | **Name** | **Definition** |
| INQU | CountersInquiry | Counters since the last counters inquiry. |
| CTXN | CustomerTransaction | Counters for a customer transaction. |
| CTOF | CutOff | Counters since the last cut-off. |
| BDAY | BusinessDay | Counters since the beginning of the business day. |
| PRTN | Operation | Counters for an operation performed by  an operator on the ATM. |
| OPER | OperatorAdjust | Counters since the last update by an operator on the ATM. |

However, none of these would be used if one wanted to send the counters since the last replenishment within either a CompletionAdvice or a ReconciliationAdvice message. This proposes we add a new Type codeset value as shown below:

|  |  |  |
| --- | --- | --- |
| **CodeName** | **Name** | **Definition** |
| SLRP | SinceLastReplenishmentPoint | Counters since last replenishment point. |

The following shows an example message element from a CompletionAdvice showing the consumer transaction had caused 4 notes to be presented, and the total number of presented notes since the last replenishment point to be 800:

##### **Example Cassette structure**



Additionally, the existing types PRTN and OPER are very confusing. The descriptions are:

|  |  |  |
| --- | --- | --- |
| **CodeName** | **Name** | **Definition** |
| PRTN | Operation | Counters for an operation performed by an operator on the ATM. |
| OPER | OperatorAdjust | Counters since the last update by an operator on the ATM. |

The two use cases covered by these are:

1. where a replenishment occurs where media is moved in or out of a cash unit (e.g. an operator adds cash),
2. where an operator counts the cash in a cassette and adjusts the count for this cassette. No media is moved in or out of the cash unit.

The proposal is to change the codesets to better reflect these semantics to the following:

|  |  |  |
| --- | --- | --- |
| **CodeName** | **Name** | **Definition** |
| OPER | Operation | Counters as a result of some replenishment operation carried out by an operator (e.g. adding cash to a cash unit). |
| OPAD | OperatorAdjust | Counters as a result of an operator adjusting the counts without removing or adding media (also known as a ‘balance set’). |

Based on this the updated codeset would become:

|  |  |  |
| --- | --- | --- |
| **CodeName** | **Name** | **Definition** |
| INQU | CountersInquiry | Counters since the last counters inquiry. |
| CTXN | CustomerTransaction | Counters for a customer transaction. |
| CTOF | CutOff | Counters since the last cut-off. |
| BDAY | BusinessDay | Counters since the beginning of the business day. |
| OPER | Operation | Counters as a result of some replenishment operation carried out by an operator (e.g. adding cash to a cash unit). |
| OPAD | OperatorAdjust | Counters as a result of an operator adjusting the counts without removing or adding media (also known as a ‘balance set’). |
| SLRP | SinceLastReplenishmentPoint | Counters since last replenishment point. |

1. Add a Reason to ATMGlobalStatus

When an ATM reports itself to be out of service, it would be very useful for the ATM Manager to know the reason the ATM is out of service. This can be achieved by adding an optional CurrentStatusReason <CurStsRsn> element to ATMGlobalStatus <ATMGblSts>, as shown below:

*Presence:* [1..1]

*Definition:* Global status of the ATM.

**ATMGlobalStatus <ATMGblSts>** contains the following **ATMStatus1** elements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| or | **MessageElement*<XML Tag>*** | **Mult.** | **Type** | **Constr. No.** | **Page** |
|  | **CurrentStatus** *<CurSts>* | [1..1] | CodeSet |  |  |
|  | **CurrentStatusReason** <CurStsRsn> | [0..\*] | Text |  |  |
|  | **DemandedStatus** *<DmnddSts>* | [0..1] | CodeSet |  |  |

Where CurrentStatusReason <CurStsRsn> is defined as:

*Presence:* [0..1]

*Definition:* A description of why the ATM is in this status. This would typically be populated with a value when CurrentStatus is OutOfService (OUTS).

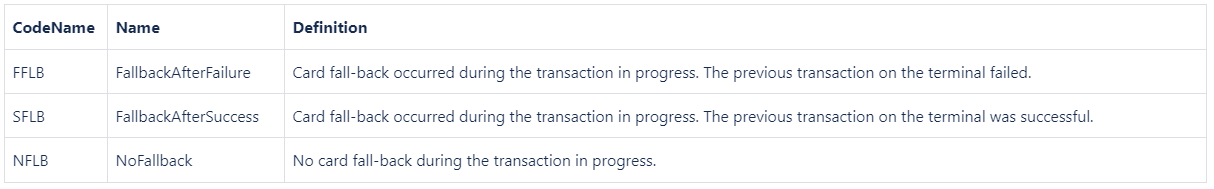
*Datatype*: "Max70Text"

1. Redesign FallbackIndicator

An EMV Fallback transaction is indicated by the usage of the following FallbackIndicator <FllbckInd> element:

*Presence:* [0..1]

*Definition:* Indicate the occurrence of a fall-back on the card entry mode.



It would appear the definition of this has been copied from the ATICA definition, which in turn has been copied from existing ISO 8583 acquirer to issuer message implementations which conform to VISA and Mastercard message requirements.

However, from the ATM client point of view this level of information is not commonly provided by an ATM. It is more common for ATM’s to simply indicate if fallback has occurred (i.e. a Boolean). The proposal therefor is to simplify this element to be:

*Presence:* [0..1]

*Definition:* Indicate the occurrence of a fall-back on the card entry mode.

*Datatype:* One of the following values must be used (see "TrueFalseIndicator" on page XXXX):

* *Meaning When True:* Fallback is used.
* *Meaning When False:* No fallback is used.

1. Corrections to Deposit messages

The Deposit messages were intended to allow one to deposit into one or more accounts. We can see this in the fact an ATMDepositRequest message contains an AccountSequenceNumber [0..1] within a DetailedRequestedAmount [0..\*] element. However, the AccountData element has a cardinality of [0..1]. This should be [0..\*] to enable multiple accounts to be defined, so that later within the message where the DetailedRequestedAmount and the DepositedMedia is defined, these can reference accounts defined within AccountData.

The message structure where AccountData is defined within the Transaction element, and then referenced within the DetailedRequestedAmount and DepositedMedia by using a sequence number. This is not an ideal structure. A better structure is to have a sub-element of the Transaction which contains the AccountData, the DetailedRequestedAmount and DepositedMedia all within the one element, and if there are multiple accounts to deposit into, then we can have multiple such sub-elements. The following describes the changes which will be required to enable this structure within the ATMDepositRequest, the the ATMDepositResponse and the ATMDepositCompletionAdvice.

The following details the changes required:

Transaction<Tx> of type ATMTransaction15 (used within the ATMDepositRequest):

|  |  |  |
| --- | --- | --- |
| MessageElement<XML Tag> | Mult. | Type |
| Transaction <Tx> | [1..1] |  |
| TransactionIdentification <TxId> | [1..1] |  |
| ReconciliationIdentification <RcncltnId> | [0..1] | Text |
| TotalRequestedAmount <TtlReqAmt> | [0..1] | Amount |
| SubDeposit <SubDpst> | [0..\*] | + |
| SubDepositIdentification <SubDpstId> | [0..1] | Text |
| AccountData <AcctData> | [0..\*] | + |
| DetailedRequestedAmount <DtldReqdAmt> | [0..\*] | + |
| DepositedMedia <DpstdMdia> | [0..\*] | + |
| RequestedReceipt <ReqdRct> | [0..1] | Indicator |
| ICCRelatedData <ICCRltdData> | [0..1] | Binary |

Transaction<Tx> of type ATMTransaction16 (used within the ATMDepositResponse):

|  |  |  |
| --- | --- | --- |
| MessageElement<XML Tag> | Mult. | Type |
| Transaction <Tx> | [1..1] |  |
| TransactionIdentification <TxId> | [1..1] |  |
| ReconciliationIdentification <RcncltnId | [0..1] | Text |
| CompletionRequired<CmpltnReqrd> | [0..1] | Amount |
| TotalAuthoriseddAmount <TtlAuthrsdAmt> | [1..1] | + |
| TotalRequestedAmount <TtlReqAmt> | [0..1] | Amount |
| SubDeposit <SubDpst> | [0..\*] | + |
| SubDepositIdentification <SubDpstId> | [0..1] | Text |
| AccountData <AcctData> | [0..\*] | + |
| DetailedRequestedAmount <DtldReqdAmt> | [0..\*] | + |
| AdditionalCharge <AddtlChrg> | [0..\*] | + |

Transaction<Tx> of type ATMTransaction19 (used within the ATMDepositCompletionAdvice):

|  |  |  |
| --- | --- | --- |
| MessageElement<XML Tag> | Mult. | Type |
| Transaction <Tx> | [1..1] |  |
| TransactionIdentification <TxId> | [1..1] |  |
| TransactionStatus<TxSts> | [1..1] | CodeSet |
| ReconciliationIdentification <RcncltnId> | [0..1] | Text |
| Incident <Incdnt> | [0..\*] | Codeset |
| IncidentDetail <IncdntDtl> | [0..\*] | Text |
| TotalDepositedAmount <TtlDpstdAmt> | [1..1] | + |
| TotalAuthoriseddAmount <TtlAuthrsdAmt> | [1..1] | + |
| TotalRequestedAmount <TtlReqAmt> | [0..1] | Amount |
| SubDeposit <SubDpst> | [0..\*] | + |
| SubDepositIdentification <SubDpstId> | [0..1] | Text |
| AccountData <AcctData> | [0..\*] | + |
| DetailedRequestedAmount <DtldReqdAmt> | [0..\*] | + |
| DepositedMedia <DpstdMdia> | [0..\*] | + |
| AdditionalCharge <AddtlChrg> | [0..\*] | + |

1. Add AccountMatchesMultipleAccounts ResponseReason

In an authorisation request it is possible to send a SelectedAccountType (e.g. Savings) with no AccountIdentifier, and for there to be multiple accounts which match this account type. In this case the response message needs to indicate the authorisation has been declined due to multiple accounts having matched the account type selected. I don’t believe there is a ResponseReason which would fit this, and hence I propose to add a new value to ResultDetail4Code codeset as suggested:

|  |  |  |
| --- | --- | --- |
| **Codename** | **Name** | **Definition** |
| AMMA | AccountMatchesMultipleAccounts | The account details provided in the request message matches multiple accounts. |
| … |  |  |

1. Support Multi-Currency Deposit

The existing Deposit messages only support a single currency (i.e. the TotalAmount element is defined with multiplicity [0..1]). To enable multiple-currency support the following changes are required:

15.1 ATMDepositRequest changes

The ATMDepositRequest TotalAmount needs to change its multiplicity to be 0..\*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | TotalAmount <TtlAmt> | [0..\*] |  |  |  |

15.2 ATMDepositResponse changes

The ATMDepositResponse needs to change the TotalAuthorisedAmount  to be 1..\*, and on TotalRequestedAmount to be 0..\*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | TotalAuthorisedAmount  <TtlAuthrsdAmt> | [1..\*] |  |  |  |
|  | TotalRequestedAmount <TtlReqAmt> | [0..\*] |  |  |  |

15.3 ATMDepositCompletionAdvice changes

The ATMDepositCompletionAdvice needs to change the multiplicity of TotalDepositedAmount and TotalAuthorisedAmount to be 1..\* and TotalRequestedAmount to be 0..\*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | TotalDepositedAmount <TtlDepstdAmt> | [1..\*] |  |  |  |
|  | TotalAuthorisedAmount  <TtlAuthrsdAmt> | [1..\*] |  |  |  |
|  | TotalRequestedAmount <TtlReqAmt> | [0..\*] |  |  |  |

1. Add Account Information to ATMWithdrawalResponse, ATMDepositResponse, and ATMTransferResponse messages

For the use case where a consumer selects an account type (say ‘savings’), the issuer may decline the authorisation if there are more than one account associated with the card used. In this case, instead of requiring a separate ATMInquiryRequest to get all the accounts associated with the card, it would be preferable if the authorisation response returned the accounts which the consumer must select from. This will make the message exchange more efficient.

The proposal is to add the <AcctInf> element within the <Tx> element which would contain the account information that matches the account type selected by the consumer. The following highlighted in yellow would be the change to the ATMWithdrawalResponse:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMWithdrawalResponse <ATMWdrwRspn> | [0..1] |  |  |  |
|  | Environment <Envt> | [1..1] |  |  |  |
|  | Context <Cntxt> | [1..1] |  |  |  |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | TransactionIdentification <TxId> | [1..1] |  |  |  |
|  | ReconciliationIdentification <RcncltnId> | [0..1] |  |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [1..1] |  |  |  |
|  | AccountInformation <AcctInf> | [0..\*] |  |  |  |
|  | … |  |  |  |  |

The intent would be if the consumer has selected an account type (e.g. savings) and there is more than one account of this type associated with the card, the response would return in the <AcctInf> element all the accounts which were of this type (that is it will not return all accounts associated with the card, only the ones which match the selection by the consumer).

The same new element would be added to the ATMDepositResponse:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMDepositResponse <ATMDpstRspn> | [0..1] |  |  |  |
|  | Environment <Envt> | [1..1] |  |  |  |
|  | Context <Cntxt> | [1..1] |  |  |  |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | TransactionIdentification <TxId> | [1..1] |  |  |  |
|  | ReconciliationIdentification <RcncltnId> | [0..1] |  |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [1..1] |  |  |  |
|  | AccountInformation <AcctInf> | [0..\*] |  |  |  |
|  | ... |  |  |  |  |

For the ATMTransferResponse:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMTransferResponse <ATMTrfRspn> | [0..1] |  |  |  |
|  | Environment <Envt> | [1..1] |  |  |  |
|  | Context <Cntxt> | [1..1] |  |  |  |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | TransactionIdentification <TxId> | [1..1] |  |  |  |
|  | ReconciliationIdentification <RcncltnId> | [0..1] |  |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [1..1] |  |  |  |
|  | AccountInformation <AcctInf> | [0..\*] |  |  |  |
|  | … |  |  |  |  |

Note in the transfer, the AccountFrom and/or the AccountTo account type may be associated with more than one account. The expectation is the response message will only return the list of accounts of whichever has multiple accounts of that type (it could be for both types).

ATMInquiryResponse does not need to be updated as it supports the AccountInformation element already.

1. Add Fee Information to ATMWithdrawalResponse, ATMDepositResponse, ATMInquiryResponse and ATMTransferResponse messages

This addresses the use case where a consumer selects a cash withdrawal and the acquirer or issuer declines the authorisation due to fees which the consumer must agree to. In this case, instead of requiring a separate ATMInquiryRequest to get fees which the consumer must agree to, it would be preferable if the authorisation response returned the corresponding fees. This will make the message exchange more efficient.

The proposal for the ATMWithdrawalResponse is to include a FeeToAdd as described below (change highlighted in yellow):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMWithdrawalResponse <ATMWdlRspn> | [0..1] |  |  |  |
|  | Environment <Envt> | [1..1] |  |  |  |
|  | Context <Cntxt> | [1..1] |  |  |  |
|  | Transaction <Tx> | [1..1] |  |  |  |
|  | TransactionIdentification <TxId> | [1..1] |  |  |  |
|  | ReconciliationIdentification <RcncltnId> | [0..1] |  |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [1..1] |  |  |  |
|  | AuthorisationEntity <AuthstnNtty> | [0..1] |  |  |  |
|  | AuthorisationResponse <AuthstnRspn> | [1..1] |  |  |  |
|  | ... |  |  |  |  |
|  | Action <Actn> | [0..\*] |  |  |  |
|  | FeeToAdd <FeeToAdd> | [0..\*] | ± |  |  |
|  | Amount <Amt> | [1..1] | Amount |  |  |
|  | Currency <Ccy> | [0..1] | Codeset |  |  |
|  | FeeLabel <FeeLabl> | [0..1] | Text |  |  |

The equivalent proposal for the ATMDepositResponse is to include a FeeToAdd as described below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMDepositResponse <ATMDpstRspn> | [0..1] | ± |  |  |
|  | Environment <Envt> | [1..1] | ± |  |  |
|  | Context <Cntxt> | [1..1] | ± |  |  |
|  | Transaction <Tx> | [1..1] | ± |  |  |
|  | TransactionIdentification <TxId> | [1..1] | Text |  |  |
|  | ReconciliationIdentification <RcncltnId> | [0..1] | Text |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [1..1] | ± |  |  |
|  | AuthorisationEntity <AuthstnNtty> | [0..1] |  |  |  |
|  | AuthorisationResponse <AuthstnRspn> | [1..1] |  |  |  |
|  | ... |  |  |  |  |
|  | Action <Actn> | [0..\*] |  |  |  |
|  | FeeToAdd <FeeToAdd> | [0..\*] | ± |  |  |
|  | Amount <Amt> | [1..1] | Amount |  |  |
|  | Currency <Ccy> | [0..1] | CodeSet |  |  |
|  | FeeLabel <FeeLabl> | [0..1] | Text |  |  |

The equivalent proposal for the ATMDepositResponse is to include a FeeToAdd as described below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMDepositResponse <ATMDpstRspn> | [0..1] | ± |  |  |
|  | Environment <Envt> | [1..1] | ± |  |  |
|  | Context <Cntxt> | [1..1] | ± |  |  |
|  | Transaction <Tx> | [1..1] | ± |  |  |
|  | TransactionIdentification <TxId> | [1..1] | Text |  |  |
|  | ReconciliationIdentification <RcncltnId> | [0..1] | Text |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [1..1] | ± |  |  |
|  | AuthorisationEntity <AuthstnNtty> | [0..1] |  |  |  |
|  | AuthorisationResponse <AuthstnRspn> | [1..1] |  |  |  |
|  | ... |  |  |  |  |
|  | Action <Actn> | [0..\*] |  |  |  |
|  | FeeToAdd <FeeToAdd> | [0..\*] | ± |  |  |
|  | Amount <Amt> | [1..1] | Amount |  |  |
|  | Currency <Ccy> | [0..1] | CodeSet |  |  |
|  | FeeLabel <FeeLabl> | [0..1] | Text |  |  |

The equivalent proposal for the ATMTransferResponse is to include a FeeToAdd as described below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMTransferResponse <ATMTrfRspn> | [0..1] | ± |  |  |
|  | Environment <Envt> | [1..1] | ± |  |  |
|  | Context <Cntxt> | [1..1] | ± |  |  |
|  | Transaction <Tx> | [1..1] | ± |  |  |
|  | TransactionIdentification <TxId> | [1..1] | Text |  |  |
|  | ReconciliationIdentification <RcncltnId> | [0..1] | Text |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [1..1] | ± |  |  |
|  | AuthorisationEntity <AuthstnNtty> | [0..1] |  |  |  |
|  | AuthorisationResponse <AuthstnRspn> | [1..1] |  |  |  |
|  | ... |  |  |  |  |
|  | Action <Actn> | [0..\*] |  |  |  |
|  | FeeToAdd <FeeToAdd> | [0..\*] | ± |  |  |
|  | Amount <Amt> | [1..1] | Amount |  |  |
|  | Currency <Ccy> | [0..1] | CodeSet |  |  |
|  | FeeLabel <FeeLabl> | [0..1] | Text |  |  |

The equivalent proposal for the ATMInquiryResponse is to include a FeeToAdd as described below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
|  | ATMInquiryResponse <ATMNqryRspn> | [[0..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=0..1&linkCreation=true&fromPageId=279412853)] | ± |  |  |
|  | Environment <Envt> | [[1..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)] | ± |  |  |
|  | Context <Cntxt> | [[1..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)] | ± |  |  |
|  | Transaction <Tx> | [[1..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)] | ± |  |  |
|  | TransactionIdentification <TxId> | [[1..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)] | Text |  |  |
|  | ReconciliationIdentification <RcncltnId> | [[0..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=0..1&linkCreation=true&fromPageId=279412853)] | Text |  |  |
|  | ... |  |  |  |  |
|  | AuthorisationResult <AuthstnRslt> | [[1..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)] | ± |  |  |
|  | AuthorisationEntity <AuthstnNtty> | [[0..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=0..1&linkCreation=true&fromPageId=279412853)] |  |  |  |
|  | AuthorisationResponse <AuthstnRspn> | [[1..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)] |  |  |  |
|  | ... |  |  |  |  |
|  | Action <Actn> | [0[..](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)\*] |  |  |  |
|  | FeeToAdd <FeeToAdd> | [0[..](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)\*] | ± |  |  |
|  | Amount <Amt> | [[1..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=1..1&linkCreation=true&fromPageId=279412853)] | Amount |  |  |
|  | Currency <Ccy> | [[0..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=0..1&linkCreation=true&fromPageId=279412853)] | CodeSet |  |  |
|  | FeeLabel <FeeLabl> | [[0..1](https://nexo-standards.atlassian.net/wiki/pages/createpage.action?spaceKey=EPASATM&title=0..1&linkCreation=true&fromPageId=279412853)] | Text |  |  |

Note this change assumes the AuthorisationResult element has been added to the ATMInquiryResponse message (i.e. a separate change request).

1. Extend the AccountIdentifier to include email and mobile phone number

It is now possible to transfer funds into an account identified by an email address or a mobile phone number (MSISDN). Examples of usage are PayPal and Zelle. The proposal is to reuse the same definitions as the CAPE specification -which already has these defined. This requires the AccountIdentifier <AcctIdr> element to be extended as following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Or | MessageElement<XML Tag> | Mult. | Type | Constr. No. | Page |
| {Or | IBAN <IBAN> | [1..1] | IdentifierSet |  |  |
| Or | BBAN <BBAN> | [1..1] | IdentifierSet |  |  |
| Or | UPIC <UPIC> | [1..1] | IdentifierSet |  |  |
| Or | Domestic <Dmst> | [1..1] | ± |  |  |
| Or | MSISDN <MSISDN> | [1..1] | Text |  |  |
| Or} | EMail <EMail> | [1..1] | Text |  |  |
|  |  |  |  |  |  |

Where MSISDN is defined:

MSISDN <MSISDN>

*Presence*: [1..1]

*Definition*: Mobile Subscriber Integrated Service Digital Network (i.e. mobile phone number of the SIM card).

*DataType*: Max16Text

and EMail is defined as:

EMail <Email>

*Presence*: [1..1]

*Definition*: E-mail identifying the account.

*DataType*: Max256Text.

1. Account Download Code

The existing ATMInquiryRequest defines within its ServiceType a list of possible data to request from an acquiring server. However this codeset does not define a value for downloading the set of accounts associated with the consumer card. This proposal adds a new codeset value which defines such an option.

ServiceType <SvcTp> is defined as:

Presence: [1..1]

Definition: Describes the type of inquiry selected by the customer or the ATM.

DataType: CodeSet ATMServiceType11Code

|  |  |  |
| --- | --- | --- |
| CodeName | Name | Definition |
| ASTS | AccountStatements | Ask for account statement information to  a related customer account. |
| ... |  |  |
| CMPF | CustomerProfile | Ask for customer profile with eventually related account information. |
| ACCD | AccountDownload | Download accounts associated to the customer card. |

1. Support Interactive Transactions

There are transactions which require some interactions between the consumer and the acquiring server. Examples are transactions in which DCC, fees, multiple account selection is involved, etc.

The proposed way to achieve this is where a transaction returns with a Response which states this transaction requires some additional information from the consumer, and if provided the transaction can be continued. In this example if the consumer agrees to the fees, another transaction request can be sent which has the same transaction ID as the previous one (i.e. as far as the acquiring server is concerned a single transaction has been executed).

At present the MUG does not describe how such a scenario could be handled. No status code exists that allows the acquiring server to inform the terminal that the transaction authorisation can be continued if the further consumer input is provided.

The proposal is to add a new Response code of ‘ApprovalSuspended’ meaning that the consumer must provide further information to continue with the approval process.

The following change needs to be made:

*Response* <Rspn>

*Presence*: [1..1]

*Definition*: Result of the requested transaction.

*Datatype*: Response5Code

|  |  |  |
| --- | --- | --- |
| CodeName | Name | Definition |
| APPR | Approved | Service has been successfully provided. |
| DECL | Declined | Service is declined. |
| PART | PartialApproved | Service has been partially provided |
| SUSP | ApprovalSuspended | To provide this service additional information needs to be provided by the customer. |

The MUG should be updated to document the advocated way to handle this functionality.

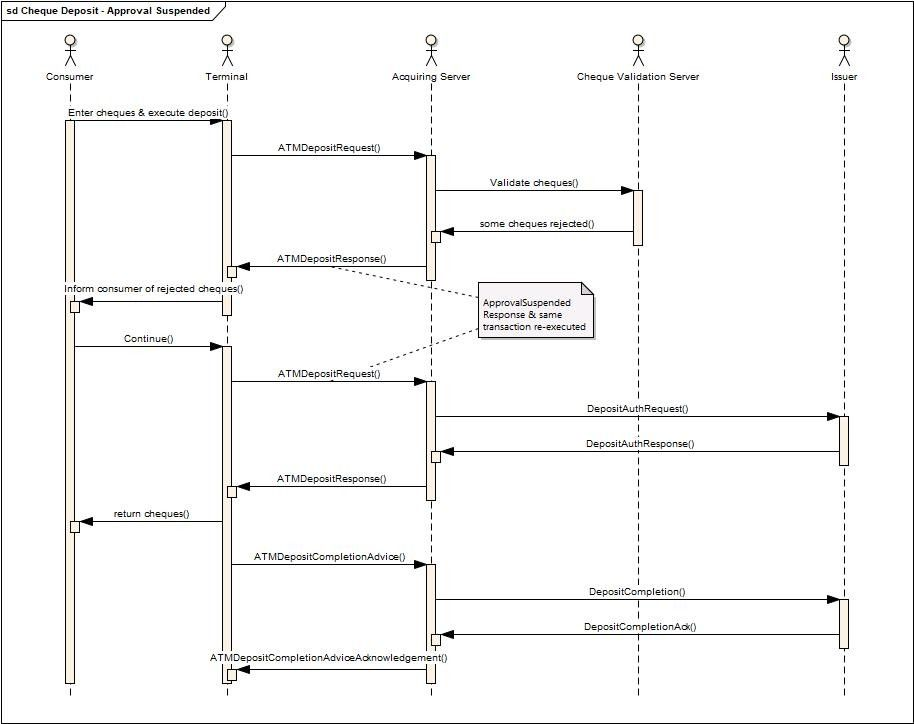
The proposal is to extend the ResponseReason <RspnRsn> element codeset ResultDetail4Code to include the additional code values:

|  |  |  |
| --- | --- | --- |
| CodeName | Name | Definition |
| AGFE | AgreeFee | Consumer must agree fee from those provided to authorise the transaction. See <Tx><AuthstnRslt><FeeToAdd>. |
| SACT | SelectAccount | Consumer must select an account from those provided to authorise the transaction. See <Tx><AcctInf>. |
| ADCC | AgreeDCC | Consumer must choose whether to accept DCC offer provided to authorise the transaction. See <Tx><CcyConvsElgblty>. |
| OTHR | Other | For reason see the AdditionalResponseInformation element. This is used in cases where the reason is not one of those defined by this codeset. |

1. Add Individual Check Authorisation Information to the ATMDepositResponse message

The existing ATMDepositResponse message returns a Response of ‘Approved’, ‘Declined’ or ‘PartialApproved’. However, it does not cater for a use case where some checks may be accepted and some rejected (that is the information of which checks are accepted/rejected cannot be included in the response message). In this situation a consumer will be informed of the checks which have been approved, return those that have not, and request the consumer agrees to making the deposit of the accepted checks .

The following sequence diagram shows a check deposit transaction where some checks are rejected and returned to the consumer, and the consumer decides to continue with the deposit transaction. The following proposal assumes another proposal to support an ‘ApprovalSuspended’ Response is supported.



The sequence of events for the transaction are as follows:

* The consumer enters multiple cheques and executes the deposit transaction.
* On receiving the ATMDepositRequest message the acquiring server checks with a cheque validation server if the cheques are to be accepted.
* The cheque validation server responds with a subset of the cheques being rejected.
* The acquiring server sends an ATMDepositResponse message to the Terminal with a Response of ApprovalSuspended and listing those cheques which have been accepted and those which have been rejected.
* The consumer agrees to continue with the transaction.
* To continue the same transaction an ATMDepositRequest is sent from the Terminal to the Acquiring Server.
* The acquiring server then sends a message to the host to authorise the subset of cheques which have been accepted.
* The host responds with an authorised response.
* The acquiring server sends an ATMDepositResponse message to the Terminal with a Response of Approved.
* The rejected cheques are returned to the consumer.
* The Terminal sends an ATMDepositCompletionAdvice message to the acquiring server.
* The acquiring server informs the host of the transaction outcome and responds to the Terminal with an ATMDepositCompletionAcknowledgement message.

There are possible alternatives on this message exchange. Two of them could be:

1. Instead of returning an ‘ApprovalSuspended’ Response, return a ‘PartialApproved’ with the information of what cheques have been rejected and those accepted. In this case the Terminal has the option to request if the consumer agrees. This is followed by an ATMDepositCompletionAdvice message to inform the acquiring server of the outcome (i.e. continue or cancel).
2. When the cheque validation server returns denoting only a subset of cheques are accepted, the acquiring server then returns a response of Decline to the Terminal. If the consumer decides to continue with the transaction, then a new ATMDepositRequest message is sent to the acquiring server. In this case a new transaction is initiated. The rest is as expected in a deposit transaction.

In any of these scenarios if the consumer decides not to continue with the transaction, then an ATMDepositCompletionAdvice message would be sent to the acquiring server with the TransactionStatus set to Failure and an Incident of CustomerCancel.

The following changes addresses these requirements. Note to correlate the cheques in the Request with those in the response we will add an ID for each cheque in the Request and the Response.

The proposed change to the ATMDepositResponse is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Or | Message Element | Mult. | Type |  |
|  | ATMDepositResponse <ATMDepstRspn> | [0..1] | ± |  |
|  | … |  |  |  |
|  | Transaction <Tx> | [1.1] | ± |  |
|  | … |  |  |  |
|  | DepositedMedia <DpstdMdia> | [0..\*] | ± |  |
|  | … |  |  |  |
|  | MediaItems <MdiaItms> | [1..\*] | ± |  |
|  | MediaID <MdiaID> | [0..1] | Text |  |
|  | Count <Cnt> | [0..1] | Quantity |  |
|  | … |  |  |  |
|  | MediaStatus <MdiaSts> | [0..1] | Codeset |  |

Where DepositedMedia is of a new type ATMDepositedMedia4 which includes a MediaItem of new type ATMDepositedMediaItems1. The ATMDepositedMediaItems1 type extends the existing ATMDepositedMedia2 with a MediaId, and MediaStatus elements as shown above.

 Where:

MediaID <MdiaID> is defined as:

*Presence*: [0..1]

*Definition*: Within a list of MediaItems uniquely identifies a specific item. Typically will be an integer index 0..n, where there are n Items in the list of MediaItems.

*DataType*: Text

MediaStatus <MdiaSts> is defined as:

*Presence*: [0..1]

*Definition*: Status of the media item.

*DataType*: CodeSet

|  |  |  |
| --- | --- | --- |
| CodeName | Name | Definition |
| ACPT | Accepted | Item (such as a check) has been accepted. |
| REJT | Rejected | Item (such as a check) has been rejected. |

The proposed change to the ATMDepositRequest is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Or | Message Element | Mult. | Type |  |
|  | ATMDepositRequest <ATMDepstReq> | [0..1] | ± |  |
|  | … |  |  |  |
|  | Transaction <Tx> | [1.1] | ± |  |
|  | … |  |  |  |
|  | DepositedMedia <DpstdMdia> | [0..\*] | ± |  |
|  | … |  |  |  |
|  | MediaItems <MdiaItms> | [1..\*] | ± |  |
|  | MediaID <MdiaID> | [0..1] | Text |  |
|  | Count <Cnt> | [0..1] | Quantity |  |
|  | … |  |  |  |
|  | MediaStatus <MdiaSts> | [0..1] | Codeset |  |

Where DepositedMedia <DpstdMdia> is of type ATMDepositedMedia3 defined previously. The MediaID, within MediaItems, may be populated in the request, but the MediaStatus will not be populated within the request. These latter element is intended to be populated optionally on the Response message if available.

No changes are required to the ATMDepositCompletionAdvice and the ATMDepositAcknowledgement messages.

1. Extend DCC for EU regulations

Based on “REGULATION (EC) No 924/2009 OF THE EUROPEAN PARLIAMENT”, “P8\_TA(2019)0124” and “2018/0076 (Article 3a)” payment service providers are required to increase transparency of DCC transactions to cardholders/consumers prior to the initiation of a payment transaction. This transparency has to be achieved by showing a “reference exchange rate” published by ECB. To allow this information to be shown on ATM the proposal is to extend existing type CurrencyConversionResult with a new element **ReferenceRate.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **or** | **MessageElement*<XML Tag>*** | Mult. | Type |  |
|  | CurrencyConversionResult |  |  |  |
|  | CurrencyConversionIdentification | [0..1] | Text |  |
|  | … |  |  |  |
|  | ReferenceRate <RefRate> | [0..1] |  |  |
|  | ExchangeRate <XchgRate> | [1..1] | Rate |  |
|  | PublishDate <PubDt> | [1..1] | DateTime |  |
|  | AddInfo <AddInfo> | [0..1] | Text |  |

Where:

ReferenceRate <RefRate> is defined as:

*Presence*: [0..1]

*Definition*: Reference exchange rate to be shown to the consumer. Requirement within the EC, see “REGULATION (EC) No 924/2009 OF THE EUROPEAN PARLIAMENT”, “P8\_TA(2019)0124” and “2018/0076 (Article 3a)”.

The ReferenceRate <RefRate> contains the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **or** | **MessageElement*<XML Tag>*** | Mult. | Type |  |
|  | ReferenceRate <RefRate> | [0..1] |  |  |
|  | ExchangeRate <XchgRate> | [1..1] | Rate |  |
|  | PublishDate <PubDt> | [1..1] | DateTime |  |
|  | AddInfo <AddInfo> | [0..1] | Text |  |

Where:

ExchangeRate <XchgRate> is defined as:

*Presence*: [1..1]

*Definition*: Exchange rate, expressed as a percentage, applied to convert the original amount into the resulting amount.

*DataType*: Rate

 PublishDate <PubDt> is defined as:

*Presence*: [1..1]

*Definition*: Date when exchange rate published.

*DataType*: DateTime

AddInfo <AddInfo> is defined as:

*Presence*: [0..1]

*Definition*: Additional information to be provided to the consumer related to the reference rate.

*DataType*: Text

1. Add Further Individual Check Authorisation Information to the ATMDepositResponse message

The following extends the proposal in section 21.

In a check deposit transaction, when an individual check is rejected, some Financial Institutions may want to provide an explanation of the reason why the item has been rejected to give that feedback to the cardholder.

When a check is accepted by the issuer, a FI may want to provide a check reference number for each check accepted for traceability reasons.

Potentially, some FI's may want to return other ad hoc data related to a check item, but which we cannot predict.

To cater for these three further data the following data elements will be added:

* RejectReason: a FI specific field with a textual description of the reason why the item was rejected.
* MediaItemReference: a string that uniquely identifies the media item in the FI's bank system.
* MediaItemAdditionalData: used to pass other information related to the media item but not modelled by the other properties. This is a key value pair that gives flexibility to model some ad hoc data a FI wishes to pass through to the ATM.

The proposed change to the ATMDepositResponse is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **or** | **MessageElement*<XML Tag>*** | Mult. | Type |  |
|  | ATMDepositResponse <ATMDepstRspn> | [0..1] | ± |  |
|  | … |  |  |  |
|  | Transaction <Tx> | [1..1] |  |  |
|  | … |  |  |  |
|  | DepositedMedia <DpstdMdia> | [0..\*] | ± |  |
|  | … |  |  |  |
|  | MediaItems <MdiaItms> | [1..\*] | ± |  |
|  | MediaID <MdiaID> | [0..1] | Text |  |
|  | Count <Cnt> | [0..1] | Quantity |  |
|  | … |  |  |  |
|  | MediaStatus <MdiaSts> | [0..1] | CodeSet |  |
|  | RejectedReason <RjctdRsn> | [0..1] | Text |  |
|  | MediaItemReference <MdiaItmRef> | [0..1] | Text |  |
|  | MediaItemAdditionalData <MdiaItemAddtlData> | [0..\*] | ± |  |
|  | Identification <Id> | [1..1] | Text |  |
|  | Value <Val> | [1..1] | Text |  |

where:

RejectedReason <RjctdRsn> is defined as:

*Presence*: [0..1]

*Definition*: The reason a media item (such as a check) has been rejected. This may be used to inform the cardholder.

*DataType*: Text

MediaItemReference <MdiaItmRef> is defined as:

*Presence*: [0..1]

*Definition*:  a string that uniquely identifies the media item in the FI's bank system. This may be used for traceability reasons.

*DataType*: Text

MediaItemAdditionalData <MdiaItemAddtlData> is defined as:

*Presence*: [0..\*]

*Definition*:  a key value pair which is associated with the media item. This provides flexibility to model some ad hoc data a FI wishes to pass through to the ATM.

*DataType*: Text

Identification <Id> is defined as:

*Presence*: [1..1]

*Definition*:  an identifier of the data item being passed to the ATM.

*DataType*: Text

Value<Val> is defined as:

*Presence*: [1..1]

*Definition*:  the value of the data item being passed to the ATM.

*DataType*: Text

1. Add Source and Destination Account Information to the ATMTransferResponse message

In 16, above, it was agreed we would add AccountInformation to various Response messages to enable account information to be returned to the terminal where a consumer is required to select a specific account to debit/credit. However, the AccountInformation element does not include information of whether the accounts returned are to be source or destination accounts. Some host system are able to provide this information, so we should change the ATMTransferResponse message to include this additional data.

The proposal is to add one new element to the AccountInformation (AcctInf) element which is in ATMTransferResponse:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Or | Message Element | Mult. | Type |  |
|  | AccountInformation<AcctInf> | [0..\*] | ± |  |
|  | AccountType <AcctTp> | [0..1] | codeset |  |
|  | AccountName<AcctNm> | [0..1] | text |  |
|  | AccountUsageCode<AcctUsgCd> | [0..1] | codeset |  |
|  | ... |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

where:

AccountUsageCode<AcctUsgCd>

*Presence*: [0..1]

*Definition*: Determines whether the account is a destination or a source (or both) account.

|  |  |  |
| --- | --- | --- |
| CodeName | **Name** | **Definition** |
| SRCE | Source | Source account |
| DSTN | Destination | Destination account |
| BOTH | Both | Can be both the source and the destination account. |

1. Mini Statement Code

A statement that contains complete information for a statement period is often referred as a legal or full statement. A statement that contains limited information or a set number of transactions is often referred to as a partial or mini statement. The former may be printed on more than one page on a full size printer, whereas the latter will typically print a single small receipt on the receipt printer.

The existing ATMInquiryRequest defines within its ServiceType a list of possible data to request from an acquiring server. This codeset defines a value for  Account Statements, however it does not clarify if this is for a Full Statement or a Mini Statement. This proposal adds a new codeset value which defines an option for mini statement, and clarifies the meaning of the AccountStatements (ASTS) refers to a full (or legal) statement.

ServiceType <SvcTp> is defined as:

Presence: [1..1]

Definition: Describes the type of inquiry selected by the customer or the ATM.

DataType: CodeSet ATMServiceType11Code

|  |  |  |
| --- | --- | --- |
| CodeName | Name | Definition |
| ASTS | AccountStatements | Request a full statement (i.e. legal statement) related to a customer account. |
| ... |  |  |
| ACCD | AccountDownload | Download accounts associated to the customer card. |
| MINI | MiniStatement | Request a mini-statement (i.e. partial statement) related to a customer account. |

1. Extending ATM Capabilities

ATM’s have the capability to offer various options to a consumer, such as DCC, fees, etc. These options are closely linked to the interactive Transactions feature which has been agreed previously. It would be useful to a Terminal Handler/acquiring server to inform it of the features which an ATM can handle. If an ATM was unable to handle say DCC, there would be no point in the terminal handler/acquiring server to offer such a feature in an authorization response. Also whether receipt printing is available would be useful information to provide, as in some cases there is a requirement to provide a receipt if DCC or fees is to be offered to a consumer.

The proposal is to add another two elements under the Capabilities <Cpblties> element which would provide this information (see yellow shaded elements).

Capabilities <Cpblties>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Or | Message Element | Mult. | Type |  |
|  | CardReadData <CardRdData> | [0..\*] | CodeSet |  |
|  | CardWriteData <CardWrtData> | [0..\*] | CodeSet |  |
|  | … |  |  |  |
|  | InteractiveTransactions <IntrctvTxs> | [0..\*] | String |  |
|  | ReceiptPrinting <RcptPrting> | [0..1] | boolean |  |

Where **InteractiveTransactions** **<IntrctvTxs>**:

Presence: [0..\*]

Definition: Interactive Transaction Capabilities such as DCC, Fees, Accounts, etc. which the terminal can support.

Datatype: string (although we can enumerate three possible values, there are likely to be many more, and we do not want to limit what these values may be).

And **ReceiptPrinting <RcptPrting>**:

Presence: [0..1]

Definition: whether the terminal can print a receipt (i.e. the device is functional, and it has paper).

Datatype: Boolean.

1. Add FlowTotals <FlowTtls> Elements for Initial amount/count

The FlowTotals <FlowTtls> element within a ReconciliationAdvice, WithdrawalCompletionAdvice and DepositCompletionAdvice, include the number of items removed, deposited, retracted, etc. but not the initial amount/count. These counters are required by current bank cash management systems when a cutover occurs. Banking systems expect to receive:

* Items at the beginning of the accounting period.
* Items added/removed during the accounting period.
* Items dispensed during the accounting period.
* Items deposited during the accounting period.
* Remaining at the end of the accounting period.

The proposed changes to FlowTotals <FlowTtls> is highlighted in yellow:

Presence: [0..\*]

Definition: Counters of media inside the cassette

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Or | Message Element | Mult. | Type |  |
|  | Type <Tp> | [1..1] | CodeSet |  |
|  | AddedNumber <AddedNb> | [0..1] | Quantity |  |
|  | RemovedNumber <RmvdNb> | [0..1] | Quantity |  |
|  | RemovedAmt <RmvdAmt> | [0..1] | Amount |  |
|  | DispensedNumber <DspnsdNb> | [0..1] | Quantity |  |
|  | DepositedNumber <DpstdNb> | [0..1] | Quantity |  |
|  | DepositedAmt <DpstdAmt> | [0..1] | Amount |  |
|  | ReclycledNumber <RcycldNb> | [0..1] | Quantity |  |
|  | RetractedNumber <RtrctdNb> | [0..1] | Quantity |  |
|  | RetractedAmt <RtrctdAmt> | [0..1] | Amount |  |
|  | RejectedNumber <RjctdNb> | [0..1] | Quantity |  |
|  | PresentedNumber <PresntdNb> | [0..1] | Quantity |  |
|  | InitialNumber <InitNb> | [0..1] | Quantity |  |
|  | InitialAmount <InitAmt> | [0..1] | Amount |  |

where

InitialNumber <InitNb>

*Presence*: [0..1]

*Definition*: Initial number added during a servicing operation.

*Datatype*: number.

InitialAmount <InitAmt>

*Presence*: [0..1]

*Definition*: Initial amount added during a servicing operation.

*Datatype*: ImpliedCurrencyAndAmount.

1. Add new Incident codeset values

The FailureReason7Code defines a set of incidents which can occur during the processing of an ATM transaction. The following suggests adding these additional incidents:

|  |  |  |
| --- | --- | --- |
| **CodeName** | **Name** | **Definition** |
| CHFG | CashDispenserRetract | Following a cash dispense operation the consumer forgot to take the notes, which have subsequently been retracted following a Present time-out. The number of notes retracted is unknown. |
| DCFG | CashDepositRetract | Following a cash return, on a deposit operation, the consumer forgot to take the returned notes, causing them to be retracted. The number of notes retracted is unknown. |
| CQFG | ChequeDepositRetract | Following cheques being returned, on a cheque deposit operation, the consumer forgot to take the returned cheques, causing them to be retracted. The number of cheques retracted is unknown. |
| CFGC | CashDispenserRetractCounted | Following a cash dispense operation the consumer forgot to take the returned notes, which have subsequently been retracted following a Present time-out. The number of notes retracted is known as the device was able to count the notes retracted. |
| DFGC | CashDepositRetractCounted | Following a cash return, on a deposit operation, the consumer forgot to take the returned notes, causing them to be retracted. The number of notes retracted is known as the device was able to count the notes retracted. |
| QFGC | ChequeDepositRetractCounted | Following cheques being returned, on a cheque deposit operation, the consumer forgot to take the returned cheques, causing them to be retracted. The number of cheques retracted is known as the device was able to count the cheques retracted |
| DCRT | DepositsRetained | Cash has been retained by the terminal since it has been unable to return it. |
| CQRT | ChequesRetained | Cheques have been retained by the terminal since it has been unable to return them. |

1. **Purpose of the change:**

Business justification:

All these change requests are due to gaps in our initial specifications and were discovered while implementing applications. One change request introduces the concept of interactive transactions, whereby a more efficient method of providing fee, DCC, account, etc. information can be provided to a terminal by an acquiring server.

1. **Urgency of the request:**

*Urgent.*

1. **Business examples:**
2. **SEG recommendation:**

*This section is not to be taken care of by the submitter of the change request. It will be completed in due time by the SEG(s) in charge of the related ISO 20022 messages.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Consider** | |  | **Timing** |  |  |  |
|  |  | | - **Next yearly cycle: 2024/2025**  (the change will be considered for implementation in the yearly maintenance cycle which starts in 2024 and completes with the publication of new message versions in the spring of 2025) | | X |  |
|  |  | | - **At the occasion of the next maintenance of the messages**  (the change will be considered for implementation, but does not justify maintenance of the messages in its own right – will be pending until more critical change requests are received for the messages) | |  |  |
|  |  | | - **Urgent unscheduled**  (the change justifies an urgent implementation outside of the normal yearly cycle) | |  |  |
|  |  | | - **Other timing:** | | |  |

Comments:

|  |  |
| --- | --- |
| **Reject** |  |

Reason for rejection: