**ISO 20022**

**Acquirer to Issuer Card Messages (ATICA)**

Approved by the Cards and Related Retail Financial Services SEG

**Message Definition Report** **- Part 1**

Edition December 2015

Table of contents

[1. Introduction 4](#_Toc437446311)

[1.1 Terms and definitions 4](#_Toc437446312)

[1.2 Glossary 5](#_Toc437446313)

[1.3 Document Scope and Objectives 6](#_Toc437446314)

[1.4 References 6](#_Toc437446315)

[2. Scope and Functionality 7](#_Toc437446316)

[2.1 Background 7](#_Toc437446317)

[2.2 Scope 7](#_Toc437446318)

[2.3 Groups of *MessageDefinitions* and Functionality 7](#_Toc437446319)

[3. *BusinessRoles* and *Participants* 9](#_Toc437446320)

[4. *BusinessProcess* Description 11](#_Toc437446321)

[4.1 *BusinessProcess* Diagram 11](#_Toc437446322)

[5. Description of *BusinessActivities* 13](#_Toc437446323)

[5.1 *BusinessProcess* – Card Payment and Cash Withdrawal Process 14](#_Toc437446324)

[5.2 *BusinessProcess* – Management Processes 17](#_Toc437446325)

[*6.* *BusinessTransactions* 20](#_Toc437446326)

[6.1 Introduction 20](#_Toc437446327)

[6.2 Authorisation and Financial exchanges 21](#_Toc437446328)

[6.2.1 Dual Message Authorisation with Subsequent Clearing 21](#_Toc437446329)

[6.2.2 Unsuccessful Dual Message Authorisation 22](#_Toc437446330)

[6.2.3 Stand-in Dual Message Authorisation by an Agent with Subsequent Clearing 23](#_Toc437446331)

[6.2.4 Dual Message Authorisation with Estimated Amount, Advice with Final Amount and Subsequent Clearing 24](#_Toc437446332)

[6.2.5 Successful Single Message 26](#_Toc437446333)

[6.2.6 Conversion from Acquirer Single Message to Issuer Dual Message 27](#_Toc437446334)

[6.2.7 Single Message Authorisation with a Maximum Amount, Financial Advice with the Final Amount 28](#_Toc437446335)

[6.2.8 Successful Dual Message Authorisation without an Agent 29](#_Toc437446336)

[6.3 Reversal exchanges 30](#_Toc437446337)

[6.3.1 Single Message with Issuer Timeout 30](#_Toc437446338)

[6.3.2 Single Message with Agent Timeout 31](#_Toc437446339)

[6.3.3 Dual Message Timeout Authorisation without an Agent 32](#_Toc437446340)

[6.3.4 Authorisation with a Subsequent Reversal Advice 33](#_Toc437446341)

[6.3.5 Authorisation with a Subsequent Approved Reversal Request 34](#_Toc437446342)

[6.4 Reconciliation 35](#_Toc437446343)

[6.4.1 Reconciliations Initiated by the Agent 35](#_Toc437446344)

[6.4.2 Single Message Reconciliation without an Agent Initiated by the Acquirer 36](#_Toc437446345)

[6.5 Management and Control 37](#_Toc437446346)

[6.5.1 Acquirer and Issuer Sign-on to an Agent 37](#_Toc437446347)

[6.5.2 Echo Test after an Inactivity Period 38](#_Toc437446348)

[6.5.3 Acquirer Key Exchange Initiated by the Agent 39](#_Toc437446349)

[6.5.4 Key Exchange Initiated by the Issuer 40](#_Toc437446350)

[6.5.5 Reject of an Authorisation Request by an Issuer 41](#_Toc437446351)

[6.5.6 Reject of an Authorisation Response by an Agent 42](#_Toc437446352)

[7. Examples 43](#_Toc437446353)

[8. Revision Record 43](#_Toc437446354)

**Preliminary note:**

The Message Definition Report (MDR) is made of three parts:

* **MDR - Part 1** describes the contextual background required to understand the functionality of the proposed message set. Part 1 is produced by the submitting organisation that developed or maintained the message set in line with a MDR Part1 template provided by the ISO 20022 Registration Authority (RA) on [www.iso20022.org](http://www.iso20022.org)
* **MDR – Part 2** is the detailed description of each message definition of the message set. Part 2 is produced by the RA using the model developed by the submitting organisation.
* **MDR – Part 3** is an extract of the ISO 20022 Business Model describing the business concepts used in the message set. Part 3 is an Excel document produced by the RA.

# Introduction

## Terms and definitions

The following terms are reserved words defined in ISO 20022 – Part1. When used in this document, they will be in italic and follow the *UpperCamelCase* notation.

|  |  |
| --- | --- |
| Term | Definition |
| *BusinessRole* | functional role played by a business actor in a particular *BusinessProcess* or *BusinessTransaction* |
| *Participant* | involvement of a *BusinessRole* in a *BusinessTransaction* |
| *BusinessProcess* | unrealized definition of the business activities undertaken by *BusinessRoles* within a *BusinessArea* whereby each *BusinessProcess* fulfils one type of business activity and whereby a *BusinessProcess* may include and extend other *BusinessProcesses* |
| *BusinessTransaction* | particular solution that meets the communication requirements and the interaction requirements of a particular *BusinessProcess* and *BusinessArea* |
| *MessageDefinition* | formal description of the structure of a *MessageInstance* |
| *Request* | Message where the sender informs the receiver that an action is in progress which requires a response from the receiver to complete the action.  Message where the sender requests a response from the receiver before to complete an action initiated by the sender. |
| *Response* | Message where the receiver instructs the sender of a request message on how the action has to be completed. |
| *Advice* | Message where the sender advices the receiver that an action has been taken, requiring an advice response. |
| *Advice Response* | Message where the receiver informs the sender of an advice message that the advice has been processed. |
| *Notification* | Message where the sender notifies the receiver that an action has been taken, without requiring any response. |

## Glossary

**Acronyms**

|  |  |
| --- | --- |
| Acronym | Definition |
| AES | Advanced Encryption Standard |
| ASN.1 | Abstract Syntax Notation 1 |
| cain | ISO 20022 Acquirer to Issuer messages (ATICA) |
| CAPE | Card Payment Exchanges |
| DES | Data Encryption Standard |
| DUKPT | Derived Unique Key Per Transaction |
| EMV | Europay, MasterCard, Visa |
| FIPS | Federal Information Processing Standard |
| FTP | File Transfer Protocol |
| ICC | Integrated Circuit Card |
| IP | Internet Protocol |
| ISO | International Organization for Standardization |
| KEK | Key Encryption Key |
| MAC | Message Authentication Code |
| MDR | Message Definition Report |
| MOTO | Mail Order, Telephone Order |
| PAN | Primary Account Number |
| PED | PIN Entry Device |
| PIN | Personal Identification Number |
| PKI | Public Key Infrastructure |
| POI | Point Of Interaction |
| POS | Point Of Sales |
| PSP | Payment Service Provider |
| RFID | Radio Frequency Identification |
| RSA | Rivest Shamir Adleman |
| SC2 | TC 68/SC2 Financial Services, security |
| SHA | Secure Hash Algorithm |
| STIP | STand-In processing |
| TMS | Terminal Management System |
| UKPT | Unique Key Per Transaction |
| UML | Unified Modelling Language |
| XML | eXtensible Mark-up Language |

## Document Scope and Objectives

This document is the first part of the ISO 20022 Message Definition Report (MDR) that describes the *BusinessTransactions* and underlying message set. For the sake of completeness, the document may also describe *BusinessActivities* that are not in the scope of the project.

This document sets:

* The *BusinessProcess* scope (business processes addressed or impacted by the project)
* The *BusinessRoles* involved in these *BusinessProcesses*

The main objectives of this document are:

* To explain what *BusinessProcesses* and *BusinessActivities* these candidate MessageDefinitions have addressed
* To give a high level description of *BusinessProcesses* and the associated *BusinessRoles*
* To document the *BusinessTransactions* and their *Participants* (sequence diagrams)
* To list the candidate *MessageDefinitions*

## References

| Document | | Version | | Date | | Author |
| --- | --- | --- | --- | --- | --- | --- |
| ISO 20022 Business Justification – Acquirer to Issuer Card Messages (ATICA) | 13 Oct. 2009 | | 2009-10-06 | | ISO TC68/SC7/TG1 | |
| Card Payment Protocols Security[[1]](#footnote-1) | 2.0 | | 2014 | | EPASOrg | |

# Scope and Functionality

## Background

This Message Definition Report covers an initial set of 13 ISO 20022 *MessageDefinitions* developed by ISO TC68/SC7/TG1 and approved by the Cards and Related Retail Financial Service Standards Evaluation Group (SEG). These messages are specifically designed to support card payment business services between an acquirer and an issuer.

## Scope

The messages can be used for exchanges between an acquirer and an issuer for card payments and cash withdrawals. The exchanges could involve intermediary agents in this process as well.

## Groups of *MessageDefinitions* and Functionality

A card transaction is the basic business service allowing a cardholder to pay for the purchase of goods and services as well as the withdrawal of cash. Other services are provided in addition to the basic card payment business service.

These services are supported by the following exchanges of messages between an Acquirer and an Issuer.

These messages are intended for use without the BAH (Business Application Header).

**Acquirer to Issuer Authorisation and Financial messages:**

* The *AcquirerAuthorisationInitiation* (cain.001.001.01): to initiate authorisation of a card payment transaction;
* The *AcquirerAuthorisationResponse* (cain.002.001.01): to return the results of an authorisation of a card payment transaction;
* The *AcquirerFinancialInitiation (*cain.003.001.01): to initiate the financial authorisation of a card payment transaction;
* The *AcquirerFinancialResponse*  (cain.004.001.01): to return the results of a financial authorisation of a card payment transaction;

**Acquirer to Issuer Reversal messages:**

* The *AcquirerReversalInitiation* (cain.005.001.01): to initiate a reversal of an authorisation or a financial authorisation and the associated card payment transaction;
* The *AcquirerReversalResponse* (cain.006.001.01): to return the results of a reversal initiation.

**Acquirer to Issuer Reconciliation messages:**

* The *ReconciliationInitiation* (cain.007.001.01): to initiate either by the acquirer or by the issuer, an exchange of totals to be reconciled for debits, credits, chargebacks and other transactions.
* The *ReconciliationResponse* (cain.008.001.01): to return either by the acquirer or by the issuer the reconciled totals for debits, credits, chargebacks and other transactions

**Acquirer to Issuer Network Management messages:**

* The *NetworkManagementInitiation* (cain.009.001.01): sent by any party, to control the operating condition of the network.
* The *NetworkManagementResponse* (cain.010.001.01): sent by any party, to answer to a *NetworkManagementInitiation* message.

**Acquirer to Issuer Security messages:**

* The *KeyExchangeInitiation* (cain.011.001.01): sent by any party, to initiate a cryptographic key exchange.
* The *KeyExchangeResponse* (cain.012.001.01): sent by any party, to answer to a *KeyExchangeInitiation* message and complete a cryptographic key exchange.

**Acquirer to Issuer Rejection messages:**

* The *AcquirerRejection* (cain.013.001.01): sent by any party, to reject an acquirer to issuer message.

# *BusinessRoles* and *Participants*

A *BusinessRole* represents an entity (or a class of entities) of the real world, physical or legal, a person, a group of persons, a corporation. Examples of *BusinessRoles*: “Acquirer”, “Issuier”, etc..

A *Participant* is a functional role performed by a *BusinessRole* in a particular BusinessProcess or BusinessTransaction: for example the “Financial Institution”, “Processor”, “Card Scheme”.

The relationship between *BusinessRoles* and *Participants* is many-to-many. One *BusinessRole* can be involved as different *Participants* at different moments in time or at the same time: "Financial Institution", etc. Different *BusinessRoles* can be involved as the same Participant.

In the context of card payments, the high-level *BusinessRoles* and typical *Participants* can be represented as follows.



| ***Participants* and *BusinessRoles* definitions** | |
| --- | --- |
| **Description** | **Definition** |
| *BusinessRoles* | |
| Acquirer | An entity acquiring card payment transactions. |
| Issuer | An entity issuing a payment card. |
| Agent | An entity acting as an intermediary between an acquirer and an issuer (e.g. card payment network). |
| *Participants* | |
| Financial Institution | A financial or related institution acquiring card payment transactions or issuing cards, and performing the settlement for acceptor and cardholder accounts, when relevant. |
| Processor | An entity providing card payment processing services. |
| Card Scheme | An entity defining rules and procedures for card payment transactions. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***BusinessRoles*/*Participants* Matrix Table** | | | |
| *Participants*  *BusinessRoles* | Financial Institution | Processor | Card Scheme | |
| Acquirer | X | X |  | |
| Agent |  | X | X | |
| Issuer | X | X |  | |

# *BusinessProcess* Description

## *BusinessProcess* Diagram

This diagram shows the high level *BusinessProcess* covered by card payment business.



**Card payment or cash withdrawal process:**

* *Definition*: The process of performing a payment of good or services with a card. The process starts with the acceptance of the payment card, involves the authorisation of the payment transaction, and the transfer of financial information (capture) from the acquirer to the issuer. In some error situations, it is necessary to reverse the authorisation.
* *Trigger*: The submission of a card payment transaction by an acquirer to an issuer for the payment of goods or services.
* *Pre-conditions*: Presentation of the card payment transaction by the acquirer for payment.
* *Post-conditions*: If the payment is successful, the Issuer is in the position of initiating the procedure to credit the merchant account and debit the customer account.
* *Role*: acquirer, agent and issuer.

**Management process:**

* *Definition*: The management process covers the range of activities required to control the operating conditions of card payment and cash withdrawal processes. It may be initiated by any party.
* *Trigger*: The variation of the operating condition for the card payment and cash withdrawal processes.
* *Pre-conditions*: The operating conditions are inappropriate for performing card payment and cash withdrawal processes.
* *Post-conditions*: The operating conditions are suitable for performing card payment and cash withdrawal processes.
* *Role*: acquirer, agent and issuer.

# Description of *BusinessActivities*

This section presents the different *BusinessActivities* within each *BusinessProcess*. *BusinessActivities* of a process are described in swim lane diagrams and are referred in this document as activity diagrams.

The development of an activity diagram is part of the ISO 20022 modelling process and allows capturing the requirements.

The activity diagram provides a zoom-in on the *BusinessActivities* taking place during each of the *BusinessProcesses* described in Section 4. It also shows the *BusinessActivities* that are triggered when another *BusinessActivity* has a negative result.

What is the activity diagram about?

* It is a diagram representing the ‘common lifecycle’ of a *BusinessProcess*
* A start point ⚫ shows where the lifecycle of the *BusinessProcess* commences and the end points show End point where the lifecycle may possibly end
* A lozenge means that a choice between several actions can be made
* A bar means that several actions are initiated in parallel
* The flow of activities between the involved *Participants* (parties)
* *BusinessActivities* may result in different actions, that is, information is conveyed from one party to another party.

Both in-scope and out-of-scope activities are included, with a different level of details. There are no information requirements for out-of-scope activities, except that they should be clearly identified in the diagram.

Activity diagrams are always accompanied with a text describing the *BusinessActivities* and their interactions.

## *BusinessProcess* – Card Payment and Cash Withdrawal Process



|  |  |
| --- | --- |
| **Descriptions of the *BusinessActivities*** | |
|  | **Initiator** |
| **Process Acceptor Request**: The acquirer processes an authorisation request (or a financial request) for a payment transaction or a cash withdrawal transaction. The acquirer sends an *AcquirerAuthorisationInitiation* message to the agent, to obtain the issuer authorisation. | **Acquirer** |
| **Process Acquirer Request**: The agent processes the authorisation by sending an *AcquirerAuthorisationInitiation* message to the issuer. | **Agent** |
| **Perform Authorisation**: The authorisation of a payment or a cash withdrawal transaction is performed by the issuer. The issuer sends the outcome of the authorisation to the agent through an *AcquirerAuthorisationResponse* message. | **Issuer** |
| **Process Issuer Response**: The agent sends an *AcquirerAuthorisationResponse* message to the acquirer either to:   * provide the outcome of the issuer authorisation received from the issuer in the *AcquirerAuthorisationResponse* message, or * decline the authorisation when the agent didn't receive a response from the issuer (timeout). | **Agent** |
| **Process Agent Response**: The acquirer sends a response to the acceptor either to:   * provide the outcome of the issuer authorisation received from the agent in the *AcquirerAuthorisationResponse* message, or * decline the authorisation when the acquirer didn't receive a response from the agent (timeout). | **Acquirer** |
| **Process Acquirer Reversal**: In case of timeout on the authorisation response from the agent, the acquirer reverses the authorisation, sending an *AcquirerReversalInitiation* message to the agent. | **Acquirer** |
| **Process Agent Reversal**: The agent reverses the authorisation sending an *AcquirerReversalInitiation* message to the issuer when:   * The agent receives an *AcquirerReversalInitiation* message from the acquirer. The agent then acknowledges the reversal by sending an *AcquirerReversalResponse* message to the acquirer. * A timeout occurs on the authorisation response from the issuer. | **Agent** |
| **Perform Reversal**: When the issuer receives an *AcquirerReversalInitiation* message from the agent, the issuer then reverses the delivered authorisation, and acknowledges the reversal by sending an *AcquirerReversalResponse* messageto the agent. | **Issuer** |
| **Process Reversal Response**: When the agent or the acquirer receives an *AcquirerReversalResponse* message from respectively the issuer or the agent, the transaction ends without success. | **Agent**  **Acquirer** |
| **Process Acquirer Financial**: If the authorisation was approved by the issuer, the acquirer performs the clearing of the transaction by sending an *AcquirerFinancialInitiation* notification message to the agent. | **Acquirer** |
| **Process Agent Financial**: The acquirer sends an *AcquirerFinancialInitiation* notification message to the agent. The agent performs the issuer clearing of the transaction by sending an *AcquirerFinancialInitiation* notification message to the issuer. | **Agent** |
| **Process Issuer Financial**: The agent sends an *AcquirerFinancialInitiation* notification message to the issuer. The issuer performs the clearing of the transaction. | **Issuer** |
| **Perform Acquirer Reconciliation**: The agent performs the reconciliation of the period for the acquirer. The agent notifies the reconciliation to the acquirer sending the totals of the period in an *AcquirerReconciliationInitiation* notification message. | **Agent** |
| **Perform Issuer Reconciliation**: The agent performs the reconciliation of the period for the issuer. The agent notifies the reconciliation to the issuer sending the totals of the period in an *AcquirerReconciliationInitiation* notification message. | **Agent** |
| **Process Reconciliation**: When the acquirer (or the issuer) receives the notification of the reconciliation by the agent, the acquirer (or the issuer) verifies the totals of the reconciliation period from the agent received in the *AcquirerReconciliationInitiation* notification message. | **Acquirer**  **Issuer** |

## *BusinessProcess* – Management Processes



|  |  |
| --- | --- |
| **Descriptions of the *BusinessActivities*** | |
|  | **Initiator** |
| **Agent Network Management**: When the agent initiates a network management process with an acquirer or an issuer, the agent then:   * Sends a *NetworkManagementInitiation* message to that party, waiting for the response, and * Receives a *NetworkManagementResponse* message from that party.   When a party initiates a network management process with an agent, the agent then:   * Receives a *NetworkManagementInitiation* message from the party, waiting for the response, and * Processes the requested network management service and sends a *NetworkManagementResponse* message to that party. | **Agent** |
| **Issuer Network Management**: When the issuer initiates a network management process with an agent, the issuer:   * Sends a *NetworkManagementInitiation* message to the agent, waiting for the response, and * Receives a *NetworkManagementResponse* message from the agent.   When an agent initiates a network management process with an issuer, the issuer:   * Receives a *NetworkManagementInitiation* message from the agent, waiting for the response, and * Processes the requested network management service and sends a *NetworkManagementResponse* message to the agent. | **Issuer** |
| **Acquirer Network Management**: When the acquirer initiates a network management process with an agent, the acquirer:   * Sends a *NetworkManagementInitiation* message to the agent, waiting for the response, and * Receives a *NetworkManagementResponse* message from the agent.   When an agent initiates a network management process with an acquirer, the acquirer:   * Receives a *NetworkManagementInitiation* message from the agent, waiting for the response, and * Processes the requested network management service and sends a *NetworkManagementResponse* message to the agent. | **Acquirer** |
| **Key Exchange Initiation**: When an acquirer or an issuer initiates a key exchange process with an agent, the acquirer or the issuer:   * Initiates the key exchange, and * Sends a *KeyExchangeInitiation* message to the agent, waiting for the response.   When an agent initiates a network management process with an acquirer or an issuer, the agent:   * Initiates the key exchange, and * Sends a *KeyExchangeInitiation* message to that party, waiting for the response. | **Acquirer**  **Agent**  **Issuer** |
| **Perform Key Exchange**: When an acquirer or an issuer initiates a key exchange process with an agent, the agent:   * Receives a *KeyExchangeInitiation* message from the relevant party, * Validates the message and sends an *AcquireRejection* message to the relevant party, if the message is invalid, * Updates the key and sends a *KeyExchangeResponse* message to the relevant party.   When an agent initiates a network management process with an acquirer or an issuer, the acquirer or issuer:   * Receives a *KeyExchangeInitiation* message from the agent, * Validates the message and sends an *AcquireRejection* message to the agent, if the message is invalid, * Updates the key and sends a *KeyExchangeResponse* message to the agent. | **Acquirer**  **Agent**  **Issuer** |
| **Key Exchange Completion**: When the acquirer or issuer initiates a key exchange process with an agent, the agent:   * Receives an *AcquireRejection* message from the relevant party, if the *KeyExchangeResponse* message it sent was invalid, * Otherwise the key exchange process is complete.   When an agent initiates a network management process with an acquirer or an issuer the relevant party:   * Receives an *AcquireRejection* message from the agent, if the *KeyExchangeResponse* message it sent was invalid, * Otherwise the key exchange process is complete. | **Acquirer**  **Agent**  **Issuer** |

# *BusinessTransactions*

This section describes the message flows based on the activity diagrams documented above. It shows the typical exchanges of information in the context of a *BusinessTransaction*.

## Introduction

The following card payment BusinessTransactionsshow how an acquirer interacts with an issuer when a cardholder has initiated the payment for the purchase of goods and services from an acceptor by using his card[[2]](#footnote-2). These BusinessTransactionsscenarios are typical card payment ones without being an exhaustive list. They are supported by exchanges of messages.

A card payment is supported by an authorisation process to request the approval of the BusinessTransaction.

A financial exchange is required when the issuer has to be notified of the outcome of the payment.

The financial data of the BusinessTransaction must be transferred to the issuer through a financial message for clearing.

## Authorisation and Financial exchanges

### Dual Message Authorisation with Subsequent Clearing



1. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is sent by the acquirer to the agent to request authorisation.
2. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is forwarded by the agent to the card issuer to request authorisation.
3. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the issuer to inform the agent about the successful outcome of the request; once the issuer has authorised the BusinessTransaction.
4. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is forwarded by the agent to inform the acquirer about the successful outcome of the request; once the issuer has authorised the BusinessTransaction.
5. The *AcquirerFinancialInitiation* (Message Function: FinancialNotification) is sent by the acquirer to the agent to complete and clear the transaction. The reconciliation totals (amount and number) are updated at both sides.
6. The *AcquirerFinancialInitiation* (Message Function: FinancialNotification) is forwarded by the agent to the issuer to complete and clear the transaction. The reconciliation totals (amount and number) are updated at both sides.

### Unsuccessful Dual Message Authorisation



1. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is sent by the acquirer to the agent to request authorisation.
2. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is forwarded by the agent to the issuer to request authorisation.
3. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the issuer to inform the agent that the transaction has been declined.
4. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the agent to inform the acquirer that the transaction has been declined.

### Stand-in Dual Message Authorisation by an Agent with Subsequent Clearing



1. The acquirer sends an *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) to the agent for authorisation
2. The issuer is unavailable. The agent authorises the transaction on behalf of the issuer (stand-in process) and sends an *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) to the acquirer.
3. The agent sends an *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationAdvice) to inform the issuer about the authorisation
4. The issuer sends an *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationAdviceResponse) to the agent to acknowledge the successful authorisation of the transaction.
5. The *AcquirerFinancialInitiation* (MessageFunction: FinancialNotification) is sent by the acquirer to the agent to initiate the clearing of the transaction
6. The AcquirerFinancialInitiation (MessageFunction: FinancialNotification) is sent by the agent to the issuer to initiate the clearing of the transaction.

### Dual Message Authorisation with Estimated Amount, Advice with Final Amount and Subsequent Clearing



1. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is sent by the acquirer to the agent to request authorisation for an estimated amount.
2. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is forwarded by the agent to the card issuer to request authorisation for this estimated amount.
3. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the issuer to inform the agent about the successful outcome of the request; once the issuer has approved the BusinessTransaction.
4. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is forwarded by the agent to inform the acquirer about the successful outcome of the request.
5. The transaction is completed with the final amount. The acquirer sends to the agent an *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationAdvice) to inform the issuer about the final amount.
6. The agent sends to the issuer an *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationAdvice) to inform the issuer about the final amount.
7. The agent sends to the acquirer an *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationAdviceResponse) to inform about the processing of the final amount.
8. The issuer sends an *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationAdviceResponse) to the agent and updates the transaction amount with the final amount.
9. The AcquirerFinancialInitiation (Message Function: FinancialNotification) is sent by the acquirer to the agent to complete and clear the transaction. The reconciliation totals (amount and number) are updated on both sides.
10. The AcquirerFinancialInitiation (Message Function: FinancialNotification) is sent by the agent to the issuer to complete and clear the transaction. The reconciliation totals (amount and number) are updated at both sides.

### Successful Single Message



1. The *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) is sent by the acquirer to the agent to request both authorisation and clearing.
2. The *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) is forwarded by the agent to the card issuer to request both authorisation and clearing.
3. The issuer approves the authorisation and updates the reconciliation totals (amount and number). An *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) is returned by the issuer to inform the agent about the successful outcome of the request.
4. The *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) is forwarded by the agent to inform the acquirer about the successful outcome of the request. The reconciliation totals (amount and number) are updated by the agent.  
   The acquirer updates the reconciliation totals (amount and number).

### Conversion from Acquirer Single Message to Issuer Dual Message



1. The *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) is sent by the acquirer to the agent to request both authorisation and clearing.
2. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is sent by the agent to the card issuer to request authorisation.
3. The issuer approves the authorisation and sends to the agent an *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse).
4. The *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) is sent by the agent to inform the acquirer about the successful outcome of the request. The acquirer reconciliation totals (amount and number) are updated by the agent.
5. The AcquirerFinancialInitiation (Message Function: FinancialNotification) is sent by the agent to the issuer to clear the transaction. The issuer reconciliation totals (amount and number) are updated at both sides.

### Single Message Authorisation with a Maximum Amount, Financial Advice with the Final Amount



1. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is sent by the acquirer to the issuer to request authorisation for a maximum amount.
2. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is forwarded by the agent to the card issuer to request authorisation for this maximum amount.
3. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the issuer to inform the agent about the successful outcome of the request; once the issuer has approved the BusinessTransaction.
4. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is forwarded by the agent to inform the acquirer about the successful outcome of the request.
5. The reconciliation totals (amount and number) are updated by the acquirer. The *AcquirerFinancialInitiation* (MessageFunction: FinancialAdvice) is sent by the acquirer to the agent to advice the issuer about the final amount of the transaction and to initiate the clearing.
6. The reconciliation totals (amount and number) are updated by the agent. The agent sends an *AcquirerFinancialResponse* (MessageFunction: FinancialAdviceResponse) to the acquirer to inform about the processing of the clearing with the final amount.
7. The agent sends an *AcquirerFinancialInitiation* (MessageFunction: FinancialAdvice) to the issuer to inform about the processing of the clearing with the final amount.
8. The *AcquirerFinancialResponse* (MessageFunction: FinancialAdviceResponse) is returned by the issuer to inform the acquirer about the completion of the transaction. The reconciliation totals (amount and number) are updated by the issuer.

### Successful Dual Message Authorisation without an Agent



1. The *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest is sent by the acquirer to the issuer to request the authorisation of the transaction
2. The *AcquireAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the issuer to the acquirer to inform him about the successful authorisation of the transaction.
3. The *AcquirerFinancialInitiation* (MessageFunction: FinancialNotification) is sent by the acquirer to the issuer to notify the issuer about the successful clearing of the transaction.

## Reversal exchanges

### Single Message with Issuer Timeout



1. The acquirer sends an *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) to the agent for authorisation and clearing.
2. The agent sends an *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) to the agent for authorisation and clearing. The issuer approves the authorisation, updates the reconciliation totals and sends an *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) to inform the acquirer about the success of the request.
3. After a timeout on the response, the agent sends an *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) to the acquirer inform him that the transaction has been declined for technical reasons.
4. The agent initiates the reversal of the transaction by sending an *AcquirerReversalInitiation* (MessageFunction: FinancialReversalAdvice) to request the issuer to initiate the reversal
5. The issuer sends back an *AcquirerReversalResponse* (MessageFunction: FinancialReversalAdviceResponse) to the agent to acknowledge the successful reversal of the transaction.

### Single Message with Agent Timeout



1. The acquirer sends an *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) to the agent for authorisation and clearing.
2. The agent forwards the *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) to the issuer for authorisation and clearing.
3. The issuer approves the authorisation, updates the reconciliation totals and sends *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) to inform the agent about the success of the request.
4. The agent updates the reconciliation totals and fails to forward the *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) to the acquirer.  
   After a timeout, the acquirer sends to the agent an *AcquirerReversalInitiation* (MessageFunction: FinancialReversalAdvice) to initiate the reversal.
5. The agent update the acquirer reconciliation totals and return to the acquirer the *AcquirerReversalResponse* (MessageFunction: FinancialReversalAdviceResponse) to inform about the success of the reversal.
6. The agent initiates the issuer reversal of the transaction by sending an *AcquirerReversalInitiation* (MessageFunction: FinancialReversalAdvice) to request the issuer to reverse the transaction.
7. The issuer reverses the transaction, updates the reconciliation totals and sends back an *AcquirerReversalResponse* (MessageFunction: FinancialReversalAdviceResponse) to the agent to acknowledge the successful reversal of the transaction.  
   The agent updates then the issuer reconciliation totals.

### Dual Message Timeout Authorisation without an Agent



1. The acquirer sends an *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) to the issuer for authorisation. Due to a communication error, no *AcquirerAuthorisationResponse* is returned by the issuer to the acquirer.
2. After timeout, the acquirer sends an *AcquirerReversalInitiation* (MessageFunction: AuthorisationReversalAdvice) to request the issuer to reverse the transaction
3. The *AcquirerReversalResponse* (MessageFunction: AuthorisationReversalAdviceResponse) is sent back by the issuer to inform the acquirer that the transaction has been reversed.

### Authorisation with a Subsequent Reversal Advice



1. The acquirer sends an *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) to the agent for authorisation.
2. The agent forwards the *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) to the issuer for authorisation.
3. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the issuer to inform the acquirer about the approval of the authorisation.
4. The agent forwards the *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) to inform the acquirer about the approval of the authorisation.
5. The transaction cannot be completed successfully. The acquirer reverses the transaction and sends an *AcquirerReversalInitiation* (MessageFunction: AuthorisationReversalAdvice) to inform the agent about the reversal.
6. The agent sends back an *AcquirerReversalResponse* (MessageFunction: AuthorisationReversalAdviceResponse) to the acquirer to acknowledge the successful reversal of the transaction.
7. An *AcquirerReversalInitiation* (MessageFunction: AuthorisationReversalAdvice) message is sent by the agent to the issuer to initiate the reversal of the transaction.
8. The *AcquirerReversalResponse* (MessageFunction: AuthorisationReversalAdviceResponse) message is sent by the issuer to the agent inform him that the transaction has been reversed.

### Authorisation with a Subsequent Approved Reversal Request



1. The acquirer sends an *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) to the agent for authorisation
2. The agent forwards the *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) to the issuer for authorisation
3. The *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the issuer to inform the acquirer about the successful outcome of the request
4. The agent forwards an *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) to the acquirer to authorise the transaction.
5. The acquirer reverses the transaction and sends an *AcquirerReversalInitiation* (MessageFunction: AuthorisationReversalRequest) to request the agent to reverse the authorisation subsequently
6. The *AcquirerReversalInitiation* (MessageFunction: AuthorisationReversalRequest) message is forwarded by the agent to the issuer to process the reversal of the transaction
7. The *AcquirerReversalResponse* (MessageFunction: AuthorisationReversalResponse) is sent by the issuer to the agent to confirm that the transaction has been reversed.
8. The agent forwards an *AcquirerReversalResponse* (MessageFunction: AuthorisationReversalResponse) to the acquirer to confirm the successful reversal of the transaction.

## Reconciliation

### Reconciliations Initiated by the Agent



1a-na: The acquirer sends an *AcquirerFinancialInitiation* (Message Function: FinancialNotification and/or any type of ReversalNotification) to the agent for transaction clearing.

na+1: The agent reconciles the acquirer session, sending to the acquirer an *AcquirerReconciliationInitiation* (MessageFunction: AcquirerReconciliationNotification). The acquirer is then ready for the settlement of the net amount.

1i-mi: The agent sends an *AcquirerFinancialInitiation* (Message Function: FinancialNotification and/or any type of ReversalNotification) to the issuer for transaction clearing.

mi+1: The agent reconciles the issuer session and sends to the issuer an *AcquirerReconciliationInitiation* (MessageFunction: AcquirerReconciliationNotification). The issuer is then ready for the settlement of the net amount.

### Single Message Reconciliation without an Agent Initiated by the Acquirer



1: The acquirer sends an *AcquirerFinancialInitiation* (Message Function: FinancialRequest) to the issuer for authorisation and clearing.

2: The issuer approves the authorisation, updates the reconciliation totals, and sends an *AcquirerFinancialresponse*(Message Function: FinancialResponse) to the acquirer.

n: The acquirer sends an *AcquirerFinancialInitiation* (Message Function: FinancialRequest) to the issuer for authorisation and clearing.

n+1: The issuer approves the authorisation, updates the reconciliation totals, and sends an *AcquirerFinancialresponse*(Message Function: FinancialResponse) to the acquirer.

n+2: The acquirer reconciles the session, sending to the issuer an *AcquirerReconciliationInitiation* (MessageFunction: AcquirerReconciliationRequest).

n+3: The issuer reconciles the session, sending to the acquirer an *AcquirerReconciliationResponse* (MessageFunction: AcquirerReconciliationResponse). The issuer and the acquirer are then ready for the settlement of the net amount.

## Management and Control

### Acquirer and Issuer Sign-on to an Agent



1. A *NetworkManagementInitiation* (MessageFunction: NetworkManagementRequest) is sent by the acquirer to the agent to open the communication at the application level.
2. The agent enables the application flow for the acquirer and sends a *NetworkManagementResponse* (MessageFunction: NetworkManagementResponse) to the acquirer.
3. A *NetworkManagementInitiation* (MessageFunction: NetworkManagementRequest) is sent by the issuer to the agent to open the communication at the application level.
4. The agent enables the application flow for the issuer and sends a *NetworkManagementResponse* (MessageFunction: NetworkManagementResponse) to the issuer.

### Echo Test after an Inactivity Period



1. The acquirer notifies the issuer about the successful clearing of a transaction bysending an *AcquirerFinancialInitiation* (MessageFunction: FinancialNotification) to the issuer (as an example of a last exchange of messages).
2. After an inactivity period without any exchange of messages, the acquirer tests the availability of the issuer application by sending a *NetworkManagementInitiation* (MessageFunction: NetworkManagementRequest) to the issuer.
3. After verifying the availability of the application, the issuer answers with a *NetworkManagementResponse* (MessageFunction: NetworkManagementResponse) to the acquirer.

### Acquirer Key Exchange Initiated by the Agent



1. Cryptographic keys have to be updated for an acquirer. The agent initiates a key exchange for this acquirer by sending a *KeyExchangeInitiation* (MessageFunction: KeyExchangeRequest) message to the acquirer containing the new generated keys.
2. The acquirer updates the keys sent by the agent, and sends a *KeyExchangeResponse* (MessageFunction: KeyExchangeResponse) message to the agent.
3. The agent validates the key exchange by sending a *KeyExchangeInitiation* (MessageFunction: KeyExchangeRequest) message to the acquirer requesting a key verification.
4. The acquirer computes the values to validate the key update and sends them in *KeyExchangeResponse* (MessageFunction: KeyExchangeResponse) message to the agent.

### Key Exchange Initiated by the Issuer



1. In this business process, an issuer must update cryptographic keys shared with an agent. The issuer then initiates a key exchange for this agent sending a *KeyExchangeInitiation* (MessageFunction: KeyExchangeRequest) message to the agent containing the new generated keys.
2. The agent updates the keys sent by the issuer, and sends a *KeyExchangeResponse* (MessageFunction: KeyExchangeResponse) message to the issuer.
3. The issuer validates the key exchange by sending a *KeyExchangeInitiation* (MessageFunction: KeyExchangeRequest) message to the agent requesting a key verification.
4. The agent computes the values to validate the key update and sends them in *KeyExchangeResponse* (MessageFunction: KeyExchangeResponse) message to the issuer.

### Reject of an Authorisation Request by an Issuer



1. An *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) message is sent by the acquirer to the agent to request authorisation.
2. The *AcquirerAuthorisationInitiation* (MessageFunction: AuthorisationRequest) is forwarded by the agent to the issuer to request authorisation.
3. The issuer cannot process the received message because the MAC (message Authentication Code) is invalid. The issuer sends an *AcquirerRejection* (MessageFunction: IssuerReject) to inform the agent that the *AcquirerAuthorisationInitiation* message has been rejected.
4. An *AcquirerAuthorisationResponse* (MessageFunction: AuthorisationResponse) is returned by the agent to inform the acquirer that the transaction has been declined for technical reasons.

### Reject of an Authorisation Response by an Agent



1. The acquirer sends an *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) to the agent for authorisation and clearing.
2. The agent forwards the *AcquirerFinancialInitiation* (MessageFunction: FinancialRequest) to the issuer for authorisation and clearing.
3. The issuer approves the authorisation, updates the reconciliation totals and sends an *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) to the agent to inform the acquirer about the success of the request.
4. The agent cannot process the received message because the MAC (message Authentication Code) is invalid. The agent sends first an *AcquirerFinancialResponse* (MessageFunction: FinancialResponse) to the acquirer inform him that the transaction has been declined for technical reasons.
5. The agent then sends an *AcquirerRejection* (MessageFunction: AgentReject) message to inform the issuer that the *AcquirerFinancialResponse* message has been rejected.
6. The agent initiates the reversal of the transaction by sending an *AcquirerReversalInitiation* (MessageFunction: FinancialReversalAdvice) to request the issuer to initiate the reversal
7. The issuer sends back an *AcquirerReversalResponse* (MessageFunction: FinancialReversalAdviceResponse) to the agent to acknowledge the successful reversal of the transaction.

# Examples

Examples of various *MessageDefinitions* will be found in the ATICA Message Usage Guide.

# Revision Record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Description** | **Sections affected** |
| 1.0 |  | ISO TC68/SC7/TG1 | First version | All |
| 1.1 | December 2015 | ISO 20022 RA | Final version for publication |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Disclaimer**:

Although the Registration Authority has used all reasonable efforts to ensure accuracy of the contents of the iso20022.org website and the information published thereon, the Registration Authority assumes no liability whatsoever for any inadvertent errors or omissions that may appear thereon. Moreover, the information is provided on an "as is" basis. The Registration Authority disclaims all warranties and conditions, either express or implied, including but not limited to implied warranties of merchantability, title, non-infringement and fitness for a particular purpose.

The Registration Authority shall not be liable for any direct, indirect, special or consequential damages arising out of the use of the information published on the iso20022.org website, even if the Registration Authority has been advised of the possibility of such damages.

**Intellectual Property Rights**:

The candidate ISO 20022 *MessageDefinitions* described in this document were contributed by ISO TC68/SC7/TG1. The ISO 20022 IPR policy is available at www.ISO20022.org > About ISO 20022 > Intellectual Property Rights.

1. Available on www.epasorg.eu [↑](#footnote-ref-1)
2. Cash withdrawal is following similar message flows of BusinessTransactions. [↑](#footnote-ref-2)