



# **TW International Online Financial Institution Maintenance Interface (FIMI)**

**February 15, 2023**

Copyright © 2023 Compass Plus (Great Britain) Limited. All rights reserved.

*This document is the property of Compass Plus (Great Britain) Limited and the information contained herein is confidential. The material contained in this document is protected under International Copyright Laws and Treaties. Any unauthorized reprint or use of this material to create other work is prohibited. No part of this document may be reproduced, disclosed, distributed, transmitted or otherwise used in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system without express written permission from Compass Plus (Great Britain) Limited, or, if any part hereof is furnished by virtue of a contract between Compass Plus (Great Britain) Limited and a third party, as expressly authorized under that contract.*

*TW INTERNATIONAL™ and TX INTERNATIONAL™ are registered trademarks of  
Compass Plus (Great Britain) Limited.*

## Terms and Abbreviations

Term	Abbreviation
TW International Online	TWO
TW International CMS	TWCMS

## Revision History

Date	Version	TWO Version	Description of Changes	Internal Code
02.12.2022 – 21.12.2022	16.25	5.3.45.2	Added the <i>EC3DSVersion</i> and <i>ECI</i> fields to the <i>StandAloneReversal</i> operation request.	TWO-75621
			Added the <i>Currency</i> , <i>TermRetailerName</i> , <i>CurrencyISOCode</i> , <i>OrigCurrencyISOCode</i> fields to the <i>Statement</i> structure of the <i>GetAcctStatement</i> operation response	TWO-76484
			Added the <i>CardUID</i> , <i>TermCountry</i> , <i>TermCity</i> , <i>OnlineIssuerFee</i> fields to the <i>Statement</i> structure of the <i>GetCardStatement</i> operation.	
			Added the <i>KeyType</i> field to the <i>GetWorkingKey</i> operation request, added the <i>PublicKey</i> field to the operation response.	TWO-75787
			Added the <i>ZPK</i> field to the requests of the <i>GetPVV_PINOffset</i> and <i>CheckPVV_PINOffset</i> operations.	
			Detailed the description in section 6.1.	
17.11.2022	16.25	5.3.45.1	Added the <i>EC3DSNetwork</i> field to the <i>POSRequest</i> operation request.	TWO-76335
			Added available value 23 – <i>Region code/name</i> for the <i>FieldType</i> field of the <i>ExtPSNames</i> structure of the <i>CreateRetailer</i> operation.	TWO-76769
			Added available value 6 – <i>Region code/name</i> for the <i>FieldType</i> field of the <i>ExtPSNames</i> structure of the <i>CreateTerminal</i> operation.	
			Changed the format and added new available values for the <i>AuthRespCodeCategory</i> field in the <i>POSRequest</i> operation response.	SPD-1175
			Added the <i>B2B Pass(805)</i> transaction to the <i>POSRequest</i> request.	TWO-74558
			Added the new invoice type "8 – outgoing request within the FPS B2B service" to the <i>GetInvoice</i> request.	
			Added the new service ID "4 – NSPK FPS B2B" for the <i>ServiceId</i> field in the <i>GenerateQRCode</i> , <i>GetQRCodeInfo</i> , <i>GetQRCodePayload</i> operations.	
			Added the <i>TaxType</i> , <i>TaxValue</i> fields to the <i>GenerateQRCode</i> operation request.	
			Added the <i>TaxType</i> , <i>TaxValue</i> , <i>UIP</i> fields to the <i>QRCodeList</i> structure of the <i>GetQRCodeInfo</i> operation response.	

Date	Version	TWO Version	Description of Changes	Internal Code
27.10.2022 – 16.11.2022	16.24	5.3.44.3	Added the <i>Currencies</i> field to the <i>List</i> structure of the <i>GetServiceMembersList</i> operation response.	TWO-75837
			Added the <i>RetailerServicePhone</i> , <i>RetailerURL</i> fields to the <i>Statement</i> structure of the responses of the <i>GetCardStatement</i> and <i>GetAcctStatement</i> operations.	TWO-75899 SPD-1139
			Added the <i>ServicePhone</i> , <i>URL</i> fields to the requests of the <i>CreateRetailer</i> and <i>UpdateRetailer</i> operations.	
			Added the following fields to the <i>AliasService</i> structure of the responses of the <i>GetCardStatement</i> and <i>GetAcctStatement</i> operations: <i>SenderCountry</i> , <i>RipientCountry</i> , <i>DebtorAmount</i> , <i>DebtorCurrency</i> , <i>CreditorAmount</i> , <i>CreditorCurrency</i> , <i>Fee</i> , <i>PaymentDetailsCategoryCode</i> . Added the <i>CountryOfResidence</i> attribute to the <i>RecipientData</i> structure of the responses of the <i>POSRequest</i> and <i>AliasVerification</i> operations.	TWO-76115 SPD-1174
09.09.2022 – 06.10.2022	16.24	5.3.44.1	Added the <i>SCAIndicators</i> structure to the <i>POSRequest</i> operation response.	TWO-74989
			Added the <i>Card Token</i> issuer history object in section 3.17.	TWO-75209
			Removed the <i>ECNeedDecoupledAuth</i> field in the <i>GetCardInfo</i> and <i>SetCard3DSecureAuthentication</i> operations.	TWO-75263 SPD-1104
			Added the <i>BranchId</i> field to the responses of the <i>GetAcctInfo</i> , <i>GetCardInfo</i> operations.	TWO-73429
			Added the <i>Branch</i> field to the responses of the <i>GetPersonInfo</i> , <i>GetRetailerList</i> operations.	
			Added the <i>BranchName</i> field to the <i>GetTermList</i> operation response.	
			Changed the description of the <i>Provider</i> field in the <i>PrepareProvisioningData</i> operation, the field is used for Visa Issuer Wallet.	TWO-74939
			Added the <i>MemberType</i> field to the request of the <i>GetServiceMembersList</i> operation and the <i>BICSwift</i> , <i>Country</i> , <i>Language</i> fields to the <i>List</i> structure of the operation response.	TWO-74854 SPD-1147
			Added the <i>C2CI Pass(872)</i> transaction to the <i>POSRequest</i> request.	
			To the <i>GetInvoice</i> request, added the new invoice type 7 – outgoing request within the C2C International FPS service.	
			Added the <i>AliasVerification</i> operation.	

Date	Version	TWO Version	Description of Changes	Internal Code
			Changed the maximum size of the <i>NewPassword</i> , <i>OldPassword</i> fields from 16 to 64 in the <i>ChangeAlphaNumericPassword</i> operation.	TWO-75440
25.08.2022	16.23	5.3.43.2	Added the <i>SubMerchantCheck</i> , <i>SubMerchantData</i> , <i>MCCCheckList</i> fields and <i>Facilitators</i> structure to the <i>CreateRetailer</i> operation request. Added the available value <i>MCC(22)</i> for the <i>FieldType</i> request field of the <i>ExtPSNames</i> structure of the <i>CreateRetailer</i> operation.	TWO-74881 SPD-1129
09.08.2022	16.23	5.3.43.2	Added the <i>CalcIssuerFee</i> , <i>CalcFeeTranId</i> fields to the <i>POSRequest</i> operation request. Added the <i>IssuerFee</i> , <i>AccountCurrency</i> fields to the <i>POSRequest</i> operation response.	TWO-73997 SPD-1121
			Added the <i>NotificationQRSubscription</i> operation. Added the <i>LegalEntityExtId</i> field to the <i>QRCodeList</i> structure of the <i>GetQRCodeInfo</i> operation response.	TWO-74381 SPD-1123
			Changed the size of the <i>FullName</i> , <i>Email</i> , <i>TaxReasonRegCode</i> , <i>OperatorTaxReasonRegCode</i> , <i>TransferCode</i> , <i>TransferAmount</i> fields in the <i>P2PData</i> structure of the <i>POSRequest</i> operation.	TWO-74740 SPD-1030
			Added the <i>ExtRRN</i> field to the <i>RelatedTran</i> structure of the <i>POSRequest</i> operation response.	TWO-74683
15.07.2022 – 06.08.2022	16.23	5.3.43.1	Added the <i>RecurringData</i> field to the <i>POSRequest</i> operation request.	TWO-74190 SPD-1105
			Changed the size of the <i>LegalEntityId</i> field from 9 to 16 characters in the following operations: <i>GetAcctInfo</i> , <i>SetAcctLegalEntity</i> , <i>GetLegalEntityInfo</i> , <i>CreateLegalEntity</i> , <i>DeleteLegalEntity</i> , <i>UpdateLegalEntity</i> , <i>SetLegalEntityUserFields</i> .	TWO-74229 SPD-1089
			Changed the description of the <i>OrigTranId</i> request field of the <i>POSRequest</i> operation.	TWO-74198
			Changed the information on whether or not the <i>Type</i> field must be filled in the <i>CreateTerminal</i> operation request.	TWO-74428
			Added the <i>BackOfficeId</i> , <i>InstName</i> fields to the requests of the <i>UpdateTerminal</i> and <i>UpdateRetailer</i> operations.	TWO-74350
20.05.2022 – 16.06.2022	16.22	5.3.42.1	Added the <i>EC3DSTranCondition</i> field to the <i>POSRequest</i> transaction request.	TWO-73272 SPD-1066
			Added the <i>CreateRetailer</i> , <i>CreateTerminal</i> operations.	TWO-71475 SPD-850
			Extended the list of allowed bits from 16 to 24 for the <i>Capabilities</i> field transferred in the <i>GetServiceMembersList</i> operation response.	TWO-73530

Date	Version	TWO Version	Description of Changes	Internal Code
			<p>Added the <i>ImpactData</i> field to the operations header.</p> <p>Allowed the following cardless transactions in the <i>POSRequest</i> operation: <i>POS P2P Debit(132)</i>, <i>POS P2P Pass(135)</i>, <i>POS Transfer Pass(149)</i>, <i>POS Payment Pass(175)</i>, <i>POS Exchange Rate Inquiry(194)</i>.</p>	TWO-73161
			<p>In section 3.20, added the new card entry mode <i>Consumer-Presented QR (03)</i>.</p> <p>Added the <i>GetConsumerPresentedQRCode</i> operation.</p>	TWO-73691
			<p>Added the <i>TrustedMerchants</i> structure in the <i>GetECMerchant</i> operation response.</p> <p>Added the <i>PAN</i>, <i>MBR</i>, <i>CardUID</i> fields in the <i>ChangeECMerchantStatus</i> operation request.</p>	TWO-73521
			<p>In the <i>POSRequest</i> operation, allowed the transfer of the <i>StandAlone</i> transaction attribute in the <i>StandAlone</i> and <i>TranCategory</i> request field for transaction 110 – <i>Purchase</i>.</p>	TWO-73752
05.05.2022	16.21	5.3.41.3	<p>Added the <i>GovernmentService</i> structure in the response of the <i>GetCardStatement</i> and <i>GetAcctStatement</i> operations.</p>	TWO-73251 SPD-1065
			<p>Increased the size of the <i>FromAmount</i>, <i>ToAmount</i> fields in the <i>GetRates</i>, <i>AddRate</i>, <i>ModifyRate</i> operations.</p>	TWO-73294 SPD-1069
11.04.2022	16.21	5.3.41.2	Added the <i>KZO</i> field in the <i>CheckQRCodeStatus</i> operation response.	TWO-72873
10.03.2022 – 24.03.2022	16.21	5.3.41.1	<p>Added the <i>ResetPassword</i> operation.</p> <p>Added the <i>MWLRequested</i> field in the <i>Invoices</i> structure of the <i>GetInvoice</i> operation response.</p> <p>Added the <i>MWLAccepted</i> field in the <i>ChangeInvoiceStatus</i> operation request.</p>	TWO-71463 SPD-1005 TWO-72407 SPD-1033
			<p>In section 3.9, added new authorizer response codes 29 – Closed Account, 30 – Blocked.</p>	TWO-72531
			<p>Added the <i>GetATMWithdrawalHold</i>, <i>CreateATMWithdrawalHold</i>, <i>ChangeStatusATMWithdrawalHold</i> operations.</p>	TWO-71502
01.03.2022	16.20	5.3.40.3	<p>Extended the list of available values for the <i>DraftCapture</i> field in the <i>POSRequest</i> and <i>GetTransInfo</i> operations.</p> <p>Detailed the description for the <i>PAN2</i> and <i>OrigTranId</i> fields of the <i>POSRequest</i> operation.</p>	TWO-71917

Date	Version	TWO Version	Description of Changes	Internal Code
16.02.2022	16.20	5.3.40.2	Extended the list of addresses for EMV scripts in section 3.3.9.	TWO-71377
12.02.2022	16.20	5.3.40.2	<p>Added the new QR code type QR-Subscription(5) in the <i>GenerateQRCode</i>, <i>GetQRCodeInfo</i>, and <i>GetQRCodePayload</i> operations.</p> <p>Added the <i>SubscriptionPurpose</i> field in the <i>GenerateQRCode</i> operation request.</p> <p>Added the <i>SubscriptionPurpose</i>, <i>TaxPayerNumber</i>, <i>OGRN</i> fields in the <i>QRCodeList</i> structure of the <i>GetQRCodeInfo</i> operation response.</p> <p>Added the <i>SubscriptionToken</i>, <i>MemberId</i>, <i>SubscriptionStatus</i>, <i>PaymentRequestStatus</i> fields in the <i>CheckQRCodeStatus</i> operation response.</p> <p>Supported the new type of <i>Invoice</i> in the <i>GetInvoice</i> operation.</p> <p>Added the <i>PullQRRequest</i> operation.</p> <p>Added the <i>AliasSubscriptions</i> structure in the <i>GetAliasInfo</i> operation response.</p> <p>Added the <i>SetAliasSubscription</i> operation.</p>	SPD-1028 TWO-70806
11.02.2022	16.20	5.3.40.2	Added the <i>RefExternalTranId</i> field in the <i>QRService</i> structure of the responses of the <i>GetAcctStatement</i> and <i>GetCardStatement</i> operations.	TWO-71764
10.02.2022	16.20	5.3.40.2	<p>Added the <i>AuthRespCodeCategory</i> and <i>Network</i> fields in the <i>POSRequest</i> operation response.</p> <p>Added section 3.23.</p>	SPD-1018 TWO-71696
10.02.2022	16.20	5.3.40.2	<p>In the <i>POSRequest</i> operation request, it is now allowed to transfer sender data in the <i>Subject</i> field of the <i>P2PData</i> structure.</p> <p>Added the <i>AFTCDK</i>, <i>OCTCDK</i> subfields in the <i>TerminalAdditionalData</i> field of the <i>POSRequest</i> operation request.</p>	SPD-987 TWO-71888
02.02.2022	16.20	5.3.40.2	<p>Added the <i>LegalEntityId</i>, <i>LegalEntityName</i> fields in the <i>GetAcctInfo</i> operation response.</p> <p>Added the following operations: <i>SetAcctLegalEntity</i>, <i>GetLegalEntityInfo</i>, <i>CreateLegalEntity</i>, <i>UpdateLegalEntity</i>, <i>DeleteLegalEntity</i>, <i>SetLegalEntityUserFields</i>.</p>	TWO-70686
27.01.2022	16.20	5.3.40.1	Added the <i>FitnessStatus</i> , <i>ReservedCash</i> , <i>AvailableCash</i> fields in the <i>Cassettes</i> structure of the <i>GetATMHoppers</i> operation response.	TWO-71201 SPD-995
25.01.2022	16.20	5.3.40.1	Added the <i>OrigTermInstName</i> , <i>OrigTermName</i> , <i>OrigRetailerName</i> fields in the <i>POSRequest</i> operation request.	TWO-71308 SPD-994

Date	Version	TWO Version	Description of Changes	Internal Code
25.01.2022	16.20	5.3.40.1	Added the <i>TranCategory</i> field in the <i>POSRequest</i> operation request. Added the <i>ExtExpDate</i> field in the <i>POSRequest</i> operation response. Marked the <i>StandAlone</i> field of the <i>POSRequest</i> operation request as obsolete.	TWO-70965
14.01.2022	16.20	5.3.40.1	Added the <i>CAVVOK</i> field in the <i>POSRequest</i> operation request.	TWO-71113
22.12.2021	16.19	5.3.39.3	Added the <i>CardAllowedEMVScript</i> field in the <i>GetCetCardInfo</i> operation response. Added the <i>AuthRespCode</i> array in the <i>GetTransInfo</i> operation request.	TWO-71166 SPD-978 TWO-70741
03.12.2021 – 20.12.2021	16.19	5.3.39.2	Added the method of communication via the protocol using the Kafka message broker. Updated the description of section 1.  Added the <i>Email</i> field in the <i>Info</i> response structure of the <i>GetPersonInfo</i> operation. Added the <i>Email</i> field in the <i>CreatePerson</i> operation request. Added the <i>Email</i> field in the <i>UpdatePerson</i> operation request. Added the <i>PersonEmail</i> field in the <i>CreateIssuerObjects</i> operation request.  Added the <i>IssuerName</i> , <i>CardBrandName</i> fields in the <i>Cards</i> structure of the <i>CheckAlias</i> operation response.  Added the <i>ExtendedAccountNumber</i> field in the <i>GetAcctInfo</i> operation response. Added the <i>ExtendedAccountNumber</i> field in the <i>CreateAccount</i> operation request. Added the <i>GetAcctTokensInfo</i> , <i>SetAcctTokenStatus</i> , <i>CreateAcctToken</i> operations.  Extended the list of allowed bits for the <i>MerchantInitiatedTran</i> field in the <i>StandAloneReversal</i> and <i>POSRequest</i> request.	TWO-70035 SPD-946 TWO-69642 TWO-70384 TWO-70595 SPD-981 TWO-70962 SPD-927
11.11.2021 – 25.11.2021	16.19	5.3.39.1	Detailed the description of the <i>Track2</i> field in the <i>POSRequest</i> operation request.  Added the <i>NewPVKI</i> field in the <i>SetCardPVV</i> operation request.  Added the <i>NeedNotify</i> field in the <i>SetCardPVV</i> operation request.	TWO-70238 TWO-60496 SPD-593 TWO-70671

Date	Version	TWO Version	Description of Changes	Internal Code
			<p>Added the <i>LastNameNat</i>, <i>FirstNameNat</i>, <i>MiddleNameNat</i> fields in the <i>Info</i> response structure of the <i>GetPersonInfo</i> operation.</p> <p>Added the <i>LastNameNat</i>, <i>FirstNameNat</i>, <i>MiddleNameNat</i> fields in the <i>CreatePerson</i> operation request.</p> <p>Added the <i>LastNameNat</i>, <i>FirstNameNat</i>, <i>MiddleNameNat</i> fields in the <i>UpdatePerson</i> operation request.</p>	TWO-69836 SPD-957
			<p>Added the <i>BranchId</i> and <i>InstName</i> fields in the <i>CreateAccount</i> operation request.</p> <p>Added the <i>PersonBranchId</i> field in the <i>CreateIssuerObjects</i> operation request.</p>	TWO-70817
			Added the <i>LegalEntityData</i> structure in the <i>POSRequest</i> operation response.	TWO-70779 SPD-979
			Added the <i>DeviceIndex</i> field in the <i>CreateCardToken</i> operation request.	TWO-70905
03.11.2021	16.18	5.3.38.3	Updated the list of transactions in section 3.8.	
18.10.2021 – 21.10.2021	16.18	5.3.38.2	Added the <i>ExternalTranId</i> field in the <i>QRService</i> structure in the response of the <i>GetAcctStatement</i> and <i>GetCardStatement</i> operations.	TWO-70287
			Added the <i>ExtRespCode</i> field in the <i>POSRequest</i> operation request.	TWO-70162
03.09.2021 – 29.09.2021	16.18	5.3.38.1	Supported the new QR code type in the <i>GenerateQRCode</i> and <i>GetQRCodePayload</i> operations: extended the list of available values of the <i>QRCodeType</i> field.	TWO-69736
			Added the <i>QRActivationId</i> field in the request of the <i>GetQRCodePayload</i> and <i>CheckQRCodeStatus</i> operations, added the <i>QRActivationId</i> field in the <i>QRPayloads</i> structure of the <i>GetQRCodePayload</i> operation response.	
			Extended the list of available values for the <i>QRStatus</i> response field of the <i>CheckQRCodeStatus</i> operation.	
			Added the <i>ModifyQRCode</i> operation.	
			Added the new <i>TranCode</i> 860 – C2G in the <i>POSRequest</i> operation. Added the <i>Invoiceld</i> field in the <i>POSRequest</i> operation response.	TWO-69071
			Added new available values for the <i>InvoiceType</i> field in the <i>GetInvoice</i> operation.	
			Added the <i>Permission</i> field in the <i>CreateAlias</i> operation request.	TWO-69520 SPD-943

Date	Version	TWO Version	Description of Changes	Internal Code
			<p>Added the <i>Account</i> and <i>AccountUID</i> fields in the <i>UpdateAlias</i> operation request.</p> <p>Added the new type of Extentions with the 'C2GRequisites' identifier in the <i>MessageExtentions</i> structure of the <i>POSRequest</i> operation request.</p>	
			<p>Added the invoice status 6 – "Confirmed by institution" in the <i>GetInvoice</i> and <i>ChangeInvoiceStatus</i> operations.</p>	TWO-69847 SPD-929
			<p>Added the <i>RedirectURL</i> field in the <i>GenerateQRCode</i> operation request. Added the <i>RedirectURL</i> and <i>AgentId</i> fields in the <i>QRCodeList</i> structure of the <i>GetQRCodeInfo</i> operation response.</p>	TWO-69752
			<p>Added the <i>PaymentTranId</i> and <i>FinalAuthRespCode</i> fields in the <i>Invoices</i> structure of the <i>GetInvoice</i> operation response.</p>	TWO-69928
			<p>Added the <i>CSCLen</i> and <i>CardVerificationType</i> fields in the <i>GetCVV</i> operation request. Added the <i>CardVerificationType</i> field in the <i>GetCVV</i> operation response. Changed the maximum size of the field in the <i>GetCVV</i> operation response for the <i>CVV</i> and <i>StrCVV</i> fields.</p>	TWO-69513 SPD-942
03.09.2021	16.17	5.3.37.3	Added the <i>SubMerchantRegion</i> field in the <i>POSRequest</i> operation request.	TWO-69541 SPD-939
03.08.2021	16.17	5.3.37.2	Added the new available value "6 – Visa Issuer Wallet" for the <i>WalletType</i> field of the <i>PrepareProvisioningData</i> operation request.	TWO-69058
12.07.2021 – 22.07.2021	16.17	5.3.37.1	In the <i>POSRequest</i> and <i>MerchantRefund</i> operations, added the <i>NSPK_FPS_125</i> subfield in the <i>Data</i> field of the <i>MessageExtentions</i> structure for the 'FPSScoring' extension.	TWO-68219
			Added the <i>RetailerAccount</i> field in the <i>GenerateQRCode</i> operation request.	SPD-748 TWO-68134
			Added the <i>Mode</i> field in the <i>GetAliasInfo</i> operation request.	TWO-68667 SPD-889
			Added the <i>TaxId</i> , <i>TaxReasonRegCode</i> , <i>OperatorTaxId</i> , <i>OperatorTaxReasonRegCode</i> subfields in the <i>P2PData</i> structure of the <i>POSRequest</i> operation.	TWO-68755 TWO-67451
			Added the <i>TokenRequestorName</i> field in the <i>Tokens</i> structure of the <i>GetCardTokensInfo</i> operation response.	TWO-66808
			Added the <i>SubMerchantTaxReasonRegCode</i> field in the <i>POSRequest</i> operation request. Added new subfields <i>VTAXID</i> , <i>VTRR</i> in the	TWO-68902

Date	Version	TWO Version	Description of Changes	Internal Code
			TerminalAdditionalData structure of the POSRequest operation.	
			Added the PlasticTypesList structure in the InitSession operation response. Added the UpdateCardMetadata, TokenInquiry operations.	TWO-66987 SPD-813
			Added the AliasType field in the UpdateAlias operation request.	TWO-68932 SPD-896
28.06.2021 – 05.07.2021	16.16	5.3.36.3	Added the incoming field SubMerchantTermId in the POSRequest request.	TWO-68118
			Added the TransferCode, TransferAmount, TransferCurrencyCode subfields in the P2PData structure of the POSRequest operation.	SPD-848 TWO-68199
			Added new transactions (TranCode) POS ECPurchase(856), POS ECRefund(857) in the POSRequest operation. Added the BSktID subfield in the TerminalAdditionalData structure of the POSRequest operation.	SPD-853 TWO-67901
			Added the incoming fields Account and AccountUID in the ChangeInvoiceStatus operation.	SPD-865 TWO-68373
			Added the new device type 17 – wearable device in the GetDevicesList, GetCardTokensInfo, CreateCardToken operations.	SPD-849 TWO-68309
04.06.2021 – 20.06.2021	16.16	5.3.36.2	Added the incoming field SubMerchantTaxId in the POSRequest request.	SPD-855 TWO-67558
			Added the incoming fields Account2, Account2UID in the CreateAlias and UpdateAlias requests. Added the Account2 and Account2UID fields in FIMI/GetAliasInfo/Rp/Aliases.	SPD-856 TWO-67516
			Extended the list of available values for the EC3DSVersion request field in the POSRequest operation: added value 4 (3D Secure Version 2.3).	TWO-68094
			Detailed the description of the TIPAmount, PAN2, OrigTranId request fields for the POSRequest operation.	TWO-66620
27.05.2021	16.16	5.3.36.1	Extended the list of available values for the P2PSourceOfFunds field of the POSRequest operation: added value 10 (Tip).	
			Detailed the description of the AcqObjInstName and AcqObjName fields for the ResetCumulativeLimit, UpdateCumulativeLimit, AddCumulativeLimit, GetCumulativeLimits, DeleteCumulativeLimit operations.	TWO-65896

Date	Version	TWO Version	Description of Changes	Internal Code
13.05.2021	16.15	5.3.35.3	Detailed the description of the <i>FIMI/PrepareProvisioningData/Rq/Nonce</i> field as ApplePay is supported for the NSPK cards.	TWO-67329
22.04.2021	16.15	5.3.35.2	Added the <i>POS QR Purchase Debit(838)</i> , <i>POS QR Purchase Credit(839)</i> , <i>POS QR Purchase Pass(840)</i> transactions in the <i>POSRequest</i> operation.	TWO-66152 SPD-785
19.04.2021	16.15	5.3.35.2	Added the <i>TIPAmount</i> field in the <i>Statement</i> structure of the response of the <i>GetCardStatement</i> and <i>GetAcctStatement</i> operations.	TWO-67257 SPD-814
19.04.2021	16.15	5.3.35.2	Added the <i>ProtectedAmount</i> field in the <i>Statement</i> structure of the response of the <i>GetCardStatement</i> and <i>GetAcctStatement</i> operations.	TWO-67093 SPD-724
12.04.2021	16.15	5.3.35.2	Added the following operations: <i>GetRetailerLimits</i> , <i>SetRetailerLimits</i> , <i>ResetRetailerLimitCounters</i> , <i>GetRetailerSchemeLimits</i> , <i>GetRetailerGroupLimits</i> , <i>SetRetailerGroupLimits</i> , <i>ResetRetailerGroupLimitCounters</i> , <i>GetRetailerGroupSchemeLimits</i> .  Changed the list of available values for the <i>Type</i> and <i>Editable</i> fields of the <i>Limits</i> structure of the <i>InitSession</i> operation response.	SPD-761 TWO-66166
12.03.2021 – 31.03.2021	16.15	5.3.35.1	Added the <i>PINVerifyType</i> field in the <i>GetCardInfo</i> operation response. Added the <i>PINVerifyType</i> field in the <i>CreateCard</i> operation request.	TWO-65837
			Added the <i>ImpactProtectedAmount</i> field in the <i>AcctDebit</i> operation request. Added the <i>ProtectedAmountDelta</i> and <i>ProtectedAmount</i> fields in the <i>AcctDebit</i> operation response. Excluded values 5 and 6 for the <i>Force</i> request field of the <i>AcctDebit</i> operation.  Added the <i>ProtectedAmountDelta</i> and <i>ProtectedAmount</i> fields in the <i>SetAcctProtectedAmount</i> operation response.	TWO-66581
			Extended the list of available values of the <i>FileFormat</i> field in the <i>List</i> structure of the <i>GetReportRequests</i> operation and <i>FileFormat</i> field of the <i>ExecuteReport</i> operation request.	TWO-66544
04.02.2021	16.14	-	Detailed the format of the <i>FPTI</i> field of the <i>POSRequest</i> request.	TWO-66127
19.01.2021 – 28.01.2021	16.14	5.3.34.1	Added the <i>Alias</i> field in the <i>GetAliasInfo</i> operation request.  Added the <i>Permission</i> field in the <i>Aliases</i> structure of the <i>GetAliasInfo</i> operation response, added the <i>AllowAliaseMemberList</i> structure.  Extended the list of available values of the <i>InvoiceType</i> field of the <i>GetInvoice</i> operation	TWO-64810 SPD-734

Date	Version	TWO Version	Description of Changes	Internal Code
			<p>request. Added the <i>Alias</i>, <i>Account</i>, and <i>AccountUID</i> fields in the <i>GetInvoice</i> operation request. Added the <i>Account</i>, <i>AccountUID</i>, <i>Alias</i>, <i>B/C</i> fields in the <i>Invoices</i> structure of the <i>GetInvoice</i> operation response.</p> <p>Added transaction 835 – <i>C2CPullPass</i> to the <i>POSRequest</i> operation request within the <i>TranCode</i> field.</p> <p>Added the <i>SetAliasPermission</i> operation.</p> <p>Added the <i>Permission</i> field in the <i>UpdateAlias</i> operation request, added available value 3 – <i>NSPK FPS C2C</i> for the <i>ServiceId</i> field.</p>	
			Added the <i>BAI</i> field in the <i>Statement</i> structure of the responses of the <i>GetAcctStatement</i> and <i>GetCardStatement</i> operations.	TWO-65729 SPD-763
06.01.2021 – 11.01.2021	16.13	5.3.33.3	<p>Added the <i>ProtectedAmount</i> field in the <i>GetAcctInfo</i> operation response.</p> <p>Added the <i>SetAcctProtectedAmount</i> operation.</p> <p>Extended the list of available values of the <i>Force</i> request field of the <i>AcctDebit</i> operation.</p> <p>Updated the list of available values in the dictionary 3.7. Account Operation Codes.</p> <p>Changed the size of the following fields providing personal payment attributes from 1000 to 4000 characters:</p> <p>FIMI/GetCardInfo/Rp/PersonalPayments.CustAcct,  FIMI/GetCardInfo/Rp/TBVendors.CustVendorAcct,  FIMI/GetCardInfo/Rp/TBTransfers.AdditionalAttributes,  FIMI/GetCardAutopayments/Rp&gt;List.CustAcct,  FIMI/GetCardAutopaymentHistory/Rp/Log.CustAcct,  FIMI/AddCardVendor/Rq/CustAcct,  FIMI/DelCardVendor/Rq/CustAcct,  FIMI/AddTBTransfer/Rq/AdditionalAttributes,  FIMI/AddTBPayment/Rq/CustAcct,  FIMI/DelTBPayment/Rq/CustAcct,  FIMI/AttachAutopaymentToCard/Rq/CustAcct,  FIMI/EditCardAutopayment/Rq/CustAcct,  FIMI/POSRequest/Rq/ToAccount2,  FIMI/StandaloneReversal/Rq/ToAccount2.</p>	TWO-64952 SPD-724
				TWO-65424 SPD-766

## Table of Contents

<b>Revision History.....</b>	<b>4</b>
<b>1 Overview.....</b>	<b>17</b>
1.1 Online Mode .....	17
1.2 Offline Mode .....	20
<b>2 FIMI Protocol Description .....</b>	<b>22</b>
2.1 Operations .....	22
2.1.1 Service Operations.....	22
2.1.2 Issuing Operations .....	23
2.1.3 Acquiring Operations .....	31
2.1.4 Transaction Requests.....	33
2.1.5 POS Transactions .....	33
2.1.6 Reports.....	33
2.1.7 Stoplists Maintenance .....	33
2.1.8 Visa Administrative Transactions.....	33
2.1.9 MasterCard Global Administrative Transactions .....	34
2.1.10 Administrative Dispute Transactions of the Local Payment System.....	34
2.1.11 System Units.....	34
2.1.12 Cryptographic Operations .....	34
2.1.13 User Operations .....	35
2.2 Set of Field Types .....	36
2.2.1 Integer Type (Int).....	36
2.2.2 Real Number Type (Num) .....	37
2.2.3 Date Type (Time) .....	37
2.2.4 String Type (Str) .....	37
2.2.5 String in Hexadecimal Format (Hex) .....	38
2.2.6 String with UAMP Structure (StrUAMP) .....	39
2.2.7 String with XML Document (StrXML) .....	40
2.2.8 Type of Personal Payment Attributes (CustAcct) .....	41
2.2.9 Array of Structures (ArrRec) .....	42
2.3 Message Contents .....	44
2.3.1 Common Header .....	44
2.3.2 Service Messages .....	47
2.3.3 Issuer Requests .....	71
2.3.4 Acquiring Requests .....	275
2.3.5 Requests on Transactions.....	344
2.3.6 Reports.....	404
2.3.7 Stoplists Maintenance .....	407
2.3.8 Visa Administrative Transactions.....	417
2.3.9 MasterCard Global Administrative Transactions .....	420
2.3.10 Administrative Dispute Transactions in the Local Payment System.....	423
2.3.11 System Units.....	424
2.3.12 Cryptographic Operations .....	429
2.3.13 User Operations .....	443
<b>3 Reference Data.....</b>	<b>444</b>
3.1 Available Values of Response Field .....	444
3.1.1 Translation of Authorizer Response Codes to Response Field Values .....	446
3.2 Card Status.....	449
3.3 Card Limits.....	450
3.4 Terminal Types .....	451
3.5 ATM Device State .....	452
3.6 ATM Media State .....	453
3.7 Account Operation Codes .....	454

3.8	Transaction Codes .....	457
3.9	Authorizer Response Codes .....	479
3.10	ATM Control Commands.....	484
3.11	POS Terminal Control Commands .....	485
3.12	Transaction Types .....	486
3.13	Visa Stoplist Regions .....	487
3.14	Mastercard Europe/Global Stoplist Regions.....	488
3.15	Messaging Service Command Codes .....	489
3.16	Messaging Service Command Decline Reason Codes .....	490
3.17	Issuer History Objects and Attributes .....	492
3.17.1	Objects Description .....	492
3.17.2	Attributes Description .....	493
3.18	System Unit Dictionaries .....	498
3.18.1	Unit, Terminal Types and Unit Subtypes .....	498
3.18.2	Unit States Bit Mask.....	502
3.18.3	Commands for Unit .....	507
3.19	Transaction Conditions .....	511
3.20	Card Entry Modes .....	513
3.21	Crypto Functions.....	514
3.22	Transaction Reason Codes.....	517
3.23	Networks.....	518
<b>4</b>	<b>Password and OTP Encryption Algorithm .....</b>	<b>519</b>
4.1	Transfer of Password in Clear or as Hash.....	519
4.2	Transfer of Password Using Encryption .....	520
4.3	OTP Hash Calculation Algorithm.....	523
<b>5</b>	<b>DS Algorithm for SOAP Messages.....</b>	<b>524</b>
5.1	DS Generation .....	524
5.2	Example of DS Creation.....	525
5.3	DS Verification .....	528
<b>6</b>	<b>Card PIN Setting Technology .....</b>	<b>529</b>
6.1	Working Key is Used.....	529
6.1.1	Working Key TPK .....	529
6.1.2	Working key RSA-SK .....	530
6.2	Working Key is not Used.....	532
6.3	Transactional Mode Usage .....	534
<b>7</b>	<b>CardUID and AccountUID Fields Usage .....</b>	<b>535</b>
<b>8</b>	<b>Request Examples .....</b>	<b>536</b>
8.1	Request for Working via UAMP .....	536
8.2	Request for Working via SOAP XML.....	538
8.3	Request for Working via JMS XML.....	542
<b>9</b>	<b>Examples of Files Generation for Offline Mode .....</b>	<b>543</b>
9.1	Examples for Working via UAMP .....	543
9.2	Examples for Working via SOAP.....	544
9.3	Examples for Working via XML .....	552

# 1 Overview

The protocol is designed to exchange messages between the FIMI terminal or server (e.g., web server) and **TWO**. The terminal in this specification stands for the site initiating the request to **TWO**.

For the service accepting requests from the terminal and generating responses stands *server, host, driver* or *TWO system*.

It is the initiative of the terminal to start the exchange of data by sending request to **TWO** that returns the response.

The FIMI protocol is based on:

- UAMP (Universal Application Messaging Protocol)
- SOAP XML v.1.2
- XML (when using the XML over JMS (Java Message Service))

The document provides the descriptions of all the message fields. For the UAMP message construction rules, refer to document **UAMP(E).pdf** and message examples, in section 8.1. For the SOAP XML protocol structure, see files **fimi.wsdl**, **fimi.xsd**, **fimi\_types.xsd**, **fimi\_userdefined.xsd** and request examples in section 8.2. For the XML format, the FIMI message construction rules are similar to those for the SOAP XML protocol. The only difference is that the HTTP Header and SOAP envelope (with SOAP Body and SOAP Header) are absent.

The protocol supports the transfer of text data in various encodings for the UAMP format – see the description of the field FIMI/Encoding. For the SOAP XML and XML format, UTF8 encoding is supported. Its definition should be initiated on the terminal site.

When extending the protocol on host side, new fields are added. Terminal must ignore the unknown fields received in **TWO** responses when communicating via UAMP protocol. If the SOAP protocol is used, the version of web-service description files (wsdl and xsd) must match the **TWO** version.

The messages exchange between the terminal and FIMI driver can be implemented both Online and Offline.

## 1.1 Online Mode

In the online mode, it is possible to transfer messages in the following formats:

- UAMP (via the TCP, Kafka channels)
- SOAP XML (via the HTTP, Kafka channels)
- XML (via the JMS channel).

For the UAMP format, when the terminal works via the TCP channel, it should send the UAMP messages (requests) as the specially framed TCP batches. For the list of frames supported by **TWO**, refer to document **Frame(E).pdf**. For the UAMP format, when the terminal works via the Kafka channel, it communicates with the Kafka message broker and sends the UAMP messages (requests) to the topic for incoming messages (request topic). The FIMI driver subscribes to the topic for incoming messages and receives incoming messages (requests) from the topic. After the request is processed, the FIMI driver sends the outgoing UAMP message (response) to the topic for outgoing messages (response topic) the terminal subscribes to. Use a unique value in the FIMI\Echo field to match the request and response. It is also allowed to specify a user attribute in the Kafka message header to match data by the header.

To transfer data in the SOAP XML format when working via the HTTP channel, TCP with the framing of messages (requests and response) via the HTTP protocol with the POST method is used. For the SOAP XML format, when the terminal works via the Kafka channel, it communicates with the Kafka message broker and sends the SOAP XML messages (requests) to the topic for incoming messages (request topic). The FIMI driver subscribes to the topic for incoming messages

and receives incoming messages (requests) from the topic. After the request is processed, the FIMI driver sends the outgoing SOAP XML message (response) to the topic for outgoing messages (response topic) the terminal subscribes to. Use a unique value in the FIMI\Echo field to match the request and response. It is also allowed to specify a user attribute in the Kafka message header to match data by the header.

To transfer data in the XML format, the JMS channel is used and the communication with the FIMI driver is implemented via the channel of the IBM WebSphere Message Queue middleware. The terminal communicates with the IBM WebSphere Message Queue middleware and sends the XML messages (requests) to the queue for incoming messages (request queue). The FIMI driver receives incoming messages (requests) from the incoming queue. After the request is processed, the FIMI driver sends the outgoing XML message (response) to the topic for outgoing messages (response queue) from which the terminal reads data. Use a unique value in the FIMI\Echo field to match the request and response.

The request of FIMI terminal to FIMI driver is arranged as **TWO** transaction.

The FIMI terminal-to-driver communication if the TCP and HTTP channels are used must be implemented as a temporary connection and pass the following stages:

- 1) connecting to driver
- 2) sending request to driver
- 3) waiting for and obtaining response from driver
- 4) connection release

For the UAMP format, it is initiated by the client side. For the SOAP XML format, the connection release attribute is copied from the Connection field of HTTP header of the message sender. The Close value indicates that the driver disconnects, whilst the Keep-alive value indicates that it keeps the connection after having generated a response. If the Connection field is not defined in the HTTP header or HTTP v.1.0 is used, the Close value is taken by default. If the terminal operates in the UAMP or SOAP XML format and keeps the connection with the driver longer than it is specified in the FIMI terminal settings (2 minutes by default, the value is calculated since the message from the client is accepted, at that, messages are not received and connection is not released by the client side), the driver forces the connection release and the message is generated to the Event log. If the request execution time exceeds the time of keeping the connection with the client, the driver initiates the connection release after sending the response.

The FIMI terminal-to-driver communication if the Kafka and JMS channels are used must be implemented as a permanent connection with the middleware (message broker).

If the terminal transmits the operator encrypted password, in the FIMI/Ver field, then it is necessary to define the protocol version 3.6 and higher. To work with the encrypted password, the *InitSession* request is the first to be sent to the host for obtaining *challenge* – password encryption component. For details, see p. 3.23 – [Password Encryption Algorithm](#).

To arrange FIMI transactions, the terminal employs the BeginTransaction, Commit, RollBack messages (see p. 2.3.2 Service Messages). FIMI transaction allows to implement such an operating mode when several messages are confirmed or rolled back in **TWO** depending on the external system conditions. For the transactional operating mode, FIMI terminal sends BeginTransaction, several requests, then Commit and RollBack within the same **TWO** session (if the TCP and HTTP channels are used). Driver forcedly rolls back the FIMI incompletely transaction continuing more than 20 seconds and then disconnects. The FIMI transaction mode supposes that it is not allowed to send messages addressed to the authorizer or host (authorize a transaction within the operation).

If the driver failed to respond to the terminal (for example, because of the connection break after the request is received or errors sending a response to the message broker) and the transaction was completed successfully, the automatic reversal to the successful transaction will be

generated for the *POSRequest* operation. For other operations, the changes (if any) are not cancelled.

***For the correct generation of responses, it is recommended to set the FIMI terminal version to not lower than 2.0 in the FIMI terminal settings for working via the SOAP XML and JMS XML protocol. The terminal operation with the version lower than 2.0 is considered outdated and not supported. It is recommended to set the terminal version to 3.0 when working via the SOAP XML and JMS XML protocols for the more optimal operation of the driver.***

For the examples of generating the driver requests and responses, refer to section 8 "Request Examples".

## 1.2 Offline Mode

In the Offline mode, the files must be placed to a special directory. FIMI driver loads files in the order they were created. The oldest one is loaded the first. The file names are optional, but the directory can contain only files with the specified protocols – UAMP or SOAP XML. Once the files are read and processed, they are moved to the BACKUP directory of the incoming directory or deleted. In case the files are moved, the file name is supplemented with the file loading time in the format “\_YYMMDDHHMISS”. For each processed file from the incoming directory, the system creates a file named after original one in the outgoing directory. The directories are set in the **Processing System Configuration** application in the unit of the VTBI terminal with the FIMI type.

The file can contain any number of requests irrespective of the format. It is recommended to mark each request with the unique value in the FIMI\Echo field for further comparing with the response. The requests from file are processed serially. The output file contains the responses in the order they are presented in the source file.

In the offline mode, it is possible to check the checksum (CRC32) that is transferred in the input file and calculate the checksum of information contained in the output file. For this purpose, the *FileHeader* operation is used. If the *FileHeader* operation is used, it must be first in the file.

In the offline mode, it is possible to send messages in the following formats:

- UAMP
- SOAP XML
- XML (available for terminal version 3 and onwards)

The requests in files of UAMP format are separated by (0x0a). HTTP header stands before each request in the SOAP XML format (the HTTP header must be always completed with the Content-Length element) for the framing and highlighting of the request if the terminal version 2.0 is used or the request must be transferred without the HTTP header for the terminal version 3.0 and onwards (in this case, requests in the file shoud be separated by (0x10)). The requests in the XML files (available for the terminal version 3.0 and onwards) are separated by (0x10).

Offline mode does not suppose the use of BeginTransaction, Commit, RollBack. It is also impossible to use the encrypted password. The password is transferred in clear. This restriction concerns with the impossibility of sequenced exchange of encryption component with the terminal.

Input and output files can be encrypted in compliance with the SMIME / CMS standard (refer to <http://tools.ietf.org/html/rfc3851>). At that, the file is as follows:

```
MIME-Version: 1.0
Content-Type: application/x-pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"
Content-Transfer-Encoding: base64
com.compassplus.header-FileType: FIMI Offline In

MIAGCSqGSIb3DQEHA6CAMIACQAxggGZMIIIB1QIBADB9MHAXCzAJBgNVBAYTA1VLMRYwFAYDVQQK
DA1PcGVuU1NMIEdyb3VwMSIwIAYDVQQLDB1GT1IgVEVTVELOrYBQVVJQT1NFUyBPTkxZMSUwIwYD
...
...
```

The file header provides the following application-specific parameters in addition to standard ones:

Parameter	Value
com.compassplus.header-FileType	‘FIMI Offline In’ – for the input file ‘FIMI Offline Out’ – for the output file

To decrypt and encrypt the file, the OpenSSL utility can be used (<https://www.openssl.org/>), for example:

```
openssl smime -decrypt -in encrypted.tip -out decrypted.tip -inkey key.pem
```

In case the output file is encrypted, its counters and checksum are not written to the *FileHeader* operation response. To get the counters and checksum, use the *FileTrailer* operation. If the *FileTrailer* operation is used, it must be at the end of the file.

***It is recommended to set the FIMI terminal version to not lower than 2.0 in the FIMI terminal settings for working via the SOAP XML and JMS XML protocol for the correct generation of responses. The terminal operation with the version lower than 2.0 is considered outdated and not supported. It is recommended to set the terminal version to 3.0 when working via the SOAP XML and JMS XML protocols for the more optimal operation of the driver.***

For the examples of generating the input and output files, refer to section 9 "Examples of Files Generation for Offline Mode".

## 2 FIMI Protocol Description

### 2.1 Operations

#### 2.1.1 Service Operations

##### **Session:**

- *InitSession* – operator session initialization
- *Logon* – operator authentication
- *Logoff* – operator session completion
- *GenerateDynamicPassword* – generating the dynamic password for the operator authentication
- *Ping* – updating the operator session activity

##### **Operator:**

- *ChangePassword* – operator password change
- *ResetPassword* – resets the operator password and sets the new one
- *GetFIMIClerks* – request for the list of operators and their terminal IDs
- *CreateFIMIClerk* – operator registration
- *SetFIMIClerkActivityStatus* – setting the operator activity attribute (blocking/unblocking)
- *SetNeedChangePassword* – setting the attribute indicating that the operator password must be changed
- *DeleteOperatorSessions* – request for deleting all operator sessions
- *GetFIMIClerkProperties* – request for the list of operator properties
- *SetFIMIClerkProperties* – removing, creating and changing operator properties
- *GetFIMIClerkAddresses* – request for addresses for connecting the operator

##### **FIMI Transaction Operations:**

- *BeginTransaction* – transaction initiation
- *Commit* – transaction completion
- *RollBack* – transaction rollback

##### **Offline Mode Operations:**

- *FileHeader* – header operation for the Offline mode
- *FileTrailer* – final operation for the Offline mode

**Operations Requesting for Reference Data:**

- *GetTranId* – request for the transaction ID
- *GetVCProducts* – request for the list of card products
- *GetServiceMembersList* – request for the list of the service members

## 2.1.2 Issuing Operations

**Card:**

- *GetCardInfo* – card/Telebank customer-related information request
- *SetCardStatus* – setting the card/Telebank customer status
- *SetCardPVV* – setting the card/Telebank customer PVV
- *SetDynamicPVV\_PINOffset* – setting dynamic PVV / PIN Offset for a card
- *SetCardUdCVV2* – setting user-defined CVV2(UdCVV2) for card
- *SetCardExpiration* – card/Telebank customer expiration date definition in the database
- *SetCardProfile* – setting the card/Telebank customer profile
- *SetCard2AcctDescr* – definition of the description for the account linked to the card/Telebank customer
- *ResetBadPINTries* – bad PIN tries counter reset
- *ResetCardEMVATC* – ATC counter reset for EMV card
- *ResetCardLimitsCounter* – card/Telebank customer limit counters reset
- *CopyCardLimitCounters* – card/Telebank customer limit counters copying
- *GetCardLimits* – request for the list of max and current limit values of the card/Telebank customer
- *SetTmpCardLimits* – card/Telebank customer temporary max limits definition
- *SetCardLimits* – changing card/Telebank customer max limits and their validity periods
- *SetCardLimitsRanges* – setting ranges of the card/Telebank customer limit
- *GetCardProfileLimits* – request for the list of limits on the card profile
- *SetCardProfileLimits* – card profile limits definition
- *CNSCardConfig* – card/Telebank customer messaging parameters definition
- *ConfigureCMSProfile* – card/Telebank customer messaging profile configuration

- *CardRiskControl*
  - card/Telebank customer risk control parameters definition
- *ResetCard2AcctLink*
  - re-creating the link of card/Telebank customer and account
- *UpdateCard2AcctLink*
  - updating the link of card/Telebank customer and account
- *DeleteCard2AcctLink*
  - deleting the link of card/Telebank customer and account
- *CreateCard*
  - recording the information on the existing card
- *CreateVCard*
  - virtual Internet card creation
- *SetEMVCardParams*
  - EMV attributes setting
- *GetCardStatement*
  - card statement request
- *AddCardVendor*
  - linking vendor to the card
- *DelCardVendor*
  - unlinking vendor from card
- *SetReissuedCardVariant*
  - reissued card variant setup
- *SetCardUserFields*
  - card/Telebank customer user fields definition
- *ChangeAlphaNumericPassword*
  - changing the card/Telebank customer alphanumeric password
- *ResetAlphaNumericPassword*
  - reset of the card/Telebank customer alphanumeric password
- *ResetBadDynAuthPwdTries*
  - reset of the bad dynamic authentication password tries counter for the card/Telebank customer abonent
- *ChangeECStatus*
  - changing card status in E-Commerce
- *GetPrefixInfo*
  - request for information by prefix
- *SetCardPerson*
  - changing card/Telebank customer owner
- *GetCardAutopayments*
  - request for the list of autopayments by the card number
- *AttachAutopaymentToCard*
  - adding the autopayment
- *DetachAutopaymentFromCard*
  - deleting the autopayment
- *EditCardAutopayment*
  - editing the autopayment
- *GetCardAutopaymentHistory*
  - request for the history of autopayment subscriptions
- *DeleteCard*
  - deleting the card
- *AddCMSAbonent*
  - adding the messaging parameters for the card/Telebank customer
- *RemoveCMSAbonent*
  - deleting the messaging parameters for the card/Telebank customer
- *ChangeCMSAbonent*
  - editing the messaging parameters for the card/Telebank customer

- *ResetCardAbonentAdditionalInfo*  
– resetting additional data on the abonent for the card/Telebank customer abonent
- *SetCard3DSecureAuthentication*  
– setting the E-Commerce (3-D Secure) authentication parameters for the card
- *GetCardCounters*  
– request for the card/Telebank customer counters
- *SetCardCounters*  
– changing the card/Telebank customer counters
- *AttachCards*  
– linking cards
- *SetCardContactlessInterface*  
– changing the status of the contactless card interface
- *GetCardTokensInfo*  
– request for the information on card tokens from the TWO storage
- *TokenInquiry*  
– request for the information on card tokens
- *SetCardTokenStatus*  
– changing the card token status
- *CreateCardToken*  
– creating the card token in the TWO storage
- *DeleteCardToken*  
– deleting the card token in the TWO storage
- *UpdateCardToken*  
– updating the card token in the TWO storage
- *ReassignCardTokens*  
– re-assigning the card tokens
- *GetCardTokenNeedConfirm*  
– request for the information on operations by the card token which require confirmation
- *CardTokenConfirmation*  
– card token operation confirmation
- *GetCardPAR*  
– request for Payment Account Reference (PAR) by card/card token
- *UpdateCardMetadata*  
– changing card metadata
- *GetConsumerPresentedQRCode*  
– request for the consumer-presented QR code with the card data for payment.

**Alias:**

- *GetAliasInfo*  
– request for the information on aliases in **TWO**
- *CreateAlias*  
– creating the alias
- *DeleteAlias*  
– deleting the alias
- *UpdateAlias*  
– changing the alias
- *CheckAlias*  
– checking the alias
- *SetAliasPermission*  
– setting additional permissions for the alias
- *SetAliasSubscription*  
– managing alias subscriptions
- *AliasVerification*  
– checking recipient alias and sender data before the transfer

**Telebank Customer:**

- *CreateTBCustomer* – Telebank customer creation
- *GetTBCustHistory* – Telebank customer history request
- *SetExtraAuthLevel* – extra authentication level setting
- *AddTBTransfer* – adding Telebank customer transfer template
- *DelTBTransfer* – deleting Telebank customer transfer template
- *AddTBPayment* – adding Telebank customer payment template
- *DelTBPayment* – deleting Telebank customer payment template
- *AddTBRetailer* – adding the link between the Telebank customer and retailer
- *DeleteTBRetailer* – deleting the link between the Telebank customer and retailer
- *AttachCardToTBCustomer* – linking the own institution card to the Telebank customer
- *DetachCardFromTBCustomer* – deleting the own institution card linked to the Telebank customer
- *AttachForeignCardToTBCustomer* – linking the external institution card to the Telebank customer
- *DetachForeignCardFromTBCustomer* – deleting the external institution card to the Telebank customer
- *DeleteTBCustomer* – deleting the Telebank customer
- *SetTBLogin* – changing the Telebank customer text/numeric login

**Account:**

- *GetAcctInfo* – account information request
- *GetAcctStatement* – account history request
- *SetAcctStatus* – account status setting
- *ChangeAcctBalances* – changing attributes concerning account balances
- *AcctDebit* – account debit
- *AcctCredit* – account credit
- *SetAccountOverdraft* – account overdraft change
- *CreateAccount* – account creation
- *GetAcctLimits* – request for the list of the account max and current limits
- *SetAcctLimits* – changing the account max limits and their validity periods

- *SetAcctLimitsRanges* – setting the account limit ranges
- *GetAcctGroupLimits* – getting limit values by the scheme level accounts
- *ResetAcctLimitCounters* – account limits counter reset
- *SetTmpAcctLimits* – setting the account temporary limits maximums
- *SetAccountUserFields* – account user fields setup
- *SetAccountPerson* – changing account owner
- *BalanceInquiry* – request for the customer account balances
- *DeleteAccount* – deleting the account
- *GetAcctCounters* – getting the account counters
- *SetAcctCounters* – changing the account counters
- *SetAccountPermissibleExcess* – setting the allowed excess of the available account balance
- *SetAcctProtectedAmount* – setting the protected account amount
- *GetAcctTokensInfo* – request for the information on account tokens from the TWO storage
- *SetAcctTokenStatus* – changing the account token status
- *CreateAcctToken* – creating the account token in the storage
- *SetAcctLegalEntity* – changing corporate customer of the account

**Card-to-Account Link:**

- *SetAcct2CardStatus* – setting account status for the card-to-account link
- *GetAcct2CardLimits* – request for the list of account-to-card link limits
- *SetAcct2CardLimits* – changing the account-to-card link limits maximums and their validity periods
- *ResetAcct2CardLimitCounters* – reset of the account-to-card link limits counter

**Customer:**

- *GetPersonInfo* – customer personal information request
- *CreatePerson* – private customer creation
- *UpdatePerson* – private customer information update
- *SetPersonUserFields* – customer user fields definition
- *AddPersonConfidentialInfo* – adding confidential information for a customer
- *DelPersonConfidentialInfo* – deleting confidential information of a customer

- *ChangePersonConfidentialInfo*
  - changing the customer personal information
- *GetPersonsHierarchy*
  - request for customers hierarchy
- *DeletePerson*
  - deleting the customer
- *GetAvatar*
  - getting the customer avatar
- *SetAvatar*
  - changing the customer avatar
- *AddPersonCMSCAbonent*
  - adding the messaging parameters for the customer
- *RemovePersonCMSCAbonent*
  - deleting the messaging parameters for the customer
- *ChangePersonCMSCAbonent*
  - changing the messaging parameters for the customer
- *ConfigurePersonCMSProfile*
  - setting the messaging profile parameters for the customer
- *ResetPersonAbonentAdditionalInfo*
  - resetting additional data on the abonent for the customer abonent
- *ResetPersonBadDynAuthPwdTries*
  - resetting counter of the dynamic password bad tries for the customer abonent
- *GetPersonLimits*
  - request for the list of customer limits with their maximum and current values
- *SetPersonLimits*
  - changing maximum values and validity periods of customer limits
- *SetPersonLimitsRanges*
  - setting customer limit ranges
- *GetPersonGroupLimits*
  - request for values of limits on customers of the scheme level
- *ResetPersonLimitCounters*
  - resetting the customer limits counter
- *SetTmpPersonLimits*
  - setting temporary maximum values for customer limits
- *GetPersonCounters*
  - getting the customer counters
- *SetPersonCounters*
  - changing the customer counters

**Corporate Customer:**

- *GetLegalEntityInfo*
  - request for the information on the corporate customer
- *CreateLegalEntity*
  - creating corporate customer
- *UpdateLegalEntity*
  - updating the information on the corporate customer
- *DeleteLegalEntity*
  - deleting the corporate customer
- *SetLegalEntityUserFields*
  - setting user-defined fields of the corporate customer

**Currency Exchange Rates:**

- *GetRateGroups* – request for the list of currency exchange rate groups
- *GetRates* – currency exchange rates request
- *AddRate* – adding the future exchange rate
- *ModifyRate* – changing the current or future exchange rate
- *DeleteRate* – removing the future exchange rate

**Request to Back-Office:**

- *GetBackOfficeInfo* – request for information from Back-Office

**Messaging Service:**

- *GetCMSBuffer* – request for the log of messages to be sent to the customer
- *GetCMSSArchive* – request for the archive of messages sent to the customer
- *GetCMSCCommands* – request for the log of commands received from the customer
- *GetSchemeItemTypes* – request for the list of available scheme items
- *SendMessage* – sending a message

**Issuer History:**

- *GetIssuerHistory* – request for the issuer history log
- *GetIssuerHistoryDetails* – request for details of the issuer history log

**EMV Scripts:**

- *GetEMVScripts* – card EMV scripts request
- *AddEMVScript* – adding EMV script
- *UpdateEMVScript* – EMV script update
- *DeleteEMVScript* – EMV script removal
- *SetEMVScriptSeqNo* – EMV script sequence number definition

**Bank Messages Offline Exchange:**

- *BMSGetForms* – request for the list of forms and groups
- *BMSGetFormFields* – request for the form fields
- *BMSGetMessages* – request for the list of customer messages
- *BMSGetMessageFields* – request for the message fields
- *BMSReadMessages* – marking the bank messages as read
- *BMSSendMessage* – sending the message to bank

**Group:**

- *CreateIssuerObjects* – request for issuer objects creation (customer, account, virtual card, credit account)

**Issuer Fees:**

- *GetIssuerFeesForOwnTerm* – request for issuer fees for own terminals
- *SetIssuerFeesForOwnTerm* – changing issuer fees for own terminals
- *AddIssuerFeesForOwnTerm* – adding issuer fees for own terminals
- *DeleteIssuerFeesForOwnTerm* – deleting issuer fees for own terminals
- *GetIssuerFeesForExternalTerm* – request for issuer fees for external terminals
- *SetIssuerFeesForExternalTerm* – changing issuer fees for external terminals
- *AddIssuerFeesForExternalTerm* – adding issuer fees for external terminals
- *DeleteIssuerFeesForExternalTerm* – deleting issuer fees for external terminals

**Devices:**

- *GetDevicesList* – request for the information on devices
- *GetDeviceHistory* – request for the device actions log
- *ChangeDeviceStatus* – changing the device status
- *SetDeviceAdditionalFields* – setting user-defined fields of the device

**Arrests:**

- *GetArrests* – request for the list of arrests
- *AddArrest* – adding an arrest
- *DeleteArrest* – deleting (canceling) the arrest
- *ChangeArrest* – changing the arrest

**E-Commerce Merchants:**

- *GetECMerchant* – request for the list of E-Commerce merchants
- *ChangeECMerchantStatus* – changing the E-Commerce merchant status
- *DetachECMerchant* – deleting the card linked to the E-Commerce merchant

**Invoice:**

- *GetInvoice* – request for the list of Invoice on card/account
- *ChangeInvoiceStatus* – changing the Invoice status

### 2.1.3 Acquiring Operations

**Terminals:**

- *GetRetailerList* – request for the list of retailers
- *CreateRetailer* – creating a retailer
- *UpdateRetailer* – request for updated information on retailer
- *GetRetailerLimits* – request for the list of retailer limits
- *SetRetailerLimits* – changing maximum values and validity periods of the retailer limits
- *ResetRetailerLimitCounters* – reset of the retailer limits counter
- *GetRetailerSchemeLimits* – request for the values of limits by the scheme-level retailers
- *GetRetailerGroupLimits* – request for the list of limits of the retailer group
- *SetRetailerGroupLimits* – changing maximum values and validity periods of limits of the retailer group
- *ResetRetailerGroupLimitCounters* – reset of the retailer group limits counter
- *GetRetailerGroupSchemeLimits* – request for the values of limits by the scheme-level retailer group
- *GetTermList* – request for the list of terminals
- *TerminalRiskControl* – setting the terminal risk level
- *CreateTerminal* – creating a terminal
- *UpdateTerminal* – updating the information on terminal
- *GetTerminalAdditionalFields* – request for the additional terminal fields
- *SetTerminalAdditionalFields* – setting the additional terminal fields
- *GetATMStatus* – ATM logical state request
- *GetATMHardware* – ATM hardware configuration request

- *GetATMFitness* – ATM fitness condition request
- *GetATMMedia* – ATM media request
- *GetATMSensors* – ATM sensors state request
- *GetATMHoppers* – ATM hoppers state request
- *GetATMActivity* – ATM last activities request
- *ATMCommand* – command to ATM driver
- *GetATMWithdrawalHold* – request for the information on the ATM holds
- *CreateATMWithdrawalHold* – creating the ATM hold
- *ChangeStatusATMWithdrawalHold* – changing the ATM hold status
- *POSState* – POS terminal state request
- *POSCommand* – command to POS terminal
- *GetNDCAuditJournal* – NDC e-journal request
- *GetDDCAuditJournal* – DDC e-journal request

**Events:**

- *GetATMEvents* – ATM events request
- *PutEvent* – writing a message to the Event log

**Correspondent Accounts:**

- *GetCorrespAcctInfo* – request for information on the institution correspondent account
- *ModifyCorrespAcct* – agent correspondent account balance change

**Acquiring Limits:**

- *GetCumulativeLimits* – getting cumulative limits
- *AddCumulativeLimit* – adding the cumulative limit
- *UpdateCumulativeLimit* – changing the cumulative limit
- *DeleteCumulativeLimit* – deleting the cumulative limit
- *ResetCumulativeLimit* – reset of current counters on the cumulative limit

**QR Codes:**

- *GenerateQRCode* – generating the merchant-presented QR code in the system
- *GetQRCodeInfo* – getting the information on the QR code and merchant
- *CheckQRCodeStatus* – checking the status by the QR code
- *GetQRCodePayload* – request for payload by QR code

- *ModifyQRCode* – changing the QR code status/availability
- *MerchantRefund* – refund generation by the merchant
- *PullQRRequest* – request the QR code payment generation
  
- *NotificationQRSubscription* – notification on the created subscription by QR code

#### **2.1.4 Transaction Requests**

- *GetTLADir* – TLA dictionary of partitions request
- *GetTransInfo* – transaction information request
- *GetFIMITrans* – FIMI transactions request
- *GetATMAdminTrans* – ATM administrative transactions request
- *ServiceOperationCancelation* – cancelling the service operation

#### **2.1.5 POS Transactions**

- *ReverseTransaction* – transaction reversal
- *POSRequest* – POS transaction generation
- *StandAloneReversal* – Stand Alone transaction reversal

#### **2.1.6 Reports**

- *GetReportList* – report list request
- *ExecuteReport* – sending the command for report execution
- *GetReportRequests* – obtaining a list of requests for the executed/scheduled reports
- *GetReport* – executed report request
- *DeleteReportRequest* – deleting report/command for report execution from a list of requests

#### **2.1.7 Stoplists Maintenance**

- *VISAStoplist* – Visa Stoplist processing
- *MCStoplist* – Mastercard Stoplist processing
- *MCGlobalStoplist* – Mastercard-Global Stoplist processing
- *LocalStoplist* – local Stoplist processing
- *AmExStoplist* – American Express Stoplist processing
- *DiscoverStoplist* – Discover Stoplist processing

#### **2.1.8 Visa Administrative Transactions**

- *AdjustToVISA* – adjustment of Visa acquiring SMS-transaction

- *RepresentToVISA* – repeated representment of Visa acquiring SMS transaction
- *ConfirmCopyRequestToVISA* – confirmation of sending a receipt copy on the issuer request
- *FeeToVISA* – Visa fees request or cash back to Visa
- *TextToVISA* – sending text message to Visa

### **2.1.9 MasterCard Global Administrative Transactions**

- *AdjustToMC* – adjustment of MasterCard Global acquiring MDS-transaction
- *RepresentToMC* – repeated representment of MasterCard Global acquiring MDS-transaction
- *ChargebackToMC* – issuer reversal of MasterCard Global MDS-transaction
- *ChargebackReversalToMC* – reversal of issuer Chargeback of MasterCard Global MDS transaction

### **2.1.10 Administrative Dispute Transactions of the Local Payment System**

- *LocalChargeback* – issuer reversal of Request transaction
- *LocalChargebackReversal* – Chargeback Reversal
- *LocalAdjustment* – adjustment of the acquirer transaction

### **2.1.11 System Units**

- *GetUnitState* – request for list of **TWO** units and their statuses
- *UnitCommand* – command to the unit
- *GetUnitEvents* – unit events request
- *GetTICConfiguration* – request for the configuration of TIC host interface connected to Interface Application
- *SetTICConfiguration* – request for changing the configuration of TIC host interface connected to Interface Application

### **2.1.12 Cryptographic Operations**

- *GetWorkingKey* – working key generation
- *GetPVV PIN Offset* – PVV/Internet PVV or PIN offset/Internet PIN offset generation
- *GetCVV* – CVV or CVV2 generation
- *Check PVV PIN Offset* – PVV/Internet PVV or PIN offset/Internet PIN offset verification

- *CryptoFunction* – Security Module function call
- *GetExcludedPINList* – request for the list of “weak” PINs
- *PrepareProvisioningData* – generating data for Push Provisioning (card registration in a wallet)
- *GetPIN* – request for the PIN block using the card PINOffset

***Key Management System Keys:***

- *GetRepositoryKey* – getting the value of the key from Key Management System
- *AssignKeyByTerm* – linking the key from Key Management System to the terminal
- *ChangeKeyStatus* – changing the status of the key from Key Management System

### 2.1.13 User Operations

- *UserDefined* – calling the algorithmix user function

## 2.2 Set of Field Types

When describing the protocol field or structure of subfields, certain types of fields provided below can be used.

It is also allowed to use the symbol '\*' (asterisk) that follows the data type designation and means that the field is mandatory. For example, "GetCardInfo/Rq/PAN Str\*" indicates that the GetCardInfo/Rq/PAN field is mandatory. In the response to the terminal, the host sends all mandatory fields and optional fields whose values are defined on host or whose sending conditions described in the specification are met. Optional fields with empty values are not sent in the response to the terminal.

It is possible to use the sequence of symbols '[' that follows the data type designation or description of the structure of subfields and means that the field is an array of objects of a particular type. For example, "FIMI/GetTransInfo/Rq/Type Int\*[]]" indicates that the GetTransInfo/Rq/Type field is an integer array.

### 2.2.1 Integer Type (Int)

The type is used for the transfer of positive and negative integer numbers. When describing the type, the maximum length of the field (regardless of the number sign) can be restricted as follows:

*Int(<max length>)*

If the maximum length for the type is not specified, the default value is Int(9).

For the UAMP exchange format, the type matches the Int and ArrInt UAMP type (refer to the **UAMP(E).pdf** specification), for XML – the type matches the "xs:integer" xsd type for the fields less than or equal to Int(9) and with the "xs:long" xsd type for the fields exceeding Int(9).

Example of the field transfer:

Description	UAMP	XML
Int(3)	FIMI/GetCardInfo/Rq/MBR=0	<fimi:GetCardInfoRq> <fimi:Request> <fimi1:MBR>0</fimi1:MBR> </fimi:Request> </fimi:GetCardInfoRq>
Int(1)	FIMI/ChangeECStatus/Rq>Status=-1	<fimi:ChangeECStatusRq> <fimi:Request> <fimi1:Status>-1</fimi1:Status> </fimi:Request> </fimi:ChangeECStatusRq>
Int[]	FIMI/GetTransInfo/Rq/Type=100FS200	<fimi:GetTransInfoRq> <fimi:Request> <fimi1:Type> <fimi1:Row> <fimi1:Value>100</fimi1:Value> </fimi1:Row> <fimi1:Row> <fimi1:Value>200</fimi1:Value> </fimi1:Row> </fimi1:Type> <fimi:Request> </fimi:GetTransInfoRq>

## 2.2.2 Real Number Type (Num)

The type is used for the transfer of positive and negative real numbers. When describing the type, the maximum length of the integer part (regardless of the sign and decimal point) and maximum length of the fractional part can be restricted as follows:

*Num(<max length of integer part>[,<max length of data after decimal point>])*

If the maximum length for the type is not specified, the default value is Num(12,3). If the length of data following the decimal point is not specified, the default value is 3.

For the UAMP exchange format, the type matches the Num and ArrNum UAMP type (refer to the **UAMP(E).pdf** specification), for XML – the type matches the "xs: decimal" xsd type.

Example of the field transfer:

Description	UAMP	XML
Num*	FIMI/AcctCredit/Rq/Amount=123.456	<fimi:AcctCreditRq> <fimi:Request> <fimi1:Amount>123.456</fimi1:Amount> </fimi:Request> </fimi:AcctCreditRq>
	FIMI/ChangeAcctBalances/Rq/ AvailDelta=-123.456	<fimi:AcctCreditRq> <fimi:Request> <fimi1:AvailDelta>-123.456</fimi1:AvailDelta> </fimi:Request> </fimi:AcctCreditRq>

## 2.2.3 Date Type (Time)

The type is used for the transfer of the date and time. For the UAMP exchange format, the type matches the Num and ArrNum UAMP type and contains the positive real number with the number of seconds (fractional part is possible) passed since 00:00:00 1/1/1901 (refer to the **UAMP(E).pdf** specification), for XML – the type matches the "xs: dateTime" xsd type.

**TWO** provides an option to define the terminal time offset when describing FIMI terminal. It is effective if FIMI terminal works in a different time zone than **TWO**. It is considered that for the *Time* field, the time is in the terminal time zone. For some operations (card expiry definition), the time is in the host time zone – see the field description.

Example of the field transfer:

Description	UAMP	XML
Time	FIMI/GetTransInfo/Rq/ ToTime=3696762222	<fimi:GetTransInfoRq> <fimi:Request> <fimi1:ToTime>2018-02- 22T14:23:42</fimi1:ToTime> </fimi:Request> </fimi:GetTransInfoRq>

## 2.2.4 String Type (Str)

The type is used for the transfer of string data. When describing the type, the maximum length of the field can be restricted as follows:

*Str(<max length>)*

If the maximum length for the type is not specified, the default value is Str(4000).

For the UAMP exchange format, the type matches the Str and ArrStr UAMP type (refer to the **UAMP(E).pdf** specification), for XML – the type matches the "xs:string" xsd type. For the XML protocol, it is required to replace markup and prohibited XML symbols in the field value with the escape tags and escape sequences ("&#x<Hex code>;" or "&#<decimal code>") or transfer the information in the CDATA section. For details, refer to the description in the protocol XML 1.0/1.1.

Example of the field transfer:

Description	UAMP	XML
Str(19)	FIMI/GetCardInfo/Rq/PAN=455555	<fimi:GetCardInfoRq> <fimi:Request> <fimi1:PAN>455555</fimi1:PAN> </fimi:Request> </fimi:GetCardInfoRq>
Str(150)	FIMI/GetRetailerList/Rq/ RetailerName='Retailer'	<fimi:GetRetailerListRq> <fimi:Request> <fimi1:RetailerName> &apos;Retailer&apos; </fimi1:RetailerName> </fimi:Request> </fimi:GetRetailerListRq>  <fimi:GetRetailerListRq> <fimi1:RetailerName> <![CDATA['Retailer']]> </fimi1:RetailerName> </fimi:Request> </fimi:GetRetailerListRq>
Str[]	FIMI/GetTransInfo/Rq/ ArrayAcqInstName=DEMO1FSDEMO2	<fimi:GetTransInfoRq> <fimi:Request> <fimi1:ArrayAcqInstName> <fimi1:Row> <fimi1:Value>DEMO1</fimi1:Value> </fimi1:Row> <fimi1:Row> <fimi1:Value>DEMO2</fimi1:Value> </fimi1:Row> </fimi1:ArrayAcqInstName> </fimi:Request> </fimi:GetTransInfoRq>

## 2.2.5 String in Hexadecimal Format (Hex)

The type is used for the transfer of string data in the hexadecimal representation. String data are converted to the hexadecimal representation and transferred as a string with allowed symbols [0-9 a-f A-F]. When describing the type, the maximum length of the field can be described as follows:

*Hex(<max length in bytes>)*

If the maximum length for the type is not specified, the default value is Hex(4000).

For the UAMP exchange format, the type matches the Str and ArrStr UAMP type (refer to the **UAMP(E).pdf** specification) with allowed symbols [0-9 a-f A-F], for XML – the type matches the "xs:hexBinary" xsd type.

Example of the field transfer:

Description	UAMP	XML
Hex(32)	FIMI/POSRequest/Rq/ ICC_IssuerData=54657377A  FIMI/POSRequest/Rq/ ICC_IssuerData=54657377a	<fimi:POSRequestRq> <fimi:Request> <fimi1:ICC_IssuerData> 54657377A </fimi1:ICC_IssuerData> </fimi:Request> </fimi:POSRequestRq>  <fimi:POSRequestRq> <fimi:Request> <fimi1:ICC_IssuerData> 54657377a </fimi1:ICC_IssuerData> </fimi:Request> </fimi:POSRequestRq>

## 2.2.6 String with UAMP Structure (StrUAMP)

The type is used for the transfer of string data presented as an UAMP structure of subfields. When describing the type, the maximum length of the field can be restricted as follows:

**StrUAMP(<max length>)**

If the maximum length for the type is not specified, the default value is Str(4000).

The field generation format depends on the exchange protocol.

**For UAMP exchange format:**

The type matches the Str and ArrStr UAMP type. For data transferred in the field, special UAMP characters are escaped (refer to the **UAMP(E).pdf** specification).

**For XML exchange format:**

The type matches the "xs:string" xsd type. For data transferred in the field, characters prohibited to be transferred in the XML format are escaped. If the request data contain the ASCII characters with the code from 01 to 31 (in the Hex format – codes from 0x01 to 0x1F, inclusive), the following requirements must be met to transfer them within this type correctly:

- characters must replaced with the escape sequences of the following format: "&#x<Hex code>;" or "&#<decimal code>;" (for example, "&#x01;")
- version of the header XML tag must be 1.1 ("<?xml version="1.1" encoding="utf-8"?>").

If the response transfers fields with this type (the field value is other than NULL), the driver changes the version of the response xml to version 1.1 (the header XML tag will be as follows: "<?xml version="1.1" encoding="utf-8"?>") and characters with codes from 01 to 31 will be replaced with the escape sequences "&#x<Hex code>;" or "&#<decimal code>;" .

Example of the field transfer:

Description	UAMP	XML
StrUAMP(100)	FIMI/POSRequest/Rq/ExtPSFields= MC22=051SPPSMC61=10	<fimi:POSRequestRq> <fimi:Request> <fimi1:ExtPSFields> MC22=051&x10;MC61=10 </fimi1:ExtPSFields> </fimi:Request> </fimi:ExtPSFields>

## 2.2.7 String with XML Document (StrXML)

The type is used for the transfer of the xml structure in the string field. When describing the type, the maximum length of the field can be restricted as follows:

*StrXML(<max length>)*

If the maximum length for the type is not specified, the default value is the maximum size of the CLOB type in the **TWO** DB.

For the UAMP exchange format, the type matches the Str and ArrStr UAMP type (refer to the **UAMP(E).pdf** specification), for XML – the type matches the "xs:string" xsd type.

For the UAMP exchange protocol, the field data are transferred as an unchanged string.

For the XML protocol, when the field is transferred in the request, the markup XML characters in the field value must be replaced with the escape tags or the information must be transferred in the CDATA section. For details, refer to the description in the XML protocol 1.0/1.1. For the XML protocol, when the field is transferred in the response, the driver transfers the string in the UTF8 encoding converted to the base64 format.

Example of the field transfer:

Description	UAMP	XML
XMLValue (for request)	FIMI/SetCardUserFields/Rq/ UserFields=ExampleFSNIFS <TestData> <Case1>Value1</Case1> <Case2 AddInfo = "Value2"/> </TestData>	<m1:SetCardUserFieldsRq> <fimi:Request> <fimi1:UserFields> <fimi1:Row> <fimi1:XMLValue> &lt;TestData&gt; &lt;Case1&gt;Value1&lt;/Case1&gt; &lt;Case2 AddInfo =&quot;Value2&quot;/&gt ;&lt;/TestData&gt; </fimi1:XMLValue> </fimi1:Row> </fimi1:UserFields> </fimi:Request> </m1:SetCardUserFieldsRq>  <m1:SetCardUserFieldsRq> <fimi:Request> <fimi1:UserFields> <fimi1:Row> <fimi1:XMLValue> <![CDATA[<TestData> <Case1>Value1</Case1> <Case2 AddInfo="Value2"/> </TestData>]]> </fimi1:XMLValue> </fimi1:Row> </fimi1:UserFields> </fimi:Request> </m1:SetCardUserFieldsRq>
XMLValue (for response)	FIMI/GetCardInfo/Rp/UserFields = ExampleFSNIFS<TestData> <Case1>Value1</Case1> <Case2 AddInfo = "Value2"/> </TestData>	<m1: GetCardInfoRp> <fimi:Response> <m0:UserFields> <m0:Row> <m0:XMLValue>  PFRlc3REYXRhPjxDYXNIMT5WYWx1ZTE8L0  Nhc2UxPjxDYXNIMiBBZGRpdGlvbmFsSW5mb

Description	UAMP	XML
		<pre>3JtYXRpb249IlZhbHVIMilvPjwvVGVzdERhdGE+     &lt;/m0:XMLValue&gt;     &lt;/m0:Row&gt;     &lt;/m0:UserFields&gt; &lt;/fimi:Response&gt; &lt;/m1:GetCardInfoRp&gt;</pre>

## 2.2.8 Type of Personal Payment Attributes (CustAcct)

The type is used for the transfer of the personal information on the customer (payment attributes). When describing the type, the maximum length of the field can be restricted as follows:

`CustAcct(<max length>)`

If the maximum length for the type is not specified, the default value is `CustAcct(4000)`.

For the UAMP exchange format, the type matches the Str and ArrStr UAMP type (refer to the **UAMP(E).pdf** specification), for XML – the type matches the "fimi\_types.xsd:CustAcct" xsd type. The field within this type is compound and consists of the ordered set of the "field name - value" pairs, the field generation format depends on the exchange protocol.

### For UAMP exchange format:

The personal information on the customer is composed by the terminal as an ordered set of the "field name - value" pairs separated by the character with the code 0x1c:

`<ID>=<VAL><0x1c><ID>=<VAL><0x1c>...`

where:

- ID – identifier of the attribute sent to the terminal in the `Name` field of the `VendorCustInfo` array of the `InitSession` operation response.
- VAL – attribute value. For example, phone number, personal account number and etc.

After the ordered set of the "field name - value" pairs is generated, the separator with the code 0x1c is escaped according to the rules of transferring special characters of the UAMP protocol (refer to the **UAMP(E).pdf** specification). If the `CustInfo` list for the vendor is empty (no personal information), the value '-' or NULL can be specified in this field.

### For SOAP XML exchange format:

The personal information on the customer is transferred as an ordered array – see the `CustAcct` type in **fimi\_types.xsd**. If the field is absent, it indicates the NULL value for the payment attributes.

Example of the field transfer:

Description	UAMP	XML
CustAcct (1000)	FIMI/POSRequest/Rq/ToAccount2=BIK =123456789SPFSINNC=SPFSKBK=55	<m1: POSRequestRq> <fimi:Request> <fimi1:ToAccount2> <fimi1:ToAccount2> <fimi1:Row> <fimi1:SEQ>1</fimi1:SEQ> <fimi1:ID>BIC</fimi1:ID> <fimi1:VAL>123456789</fimi1:VAL> </fimi1:Row> <fimi1:Row> <fimi1:SEQ>2</fimi1:SEQ> <fimi1:ID>INNC</fimi1:ID> <fimi1:VAL/> </fimi1:Row> <fimi1:Row> <fimi1:SEQ>3</fimi1:SEQ> <fimi1:ID>KBK</fimi1:ID> <fimi1:VAL>55</fimi1:VAL> </fimi1:Row> </fimi1:ToAccount2> </fimi1:ToAccount2> </fimi:Request> </m1: POSRequestRq>

## 2.2.9 Array of Structures (ArrRec)

The type is used for the transfer of the ordered array of structures. The structure is a certain set of fields with base types (Int, Num, Time, Str and etc.) compiled into a group (record). When describing the type, the maximum number of elements (records) of the array can be restricted:

ArrRec(<max number of characters>)

If the maximum value for the type is not specified, the number of records in the array is unlimited.

For the UAMP exchange format, the type matches the ArrRec UAMP type and is generated according to the UAMP protocol for the array of structures (refer to the **UAMP(E).pdf** specification), for XML – the field with this type is compound and consists of the unlimited number of the Row xml elements (fields) consisting of the limited number of sub-elements (subfields) with the set base types.

Example of the field transfer:

Description	UAMP	XML
ArrRec	FIMI/InitSession/Rp/AuthSchemes=1FSPositiveFS2IS2FSNegativeFS12	<pre> &lt;m1:InitSessionRp&gt;   &lt;m1:Response&gt;     &lt;m0:AuthSchemes&gt;       &lt;m0:Row&gt;         &lt;m0:Id&gt;1&lt;/m0:Id&gt;         &lt;m0:Title&gt;Positive&lt;/m0:Title&gt;         &lt;m0:Type&gt;2&lt;/m0:Type&gt;       &lt;/m0:Row&gt;       &lt;m0:Row&gt;         &lt;m0:Id&gt;2&lt;/m0:Id&gt;         &lt;m0:Title&gt;Negative&lt;/m0:Title&gt;         &lt;m0:Type&gt;12&lt;/m0:Type&gt;       &lt;/m0:Row&gt;     &lt;/m0:AuthSchemes&gt;   &lt;/m1:Response&gt; &lt;/m1:InitSessionRp&gt; </pre>

## 2.3 Message Contents

### 2.3.1 Common Header

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>Common Header</b>			
Ver	Int	In, out	Version for the products  If the messages are exchanged via SOAP, this field is ignored
Product	Str*	In, out	Title of the product: "FIMI"
Echo	Str	In, out	It is returned unchanged in the response
Encoding	Str	In, out	Message encoding. Available values: 'Cp866', 'ANSI', 'Cp1251', 'UTF8'  If not defined, encoding ='Cp866' is used. It is sent unchanged in response.  'ANSI' for: - Windows OS – ANSI encoding with single-byte code page defined at the OS level; - UNIX OS – OS base encoding.  On exchanging messages via SOAP, the field is ignored.
FIMI/Session	Int	In	Session ID. The field is mandatory for all the requests to the exclusion of InitSession if the encrypted password is used.
FIMI/Ver	Num*	In, out	Version for FIMI product. For requests, the terminal indicates the supported protocol version in this field. It is used to select the password encryption method. See p.4. The response returns the maximum allowed version of the protocol supported by the FIMI driver.
FIMI/Operation	Str*	In, out	Operation title: Logon, GetCardInfo, GetTermState.  For UAMP only.
FIMI/BusinessDate	Time	In	Operation business day. It may proceed <b>TWO</b> business day. If not defined, it is taken equal to <b>TWO</b> business day.  It is used by the external systems for the operations AcctDebit, AcctCredit.
FIMI/Clerk	Str(30)*	In	Name of the operator who initiated a transaction

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/Password	Str(64)*	In	<p>Operator password encrypted using NextChallenge in the ASCII HEX format or packed string.</p> <p><b>For password hash version 1(3DES).</b> If the password encryption is used (FIMI/Ver &gt;= 3.6), the field contains the encrypted password in the ASCII HEX format whose length does not exceed 16 characters.</p> <p>If the password encryption is not used (FIMI/Ver &lt;= 3.5), the field contains the clear password (if the "Calculate hash for clear password" option is enabled on the terminal) or password hash in the ASCII HEX format whose length does not exceed 16 characters. If the length is exceeded, the value is truncated to the maximum available length.</p> <p><b>For password hash version 2(SHA3-256).</b> If the password encryption is used (FIMI/Ver &gt;= 16.13), the field contains the encrypted password in the ASCII HEX format whose length does not exceed 64 characters.</p> <p>If the password encryption is not used (FIMI/Ver &lt;= 3.5), the field contains the clear password whose length does not exceed 64 characters.</p> <p><i>The password hash version used by the operator is returned in the PasswordVersion response field of the InitSession operation.</i></p>
FIMI/RetAddress	Str(300)	In	Return address of the customer (IP-address of the working station that initiates a request)
FIMI/Comment	Str	In	Textual comment to the transaction. It is saved to <b>TWO</b> Transaction log.
FIMI/Savepoint	Str	In	<p>Name of the transaction savepoint.</p> <p>It can be defined in a message within the FIMI open transaction. If defined, it is identical to the BeginTransaction/Rq/Savepoint field</p>
FIMI/TransactionNumber	Str(50)	In	Transaction number in the initiator system
FIMI/Language	Int(1)	In	<p>The language being requested. If not defined, the text descriptions will return in language set currently in <b>TWO</b>.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>NULL – system language.</li> <li>0 – English.</li> </ul>

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Dirac- tion</b> (relative to TWO)	<b>Description</b>
FIMI/OrigTime	Time	In	Time of the operation execution in the originator time zone. If it is not defined, the current time on server is used considering the time offset on the terminal.
FIMI/TransactionUserData	Str	In	User data of the transaction. The field value is placed in the additional transaction field with ID 1000(User_Data) in all transactions generated when executing the FIMI operation.
FIMI/ImpactData	Str	In	Data on the impact within the operation, it is transferred as a Base64-encoded UAMP string with the parameters. Available parameters: <ul style="list-style-type: none"> <li>• Str InitiatorRid – originator ID.</li> <li>• Str OrderRid – order ID.</li> <li>• Str ObjectRid – customer ID.</li> <li>• Str Stream – stream (order execution mode).</li> <li>• Int Csn – counter of the customer object changesw within the stream.</li> </ul>
FIMI/Response	Int*	Out	Response code, see the table with a list of available values (p. 3.1). For the SOAP protocol, in case of FIMI transaction declination, the <i>Response</i> field is specified in the standard <i>Fault</i> envelope (the <i>Detail/DeclineRp/Response/Response</i> tag).
FIMI/DeclineReason	Str	In	FIMI transaction decline reason. For the SOAP protocol, the field value will be used in the <i>Fault</i> envelope, in the <i>Reason/Text</i> tag.
FIMI/NextChallenge	Str	Out	The password encryption component value for the next terminal request (it is transmitted starting from v. 3.6)  If NULL – use the current encryption component  For the SOAP XML format, the field is transferred as a part of <i>Fault</i> (the <i>Detail/DeclineRp/Response/NextChallenge</i> tag).
FIMI/TranId	Int (12)	Out	Transaction ID in TWO

## 2.3.2 Service Messages

### 2.3.2.1 On Session:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description																																																				
<b><i>InitSession – session initialization</i></b>																																																							
FIMI/InitSession/Rq/NeedDicts	Int	In	<p>If NULL then 1</p> <p>=1 – requested dictionaries – the response will contain all the fields (dictionaries) described in the “Requested Values” column in the table below.</p> <p>=0 – the dictionaries are not sent in the response</p> <p>Besides, the field allows to define the bit mask with the requested parameters of response.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Bit Number</th><th>Requested Values</th></tr> </thead> <tbody> <tr><td>2</td><td>CNSSchemeList and CNSChannelList</td></tr> <tr><td>3</td><td>AcctOperations</td></tr> <tr><td>4</td><td>AuthSchemes</td></tr> <tr><td>5</td><td>CardProfileList, PrefixTypeList, PlasticTypesList</td></tr> <tr><td>6</td><td>CardStatList</td></tr> <tr><td>7</td><td>VISATranReasonList</td></tr> <tr><td>8</td><td>AllowedFIID</td></tr> <tr><td>9</td><td>CardVendors, VendorGroups, VendorsCustInfo and VendorGroupLinks</td></tr> <tr><td>10</td><td>Instances</td></tr> <tr><td>11</td><td>AllowedEMVScript</td></tr> <tr><td>12</td><td>AllowedCommand</td></tr> <tr><td>13</td><td>EventCodes</td></tr> <tr><td>14</td><td>ReportGroups</td></tr> <tr><td>15</td><td>CardProgramList</td></tr> <tr><td>16</td><td>Hosts</td></tr> <tr><td>17</td><td>LanguageList</td></tr> <tr><td>18</td><td>CountryList</td></tr> <tr><td>19</td><td>CurrencyList</td></tr> <tr><td>20</td><td>MerchantCategoryList, PersonCategoryList</td></tr> <tr><td>21</td><td>UserFields</td></tr> <tr><td>22</td><td>Limits, Counters</td></tr> <tr><td>23</td><td>PaymentSystem</td></tr> <tr><td>24</td><td>GroupTemplates, AcquiringGroups, RetailerGroups</td></tr> <tr><td>25</td><td>BonusPrograms</td></tr> <tr><td>26</td><td>FIMIProfiles, AccessGroups</td></tr> </tbody> </table>	Bit Number	Requested Values	2	CNSSchemeList and CNSChannelList	3	AcctOperations	4	AuthSchemes	5	CardProfileList, PrefixTypeList, PlasticTypesList	6	CardStatList	7	VISATranReasonList	8	AllowedFIID	9	CardVendors, VendorGroups, VendorsCustInfo and VendorGroupLinks	10	Instances	11	AllowedEMVScript	12	AllowedCommand	13	EventCodes	14	ReportGroups	15	CardProgramList	16	Hosts	17	LanguageList	18	CountryList	19	CurrencyList	20	MerchantCategoryList, PersonCategoryList	21	UserFields	22	Limits, Counters	23	PaymentSystem	24	GroupTemplates, AcquiringGroups, RetailerGroups	25	BonusPrograms	26	FIMIProfiles, AccessGroups
Bit Number	Requested Values																																																						
2	CNSSchemeList and CNSChannelList																																																						
3	AcctOperations																																																						
4	AuthSchemes																																																						
5	CardProfileList, PrefixTypeList, PlasticTypesList																																																						
6	CardStatList																																																						
7	VISATranReasonList																																																						
8	AllowedFIID																																																						
9	CardVendors, VendorGroups, VendorsCustInfo and VendorGroupLinks																																																						
10	Instances																																																						
11	AllowedEMVScript																																																						
12	AllowedCommand																																																						
13	EventCodes																																																						
14	ReportGroups																																																						
15	CardProgramList																																																						
16	Hosts																																																						
17	LanguageList																																																						
18	CountryList																																																						
19	CurrencyList																																																						
20	MerchantCategoryList, PersonCategoryList																																																						
21	UserFields																																																						
22	Limits, Counters																																																						
23	PaymentSystem																																																						
24	GroupTemplates, AcquiringGroups, RetailerGroups																																																						
25	BonusPrograms																																																						
26	FIMIProfiles, AccessGroups																																																						

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/InitSession/Rq/AllVendors	Int	In	If NULL then 0 =1 – CardVendors, VendorGroups and VendorsCustInfo will contain all the vendors. =0 – CardVendors, VendorGroups and VendorsCustInfo will contain the card vendors only.
FIMI/InitSession/Rq/CurrencyCodes	Int[...]	In	A list of codes of the requested currencies from the Dictionary of Currencies. If defined, bits 1 (only for the CurrencyList structure) and 19 of the <i>FIMI/InitSession/Rq/NeedDicts</i> bit mask are ignored and only the requested currencies are returned.
FIMI/InitSession/Rq/AvoidSession	Int(1)	In	=1 – without session initialization, the FIMI/Session field must be specified.
FIMI/InitSession/Rp/Id	Int*	Out	Session ID
FIMI/InitSession/Rp/NeedCAPAuth	Int(1)*	Out	DPA/CAP authentication flag
FIMI/InitSession/Rp/PasswordVersion	Int(1)*	Out	Password hash algorithm available for the operator: 1 – 3DES 2 – SHA3-256 For the algorithm of calculating the hash by the password algorithms 1(3DES) and 2(SHA3-256), refer to section 4.
FIMI/InitSession/Rp/CNSSchemeList	ArrRec	Out	{ str(20)* Name; str(200) Title; } [] – messaging schemes list Further, see the requests <i>CNSCardConfig</i> , <i>GetCardInfo</i> , <i>GetCMSBuffer</i> , <i>GetCMSArchive</i> , <i>GetCMSCommands</i> requiring information from this field
FIMI/InitSession/Rp/CNSChannelList	ArrRec	Out	{ str(20) Name; str(200) Title; int (1)* UsedForPush; } [] – messaging channels list UsedForPush – the channel is used for PUSH notifications. Further see the requests <i>CNSCardConfig</i> , <i>GetCMSBuffer</i> , <i>GetCMSArchive</i> , <i>GetCMSCommands</i> requiring information from this field

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/InitSession/Rp/ AcctOperations	ArrRec	Out	{ int Id; str(80) Name; str(80) Name_Rus; } [] – operation codes dictionary
FIMI/InitSession/Rp/ AuthSchemes	ArrRec	Out	{ int Id; str(150) Title; int(2)* Type; } [] – authorization schemes dictionary; Further, see the GetCardLimits request where you should specify the authorization scheme ID for obtaining the current value of limits counter Type – scheme type. Available values: 1 – Positive; 2 – Positive with account access; 3 – Positive with account search; 11 – Negative; 12 – Negative with accumulation; 21 – Cumulative acquiring limits.
FIMI/InitSession/Rp/ CardProfileList	ArrRec	Out	{ int Id; str(150) Title; str(4) InstName } [] – the list of card profiles; Further see the requests <i>SetCardProfile</i> , <i>GetCMSBuffer</i> , <i>GetCMSArchive</i> , <i>GetCMSCommands</i> requiring information from this field. InstName – card profile institution.
FIMI/InitSession/Rp/ GroupTemplates	ArrRec	Out	{ int* Id; int*(1) Type; str(50) Title; } [] – a list of group templates: Type – group template type: 1 – card profile templates, 2 – acquiring group templates; Title – name of group template.
FIMI/InitSession/Rp/ AcquiringGroups	ArrRec	Out	{ int* Id; str(150) Title; } [] – a list of acquiring groups.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																
FIMI/InitSession/Rp/ RetailerGroups	ArrRec	Out	{ int* Id; str(30)* Name; str(50) Title; str(4)* InstName; } [] – a list of retailers group InstName – retailer groups																
FIMI/InitSession/Rp/CardStatList	ArrRec	Out	{ int Id; str(240) Name; } [] – card status dictionary																
FIMI/InitSession/Rp/ VISATranReasonList	ArrRec	Out	{ int Id; str(450) Name; int PSType; int Usage; } [] – dictionary of Visa administrative transaction generation reasons Name – transaction reason name; PSType – payment system type (1=VISA, 2=MasterCard, 3=Local) Usage – indicates whether it is allowed to use the reason for generating the administrative transaction. Bit mask: <table border="1"> <thead> <tr> <th>Bit Number</th><th>Allowed Transaction</th></tr> </thead> <tbody> <tr> <td>1</td><td>Adjustment</td></tr> <tr> <td>2</td><td>Chargeback</td></tr> <tr> <td>3</td><td>Chargeback Reversal</td></tr> <tr> <td>4</td><td>Representment</td></tr> <tr> <td>5</td><td>Fee Collection / Funds Disbursement</td></tr> <tr> <td>6</td><td>Acquirer Reversal</td></tr> <tr> <td>7</td><td>Visa-Generated Reversal Advice</td></tr> </tbody> </table>	Bit Number	Allowed Transaction	1	Adjustment	2	Chargeback	3	Chargeback Reversal	4	Representment	5	Fee Collection / Funds Disbursement	6	Acquirer Reversal	7	Visa-Generated Reversal Advice
Bit Number	Allowed Transaction																		
1	Adjustment																		
2	Chargeback																		
3	Chargeback Reversal																		
4	Representment																		
5	Fee Collection / Funds Disbursement																		
6	Acquirer Reversal																		
7	Visa-Generated Reversal Advice																		
FIMI/InitSession/Rp/ PaymentSystems	ArrRec	Out	{ int* Id; str(30)* Name; str(150) Title; } [] – list of payment systems: Name – unique name of payment system; Title – payment system title.																

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>																														
FIMI/InitSession/Rp/AllowedFIID	ArrRec	Out	{ str(4) Name; str(150) Title; int AccessMask; int(1)* External; int* Id } [] – list of the allowed institutions:																														
Name – institution unique name. Title – institution title. AccessMask – objects that can be accessed, bit mask:																																	
<table border="1"> <thead> <tr> <th><b>Bit Number</b></th><th><b>Allowed Object</b></th></tr> </thead> <tbody> <tr><td>1</td><td>All</td></tr> <tr><td>2</td><td>Card</td></tr> <tr><td>3</td><td>Account</td></tr> <tr><td>4</td><td>Customer – private customer</td></tr> <tr><td>5</td><td>Issuer History</td></tr> <tr><td>6</td><td>Retailer</td></tr> <tr><td>7</td><td>Terminal</td></tr> <tr><td>8</td><td>Currency Rates</td></tr> <tr><td>9</td><td>Messages</td></tr> <tr><td>10</td><td>Devices</td></tr> <tr><td>11</td><td>Arrests</td></tr> <tr><td>12</td><td>E-Commerce merchants</td></tr> <tr><td>13</td><td>Invoice</td></tr> <tr><td>14</td><td>Private customers</td></tr> </tbody> </table>				<b>Bit Number</b>	<b>Allowed Object</b>	1	All	2	Card	3	Account	4	Customer – private customer	5	Issuer History	6	Retailer	7	Terminal	8	Currency Rates	9	Messages	10	Devices	11	Arrests	12	E-Commerce merchants	13	Invoice	14	Private customers
<b>Bit Number</b>	<b>Allowed Object</b>																																
1	All																																
2	Card																																
3	Account																																
4	Customer – private customer																																
5	Issuer History																																
6	Retailer																																
7	Terminal																																
8	Currency Rates																																
9	Messages																																
10	Devices																																
11	Arrests																																
12	E-Commerce merchants																																
13	Invoice																																
14	Private customers																																
External – indicates whether the payment system of the financial institution is external (0 – local payment system, 1 – external payment system). Id – financial institution ID.																																	
FIMI/InitSession/Rp/AllowedCommand	ArrRec	Out	{ int* Id; str(240) Title; } – list of allowed commands:  Id – command code; Title – command title																														
FIMI/InitSession/Rp/CardVendors	ArrRec	Out	{ str(30)* Acct; str(300) Title; str(150) Phone; str(300) Address; str(30) SettlementAccount; str(30) CorrespAccount; str(20) INN; str(15) BIC; int* VendorGroupId; int(1)* Telebank; int(1)* Personal; int(1)* Mobile; int(1)* Free; int(1)* Auto																														

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
			<p>} [] – list of vendors which can be linked to a card by the AddCardVendor operation:</p> <p>Acct – vendor account number or vendor code in Back-Office.</p> <p>Title – vendor title in national characters.</p> <p>SettlementAccount – settlement account.</p> <p>CorrespAccount – correspondent account in SWIFT, TPN, BIC.</p> <p>VendorGroupId – vendor group ID.</p> <p>The structure is sorted by the VendorGroupId field.</p> <p>Telebank – Telebank payments (0 – not allowed, 1 – allowed, 2 – temporary not allowed).</p> <p>Personal – personal payments (0 – not allowed, 1 – allowed, 2 – temporary not allowed).</p> <p>Mobile – mobile payments (0 – not allowed, 1 – allowed, 2 – temporary not allowed).</p> <p>Free – free payments (0 – not allowed, 1 – allowed, 2 – temporary not allowed).</p> <p>Auto – available autopayments (0 – not allowed, 1 – allowed, 2 – temporary not allowed).</p>
FIMI/InitSession/Rp/VendorGroups	ArrRec	Out	<pre>{     Int* GroupId;     str(30)* Name;     str(50) Title; } [] – array of structures – list of vendor groups</pre>
FIMI/InitSession/Rp/VendorGroupLinks	ArrRec	Out	<pre>{     int* GroupId;     str(30)* VendorAcct; } [] – array of structures – list of links between vendors and vendor groups</pre>
FIMI/InitSession/Rp/VendorsCustInfo	ArrRec	Out	<pre>{     str(30)* VendorAcct;     str(20)* Name;     str(180) Title;     str(1500) Annotation;     str(1000) Mask; } [] – array of structures – list of the vendor groups</pre>
Payer extra personal attributes:			
<p>VendorAcct – vendor account defined in the CardVendors field;</p> <p>Name – line field identifier;</p> <p>Title – short field title;</p> <p>Annotation – full description of the personal information;</p> <p>Mask – input mask (controlled by the terminal).</p>			
FIMI/InitSession/Rp/VendorsServiceCustInfo	ArrRec	Out	<pre>{     str(30)* VendorAcct;     str(20)* Name;     int* SeqNo;     int(1)* Mandatory;     str(180) Title;     str(1500) Annotation;     str(1000) Mask;     int(1)* EditAllowed; } [] – array of structures</pre>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																
Payer extra personal attributes required for performing the autopayment to vendor:																			
VendorAcct – vendor account defined in the CardVendors field; Name – line field identifier; SeqNo – field sequence number; Mandatory – indicates that the payment attribute value is mandatory; Title – short field title; Annotation – full description of the personal information; Mask – input mask (controlled by the terminal); EditAllowed – indicates whether the attribute is editable (controlled by the terminal).																			
FIMI/InitSession/Rp/Instances	ArrRec	Out	{ int* Id; str(150) Title; } [] – list of the system instance																
FIMI/InitSession/Rp/ AllowedEMVScript	Int	Out	Allowed types of EMV scripts for an operator, bit mask: <table border="1"> <thead> <tr> <th>Bit Number</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Application block</td> </tr> <tr> <td>2</td> <td>Card block</td> </tr> <tr> <td>3</td> <td>Pin unblock</td> </tr> <tr> <td>4</td> <td>Put data</td> </tr> <tr> <td>5</td> <td>Update record</td> </tr> <tr> <td>6</td> <td>Application unblock</td> </tr> <tr> <td>7</td> <td>PIN Change</td> </tr> </tbody> </table>	Bit Number	Value	1	Application block	2	Card block	3	Pin unblock	4	Put data	5	Update record	6	Application unblock	7	PIN Change
Bit Number	Value																		
1	Application block																		
2	Card block																		
3	Pin unblock																		
4	Put data																		
5	Update record																		
6	Application unblock																		
7	PIN Change																		
FIMI/InitSession/Rp/ CardStatusTransitions	ArrRec	Out	{ int* InitialStatus; int* AllowableStatus; } [] – list of the allowed transitions by changing the card status (the SetCardStatus operation) For the codes list, see p. 3.2.																
FIMI/InitSession/Rp/ NoAllowedTransitions	Int(1)	Out	Indicates that there are no allowed transitions of card statuses																
FIMI/InitSession/Rp/EventCodes	ArrRec	Out	{ int* Code; str(500)* Mess; } [] – list of the event codes. Mess – message (transferred depending on the value in the FIMI/Language field).																

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/InitSession/Rp/ReportGroups	ArrRec	Out	{ int* Id; int ParentId; str(300) Title; } [] – a list of report groups and subgroups. Title – group title (transferred depending on the value in the <i>FIMI/Language</i> field).
FIMI/InitSession/Rp/CardProgramList	ArrRec	Out	{ int* Id; str(300) Title; } [] – a list of card products.
FIMI/InitSession/Rp/Hosts	ArrRec	Out	{ int* Id; str*(200) Name; } [] – list of hosts.
FIMI/InitSession/Rp/LanguageList	ArrRec	Out	{ int* Id; str(240) Language; str(2)* Alpha2Code; } [] – a list of languages.
FIMI/InitSession/Rp/CountryList	ArrRec	Out	{ int* Id; str(240) Country; str(2) Alpha2Code; str(3) Alpha3Code; int(3) RiskLevel; } [] – a list of countries.
FIMI/InitSession/Rp/CurrencyList	ArrRec	Out	{ int* Id; str(240) Currency; str(3) Alpha3Code; str(10) AfterPrint; str(10) BeforePrint; int DecimalPlaces; } [] – a list of currencies
FIMI/InitSession/Rp/MerchantCategoryList	ArrRec	Out	{ int* MCC; str(240) Name; int(3) RiskLevel; str(1)* TCC; } [] – a list of SIC/MCC
FIMI/InitSession/Rp/LimitedAccessToUserFields	Int(1)	Out	Indicates whether the access to the Card, Account and Customer user fields is restricted. If =1, only the user fields specified in the <i>FIMI/InitSession/Rp/UserFields</i> structure are available.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																																						
FIMI/InitSession/Rp/UserFields	ArrRec	Out	{ int*(1) Type; str*(100) Name; int*(1) Editable } [] – a list of available user fields.																																						
Type – field type: 1 – Customer, 2 – Card, 3 – Telebank customer, 4 – Account. Name – field name. Editable – attribute indicating that the field is available for editing, adding and deleting.																																									
FIMI/InitSession/Rp/Limits	ArrRec	Out	{ int* Id; str(150)* Name; int(1)* Type; str(100) Title; int(4)* Editable; int(1)* IsCounter; } [] – a list of created limits:																																						
Id – limit ID. Name – limit name. Type – bit mask indicating the limit belonging:																																									
<table border="1"><thead><tr><th>Bit Number</th><th>Belonging</th></tr></thead><tbody><tr><td>1</td><td>Card</td></tr><tr><td>2</td><td>Account</td></tr><tr><td>3</td><td>Card-to-account link</td></tr><tr><td>4</td><td>Customer</td></tr><tr><td>5</td><td>Token</td></tr><tr><td>6</td><td>Retailer</td></tr><tr><td>7</td><td>Group of retailers</td></tr></tbody></table>	Bit Number	Belonging	1	Card	2	Account	3	Card-to-account link	4	Customer	5	Token	6	Retailer	7	Group of retailers																									
Bit Number	Belonging																																								
1	Card																																								
2	Account																																								
3	Card-to-account link																																								
4	Customer																																								
5	Token																																								
6	Retailer																																								
7	Group of retailers																																								
Title – limit description (transferred depending on the value in the FIMI/Language field). Editable – bit mask indicating the types of limit that can be edited:																																									
<table border="1"><thead><tr><th>Bit Number</th><th>Limit Object</th></tr></thead><tbody><tr><td>1</td><td>Card limit</td></tr><tr><td>2</td><td>Temporary card limit</td></tr><tr><td>3</td><td>Accouint limit</td></tr><tr><td>4</td><td>Temporary account limit</td></tr><tr><td>5</td><td>Card-to-account link limit</td></tr><tr><td>6</td><td>Customer limit</td></tr><tr><td>7</td><td>Temporary customer limit</td></tr><tr><td>8</td><td>Card limit range</td></tr><tr><td>9</td><td>Account limit range</td></tr><tr><td>10</td><td>Customer limit range</td></tr><tr><td>11</td><td>Card limit counter</td></tr><tr><td>12</td><td>Account limit counter</td></tr><tr><td>13</td><td>Customer limit counter</td></tr><tr><td>14</td><td>Counter of the card-to-account link</td></tr><tr><td>15</td><td>Retailer limit</td></tr><tr><td>16</td><td>Retailer limit counter</td></tr><tr><td>17</td><td>Limit of group of retailers</td></tr><tr><td>18</td><td>Counter of retailer group limits</td></tr></tbody></table>	Bit Number	Limit Object	1	Card limit	2	Temporary card limit	3	Accouint limit	4	Temporary account limit	5	Card-to-account link limit	6	Customer limit	7	Temporary customer limit	8	Card limit range	9	Account limit range	10	Customer limit range	11	Card limit counter	12	Account limit counter	13	Customer limit counter	14	Counter of the card-to-account link	15	Retailer limit	16	Retailer limit counter	17	Limit of group of retailers	18	Counter of retailer group limits			
Bit Number	Limit Object																																								
1	Card limit																																								
2	Temporary card limit																																								
3	Accouint limit																																								
4	Temporary account limit																																								
5	Card-to-account link limit																																								
6	Customer limit																																								
7	Temporary customer limit																																								
8	Card limit range																																								
9	Account limit range																																								
10	Customer limit range																																								
11	Card limit counter																																								
12	Account limit counter																																								
13	Customer limit counter																																								
14	Counter of the card-to-account link																																								
15	Retailer limit																																								
16	Retailer limit counter																																								
17	Limit of group of retailers																																								
18	Counter of retailer group limits																																								
IsCounter – attribute of the quantitative limit (counter).																																									

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description										
FIMI/InitSession/Rp/ BonusPrograms	ArrRec	Out	<pre>{     int* Id;     int*(20) Name;     str(100) Title; }</pre> <p>[]} – list of bonus programs:          Id – bonus program ID.          Name – bonus program name.          Title – bonus program description</p>										
FIMI/InitSession/Rp/ PersonCategoryList	ArrRec	Out	<pre>{     int* Id;     str(500) Title; }</pre> <p>[]} – list of customer categories          Id – category ID          Title – customer category description</p>										
FIMI/InitSession/Rp/ PrefixTypeList	ArrRec	Out	<pre>{     int* Id;     str(240) Title; }</pre> <p>[]} – list of prefix types          Id – prefix type ID          Title – prefix type title</p>										
FIMI/InitSession/Rp/Counters	ArrRec	Out	<pre>{     int* Id;     str(150)* Name;     int(1)* Type;     str(100) Title;     int(1)* IsCounter; }</pre> <p>[]} – list of existing counters:</p> <p>Id – counter ID.          Name – counter name.          Type – bit mask indicating the counter belonging:</p>										
<table border="1" style="width: 100%;"> <thead> <tr> <th>Bit Number</th> <th>Belonging</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Card</td> </tr> <tr> <td>2</td> <td>Account</td> </tr> <tr> <td>3</td> <td>Card-to-account link</td> </tr> <tr> <td>4</td> <td>Customer</td> </tr> </tbody> </table>				Bit Number	Belonging	1	Card	2	Account	3	Card-to-account link	4	Customer
Bit Number	Belonging												
1	Card												
2	Account												
3	Card-to-account link												
4	Customer												
<p>Title – counter description (transferred depending on the value in the <i>FIMI/Language</i> field).          IsCounter – attribute of the quantitative counter.</p>													
FIMI/InitSession/Rp/ FIMIProfiles	ArrRec	Out	<pre>{     str(180)* Name;     str(1000) Title; }</pre> <p>[]} – list of the FIMI profile types:          Id – FIMI profile name.          Title – FIMI profile description.</p>										

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/InitSession/Rp/AccessGroups	ArrRec	Out	{           int* Id;           str(180) Title;         } [] – list of the access groups: Id – access group ID. Title – access group description.
FIMI/InitSession/Rp/PlasticTypesList	ArrRec	Out	{           Str(32)* Id;           str(250) Title;         } [] – list of plastic types
<b>Logon – operator authentication</b>			
FIMI/Logon/Rq/CAPToken	Str(40)	In	CAP-token or SMS\E-Mail dynamic password. In case of SMS\E-Mail dynamic password, it contains its hash calculated by the DES algorithm of FIMI operator password hash calculation.
FIMI/Logon/Rp/Operations	Str(50) [...]*	Out	List of the allowed operations, e.g.: “GetTermList”, “GetTransInfo”, “SetCardStatus”.
FIMI/Logon/Rp/ClerkExpiration	Time	Out	Expiration date of the operator account, if NULL – user account is unlimited
FIMI/Logon/Rp/AddressList	ArrRec	Out	{           str(20)* Channel;           str(200) Provider;           str(500)* Address;           str(100) FullAddress;           str(150) Title;           int(1)* Lock;           int(1)* Default;         } [] – list of the addresses which can be used for dynamic authentication: Channel – exchange channel Provider – channel description Address – abonent address in the system FullAddress – abonent full address Title – abonent description Lock – address is blocked Default – use by default
<b>Logoff – operator session completion</b>			
<b>GenerateDynamicPassword – dynamic password generation for operator authentication</b>			
FIMI/GenerateDynamicPassword/Rq/Channel	Str(20)*	In	Exchange channel.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GenerateDynamicPassword/ Rq/Address	Str(500)*	In	Address in the system.
FIMI/GenerateDynamicPassword/ Rp/PasswordCount	Int*	Out	Number of the generated dynamic password.
<b>Ping – updating operator session activity</b>			
<i>The operation is used to hold the session activity in case it is not required to send other requests from the FIMI customer. The operation can be also used to check whether the operator session is available and active.</i>			

### 2.3.2.2 On Operator:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>ChangePassword – operator password change</b>			
FIMI/ChangePassword/Rq/ NewPassword	Str(64)*	In	New encrypted password of FIMI operator.
<b>For password hash version 1(3DES).</b> If the protocol version is not specified or it is <= 3.5, the new password is transferred as a clear value (if the "Calculate hash for clear password" option is enabled on the terminal or password hash in the ASCII HEX format. In this case, the password must have the maximum length of 16 characters. For the protocol version later than 3.5, the new password must be encrypted with the hash of the old password by the des-ecb algorithm. If the password length is not multiple of 8, the password must be padded with spaces up to 8, 16 characters. The encrypted password must be converted to ASCII HEX.			
<b>For password hash version 2(SHA3-256).</b> If the protocol version is not specified or it is <= 3.5, the new password is transferred as a clear value (if the "Calculate hash for clear password" option is enabled on the terminal or password hash in the ASCII HEX format. In this case, the password must have the maximum length of 64 characters. If the protocol version is >= 16.13, the new password must be encrypted with the hash of the old password by the aes-256-ecb-nopad algorithm. If the password length is not multiple of 16, the password must be padded with spaces up to 16, 32, 48 characters. The encrypted password must be converted to ASCII HEX. <i>The password hash version used by the operator is returned in the PasswordVersion response field of the InitSession operation.</i>			
FIMI/ChangePassword/Rp/ NewPasswordVersion	Int(1)*	Out	Password hashing algorithm available for the operator for the new password after the change: 1 – 3DES 2 – SHA3-256 For the algorithm of calculating the hash on the password algorithms 1(3DES) and 2(SHA3-256), refer to section 4.

Parameter	Type (*- mandatory parameter)	Direction	Description
<b>ResetPassword – reset the operator password and set the new one</b>			
FIMI/ResetPassword/Rq/Address	Str(500)	In	<p>Operator address</p> <p>The field is mandatory when initiating the password reset and OTP generation.</p>
FIMI/ResetPassword/Rq/NewPassword	Str(64)	In	<p>New operator password</p> <p>The field is mandatory when the password reset process is continued (response code 52 – DynamicPasswordRequired was received to the initial request), it is transferred with the OTP field.</p> <p>It is transferred encrypted using challenge:</p>
<p><b>For password hash version 1(3DES).</b></p> <p>The new password must be encrypted with challenge (transferred in the FIMI/NextChallenge field when the password reset and OTP sending are initiated) by the des-ecb algorithm. If the password length is not multiple of 8, the password must be padded with spaces up to 8, 16 characters. The encrypted password must be converted to ASCII HEX.</p> <p><b>For password hash version 2(SHA3-256).</b></p> <p>The new password must be encrypted with challenge (transferred in the FIMI/NextChallenge field when the password reset and OTP sending are initiated) by the aes-128-ecb-nopad algorithms. If the password length is not multiple of 16, the password must be padded with spaces up to 16, 32, 48 characters. The encrypted password must be converted to ASCII HEX.</p> <p><i>The password hash version used by the operator is returned in the PasswordVersion response field of the InitSession operation.</i></p>			
FIMI/ResetPassword/Rq/OTP	Str(40)	In	<p>Dynamic password hash.</p> <p>For the hash calculation algorithm, refer to section 4.3.</p> <p>The field is mandatory when the password reset process is continued (response code 52 – DynamicPasswordRequired was received to the initial request), it is transferred with the NewPassword field.</p>
FIMI/ResetPassword/Rq/RecoveryLink	Str(1000)	In	<p>URL link for the password recovery. If specified, it is transferred in the message with OTP.</p> <p>The field is used when password reset and OTP sending are initiated.</p>
FIMI/ResetPassword/Rp/NewPasswordVersion	Int(1)*	Out	<p>Password hashin algorithm available for the operator for the new password after the change:</p> <p>1 – 3DES</p> <p>2 – SHA3-256</p> <p>For the algorithm of calculating the hash on the password algorithms 1(3DES) and 2(SHA3-256), refer to section 4.</p>

Parameter	Type (*- mandatory parameter)	Direction	Description
<b>GetFIMIClerks – request for list of operators with terminal IDs</b>			
FIMI/GetFIMIClerks/Rq/Clerk	Str(30)*	In	<p>Name of the FIMI terminal operator.</p> <p><i>The mask can be defined in the format of SQL-operator 'like', i.e. applying the metacharacters '_' (any single character), '%' (any number of any characters).</i></p>
FIMI/GetFIMIClerks/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution of the FIMI terminal.</p> <p>It is used if the field is transferred in the TerminalName field.</p> <p><i>If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.</i></p>
FIMI/GetFIMIClerks/Rq/TerminalName	Str(16)	In	<p>FIMI terminal name.</p> <p><i>The mask can be defined in the format of SQL-operator 'like', i.e. applying the metacharacters '_' (any single character), '%' (any number of any characters).</i></p>
FIMI/GetFIMIClerks/Rp>List	ArrRec	Out	<pre>{     str(30)* Clerk;     int* TermId;     time* LastPwdChangeTime;     int(4)* PwdValidPeriod;     int(3)* MaskPAN;     int(1)* MaskBalances;     str(16)* TerminalName;     int(1) Blocked;     time BlockTime;     int AccessGroup;     str(180) FIMIProfile;     int(1) Access2VIP;     int(4) MaxInactivityPeriod;     int(3) BadPasswordMax;     time LastLogonTime;     time CreationDate;     str(19) AuthPAN;     int(3) AuthMBR;     str(32) AuthCardUID;     int(1) AuthType;     str(100) Description;     str(180) FamilyName;     str(25) StaffID;     str(180) Department;     str(180) PhoneNumber;     str(100) Email;     str(180) Designation;     time StartDate;     time EndDate; }</pre> <p>} [] – list of the FIMI terminal operators with the terminal identifiers:</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																						
Clerk – name of the FIMI terminal operator.																									
TermId – terminal ID.																									
LastPwdChangeTime – time of the last change of the operator password.																									
PwdValidPeriod – password validity period (in days), 0 – password validity period is unlimited.																									
MaskPAN – masking data in response messages:																									
	<table border="1"> <thead> <tr> <th>Bit Number</th><th>Description</th></tr> </thead> <tbody> <tr><td>1</td><td>Card numbers</td></tr> <tr><td>2</td><td>Account numbers</td></tr> <tr><td>3</td><td>Magnetic tracks</td></tr> <tr><td>4</td><td>PVV / IPVV;</td></tr> <tr><td>5</td><td>Telebank customer ID</td></tr> <tr><td>6</td><td>Name on card</td></tr> <tr><td>7</td><td>Full name of the customer</td></tr> <tr><td>8</td><td>Document number</td></tr> <tr><td>9</td><td>Customer birth place</td></tr> <tr><td>10</td><td>Customer address</td></tr> </tbody> </table>	Bit Number	Description	1	Card numbers	2	Account numbers	3	Magnetic tracks	4	PVV / IPVV;	5	Telebank customer ID	6	Name on card	7	Full name of the customer	8	Document number	9	Customer birth place	10	Customer address		
Bit Number	Description																								
1	Card numbers																								
2	Account numbers																								
3	Magnetic tracks																								
4	PVV / IPVV;																								
5	Telebank customer ID																								
6	Name on card																								
7	Full name of the customer																								
8	Document number																								
9	Customer birth place																								
10	Customer address																								
MaskBalances – 'Mask ledger and available balances for operator' attribute.																									
TerminalName – terminal name.																									
Blocked – block operator:																									
0 – not blocked.																									
1 – blocked without any reason.																									
2 – blocked due to the inactivity timeout.																									
3 – blocked as the number of invalid password tries exceeded.																									
4 – manual blocking.																									
BlockTime – blocking time.																									
AccessGroup – access group (for the list of available values, see <i>FIMI/InitSession/Rp/AccessGroups</i> ).																									
FIMIProfile – operator FIMI profile (for the list of available values, see <i>FIMI/InitSession/Rp/FIMIProfiles</i> ).																									
Access2VIP – access to VIP customers:																									
0 – all, except for VIP.																									
1 – all.																									
2 – VIP only.																									
MaxInactivityPeriod – maximum number of days of inactivity before blocking, 0 – unlimited.																									
BadPasswordMax – maximum number of invalid password tries, 0 – unlimited.																									
LastLogonTime – time of the last logon (executing the Logon operation).																									
CreationDate – creation date.																									
AuthPAN, AuthMBR, AuthCardUID – card for the dynamic authentication.																									
AuthType – authentication type:																									
0 – password.																									
1 – password and CAP dynamic authentication.																									
3 – password and SMS/Email dynamic authentication.																									
Description – description.																									
FamilyName – personal name.																									
StaffID – employee number.																									
Department – department.																									
PhoneNumber – phone numbers.																									
Email – E-mail.																									
Designation – position.																									
StartDate – start date of the work.																									
EndDate – end date of the work.																									
<b>Fields Blocked, BlockTime, AccessGroup, FIMIProfile, Access2VIP, MaxInactivityPeriod, BadPasswordMax, LastLogonTime, CreationDate, AuthPAN, AuthMBR, AuthCardUID, AuthType, Description, FamilyName, StaffID, Department, PhoneNumber, Email, Designation, StartDate, EndDate are returned only if the operator executing the request is allowed to manage FIMI operators.</b>																									

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>CreateFIMIClerk – operator registration</b>			
FIMI/CreateFIMIClerk/Rq/Clerk	Str(30)*	In	Name of the FIMI terminal operator
FIMI/CreateFIMIClerk/Rq/Password	Str(64)*	In	Encrypted password of the FIMI operator
<b>For password hash version 1(3DES).</b> If the protocol version is not specified or it is <= 3.5, the new password is transferred as a clear value (if the "Calculate hash for clear password" option is enabled on the terminal or password hash in the ASCII HEX format. In this case, the password must have the maximum length of 16 characters.  For the protocol version later than 3.5, the new password must be encrypted with the hash of the old password by the des-ecb algorithm. If the password length is not multiple of 8, the password must be padded with spaces up to 8, 16 characters. The encrypted password must be converted to ASCII HEX.			
<b>For password hash version 2(SHA3-256).</b> If the protocol version is not specified or it is <= 3.5, the new password is transferred as a clear value (if the "Calculate hash for clear password" option is enabled on the terminal or password hash in the ASCII HEX format. In this case, the password must have the maximum length of 64 characters.  If the protocol version is >= 16.13, the new password must be encrypted with the hash of the old password by the aes-256-ecb-nopad algorithm. If the password length is not multiple of 16, the password must be padded with spaces up to 16, 32, 48 characters. The encrypted password must be converted to ASCII HEX.			
<i>The password hash version used by the operator is returned in the PasswordVersion response field of the InitSession operation.</i>  For the algorithm of calculating the hash on the password algorithms 1(3DES) and 2(SHA3-256), refer to section 4.			
FIMI/CreateFIMIClerk/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution of the FIMI terminal.  It is used if the field is transferred in the TerminalName field. <i>If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.</i>
FIMI/CreateFIMIClerk/Rq/TermName	Str(16)*	In	FIMI terminal name
FIMI/CreateFIMIClerk/Rq/AccessGroup	Int*	In	Access group  See the list of available values in FIMI/InitSession/Rp/AccessGroups.
FIMI/CreateFIMIClerk/Rq/FIMIProfile	Str(180)	In	Operator FIMI profile.  See the list of available values in FIMI/InitSession/Rp/FIMIProfiles.
FIMI/CreateFIMIClerk/Rq/Access2VIP	Int(1)	In	Access to VIP customers: 0 – all, except for VIP. 1 – all. 2 – VIP only. If NULL, it is 1

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																						
FIMI/CreateFIMIClerk/Rq/ MaskPAN	Int(3)	In	<p>Masking data in response messages:</p> <table border="1"> <thead> <tr> <th>Bit Number</th><th>Description</th></tr> </thead> <tbody> <tr><td>1</td><td>Card numbers</td></tr> <tr><td>2</td><td>Account numbers</td></tr> <tr><td>3</td><td>Magnetic tracks</td></tr> <tr><td>4</td><td>PVV / IPVV;</td></tr> <tr><td>5</td><td>Telebank customer ID</td></tr> <tr><td>6</td><td>Name on card</td></tr> <tr><td>7</td><td>Full name of the customer</td></tr> <tr><td>8</td><td>Document number</td></tr> <tr><td>9</td><td>Customer birth place</td></tr> <tr><td>10</td><td>Customer address</td></tr> </tbody> </table> <p>If NULL, it is 13 (1+3+4 bits).</p>	Bit Number	Description	1	Card numbers	2	Account numbers	3	Magnetic tracks	4	PVV / IPVV;	5	Telebank customer ID	6	Name on card	7	Full name of the customer	8	Document number	9	Customer birth place	10	Customer address
Bit Number	Description																								
1	Card numbers																								
2	Account numbers																								
3	Magnetic tracks																								
4	PVV / IPVV;																								
5	Telebank customer ID																								
6	Name on card																								
7	Full name of the customer																								
8	Document number																								
9	Customer birth place																								
10	Customer address																								
FIMI/CreateFIMIClerk/Rq/ MaskBalances	Int(1)	In	<p>Attribute indicating that ledger and available balances must be masked for a certain operator.</p> <p>If NULL, it is 0.</p>																						
FIMI/CreateFIMIClerk/Rq/ PwdValidPeriod	Int(4)	In	<p>Password validity period (in days), 0 – password validity period is unlimited.</p> <p>If NULL, it is 0.</p>																						
FIMI/CreateFIMIClerk/Rq/ MaxInactivityPeriod	Int(4)	In	<p>Maximum days of inactivity before blocking, 0 – unlimited</p> <p>If NULL, it is 0.</p>																						
FIMI/CreateFIMIClerk/Rq/ BadPasswordMax	Int(3)	In	<p>Maximum number of invalid password tries, 0 – unlimited.</p> <p>If NULL, it is 0.</p>																						
FIMI/CreateFIMIClerk/Rq/AuthType	Int(1)	In	<p>Authentication type:</p> <ul style="list-style-type: none"> <li>0 – password.</li> <li>1 – password and CAP dynamic authentication.</li> <li>3 – password and SMS/Email dynamic authentication.</li> </ul> <p>If NULL, it is 0.</p>																						
FIMI/CreateFIMIClerk/Rq/AuthPAN	Str(19)	In	<p>Card for dynamic authentication. UID of the card (AuthCardUID) can be transferred instead of AuthPAN and AuthMBR.</p> <p>Card is mandatory for dynamic authentication (AuthType=1 or 3).</p>																						
FIMI/CreateFIMIClerk/Rq/AuthMBR	Int(3)	In																							
FIMI/CreateFIMIClerk/Rq/ AuthCardUID	Str(32)	In																							
FIMI/CreateFIMIClerk/Rq/ Description	Str(100)	In	Description																						
FIMI/CreateFIMIClerk/Rq/ FamilyName	Str(180)	In	Personal name																						
FIMI/CreateFIMIClerk/Rq/ StaffID	Str(25)	In	Employee number																						
FIMI/CreateFIMIClerk/Rq/ Department	Str(180)	In	Department																						
FIMI/CreateFIMIClerk/Rq/ PhoneNumber	Str(180)	In	Phone numbers																						
FIMI/CreateFIMIClerk/Rq/ Email	Str(100)	In	E-mail																						

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateFIMIClerk/Rq/Designation	Str(180)	In	Position
FIMI/CreateFIMIClerk/Rq/StartDate	Time	In	Start date of the work. If NULL, starting from the creation time.
FIMI/CreateFIMIClerk/Rq/EndDate	Time	In	End date of the work If NULL, it is unlimited.
<b><i>SetFIMIClerkActivityStatus – setting operator activity attribute (blocking/unblocking)</i></b>			
FIMI/SetFIMIClerkActivityStatus/Rq/Clerk	Str(30)*	In	Name of FIMI terminal operator
FIMI/SetFIMIClerkActivityStatus/Rq/NewStatus	Int(1)*	In	Operator status: 0 – blocked. 1 – active.
FIMI/SetFIMIClerkActivityStatus/Rq/ChangeReason	Str(1000)	In	Status change reason
<b><i>SetNeedChangePassword – setting attribute indicating that operator password must be changed</i></b>			
FIMI/SetNeedChangePassword/Rq/Clerk	Str(30)*	In	Name of FIMI terminal operator
FIMI/SetNeedChangePassword/Rq/ChangeReason	Str(1000)	In	Reason for setting the password change attribute
<b><i>DeleteOperatorSessions – deleting all operator sessions</i></b>			
FIMI/DeleteOperatorSessions/Rq/Clerk	Str(30)*	In	Name of FIMI terminal operator
FIMI/DeleteOperatorSessions/Rq/ChangeReason	Str(1000)	In	Reason for deleting the FIMI operator sessions
<b><i>GetFIMIClerkProperties – request for list of operator properties</i></b>			
FIMI/GetFIMIClerkProperties/Rq>List	Str(60) [...]	In	List of properties being requested. If not defined, the list of all properties of the FIMI terminal operator will be returned.
FIMI/GetFIMIClerkProperties/Rp/List	ArrRec	Out	{ str(60)* Property: str Value } [] – list of parameters and their values; Property – property name. Value – property value.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>SetFIMIClerkProperties – deleting, creating and changing operator properties</b>			
FIMI/SetFIMIClerkProperties/Rq/List	ArrRec	In	<pre>{   Str(60)* Property;   Str Value } [] – list of parameters and their values; Property – property name Value – property value. If it = NULL, delete the property, &lt;&gt; NULL, add/change the property</pre>
<b>GetFIMIClerkAddresses – request for addresses for operator connections</b>			
FIMI/GetFIMIClerkAddresses/Rq/Clerk	Str(30)*	In	Name of FIMI terminal operator
FIMI/GetFIMIClerkAddresses/Rp/TermId	Int*	Out	Identifier of the terminal to which the FIMI user belongs
FIMI/GetFIMIClerkAddresses/Rp/Addresses	ArrRec	Out	<pre>{   int* UnitId;   str(150) Address;   int(1)* UnitOnline } [] – a list of units and their addresses; UnitId – unit ID; Address – network port address. UnitOnline – indicates whether the unit is available for the requests execution.</pre>

### 2.3.2.3 FIMI Transaction Operations:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>BeginTransaction – transaction start</b>			
FIMI/BeginTransaction/Rq/Savepoint	Str(30)	In	<p>Name of the savepoint which is defined within the current transaction.  The field name must comply with the rules of generating identifiers in Oracle:  - it must contain the Latin letters, digits and character “_”;  - name length must not exceed 30 characters;  - name must begin with the Latin letter.</p> <p>If the field is empty, it is considered that a new transaction is requested.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/BeginTransaction/Rq/Force	Int(1)	In	If =1, the repeated processing of the name of savepoint already in use that was sent earlier is allowed
<b>Commit – transaction completion</b>			
<b>RollBack – transaction rollback</b>			
FIMI/RollBack/Rq/Savepoint	Str	In	Name of the savepoint up to which the current transaction is rolled back  If the field is empty, the transaction full rollback is requested

#### 2.3.2.4 Offline Mode Operations:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description														
<b>FileHeader – header operation for Offline mode</b>																	
FIMI/FileHeader/Rq/Counters	Int(2)	In	<p>Bit mask of the requested counters. If NULL, the counters are not returned. If the output file is encrypted, counters are not returned; it is required to use <i>FileTrailer</i>.</p> <table border="1"> <thead> <tr> <th>Bit Number</th><th>Counter Type in Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>TotalRequests</td></tr> <tr> <td>2</td><td>ApprovedRequests</td></tr> <tr> <td>3</td><td>DeclinedRequests</td></tr> <tr> <td>4</td><td>TotalAmount</td></tr> <tr> <td>5</td><td>ApprovedAmount</td></tr> <tr> <td>6</td><td>DeclinedAmount</td></tr> </tbody> </table>	Bit Number	Counter Type in Response	1	TotalRequests	2	ApprovedRequests	3	DeclinedRequests	4	TotalAmount	5	ApprovedAmount	6	DeclinedAmount
Bit Number	Counter Type in Response																
1	TotalRequests																
2	ApprovedRequests																
3	DeclinedRequests																
4	TotalAmount																
5	ApprovedAmount																
6	DeclinedAmount																
FIMI/FileHeader/Rq/CRC	Str(8)	In	Checksum of the input file (CRC32). It is calculated on all requests in the file, except for <i>FileTrailer</i> , <i>FileHeader</i> . If the field is filled, the checksum of the input file is checked and the CRC field is filled in the response. If the output file is encrypted, the CRC field in the response is not filled; it is required to use <i>FileTrailer</i> .														
FIMI/FileHeader/Rp/TotalRequests	Int	Out	Counter of all requests in the file (the <i>FileTrailer</i> , <i>FileHeader</i> requests are not considered)														

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/FileHeader/Rp/ ApprovedRequests	Int	Out	Counter of approved requests (the <i>FileTrailer</i> , <i>FileHeader</i> requests are not considered)
FIMI/FileHeader/Rp/ DeclinedRequests	Int	Out	Counter of declined requests (the <i>FileTrailer</i> , <i>FileHeader</i> requests are not considered).  If the authorizer declines the transaction (to the <i>POSRequest</i> request) generated by FIMI terminal (it is approved by FIMI terminal), such request is considered in the <i>DeclinedRequests</i> counter.
FIMI/FileHeader/Rp/TotalAmount	Str(16)	Out	Amount counter for the <i>Amount</i> field of the following transactions: <i>AcctCredit</i> , <i>AcctDebit</i> and <i>POSRequest</i> . The response string is in the <i>Num(12,3)</i> format, the nonsignificant bytes are left or right-padded with zeros
FIMI/FileHeader/Rp/ ApprovedAmount	Str(16)	Out	Amount counter for the <i>Amount</i> field of the approved transactions: <i>AcctCredit</i> , <i>AcctDebit</i> and <i>POSRequest</i> . The response string is in the <i>Num(12,3)</i> format, the nonsignificant bytes are left or right-padded with zeros
FIMI/FileHeader/Rp/ DeclinedAmount	Str(16)	Out	Amount counter for the <i>Amount</i> field of the declined transactions: <i>AcctCredit</i> , <i>AcctDebit</i> and <i>POSRequest</i> . The response string is in the <i>Num(12,3)</i> format, the nonsignificant bytes are left or right-padded with zeros.  If the authorizer declines the transaction (to the <i>POSRequest</i> request) generated by FIMI terminal (it is approved by FIMI terminal), such request is considered in the <i>DeclinedAmount</i> counter
FIMI/FileHeader/Rp/CRC	Str(8)	Out	Checksum of the input file (CRC32). It is calculated on all requests in the file, except for <i>FileTrailer</i> , <i>FileHeader</i> .
<b><i>FileTrailer – final operation for Offline mode</i></b>			
FIMI/FileTrailer/Rq/Counters	Int(2)	In	Bit mask of the requested counters. If NULL, the counters are not returned. For the list of available values, see FIMI/FileHeader/Rq/Counters.
FIMI/FileTrailer/Rq/CalculateCRC	Int(1)	In	Checksum of the output file (CRC32). If = 1, the response returns the CRC field. If NULL then 0.
FIMI/FileTrailer/Rp/TotalRequests	Int	Out	Counter of all requests in the file (the <i>FileTrailer</i> , <i>FileHeader</i> requests are not considered)
FIMI/FileTrailer/Rp/ ApprovedRequests	Int	Out	Counter of approved requests (the <i>FileTrailer</i> , <i>FileHeader</i> requests are not considered)

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/FileTrailer/Rp/ DeclinedRequests	Int	Out	Counter of declined requests (the <i>FileTrailer</i> , <i>FileHeader</i> requests are not considered).  If the authorizer declines the transaction (to the <i>POSRequest</i> request) generated by FIMI terminal (it is approved by FIMI terminal), such request is considered in the <i>DeclinedRequests</i> counter.
FIMI/FileTrailer/Rp/TotalAmount	Str(16)	Out	Amount counter for the <i>Amount</i> field of the following transactions: <i>AcctCredit</i> , <i>AcctDebit</i> and <i>POSRequest</i> . The response string is in the <i>Num(12,3)</i> format, the nonsignificant bytes are left or right-padded with zeros.
FIMI/FileTrailer/Rp/ ApprovedAmount	Str(16)	Out	Amount counter for the <i>Amount</i> field of the approved transactions: <i>AcctCredit</i> , <i>AcctDebit</i> and <i>POSRequest</i> . The response string is in the <i>Num(12,3)</i> format, the nonsignificant bytes are left or right-padded with zeros.
FIMI/FileTrailer/Rp/ DeclinedAmount	Str(16)	Out	Amount counter for the <i>Amount</i> field of the declined transactions: <i>AcctCredit</i> , <i>AcctDebit</i> and <i>POSRequest</i> . The response string is in the <i>Num(12,3)</i> format, the nonsignificant bytes are left or right-padded with zeros.  If the authorizer declines the transaction (to the <i>POSRequest</i> request) generated by FIMI terminal (it is approved by FIMI terminal), such request is considered in the <i>DeclinedAmount</i> counter
FIMI/FileTrailer/Rp/CRC	Str(8)	Out	Checksum of the output file (CRC32). It is calculated on all requests in the file, except for <i>FileTrailer</i> , <i>FileHeader</i> .

### 2.3.2.5 Operations for Getting Reference Data:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetTranId – request for transaction ID</b>			

Parameter	Type (* - mandatory parameter)	Direction	Description
<b>GetVCProducts – request for card products list</b>			
FIMI/GetVCProducts/Rq/CardProfile	Int	In	Card profile ID
FIMI/GetVCProducts/Rp/List	ArrRec	Out	{ int* ProductId; str(250)* Title; } } – a list of card products
<b>GetServiceMembersList – request for list of service members</b>			
FIMI/GetServiceMembersList/Rq/Serviceld	Int(1)*	In	Service ID  Available value: 3 – NSPK FPS
FIMI/GetServiceMembersList/Rq/MemberType	Int(1)	In	Membership type: 0, NULL – domestic. 1 – international.
FIMI/GetServiceMembersList/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/GetServiceMembersList/Rq/Capabilities	Int(12)	In	Member capabilities.  If the value is not defined, the members are filtered by capabilities required of them. The capabilities are specified as a bit mask. If NULL, it can be any. <i>For details on the description of bits, see the description of the Capabilities field of the List structure of the response.</i>
FIMI/GetServiceMembersList/Rp>List	ArrRec	Out	{ str(12)* Id; str Title; str LatinTitle; str(1)* ServiceMemberType; int(12) Capabilities; str(20) BIC; str(11) BICSwift; int(3) Country; str(2) Language; str(1000) Currencies; } } – list of the service members:
Id – member ID in the service. Title – member title. LatinTitle – member title in Latin characters. ServiceMemberType – type of participating in the service (M – for the FPS members, S – FPS OPCC).			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description			
Capabilities – member capabilities, bit mask:						
<b>For NSPK FPS</b>						
Bit Number	<b>Member Capability</b>					
1	Fast payment C2C Push - sender					
2	Fast payment C2C Push - recipient					
3	C2B QR at the cashier desk - sender					
4	C2B QR at the cashier desk - recipient					
5	C2B QR sticker - sender					
6	C2B QR sticker - recipient					
7	C2B instant account - sender					
8	C2B instant account - recipient					
9	Request for transfer from own accounts Me2Me Pull - sender					
10	Request for transfer from own accounts Me2Me Pull - recipient					
11	Fast payment Me2Me Push - sender					
12	Fast payment Me2Me Push - recipient					
13	Other payments to private customer - sender					
14	Other payments to private customer - recipient					
15	Return in the C2B FPS operation - sender					
16	Return in the C2B FPS operation - recipient					
17	Fast payment C2G Push - sender					
18	Fast payment C2G Push - recipient					
19	Fast payment B2B Push - sender					
20	Fast payment B2B Push - recipient					
21	Fast cross-border payment C2C Push - sender					
22	Fast cross-border payment C2C Push - recipient					
23	Linking the account, payment from the linked account - sender					
24	Linking the account, payment from the linked account - recipient					
If NULL, all are allowed.						
BIC – member BIC.						
BICSwift – SWIFT code of the member bank.						
Country – country ISO code.						
Language – two-character ISO code of the language.						
Currencies – allowed member currencies, list of numeric ISO codes transferred via separator ','.						

### 2.3.3 Issuer Requests

#### 2.3.3.1 On Card:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description														
<b>GetCardInfo – request for card/Telebank customer-related information</b>																	
FIMI/GetCardInfo/Rq/PAN	Str(19)	In	<p>Card number or Telebank customer ID.            The card number mask can be defined in the format of SQL-operator 'like', i.e. applying the metacharacters '_' (any single character), '%' (any number of any characters). E.g.: to request cards with the rightmost characters '4567' and leftmost characters '5678', transmit '4567%5678' in the field.</p>														
FIMI/GetCardInfo/Rq/MBR	Int(3)	In	<p>Card member number            NULL – any            For the Telebank customer ID, the PAN field transfers the Telebank customer ID and MBR field – the value 0.</p>														
FIMI/GetCardInfo/Rq/CardUID	Str(64)	In	Unique ID of card/Telebank customer. It can stand for the card number or Telebank customer ID														
FIMI/GetCardInfo/Rq/PersonId	Int(16)	In	Customer ID														
FIMI/GetCardInfo/Rq/Type	Int(1)	In	Card type: 1-plastic; 2-TelebankID; 3-virtual. NULL – card of any type is requested.														
FIMI/GetCardInfo/Rq/RequiredData	Int	In	<p>Bit mask of the information requested on card.            0 or NULL – no restrictions. It applies if only one card is found by the search condition.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Bit Number</th><th>Field Names in Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>FoundPAN, FoundMBR, CardUID, Type</td></tr> <tr> <td>2</td><td>Status, ECStatus, TmpECStatus, TmpECStatusStartTime, TmpECStatusExpTime</td></tr> <tr> <td>3</td><td>Accounts</td></tr> <tr> <td>4</td><td>IssueTechnology, ContactlessStatus, CardAllowedEMVScript</td></tr> <tr> <td>5</td><td>ECUseCardSettingsAuth, ECNeedStaticAuth, ECNeedDynPwdAuth, ECNeedCAPAuth, ECNeedTokenAuth, ECUseDecoupledAuth</td></tr> <tr> <td>6</td><td>PersonFIO, PersonId, PersonVIP, PersonExtId, PersonConfidential</td></tr> </tbody> </table>	Bit Number	Field Names in Response	1	FoundPAN, FoundMBR, CardUID, Type	2	Status, ECStatus, TmpECStatus, TmpECStatusStartTime, TmpECStatusExpTime	3	Accounts	4	IssueTechnology, ContactlessStatus, CardAllowedEMVScript	5	ECUseCardSettingsAuth, ECNeedStaticAuth, ECNeedDynPwdAuth, ECNeedCAPAuth, ECNeedTokenAuth, ECUseDecoupledAuth	6	PersonFIO, PersonId, PersonVIP, PersonExtId, PersonConfidential
Bit Number	Field Names in Response																
1	FoundPAN, FoundMBR, CardUID, Type																
2	Status, ECStatus, TmpECStatus, TmpECStatusStartTime, TmpECStatusExpTime																
3	Accounts																
4	IssueTechnology, ContactlessStatus, CardAllowedEMVScript																
5	ECUseCardSettingsAuth, ECNeedStaticAuth, ECNeedDynPwdAuth, ECNeedCAPAuth, ECNeedTokenAuth, ECUseDecoupledAuth																
6	PersonFIO, PersonId, PersonVIP, PersonExtId, PersonConfidential																

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description	
			7	InstName, Branch, ExpDate, NameOnCard, PlasticType, CardProfiles, RefId
			8	AlternativeMessaging, MessagingProfile
			9	UserFields
			10	BackPayments, PersonalPayments
			11	TB Vendors, TB Transfers, TB Retailers, TB Cards, TB ExtraAuthLevel, TB NumericLogin, TB TextLogin, TB LastLogonTime, TB LastCustLocation, TB SecretWord
			12	CardLinks
			30	Other (all fields not described above)
FIMI/GetCardInfo/Rq/Status	Int[...]	In	Array of card statuses to be returned in the response (see a list of available values in the field InitSession/Rp/CardStatList.Id or in section 3.2)	
FIMI/GetCardInfo/Rq/ExpirationDate	Time	In	Card expiration date. The card will be searched by year and month, whereas the date, hour and seconds will be ignored.	
FIMI/GetCardInfo/Rq/NameOnCard	Str(250)	In	<p>Name on the card. The card name mask can be defined in the format of SQL-operator 'like', i.e., applying the metacharacters '_' (any single character), '%' (any number of any characters).</p>	
FIMI/GetCardInfo/Rp/FoundCards	ArrRec	Out	<p>{ str(19)* PAN; int(3)* MBR; int(16) PersonID; str(300) PersonFIO; str(64) CardUID; str(240) Branch; time ExpirationDate; str(90) BranchId; } []} – list of the found cards, not more than 100; The field is defined if in the request, in the PAN field there was defined a mask on which more than one card is found; If defined, all the other fields in the response beginning with prefix FIMI/GetCardInfo/Rp will be empty.</p>	
FIMI/GetCardInfo/Rp/InstName	Str(4)	Out	Name of the issuing financial institution	
FIMI/GetCardInfo/Rp/FoundPAN	Str(19)	Out	Number of the found card	
FIMI/GetCardInfo/Rp/FoundMBR	Int(3)	Out	MBR of the found card	

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetCardInfo/Rp/CardUID	Str(64)	Out	Unique card identifier. It can stand for the card number in the subsequent requests when working with the card.
FIMI/GetCardInfo/Rp/Branch	Str(240)	Out	Card branch name
FIMI/GetCardInfo/Rp/BranchId	Str(90)	Out	Card branch identifier
FIMI/GetCardInfo/Rp/Type	Int	Out	Card type: 1-plastic; 2-TelebankID; 3-virtual
FIMI/GetCardInfo/Rp/Acct2CardAttachType	Int	Out	Account-card link types: 0 – Main (all the card accounts belong to the same cardholder). 1 – Additional (all the card accounts belong to the customers different from the cardholder). 2 – Composite (a card has got accounts belonging to one customer and accounts of other customers). 3 – no account is linked to the card
FIMI/GetCardInfo/Rp>Status	Int	Out	Card status (for the available values, see field InitSession/Rp/CardStatList.Id or p. 3.2)
FIMI/GetCardInfo/Rp/PrevStatus	Int	Out	Status of the previous card (for the available values, see field InitSession/Rp/CardStatList.Id or p. 3.2)
FIMI/GetCardInfo/Rp/ECStatus	Int	Out	Card status in E-Commerce: -1 – not defined 0 – not active 1 – ready for enrollment 2 – active 3 – pause
FIMI/GetCardInfo/Rp/TmpECStatus	Int	Out	Temporary card status in E-Commerce: -1 – not used 0 – not active 2 – active 3 – pause
FIMI/GetCardInfo/Rp/TmpECStatusStartTime	Time	Out	Start time of the temporary card status in E-Commerce.  It is returned if the temporary E-Commerce status is used (TmpECStatus is not -1).
FIMI/GetCardInfo/Rp/TmpECStatusExpTime	Time	Out	End time of the temporary card status in E-Commerce.  It is returned if the temporary E-Commerce status is used (TmpECStatus is not -1).
FIMI/GetCardInfo/Rp/NameOnCard	Str(250)	Out	Name on card in Latin characters  For plastic cards, it is the customer name that was embossed on card and written to Track1
FIMI/GetCardInfo/Rp/ExpDate	Time	Out	Card expiration date

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCardInfo/Rp/PrevExpDate	Time	Out	Previous expiration date of the reissued card
FIMI/GetCardInfo/Rp/ReissueVariant	Int	Out	Reissued card variants: NULL – unknown 1 – old card 2 – new card The attribute is managed by the <i>SetReissuedCardVariant</i> message. The value can be changed if PrevExpDate!=NULL
FIMI/GetCardInfo/Rp/PersonId	Int(16)	Out	Card owner ID
FIMI/GetCardInfo/Rp/PersonExtId	Str(50)	Out	Customer ID in the external system
FIMI/GetCardInfo/Rp/PersonFIO	Str(300)	Out	Full name of the cardholder
FIMI/GetCardInfo/Rp/PersonConfidential	ArrRec	Out	{ str(150)* What; str(300) Value; int(1)* IsAllowedCST; int(1)* IsAllowedADS; int(1)* IsAllowedTB; } [] – customer confidential data in the form of “question-answer”: What – question, Value – answer. IsAllowedCST – question is allowed in the Customer Support Terminal application. Available values: 0 and 1; IsAllowedADS – question is allowed in the 3-D Secure ADS (Activation During Shopping) service. Available values: 0 and 1. IsAllowedTB – question is allowed in Telebank. Available values: 0 and 1.
FIMI/GetCardInfo/Rp/PersonVIP	Int(1)	Out	"VIP" customer indicator
FIMI/GetCardInfo/Rp/PVV	Int(12)	Out	PIN Verification Value
FIMI/GetCardInfo/Rp/PINVerifyType	Int(1)	Out	Actual PIN verification type. Available values: 0 – by the prefix settings 1 - Visa PVV 2 - IBM PIN offset
FIMI/GetCardInfo/Rp/IPVV	Int(12)	Out	Internet PIN Verification Value
FIMI/GetCardInfo/Rp/LastTranId	Int(12)	Out	Last transaction ID
FIMI/GetCardInfo/Rp/LastATMUsed	Time	Out	Time of the card last use at ATM
FIMI/GetCardInfo/Rp/LastPOSUsed	Time	Out	Time of the card last use at POS terminal
FIMI/GetCardInfo/Rp/LastRefreshTime	Time	Out	Time when the card was updated for the last time by the <i>Refresh</i> procedure

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCardInfo/Rp/LastChangeStatusTime	Time	Out	Time when the card status was changed for the last time in <b>TWO</b>
FIMI/GetCardInfo/Rp/LastChangePrevStatusTime	Time	Out	Time when the status of the previous card was changed for the last time in <b>TWO</b>
FIMI/GetCardInfo/Rp/LastTranTime	Time	Out	Time of the last transaction
FIMI/GetCardInfo/Rp/LastPVVChangeTime	Time	Out	Time of PVV last change
FIMI/GetCardInfo/Rp/LastIPVVChangeTime	Time	Out	Time of the Internet PVV last change
FIMI/GetCardInfo/Rp/LastPasswordChangeTime	Time	Out	Time when the alphanumeric password was last changed
FIMI/GetCardInfo/Rp/Accounts	ArrRec	Out	<pre>{     str(30)* AcctNo;     int(1)* Status;     str(250) Descr;     str(32) AccountUID;     num* LedgerBalance;     num* AvailBalance;     int* Currency;     int* Type;     int* AccountStatus;     str(40) AggregateId;     int AggregatePriority; }</pre> <p>} [] – array of structures, List of the card-linked accounts:</p>
AcctNo – account number			
Status – status of the card-linked account (account status in link):			
0 – Inactive account			
1 – Open			
2 – Deposit only			
3 – Open primary account			
4 – Deposit only primary account			
5 – Information only			
9 – Closed			
Descr – description of the card-linked account			
LedgerBalance – account ledger balance			
AvailBalance – account available balance			
Currency – account currency			
Type – account type (see the description in the <i>GetAcctInfo</i> operation)			
AccountStatus – account status (see the description in the <i>GetAcctInfo</i> operation)			
AggregateId – accounts aggregate ID			
AggregatePriority – priority of the account in the aggregate			
FIMI/GetCardInfo/Rp/MaskBalances	Int(1)	Out	=1 — account balance must be masked
FIMI/GetCardInfo/Rp/BackPayments	ArrRec	Out	<pre>{     str(30)* Id;     str(100) Descr; }</pre> <p>} [] – array of structures,</p>

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
			List of payments defined in Back-Office: Id – payment ID in Back-Office; Descr – payment description to be displayed for the customer. The field is outdated. It is replaced by the PersonalPayments field.
FIMI/GetCardInfo/Rp/ PersonalPayments	ArrRec	Out	{ str(30)* Vendor; custAcct(4000)* CustAcct; str(100) Descr; str(20) Name; } [] – array of structures, List of personal payments on card:
Vendor – vendor (payment recipient) account. CustAcct – customer personal information for sending payment to vendor. Descr – customer-defined payment description to be displayed to the customer during payment. Name – payment name to be used in SMS commands.			
FIMI/GetCardInfo/Rp/IB_Registered	Int(1)	Out	Flag 'Registered Card for Buying Information': = NULL,0 – all the other fields with the prefix IB_=NULL; = 1 – other fields with the IB_ prefix can be defined
FIMI/GetCardInfo/Rp/IB_IsPIN	Int(1)	Out	Flag 'Information Buying PIN Present'
FIMI/GetCardInfo/Rp/ IB_BadPINTries	Int	Out	Number of bad PIN tries when buying information
FIMI/GetCardInfo/Rp/IB_LastTranId	Int(12)	Out	ID of the information buying last transaction
FIMI/GetCardInfo/Rp/ IB_LastBuyBalance	Time	Out	Time of the balance buying last transaction
FIMI/GetCardInfo/Rp/ IB_LastBuyMiniStmt	Time	Out	Time of the ministatement buying last transaction
FIMI/GetCardInfo/Rp/ IB_LastResetTime	Time	Out	Time when the bad PIN tries counter for buying information was last reset
FIMI/GetCardInfo/Rp/CNSDisabled	Int(1)	Out	Flag 'Card Notification Disabled'; It is used to temporary disable notification
FIMI/GetCardInfo/Rp/ AlternativeMessaging	ArrRec	Out	{ Str(20)* Channel; Str(500)* Address; Str(20) Scheme; Str(20) Name; Str(150) Title; Int(1)* Disabled; Int(1) UseForDynAuth; Int(1) IsDefault;

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
			<p>Int(1)* Broadcast;          Int(1)* DynAuthBlocked;          Time ErrorFirstTime;          Int ErrorCount;</p> <p>} [] – list of the channel abonents and card SMS command delivery addresses, where:</p> <p>Channel, Address, Scheme – channel, address, scheme for the commands delivery (list of available values for the channel and scheme is provided in the fields <i>InitSession/Rp/CNSChannelList.Name</i> and <i>InitSession/Rp/CNSSchemeList.Name</i>).</p> <p>Name – name of the messaging abonent to be used in the commands.</p> <p>Title – any description.</p> <p>Disabled – ‘Abonent Disabled’ attribute.</p> <p>UseForDynAuth – use for dynamic authentication.</p> <p>IsDefault– ‘Default Abonent for Dynamic Authentication’ attribute. It can be enabled only for one abonent of the Telebank customer.</p> <p>Broadcast – allow the broadcast messages.</p> <p>DynAuthBlocked – “Abonent is Blocked for Dynamic Authentication” attribute.</p> <p>ErrorFirstTime – first time when the current dynamic authentication password was entered incorrectly; it is filled if UseForDynAuth=1 and ErrorCount&gt;0.</p> <p>ErrorCount – number of dynamic authentication password bad tries; it is filled if UseForDynAuth=1.</p>
FIMI/GetCardInfo/Rp/RiskLevel	Int(3)	Out	Card risk level; available values – from 0 up to 999 or NULL
FIMI/GetCardInfo/Rp/RiskControlDisabled	Int(1)	Out	<p>Flag ‘Risk Control Disabled’</p> <p>If it =1, it indicates that one transaction will be authorized without the control of risks. After that the flag will be automatically disabled</p>
FIMI/GetCardInfo/Rp/EMVOptionsCheckDisabled	Int(1)	Out	<p>‘Check EMV Options’ attribute</p> <p>If it =1, it indicates that one transaction will be authorized without the EMV options check. After that the flag will be automatically disabled.</p>
FIMI/GetCardInfo/Rp/CardProfiles	ArrRec	Out	<p>{</p> <p>int* Id;          str(150) Title;</p> <p>} [] – array of structures,          List of the card profiles;</p> <p>in the GetCardLimits request, specify the card profile ID for obtaining current values of the limits counter</p>
FIMI/GetCardInfo/Rp/MessagingProfile	ArrRec	Out	<p>{</p> <p>str(19)* PAN;          int(3)* MBR;          str(20)* Channel;          str(500)* Address;          int* ItemType;</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<p>int* ItemCode;          int MinAmount;          int(1) Disabled;</p> <p>} [] – array of structures,          List of the card messaging profiles.</p>
Channel, Address, Scheme – channel, address, scheme used for sending/receiving commands (for the list of available values for the channel and scheme, see fields <i>InitSession/Rp/CNSChannelList.Name</i> and <i>InitSession/Rp/CNSSchemeList.Name</i> ).  ItemType – scheme item type (for the list of available values, see structure <i>GetSchemeItemTypes/Rp/SchemeItemTypes</i> , p.2.2.3.6).  ItemCode – scheme item code (for the list of available values, see structure <i>GetSchemeItemTypes/Rq/SchemeItemTypes/ItemCode</i> , p.2.2.3.6).  MinAmount – minimum amount requiring the notification.  Disabled – flag ‘Disabled’.			
FIMI/GetCardInfo/Rp/UserFields	ArrRec	ArrRec	<p>{</p> <p>str(100)* Name;          str(4000) TextValue;          strXML XMLValue;</p> <p>} [] – array of structures,</p> <p>List of the card user fields; it is filled if only one card is found;</p> <p>Name, TextValue, XMLValue – name, textual or XML values of the user fields.</p>
FIMI/GetCardInfo/Rp/AuthCustPAN	Str(19)	Out	Telebank customer authenticated by a card
FIMI/GetCardInfo/Rp/AuthCustMBR	Int(3)	Out	MBR of the Telebank customer being authenticated
FIMI/GetCardInfo/Rp/ParentPAN	Str(19)	Out	Parent card from which the current card inherits the limits and limit counters that are active at the card activation in the Refresh
FIMI/GetCardInfo/Rp/ParentMBR	Int(3)	Out	Parent card MBR
FIMI/GetCardInfo/Rp/ParentCardUID	Str(64)	Out	Unique identifier of the parent card. It can be specified instead of the parent card number in the requests generated when working with the card.
FIMI/GetCardInfo/Rp/IssueTechnology	Int(2)	Out	Card issue technology. Available values:  0 – unknown; 1 – mag stripe; 11 – EMV.
FIMI/GetCardInfo/Rp/ECUseCardSettingsAuth	Int(1)	Out	Indicates that the card level settings are used for authentication in E-Commerce (3-D Secure).  For TelebankID (see GetCardInfo/Rp/Type), it is NULL.

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetCardInfo/Rp/ ECNeedStaticAuth	Int(1)	Out	Static authentication is required for E-Commerce (3-D Secure). Available values: 0 – Not required; 1 – Required. For TelebankID (see GetCardInfo/Rp/Type), it is NULL.
FIMI/GetCardInfo/Rp/ ECNeedDynPwdAuth	Int(1)	Out	Authentication by the dynamic password (SMS/E-mail) is required for E-Commerce (3-D Secure). Available values: 0 – Not required; 1 – Required. For TelebankID (see GetCardInfo/Rp/Type), it is NULL.
FIMI/GetCardInfo/Rp/ ECNeedCAPAuth	Int(1)	Out	CAP authentication is required for E-Commerce (3-D Secure). Available values: 0 – Not required; 1 – CAP required; 2 – CAP+Challenge required. For TelebankID (see GetCardInfo/Rp/Type), it is NULL.
FIMI/GetCardInfo/Rp/ ECNeedTokenAuth	Int(1)	Out	Authentication by the dynamic password (using the hardware token) is required for E-Commerce (3-D Secure). Available values: 0 – Not required; 1 – Required. For TelebankID (see GetCardInfo/Rp/Type), it is NULL.
FIMI/GetCardInfo/Rp/ ECUseDecoupledAuth	Int(1)	In	Attribute indicating that the card supports decoupled authentication. Available values: 0 – not supported. 1 – supported. For TelebankID (see GetCardInfo/Rp/Type), its value is NULL.
FIMI/GetCardInfo/Rp/CardLinks	ArrRec	Out	{ str(19)* LinkedPAN; int(3)* LinkedMBR; str(64) LinkedCardUID; int(1)* CardType; int(1)* Type; } [] – list of linked cards:
LinkedPAN, LinkedMBR – number of the linked card. CardType – type of the card in the link (1 - main, 2 – additional). Type – link type (1 – co-branding, 2 – dual card).			
FIMI/GetCardInfo/Rp/ ContactlessStatus	Int(1)	Out	Status of the contactless card interface: 0 – disabled 1 – enabled
FIMI/GetCardInfo/Rp/RefId	Str(48)	Out	Unique card ID assigned by the payment system

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<u>Telebank customer attributes:</u>			
FIMI/GetCardInfo/Rp/TBVendors	ArrRec	Out	<pre>{     str(30)* Vendor;     custAcct(4000)* CustVendorAcct;     num Amount;     int Currency;     str(750) Description; } [] – array of structures, List of templates for payment of the Telebank customer to vendors:</pre>
Vendor – payee. CustVendorAcct – personal account. Amount – amount. Currency – currency. Description – payment description.			
FIMI/GetCardInfo/Rp/TBTransfers	ArrRec	Out	<pre>{     int(12) SeqNo;     str(30) FromAcct;     str(30) ToAcct;     str(19) ToPAN;     int(3) ToMBR;     num Amount;     int Currency;     str(750) Description;     custAcct(4000) AdditionalAttributes } [] – array of structures, List of transfers templates described by Telebank customer:</pre>
SeqNo – template identifier. FromAcct – source account. ToAcct – destination account. ToPAN – destination card. ToMBR – MBR of destination card. Amount – amount. Currency – currency. Description – transfer description. AdditionalAttributes – additional attributes.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																																
FIMI/GetCardInfo/Rp/TBExtraAuthLevel	Int	Out	<p>Extra authentication type, NULL- no</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit Number</th> <th>Authentication Type</th> </tr> </thead> <tbody> <tr> <td colspan="2"><u>SMS/E-Mail Authentication</u></td> </tr> <tr> <td>1</td> <td>Transactions with account or card access</td> </tr> <tr> <td>2</td> <td>Logon</td> </tr> <tr> <td>3</td> <td>External Transfers</td> </tr> <tr> <td>4</td> <td>Financial Transactions</td> </tr> <tr> <td colspan="2"><u>CAP Authentication:</u></td> </tr> <tr> <td>9</td> <td>Transactions with account or card access</td> </tr> <tr> <td>10</td> <td>Logon</td> </tr> <tr> <td>11</td> <td>External Transfers</td> </tr> <tr> <td>12</td> <td>Financial Transactions</td> </tr> <tr> <td colspan="2"><u>CAP Signature:</u></td> </tr> <tr> <td>17</td> <td>Transactions with account or card access</td> </tr> <tr> <td>18</td> <td>Logon</td> </tr> <tr> <td>19</td> <td>External Transfers</td> </tr> <tr> <td>20</td> <td>Financial Transactions</td> </tr> </tbody> </table>	Bit Number	Authentication Type	<u>SMS/E-Mail Authentication</u>		1	Transactions with account or card access	2	Logon	3	External Transfers	4	Financial Transactions	<u>CAP Authentication:</u>		9	Transactions with account or card access	10	Logon	11	External Transfers	12	Financial Transactions	<u>CAP Signature:</u>		17	Transactions with account or card access	18	Logon	19	External Transfers	20	Financial Transactions
Bit Number	Authentication Type																																		
<u>SMS/E-Mail Authentication</u>																																			
1	Transactions with account or card access																																		
2	Logon																																		
3	External Transfers																																		
4	Financial Transactions																																		
<u>CAP Authentication:</u>																																			
9	Transactions with account or card access																																		
10	Logon																																		
11	External Transfers																																		
12	Financial Transactions																																		
<u>CAP Signature:</u>																																			
17	Transactions with account or card access																																		
18	Logon																																		
19	External Transfers																																		
20	Financial Transactions																																		
FIMI/GetCardInfo/Rp/TBRetailers	ArrRec	Out	{ int* RetailerId; str(250) Description; } } [] – array of structures, List of retailers linked to the Telebank customer; RetailerId – retailer ID Description – link description																																
FIMI/GetCardInfo/Rp/TBTextLogin	Str(50)	Out	Text login																																
FIMI/GetCardInfo/Rp/TBNumericLogin	Str(20)	Out	Numeric login																																
FIMI/GetCardInfo/Rp/TBLastLoginTime	Time	Out	Last logon time																																
FIMI/GetCardInfo/Rp/TBLastCustLocation	Str(100)	Out	IP address of the customer last logon																																
FIMI/GetCardInfo/Rp/TBSecretWord	Str(60)	Out	Secret word																																
FIMI/GetCardInfo/Rp/UseUdCVV2	Int(1)*	Out	Attribute of using user-defined CVV2 for a card																																
FIMI/GetCardInfo/Rp/PasswordFlag	Int(1)*	Out	Flag 'Card Alphanumeric Password'																																

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCardInfo/Rp/PasswordVersion	Int(1)	Out	<p>Version of generating the current alphanumeric password of the card.  The field is returned if the password is set for the card (PasswordFlag = 1).  The field is used for the <i>ChangeAlphaNumericPassword</i> operation when generating the old alphanumeric password.</p> <p>Available values:  1 – 1(SHA-1)  2 – 2(SHA-256)</p>
FIMI/GetCardInfo/Rp/RequiredPasswordVersion	Int(1)*	Out	<p>Allowed version of generating the current alphanumeric password of the card.  The field is used for the <i>ChangeAlphaNumericPassword</i> operation when generating the new alphanumeric password.</p> <p>For the available values, see the field <i>FIMI/GetCardInfo/Rp/PasswordVersion</i></p>
FIMI/GetCardInfo/Rp/TBCards	ArrRec	Out	<pre>{     str(19) PAN,     int(3) MBR,     str(64) CardUID,     str(250) Description }</pre> <p>[] – array of cards linked to the Telebank customer;</p> <p>Description – link description.</p>
<u>Card EMV attributes:</u>			
FIMI/GetCardInfo/Rp/LastATC	Int	Out	Application Transaction Counter value of the last EMV-online transaction
FIMI/GetCardInfo/Rp/EMVAppCurrency	Int	Out	EMV application currency
FIMI/GetCardInfo/Rp/OfflineLimit	Num	Out	Offline limit in the EMVAppCurrency currency
FIMI/GetCardInfo/Rp/OfflinePending	Num	Out	Amount used offline (in EMVAppCurrency currency)
FIMI/GetCardInfo/Rp/OSC_Algo	Int	Out	NULL – No offline spending control; 0 – Do not hold offline limit; 1 – Hold offline limit; 2 – Hold offline limit but use it for EMV-online
Description of the algorithm considering offline amounts when authorizing EMV transactions (the OSC_Algo field): <b>NULL</b> – no Offline Spending Control (in this case Offline Limit and the funds spent offline are not considered when calculating the available balance during authorization). <b>0</b> – do not hold Offline Limit (in this case, the amounts already spent Offline (i.e. the OfflinePending field) are considered by calculating the available balance during authorization). <b>1</b> – hold Offline Limit and do not use it Online (amounts already spent Offline /the OfflinePending field/ and also the Offline limit /the OfflineLimit field/ \affect the available balance).			

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>																
<b>2 – hold Offline Limit but use it for EMV online (amounts already spent Offline/the Offline Pending field/ and also the Offline limit for the not online and not EMV transactions affect the available balance).</b>																			
FIMI/GetCardInfo/Rp/PlasticType	Str(32)	Out	<p>Code of the card plastic type</p> <p><i>The list of available values is transferred in the PlasticTypesList structure of the InitSession operation.</i></p>																
FIMI/GetCardInfo/Rp/MaskPVV	Int	Out	<p>Attribute indicating that PVV\IPVV must be masked.</p> <p>0 – do not mask 1 – mask</p>																
FIMI/GetCardInfo/Rp/CardAllowedEMVScript	Int	Out	<p>Allowed types of EMV scripts for a card:</p> <table border="1"> <thead> <tr> <th>Bit Number</th><th>EMV Script Type</th></tr> </thead> <tbody> <tr> <td>1</td><td>Application block</td></tr> <tr> <td>2</td><td>Card block</td></tr> <tr> <td>3</td><td>Pin unblock</td></tr> <tr> <td>4</td><td>Put data</td></tr> <tr> <td>5</td><td>Update record</td></tr> <tr> <td>6</td><td>Application unblock</td></tr> <tr> <td>7</td><td>PIN Change</td></tr> </tbody> </table>	Bit Number	EMV Script Type	1	Application block	2	Card block	3	Pin unblock	4	Put data	5	Update record	6	Application unblock	7	PIN Change
Bit Number	EMV Script Type																		
1	Application block																		
2	Card block																		
3	Pin unblock																		
4	Put data																		
5	Update record																		
6	Application unblock																		
7	PIN Change																		
<b>SetCardStatus – setting card/Telebank customer status</b>																			
FIMI/SetCardStatus/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.																
FIMI/SetCardStatus/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.																
FIMI/SetCardStatus/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.																
FIMI/SetCardStatus/Rq/PersonId	Int(16)	In	Customer ID																
FIMI/SetCardStatus/Rq/ParentPAN	Str(19)	In	Parent card number																
FIMI/SetCardStatus/Rq/ParentMBR	Int(3)	In	Parent card MBR																
FIMI/SetCardStatus/Rq/ParentCardUID	Str(64)	In	Unique identifier of the parent card. It can be specified instead of the parent card number.																
FIMI/SetCardStatus/Rq>Status	Int*	In	New card status (for the available values, see field InitSesion/Rp/CardStatList.Id or p. 3.2)																
FIMI/SetCardStatus/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the card is enabled =0 – do not notify a customer even if notification on the card is enabled																
FIMI/SetCardStatus/Rq/Reissued	Int(1)	In	Attribute indicating that the card was reissued: = 0, NULL – common change of the status = 1 – reissuing of the card, it is specified when closing (changing the status to Closed) of the parent card																

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetCardStatus/Rq/ChangeReason	Str(1000)	In	Card status setting reason
If in <b>TWO</b> , the "Copy Card Attributes" setting is enabled for the FIMI terminal and the <i>ParentPAN</i> and <i>ParentMBR</i> (or <i>ParentCardUID</i> ) parameters are defined, when changing the card status from <i>Closed</i> to <i>Open</i> , <i>VIP</i> or <i>Open Domestic</i> , the attributes of the parent card are copied.			
<b>SetCardPVV – setting card/Telebank customer PVV</b>			
FIMI/SetCardPVV/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/SetCardPVV/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCardPVV/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardPVV/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetCardPVV/Rq/NewPVV	Int(12)	In	New PVV – available values fall into the range from 0 up to 999999999999 or NULL In case the algorithm of generation and Internet PIN = VISA PVV check is employed, the value posted in this field can contain 4 characters only If NULL, the card PVV is set to null value
FIMI/SetCardPVV/Rq/NewPVKI	Int(1)	In	New value of PIN Verification Key Index for the VISA PVV PIN verification algorithm. Available values: 0-6 – certain number of a key pair to be used for the PIN verification. -1 – default key pair set for a card prefix will be used. NULL – keep unchanged.
FIMI/SetCardPVV/Rq/NewIPVV	Int(12)	In	New Internet PVV – available values fall into the range from 0 up to 999999999999 or NULL In case the algorithm of generation and Internet PIN = VISA PVV check is employed, the value transmitted in this field can contain 4 characters only If NULL, the card IPVV is set to null value
FIMI/SetCardPVV/Rq/ ChangeReason	Str(1000)	In	Card PVV setting reason

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetCardPVV/Rq/AuthScheme	Int	In	<p>Authorization scheme.</p> <p>The field is required for the correct processing of the "First Allowed Transaction" authorization scheme setting.</p> <p>The field is optional. If it is not defined, the FIMI driver determines the authorization scheme by the card prefix.</p>
FIMI/SetCardPVV/Rq/SetMode	Int(1)	In	<p>Setting mode:</p> <p>1 – PVV, PVKI (the value of the NewIPVV field is ignored, card IPVV is not changed);</p> <p>2 – InternetPVV (the value of the NewPVV, NewPVKI field is ignored, card PVV, PVKI is not changed);</p> <p>3 – PVV/InternetPVV.</p> <p>If NULL, 3.</p>
FIMI/SetCardPVV/Rq/NeedNotify	Int(1)	In	<p>=1 – inform a customer if the notification on card is enabled.</p> <p>=0 or NULL – do not notify a customer even if the notification on card is enabled.</p>

***SetDynamicPVV\_PINOffset – setting dynamic PVV / PIN Offset for card***

FIMI/SetDynamicPVV_PINOffset/ Rq/PAN	Str(19)	In	Card number
FIMI/SetDynamicPVV_PINOffset/ Rq/MBR	Int(3)	In	Card member number
FIMI/SetDynamicPVV_PINOffset/ Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetDynamicPVV_PINOffset/ Rq/PINBlock	Hex(16)	In	<p>PIN block under TPK.</p> <p>If PIN block = NULL, the dynamic PVV/PIN Offset is reset.</p>
FIMI/SetDynamicPVV_PINOffset/ Rq/KeyId	Int	In	<p>ID of working key</p> <p>It must be specified if PIN block is defined</p>
FIMI/SetDynamicPVV_PINOffset/ Rq/PVKI	Int(1)	In	<p>PIN Verification Key Index for VISA PVV PIN verification algorithm.</p> <p>Available values:</p> <p>1-6 – number of the key pair that will be used to verify PIN;</p> <p>If NULL=0, the key pair specified by default for the card prefix is used</p>
FIMI/SetDynamicPVV_PINOffset/ Rq/ExpDate	Time	In	<p>Expiration date of dynamic PVV/PIN Offset in <b>TWO</b> time zone.</p> <p>It must be specified if the PIN block is defined.</p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetDynamicPVV_PINOffset/SingleOperation	Int(1)	In	Type of the validity period of the dynamic PVV/PIN Offset (it is used together with the ExpDate field): 1 – per a single operation till the specified ExpDate. 0 – per an unlimited number of operations till the specified ExpDate. If NULL, it is 1.
FIMI/SetDynamicPVV_PINOffset/Rq/ChangeReason	Str(1000)	In	Reason why the dynamic PVV/PIN Offset was set.
<b>SetCardUdCVV2 – setting user-defined CVV2 (UdCVV2) for card</b>			
FIMI/SetCardUdCVV2/Rq/PAN	Str(19)	In	Number of card and card owner.
FIMI/SetCardUdCVV2/Rq/MBR	Int(3)	In	If MBR is not transferred, it is 0.
FIMI/SetCardUdCVV2/Rq/CardUID	Str(64)	In	Unique ID of the card. It can stand for the card number.
FIMI/SetCardUdCVV2/Rq/UdCVV2	Str(3)	In	Value of user-defined CVV2. If NULL, user-defined CVV2 of a card must be reset.
FIMI/SetUdCardCVV2/Rq/ChangeReason	Str(1000)	In	Reason for setting/resetting user-defined CVV2.
<b>SetCardExpiration – setting card/Telebank customer expiration date in database</b>			
FIMI/SetCardExpiration/Rq/PAN	Str(19)	In	Number of card and card owner or Telebank customer ID.
FIMI/SetCardExpiration/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCardExpiration/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardExpiration/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/SetCardExpiration/Rq/NewExpDate	Time*	In	Card new expiration date in TWO time zone
FIMI/SetCardPVV/Rq/ ChangeReason	Str(1000)	In	Card expiration date setting reason
<b>SetCardProfile – setting card/Telebank customer profile</b>			
FIMI/SetCardProfile/Rq/PAN	Str(19)	In	Number of card and card owner or Telebank customer ID.
FIMI/SetCardProfile/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetCardProfile/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardProfile/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/SetCardProfile/Rq/NewProfile	Int	In	New value of the card profile <i>List of available values is transmitted in the response to the InitSession message in the CardProfileList field</i>
FIMI/SetCardProfile/Rq/NewBackProfile	Str(50)	In	ID of the card profile new value in Back-Office The field is optional. Its value is used if the SetCardProfile/Rq/NewProfile field is not defined. When searching for profile by Back-Office ID, the card institution is considered. If the system contains several profiles with ID specified in NewBackProfile, the profile belonging to the card institution is used.
FIMI/SetCardProfile/Rq/ChangeReason	Str(1000)	In	Card profile setting reason
<b>SetCard2AcctDescr – setting description of account/Telebank customer linked to card</b>			
FIMI/SetCard2AcctDescr/Rq/PAN	Str(19)	In	Number of card and card owner or Telebank customer ID.
FIMI/SetCard2AcctDescr/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCard2AcctDescr/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCard2AcctDescr/Rq/Account	Str(30)	In	Card-linked account
FIMI/SetCard2AcctDescr/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/SetCard2AcctDescr/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/SetCard2AcctDescr/Rq/Description	Str(250)	In	New description of the card-linked account. This description can display on the ATM screen.
FIMI/SetCard2AcctDescr/Rq/ChangeReason	Str(1000)	In	Account description setting reason
<b>ResetBadPINtries – bad PIN tries counter reset</b>			
FIMI/ResetBadPINtries/Rq/PAN	Str(19)	In	Card number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ResetBadPINtries/Rq/MBR	Int(3)	In	Card member number
FIMI/ResetBadPINtries/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/ResetBadPINtries/Rq/PersonId	Int(16)	In	Customer ID
FIMI/ResetBadPINtries/Rq/IsInet	Int(1)	In	1 – reset bad Internet PIN tries counter. NULL,0 – reset bad PIN tries counter.
FIMI/ResetBadPINtries/Rq/ChangeReason	Str(1000)	In	Bad PIN tries counter reset reason
<b>ResetCardEMVATC – reset of ATC counter for EMV cards</b>			
FIMI/ResetCardEMVATC/Rq/PAN	Str(19)	In	Card number
FIMI/ResetCardEMVATC/Rq/MBR	Int(3)	In	Cardholder number
FIMI/ResetCardEMVATC/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/ResetCardEMVATC/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/ResetCardEMVATC/Rq/ChangeReason	Str(1000)	In	EMV card counter reset reason
<b>ResetCardLimitsCounter – reset of card/Telebank customer limit counters</b>			
FIMI/ResetCardLimitsCounter/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/ResetCardLimitsCounter/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ResetCardLimitsCounter/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/ResetCardLimitsCounter/Rq/Limits	Int[]*	In	IDs of limits for which the counters are reset.
FIMI/ResetCardLimitsCounter/Rq/ChangeReason	Str(1000)	In	Reason of limits counters reset
<b>CopyCardLimitCounters – card/Telebank customer limit counters copying</b>			
FIMI/CopyCardLimitCounters/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/CopyCardLimitCounters/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CopyCardLimitCounters/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/CopyCardLimitCounters/Rq/ParentPAN	Str(19)	In	Number of the parent card from which the limit counters are copied
FIMI/CopyCardLimitCounters/Rq/ParentMBR	Int(3)	In	Cardholder number from which the limit counters are copied
FIMI/CopyCardLimitCounters/Rq/ParentCardUID	Str(64)	In	Unique identifier of the parent card. It can be specified instead of the card number.
FIMI/CopyCardLimitCounters/Rq/Limits	Int[]*	In	IDs of limits for which the counters are copied.
FIMI/CopyCardLimitCounters/Rq/ChangeReason	Str(1000)	In	Reason of limits counters copying
If the destination card already has the limit counter that is copied from the parent card, the counter value will be updated with the value of the parent card limit counter.			
<b>GetCardLimits – request for list of card/Telebank customer max and current limit values</b>			
FIMI/GetCardLimits/Rq/PAN	Str(19)	In	Number of the card and card owner number or Telebank customer ID.
FIMI/GetCardLimits/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/GetCardLimits/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/GetCardLimits/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/GetCardLimits/Rq/GetCurrentCounters	Int(1)	In	Return only counters assigned to the card. if NULL then = 0, if it = 1, current limits will be returned in the response, in the fields <i>Limits[i].Max</i> and <i>Limits[i].Current</i> ; at that, the following fields should be defined in the request – <i>AuthScheme</i> and <i>CardProfile</i>
FIMI/GetCardLimits/Rq/AuthScheme	Int	In	Authorization scheme ID; the field should be defined, if <i>GetCurrenCounters</i> =1; For the list of the available values, see the field <i>InitSession/Rp/AuthSchemes</i>
FIMI/GetCardLimits/Rq/CardProfile	Int	In	Card profile ID; the field should be defined if <i>GetCurrenCounters</i> =1; For the list of the available values, see the field <i>GetCardInfo/Rp/CardProfiles</i>
FIMI/GetCardLimits/Rp/LimCurrency	Int*	Out	Currency of the card financial limits

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCardLimits/Rp/LastResetTime	Time	Out	Time of the counter (the limit current values) last reset
FIMI/GetCardLimits/Rp/Limits	ArrRec	Out	<pre>{     int* LimitId;     num Max;     num* Current;     int(2) PeriodType;     int Period;     num TmpMax;     time TmpMaxExpiration;     int(1)* MaxPresentedOnCard;     num TempMaxRange;     time TmpMaxStartTime;     int(1) TmpMaxPeriodType;     time DateTimeForReset }</pre> <p>} [] – array of structures, List of the card max and current limits:</p>
LimitId – limit ID in TWO (see p. 3.3). Max – max limit (-1 – as for a card profile, 0 – restricted, NULL – unlimited, >0 – defined). TmpMax – limit temporary max (0 – restricted, NULL – not defined, >0 – defined). TmpMaxExpiration – expiration date of the temporary max (NULL – until the next reset of the limit counter). Current – current limit. PeriodType, Period – limit validity period (see below, in SetCardLimits). MaxPresentedOnCard – flag ‘Max Limit Presented on Card’. TmpMaxRange – the value limiting the TmpMax (0 – restricted, NULL – not defined, >0 – defined) parameter range. TmpMaxStartTime – date and time when the limit temporary maximum becomes active. TmpMaxPeriodType – type of the limit temporary maximum validity period (NULL – for a period; 1 – per a single operation). DateTimeForReset – calendar date and time of the limit current value reset.			
FIMI/GetCardLimits/Rp/LimitsRanges	ArrRec	Out	<pre>{     int* LimitId;     time* StartDate;     time EndDate;     num Max; }</pre> <p>} [] – array of structures, list of the limit ranges returned in the <i>Limits</i> structure:</p>
LimitId – limit ID in TWO (see p. 3.3). StartDate – start date of the range. EndDate – end date of the range (NULL – range is unlimited). Max – maximum value of the limit in the range (NULL – as maximum limit value).			
1. If GetCurrentCounters = 0, the Limits array of records returns the following: 1.1. maximum values of the limits assigned to the card and their counters; 1.2. values of counters kept for the card and belonging to the limits not assigned to the card (scheme level limits); 1.3. values of counters (=0) that are not kept for the card but the limits are created for them in TWO. 2. If GetCurrentCounters = 0 and AuthScheme and CardProfile are set, the Limits array of records returns the following: 2.1. maximum values of the limits assigned to the card and their counters; 2.2. values of counters kept for the card and belonging to the limits not assigned to the card (scheme level limits), the values are recalculated in compliance with the reset settings for the specified authorization schemes and card profile; 2.3. values of counters (=0) that are not kept for the card but the limits are created for them in TWO. 3. If GetCurrentCounters = 1 and AuthScheme and CardProfile are set, the Limits array of records returns the following: 3.1. maximum values of the limits assigned to the card and their counters; 3.2. values of counters kept for the card and belonging to the limits not assigned to the card (scheme level limits), the values are recalculated in compliance with the reset settings for the specified authorization schemes and card profile.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetCardProfileLimits – request for list of limits on card profile</b>			
FIMI/GetCardProfileLimits/Rq/AuthScheme	Int*	In	Authorization scheme ID For the list of available values, see the field <i>InitSession/Rp/AuthSchemes</i>
FIMI/GetCardProfileLimits/Rq/CardProfile	Int*	In	Card profile ID For the list of available values, see the field <i>GetCardInfo/Rp/CardProfiles</i>
FIMI/GetCardProfileLimits/Rp/LimCurrency	Int*	Out	Currency of financial limits card profile
FIMI/GetCardProfileLimits/Rp/Limits	ArrRec	Out	<pre>{     int* LimitId;     num.MaxValue;     int(2) PeriodType;     int Period;     time DateTimeForReset; } [] – array of structures,</pre> List of max limits and periods of validity (for the description of the periods of validity, see the <i>SetCardProfileLimits</i> operation)
<b>SetTmpCardLimits – setting card/Telebank customer temporary max limits</b>			
FIMI/SetTmpCardLimits/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/SetTmpCardLimits/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetTmpCardLimits/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetTmpCardLimits/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/SetTmpCardLimits/Rq/Limits	ArrRec*	In	<pre>{     int LimitId;     num TmpMax;     time TmpMaxExpiration;     time TmpMaxStartTime;     int TmpMaxPeriodType; } [] – array of structures,</pre> List of the temporary max limits:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
LimitId – limit ID in TWO (see p.3.3). TmpMax – temporary max limit (it is cleared when the limit expires or certain time is reached). TmpMaxExpiration – temporary max validity period (NULL-until the limit counter next reset). TmpMaxStartTime – date and time when the limit temporary maximum becomes active. TmpMaxPeriodType – type of the limit temporary maximum validity period (NULL – for a period; 1 – per a single operation).			
FIMI/SetTmpCardLimits/Rq/ NeedNotify			
FIMI/SetTmpCardLimits/Rq/ ChangeReason			
<b>SetCardLimits – modification of card/Telebank customer max limits and their validity periods</b>			
FIMI/SetCardLimits/Rq/PAN	Str(19)	In	Card and card owner number or Telebank customer ID.
FIMI/SetCardLimits/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCardLimits/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardLimits/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/SetCardLimits/Rq/ RemoveMissingLimits	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing limits. 2 – remove received limits. If NULL, it is 0.
FIMI/SetCardLimits/Rq/LimCurrency	Int	In	Limit currency code. The limit currency is defined only if all the limit values of the card are defined. The parameter can be defined if RemoveMissingLimits=1.
FIMI/SetCardLimits/Rq/Limits	ArrRec	In	{ int LimitId; num Max; int(2) PeriodType; int Period; Time DateTimeForReset; } [] – array of structures, List of the card max limits:
LimitId – limit ID in TWO (see p.3.3). Max – card max limit (-1 – to set the limit as for a card profile).			

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
PeriodType – period type:			
			<ul style="list-style-type: none"> <li>- NULL – period type and limit reset time within the period are unknown.</li> <li>- 0 – Absolute (Daily) – absolute period; it starts from the beginning of the day (1440 – it is reset every day at 0:00, 2880+420 – at 7:00 every second day).</li> <li>- 1 – Weekly – calendar weekly period; it starts from the beginning of the week (2880+420 – it is reset every Wednesday at 7:00).</li> <li>- 2 – Monthly – calendar monthly period; it starts from the beginning of the month (2880+420 – it is reset on the third day of every month at 7:00).</li> <li>- 3 – Quarterly – calendar quarterly period; it starts from the beginning of the quarter (2880+420 – it is reset on the third day of every quarter at 7:00).</li> <li>- 4 – Infinite – limit counter will never be reset; the period is optional.</li> <li>- 5 – Single operation – one-time limit (limit for a single operation); limit counter is always reset before the limit check; the period is optional.</li> <li>- 6 – Reset in Refresh – limit counter is reset with Refresh; the period is optional.</li> <li>- 7 – Yearly – the calendar year limit, the period starts from the beginning of the year.</li> <li>- 8 – Sliding – sliding limit, the real value of the limit is calculated during the transaction authorization and considers all transactions executed over the time period starting from the moment in the past (it is the difference between the current time value and specified period value) till the current moment.</li> <li>- 11 - Calendar – reset on the calendar date and time (for a particular type, the DateTimeForReset field is used instead of the Period field).</li> </ul> <p>(Period=0 - the counter is reset on the 1st of January at 00:00;      Period=2880+420 – the counter is reset each year on the 3<sup>rd</sup> of January at 7:00.</p> <p>Period – limit reset time (min):      Period/1440 - number of day (starts from 0); Mod(LimitPeriod,1440) – reset time within a day (min)</p> <p>Restrictions:      PeriodType=0 → 1440&lt;=Period&lt;=1440000    PeriodType=1 → 0&lt;=Period&lt;1440*7      PeriodType=2 → 0&lt;=Period&lt;1440*31      PeriodType=3 → 0&lt;=Period&lt;1440*90      PeriodType=7 → 0&lt;=Period&lt;1440*360      PeriodType=8 → 1&lt;=Period&lt;1440*30</p>
FIMI/SetCardLimits/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled =0 – do not notify a customer even if notification on the cards is enabled
FIMI/SetCardLimits/Rq/ChangeReason	Str(1000)	In	Reason for changing the card max limits and their validity periods
<b>SetCardLimitsRanges – setting ranges of card/Telebank customer limit</b>			
FIMI/SetCardLimitsRanges/Rq/PAN	Str(19)	In	Card and card owner number or Telebank customer ID.
FIMI/SetCardLimitsRanges/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCardLimitsRanges/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardLimitsRanges/Rq/LimitId	Int*	In	Limit ID in TWO (see p. 3.3)
FIMI/SetCardLimitsRanges/Rq/RemoveMissing	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing ranges. 2 – remove received ranges. If NULL, it is 0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetCardLimitsRanges/Rq/LimitsRanges	ArrRec	In	<p>{ time StartDate; time EndDate; num Max; } [] – array of structures, list of limit ranges:</p> <p>StartDate – start time of the range (NULL – starting from the operation execution). EndDate – end time of the range (NULL – range is unlimited). Max – maximum value of the limit in the range (NULL – as maximum limit value).</p>
FIMI/SetCardLimitsRanges/Rq/ChangeReason	Str(1000)	In	Reason for changing the card max limits and their validity periods
FIMI/SetCardLimitsRanges/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled =0 – do not notify a customer even if notification on the cards is enabled
<b>SetCardProfileLimits – setting limits by card profile</b>			
FIMI/SetCardProfileLimits/Rq/AuthScheme	Int*	In	<p>Authorization scheme ID For the list of available values, see the field <i>InitSession/Rp/AuthSchemes</i></p>
FIMI/SetCardProfileLimits/Rq/CardProfile	Int*	In	<p>Card profile ID For the list of available values, see the field <i>GetCardInfo/Rp/CardProfiles</i></p>
FIMI/SetCardProfileLimits/Rq/Limits	ArrRec*	In	<p>{ int* LimitId; num.MaxValue; int(2) PeriodType; int Period; time DateTimeForReset } [] – array of structures, List of the card maximum limits and periods of validity:</p>
<p>LimitId – limit ID in TWO (see p. 3.3). .MaxValue - limit max value. PeriodType, Period, DateTimeForReset – limit validity period (see the <i>SetCardLimits</i> operation above).</p>			
FIMI/SetCardProfileLimits/Rq/ChangeReason	Str(1000)	In	Reason for setting the card max limits and validity periods of the card profile
<b>CNSCardConfig – defining card/Telebank customer messaging parameters</b>			
FIMI/CNSCardConfig/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/CNSCardConfig/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CNSCardConfig/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/CNSCardConfig/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/CNSCardConfig/Rq/DisableNotification	Int(1)	In	Flag 'Disable Notification' if NULL then =0
FIMI/CNSCardConfig/Rq/AlternativeMessaging	ArrRec	In	{ Str(20)* Channel; Str(500)* Address; Str(20) Scheme; Str(20) Name; Str(150) Title; Int(1) Disabled; Str TestMessage; Int(1) UseForDynAuth; Int(1) IsDefault; Int(1) Priority; Int(1) Broadcast; Str(20) PrevChannel; Str(500) PrevAddress; } } [] – list of channels and addresses for SMS commands delivery and reception:
For the structure field values, see the description of the field <i>FIMI/GetCardInfo/Rp/AlternativeMessaging</i> .			
If the AlternativeMessaging array is transmitted, the fields Channel, Address are mandatory (must not be NULL)			
If TestMessage is not NULL, the defined test message is sent to the customer via the defined channel and to the respective address			
Priority – priority of test message sending (0 – low; 1 – below average; 2 – average; 3 – above average; 4 – high). By default - 4 (high).			
The PrevChannel and PrevAddress fields are filled only if the existing notification element is filled with the previous data.			
If Disabled=NULL, =0.			
If UseForDynAuth=NULL, =0.			
If Broadcast =NULL, =1			
FIMI/CNSCardConfig/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled =0 – do not notify a customer even if notification on the cards is enabled
FIMI/CNSCardConfig/Rq/ChangeReason	Str(1000)	In	Reason for setting the card messaging parameters
<b><i>The CNSCardConfig operation is outdated, it is recommended to use the AddCMSAbonent, ChangeCMSAbonent, RemoveCMSAbonent operations.</i></b>			
The FIMI/CNSCardConfig/Rq/AlternativeMessaging/Disabled field affects the customer messaging only. If Disabled = 1, the notifications will not be generated but the commands incoming via sms from customers will be received and processed successfully. If FIMI/CNSCardConfig/Rq/AlternativeMessaging/Scheme = NULL, the incoming customer commands will not be processed.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>ConfigureCMSProfile – setting messaging profile parameters for card/Telebank customer</b>			
FIMI/ConfigureCMSProfile/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/ConfigureCMSProfile/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ConfigureCMSProfile/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/ConfigureCMSProfile/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/ConfigureCMSProfile/Rq/MessagingProfile	ArrRec	In	<pre>{     str(20)* Channel;     str(500)* Address;     int* ItemType;     int* ItemCode;     num MinAmount;     int(1) Disabled; } [] – array of structures;</pre> <p>values of the elements responsible for the SMS notifications delivery:</p>
The existing elements are replaced by the elements from the array. If the array is empty, the existing ones are removed.			
Description of the fields is similar to the description of the field FIMI/GetCardInfo/MessagingProfile in the GetCardInfo operation. The ItemCode field is extended by the following values: -1 – all. 0 – any.			
FIMI/ConfigureCMSProfile/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled =0 – do not notify a customer even if notification on the cards is enabled
FIMI/ConfigureCMSProfile/Rq/ChangeReason	Str(1000)	In	Messaging profile setting reason
<b>CardRiskControl – setting card/Telebank customer risk control parameters</b>			
FIMI/CardRiskControl/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/CardRiskControl/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/CardRiskControl/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CardRiskControl/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/CardRiskControl/Rq/RiskLevel	Int(3)	In	Card risk level; available values – from 0 to 999 or NULL
FIMI/CardRiskControl/Rq/Disable	Int(1)	In	Flag 'Disable Risk Control' if NULL then =0  if it =1, one transaction will be authorized without risk control. Then the flag is automatically disabled
<b>ResetCard2AcctLink – re-creating link of card/Telebank customer and account</b>			
FIMI/ResetCard2AcctLink/Rq/PAN	Str(19)	In	Number of the card and card owner number or Telebank customer ID.
FIMI/ResetCard2AcctLink/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ResetCard2AcctLink/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/ResetCard2AcctLink/Rq/Account	Str(30)	In	Number of the account to be linked to the card instead of the existing links
FIMI/ResetCard2AcctLink/Rq/AccountUID	Str(32)	In	Unique card identifier within a session. It can stand for the card number.
FIMI/ResetCard2AcctLink/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/ResetCard2AcctLink/Rq/CardPersonID	Int(16)	In	Card member ID
FIMI/ResetCard2AcctLink/Rq/Description	Str(250)	In	Description of account linked to card. The description can be displayed on the ATM screen.
FIMI/ResetCard2AcctLink/Rq/ChangeReason	Str(1000)	In	Card-to-account link reset reason
<b>UpdateCard2AcctLink – creating link of card/Telebank customer and account</b>			
FIMI/UpdateCard2AcctLink/Rq/PAN	Str(19)	In	Number of card and card owner or Telebank customer ID.
FIMI/UpdateCard2AcctLink/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/UpdateCard2AcctLink/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateCard2AcctLink/Rq/Account	Str(30)	In	Number of account to be linked to the card instead of the existing links
FIMI/UpdateCard2AcctLink/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/ResetCard2AcctLink/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/UpdateCard2AcctLink/Rq/AcctStatus	Int*	In	Status of the account linked to the card:  0 – Inactive account 1 – Open 2 – Deposit only 3 – Open primary account 4 – Deposit only primary account 5 – Information only 9 – Closed
FIMI/UpdateCard2AcctLink/Rq/Description	Str(250)	In	Description of account linked to card. The description can be displayed on the ATM screen.
FIMI/UpdateCard2AcctLink/Rq/ChangeReason	Str(1000)	In	Card-to-account link creation reason
<b>DeleteCard2AcctLink – deleting link of card/Telebank customer and account</b>			
FIMI/DeleteCard2AcctLink/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/DeleteCard2AcctLink/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/DeleteCard2AcctLink/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/DeleteCard2AcctLink/Rq/Account	Str(30)	In	Account number
FIMI/DeleteCard2AcctLink/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/DeleteCard2AcctLink/Rq/ChangeReason	Str(1000)	In	Reason for deleting the card-to-account link
FIMI/DeleteCard2AcctLink/Rq/IgnoreChecking	Int(1)	In	If =0 or NULL, the check of whether at least one link to the account remains for the card is performed. If =1, the check is ignored
<b>CreateCard – recording information on existing card</b>			
FIMI/CreateCard/Rq/PAN	Str(19)*	In	Card number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateCard/Rq/MBR	Int(3)*	In	Card member number
FIMI/CreateCard/Rq/InstName	Str(4)*	In	Institution the card belongs to
FIMI/CreateCard/Rq/BranchId	Str(90)	In	Branch ID
FIMI/CreateCard/Rq/Expiration	Time*	In	The card expiration date
FIMI/CreateCard/Rq/PersonId	Int(16)	In	Customer ID
FIMI/CreateCard/Rq>Status	Int(2)*	In	Card status (see a list of available values in the field InitSession/Rp/CardStatList.Id or Section 3.2)
FIMI/CreateCard/Rq/NameOnCard	Str(250)	In	Name on the card
FIMI/CreateCard/Rq/PVKI	Int(1)	In	PIN Verification Key Index for the PIN - VISA PVV check algorithm. Available values: 1-6 – number of key pair that will be used to check PIN; NULL=0 – PIN will be checked using the default key pair (specified for the card prefix).
FIMI/CreateCard/Rq/PINVerifyType	Int(1)	In	Actual PIN verification type. Available values: NULL, 0 – by the prefix settings 1 - Visa PVV 2 - IBM PIN offset
FIMI/CreateCard/Rq/ParentPAN	Str(19)	In	Parent card number. It is specified when reissuing the card.
FIMI/CreateCard/Rq/ParentMBR	Int(3)	In	Parent card member number. It is specified when reissuing the card.
FIMI/CreateCard/Rq/ParentCardUID	Str(64)	In	Unique ID of the parent card. It can stand for the parent card number.
FIMI/CreateCard/Rq/ObjectId	Str(64)	In	Unique card ID.
FIMI/CreateCard/Rq/ExternalObjectId	Str(64)	In	External unique ID of the card
<b>CreateVCard – creating virtual Internet card</b>			
FIMI/CreateVCard/Rq/Account	Str(30)	In	Number of account to link a virtual card to  The account must be defined when creating a virtual card; for the prepaid card, the account can be left undefined.
FIMI/CreateVCard/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/CreateVCard/Rq/Expiration	Time	In	Expiration date — time when the virtual card will expire (in TWO time zone)

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateVCard/Rq/LimitPeriodType	Int(1)	In	Limit period type (see the description below)
FIMI/CreateVCard/Rq/LimitPeriod	Int	In	Period of limit validity (see the description below)
FIMI/CreateVCard/Rq/LimitCurrency	Int	In	Currency for the card financial limit (if not defined, FIMI-terminal currency is taken by default)
FIMI/CreateVCard/Rq/LimitAmount	Num	In	Purchase max amount for the specified period
FIMI/CreateVCard/Rq/LimitCount	Int	In	Max number of purchases for the specified period
FIMI/CreateVCard/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if the card notification is enabled  =0 – do not notify a customer even if the card notification is enabled
FIMI/CreateVCard/Rq/CardProfile	Int	In	Card profile identifier.  If NULL and CardProductId=-4, it is identified automatically using the found Domestic prefix.
FIMI/CreateVCard/Rq/NameOnCard	Str(250)	In	Customer name on card in Latin characters
FIMI/CreateVCard/Rq/BranchId	Str(90)	In	Branch ID
FIMI/CreateVCard/Rq/CardProductId	Int	In	Card product identifier  Available values: -1 – Select prefixes only with card product type = virtual card. -2 - Select prefixes only with card product type = prepaid card. -3 – Select prefixes with any type of car product.  -4 – Select the Domestic prefix by the card number transferred in the PAN field. >0 – ID of card product (it can be obtained using the GetVCProducts operation).  When using values -1, -2 and -3, only one prefix of the selected type must be described in the authorization scheme. If several prefixes satisfying the specified conditions are found, the transaction will be declined.  When using value -4, prefixes on the transferred PAN values must be described in the authorization scheme. Otherwise, the request will be declined.  If =NULL, CardProductId =-1
FIMI/CreateVCard/Rq/ExternalObjectId	Str(64)	In	Unique external ID of a card

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateVCard/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/CreateVCard/Rq/MBR	Int(3)	In	They are transferred if it is required to create a card with the specified number. If the card PAN is not transferred, they are generated automatically according to the settings of the card product in use. If MBR is not transferred, value 0 is set.
LimitPeriodType – period type: <ul style="list-style-type: none"> <li>- 0 – Absolute (Daily) – absolute period; it is started from the beginning of the day (1440 – it is reset every day at 0:00, 2880+420 – at 7:00 every second day);</li> <li>- 1 – Weekly – calendar weekly period; it is started from the beginning of the week (2880+420 – it is reset every Wednesday at 7:00 );</li> <li>- 2 – Monthly – calendar monthly period; it is started from the beginning of the month (2880+420 – it is reset on the third day of every month at 7:00);</li> <li>- 3 – Quarterly – calendar quarterly period; it is started from the beginning of a quarter (2880+420 – it is reset on the third day of every quarter at 7:00);</li> <li>- 4 – Infinite – the counter will never be reset; the period is optional;</li> <li>- 5 – Single operation – the limit counter is always reset before the limit check; the period is optional;</li> </ul>			
Period – limit reset time within the calendar period(min): Period/1440 – number of the day (starts from 0); Mod(Period,1440) – reset time during the day (min.) Restrictions: PeriodType=0 → 1440<=Period<=1440000    PeriodType=1 → 0<=Period<1440*7 PeriodType=2 → 0<=Period<1440*31    PeriodType=3 → 0<=Period<1440*90			
FIMI/CreateVCard/Rp/PAN	Str(19)*	Out	Virtual card PAN
FIMI/CreateVCard/Rp/MBR	Int(3)	Out	Virtual card MBR = 0
FIMI/CreateVCard/Rp/CardUID	Str(64)	Out	Unique card identifier. It can stand for the card number.
FIMI/CreateVCard/Rp/Expiration	Time*	Out	Virtual card expiration date in <b>TWO</b> time zone (can be adjusted by <b>TWO</b> authorizer)
FIMI/CreateVCard/Rp/CVV2	Int(3)	Out	Virtual card CVV2.  It is returned if the CVV2 calculation is allowed in the settings of the <b>TWO</b> authorization scheme for virtual cards.
FIMI/CreateVCard/Rp/StrCVV2	Str(3)	Out	Virtual card CVV2 in the string format.  It is returned if the CVV2 calculation is allowed in the settings of the <b>TWO</b> authorization scheme for virtual cards.
FIMI/CreateVCard/Rp/LimitCurrency	Int*	Out	Financial limit currency
FIMI/CreateVCard/Rp/LimitAmount	Num*	Out	Purchase max amount for a specified period (can be adjusted by <b>TWO</b> authorizer)
FIMI/CreateVCard/Rp/LimitCount	Int*	Out	Max number of purchases for the specified period (can be adjusted by <b>TWO</b> authorizer)
FIMI/CreateVCard/Rp/Account	Str(30)	Out	Number of account linked to the prepaid card.
FIMI/CreateVCard/Rp/AccountUID	Str(32)	Out	Account unique identifier within the session. It can be specified instead of the account number in the next account requests.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateVCard/Rp/InstName	Str(4)	Out	Name of the financial institution that issued the card. It is filled together with the field <i>FIMI/CreateVCard/Rp/Account</i> .
The card product settings in Back-Office define whether the plastic card will be issued or the card will remain virtual.  Either CardProfile field or LimitAmount and LimitCount fields must be filled in the <i>CreateVCard</i> operation. If they are absent in the operation at the same time, the transaction will be declined at the authorization stage.			
<b>SetEMVCardParams – setting EMV attributes</b>			
FIMI/SetEMVCardParams/Rq/PAN	Str(19)	In	Card number
FIMI/SetEMVCardParams/Rq/MBR	Int(3)	In	Card member number
FIMI/SetEMVCardParams/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetEMVCardParams/Rq/PersonId	Int(16)	In	Customer ID  Reserved, not used for processing
FIMI/SetEMVCardParams/Rq/OfflineLimit	Num	In	New value of the offline spending limit in the EMVAppCurrency currency.  If not NULL, new value of the offline spending will be set.
FIMI/SetEMVCardParams/Rq/Disable	Int(1)	In	'Prohibit EMV Options Check' attribute.  If NULL, then = 0.  If it =1, it indicates that one transaction will be authorized without the EMV options check. After that the flag will be automatically disabled.
FIMI/SetEMVCardParams/Rq/ChangeReason	Str(1000)	In	Reason for EMV attributes setting
<b>GetCardStatement – request for card statement</b>			
FIMI/GetCardStatement/Rq/PAN	Str(19)	In	Card number
FIMI/GetCardStatement/Rq/MBR	Int(3)	In	Card member number
FIMI/GetCardStatement/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetCardStatement/Rq/PersonId	Int(16)	In	Customer ID  Reserved, not used for processing
FIMI/GetCardStatement/Rq/FromTime	Time	In	Starting from, NULL – from the beginning
FIMI/GetCardStatement/Rq/ToTime	Time	In	Ending with, NULL – up to the current moment.
FIMI/GetCardStatement/Rq/Count	Int(4)	In	Number of the last requested operations. If not defined – all the operations are returned but not more than 1000.
FIMI/GetCardStatement/Rq/Language	Int	In	ID of language in which the <i>AnotherTitle</i> field must be returned (the <i>Statement</i> structure). The language code is taken from the <b>TWO</b> dictionary.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			If the field is not defined, the standard description of field returns. This parameter is used only if the operations history is requested via Core Banking authorizer.
FIMI/GetCardStatement/Rp/ Statement	ArrRec	Out	<pre>{     int(12) FrontId;     str(12) BackId;     int(1)* Origin;     int(1)* Type;     int* OperCode;     str(300) AnotherTitle;     str(4000) Description;     num* Amount;     str(3) Currency;     time* OperDate;     time TranTime;     num OrigAmount;     str(3) OrigCurrency;     str(19) PAN;     int(3) MBR;     int TermClass;     str(16) TermName;     str(150) TermRetailerName;     int TermSIC;     str(750) TermLocation;     str(30) BackAcct;     num BackAmount;     num Remain;     str(8) ApprovalCode;     int(3)* CurrencyISOCode;     int(3) OrigCurrencyISOCode;     int(3)* CurrencyISO Code;     int(3) OrigCurrencyISOCode;     int* SeqNo;     time OrigTime;     str(750) MessToRecipient;     str(2) BAI;     num TIPAmount;     num ProtectedAmount;     str(16) RetailerServicePhone;     str(255) RetailerURL;     str(64) CardUID;     int(3) TermCountry;     str(90) TermCity;     num OnlineIssuerFee; } [] – array of structures, history of the accounts linked to the card, sorted by the time in the descending order (starting from the last operations). The number of the operations should not exceed Min (Rq/Count, 1000). For the description of the array fields, see the GetAcctStatement operation, the Statement structure.</pre>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCardStatement/Rp/ AncillaryService	ArrRec	Out	<pre>{     int*      SeqNo;     str(2)*   Code     num(12, 3)* Amount } [] – array of structures, list of additional service payments on the operation:</pre> <p>SeqNo – sequence number of the operation, it matches the SeqNo field of the Statement structure.</p> <p>Code – code of the service payment.</p> <p>Amount – amount of the service payment in the operation currency (the OrigCurrency field of the Statement structure).</p>
FIMI/GetCardStatement/Rp/ AliasService	ArrRec	Out	<pre>{     int*      SeqNo;     str       SenderMessage;     str(32)   ExternalTranId;     str(12)   RecipientMemberId;     str       RecipientFIO;     str(500)  RecipientAlias;     str(12)   SenderMemberId;     str       SenderFIO;     str(500)  SenderAlias;     int(3)    SenderCountry;     int(3)    RecipientCountry;     num      DebtorAmount;     int(3)    DebtorCurrency;     num      CreditorAmount;     int(3)    CreditorCurrency;     num      Fee;     str(10)   PaymentDetailsCategoryCode } [] – array of structures, additional data for the operation of transfer by alias:</pre> <p>SeqNo – sequence number of the operation, matches the SeqNo field of the Statement structure.</p> <p>SenderId – external transaction ID.</p> <p>RecipientMemberId – ID of the transfer service participant on the recipient side.</p> <p>RecipientFIO – full name of the recipient.</p> <p>RecipientAlias – recipient alias.</p> <p>SenderId – ID of the transfer service participant on the sender side.</p> <p>SenderFIO – full name of the sender.</p> <p>SenderAlias – sender alias.</p> <p>SenderCountry – sender country (returned for international transfers).</p> <p>RecipientCountry – recipient country (returned for international transfers).</p> <p>DebtorAmount – amount to be debited from the sender (returned for international transfers).</p> <p>DebtorCurrency – ISO code of the currency for the amount to be debited (returned for international transfers).</p> <p>CreditorAmount – amount to be credited to the recipient (returned for international transfers).</p> <p>CreditorCurrency – ISO code of the currency of the amount to be credited (returned for international transfers).</p> <p>Fee – operation fee in the debit currency.</p> <p>PaymentDetailsCategoryCode – transfer category code.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCardStatement/Rp/ QRService	ArrRec	Out	<pre>{     int* SeqNo;     str(140) PaymentPurpose;     str(140) RetailerTitle;     str(140) RetailerName;     str(140) RetailerAddress;     str(12) MemberId;     int MCC;     str(4) QR.PayloadCRC;     str(32) ExternalTranId;     str(32) RefExternalTranId; } [] – array of structures, additional data on the operation executed using the QR code:</pre> <p>SeqNo – sequence number of the operation, matches the SeqNo field of the Statement structure.</p> <p>RetailerTitle – title of the merchant that issued the QR code.</p> <p>RetailerName – name of the merchant that issued the QR code.</p> <p>RetailerAddress – address of the merchant that issued the QR code.</p> <p>MemberId – member ID within the service (the list of available options is requested in the <i>GetServiceMembersList</i> operation).</p> <p>MCC – MCC code.</p> <p>QR.PayloadCRC – value of the checksum of the QR code payload.</p> <p>ExternalTranId – external identifier of a transaction.</p> <p>RefExternalTranId – external identifier of the linked transaction, it contains the external identifier of the original transaction when the operation is cancelled.</p>
FIMI/GetCardStatement/Rp/ GovernmentService	ArrRec	Out	<pre>{     int* SeqNo;     str(32) UID;     str(25) PUID;     str(140) LegalEntityName;     str(32) ExternalTranId;     str(140) PayerName;     str(12) PayerTaxNumber;     str(9) PayerTaxReasonRegCode; } [] – array of structures, additional data for the operation of transfer in favour of the government:</pre> <p>SeqNo – sequence number of the operation, it is the same as the SeqNo field of the Statement structure.</p> <p>UID – unique identifier assigned by the payer bank.</p> <p>PUID – unique payment identifier.</p> <p>LegalEntityName – corporate customer name.</p> <p>ExternalTranId – external transaction identifier.</p> <p>PayerName – real payer: corporate customer name or title.</p> <p>PayerTaxNumber – TPN of a real payer.</p> <p>PayerTaxReasonRegCode – tax registration reason code of a real payer.</p>
FIMI/GetCardStatement/Rp/ MultiAcct	ArrRec	Out	<pre>{     int* SeqNo;     str(30)* Account;     int* Currency;     num* Amount</pre>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<pre>} [] – array of structures, list of accounts with amounts participating in the authorization (if several accounts were used):</pre>
SeqNo – sequence number of the operation. Account – account. Currency – account currency ID. Amount – amount debited from the account.			
FIMI/GetCardStatement/Rp/ ArrestImpact	ArrRec	Out	<pre>{     int* SeqNo;     str(4)* InstName;     str(32)* ArrestId;     num* Amount;     str Description; }</pre> <p>} [] – array of structures, list of impacts on the arrest under the credit operation:</p>
<pre>SeqNo – sequence number of the operation. Account – account. Amount – arrested amount debited from the account in the account currency. InstName – institution name. ArrestId – arrest ID (data on the arrest can be received using the GetArrests operation). Description – description.</pre>			
<b>CreateTBCustomer – creating Telebank-customer</b>			
FIMI/CreateTBCustomer/Rq/Account	Str(30)*	In	Account number
FIMI/CreateTBCustomer/Rq>Status	Int(2)*	In	Telebank-customer account status. For the list of available values, see p. 3.2
FIMI/CreateTBCustomer/Rq/ExpDate	Time*	In	Expiration date of the Telebank-customer account
FIMI/CreateTBCustomer/Rq/PersonId	Int(16)	In	Customer ID
FIMI/CreateTBCustomer/Rq/ TextLogin	Str(50)	In	Text login
FIMI/CreateTBCustomer/Rq/ NumericLogin	Str(20)	In	Numeric login
FIMI/CreateTBCustomer/Rq/PVKI	Int(1)	In	PIN Verification Key Index for the PIN - VISA PVV check algorithm. Available values: 1-6 – number of key pair that will be used to check PIN; NULL=0 – PIN will be checked using the default key pair (specified for the Telebank-customer prefix).
FIMI/CreateTBCustomer/Rq/ CustomerId	Str(19)	In	Telebank customer ID. If the field is defined, this ID will be assigned to the Telebank customer; otherwise, the identifier will be generated automatically.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateTBCustomer/Rq/ExternalObjectId	Str(64)	In	Unique external ID of the Telebank customer
FIMI/CreateTBCustomer/Rp/CustomerId	Str(19)*	Out	Telebank customer ID
<b>AddCardVendor – linking vendor to card</b>			
FIMI/AddCardVendor/Rq/PAN	Str(19)	In	Customer card
FIMI/AddCardVendor/Rq/MBR	Int(3)	In	
FIMI/AddCardVendor/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AddCardVendor/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/AddCardVendor/Rq/VendorAcct	Str(30)*	In	Vendor account
FIMI/AddCardVendor/Rq/CustAcct	CustAcct (4000)*	In	Customer personal information
FIMI/AddCardVendor/Rq/Comment	Str(100)*	In	Payment name to be displayed at ATM or a mobile phone
FIMI/AddCardVendor/Rq/Name	Str(20)	In	Payment short name in Latin characters to be used in the CMS commands (at the mobile phone)
<b>DelCardVendor – unlinking vendor from card</b>			
FIMI/DelCardVendor/Rq/PAN	Str(19)	In	Customer card
FIMI/DelCardVendor/Rq/MBR	Int(3)	In	
FIMI/DelCardVendor/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/DelCardVendor/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/DelCardVendor/Rq/VendorAcct	Str(30)*	In	Vendor account
FIMI/DelCardVendor/Rq/CustAcct	CustAcct (4000)*	In	Customer personal information
<b>SetReissuedCardVariant – setting reissued card variants</b>			
FIMI/SetReissuedCardVariant/Rq/PAN	Str(19)	In	Customer card
FIMI/SetReissuedCardVariant/Rq/MBR	Int(3)	In	
FIMI/SetReissuedCardVariant/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetReissuedCardVariant/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetReissuedCardVariant/Rq/ Variant	Int(1)	In	Reissued card variant: NULL – unknown 1 – old card 2 – new card
<b>GetTBCustHistory – request for Telebank customer history</b>			
FIMI/GetTBCustHistory/Rq/Customer	Str(19)*	In	Telebank-customer ID
FIMI/GetTBCustHistory/Rq/MBR	Int(3)*	In	Card member number (for Telebank customer, the value is 0)
FIMI/GetTBCustHistory/Rq/FromTime	Time	In	Transaction starting date and time
FIMI/GetTBCustHistory/Rq/ToTime	Time	In	Transaction ending date and time
FIMI/GetTBCustHistory/Rp/History	ArrRec	Out	<pre>{ int(12)* TranId; int PayClass; int PayType; Time TranTime; Int RespCode; Str(8) ApprovalCode; Str(30) FromAcct; Str(30) ToAcct; Str(19) ToPAN; Int(3) ToMBR; Str(1000) CustVendorAcct; Num Amount; Int Currency; Str(4000) Description; Int(1) SecurityLevel; Int(1) OutsideTransfer; Int TranType; }</pre> <p>} [] – array of structures,</p> <p>TranId – transaction ID          PayClass – payment class (0 – to vendor, 1 – defined in Back-Office)          PayType – payment/transfer type (0 – Immediate, 1 – Future, 2 – Recurring)          TranTime – transaction time          RespCode – response code (in addition to the other codes: 1 - reversal)          ApprovalCode – approval code          FromAcct – source account          ToAcct – destination account          ToPAN – destination card number          ToMBR – destination card owner          CustVendorAcct – personal account          Amount – amount          Currency – currency          Description – payment description          SecurityLevel – extra authentication (1 – Software, 2 – HardWare)          TranId – transaction ID          PayClass – payment class (0 – to vendor, 1 – defined in Back-Office)          PayType – payment/transfer type (0 – Immediate, 1 – Future, 2 – Recurring)          TranTime – transaction time          RespCode – response code (in addition to the other codes: 1 – reversal)          ApprovalCode – approval code          FromAcct – source account          ToAcct – destination account</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
ToPAN – destination card number ToMBR – destination card owner CustVendorAcct – personal account Amount – amount Currency – currency Description – payment description SecurityLevel – extra authentication (1 – Software, 2 – HardWare) OutsideTransfer – flag “Transfer to External Card or Account” TranType – transaction type			
<b>SetExtraAuthLevel – setting Telebank customer extra authentication</b>			
FIMI/SetExtraAuthLevel/Rq/Customer	Str(19)*	In	Telebank customer ID
FIMI/SetExtraAuthLevel/Rq/MBR	Int(3)*	In	Card member number (for Telebank customer, the value is 0)
FIMI/SetExtraAuthLevel/Rq/NewLevel	Int*	In	New value of the extra authentication level of the Telebank customer. Bit mask. For the bit values, see the description of the field <i>FIMI/GetCardInfo/Rp/TBExtraAuthLevel</i> . Respective bits from the three groups are mutually exclusive. To use CAP, either the 10th or the 18th bits should be included.
<b>SetCardUserFields – setting card/Telebank customer user fields</b>			
FIMI/SetCardUserFields/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/SetCardUserFields/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCardUserFields/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardUserFields/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/SetCardUserFields/Rq/ UserFields	ArrRec	In	{ Str(100)* Name; Str(3800) TextValue; StrXML XMLValue. } [] – array of structures
The total length of the Name and TextValue fields in all the records of the <i>FIMI/SetCardUserFields/Rq/UserFields</i> array must not exceed 3900 symbols.			
If the <i>TextValue</i> and <i>XMLValue</i> fields are absent in the structure and card has the user field, this user field will not be changed. To reset the <i>TextValue</i> and <i>XMLValue</i> values, specify empty values of the <i>TextValue</i> and <i>XMLValue</i> fields.			

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetCardUserFields/Rq/ RemoveMissingFields	Int(1)	In	1 – remove missing fields 0 – by default If FIMI/InitSession/Rp/ LimitedAccessToUserFields = 1, only the user fields available for editing will be deleted
FIMI/SetCardUserFields/Rq/ NeedNotify	Int(1)	In	=1 – notify a customer if notification is enabled =0 or NULL – do not notify a customer even if a notification is enabled
<b>ChangeAlphaNumericPassword – changing card/Telebank customer alphanumeric password</b>			
FIMI/ChangeAlphaNumericPassword /Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/ChangeAlphaNumericPassword /Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ChangeAlphaNumericPassword /Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/ChangeAlphaNumericPassword /Rq/OldPassword	Str(64)	In	Old password hash. If specified, it is checked. It is generated according to the version returned by the <i>GetCardInfo</i> request in the <i>PasswordVersion</i> field.
FIMI/ChangeAlphaNumericPassword /Rq/NewPassword	Str(64)	In	New password hash. It is generated according to the version returned by the <i>GetCardInfo</i> request in the <i>RequiredPasswordVersion</i> field.
FIMI/ChangeAlphaNumericPassword /Rq/NewPasswordVersion	Int(1)	In	Version of the new password hash. Available values: 1,NULL – 1(SHA-1) 2 – 2(SHA-256)
FIMI/ChangeAlphaNumericPassword /Rq/Force	Int(1)	In	Indicates that the new password hash must be compared with the previous values. If =1, the check is performed
Password encryption algorithm can be presented by the pseudocode:			
for the 1(SHA-1) version: passwordHash = StrToPackedHex(SubStr(SHA1Hash(PAN+ToUTF8(password)),1,8))			
for the 2(SHA-256) version: passwordHash=StrToPackedHex(SHA256(PAN + UTF8(password)))			
where: PAN – card number. password – password plain text. passwordHash – password hash.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>ResetAlphaNumericPassword – reset alphanumeric password of card/Telebank customer</b>			
FIMI/ResetAlphaNumericPassword /Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/ResetAlphaNumericPassword /Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ResetAlphaNumericPassword /Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
<b>ResetBadDynAuthPwdTries – reset of bad dynamic authentication password tries counter for card/Telebank customer abonent</b>			
FIMI/ResetBadDynAuthPwdTries/Rq/Customer	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/ResetBadDynAuthPwdTries/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ResetBadDynAuthPwdTries/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/ResetBadDynAuthPwdTries/Rq/Channel	Str(20)*	In	Channel
FIMI/ResetBadDynAuthPwdTries/Rq/Address	Str(500)*	In	Address
<b>ChangeECStatus – changing Card Status in e-Commerce</b>			
FIMI/ChangeCardECStatus/Rq/PAN	Str(19)	In	Card number
FIMI/ChangeCardECStatus/Rq/MBR	Int(3)	In	Cardholder number
FIMI/ChangeCardECStatus/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/ChangeECStatus/Rq>Status	Int(1)	In	<p>Card status in E-Commerce. Available values:</p> <ul style="list-style-type: none"> <li>-1 – Not defined</li> <li>0 – Not active</li> <li>1 – Ready for enrollment</li> <li>2 – Active</li> <li>3 – Pause</li> </ul> <p>Possible transitions:</p> <ul style="list-style-type: none"> <li>Not defined → Ready for enrollment</li> <li>Not defined → Active</li> <li>Not active → Ready for enrollment</li> <li>Not active → Active</li> <li>Active → Not active</li> <li>Active → Ready for enrollment</li> <li>Active → Pause</li> </ul>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
			Pause → Not active Pause → Ready for enrollment Pause → Active Ready for enrollment → Active If <i>Not defined</i> , the transitions are not allowed. If NULL, the card status in E-Commerce is not changed.
FIMI/ChangeECStatus/Rq/Type	Int(1)	In	Card type: 1 - plastic; 3 - virtual. NULL indicates that the card of the type 1 – plastic has been requested.
FIMI/ChangeECStatus/Rq/ TmpECStatus	Int	In	Temporary card status in E-Commerce: -1 – Not used 0 – Not active 2 – Active 3 – Pause  Available transitions depend on the current card status (or status being set) in E-Commerce: - If E-Commerce status is “Active”, the temporary E-Commerce status can be “Not active”, “Not used”, “Pause”. - If E-Commerce status is “Pause”, the temporary E-Commerce status can be “Active”, “Not Used”. - If E-Commerce status is not “Pause” or not “Active”, it is prohibited to use the temporary E-Commerce status. If NULL, it is -1.
FIMI/ChangeECStatus/Rq/ TmpECStatusStartTime	Time	In	Start time of the temporary card status in E-Commerce.  It is mandatory if the temporary E-Commerce status is set (TmpECStatus is other than -1). If the value of the TmpECStatus field is -1, the field value is ignored.
FIMI/ChangeECStatus/Rq/ TmpECStatusExpTime	Time	In	End time of the temporary card status in E-Commerce.  It is mandatory if the temporary E-Commerce status is set (TmpECStatus is other than -1). If the value of the TmpECStatus field is -1, the field value is ignored.
FIMI/ChangeCardECStatus/Rq/ ChangeReason	Str(1000)	In	Change reason
<b>GetPrefixInfo – getting information by prefix</b>			
FIMI/GetPrefixInfo/Rq/PAN	Str(19)	In	Card number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetPrefixInfo/Rq/CardUID	Str(64)	In	Unique card ID. It can stand for the card number.
FIMI/GetPrefixInfo/Rq/TransactionType	Int(3)	In	Transaction type: 100=AuthRequest (authorization request, online authorization is performed); 120=AuthAdvice (authorization request, unconditional receiving and approval of the transaction advice); 200=Request (financial request, online authorization is performed); 220=Advice (financial request, unconditional receiving and approval of the transaction advice)
FIMI/GetPrefixInfo/Rq/TransactionCode	Int(4)	In	Transaction code. For the list of available values, see p. 3.8 in the POS transaction section
FIMI/GetPrefixInfo/Rq/Amount	Num	In	Transaction amount
FIMI/GetPrefixInfo/Rq/Currency	Int	In	Transaction currency
FIMI/GetPrefixInfo/Rq/FromAccountType	Int(2)	In	Source account type. Available values: 0-Default, 1-Checking, 11-Savings, 31-Credit, 91-Bonus
FIMI/GetPrefixInfo/Rq/ToAccountType	Int(2)	In	Destination account type. Available values: 0-Default, 1-Checking, 11-Savings, 31-Credit, 91-Bonus
FIMI/GetPrefixInfo/Rq/Track2	Str(39)	In	Magnetic stripe contents in the following format: “PAN=YYMM[SRV]...” or “PANDYYMM[SRV]...” where: PAN – value transferred in the PAN field; YYMM – card expiration date (two characters – year, two characters – month); SRV – 3-character value of the card service code; ... – additional information from the magnetic stripe
FIMI/GetPrefixInfo/Rq/Amount2	Num	In	Transaction real amount, cashback amount, etc.
FIMI/GetPrefixInfo/Rq/Message	Int	In	Used to transfer the additional subcode in the transaction
FIMI/GetPrefixInfo/Rq/Condition	Int(2)	In	Transaction execution condition. For the list of available values, refer to p. 3.19
FIMI/GetPrefixInfo/Rq/EntryMode	Int(3)	In	PAN entry mode. For the list of available values, refer to p. 3.20

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetPrefixInfo/Rq/ICCAdditionalData	StrUAMP	In	Additional EMV tags in the UAMP format. For details on the field format, refer to "TW International Container Fields Specification".
FIMI/GetPrefixInfo/Rp/Prefix	Str(19)*	Out	Card prefix
FIMI/GetPrefixInfo/Rp/CardProgram1	Int	Out	First card program. Interpretation of codes in the CardProfileList parameter of the CardProgramList request.
FIMI/GetPrefixInfo/Rp/CardProgram2	Int	Out	Second card program. Interpretation of codes in the CardProfileList parameter of the CardProgramList request.
FIMI/GetPrefixInfo/Rp/Technology	Int*	Out	Issue technology. Available values: 0 – unknown 1 – Mag stripe 11 – EMV
FIMI/GetPrefixInfo/Rp/CardProfile	Int*	Out	Card profile. Interpretation of codes in the CardProfileList parameter of the <i>InitSession</i> request.
FIMI/GetPrefixInfo/Rp/County	Int	Out	ISO
FIMI/GetPrefixInfo/Rp/Region	Str(50)	Out	Region
FIMI/GetPrefixInfo/Rp/Type	Int	Out	Prefix type. For the list of available values, see <i>FIMI/InitSession/Rp/PrefixTypeList</i> .
FIMI/GetPrefixInfo/Rp>Title	Str(150)	Out	Prefix title
FIMI/GetPrefixInfo/Rp/IssInstName	Str(4)	Out	Character ID (name) of the prefix issuer financial institution. For the list of available values, see <i>InitSession/Rp/AllowedFIID.Name</i> .

**AddTBTransfer – adding Telebank customer transfer template**

FIMI/AddTBTransfer/Rq/PAN	Str(19)	In	Telebank customer ID
FIMI/AddTBTransfer/Rq/MBR	Int(3)	In	Card member number
FIMI/AddTBTransfer/Rq/CardUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID.
FIMI/AddTBTransfer/Rq/FromAcct	Str(30)	In	Source account number. The source account can be present only by one of the Telebank customer accounts.
FIMI/AddTBTransfer/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can be specified instead of the account number
FIMI/AddTBTransfer/Rq/ToAcct	Str(30)	In	Destination account number. If the field is filled, the field <i>FIMI/AddTBTransfer/Rq/ToPAN</i> is left undefined
FIMI/AddTBTransfer/Rq/ToPAN	Str(19)	In	Destination card PAN. If the field is filled, the field <i>FIMI/AddTBTransfer/Rq/ToAcct</i> is left undefined

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddTBTransfer/Rq/ToMBR	Int(3)	In	Destination card member number
FIMI/AddTBTransfer/Rq/Amount	Num	In	Amount
FIMI/AddTBTransfer/Rq/Currency	Int	In	Currency code
FIMI/AddTBTransfer/Rq/Description	Str(750)	In	Transfer description
FIMI/AddTBTransfer/Rq/AdditionalAttributes	CustAcct (4000)*	In	Additional attributes of the transfer template
FIMI/AddTBTransfer/Rp/SeqNo	Int*	Out	Transfer template number
<b><i>DelTBTransfer – deleting Telebank customer transfer template</i></b>			
FIMI/DelTBTransfer/Rp/SeqNo	Int*	In	Number of transfer template being deleted
<b><i>AddTBPayment – adding Telebank customer payment template</i></b>			
FIMI/AddTBPayment/Rq/Customer	Str(19)	In	Telebank-customer ID
FIMI/AddTBPayment/Rq/MBR	Int(3)	In	Card member number (for the Telebank customer, the value is 0)
FIMI/AddTBPayment/Rq/CardUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID
FIMI/AddTBPayment/Rq/VendorAcct	Str(30)*	In	Vendor account
FIMI/AddTBPayment/Rq/CustAcct	CustAcct (4000)*	In	Payment-specific attributes
FIMI/AddTBPayment/Rq/Description	Str(750)	In	Comment
FIMI/AddTBPayment/Rq/Amount	Num	In	Payment amount
FIMI/AddTBPayment/Rq/Currency	Int	In	Payment currency
FIMI/AddTBPayment/Rq/FromAcct	Str(30)	In	Source-account
FIMI/AddTBPayment/Rq/CMSName	Str(20)	In	CMS name
<b><i>DelTBPayment – deleting Telebank customer payment template</i></b>			
FIMI/DelTBPayment/Rq/Customer	Str(19)	In	Telebank-customer ID
FIMI/DelTBPayment/Rq/MBR	Int(3)	In	Card member number (for Telebank customer, the value is 0)
FIMI/DelTBPayment/Rq/CardUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID
FIMI/DelTBPayment/Rq/VendorAcct	Str(30)	In	Vendor account
FIMI/DelTBPayment/Rq/CustAcct	CustAcct (4000)*	In	Payment-specific attributes
FIMI/DelTBPayment/Rq/SeqNo	Int	In	ID of vendor assigned to Telebank customer (if it is specified, the vendor is searched by SeqNo; otherwise by VendorAcct and CustAcct)

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b><i>SetCardPerson – changing owner of card/Telebank customer</i></b>			
FIMI/SetCardPerson/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/SetCardPerson/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN field transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCardPerson/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardPerson/Rq/NewPersonId	Int(16)*	In	New ID of the cardholder
FIMI/SetCardPerson/Rq/ChangeReason	Str(1000)	In	Change reason
<b><i>AddTBRetailer – adding link between Telebank customer and retailer</i></b>			
FIMI/AddTBRetailer/Rq/Customer	Str(19)	In	Telebank customer ID
FIMI/AddTBRetailer/Rq/MBR	Int(3)	In	Card member number (for the Telebank customer, the value is 0)
FIMI/AddTBRetailer/Rq/CardUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID.
FIMI/AddTBRetailer/Rq/RetailerId	Int*	In	Retailer ID
FIMI/AddTBRetailer/Rq/Description	Str(250)	In	Description of retailer linked to the Telebank customer
<b><i>DeleteTBRetailer – deleting link between Telebank customer and retailer</i></b>			
FIMI/DeleteTBRetailer/Rq/Customer	Str(19)	In	Telebank customer ID
FIMI/DeleteTBRetailer/Rq/MBR	Int(3)	In	Card member number (for the Telebank customer, the value is 0)
FIMI/DeleteTBRetailer/Rq/CardUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID
FIMI/DeleteTBRetailer/Rq/RetailerId	Int*	In	Retailer ID
<b><i>AttachCardToTBCustomer – linking own institution card to Telebank customer</i></b>			
FIMI/AttachCardToTBCustomer/Rq/TelebankId	Str(19)	In	Telebank customer ID
FIMI/AttachCardToTBCustomer/Rq/TelebankMBR	Int(3)	In	Telebank customer member number (for the Telebank customer, the value is 0)
FIMI/AttachCardToTBCustomer/Rq/TelebankUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AttachCardToTBCustomer/Rq/PAN	Str(19)	In	Card number
FIMI/AttachCardToTBCustomer/Rq/MBR	Int(3)	In	Card member number
FIMI/AttachCardToTBCustomer/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AttachCardToTBCustomer/Rq/Description	Str(250)	In	Link description
<b><i>DetachCardFromTBCustomer – deleting own institution card linked to Telebank customer</i></b>			
FIMI/DetachCardFromTBCustomer/Rq/TelebankId	Str(19)	In	Telebank customer ID
FIMI/DetachCardFromTBCustomer/Rq/TelebankMBR	Int(3)	In	Telebank customer member number (for the Telebank customer, the value is 0)
FIMI/DetachCardFromTBCustomer/Rq/TelebankUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID
FIMI/DetachCardFromTBCustomer/Rq/PAN	Str(19)	In	Card number
FIMI/DetachCardFromTBCustomer/Rq/MBR	Int(3)	In	Card member number
FIMI/DetachCardFromTBCustomer/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
<b><i>AttachForeignCardToTBCustomer – linking external institution card to Telebank customer</i></b>			
FIMI/AttachForeignCardToTBCustomer/Rq/TelebankId	Str(19)	In	Telebank customer ID
FIMI/AttachForeignCardToTBCustomer/Rq/TelebankMBR	Int(3)	In	Telebank customer member number (for the Telebank customer, the value is 0)
FIMI/AttachForeignCardToTBCustomer/Rq/TelebankUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID
FIMI/AttachForeignCardToTBCustomer/Rq/PAN	Str(19)	In	Card number
FIMI/AttachForeignCardToTBCustomer/Rq/MBR	Int(3)	In	Card member number
FIMI/AttachForeignCardToTBCustomer/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ AttachForeignCardToTBCustomer/ Rq/ExpirationDate	Time*	In	Card expiration date
FIMI/ AttachForeignCardToTBCustomer/ Rq/NameOnCard	Str(250)	In	Name on card in Latin letters.  For the physically issued plastic cards, it is the name of the customer embossed on the card and specified in <i>Track1</i>
FIMI/ AttachForeignCardToTBCustomer/ Rq/Description	Str(250)	In	Link description
<b>DetachForeignCardFromTBCustomer – deleting external institution card linked to Telebank customer</b>			
FIMI/ DetachForeignCardFromTBCustomer/ Rq/TelebankId	Str(19)	In	Telebank customer ID
FIMI/ DetachForeignCardFromTBCustomer/ Rq/TelebankMBR	Int(3)	In	Telebank customer member number  (for the Telebank customer, the value is 0)
FIMI/ DetachForeignCardFromTBCustomer/ Rq/TelebankUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID
FIMI/ DetachForeignCardFromTBCustomer/ Rq/PAN	Str(19)	In	Card number
FIMI/ DetachForeignCardFromTBCustomer/ Rq/MBR	Int(3)	In	Card member number
FIMI/ DetachForeignCardFromTBCustomer/ Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
<b>GetCardAutopayments – request for list of autopayments by card number</b>			
FIMI/GetCardAutopayments/Rq/PAN	Str(19)	In	Card number
FIMI/GetCardAutopayments/Rq/MBR	Int(3)	In	Card member number
FIMI/GetCardAutopayments/Rq/ CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetCardAutopayments/Rp>List	ArrRec	Out	{ Int(9)* Id; Str(30)* VendorAcct; Str(100)* ExtId; Int* Status; Str(250) Description; CustAcct(4000) CustAcct } []- array of structures

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
VendorAcct – vendor account number. ExtId – subscription external identifier. Status – subscription status: <ul style="list-style-type: none"> <li>- 0 – status is not defined, its check is required.</li> <li>- 1 – request for subscription is created.</li> <li>- 2, 5 – request for subscription is declined.</li> <li>- 10 – active.</li> <li>- 31 – request for the parameters change is created.</li> <li>- 32,35 – request for the parameters change is declined.</li> <li>- 51 – request to disable the subscription is created.</li> <li>- 52, 55 – request to disable the subscription is declined.</li> <li>- 59 – subscription deleted.</li> <li>- 60 – request for the subscription block is created.</li> <li>- 61 – blocked.</li> <li>- 62 – request for the subscription unblock is created.</li> <li>- 63 – unblocked.</li> <li>- 999 – in process.</li> </ul> Description – description. CustAcct – subscription attributes. Examples: Type – prepaid or postpaid tariff. Provider – operator providing services for such suppliers as Cyberplat. Threshold amount – when this amount is reached, the autopayment must be performed. Replenishment amount – autopayment amount.			
<b><i>AttachAutopaymentToCard – request for adding autopayment</i></b>			
FIMI/AttachAutopaymentToCard/Rq/ PAN	Str(19)	In	Card number
FIMI/AttachAutopaymentToCard /Rq/MBR	Int(3)	In	Card member number
FIMI/AttachAutopaymentToCard /Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AttachAutopaymentToCard/Rq/ VendorAcct	Str(30)*	In	Vendor account
FIMI/AttachAutopaymentToCard/Rq/ CustAcct	CustAcct (4000)	In	Subscription attributes
FIMI/AttachAutopaymentToCard /Rp/Id	Int(9)*	Out	ID of the created subscription
<b><i>DetachAutopaymentFromCard – request for deleting the autopayment</i></b>			
FIMI/DetachAutopaymentFromCard/ Rq/PAN	Str(19)	In	Card number
FIMI/DetachAutopaymentFromCard /Rq/MBR	Int(3)	In	Card member number
FIMI/DetachAutopaymentFromCard /Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ DetachAutopaymentFromCard /Rq/VendorAcct	Str(30)*	In	Vendor account
FIMI/DetachAutopaymentFromCard /Rq/Id	Int(9)*	In	Subscription ID
<b>EditCardAutopayment – request for editing the autopayment</b>			
FIMI/EditCardAutopayment/Rq/PAN	Str(19)	In	Card number
FIMI/EditCardAutopayment/Rq/MBR	Int(3)	In	Card member number
FIMI/EditCardAutopayment/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/EditCardAutopayment/Rq/VendorAcct	Str(30)*	In	Vendor account from the template
FIMI/EditCardAutopayment/Rq/CustAcct	CustAcct (4000)	In	Subscription attributes from the template
FIMI/EditCardAutopayment/Rq/Id	Int(9)*	In	Subscription ID from the template
<b>GetCardAutopaymentHistory – request for history of autopayment subscriptions</b>			
FIMI/GetCardAutopaymentHistory/Rq/PAN	Str(19)	In	Card number
FIMI/GetCardAutopaymentHistory/Rq/MBR	Int(3)	In	Card member number
FIMI/GetCardAutopaymentHistory/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetCardAutopaymentHistory/Rq/FromTime	Time	In	Starting from, NULL – from
FIMI/GetCardAutopaymentHistory/Rq/ToTime	Time	In	Till, NULL – till the current time
FIMI/GetCardAutopaymentHistory/Rp/Log	ArrRec	Out	{ Int(9)* Id; Str(30)* VendorAcct; Str(100)* ExtId; Int* Status; Str(250) Description; CustAcct(4000) CustAcct; Time ChangeTime; Str(1) OperationType; } [] – array of structures.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																
Id – template ID. VendorAcct – vendor account number. ExtId – subscription external ID. Status – autopayment subscription status (for the list of values, see <i>FIMI/GetCardAutopayments/Rp&gt;List&gt;Status</i> ). Description – description. CustAcct – subscription attributes. ChangeTime – subscription change time. OperationType – operation type: U – update. D – deleting. I – adding.																			
<b>DeleteCard – deleting card</b>																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">FIMI/DeleteCard/Rq/PAN</td> <td style="padding: 2px;">Str(19)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Card number</td> </tr> <tr> <td style="padding: 2px;">FIMI/DeleteCard/Rq/MBR</td> <td style="padding: 2px;">Int(3)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Card member number</td> </tr> <tr> <td style="padding: 2px;">FIMI/DeleteCard/Rq/CardUID</td> <td style="padding: 2px;">Str(64)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Unique card identifier. It can stand for the card number.</td> </tr> <tr> <td style="padding: 2px;">FIMI/DeleteCard/Rq/Force</td> <td style="padding: 2px;">Int(1)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px; vertical-align: top;">           =NULL or 0 – the check is performed; if transactions (except for the Create FIMI transaction) are found for the card in the Online Transaction log, the operation is declined.            =1 – card is deleted without the check of whether transactions on card are present.         </td> </tr> </table>				FIMI/DeleteCard/Rq/PAN	Str(19)	In	Card number	FIMI/DeleteCard/Rq/MBR	Int(3)	In	Card member number	FIMI/DeleteCard/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.	FIMI/DeleteCard/Rq/Force	Int(1)	In	=NULL or 0 – the check is performed; if transactions (except for the Create FIMI transaction) are found for the card in the Online Transaction log, the operation is declined. =1 – card is deleted without the check of whether transactions on card are present.
FIMI/DeleteCard/Rq/PAN	Str(19)	In	Card number																
FIMI/DeleteCard/Rq/MBR	Int(3)	In	Card member number																
FIMI/DeleteCard/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.																
FIMI/DeleteCard/Rq/Force	Int(1)	In	=NULL or 0 – the check is performed; if transactions (except for the Create FIMI transaction) are found for the card in the Online Transaction log, the operation is declined. =1 – card is deleted without the check of whether transactions on card are present.																
<b>DeleteTBCustomer – deleting Telebank customer</b>																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">FIMI/DeleteTBCustomer/Rq/CustomerId</td> <td style="padding: 2px;">Str(19)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Telebank customer ID</td> </tr> <tr> <td style="padding: 2px;">FIMI/DeleteTBCustomer/Rq/MBR</td> <td style="padding: 2px;">Int(3)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Telebank customer owner number (for the Telebank customer, the value is 0)</td> </tr> <tr> <td style="padding: 2px;">FIMI/DeleteTBCustomer/Rq/CustomerUID</td> <td style="padding: 2px;">Str(64)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Unique ID of the Telebank customer. It can stand for the Telebank customer ID</td> </tr> <tr> <td style="padding: 2px;">FIMI/DeleteTBCustomer/Rq/Force</td> <td style="padding: 2px;">Int(1)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px; vertical-align: top;">           =NULL or 0 – the check is performed; if transactions (except for the Create FIMI transaction) are found for the Telebank customer in the Online Transaction log, the operation is declined            =1 – Telebank customer is deleted without the check of whether transactions for the Telebank customer are present         </td> </tr> </table>				FIMI/DeleteTBCustomer/Rq/CustomerId	Str(19)	In	Telebank customer ID	FIMI/DeleteTBCustomer/Rq/MBR	Int(3)	In	Telebank customer owner number (for the Telebank customer, the value is 0)	FIMI/DeleteTBCustomer/Rq/CustomerUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID	FIMI/DeleteTBCustomer/Rq/Force	Int(1)	In	=NULL or 0 – the check is performed; if transactions (except for the Create FIMI transaction) are found for the Telebank customer in the Online Transaction log, the operation is declined =1 – Telebank customer is deleted without the check of whether transactions for the Telebank customer are present
FIMI/DeleteTBCustomer/Rq/CustomerId	Str(19)	In	Telebank customer ID																
FIMI/DeleteTBCustomer/Rq/MBR	Int(3)	In	Telebank customer owner number (for the Telebank customer, the value is 0)																
FIMI/DeleteTBCustomer/Rq/CustomerUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID																
FIMI/DeleteTBCustomer/Rq/Force	Int(1)	In	=NULL or 0 – the check is performed; if transactions (except for the Create FIMI transaction) are found for the Telebank customer in the Online Transaction log, the operation is declined =1 – Telebank customer is deleted without the check of whether transactions for the Telebank customer are present																

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>SetTBLogin – changing Telebank customer text/numeric login</b>			
FIMI/SetTBLogin/Rq/CustomerId	Str(19)	In	Telebank customer ID
FIMI/SetTBLogin/Rq/MBR	Int(3)	In	Telebank customer owner number (for Telebank customer, the value is 0)
FIMI/SetTBLogin/Rq/CustomerUID	Str(64)	In	Unique ID of the Telebank customer. It can stand for the Telebank customer ID
FIMI/SetTBLogin/Rq/TextLogin	Str(50)	In	Telebank customer text login. If NULL (field is not sent), it is not changed. If the value is empty, it is reset.
FIMI/SetTBLogin/Rq/NumericLogin	Str(20)	In	Telebank customer numeric login. If NULL (field is not sent), it is not changed. If the value is empty, it is reset.
<b>AddCMSAbonent – adding messaging parameters for card/Telebank customer</b>			
FIMI/AddCMSAbonent/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/AddCMSAbonent/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/AddCMSAbonent/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/AddCMSAbonent/Rq/AlternativeMessaging	ArrRec*	In	{ Str(20)* Channel; Str(500)* Address; Str(20) Scheme; Str(20) Name; Str(150) Title; Int(1) Disabled; Str TestMessage; Int(1) UseForDynAuth; Int(1) IsDefault; Int(1) Priority; Int(1) Broadcast } [] – list of channels and addresses for SMS commands delivery and reception:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
Channel, Address, Scheme – channel, address, scheme used for the commands delivery (list of available values for the channel and scheme is provided in the fields <i>InitSession/Rp/CNSChannelList.Name</i> and <i>InitSession/Rp/CNSSchemeList.Name</i> ).			
Name – name of the messaging abonent to be used in the commands.			
Title – any description.			
Disabled – “Abonent Disabled” attribute. If NULL, it=0.			
TestMessage – test message. If not NULL, the defined test message is sent to the customer via the defined channel and to the respective address.			
UseForDynAuth – use for dynamic authentication.			
IsDefault – “Default Abonent for Dynamic Authentication” attribute. It can be enabled only for one abonent for a certain card.			
Priority – priority of test message sending (0 – low; 1 – below average; 2 – average; 3 – above average; 4 – high). By default, 4 – high.			
Broadcast – allow the broadcast messages.			
FIMI/AddCMSAbonent/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled =0 – do not notify a customer even if notification on the cards is enabled
FIMI/AddCMSAbonent/Rq/ChangeReason	Str(1000)	In	Reason for setting the messaging parameters
The FIMI/AddCMSAbonent/Rq/AlternativeMessaging/Disabled field affects sending of the customer notifications only. If FIMI/ChangeCMSAbonent/Rq/AlternativeMessaging/Disabled = 1, the notifications will not be generated but the commands incoming via sms from customers will be accepted and processed successfully. If FIMI/ChangeCMSAbonent/Rq/AlternativeMessaging//Scheme = NULL, the incoming customer commands will not be processed.			
<b>RemoveCMSAbonent – deleting messaging parameters for card/Telebank customer</b>			
FIMI/RemoveCMSAbonent/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/RemoveCMSAbonent/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/RemoveCMSAbonent/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/RemoveCMSAbonent/Rq/AlternativeMessaging	ArrRec*	In	{ Str(20)* Channel; Str(500)* Address } [] – list of channels and addresses for SMS commands delivery and reception  Channel, Address – channel, address for the commands delivery (list of available values for the channel is provided in the field <i>InitSession/Rp/CNSChannelList.Name</i> )

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/RemoveCMSAbonent/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled =0 – do not notify a customer even if notification on the cards is enabled
FIMI/RemoveCMSAbonent/Rq/ChangeReason	Str(1000)	In	Reason for setting the messaging parameters
<b>ChangeCMSAbonent – editing messaging parameters for card/Telebank customer</b>			
FIMI/ChangeCMSAbonent/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/ChangeCMSAbonent/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ChangeCMSAbonent/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/ChangeCMSAbonent/Rq/AlternativeMessaging	ArrRec	In	{ Str(20)* Channel; Str(500)* Address; Str(20) Scheme; Str(20) Name; Str(150) Title; Int(1) Disabled; Str TestMessage; Int(1) UseForDynAuth; Int(1) IsDefault; Int(1) Priority; Int(1) Broadcast; Str(20) PrevChannel; Str(500) PrevAddress } } [] – list of channels and addresses for SMS commands delivery and reception:
<p>Channel, Address, Scheme – channel, address, scheme used for the commands delivery (list of available values for the channel and scheme provided in the fields <i>InitSession/Rp/CNSChannelList.Name</i> and <i>InitSession/Rp/CNSSchemeList.Name</i>).</p> <p>Name – name of the messaging abonent to be used in the commands.</p> <p>Title – any description.</p> <p>Disabled – “Abonent Disabled” attribute. If NULL, it=0.</p> <p>TestMessage – test message. If not NULL, the defined test message is sent to the customer via the defined channel and to the respective address.</p> <p>UseForDynAuth – use for dynamic authentication.</p> <p>IsDefault – “Default Abonent for Dynamic Authentication” attribute. It can be enabled only for one abonent for a certain card.</p> <p>Priority – priority of test message sending (0 – low; 1 – below average; 2 – average; 3 – above average; 4 – high). By default, 4 – high.</p> <p>Broadcast – allow the broadcast messages.</p>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
PrevChannel, PrevAddress – previous values of the channel and address. The PrevChannel and PrevAddress fields are filled only if the existing notification element is changed and filled with the previous change.			
FIMI/ChangeCMSAbonent/Rq/DisableNotification	Int(1)	In	Flag "Disable Notification on Card". If NULL, it=0
FIMI/ChangeCMSAbonent/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled =0 – do not notify a customer even if notification on the cards is enabled
FIMI/ChangeCMSAbonent/Rq/ChangeReason	Str(1000)	In	Reason for setting the messaging parameters
The FIMI/ChangeCMSAbonent/Rq/AlternativeMessaging/Disabled field affects sending of the customer notifications only. If FIMI/ChangeCMSAbonent/Rq/AlternativeMessaging/Disabled = 1, the notifications will not be generated but the commands incoming via sms from customers will be accepted and processed successfully. If FIMI/ChangeCMSAbonent/Rq/AlternativeMessaging/Scheme = NULL, the incoming customer commands will not be processed.			

***SetCard3DSecureAuthentication – setting E-Commerce (3-D Secure) authentication parameters for card***

FIMI/SetCard3DSecureAuthentication/Rq/PAN	Str(19)	In	Card number
FIMI/SetCard3DSecureAuthentication/Rq/MBR	Int(3)	In	Card member number
FIMI/SetCard3DSecureAuthentication/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetCard3DSecureAuthentication/Rq/ECUseCardSettingsAuth	Int(1)	In	Indicates that the card level settings are used for authentication in E-Commerce (3-D Secure).
FIMI/SetCard3DSecureAuthentication/Rq/ECNeedStaticAuth	Int(1)	In	Static authentication is required for E-Commerce (3-D Secure). Available values: 0 – Not required 1 – Required If NULL, the card parameter is not changed.
FIMI/SetCard3DSecureAuthentication/Rq/ECNeedDynPwdAuth	Int(1)	In	Authentication by the dynamic password (SMS/E-mail) is required for E-Commerce (3-D Secure). Available values: 0 – Not required 1 – Required If NULL, the card parameter is not changed.
FIMI/SetCard3DSecureAuthentication/Rq/ECNeedCAPAuth	Int(1)	In	CAP authentication is required for E-Commerce (3-D Secure). Available values: 0 – Not required 1 – CAP required 2 – CAP+Challenge required If NULL, the card parameter is not changed.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetCard3DSecureAuthentication/ Rq/ECNeedTokenAuth	Int(1)	In	Authentication by the dynamic password (using the hardware token) is required for E-Commerce (3-D Secure). Available values: 0 – Not required 1 – Required If NULL, the card parameter is not changed.
FIMI/SetCard3DSecureAuthentication/ Rq/ECUseDecoupledAuth	Int(1)	In	Attribute indicating whether the card supports decoupled authentication. Available values: 0 – not supported. 1 – supported. If the value is NULL, the card parameter is not changed.
FIMI/SetCard3DSecureAuthentication/ Rq/ChangeReason	Str(1000)	In	Reason for setting the E-Commerce (3-D Secure) authentication parameters
<b>ResetCardAbonentAdditionalInfo – reset of additional data on abonent for card/Telebank customer abonent</b>			
FIMI/ResetCardAbonentAdditionalInfo/ Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/ResetCardAbonentAdditionalInfo/ Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN field transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/ResetCardAbonentAdditionalInfo/ Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/ResetCardAbonentAdditionalInfo/ Rq/Channel	Str(20)*	In	Channel
FIMI/ResetCardAbonentAdditionalInfo/ Rq/Address	Str(500)*	In	Address
FIMI/ResetCardAbonentAdditionalInfo/ Rq/ChangeReason	Str(1000)	In	Reason for the abonent additional data reset
<b>TokenInquiry – request for information on card tokens</b>			
FIMI/TokenInquiry/Rq/PAN	Str(19)	In	Card number
FIMI/TokenInquiry/Rq/MBR	Int(3)	In	Card member number. If NULL, it is 0.
FIMI/TokenInquiry/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/TokenInquiry/Rq/Token	Str(19)	In	Token If NULL, return all card tokens.

<b>Parameter</b>	<b>Type</b> (* - mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/TokenInquiry/Rq/ NeedDevices	Int(1)	In	Indicates that the list of devices associated with the token must be present in the <i>Devices</i> structure, it is returned if the <i>Token</i> field is specified.  If NULL, it is 1.
FIMI/TokenInquiry/Rp/Tokens	ArrRec	Out	{ Str(19) Token; Str(48) TokenUID; Int (1) Type; Int(2) Status; Time ExpDate; Str(2) AssuranceLevel; Str(64) WalletAccountId; Str(48) DeviceId; Int(2) DeviceType; Str(99) DeviceName; Str(15) DeviceNumber; Int(11) TokenRequestorId; Str(64) PANRefId; Int(1) EventRequestor; Int WPDeviceScore; Int WPAccountScore; Int TokenScore; } } [] – list of card tokens:
<p>Token – token value.          Type – token type (see the list of available values in <i>FIMI/CreateCardToken/Rq/Type</i>).          TokenUID – unique token ID.          Status – token status (see the list of available values in <i>FIMI/GetCardTokensInfo/Rq&gt;Status</i>).          ExpDate – expiry date.          AssuranceLevel – assurance level.          WalletAccountId – wallet account ID.          DeviceId – device ID.          DeviceType – device type (see the list of available values in <i>FIMI/GetDevicesList/Rp/DeviceList</i>, the <i>Type</i> field).          DeviceName – device name.          DeviceNumber – device phone number or part of it.          TokenRequestorId – unique ID of the object that initiated the tokenization process.          PANRefId – PAN Reference ID.          EventRequestor – organization that executed the last action with token (0 – wallet, 1 – issuer, 2 – card owner, 3 – Mobile PIN Security, 4 – Mobile PIN Change Security, 5 – issuer communications provider, 6 – third-party vendor).          WPDeviceScore – Wallet Provider Device Score.          WPAccountScore – Wallet Provider Account Score.          TokenScore – Token Score.</p>			
FIMI/TokenInquiry/Rp/Devices	ArrRec	Out	{ Str(19)* Token; Int* DeviceIndex; Str(48)* DeviceId; } } [] – list of token devices

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetCardTokensInfo – request for information on card tokens in TWO storage</b>			
FIMI/GetCardTokensInfo/Rq/PAN	Str(19)	In	Card number
FIMI/GetCardTokensInfo/Rq/MBR	Int(3)	In	Card member number. If NULL, it=0
FIMI/GetCardTokensInfo/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetCardTokensInfo/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer The field value is used if the PersonId field is defined.  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .  <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name – it is allowed to request a customer for the institution if AllowedFIID.AccessMask.Bit[1]=1 or AllowedFIID.AccessMask.Bit[4]=1</i>
FIMI/GetCardTokensInfo/Rq/PersonId	Int(16)	In	Customer ID. The field value is used if the CardUID and PAN fields are not defined.
FIMI/GetCardTokensInfo/Rq/RefId	Str(48)	In	Unique card ID assigned by the payment system. The field value is used if the CardUID, PAN and PersonId fields are not defined.
FIMI/GetCardTokensInfo/Rq/DeviceType	Int(2)	In	Device type: See the list of available values in <b>FIMI/GetDevicesList/Rp/DeviceList</b> , field Type. If not set – any.
FIMI/GetCardTokensInfo/Rq/DeviceNumber	Str(15)	In	Phone number. The card number mask can be defined in the format of SQL-operator 'like', i.e., applying the metacharacters '_' (any single character), '%' (any number of any characters). If not set – any.
FIMI/GetCardTokensInfo/Rq/DeviceId	Str(48)	In	Device ID. If not set – any.
FIMI/GetCardTokensInfo/Rq/WalletId	Str(3)	In	Wallet ID: 101 - Masterpass by Mastercard 103 - Apple Pay 216 - Google Pay 217 - Samsung Pay 327 - Merchant Tokenization Program If not set – any.
FIMI/GetCardTokensInfo/Rq/TokenRequestorId	Int(11)	In	Unique ID of the object that initiated the tokenization process. If not set – any.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCardTokensInfo/Rq>Status	Int[]	In	Token statuses: 1 – active 2 – inactive 3 – suspended 4 – disabled 10 – token is removed from the device If not set – any.  The tokens can be searched for by card (if the PAN or CardUID field is defined), by customer (if the PersonId field is defined), by unique card ID (if the RefId field is defined). At least one of the CardUID, PAN, PersonId, RefId fields must be defined.
FIMI/GetCardTokensInfo/Rp/Tokens	ArrRec	Out	<pre>{   Str(19)* PAN;   Int(3)* MBR;   Str(64) CardUID;   Str(19)* Token;   Str(48) TokenUID;   Int(11) TokenRequestorId;   Str(2) AssuranceLevel;   Time* ExpDate;   Int(2) Status;   Int(2)* DeviceType;   Str(99) DeviceName;   Str(15) DeviceNumber;   Str(48) DeviceId;   Str(3) WalletId;   Str(64) WalletAccountId;   Int(1)* Type;   Int(1) CardOnFileUsage;   Int(1) ContactlessUsage;   Int(1) MobileWalletECUsage;   Str(48) PaymentAppliance;   Time TokenActivationTime;   Time LastChangeStatusTime;   Str(32) InternalDeviceId;   Str(240) TokenRequestorName; } [] – list of card tokens:</pre> <p>PAN – card number.            MBR – card member number.            CardUID – unique card ID.            Token – token value.            TokenUID – token unique ID.            TokenRequestorId – ID of the entity that requested the token creation (tokenization).            AssuranceLevel – assurance level.            ExpDate – expiry date.            Status – token status (see the list of available values in <i>FIMI/GetCardTokensInfo/Rq&gt;Status</i>).            DeviceType – device type (see the list of available values in <i>FIMI/GetDevicesList/Rp/DeviceList</i>, field <i>Type</i>).            DeviceName – device name.            DeviceNumber – device phone number or the part of it.            DeviceId – device ID.            WalletId – wallet ID (see the list of available values in <i>FIMI/GetCardTokensInfo/Rq/WalletId</i>).            ContactlessUsage – “Contactless Use” attribute.            CardOnFileUsage – “Merchant Stores Token” attribute.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																																																				
<p>MobileWalletECUsage – “Used in E-Commerce (3-D Secure)” attribute.</p> <p>Type – token type (see the list of available values in <i>FIMI/CreateCardToken/Rq/Type</i>).</p> <p>PaymentApplInstance – ID of the payment application.</p> <p>WalletAccountId – wallet profile ID.</p> <p>TokenActivationTime – date of activating/creating a token.</p> <p>LastChangeStatusTime – date of the last token status change.</p> <p>InternalDeviceId – device ID in <b>TWO</b>.</p> <p>TokenRequestorName – name of the object (based on the <b>TWO</b> dictionary) that initiated the tokenization process (it is transferred depending on the value in the <i>FIMI/Language</i> field).</p>																																																							
<b>CreateCardToken – creating card token in TWO storage</b>																																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/PAN</td> <td style="padding: 2px;">Str(19)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Card number</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/MBR</td> <td style="padding: 2px;">Int(3)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Card member number. If NULL, it=0</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/CardUID</td> <td style="padding: 2px;">Str(64)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Unique card identifier. It can stand for the card number.</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/Token</td> <td style="padding: 2px;">Str(19)*</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Token</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/TokenUID</td> <td style="padding: 2px;">Str(48)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Unique token ID</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/RefId</td> <td style="padding: 2px;">Str(48)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Unique card ID assigned by the payment system for tokenization.  If the value is transferred in the field, it is assigned to a card. If NULL, it is not changed.</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/TokenRequestorId</td> <td style="padding: 2px;">Int(11)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Unique ID of the object that initiated the tokenization process.</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/AssuranceLevel</td> <td style="padding: 2px;">Str(2)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Assurance level</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/ExpDate</td> <td style="padding: 2px;">Time*</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Expiration date</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/Language</td> <td style="padding: 2px;">Str(3)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Language</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq&gt;Status</td> <td style="padding: 2px;">Int(2)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Token statuses: 1 – active 2 – inactive 3 – suspended 4 – disabled 10 – token is removed from the device</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/Type</td> <td style="padding: 2px;">Int (1)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Token type: 0,NULL – unknown. 1 - E-Commerce/card on file. 2 - Secure element. 3 - Cloud-based payment. 5 - E-Commerce enabler. 6 - Pseudo account.</td> </tr> <tr> <td style="padding: 2px;">FIMI/CreateCardToken/Rq/DeviceNumber</td> <td style="padding: 2px;">Str(15)</td> <td style="padding: 2px;">In</td> <td style="padding: 2px;">Phone number</td> </tr> </table>				FIMI/CreateCardToken/Rq/PAN	Str(19)	In	Card number	FIMI/CreateCardToken/Rq/MBR	Int(3)	In	Card member number. If NULL, it=0	FIMI/CreateCardToken/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.	FIMI/CreateCardToken/Rq/Token	Str(19)*	In	Token	FIMI/CreateCardToken/Rq/TokenUID	Str(48)	In	Unique token ID	FIMI/CreateCardToken/Rq/RefId	Str(48)	In	Unique card ID assigned by the payment system for tokenization.  If the value is transferred in the field, it is assigned to a card. If NULL, it is not changed.	FIMI/CreateCardToken/Rq/TokenRequestorId	Int(11)	In	Unique ID of the object that initiated the tokenization process.	FIMI/CreateCardToken/Rq/AssuranceLevel	Str(2)	In	Assurance level	FIMI/CreateCardToken/Rq/ExpDate	Time*	In	Expiration date	FIMI/CreateCardToken/Rq/Language	Str(3)	In	Language	FIMI/CreateCardToken/Rq>Status	Int(2)	In	Token statuses: 1 – active 2 – inactive 3 – suspended 4 – disabled 10 – token is removed from the device	FIMI/CreateCardToken/Rq/Type	Int (1)	In	Token type: 0,NULL – unknown. 1 - E-Commerce/card on file. 2 - Secure element. 3 - Cloud-based payment. 5 - E-Commerce enabler. 6 - Pseudo account.	FIMI/CreateCardToken/Rq/DeviceNumber	Str(15)	In	Phone number
FIMI/CreateCardToken/Rq/PAN	Str(19)	In	Card number																																																				
FIMI/CreateCardToken/Rq/MBR	Int(3)	In	Card member number. If NULL, it=0																																																				
FIMI/CreateCardToken/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.																																																				
FIMI/CreateCardToken/Rq/Token	Str(19)*	In	Token																																																				
FIMI/CreateCardToken/Rq/TokenUID	Str(48)	In	Unique token ID																																																				
FIMI/CreateCardToken/Rq/RefId	Str(48)	In	Unique card ID assigned by the payment system for tokenization.  If the value is transferred in the field, it is assigned to a card. If NULL, it is not changed.																																																				
FIMI/CreateCardToken/Rq/TokenRequestorId	Int(11)	In	Unique ID of the object that initiated the tokenization process.																																																				
FIMI/CreateCardToken/Rq/AssuranceLevel	Str(2)	In	Assurance level																																																				
FIMI/CreateCardToken/Rq/ExpDate	Time*	In	Expiration date																																																				
FIMI/CreateCardToken/Rq/Language	Str(3)	In	Language																																																				
FIMI/CreateCardToken/Rq>Status	Int(2)	In	Token statuses: 1 – active 2 – inactive 3 – suspended 4 – disabled 10 – token is removed from the device																																																				
FIMI/CreateCardToken/Rq/Type	Int (1)	In	Token type: 0,NULL – unknown. 1 - E-Commerce/card on file. 2 - Secure element. 3 - Cloud-based payment. 5 - E-Commerce enabler. 6 - Pseudo account.																																																				
FIMI/CreateCardToken/Rq/DeviceNumber	Str(15)	In	Phone number																																																				

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateCardToken/Rq/DeviceType	Int(2)	In	Device type: See the list of available values in <i>FIMI/GetDevicesList/Rp/DeviceList</i> , field <i>Type</i> . If NULL, it is 0 – Unknown.
FIMI/CreateCardToken/Rq/DeviceName	Str(99)	In	Device name
FIMI/CreateCardToken/Rq/DeviceIndex	Str(4)	In	Device index
FIMI/CreateCardToken/Rq/DeviceId	Str(48)	In	Device ID
FIMI/CreateCardToken/Rq/WalletId	Str(3)	In	Wallet ID: 101 - Masterpass by Mastercard 103 - Apple Pay 216 - Google Pay 217 - Samsung Pay 327 - Merchant Tokenization Program
FIMI/CreateCardToken/Rq/WalletAccountId	Str(64)	In	Wallet account ID
FIMI/CreateCardToken/Rq/CardOnFileUsage	Int (1)	In	"Merchant Stores Token" attribute
FIMI/CreateCardToken/Rq/ContactlessUsage	Int (1)	In	"Contactless Use" attribute
FIMI/CreateCardToken/Rq/MobileWalletECUsage	Int (1)	In	"Used in E-Commerce (3-D Secure)" attribute
FIMI/CreateCardToken/Rq/PANSource	Int (1)	In	PAN source: 1 – manually 2 – card on file 3 – application (mobile bank)
FIMI/CreateCardToken/Rq/PaymentApplInstance	Str(48)	In	Payment application ID
<b>DeleteCardToken – deleting card token in TWO storage</b>			
FIMI/DeleteCardToken/Rq/PAN	Str(19)	In	Card number
FIMI/DeleteCardToken/Rq/MBR	Int(3)	In	Card member number. If NULL, it=0
FIMI/DeleteCardToken/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/DeleteCardToken/Rq/Token	Str(19)*	In	Token
<b>UpdateCardToken – updating card token in TWO storage</b>			
FIMI/UpdateCardToken/Rq/PAN	Str(19)	In	Card number
FIMI/UpdateCardToken/Rq/MBR	Int(3)	In	Card member number. If NULL, it=0
FIMI/UpdateCardToken/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateCardToken/Rq/Token	Str(19)*	In	Token
FIMI/UpdateCardToken/Rq/Status	Int(2)	In	Token statuses: 1 – active 2 – inactive 3 – suspended 4 – disabled 10 – token is removed from the device NULL – do not change
FIMI/UpdateCardToken/Rq/ExpDate	Time	In	Expiration date NULL – do not change
<b>ReassignCardTokens – re-assigning card tokens</b>			
FIMI/ReassignCardTokens/Rq/PAN	Str(19)	In	Card number
FIMI/ReassignCardTokens/Rq/MBR	Int(3)	In	Card member number. If NULL, it=0
FIMI/ReassignCardTokens/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/ReassignCardTokens/Rq/ParentPAN	Str(19)	In	Parent card number If NULL, the parent card number will be determined by the card number on the basis of data on the parent card.
FIMI/ReassignCardTokens/Rq/ParentMBR	Int(3)	In	Parent card MBR. If NULL, it=0
FIMI/ReassignCardTokens/Rq/ParentCardUID	Str(64)	In	Unique identifier of the parent card. It can stand for number of the new card.
FIMI/ReassignCardTokens/Rq/NeedCreateReassignTokens	Int(1)	In	Attribute indicating that it is required to reassign card tokens from the parent card in the TWO DB. If NULL, it=0
<b>SetCardTokenStatus – changing card token status</b>			
FIMI/SetCardTokenStatus/Rq/PAN	Str(19)	In	Card number
FIMI/SetCardTokenStatus/Rq/MBR	Int(3)	In	Card member number
FIMI/SetCardTokenStatus/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetCardTokenStatus/Rq/Token	Str(19)	In	Token. It can be used for cards with the Mastercard, Visa, Mir, UPI card program. If NULL, the status is changed for all card tokens (the value is allowed for the Mastercard, Mir cards).

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetCardTokenStatus/Rq/Status	Int(2)*	In	New token status. Available values: 1 – active 2 – inactive 3 – suspended 4 – disabled 10 – token is removed from the device
FIMI/SetCardTokenStatus/Rq/NeedCheckToken	Int(1)	In	Attribute indicating that it is required to check whether the card token is present in the <b>TWO</b> DB. If NULL, it=0
FIMI/SetCardTokenStatus/Rq/ChangeReason	Str(1000)	In	Reason for setting the token status
<b>GetCardTokenNeedConfirm – request for information on card token operations which require confirmation</b>			
FIMI/GetCardTokenNeedConfirm/Rq/PAN	Str(19)	In	Card number
FIMI/GetCardTokenNeedConfirm/Rq/MBR	Int(3)	In	Card member number
FIMI/GetCardTokenNeedConfirm/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetCardTokenNeedConfirm/Rq/Token	Str(19)	In	Token If NULL, it can be any
FIMI/GetCardTokenNeedConfirm/Rp/Tokens	ArrRec	Out	{ Str(19)* PAN; Int(3)* MBR; Str(64) CardUID; Str(19)* Token; Int(11)* TokenRequestorId; Time* OrigTime; Int(1)* Reason; Int(12)* RequestTranId; Str TokenData; } [] – list of operations which require confirmation
PAN – card number.			
MBR – card member number.			
CardUID – unique card ID.			
Token – token value.			
TokenRequestorId – unique ID of the object that initiated the tokenization process.			
OrigTime – confirmation request origination time.			
Reason – confirmation reason (4 - Device Binding, 5 - Cardholder StepUp).			
RequestTranId – ID of the confirmation request transaction.			
TokenData – contains token data in the UAMP format. For details on the field format, refer to <i>TW International Container Fields Specification</i> . It is transferred in the Base64 encoding.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>CardTokenConfirmation – card token operation confirmation</b>			
FIMI/CardTokenConfirmation/Rq/PAN	Str(19)	In	Card number
FIMI/CardTokenConfirmation/Rq/MBR	Int(3)	In	Card member number
FIMI/CardTokenConfirmation/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/CardTokenConfirmation/Rq/RequestTranId	Int(12)*	In	ID of the confirmation request transaction. The value received within the <i>GetCardTokenNeedConfirm</i> request.
FIMI/CardTokenConfirmation/Rq/Source	Int(1)*	In	Confirmation source: 1 – Call center. 2 – mobile bank.
<b>GetCardPAR – request for Payment Account Reference (PAR) by card/card token</b>			
FIMI/GetCardPAR/Rq/PAN	Str(19)	In	Card number and card member number.
FIMI/GetCardPAR/Rq/MBR	Int(3)	In	If MBR is not transferred, it is 0.
FIMI/GetCardPAR/Rq/CardUID	Str(64)	In	Unique card ID. It can stand for the card number.
FIMI/GetCardPAR/Rq TokenName	Str(19)	In	Token
FIMI/GetCardPAR/Rq/NeedCheckToken	Int(1)	In	Attribute indicating that it is required to check whether the card token is present in the TWO DB. It is used if the card token is transferred in the Token field. If NULL, it is 0.  If 1, it is required to transfer the card number in the PAN field (or CardUID).
To request for the PAR data, it is required to transfer one of the key fields in the request: Token, PAN (or CardUID). Otherwise, the request will be declined with code 62 - InvalidParameterValue. In case of the PAR request by token, the data is to be transferred in the Token field. In case of the PAR request by card number, the card number is to be transferred in the PAN field or card UID – in the CardUID field.			
FIMI/GetCardPAR/Rq/PAR	Str(29)	Out	Payment Account Reference (PAR)
FIMI/GetCardPAR/Rq/PARCreationDate	Time	Out	Payment Account Reference creation date
<b>GetCardCounters – getting card/Telebank customer counters</b>			
FIMI/GetCardCounters/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/GetCardCounters/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/GetCardCounters/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description			
FIMI/GetCardCounters/Rq/AuthScheme	Int	In	<p>Authorization scheme ID.</p> <p>If defined, the counter parameters are returned considering the authorization scheme parameters.</p> <p>See the list of available values in <i>InitSession/Rp/AuthSchemes</i>.</p>			
FIMI/GetCardCounters/Rq/CardProfile	Int	In	<p>Card profile ID; the field must be defined if <i>AuthScheme</i> is transferred.</p> <p>See the list of available values in <i>GetCardInfo/Rp/CardProfiles</i>.</p>			
FIMI/GetCardCounters/Rp/Counters	ArrRec	Out	<pre>{     int* CounterId;     num*.MaxValue;     num* Current;     int(2)* PeriodType;     time DateTimeForReset;     time LastResetTime;     int Period;  }</pre> <p>} [] – list of the card counters</p>			
<p>CounterId – counter ID in <b>TWO</b> (see the list of available values in <i>FIMI/InitSession/Rp/Counters</i>);</p> <p>.MaxValue – maximum counter value (available values &gt;=0, -1 – “as the scheme”, -2 – “unlimited”);</p> <p>Current – current counter value;</p> <p>LastResetTime – time of the last reset;</p> <p>PeriodType – period type:</p> <ul style="list-style-type: none"> <li>-1 – as the authorization scheme</li> <li>- 0 – Absolute (Daily) – absolute period; it starts from the beginning of the day (1440 – it is reset every day at 0:00, 2880+420 – at 7:00 every second day)</li> <li>- 1 – Weekly – calendar weekly period; it starts from the beginning of the week (2880+420 – it is reset every Wednesday at 7:00)</li> <li>- 2 – Monthly – calendar monthly period; it starts from the beginning of the month (2880+420 – it is reset on the third day of every month at 7:00)</li> <li>- 3 – Quarterly – calendar quarterly period; it starts from the beginning of the quarter (2880+420 – it is reset on the third day of every quarter at 7:00)</li> <li>- 4 – Infinite – counter with the unlimited validity period (non-resettable)</li> <li>- 5 – Single operation – one-time counter</li> <li>- 6 – Reset in Refresh – counter is reset on Refresh</li> <li>- 7 – Yearly – the calendar year limit, the period starts from the beginning of the year (Period=0 - the counter is reset on the 1st of January at 00:00; Period=2880+420 – the counter is reset each year on the 3<sup>rd</sup> of January at 7:00)</li> <li>- 11 – Calendar – counter is reset on a calendar date and time (the <i>DateTimeForReset</i> field is used for this type)</li> </ul> <p>Period – limit reset time (min):</p> <p>Period/1440 - number of day (starts from 0); Mod(LimitPeriod,1440) – reset time within a day (min.)</p> <p>Restrictions:</p> <table style="margin-left: 20px;"> <tr> <td>PeriodType=0 → 1&lt;=Period&lt;=1440000</td> <td>PeriodType=1 → 0&lt;=Period&lt;1440*7</td> </tr> <tr> <td>PeriodType=2 → 0&lt;=Period&lt;1440*31</td> <td>PeriodType=3 → 0&lt;=Period&lt;1440*90</td> </tr> <tr> <td>PeriodType=7 → 0&lt;=Period&lt;1440*360</td> <td></td> </tr> </table>	PeriodType=0 → 1<=Period<=1440000	PeriodType=1 → 0<=Period<1440*7	PeriodType=2 → 0<=Period<1440*31	PeriodType=3 → 0<=Period<1440*90	PeriodType=7 → 0<=Period<1440*360	
PeriodType=0 → 1<=Period<=1440000	PeriodType=1 → 0<=Period<1440*7					
PeriodType=2 → 0<=Period<1440*31	PeriodType=3 → 0<=Period<1440*90					
PeriodType=7 → 0<=Period<1440*360						
<b>SetCardCounters – changing card/Telebank customer counters</b>						
FIMI/SetCardCounters/Rq/PAN	Str(19)	In	Number of the card and card owner number or Telebank customer ID.			
FIMI/SetCardCounters/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SetCardCounters/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/SetCardCounters/Rq/RemoveMissing	Int(1)	In	=1 – remove unsent counters By default = 0
FIMI/SetCardCounters/Rq/Counters	ArrRec	In	{ int* CounterId; num.MaxValue; num.CurrentDelta; int(2).PeriodType; time.DateTimeForReset; int.Period         } [] – list of card counters:
<p>CounterId – counter ID in <b>TWO</b> (see the list of available values in <i>FIMI/InitSession/Rp/Counters</i>).            MaxValue – maximum counter value (available values &gt;=0, -1 – “as the scheme”, -2 – “unlimited”).            CurrentDelta – positive or negative delta of the counter value change, if 0, the counter is reset.            PeriodType, DateTimeForReset, Period – period type and value (see the description in <i>GetCardCounters</i>).            If MaxValue, CurrentDelta, PeriodType = NULL, no impact on the respective attribute is performed.</p>			
FIMI/SetCardCounters/Rq/ChangeReason	Str(1000)	In	Reason for the card counter change
<b>AttachCards – linking cards</b>			
FIMI/AttachCards/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/AttachCards/Rq/MBR	Int(3)	In	
FIMI/AttachCards/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AttachCards/Rq/LinkedPAN	Str(19)	In	Number of the card to be linked
FIMI/AttachCards/Rq/LinkedMBR	Int(3)	In	MBR of the card to be linked
FIMI/AttachCards/Rq/LinkedCardUID	Str(64)	In	Unique ID of the card to be linked. It can stand for the card number.
FIMI/AttachCards/Rq/LinkType	Int(1)*	In	Cards link type: 1 – co-branding 2 – dual card
FIMI/AttachCards/Rq/Force	Int(1)	In	=NULL or 0 – check is performed; if transactions for linked cards are found in TLG (except for the FIMI creation transactions), the operation is declined =1 – cards are linked and no check of whether transactions on linked cards are present is performed

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b><i>SetCardContactlessInterface – changing status of contactless card interface</i></b>			
FIMI/SetCardContactlessInterface/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/SetCardContactlessInterface/Rq/MBR	Int(3)	In	
FIMI/SetCardContactlessInterface/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetCardContactlessInterface/Rq/ContactlessStatus	Int(1)*	In	Contactless interface status to be set: 0 – disabled 1 – enabled
FIMI/SetCardContactlessInterface/Rq/ChangeReason	Str(1000)	In	Change reason
<b><i>UpdateCardMetadata – changing card metadata</i></b>			
FIMI/UpdateCardMetadata/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/UpdateCardMetadata/Rq/MBR	Int(3)	In	
FIMI/UpdateCardMetadata/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/UpdateCardMetadata/Rq/NewPlasticType	Str(32)*	In	New value of the card plastic type code <i>The list of available values is transferred in the PlasticTypesList structure of the InitSession operation</i>
FIMI/UpdateCardMetadata/Rq/NeedAdvice	Int(1)	In	Attribute indicating that it is required to generate the notification on the change for the external system. If NULL, it is 1.
FIMI/SetCardContactlessInterface/Rq/ChangeReason	Str(1000)	In	Change reason
<b><i>GetConsumerPresentedQRCode – request for consumer-presented QR code with card data for payment</i></b>			
FIMI/GetConsumerPresentedQRCode/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/GetConsumerPresentedQRCode/Rq/MBR	Int(3)	In	
FIMI/GetConsumerPresentedQRCode/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetConsumerPresentedQRCode/Rq/ImageType	Int(1)	In	QR code image format: 0 – no image is required. 1 – PNG. 2 – SVG. If NULL, it is 0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetConsumerPresentedQRCode/Rq/ImageWidth	Int(4)	In	Image width in pixels. The minimum length is 200.
FIMI/GetConsumerPresentedQRCode/Rq/ImageHeight	Int(4)	In	Image height in pixels. The minimum length is 200.
FIMI/GetConsumerPresentedQRCode/Rp/QRPayload	Str*	Out	Payload of the QR code in the Base64 encoding.
FIMI/GetConsumerPresentedQRCode/Rp/QRImage	Str	Out	QR code image. It is transferred in the Base64 format if the ImageType value is 1 or 2.

### 2.3.3.2 On Account:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetAcctInfo – account data request</b>			
FIMI/GetAcctInfo/Rq/InstName	Str(4)	In	Character identifier (name) of a financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.  <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name – it is allowed to request an account for the institution if AllowedFIID.AccessMask.Bit[1]=1 or AllowedFIID.AccessMask.Bit[3]=1</i>
FIMI/GetAcctInfo/Rq/Account	Str(30)	In	Account number (permissible mask is identical to the one used in the GetCardInfo request in the field Rq/PAN)
FIMI/GetAcctInfo/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/GetAcctInfo/Rq/PersonId	Int(16)	In	Customer ID. It can be used for the definite account search if masking is enabled.
FIMI/GetAcctInfo/Rq/AggregateId	Str(40)	In	Account aggregate ID (permissible mask is identical to the one used in the GetCardInfo request in the field Rq/PAN). The field is used if the Account and AccountUID fields are not defined.
FIMI/GetAcctInfo/Rp/FoundAccts	ArrRec	Out	{ Str(30)* Account; int(16) PersonID; str(300) PersonFIO; str(32) AccountUID; str(240) Branch;

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			int* Type; int* Status; str(40) Aggregateld; int AggregatePriority; str(90) BranchId;  } [] – list of the found accounts, not more than 100; The field is defined if more than one account is found (in the request, in the Account field, there was defined a mask on which more than one account is found); If defined, all the other fields in the response beginning with the prefix FIMI/GetAcctInfo/Rp will be empty.
FIMI/GetAcctInfo/Rp/FoundAccount	Str(30)	Out	Number of the found account
FIMI/GetAcctInfo/Rp/FoundAccountUID	Str(32)	Out	Account unique identifier within a session. It can stand for the account number in the following requests
FIMI/GetAcctInfo/Rp/ExtendedAccountNumber	Str(30)	Out	Extended account number
FIMI/GetAcctInfo/Rq/ParentAcct	Str(30)	Out	Parent account number
FIMI/GetAcctInfo/Rq/ParentAcctUID	Str(32)	Out	Unique identifier of the parent account within a session. It can stand for the account number in the following requests
FIMI/GetAcctInfo/Rq/Branch	Str(240)	Out	Account branch name
FIMI/GetAcctInfo/Rq/BranchId	Str(90)	Out	Account branch identifier
FIMI/GetAcctInfo/Rp/Type	Int(2)	Out	Account type: 1 – checking 11 – savings 31 – credit 91 – bonus
FIMI/GetAcctInfo/Rp>Status	Int(1)	Out	Account status: 0 – Inactive account 1 – Open 2 – Deposit only 3 – Open primary account 4 – Deposit only primary account 5 – Information only 9 – Closed
FIMI/GetAcctInfo/Rp/Currency	Int	Out	Account currency
FIMI/GetAcctInfo/Rp/Ledger	Num	Out	Ledger balance
FIMI/GetAcctInfo/Rp/Avail	Num	Out	Available balance
FIMI/GetAcctInfo/Rp/BalanceText	Str(1000)	Out	Balance explanatory text

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetAcctInfo/Rp/DebitHold	Num	Out	Debit hold (amount of the debit authorizations)
FIMI/GetAcctInfo/Rp/CreditHold	Num	Out	Credit hold (amount of the deposit authorizations)
FIMI/GetAcctInfo/Rp/ProtectedAmount	Num	Out	Protected amount
FIMI/GetAcctInfo/Rp/Bonus	Num	Out	Bonus/Debt If it >0, it is bonus; If it <0, it is a debt.
FIMI/GetAcctInfo/Rp/Overdraft	Num	Out	Permissible overdraft
FIMI/GetAcctInfo/Rp/TmpOverdraft	Num	Out	Temporary overdraft
FIMI/GetAcctInfo/Rp/TmpOverdraftExpiration	Time	Out	Expiration date of the temporary overdraft
FIMI/GetAcctInfo/Rp/TmpOverdraftPeriodType	Int(1)	Out	Temporary overdraft period type. Available values: 1 – overdraft per one operation NULL – overdraft per period
FIMI/GetAcctInfo/Rp/DropTmpOverOnRefresh	Int(1)	Out	Flag 'Reset Temporary Overdraft in Refresh' NULL – do not reset 1 – reset on receiving the next Refresh on account 2 – reset on receiving the Nth Refresh on account
FIMI/GetAcctInfo/Rp/DropTmpOverOnDate	Time	Out	Reset the temporary overdraft when receiving Refresh on the specified date
FIMI/GetAcctInfo/Rp/DropTmpOverNumber	Int	Out	Number of account Refreshes in which temporary overdraft must be reset
FIMI/GetAcctInfo/Rp/Remain	Num	Out	Bank account remain
FIMI/GetAcctInfo/Rp/PersonId	Int(16)	Out	Account owner ID
FIMI/GetAcctInfo/Rp/PersonExtId	Str(50)	Out	Customer ID in the external system
FIMI/GetAcctInfo/Rp/PersonFIO	Str(300)	Out	Full name of the account owner
FIMI/GetAcctInfo/Rp/LastTranId	Int(12)	Out	Last transaction ID
FIMI/GetAcctInfo/Rp/LastWdlTime	Time	Out	Last withdrawal time
FIMI/GetAcctInfo/Rp/LastWdlAmount	Num	Out	Last withdrawal amount
FIMI/GetAcctInfo/Rp/LastDepTime	Time	Out	Last deposit time
FIMI/GetAcctInfo/Rp/LastDepAmount	Num	Out	Last deposit amount
FIMI/GetAcctInfo/Rp/LastRefreshTime	Time	Out	Time when the account was updated for the last time by the Refresh procedure

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetAcctInfo/Rp/AggregateId	Str(40)	Out	Account aggregate ID
FIMI/GetAcctInfo/Rp/AggregatePriority	Int	Out	Account priority in the aggregate
FIMI/GetAcctInfo/Rp/PermissibleExcessType	Int(1)	Out	Type of the allowed excess of the available account balance: -1 – not used. 0 – percent of the balance amount. 1 – excess in the account currency.
FIMI/GetAcctInfo/Rp/PermissibleExcess	Num	Out	Value of the allowed excess of the available account balance in percents or amount. If PermissibleExcessType=0, it contains percents in the range of 0.001 to 99.999. If PermissibleExcessType=1, it contains the amount in the account currency.
FIMI/GetAcctInfo/Rp/LegalEntityId	Int(16)	Out	Identifier of a corporate customer linked to the account
FIMI/GetAcctInfo/Rp/LegalEntityName	Str(150)	Out	Name of a corporate customer linked to the account
FIMI/GetAcctInfo/Rp/Cards	ArrRec	Out	{ str(19)* PAN; int(3)* MBR; str(250) Descr; str(64) CardUID; int(2)* Status; int(1)* Type. } [] – array of structures, List of the account-linked cards: PAN – card number. MBR – card member number. Descr – description of the card linked to account. CardUID – unique card identifier. Status – card status (it is not account status in the link). See the list of available values in the field <i>InitSession/Rp/CardStatList.Id</i> or in section 3.2). Type – card type (see the list of available values in the field <i>FIMI/GetCardInfo/Rq/Type</i> ).}
FIMI/GetAcctInfo/Rp/MaskBalances	Int(1)	Out	=1 — account balances should be masked
FIMI/GetAcctInfo/Rp/UserFields	ArrRec	Out	{ str(100)* Name; str(4000) TextValue; strXML XMLValue;

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<p>} [] – array of structures,  List of the account user fields; if is defined if only one account is found  Name, TextValue, XMLValue – name, text and XML values of the user fields</p>
<b>GetAcctStatement – account statement request</b>			
FIMI/GetAcctStatement/Rq/InstName	Str(4)	In	<p>Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.</p>
FIMI/GetAcctStatement/Rq/Account	Str(30)	In	Account number
FIMI/GetAcctStatement/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/GetAcctStatement/Rq/PersonId	Int(16)	In	Customer ID
FIMI/GetAcctStatement/Rq/Count	Int(4)	In	Number of the last requested operations. If not defined, all the operations are returned (not more than 1000)
FIMI/GetAcctStatement/Rq/Origin	Int(1)	In	<p>Operation origin:  1 – Processing – operation conducted in the current business day and not yet confirmed in Back-Office;  3 – Retail – operation received with Refresh from Back-Office.  If not defined, both the processing and retail operations are returned.</p>
FIMI/GetAcctStatement/Rq/Type	Int(1)	In	<p>Operation type:  1 – financial  2 – intracontractual (on the composite account)  3 – authorizational  If not defined, operations of all the types return.</p>
FIMI/GetAcctStatement/Rq/FromTime	Time	In	Starting time, NULL – from the beginning
FIMI/GetAcctStatement/Rq/ToTime	Time	In	End time, NULL – till the current moment

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetAcctStatement/Rq/ Language	Int	In	<p>ID of language in which the <i>AnotherTitle</i> field must be returned (the <i>Statement</i> structure). The language code is taken from the <b>TWO</b> dictionary.</p> <p>If the field is not defined, the standard description of field returns.</p> <p>This parameter is used only if the operations history is requested via Core Banking authorizer.</p>
FIMI/GetAcctStatement/Rp/ Statement	ArrRec	Out	<pre>{     int(12) FrontId;     str(12) BackId;     int(1)* Origin;     int(1)* Type;     int(2)* OperCode;     str(300) AnotherTitle;     str(4000) Description;     num* Amount;     time* OperDate;     time TranTime;     num OrigAmount;     str(3) OrigCurrency;     str(19) PAN;     int(3) MBR;     int(1) TermClass;     str(16) TermName;     int TermSIC;     str(750) TermLocation;     str(30) BackAcct;     num BackAmount;     num Remain;     str(8) ApprovalCode;     str(64) CardUID;     int* SeqNo;     int(3) TermCountry;     str(90) TermCity;     num OnlinelssuerFee;     time OrigTime;     str(750) MessToRecipient;     str(2) BAI;     num TIPAmount;     num ProtectedAmount;     str(16) RetailerServicePhone;     str(255) RetailerURL;     str(3) Currency;     str(150) TermRetailerName;     int(3)* CurrencyISOCode;     int(3) OrigCurrencyISOCode; }</pre> <p>} [] – array of structures,</p> <p>Operations history on the account:</p> <p>The history is sorted by time in the descending order (starting from the last</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<p>operations). The number of the operations should not exceed Min(Rq/Count,1000);</p> <p>FrontId – source transaction ID in Front-Office – for the processing operations.</p> <p>BackId – entry ID in Back-Office.</p> <p>Origin – operation origin: 1 – Processing, 3 – Retail.</p> <p>Type – operation type: 1–financial 2–intracontractual, 3–authorization.</p> <p>OperCode – operation code (see p. 3.6).</p> <p>AnotherTitle – priority operation title, if not NULL, it overlays OperCode.</p> <p>Description – operation description – can be defined for the bank operations.</p> <p>Amount – operation amount in the account currency with the sign.</p> <p>OperDate – Front-Office and Back-Office business day.</p> <p>TranTime – transaction time – for the processing operations.</p> <p>OrigAmount – operation original amount.</p> <p>OrigCurrency – operation original currency – 3-char value.</p> <p>PAN, MBR – number of the card on which a processing operation was conducted.</p> <p>TermClass – terminal class: 1-ATM, 2-POS, 3-CRT, 4-TELEBANK.</p> <p>TermName – terminal name.</p> <p>TermSIC – MCC terminal.</p> <p>TermLocation – terminal location.</p> <p>BackAcct – account number in Back-Office for the contractual operation (Type=2); If ! =t NULL and differs from the main account number, the operation has been conducted in Back-Office on the contractual account.</p> <p>BackAmount – operation amount in the Back-Office account currency.</p> <p>Remain – bank account balance after conducting an operation – known for the bank operations only.</p> <p>ApprovalCode – approval code.</p> <p>CardUID – unique card identifier.</p> <p>SeqNo – operation sequence number.</p> <p>TermCountry – terminal country code.</p> <p>TermCity – terminal city.</p> <p>OnlineIssuerFee – online issuer fee.</p> <p>OrigTime – transaction time in the originator time zone.</p> <p>MessToRecipient – message to the recipient.</p> <p>BAI – Business Application Identifier (for details, see <i>FIMI/POSRequest/Rq/BusinessApplicationIdent</i>).</p> <p>TIPAmount – tip amount.</p> <p>ProtectedAmount – protected amount.</p> <p>RetailerServicePhone – phone number of the support service.</p> <p>RetailerURL – URL of the online store of the merchant.</p> <p>Currency – 3-character designation of the Amount field currency.</p> <p>CurrencyISOCode – ISO code of the Amount field currency.</p> <p>OrigCurrencyISOCode – ISO code of the OrigAmount field currency.</p> <p>TermRetailerName – merchant name.</p> <p>If FIMI/GetAcctStatement/Rq/Origin = 1, ID from the transactions log is used as SeqNo.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetAcctStatement/Rp/ AliasService	ArrRec	Out	<pre>{     int*      SeqNo;     str       SenderMessage;     str(32)   ExternalTranId;     str(12)   RecipientMemberId;     str       RecipientFIO;     str(500)  RecipientAlias;     str(12)   SenderMemberId;     str       SenderFIO;     str(500)  SenderAlias;     int(3)    SenderCountry;     int(3)    RecipientCountry;     num      DebtorAmount;     int(3)    DebtorCurrency;     num      CreditorAmount;     int(3)    CreditorCurrency;     num      Fee;     str(10)   PaymentDetailsCategoryCode; } [] – array of structures, additional data for the operation of transfer by alias:</pre> <p>SeqNo – sequence number of the operation it matches the SeqNo field of the Statement structure.</p> <p>SenderId – sender message for the recipient specified during the transfer.</p> <p>ExternalTranId – external transaction ID.</p> <p>RecipientMemberId – ID of the transfer service participant on the recipient side.</p> <p>RecipientFIO – full name of the recipient.</p> <p>RecipientAlias – recipient alias.</p> <p>SenderId – ID of the transfer service participant on the sender side.</p> <p>SenderFIO – full name of the sender.</p> <p>SenderAlias – sender alias.</p> <p>SenderId – sender country (returned for international transfers).</p> <p>RecipientCountry – recipient country (returned for international transfers).</p> <p>DebtorAmount – amount to be debited from the sender (returned for international transfers).</p> <p>DebtorCurrency – ISO code of the currency for the amount to be debited (returned for international transfers).</p> <p>CreditorAmount – amount to be credited to the recipient (returned for international transfers).</p> <p>CreditorCurrency – ISO code of the currency for the amount to be credited (returned for international transfers).</p> <p>Fee – operation fee in the debit currency.</p> <p>PaymentDetailsCategoryCode – transfer category code.</p>
FIMI/GetAcctStatement/Rp/ QRService	ArrRec	Out	<pre>{     int*      SeqNo;     str(140)  PaymentPurpose;     str(140)  RetailerTitle;     str(140)  RetailerName;     str(140)  RetailerAddress;     str(12)   MemberId;     int      MCC;</pre>

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
			<pre>str(4) QR.Payload.CRC; str(32) External.TranId; str(32) Ref.External.TranId; } [] – array of structures, additional data for the operation executed using the QR code:</pre> <p>SeqNo – sequence number of the operation it matches the SeqNo field of the Statement structure.      RetailerTitle – title of the merchant that issued the QR code.      RetailerName – name of the merchant that issued the QR code.      RetailerAddress – address of the merchant that issued the QR code.      MemberId – member ID within the service (the list of available options is requested in the <i>GetServiceMembersList</i> operation).      MCC – merchant category code.      QR.Payload.CRC – value of the checksum of the QR code payload.      External.TranId – external identifier of a transaction.      Ref.External.TranId – external identifier of the linked transaction, it contains the external identifier of the original transaction when the operation is canceled.</p>
FIMI/GetAcctStatement/Rp/ GovernmentService	ArrRec	Out	<pre>{ int* SeqNo; str(32) UID; str(25) PUID; str(140) LegalEntityName; str(32) External.TranId; str(140) PayerName; str(12) PayerTaxNumber; str(9) PayerTaxReasonRegCode; } [] – array of structures, additional data for the operation of transfer in favour of the government.</pre> <p>SeqNo – sequence number of the operation, it is the same as the SeqNo field of the Statement structure.      UID – unique identifier assigned by the payer bank.      PUID – unique payment identifier.      LegalEntityName – corporate customer name.      External.TranId – external transaction identifier.      PayerName – real payer: corporate customer name or title.      PayerTaxNumber – TPN of a real payer.      PayerTaxReasonRegCode – tax registration reason code of a real payer.</p>
FIMI/GetAcctStatement/Rp/ MultiAcct	ArrRec	Out	<pre>{ int(12)* SeqNo; str(30)* Account; int* Currency; num* Amount; } [] – array of structures, list of accounts with amounts participated in the authorization (if several accounts were used):</pre> <p>SeqNo - operation sequence number. If FIMI/GetAcctStatement/Rq/Origin = 1, ID from the transactions log is used as SeqNo.      Account – account.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
Currency – account currency ID. Amount – amount debited from account.			
FIMI/GetAcctStatement/Rp/ ArrestImpact	ArrRec	Out	{ int* SeqNo; str(4)* InstName; str(32)* ArrestId; num* Amount; str Description; } [] – array of structures, list of impacts on the arrest during the credit operation.
SeqNo – sequence number of the operation. Amount – arrested amount in the account currency debited from the account. InstName – institution name. ArrestId – arrest ID (data on the arrest can be obtained using the GetArrests operation). Description – description.			
<b>SetAcctStatus – setting account status</b>			
FIMI/SetAcctStatus/Rq/InstName	Str(4)	In	Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/SetAcctStatus/Rq/Account	Str(30)	In	Account number
FIMI/SetAcctStatus/Rq/ AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/SetAcctStatus/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetAcctStatus/Rq/Status	Int(1)*	In	Account new status: 0 – Inactive account 1 – Open 2 – Deposit only 3 – Open primary account 4 – Deposit only primary account 5 – Information only 9 – Closed
FIMI/SetAcctStatus/Rq/ ChangeReason	Str(1000)	In	Reason for setting the account status
<b>ChangeAcctBalances – changing attributes concerning account balances</b>			
FIMI/ChangeAcctBalances/Rq/ InstName	Str(4)	In	Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/ChangeAcctBalances/Rq/ Account	Str(30)	In	Account number
FIMI/ChangeAcctBalances/Rq/ AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ChangeAcctBalances/Rq/PersonId	Int(16)	In	Customer ID
FIMI/ChangeAcctBalances/Rq/LedgerDelta	Num	In	Increment of ledger balance in the account currency. This value is added to the current value of the account ledger balance. The value can be both positive and negative. If not defined, the account ledger balance is left unchanged.
FIMI/ChangeAcctBalances/Rq/AvailDelta	Num	In	Increment of available balance in the account currency. This value is added to the current value of the account available balance. The value can be both positive and negative. If not defined, the account available balance is left unchanged.
FIMI/ChangeAcctBalances/Rq/BonusDelta	Num	In	Bonus increment in the account currency. This value is added to the current value of the account "bonus/debt". The value can be both positive and negative. If not defined, the account "bonus/debt" is left unchanged.
FIMI/ChangeAcctBalances/Rq/OverdraftDelta	Num	In	Increment of available overdraft in the account currency. This value is added to the current value of the account available overdraft. The value can be both positive and negative. If not defined, the account available overdraft is left unchanged.
FIMI/ChangeAcctBalances/Rq/TranNumber	Str(50)	In	Operation ID in the external system.  If the field is filled, it overlays the value of the <i>FIMI/TransactionNumber</i> field from the request header.  If this field or the <i>FIMI/TransactionNumber</i> field is filled, the system searches for the duplicates. If the duplicate is found, the account balance is not changed.
FIMI/ChangeAcctBalances/Rq/TmpOverdraftDelta	Num	In	Increment of temporary overdraft in the account currency. This value is added to the current value of the account temporary overdraft. The value can be both positive and negative. If not defined, the account temporary overdraft is left unchanged.  Attention! To activate the temporary overdraft, specify valid expiration date in the TmpOverdraftExpiration field.
FIMI/ChangeAcctBalances/Rq/TmpOverdraftExpiration	Time	In	Expiration date of the temporary overdraft. It is used together with the TmpOverdraftDelta. If the TmpOverdraftExpiration is not defined, this attribute is left unchanged.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ChangeAcctBalances/Rq/TmpOverdraftPeriodType	Int(1)	In	Temporary overdraft period type. Available values: NULL – leave previous value; 1 – overdraft per one operation 0 – overdraft per period
FIMI/ChangeAcctBalances/Rq/DropTmpOverOnRefresh	Int(1)	In	Flag 'Rest Temporary Overdraft in Refresh'. NULL, 0 – do not reset; 1 – reset on receiving the next Refresh on account 2 – reset on receiving the Nth Refresh on account
FIMI/ChangeAcctBalances/Rq/DropTmpOverOnDate	Time	In	Reset the temporary overdraft when receiving Refresh on the specified date. It is filled only if FIMI/ChangeAcctBalances/Rq/DropTmpOverOnRefresh=1 Available values: Value of Time type, -1 – set to NULL, NULL or undefined field – leave the previous value.
FIMI/ChangeAcctBalances/Rq/DropTmpOverNumber	Int	In	Number of account Refreshes in which temporary overdraft must be reset. It is filled only if FIMI/ChangeAcctBalances/Rq/DropTmpOverOnRefresh=2 Available values: Value of Int >= 1 type, -1 – set to NULL, NULL or undefined field – leave the previous value.
FIMI/ChangeAcctBalance/Rq/ChangeReason	Str(1000)	In	Account balances change reason
FIMI/ChangeAcctBalances/Rq/Force	Int(1)	In	=2 – the balance is decreased with the check of available balance and only from the own funds; =1 – the balance is decreased without the available balance check (in the forced mode); =0 or NULL – the balance is decreased with the check of account available balance
<b>AcctDebit – debiting account</b>			
FIMI/AcctDebit/Rq/InstName	Str(4)	In	Name of the financial institution owning the account. If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/AcctDebit/Rq/Account	Str(30)	In	Account number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<p>The field may be undefined if the search for the account by the card number is allowed for the terminal for the <i>AcctDebit/AcctCredit</i> operation and PAN or CardUID field is defined; the account number will be determined among the accounts linked to the card with the Open, Open primary account status of the account and link according to the following priority:</p> <ul style="list-style-type: none"> <li>- Account type (checking, savings, credit, bonus)</li> <li>- Link status (Open primary account, Open)</li> <li>- Account status (Open primary account, Open)</li> </ul>
FIMI/AcctDebit/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/AcctDebit/Rq/PAN	Str(19)	In	<p>Card PAN.</p> <p>For this operation, card attributes (PAN, MBR or CardUID fields) are optional.</p>
FIMI/AcctDebit/Rq/MBR	Int(3)	In	<p>If filled, the card will be saved in the transaction (the card must be linked to the account) and the notification will be generated on this card on a priority basis (the abonent must be created for the card)</p>
FIMI/AcctDebit/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AcctDebit/Rq/PersonId	Int(16)	In	Account owner ID
FIMI/AcctDebit/Rq/Amount	Num*	In	Operation amount (in the account currency) to be debited from the account in <b>TWO</b>
FIMI/AcctDebit/Rq/CorrespAcct	Str(30)	In	Corresponding account number
FIMI/AcctDebit/Rq/CorrespAcctUID	Str(32)	In	Unique identifier of the correspondent account within a session. It can stand for the account number
FIMI/AcctDebit/Rq/TranNumber	Str(50)	In	<p>Operation ID in the external system.</p> <p>If the field is filled, it overlays the value of the <i>FIMI/TransactionNumber</i> field from the request header.</p> <p>If this field or the <i>FIMI/TransactionNumber</i> field is filled, the system searches for the duplicates.</p>
FIMI/AcctDebit/Rq/OrigAmount	Num	In	Operation original amount
FIMI/AcctDebit/Rq/OrigCurrency	Int	In	Operation original currency

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AcctDebit/Rq/IgnoreImpact	Int(1)	In	=1 – operation is allowed in the Impact-period  =0 or NULL – operation is prohibited in the Impact-period
FIMI/AcctDebit/Rq/Force	Int(1)	In	=2 – account is debited after checking the account available balance and from the proprietary funds only.  =1 – account is debited without checking the account available balance (in the forced mode).  =0 or NULL – account is debited after checking the account available balance.  =3 – account is debited with the available balance check in the partial authorization mode.  =4 – account is debited with the available balance check and only from the own funds in the partial authorization mode.
FIMI/AcctDebit/Rq/ImpactProtectedAmount	Int(1)	In	Attribute indicating that it is required to impact the protected amount during the operation authorization:  0, NULL – do not impact. 1 – impact.  It is used when processing the operation in the authorization mode.
FIMI/AcctDebit/Rq/NeedNotify	Int(1)	In	=1 or NULL – notify a customer if notification on the cards linked to account is enabled  =0 – do not notify a customer even if notification on the card is enabled
FIMI/AcctDebit/Rq/PrevTranId	Int(12)	In	ID of the previous credit transaction; it is used to cancel the transaction;  If defined, a new debit transaction is linked to the specified credit one and the operation details are removed from the account history and not considered in the Impact period
FIMI/AcctDebit/Rq/RefreshSeqNo	Int(12)	In	Number of the last Refresh batch (in Back-Office) including account. If defined, it is used for the debit Impact during the Refresh.  The field is mandatory for online and offline exchange with Back-Office.
FIMI/AcctDebit/Rq/ChangeReason	Str(1000)	In	Account debit reason

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AcctDebit/Rq/UseBonusDebt	Int(1)	In	NULL or not 0 – consider bonus/debt when debiting the account 0 – do not consider bonus/debt when debiting the account
FIMI/AcctDebit/Rq/BonusProgramName	Str(20)	In	Bonus program name (see the list of available values in <i>FIMI/InitSession/Rp/BonusPrograms</i> ) The field is used if the <i>AcctDebit</i> operation is performed on the bonus account.
FIMI/AcctDebit/Rq/PrizeID	Str(40)	In	ID of the prize to be exchanged for bonuses The field is used if the <i>AcctDebit</i> operation is performed on the bonus account.
FIMI/AcctDebit/Rq/PrizeQuantity	Num	In	Number of prizes to be exchanged for bonuses The field is used if the <i>AcctDebit</i> operation is performed on the bonus account.
FIMI/AcctDebit/Rp/ApprovalCode	Str(6)*	Out	Approval code for the approved debit operation
FIMI/AcctDebit/Rp/Amount	Num	Out	Operation amount. If not NULL, the partial authorization was executed.
FIMI/AcctDebit/Rp/AvailBalance	Num*	Out	Value of the available balance of the source account in the account currency after the transaction authorization
FIMI/AcctDebit/Rp/LedgerBalance	Num*	Out	Value of the ledger balance of the source account in the account currency after the transaction authorization
FIMI/AcctDebit/Rp/ProtectedAmount	Num	Out	Absolute value of the protected amount after the operation execution. It is returned when processing the operation in the authorization mode.
FIMI/AcctDebit/Rp/ProtectedAmountDelta	Num	Out	Increment of the protected amount in the account currency. It is returned when processing the operation in the authorization mode.
<p>The description of the debit operation is defined in the field FIMI/Comment.</p> <p>ID of the debit transaction is returned in the field FIMI/TranId.</p> <p>Fields OrigAmount and OrigCurrency are of the informative character in TWO and transmitted to Back-Office in the Extract.</p> <p>If the <i>AcctDebit</i> operation is successfully executed, the <i>FIMI/Response</i> field can be =1 or =65.</p> <p>The full or partial reversal by means of the <i>ReverseTransaction</i> operation is available for this operation when authorizing the account balance change transaction.</p>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>AcctCredit – crediting account</b>			
FIMI/AcctCredit/Rq/InstName	Str(4)	In	Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/AcctCredit/Rq/Account	Str(30)	In	Account number  The field may be undefined if the search for the account by the card number is allowed for the terminal for the <i>AcctDebit/AcctCredit</i> operation and PAN or CardUID field is defined; the account number will be determined among the accounts linked to the card with the Open, Open primary account, Deposit only, Deposit only primary account status of the account and link according to the following priority: - Account type (checking, savings, credit, bonus) - Link status (Open primary account, Open, Deposit only primary account, Deposit only) - Account status (Open primary account, Open, Deposit only primary account, Deposit only)
FIMI/AcctCredit/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number.
FIMI/AcctCredit/Rq/PAN	Str(19)	In	Card PAN.  For this operation, card attributes (PAN, MBR or CardUID fields) are optional.  If filled, the card will be saved in the transaction (the card must be linked to the account) and the notification will be generated on this card on a priority basis (the abonent must be created for the card).
FIMI/AcctCredit/Rq/MBR	Int(3)	In	Card member number.  If =NULL, it =0.
FIMI/AcctCredit/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AcctCredit/Rq/PersonId	Int(16)	In	Account owner ID
FIMI/AcctCredit/Rq/Amount	Num*	In	Operation amount in the account currency to be credited in <b>TWO</b>
FIMI/AcctCredit/Rq/CorrespAcct	Str(30)	In	Corresponding account number
FIMI/AcctCredit/Rq/CorrespAcctUID	Str(32)	In	Unique identifier of the correspondent account within a session. It can stand for the account number

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AcctCredit/Rq/TranNumber	Str(50)	In	<p>Operation ID in the external system.</p> <p>If the field is filled, it overlays the value of the <i>FIMI/TransactionNumber</i> field from the request header.</p> <p>If this field or the <i>FIMI/TransactionNumber</i> field is filled, the system searches for the duplicates.</p>
FIMI/AcctCredit/Rq/OrigAmount	Num	In	Operation original amount
FIMI/AcctCredit/Rq/OrigCurrency	Int	In	Operation original currency
FIMI/AcctCredit/Rq/IgnoreImpact	Int(1)	In	<p>=1 – operation is allowed in the Impact-period</p> <p>=0 or NULL – operation is prohibited in the Impact-period</p>
FIMI/AcctCredit/Rq/NeedNotify	Int(1)	In	<p>=1 or NULL – notify a customer if notification on the cards linked to account is enabled.</p> <p>=0 – do not notify a customer even if notification on the cards is enabled.</p>
FIMI/AcctCredit/Rq/PrevTranId	Int(12)	In	<p>ID of the previous debit transaction; it used to cancel the transaction;</p> <p>If defined, a new credit transaction is linked to the debit one, the operation details are removed from the account history and not considered in the Impact period</p>
FIMI/AcctCredit/Rq/RefreshSeqNo	Int(12)	In	<p>Number of the last Refresh batch (in Back-Office) including account. If defined, it is used for the debit Impact during the Refresh.</p> <p>The field is mandatory under online and offline exchange with Back-Office.</p>
FIMI/AcctCredit/Rq/ChangeReason	Str(1000)	In	Account credit reason
FIMI/AcctCredit/Rq/UseBonusDebt	Int(1)	In	<p>NULL or not 0 – consider bonus/debt when crediting the account</p> <p>0 – do not consider bonus/debt when crediting the account,</p>
FIMI/AcctCredit/Rq/BonusProgramName	Str(20)	In	<p>Bonus program name (see the list of available values in <i>FIMI/InitSession/Rp/BonusPrograms</i>)</p> <p>The field is used if the <i>AcctCredit</i> operation is performed on the bonus account.</p>
FIMI/AcctCredit/Rq/PrizeID	Str(40)	In	<p>ID of the prize to be exchanged for bonuses</p> <p>The field is used if the <i>AcctCredit</i> operation is performed on the bonus account.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AcctCredit/Rq/PrizeQuantity	Num	In	Number of prizes to be exchanged for bonuses  The field is used if the <i>AcctCredit</i> operation is performed on the bonus account.
FIMI/AcctCredit/Rp/ApprovalCode	Str(6)*	Out	Approval code for the approved credit operation
FIMI/AcctCredit/Rp/AvailBalance	Num*	Out	Value of the available balance of the source account in the account currency by the time after the transaction authorization
FIMI/AcctCredit/Rp/LedgerBalance	Num*	Out	Value of the ledger balance of the source account in the account currency after the transaction authorization

The *FIMI/Comment* field contains the description of the account credit operation.

The *FIMI/TranId* field returns ID of the received credit transaction.

The *OrigAmount* and *OrigCurrency* fields are used in **TWO** as informational and transferred to Back-Office by means of *Extract* procedure

The full or partial reversal by means of the *ReverseTransaction* operation is available for this operation when authorizing the account balance change transaction.

#### ***SetAccountOverdraft – account overdraft change***

FIMI/SetAccountOverdraft/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the FIMI terminal owner institution stands for the account intuition in <b>TWO</b>
FIMI/SetAccountOverdraft/Rq/Account	Str(30)	In	Account number
FIMI/SetAccountOverdraft/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can be specified instead of the account number
FIMI/SetAccountOverdraft/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetAccountOverdraft/Rq/OverdraftDelta	Num	In	Increment of the allowed overdraft in the account currency. This value is added to the current allowed overdraft of the account. The increment can be both positive and negative. The overdraft total amount cannot be less than 0. If the field is not filled, the allowed overdraft is not changed.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetAccountOverdraft/Rq/Overdraft	Num	In	Absolute value of the allowed overdraft in the account currency.  If the field is transferred, the OverdraftDelta field value is ignored and increment is calculated automatically on the basis of the field value and current account overdraft in order the account overdraft is equal to the transferred value after the change. The value can exceed or equal 0.
FIMI/SetAccountOverdraft/Rq/TmpOverdraftDelta	Num	In	Increment of the temporary overdraft in the account currency. This value is added to the current temporary overdraft of the account. The increment can be both positive and negative. If the field is not filled, the temporary overdraft is not changed.  Attention! To activate the temporary overdraft, the <i>TmpOverdraftExpiration</i> field must contain the valid expiration date.
FIMI/SetAccountOverdraft/Rq/TmpOverdraft	Num	In	Absolute value of the temporary overdraft in the account currency.  If the field is transferred, the TmpOverdraftDelta field value is ignored and increment is calculated automatically on the basis of the field value and current temporary overdraft of the account in order the temporary overdraft is equal to the transferred value after the change.
FIMI/SetAccountOverdraft/Rq/TmpOverdraftExpiration	Time	In	Expiration date and time of the temporary overdraft. This field is defined together with the <i>TmpOverdraftDelta</i> field. If the value in the <i>TmpOverdraftExpiration</i> field is not defined, this account attribute won't be changed.
FIMI/ SetAccountOverdraft /Rq/TmpOverdraftPeriodType	Int(1)	In	Temporary overdraft period type. Available values:  1 – overdraft per one operation 0 – overdraft per period NULL – leave the previous value
FIMI/SetAccountOverdraft/Rq/DropTmpOverOnRefresh	Int(1)	In	Reset temporary overdraft when receiving Refresh:  0 – do not reset 1 – reset when receiving the next Refresh on the account 2 – reset when receiving the N-th Refresh on the account; NULL – use the previous value  It can be used when authorizing the <i>SetAccountOverdraft</i> operation.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetAccountOverdraft/Rq/DropTmpOverOnDate	Time	In	<p>Reset temporary overdraft when receiving Refresh on the specified date. It is filled only if FIMI/SetAccountOverdraft/Rq/DropTmpOverOnRefresh=1. Available values:</p> <p>Value of the Time type,            -1 – reset to NULL,            NULL – use the previous value</p> <p>It can be used when authorizing the SetAccountOverdraft operation.</p>
FIMI/SetAccountOverdraft/Rq/DropTmpOverNumber	Int	In	<p>Number of Refresh procedures on the account during which the temporary overdraft must be reset. It is filled only if FIMI/SetAccountOverdraft/Rq/DropTmpOverOnRefresh=2. Available values:</p> <p>Value of the Int type &gt;= 1,            -1 – reset to NULL,            NULL – use the previous value.</p> <p>It can be used when authorizing the SetAccountOverdraft operation.</p>
FIMI/SetAccountOverdraft/Rq/IgnoreImpact	Int(1)	In	<p>=1 – operation is allowed during the Impact period            =0 or NULL – operation is not allowed during the Impact period</p>
FIMI/SetAccountOverdraft/Rq/RefreshSeqNo	Int(12)	In	<p>Number of the Refresh batch that includes the account in Back-Office; if the field is filled, it is used for the credit Impact when executing account Refresh.</p> <p>The field is defined if the online and offline exchange with Back-Office is used.</p>
FIMI/SetAccountOverdraft/Rq/ChangeReason	Str(1000)	In	Account overdraft change reason
FIMI/SetAccountOverdraft/Rq/NeedNotify	Int(1)	In	<p>=1 or NULL – notify customer if the notification is enabled for the cards linked to the account            =0 – do not notify customer even if the notification is enabled for the cards linked to the account</p>
FIMI/SetAccountOverdraft/Rp/ApprovalCode	Str(6)*	Out	Approval code for the approved operation of the account overdraft change
The FIMI/Comment field contains the description of the account overdraft change operation.			
The FIMI/TranId field returns the ID of the received transaction of account overdraft change.			
<b>CreateAccount – creating account</b>			
FIMI/CreateAccount/Rq/Account	Str(30)*	In	Account number

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/CreateAccount/Rq/BranchId	Str(90)	In	Account branch ID
FIMI/CreateAccount/Rq/InstName	Str(30)	In	Character ID (name) of the financial institution owning the account. If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/CreateAccount/Rq/Type	Int(2)*	In	Account type in TWO: 1 – checking 11 – savings 31 – credit 91 – bonus
FIMI/CreateAccount /Rq/Status	Int(1)*	In	Account status: 0 – Inactive account 1 – Open 2 – Deposit only 3 – Open primary account 4 – Deposit only primary account 5 – Information only 9 – Closed
FIMI/CreateAccount /Rq/Currency	Int*	In	Account currency ISO code
FIMI/CreateAccount /Rq/ LedgerBalance	Num*	In	Ledger balance
FIMI/CreateAccount /Rq/ AvailBalance	Num*	In	Available balance
FIMI/CreateAccount/Rq/ ExtendedAccountNumber	Str(30)	In	Extended account number
FIMI/CreateAccount /Rq/PersonID	Int(16)	In	Customer ID

The field FIMI/Comment allows to transfer any information for creating an account in Back-Office in the format <field1>~<field2>, i.e. several fields are ‘~’-separated. The value will be saved in the transaction and posted to Extract for further processing in Back-Office.

If the *CreateAccount* operation is to be posted to Back-Office for further processing, the account type in Back-Office and Branch ID (if the BranchId field is not used) should be specified in the field FIMI/Comment between ‘~’ separator. If <field2> is filled or the BranchId field is transferred, the account will be created with the specified Branch ID.

#### **GetAcctLimits – request for list of account max and current limits**

FIMI/GetAcctLimits/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the account If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO. For the list of available values, see the field <i>InitSession/Rp/AllowedFIID.Name</i> – it is allowed to request an account for the
--------------------------------	--------	----	--

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
			institution with <i>AllowedFIID.AccessMask.Bit[1]=1</i> or <i>AllowedFIID.AccessMask.Bit[3]=1</i>
FIMI/GetAcctLimits/Rq/Account	Str(30)	In	Account number
FIMI/GetAcctLimits/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/GetAcctLimits/Rq/PersonId	Int(16)	In	Customer ID
FIMI/GetAcctLimits/Rq/GetCurrentCounters	Int(1)	In	if NULL then = 0 if it = 1, the current values of counters are returned in the response, in the fields <i>Limits[i].Current</i> ; at that, the <i>AuthScheme</i> field should be defined in the request
FIMI/GetAcctLimits/Rq/AuthScheme	Int	In	Authorization scheme ID; the field should be defined if <i>GetCurrentCounters=1</i> ; It is used to request the values of the group limit assigned to authorization scheme account. If NULL, the specific values of the group limit are not provided; For the list of available values, see the field <i>InitSession/Rp/AuthSchemes</i>
FIMI/GetAcctLimits/Rp/Limits	ArrRec	Out	<pre>{     int* LimitId;     num Max;     num* Current;     int(2) PeriodType;     int Period;     num TmpMax;     time TmpMaxExpiration;     num TmpMaxRange;     int(1) TmpMaxPeriodType;     time DateTimeForReset; }</pre> <p>} [] – array of structures, List of the account max and current limits:</p>
<p>LimitId – limit ID in TWO (see p. 3.3).          Max – max limit value in the account currency (0-restricted, NULL - unlimited, &gt;0 - defined), -1 – as auth. scheme value).          Current – current limit value.          PeriodType, Period – limit validity period (see the description in the <i>SetCardLimits</i> message).          TmpMax – maximum value of the temporary limit (0-restricted, NULL – not defined, &gt;0 - defined).          TmpMaxExpiration – validity period of the temporary maximum limit (NULL- until the next reset of the limit counter).          TmpMaxRange – the value restricting the TmpMax parameter range (0-restricted, NULL-not defined, &gt;0 defined).          TmpMaxPeriodType – type of the limit temporary maximum period (NULL – for a period; 1 – per a single operation).          DateTimeForReset – calendar date and time of the limit current value reset.</p>			
FIMI/GetAcctLimits/Rp/LimitsRanges	ArrRec	Out	<pre>{     int* LimitId;     time* StartDate;     time EndDate;     num Max; }</pre> <p>} [] – array of structures, list of limit ranges returned in the <i>Limits</i> structure:</p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
LimitId – limit ID in <b>TWO</b> (see p. 3.3). StartDate – start date of the range. EndDate – end date of the range (NULL – range is unlimited). Max – maximum value of the limit in the range (NULL – as maximum limit value).			
<b>SetAcctLimits – changing account max limits and their validity periods</b>			
FIMI/SetAcctLimits/Rq/InstName	Str(4)	In	Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/SetAcctLimits/Rq/Account	Str(30)	In	Account number
FIMI/SetAcctLimits/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number.
FIMI/SetAcctLimits/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetAcctLimits/Rq/RemoveMissingLimits	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing limits. 2 – remove received limits. If NULL, it is 0.
FIMI/SetAcctLimits/Rq/Limits	ArrRec*	In	{ int LimitId; num Max; int(2) PeriodType; int Period; time DateTimeForReset; } [] – array structures;  List of the account max limits:
LimitId – limit ID in <b>TWO</b> (see p.3.3). Max – limit max in the account currency (0-restricted, NULL - unlimited, >0 - defined), -1 - as auth. scheme value). PeriodType, Period, DateTimeForReset – limit period of validity (for the description, see <i>SetCardLimits</i> ).			
FIMI/SetAcctLimits/Rq/ChangeReason	Str(1000)	In	Reason for changing the account limit maximums and their periods of validity
<b>SetAcctLimitsRanges – setting account limit ranges</b>			
FIMI/SetAcctLimitsRanges/Rq/InstName	Str(4)	In	Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/SetAcctLimitsRanges/Rq/Account	Str(30)	In	Account number
FIMI/SetAcctLimitsRanges/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetAcctLimitsRanges/Rq/ LimitId	Int*	In	Limit ID in TWO (see p. 3.3)
FIMI/SetAcctLimitsRanges/Rq/ RemoveMissing	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing ranges. 2 – remove received ranges. If NULL, it is 0.
FIMI/SetAcctLimitsRanges/Rq/ LimitsRanges	ArrRec	In	{ time StartDate; time EndDate; num Max; } [] – array of structures, list of the limit ranges:  StartDate – start date of the range (NULL – starting from the operation execution). EndDate – end date of the range (NULL – range is unlimited). Max – maximum value of the limit in the range (NULL – as maximum limit value).
FIMI/SetAcctLimitsRanges/Rq/ ChangeReason	Str(1000)	In	Reason for changing the account limit maximums and their periods of validity
<b>GetAcctGroupLimits – request for limit values by scheme level accounts</b>			
FIMI/GetAcctGroupLimits/Rq/ AuthScheme	Int*	In	ID of the authorization scheme  See a list of available values in the <i>InitSession/Rp/AuthSchemes</i> field
FIMI/GetAcctGroupLimits/Rp/ Limits	ArrRec	Out	{ int* LimitId; int* Currency; num Max; int(2) PeriodType; int Period; time DateTimeForReset; } [] – array of structures,  List of limits with the maximum values and validity periods (see the description of limit validity periods in the <i>SetCardLimits</i> operation)
<b>ResetAcctLimitsCounter – reset of account limits counter</b>			
FIMI/ResetAcctLimitsCounter/Rq/ InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/ResetAcctLimitsCounter/Rq/ Account	Str(30)	In	Account number
FIMI/ResetAcctLimitsCounter/Rq/ AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ResetAcctLimitsCounter/Rq/Limits	Int[]*	In	{ int LimitId } [] – array of structures, Identifiers if limits whose counters are reset.
FIMI/ResetAcctLimitsCounter/Rq/ChangeReason	Str(1000)	In	Reason for the limits counter reset
<b><i>SetTmpAcctLimits – setting account temporary limit maximums</i></b>			
FIMI/SetTmpAcctLimits/Rq/InstName	Str(4)	In	Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/SetTmpAcctLimits/Rq/Account	Str(30)	In	Account number
FIMI/SetTmpAcctLimits/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/SetTmpAcctLimits/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetTmpAcctLimits/Rq/Limits	ArrRec*	In	{ int* LimitId; num TmpMax; time TmpMaxExpiration; int TmpMaxPeriodType; } } [] – array of structures, List of the account temporary max limits:
 LimitId – limit ID in TWO (see p. 3.3). TmpMax – temporary max limits (it is reset when the limit expires or when the time is reached). TmpMaxExpiration – expiration date of the temporary maximum (NULL- until the next reset of the limit counter). TmpMaxPeriodType – type of the limit temporary maximum period (NULL – for a period; 1 – per a single operation).			
FIMI/SetTmpAcctLimits/Rq/ChangeReason	Str(1000)	In	Reason for changing the account temporary limit maximums
<b><i>SetAccountUserFields – setting the account user fields</i></b>			
FIMI/SetAccountUserFields/Rq/InstName	Str(4)	In	Name of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/SetAccountUserFields/Rq/Account	Str(30)	In	Account number
FIMI/SetAccountUserFields/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/SetAccountUserFields/Rq/PersonId	Int(16)	In	Customer ID

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetAccountUserFields/Rq/UserFields	ArrRec	In	<pre>{     Str(100)* Name;     Str(3800) TextValue;     StrXML XMLValue. }</pre> <p>[]} – array of structures</p>
<p>The total length of the <i>Name</i> and <i>TextValue</i> fields in all the records of the array <i>FIMI/SetAccountUserFields/Rq/UserFields</i> must not exceed 3900 symbols.</p> <p>If the <i>TextValue</i> and <i>XMLValue</i> fields are absent in the structure and account has the user field, this user field will not be changed. To reset the <i>TextValue</i> and <i>XMLValue</i> values, specify empty values of the <i>TextValue</i> and <i>XMLValue</i> fields.</p>			
FIMI/SetAccountUserFields/Rq/RemoveMissingFields	Int(1)	In	<p>1 – remove missing fields 0 – by default</p> <p>If FIMI/InitSession/Rp/LimitedAccessToUserFields = 1, only the user fields available for editing will be deleted</p>
FIMI/SetAccountUserFields/Rq/NeedNotify	Int(1)	In	<p>=1 – notify a customer if the notification is enabled =0 or NULL – do not notify a customer even if the notification is enabled</p>
<b><i>SetAccountPerson – changing account owner</i></b>			
FIMI/SetAccountPerson/Rq/InstName	Str(4)	In	<p>Name of the financial institution owning the account. If not defined, the FIMI terminal owner institution stands for the account institution in <b>TWO</b>.</p>
FIMI/SetAccountPerson/Rq/Account	Str(30)	In	Account number
FIMI/SetAccountPerson/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can be specified instead of the account number.
FIMI/SetAccountPerson/Rq/NewPersonId	Int(16)*	In	New ID of customer.
FIMI/SetAccountPerson/Rq/ChangeContractPerson	Int(1)	In	'Change Contract Owner' attribute. If =1, the owner is changed to <i>NewPersonId</i> for the contract linked to <i>Account</i>
FIMI/SetAccountPerson/Rq/ChangeReason	Str(1000)	In	Change reason
<p>The <i>ChangeContractPerson</i> field not saved in <b>TWO</b>; it is only used to transfer the attribute for transactions offline exchange between <b>TWO</b> and <b>TWCMS</b>.</p>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b><i>BalanceInquiry – request for customer account balances</i></b>			
FIMI/BalanceInquiry/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/BalanceInquiry/Rq/Account	Str(30)	In	Account number
FIMI/BalanceInquiry/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/BalanceInquiry/Rq/PAN	Str(19)	In	Account-linked card number.  If not defined, the terminal will search for the card linked to the account
FIMI/BalanceInquiry/Rq/MBR	Int(3)	In	Account-linked card member number
FIMI/BalanceInquiry/Rq/CardUID	Ctr(64)	In	Unique identifier of the account-linked card
FIMI/BalanceInquiry/Rq/PersonId	Int(16)	In	Customer ID. It can be used for the definite account search if masking is enabled.
FIMI/BalanceInquiry/Rq/Currency	Int	In	Balances currency.  NULL – requesting for balances in the terminal currency
FIMI/BalanceInquiry/Rp/Ledger	Num*	Out	Ledger balance
FIMI/BalanceInquiry/Rp/Avail	Num*	Out	Available balance. If permitted in the host settings, it includes the allowed overdrafts.
FIMI/BalanceInquiry/Rp/Currency	Int*	Out	Balances currency
FIMI/BalanceInquiry/Rp/OverdraftOn	Int(1)*	Out	Indicates whether overdraft is included in the available balance
The <i>BalanceInquiry</i> transaction is authorized at the Stand-In or TCI host.			
<b><i>DeleteAccount – deleting account</i></b>			
FIMI/DeleteAccount/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/DeleteAccount/Rq/Account	Str(30)	In	Account number
FIMI/DeleteAccount/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/DeleteAccount/Rq/Force	Int(1)	In	=NULL or 0 – the check is performed; if transactions (except for the Create FIMI transaction) are found for the account in the Online Transaction log, the operation is declined  =1 – account is deleted without the check of whether the transactions on the account are present
<b>GetAcctCounters – getting account counters</b>			
FIMI/GetAcctCounters/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/GetAcctCounters/Rq/Account	Str(30)	In	Account number
FIMI/GetAcctCounters/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number.
FIMI/GetAcctCounters/Rq/AuthScheme	Int	In	Authorization scheme ID.  If defined, the counter parameters are returned considering the authorization scheme parameters. See the list of available values in <i>InitSession/Rp/AuthSchemes</i> .
FIMI/GetAcctCounters/Rp/Counters	ArrRec	Out	{ int* CounterId; num* MaxValue; num* Current; int(2)* PeriodType; time DateTimeForReset; time LastResetTime; int Period } [] – list of account counters:  CounterId – counter ID in <b>TWO</b> (see the list of available values in <i>FIMI/InitSession/Rp/Counters</i> ). MaxValue – maximum counter value (available values >=0, -1 – “as the scheme”, -2 – “unlimited”). Current – current counter value. LastResetTime – time of the last reset. PeriodType, DateTimeForReset, Period – period type and value (see the description in <i>GetCardCounters</i> ).
<b>SetAcctCounters – changing account counters</b>			
FIMI/SetAcctCounters/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in <b>TWO</b> .

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetAcctCounters/Rq/Account	Str(30)	In	Account number
FIMI/SetAcctCounters/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number.
FIMI/SetAcctCounters/Rq/RemoveMissing	Int(1)	In	=1 – remove missing counters By default = 0
FIMI/SetAcctCounters/Rq/Counters	ArrRec	In	{ int* CounterId; num.MaxValue; num.CurrentDelta; int(2).PeriodType; time.DateTimeForReset; int.Period } [] – list of account counters:
CounterId – counter ID in TWO (see the list of available values <i>FIMI/InitSession/Rp/Counters</i> ). .MaxValue – maximum counter value (available values >=0, -1 – “as the scheme”, -2 – “unlimited”). CurrentDelta – positive or negative delta of the counter value change, if 0, the counter is reset. PeriodType, DateTimeForReset, Period – period type and value (see the description in <i>GetCardCounters</i> ). If MaxValue, CurrentDelta, PeriodType = NULL, the impact on the respective attribute is not performed.			
FIMI/SetAcctCounters/Rq/ChangeReason	Str(1000)	In	Reason for the account counters change
<b>SetAccountPermissibleExcess – setting allowed excess of available account balance</b>			
FIMI/SetAccountPermissibleExcess/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in TWO.
FIMI/SetAccountPermissibleExcess/Rq/Account	Str(30)	In	Account number
FIMI/SetAccountPermissibleExcess/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number.
FIMI/SetAccountPermissibleExcess/PermissibleExcessType	Int(1)*	In	Type of the allowed excess of the available account balance: -1 – not used. 0 – percent of the balance amount. 1 – excess in the account currency.
FIMI/SetAccountPermissibleExcess/Rq/PermissibleExcess	Num*	In	Value of the allowed excess of the available account balance in percents or amount. If PermissibleExcessType=0, it contains percents in the range of 0.001 to 99.999. If PermissibleExcessType=1, it contains the amount in the account currency.
FIMI/SetAccountPermissibleExcess/Rq/ChangeReason	Str(1000)	In	Change reason

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>SetAcctProtectedAmount – setting protected account amount</b>			
FIMI/SetAcctProtectedAmount/Rq/InstName	Str(4)	In	<p>Name of the financial institution owning the account.</p> <p>If not defined, the institution owning the FIMI terminal stands for the account institution in TWO.</p>
FIMI/SetAcctProtectedAmount/Rq/Account	Str(30)	In	Account number
FIMI/SetAcctProtectedAmount/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.
FIMI/SetAcctProtectedAmount/Rq/ClearProtectedAmount	Int(1)	In	<p>Attribute indicating that keeping of the protected amount is disabled.</p> <p>1 – disable keeping of the protected amount and clear the value, the values transferred in the <i>ProtectedAmountDelta</i> and <i>ProtectedAmount</i> fields are ignored.</p> <p>0, NULL – activate or change, impact the protected amount within the <i>ProtectedAmountDelta</i> or <i>ProtectedAmount</i> fields.</p>
FIMI/SetAcctProtectedAmount/Rq/ProtectedAmountDelta	Num	In	Increment of the protected amount in the account currency
This value is added arithmetically to the current value of the protected amount of the account. Both positive and negative increment values can be specified.			
It is used if the field with the absolute ProtectedAmount value is not transferred.			
If the ProtectedAmountDelta value is negative (protected amount is decreased) and the protected amount becomes less than 0 according to the processing results, the maximum possible increment is applied for the value to become 0, the applied value returns in the ProtectedAmountDelta response field.			
If the ProtectedAmountDelta value is negative (protected amount is decreased) and the protected amount value is negative at the time of execution, no impact is performed, value 0 is transferred in the ProtectedAmountDelta response field.			
FIMI/SetAcctProtectedAmount/Rq/ProtectedAmount	Num	In	<p>Absolute value of the protected amount.</p> <p>Available value is &gt; 0, if =0, the amount must be reset to 0.</p>
FIMI/SetAcctProtectedAmount/Rq/ChangeReason	Str(1000)	In	Change reason
FIMI/SetAcctProtectedAmount/Rp/ProtectedAmount	Num	Out	Absolute value of the protected amount after the operation execution
FIMI/SetAcctProtectedAmount/Rp/ProtectedAmountDelta	Num	Out	Increment of the protected amount in the account currency

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetAcctTokensInfo – request for information on account tokens from TWO storage</b>			
FIMI/GetAcctTokensInfo/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in TWO.
FIMI/GetAcctTokensInfo/Rq/Account	Str(30)	In	Account number
FIMI/GetAcctTokensInfo/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.
FIMI/GetAcctTokensInfo/Rq/ServiceId	Int(1)*	In	Service ID: 1 – SBPay
FIMI/GetAcctTokensInfo/Rq/Token	Str(100)	In	Token  If not defined, it can be any.
FIMI/GetAcctTokensInfo/Rq/TokenRequestorId	Str(100)	In	Unique identifier of the object that initiated the tokenization process.  If not defined, it can be any.
The search for tokens can be performed by the account (if the Account or AccountUID field is defined). The value in one of the fields (Account or AccountUID) is mandatory.			
FIMI/GetAcctTokensInfo/Rp/Tokens	ArrRec	Out	{ Str(30)* Account; Str(32) AccountUID; Str(100)* Token; Str(100)* TokenRequestorId; Str(100)* TokenIntentId; Str(30) ExtendedAccountNumber; Int(1)* Status; Str (20) PhoneNumber; Int(12) LastTranId; Time* CreationTime; Time LastChangeTime; } [] – list of the account tokens:
Account – account number. AccountUID – unique account identifier within the session. Token – token value. TokenRequestorId – unique identifier of the object that initiated the tokenization process. TokenIntentId – identifier of the token creation request. ExtendedAccountNumber – extended account number. Status – token status, available values: 0 – new. 1 – authorized. 2 – active. 3 – suspended. 4 – blocked. 5 – being deleted. 6 – deleted. PhoneNumber – phone number. LastTranId – identifier of the last token transaction.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
CreationTime – token creation date. ChangeTime – date of the last token change.			
<b>SetAcctTokenStatus – changing account token status</b>			
FIMI/SetAcctTokenStatus/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/SetAcctTokenStatus/Rq/Account	Str(30)	In	Account number
FIMI/SetAcctTokenStatus/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.
FIMI/SetAcctTokenStatus/Rq/ServiceId	Int(1)*	In	Service ID: 1 – SBPay
FIMI/SetAcctTokenStatus/Rq/Token	Str(100)*	In	Token
FIMI/SetAcctTokenStatus/Rq/TokenRequestorId	Str(100)*	In	Unique identifier of the object that initiated the tokenization process.
FIMI/SetAcctTokenStatus/Rq/Status	Int(2)*	In	New token status  Available values: 2 – Active. 3 – Suspended. 4 – Blocked. 6 – Deleted.  Available status transitions: “Active” → “Suspended”; “Active” → “Blocked”; “Active” → “Deleted”; “Suspended” → “Active”; “Suspended” → “Blocked”; “Suspended” → “Deleted”; “Blocked” → “Deleted”, “Being deleted” → “Deleted”.
FIMI/SetAcctTokenStatus/Rq/ChangeReason	Str(1000)	In	Token status change reason
<b>CreateAcctToken – creating account token in TWO storage</b>			
FIMI/CreateAcctToken/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/CreateAcctToken/Rq/Account	Str(30)	In	Account number
FIMI/CreateAcctToken/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateAcctToken/Rq/ServiceId	Int(1)*	In	Service ID: 1 – SBPay
FIMI/CreateAcctToken/Rq/TokenIntentId	Str(100)*	In	Identifier if the token creation request
FIMI/CreateAcctToken/Rq/ExtendedAccountNumber	Str(30)	In	Extended account number If NULL, it is identified by the account data in TWO.
FIMI/CreateAcctToken/Rp/Token	Str(100)*	Out	Token
<b>SetAcctLegalEntity – changing corporate customer of account</b>			
FIMI/SetAcctLegalEntity/Rq/InstName	Str(4)	In	Name of the financial institution owning the account.  If not defined, the institution owning the FIMI terminal stands for the account institution in TWO.
FIMI/SetAcctLegalEntity/Rq/Account	Str(30)	In	Account number
FIMI/SetAcctLegalEntity/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.
FIMI/SetAcctLegalEntity/Rq/NewLegalEntityId	Int(16)*	In	New identifier of a corporate customer -1 – not defined. >0 – corporate customer identifier (list of available values can be obtained using the GetLegalEntityInfo operation).
FIMI/SetAccountPerson/Rq/ChangeReason	Str(1000)	In	Change reason

### 2.3.3.3 On Card-to-Account Link:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>SetAcct2CardStatus – setting account status for card-to-account link</b>			
FIMI/SetAcct2CardStatus/Rq/PAN	Str(19)	In	Card number
FIMI/SetAcct2CardStatus/Rq/MBR	Int(3)	In	Card member number
FIMI/SetAcct2CardStatus/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetAcct2CardStatus/Rq/Account	Str(30)	In	Number of the card linked account. It is obtained from GetCardInfo/Rp/Accounts.AcctNo.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetAcct2CardStatus/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.
FIMI/SetAcct2CardStatus/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetAcct2CardStatus/Rq/Status	Int(1)*	In	New account status in the card-to-account link:  0 – Inactive account 1 – Open 2 – Deposit only 3 – Open primary account 4 – Deposit only primary account; 5 – Information only 9 – Closed
FIMI/SetAcct2CardStatus/Rq/ChangeReason	Str(1000)	In	Reason for changing the account status for the card-to-account link
To determine the card-to-account link, the account number must be transferred in the Account field (or AccountUID if the UID accounts are used) and card number must be transferred in the PAN field (or CardUID if the UID cards are used).			
<b>GetAcct2CardLimits – request for list of account-to-card link limits</b>			
FIMI/GetAcct2CardLimits/Rq/PAN	Str(19)	In	Card number
FIMI/GetAcct2CardLimits/Rq/MBR	Int(3)	In	Card member number
FIMI/GetAcct2CardLimits/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetAcct2CardLimits/Rq/Account	Str(30)	In	Account number
FIMI/GetAcct2CardLimits/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.
FIMI/GetAcct2CardLimits/Rq/PersonId	Int(16)	In	Customer ID
FIMI/GetAcct2CardLimits/Rq/GetCurrentCounters	Int(1)	In	If NULL then = 0  If it = 1, the counter current values will be returned in the response, in the field <i>Limits[i].Current</i> ; at that, the <i>AuthScheme</i> field should be defined in the request
FIMI/GetAcct2CardLimits/Rq/AuthScheme	Int	In	Authorization scheme ID; the field should be defined if <i>GetCurrentCounters</i> =1.  For the list of available values, see the field <i>InitSession/Rp/AuthSchemes</i> .
To determine the card-to-account link, the account number must be transferred in the Account field (or AccountUID if the UID accounts are used) and card number must be transferred in the PAN field (or CardUID if the UID cards are used).			

<b>Parameter</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
FIMI/GetAcct2CardLimits/Rp/Limits	ArrRec	Out	<pre>{     int* LimitId;     num Max;     num* Current;     int(2) PeriodType;     int Period;     time DateTimeForReset; }</pre> <p>} [] – array of structures,  List of the account max and current limits:  LimitId – limit ID in <b>TWO</b>  (see p. 3.3);  Max – max limit in the account currency  (0-restricted, NULL - unlimited, &gt;0 - defined);  Current – current limit;  PeriodType, Period, DateTimeForReset – limit period of validity (for details, refer to the <i>SetCardLimits</i> message)</p>
<b><i>SetAcct2CardLimits</i> – changing account-to-card link limit maximums and their validity periods</b>			
FIMI/SetAcct2CardLimits/Rq/PAN	Str(19)	In	Card number
FIMI/SetAcct2CardLimits/Rq/MBR	Int(3)	In	Card member number
FIMI/SetAcct2CardLimits/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetAcct2CardLimits/Rq/Account	Str(30)	In	Account number
FIMI/SetAcct2CardLimits/Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number.
FIMI/SetAcct2CardLimits/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetAcct2CardLimits/Rq/RemoveMissingLimits	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing limits. 2 – remove received limits. If NULL, it is 0.
FIMI/SetAcct2CardLimits/Rq/Limits	ArrRec	In	<pre>{     int LimitId;     num Max;     int(2) PeriodType;     int Period;     time DateTimeForReset; }</pre> <p>} [] – array of structures,  List of the account max limits:</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
LimitId – limit ID in <b>TWO</b> (see p. 3.3). Max – limit max in the account currency (0-restricted, NULL - unlimited, >0 - defined). PeriodType, Period, DateTimeForReset – limit period of validity (for the description, see <i>SetCardLimits</i> ). To determine the card-to-account link, the account number must be transferred in the Account field (or AccountUID if the UID accounts are used) and card number must be transferred in the PAN field (or CardUID if the UID cards are used).			
<b>ResetAcct2CardLimitsCounter – reset of account-card link limits counter</b>			
FIMI/ ResetAcct2CardLimitsCounter/ Rq/PAN	Str(19)	In	Card number
FIMI/ ResetAcct2CardLimitsCounter/ Rq/MBR	Int(3)	In	Cardholder number
FIMI/ ResetAcct2CardLimitsCounter/ Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/ ResetAcct2CardLimitsCounter/ Rq/InstName	Str(4)	In	Name of the financial institution owning the account  If not defined, the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/ ResetAcct2CardLimitsCounter/ Rq/Account	Str(30)	In	Account number
FIMI/ ResetAcct2CardLimitsCounter/ Rq/AccountUID	Str(32)	In	Unique account identifier within a session. It can stand for the account number
FIMI/ ResetAcct2CardLimitsCounter/ Rq/Limits	Int[]*	In	Identifiers of limits whose counters are reset
FIMI/ ResetAcct2CardLimitsCounter/ Rq/ChangeReason	Str(1000)	In	Reason for limits counter reset
To determine the card-to-account link, the account number must be transferred in the Account field (or AccountUID if the UID accounts are used) and card number must be transferred in the PAN field (or CardUID if the UID cards are used).			

## 2.3.3.4 On Customer:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetPersonInfo – request for customer personal information</b>			
FIMI/GetPersonInfo/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution where the customer is created</p> <p>If not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in <b>TWO</b>.</p> <p><i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name – it is allowed to request an account for the institution with AllowedFIID.AccessMask.Bit[1]=1 or AllowedFIID.AccessMask.Bit[3]=1</i></p>
FIMI/GetPersonInfo/Rq/PersonId	Int(16)	In	Customer ID
FIMI/GetPersonInfo/Rq/FIO	Str(300)	In	Full name (or the first letters) is not case sensitive
FIMI/GetPersonInfo/Rq/IdentType	Int(1)	In	<p>Identifier type – type of the number of the document identifying the customer</p> <p><i>For the list of available values, see the field GetPersonInfo/Rp/Info</i></p>
FIMI/GetPersonInfo/Rq/Identity	Str(120)	In	Identifier – number of the identity paper
FIMI/GetPersonInfo/Rq/Birthday	Time	In	Birthday
FIMI/GetPersonInfo/Rq/PersonExtId	Str(50)	In	External customer ID
FIMI/GetPersonInfo/Rq/AbonentAddress	Str(500)	In	<p>Abonent address for the messaging service.</p> <p>The search by address is performed among all abonents on cards linked to customers and customer abonents.</p>
<p><b>Note:</b></p> <p>In the <i>FIO</i>, <i>Identity</i>, <i>AbonentAddress</i>, <i>PersonExtId</i> parameters, the mask can be defined in the format of the SQL operator 'like', i.e. applying the metacharacters '_' (any single character), '%' (any number of any characters).</p> <p>When searching for the customer, the request field values are analyzed and the following search strategy is created:</p> <ul style="list-style-type: none"> <li>- search by the customer ID if the <i>Id</i> field is defined;</li> <li>- search by the external customer ID if the <i>ExtId</i> field is defined;</li> <li>- search by the document type and number if the <i>Identity</i> and <i>IdentType</i> fields are defined (the <i>Birthday</i> field value is considered during the search if the value is set)</li> <li>- search by the address of the messaging service abonent if the <i>AbonentAddress</i> field is defined (the <i>FIO</i>, <i>Identity</i> or <i>IdentType</i>, <i>Birthday</i> field values are considered during the search if the values are set);</li> <li>- search by the customer last name, first name and middle name if the <i>FIO</i> field is defined (the <i>Identity</i> or <i>IdentType</i>, <i>Birthday</i> fields values are considered during the search if the values are set).</li> </ul> <p>In case, the suitable search strategy is not selected, the request will be declined.</p>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetPersonInfo/Rp/Info	ArrRec	Out	<pre>{     Int(16)* PersonId;     str(300) FIO;     str(1)* Sex;     int(1) IdentType;     str(120) Identity;     str(60) SecretWord;     time Birthday;     str(250) Birthplace;     int(1) VIP;     str(4) InstName;     str(50) PersonExtId;     int ResidentCountry;     str(100) ResidentCityInLatin;     str(30) PostalCode     str(200) AddressInLatin;     str(90) BranchId;     int(16) ParentPersonId;     str(4) ParentInstName;     str(50) ResidentState;     int CategoryId;     str(100) FirstName;     str(100) LastName;     str(100) MiddleName;     str(20) PhoneNumber;     str(100) FirstNameNat;     str(100) LastNameNat;     str(100) MiddleNameNat;     str(300) Email;     str(240) Branch; }</pre> <p>} [] – customer personal information; not more than 100 customers</p> <p>PersonId – customer ID.  FIO – full name.  Sex - 'M' – male, 'F' – female, 'N' – not specified, 'U' – unknown).  IdentType – type of the identity document:  1 – passport.  2 – driving licence.  3 – social insurance number.  4 – TPN.  5 – national ID.  6 – international passport.  7 – military identity card.  8 – other document type.  SecretWord – customer secret word.  Birthday – date of birth.  Birthplace – place of birth.  InstName – name of the institution where the customer is defined.  ResidentCountry – country of the resident.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
ResidentState – state/region of the resident. ResidentCityInLatin – city of the resident (in Latin characters). PostalCode – postal code. AddressInLatin – address in Latin characters. BranchId – customer branch ID. Branch – customer branch name. ParentPersonId – parent customer ID. ParentInstName – name of the parent customer institution. CategoryId – customer category (see the list of available values in <i>FIMI/InitSession/Rp/PersonCategoryList</i> ). FirstName, LastName, MiddleName – name, last name, middle name of a customer (in Latin characters). PhoneNumber – phone number. FirstNameNat, LastNameNat, MiddleNameNat – name, last name, middle name of a customer (in national characters). Email – E-mail address.			
FIMI/GetPersonInfo/Rp/Confidential	ArrRec	Out	{ Int(16)* PersonId; str(150)* What; str(300) Value; int(1)* IsAllowedCST; int(1)* IsAllowedADS; int(1)* IsAllowedTB; } [] – customer confidential information in the “question-answer” form:
What – question, e.g.: “What is your mother’s maiden name?” Value – answer. IsAllowedCST – question is allowed in the Customer Support Terminal application. Available values: 0 and 1; IsAllowedADS – question is allowed in the 3-D Secure ADS (Activation During Shopping) service. Available values: 0 and 1. IsAllowedTB – question is allowed in Telebank. Available values: 0 and 1.			
FIMI/GetPersonInfo/Rp/Cards	ArrRec	Out	{ Int(16)* PersonId; str(19)* PAN; int(3)* MBR; int* Status; str(64) CardUID; time* ExpDate; int(1)* Type. } [] – customer cards, not more than 100 cards for each customer:
Status – card status (see <i>InitSession/Rp/CardStatList/Id</i> or p. 3.2). ExpDate – card expiration date. Type – card type (see the list of available values in <i>FIMI/GetCardInfo/Rq/Type</i> ).			

<b>Parameter</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
FIMI/GetPersonInfo/Rp/Accounts	ArrRec	Out	{           Int(16)* PersonId;           str(30)* Account;           int(2)* Type;           int(1)* Status;           str(32) AccountUID;           str(40) AggregateId;           int AggregatePriority;         } [] – customer accounts, not more than 100 accounts for each customer:
Type – account type (1–checking; 11–savings; 31–credit; 91 - bonus).			
Status – account status:			
0 – Inactive account 1 – Open 2 – Deposit only 3 – Primary open account 4 – Deposit only primary account 5 – Information only 9 – Closed			
AggregateId – accounts aggregate ID.			
AggregatePriority – priority of the account in aggregate.			
FIMI/GetPersonInfo/Rp/UserFields	ArrRec	Out	{           str(100)* Name;           str(4000) TextValue;           strXML XMLValue;         } [] – array of structures  List of the customer user fields; it is filled if only one customer is found Name, TextValue, XMLValue – name, text and XML values of the user fields.
FIMI/GetPersonInfo/Rp/ AlternativeMessaging	ArrRec	Out	{           Str(20)* Channel;           Str(500)* Address;           Str(20) Scheme;           Str(20) Name;           Str(150) Title;           Int(1)* Disabled;           Int(1)* UseForDynAuth;           Int(1)* IsDefault;           Int(1)* Broadcast;           Int(1)* DynAuthBlocked;           Time ErrorFirstTime;           Int ErrorCode;         } [] – list of channel and address abonents for SMS commands delivery and reception by the customer, where:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<p>Channel, Address, Scheme – channel, address, scheme used to send/receive commands (list of available values for the channel and scheme provided in the fields <i>InitSession/Rp/CNSChannelList.Name</i> and <i>InitSession/Rp/CNSSchemeList.Name</i>).</p> <p>Name – name of the messaging abonent to be used in the commands.</p> <p>Title – any description.</p> <p>Disabled – “Abonent Disabled” attribute.</p> <p>UseForDynAuth – use for dynamic authentication.</p> <p>IsDefault – “Default Abonent for Dynamic Authentication” attribute. It can be enabled only for one abonent for a certain customer.</p> <p>Broadcast – allow the broadcast messages.</p> <p>DynAuthBlocked – “Abonent is Blocked for Dynamic Authentication” attribute.</p> <p>ErrorFirstTime – first time when the current dynamic authentication password was entered incorrectly, it is filled if UseForDynAuth=1 and ErrorCount&gt;0.</p> <p>ErrorCount – number of dynamic authentication password bad tries, it is filled if UseForDynAuth=1.</p> <p>The list of customer abonents, it is filled if only one customer is found.</p>			
<p>FIMI/GetPersonInfo/Rp/ MessagingProfile</p> <p>ArrRec</p> <p>Out</p> <pre>{     str(20)* Channel;     str(500)* Address;     int* ItemType;     int* ItemCode;     int MinAmount;     int(1) Disabled; }</pre> <p>} [] – array of structures, list of the customer messaging profiles:</p>			
<p>Channel, Address, Scheme – channel, address, scheme used to send/receive commands (for the list of available values for the channel and scheme, see fields <i>InitSession/Rp/CNSChannelList.Name</i> and <i>InitSession/Rp/CNSSchemeList.Name</i>).</p> <p>ItemType – scheme item type (for the list of available values, see structure <i>GetSchemeItemTypes/Rp/SchemeItemTypes</i>, see p.2.2.3.6).</p> <p>ItemCode – scheme item code (for the list of available values, see structure <i>GetSchemeItemTypes/Rq/SchemeItemTypes/ItemCode</i>, see p.2.2.3.6).</p> <p>MinAmount – minimum amount requiring the notification.</p> <p>Disabled – “Profile Disabled” attribute.</p> <p>The list of customer abonent profiles, it is filled if only one customer is found.</p>			
<b>CreatePerson – creating customer</b>			
FIMI/CreatePerson/Rq/InstName	Str(4)*	In	Institution name
FIMI/CreatePerson/Rq/BranchId	Str(90)	In	Branch ID
FIMI/CreatePerson/Rq/FIO	Str(300)*	In	Customer full name
FIMI/CreatePerson/Rq/VIP	Int(1)*	In	Flag “VIP”
FIMI/CreatePerson/Rq/Sex	Str(1)*	In	Sex M – male F – female N – not specified U – unknown

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreatePerson/Rq/IdentType	Int(1)	In	Type of identification. For the list of available values, see the <i>GetPersonInfo/Rp/Info</i> field.
FIMI/CreatePerson/Rq/Identity	Str(120)	In	Identifier – number of the identity paper
FIMI/CreatePerson/Rq/SecretWord	Str(60)	In	Secret word
FIMI/CreatePerson/Rq/Birthday	Time	In	Date of birth
FIMI/CreatePerson/Rq/Birthplace	Str(250)	In	Place of birth
FIMI/CreatePerson/Rq/ResidentCountry	Int	In	Resident country (ISO code)
FIMI/CreatePerson/Rq/ResidentState	Str(50)	In	Resident state/region
FIMI/CreatePerson/Rq/ResidentStateExternalId	Str(20)	In	Resident state/region external ID The field must be sent for the correct processing of the operation in Back-Office if the <i>ResidentState</i> field is filled.
FIMI/CreatePerson/Rq/ResidentCityInLatin	Str(100)	In	Resident city (in Latin characters)
FIMI/CreatePerson/Rq/AddressInLatin	Str(200)	In	Address (in Latin characters)
FIMI/CreatePerson/Rq/PostalCode	Str(30)	In	Postal code
FIMI/CreatePerson/Rq/PersonId	Int(16)	In	ID of customer being created
FIMI/CreatePerson/Rq/ChangeReason	Str(1000)	In	Customer creation reason
FIMI/CreatePerson/Rq/CategoryId	Int	In	Customer category (see the list of available values in <i>FIMI/InitSession/Rp/PersonCategoryList</i> )
FIMI/CreatePerson/Rq/FirstName	Str(100)	In	Name of a customer (in Latin characters)
FIMI/CreatePerson/Rq/LastName	Str(100)	In	Last name of a customer (in Latin characters)
FIMI/CreatePerson/Rq/MiddleName	Str(100)	In	Middle name of a customer (in Latin characters)
FIMI/CreatePerson/Rq/FirstNameNat	Str(100)	In	Name of a customer (in national characters)
FIMI/CreatePerson/Rq/LastNameNat	Str(100)	In	Last name of a customer (in national characters)
FIMI/CreatePerson/Rq/MiddleNameNat	Str(100)	In	Middle name of a customer (in national characters)
FIMI/CreatePerson/Rq/PhoneNumber	Str(20)	In	Phone number
FIMI/CreatePerson/Rq/Email	Str(300)	In	E-mail address
FIMI/CreatePerson/Rq/PersonExtId	Str(50)	In	External customer ID

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreatePerson/Rp/PersonId	Int(16)	Out	Created customer ID
<b>UpdatePerson – updating information on private customer</b>			
FIMI/UpdatePerson/Rq/Id	Int(16)*	In	Customer ID (unchangeable)
FIMI/UpdatePerson/Rq/InstName	Str(4)*	In	Institution name (unchangeable)
FIMI/UpdatePerson/Rq/BranchId	Str(90)	In	Branch ID
FIMI/UpdatePerson/Rq/FIO	Str(300)	In	Customer full name
FIMI/UpdatePerson/Rq/VIP	Int(1)	In	“VIP” sign
FIMI/UpdatePerson/Rq/Sex	Str(1)	In	Sex M – male F – female N – not specified U – unknown
FIMI/UpdatePerson/Rq/IdentType	Int	In	Type of identification. For the list of available values, see the <i>GetPersonInfo/Rp/Info</i> field.
FIMI/UpdatePerson/Rq/Identity	Str(120)	In	Number of the customer identity document
FIMI/UpdatePerson/Rq/SecretWord	Str(60)	In	Secret word
FIMI/UpdatePerson/Rq/Birthday	Time	In	Birth date
FIMI/UpdatePerson/Rq/Birthplace	Str(250)	In	Birth place
FIMI/UpdatePerson/Rq/ResidentCountry	Int	In	Resident country (ISO code)
FIMI/UpdatePerson/Rq/ResidentState	Str(50)	In	Resident state/region
FIMI/UpdatePerson/Rq/ResidentStateExternalId	Str(20)	In	Resident state/region external ID The field must be sent for the correct processing of the operation in Back-Office if the <i>ResidentState</i> field is filled.
FIMI/UpdatePerson/Rq/ResidentCityInLatin	Str(100)	In	Resident city (in Latin characters)
FIMI/UpdatePerson/Rq/AddressInLatin	Str(200)	In	Address (in Latin characters)
FIMI/UpdatePerson/Rq/PostalCode	Str(30)	In	Postal code
FIMI/UpdatePerson/Rq/ParentPersonId	Int(16)	In	Parent customer ID
FIMI/UpdatePerson/Rq/ParentInstName	Str(4)	In	Name of the parent customer institution
FIMI/UpdatePerson/Rq/ChangeReason	Str(1000)	In	Reason for the customer information change

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdatePerson/Rq/UpdateFieldsMethod	Int(1)	In	If = 0 or NULL – only the transferred fields will be updated. If = 1 – the optional fields that were not sent will be set to NULL. The FIO, VIP and Sex fields cannot be set to NULL, they will be left unchanged
FIMI/UpdatePerson/Rq/CategoryId	Int	In	Customer category (see the list of available values in <i>FIMI/InitSession/Rp/PersonCategoryList</i> )
FIMI/UpdatePerson/Rq/FirstName	Str(100)	In	Name of a customer (in Latin characters)
FIMI/UpdatePerson/Rq/LastName	Str(100)	In	Last name of a customer (in Latin characters)
FIMI/UpdatePerson/Rq/MiddleName	Str(100)	In	Middle name of a customer (in Latin characters)
FIMI/UpdatePerson/Rq/FirstNameNat	Str(100)	In	Name of a customer (in national characters)
FIMI/UpdatePerson/Rq/LastNameNat	Str(100)	In	Last name of a customer (in national characters)
FIMI/UpdatePerson/Rq/MiddleNameNat	Str(100)	In	Middle name of a customer (in national characters)
FIMI/UpdatePerson/Rq/PhoneNumber	Str(20)	In	Phone number
FIMI/UpdatePerson/Rq/Email	Str(300)	In	E-mail address
FIMI/UpdatePerson/Rq/PersonExtId	Str(50)	In	External customer ID

#### ***SetPersonUserFields – setting customer user fields***

FIMI/SetUserFields/Rq/InstName	Str(4)	In	Identifier (name) of the customer financial institution If not defined, the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/SetUserFields/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/SetUserFields/Rq/UserFields	ArrRec	In	{ Str(100)* Name; Str(3800) TextValue; StrXML XMLValue. } [] – array of structures with user fields:

Name – field name.

TextValue – field text value.

XMLValue – XML value of the field.

The total length of the *Name* and *TextValue* fields in all the records of the array *FIMI/SetPersonUserFields/Rq/UserFields* must not exceed 3900 symbols.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
If the <i>TextValue</i> and <i>XMLValue</i> fields are absent in the structure and customer has the user field, this user field will not be changed. To reset the <i>TextValue</i> and <i>XMLValue</i> values, specify empty values of the <i>TextValue</i> and <i>XMLValue</i> fields.			
FIMI/SetUserFields/Rq/ RemoveMissingFields	Int(1)	In	1 – remove missing fields by default = 0  If FIMI/InitSession/Rp/ LimitedAccessToUserFields = 1, only the user fields available for editing will be deleted
FIMI/SetPersonUserFields/Rq/ NeedNotify	Int(1)	In	=1 – notify a customer if the notification is enabled  =0 or NULL – do not notify a customer event if the notification is enabled
<b>AddPersonConfidentialInfo – adding customer confidential information</b>			
FIMI/AddPersonConfidentialInfo/Rq/ InstName	Str(4)	In	Identifier (name) of the customer financial institution  If not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/AddPersonConfidentialInfo/Rq/ PersonId	Int(16)*	In	Customer ID
FIMI/AddPersonConfidentialInfo/Rq/ What	Str(150)*	In	Question
FIMI/AddPersonConfidentialInfo/Rq/ Value	Str(300)*	In	Answer
FIMI/AddPersonConfidentialInfo/Rq/ IsAllowedCST	Int(1)	In	Indicates that the question is in the Customer Support Terminal application.  If NULL, then 1 – the question is allowed
FIMI/AddPersonConfidentialInfo/Rq/ IsAllowedADS	Int(1)	In	Indicates that the question is allowed in the 3-D Secure ADS (Activation During Shopping) service.  If NULL, then 1 – the question is allowed
FIMI/AddPersonConfidentialInfo/Rq/ IsAllowedTB	Int(1)	In	Indicates that the question is allowed in Telebank.  If NULL, then 1 – the question is allowed
<b>DelPersonConfidentialInfo – deleting customer confidential information</b>			
FIMI/DelPersonConfidentialInfo/Rq/ InstName	Str(4)	In	Identifier (name) of the customer financial institution  If not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/DelPersonConfidentialInfo/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/DelPersonConfidentialInfo/Rq/What	Str(150)*	In	Question
<b>ChangePersonConfidentialInfo – changing customer confidential information</b>			
FIMI/ChangePersonConfidentialInfo/Rq/InstName	Str(4)	In	Character identifier (name) of financial institution where the customer is created  If not defined, the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/ChangePersonConfidentialInfo/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/ChangePersonConfidentialInfo/Rq/What	Str(150)*	In	Question
FIMI/ChangePersonConfidentialInfo/Rq/Value	Str(300)	In	Answer
FIMI/ChangePersonConfidentialInfo/Rq/IsAllowedCST	Int(1)	In	Attribute indicating that the question is allowed in the “Customer Support Terminal” application.
FIMI/ChangePersonConfidentialInfo/Rq/IsAllowedADS	Int(1)	In	Attribute indicating that the question is allowed in 3-D Secure ADS (Activation During Shopping).
FIMI/ChangePersonConfidentialInfo/Rq/IsAllowedTB	Int(1)	In	Attribute indicating that the question is allowed in Telebank.
<b>GetPersonsHierarchy – request for customers hierarchy</b>			
FIMI/GetPersonsHierarchy/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the customer.  If not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/GetPersonsHierarchy/Rq/Id	Int(16)*	In	Customer ID
FIMI/GetPersonsHierarchy/Rq/Direction	Int(1)	In	Hierarchy direction:  1 – from the first child customer to the current one (specified in the Id field) inclusive; 2 – from the current customer inclusive to the last parent customer; 3 – full hierarchy: from the first child customer to the last parent customer including the current customer regardless of its location in the hierarchy.  If = NULL, it = 3.
FIMI/GetPersonsHierarchy/Rp/Hierarchy	ArrRec	Out	{ str(4)* InstName; int(16)* PersonId;

<b>Parameter</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
			str(4) ParentInstName; int(16) ParentPersonId; str(300) Name; str(1) Sex; int(1) IdentType; str(120) Identity; time Birthday; } [] – array of structures with the customers hierarchy;
InstName – customer institution name. PersonId – customer ID. ParentInstName – name of the parent customer institution. ParentPersonId – parent customer ID. Name – full name. Sex – sex ('M' – male, 'F' – female, 'N' – not specified, 'U' – unknown). IdentType – customer identification type (for the list of available values, see the <i>GetPersonInfo/Rp/Info</i> field). Identity – number of the document identifying the customer. Birthday – date of birth.			
<b>DeletePerson – deleting customer</b>			
FIMI/DeletePerson/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the customer. If not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/DeletePerson/Rq/Id	Int(16)*	In	Customer ID
<b>GetAvatar – getting customer avatar</b>			
FIMI/GetAvatar/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the customer. If not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> . It is defined if the customer is searched for by the customer ID (PersonId field is set).
FIMI/GetAvatar/Rq/PersonId	Int(16)	In	Customer ID
FIMI/GetAvatar/Rq/TelebankId	Str(19)	In	Telebank customer ID. It is used if the PersonId field is not set.
FIMI/GetAvatar/Rq/TelebankMBR	Int(3)	In	Number of the Telebank customer owner. It is used if the PersonId field is not set.
FIMI/GetAvatar/Rq/TelebankUID	Str(64)	In	Telebank customer unique ID It can be specified instead of the Telebank customer ID. It is used if the PersonId field is not set.
FIMI/GetAvatar/Rp/AvatarId	Int(9)	Out	Avatar ID stored in ACS/TWIB/other systems. It is used for standard avatars.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetAvatar/Rp/AvatarData	Str(65535)	Out	Avatar data in the Base64 encoding. It is used for user avatars.
FIMI/GetAvatar/Rp/AvatarInfo	Str(100)	Out	Avatar information – MIME type of data transferred in the AvatarData field. It is used for the correct displaying of the user avatar in ACS/TWIB/other systems.
<b>SetAvatar – changing customer avatar</b>			
FIMI/SetAvatar/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the customer.  If not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in TWO.  It is defined if the customer is accessed by the customer ID (PersonId field is set).
FIMI/SetAvatar/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SetAvatar/Rq/TelebankId	Str(19)	In	Telebank customer ID. It is used if the PersonId field is not set.
FIMI/SetAvatar/Rq/TelebankMBR	Int(3)	In	Number of the Telebank customer owner. It is used if the PersonId field is not set.
FIMI/SetAvatar/Rq/TelebankUID	Str(64)	In	Telebank customer unique ID It can be specified instead of the Telebank customer ID. It is used if the PersonId field is not set.
FIMI/SetAvatar/Rq/AlphaNumericPassword	Str(16)	In	Hash of the Telebank customer alphanumeric password. It is used if the PersonId field is not set.
FIMI/SetAvatar/Rq/PINBlock	Hex(16)	In	Telebank customer PIN block. If set, it is also required to specify the working key ID in the KeyId field.  PIN block is presented in the ANSI X9.8 (ISO Format 0) format and transferred by the encrypted working key. It is used if the PersonId field is not set.
FIMI/SetAvatar/Rq/KeyId	Int	In	ID of the working key received by means of the GetWorkingKey request. The field must be set if the PIN block is transferred.
FIMI/SetAvatar/Rq/AvatarId	Int(9)	In	Avatar ID stored in ACS/TWIB/other systems. It is used for standard avatars. -1 – reset the field value. The field can be transferred with the value other than -1 if the AvatarData field is not set.
FIMI/SetAvatar/Rq/AvatarData	Str(65535)	In	Avatar data in the Base64 encoding. It is used for user avatars. To reset, specify the field empty value. The field can be transferred with the value other than empty if the AvatarId field is not set.

<b>Parameter</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
FIMI/SetAvatar/Rq/AvatarInfo	Str(100)	In	Avatar information – MIME type of data transferred in the AvatarData field. It is required for the correct displaying of the user avatar in ACS/TWIB/other systems. It is used if the AvatarData field is transferred. To reset, specify the empty value for the AvatarData field.
<b>AddPersonCMSAbonent – adding messaging parameters for customer</b>			
FIMI/AddPersonCMSAbonent/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer. If not defined, the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/AddPersonCMSAbonent/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/AddPersonCMSAbonent/Rq/AlternativeMessaging	ArrRec*	In	{ Str(20)* Channel; Str(500)* Address; Str(20) Scheme; Str(20) Name; Str(150) Title; Int(1) Disabled; Str TestMessage; Int(1) UseForDynAuth; Int(1) IsDefault; Int(1) Priority; Int(1) Broadcast } [] – list of channels and addresses for delivery and reception of SMS commands by card:
Channel, Address, Scheme – channel, address, scheme used to send/receive commands (list of available values for the channel and scheme is provided in the fields <i>InitSession/Rp/CNSChannelList.Name</i> and <i>InitSession/Rp/CNSSchemeList.Name</i> ).  Name – name of the messaging abonent to be used in the commands.  Title – any description.  Disabled – "Abonent Disabled" attribute. If NULL, it=0.  TestMessage – test message. If not NULL, the defined test message is sent to the customer via the defined channel and to the respective address.  UseForDynAuth – use for dynamic authentication.  IsDefault – "Default Abonent for Dynamic Authentication" attribute. It can be enabled only for one abonent for a certain card.  Priority – priority of test message sending (0 – low; 1 – below average; 2 – average; 3 – above average; 4 – high). By default, 4 – high.  Broadcast – allow the broadcast messages.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddPersonCMSAbonent/Rq/NeedNotify	Int(1)	In	= 1 or NULL – notify a customer if notification on the customer is enabled = 0 – do not notify a customer even if notification on the customer is enabled
FIMI/AddPersonCMSAbonent/Rq/ChangeReason	Str(1000)	In	Reason for setting the messaging parameters
The FIMI/AddPersonCMSAbonent/Rq/AlternativeMessaging/Disabled field affects sending of the customer notifications only. If FIMI/AddPersonCMSAbonent/Rq/AlternativeMessaging/Disabled = 1, the notifications will not be generated but the commands incoming via sms from customers will be accepted and processed successfully. If FIMI/AddPersonCMSAbonent/Rq/AlternativeMessaging/Scheme = NULL, the incoming customer commands will not be processed.			
<b>RemovePersonCMSAbonent – deleting customer messaging parameters</b>			
FIMI/RemovePersonCMSAbonent/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer. If not defined, the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/RemovePersonCMSAbonent/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/RemovePersonCMSAbonent/Rq/AlternativeMessaging	ArrRec*	In	{ Str(20)* Channel; Str(500)* Address } [] – list of channels and addresses for delivery and reception of SMS commands by card: Channel, Address – channel, address used to send/receive commands (list of available values is provided in the field InitSession/Rp/CNSChannelList.Name)
FIMI/RemovePersonCMSAbonent/Rq/NeedNotify	Int(1)	In	= 1 or NULL – notify a customer if notification on the customer is enabled = 0 – do not notify a customer even if notification on the customer is enabled
FIMI/RemovePersonCMSAbonent/Rq/ChangeReason	Str(1000)	In	Reason for setting the messaging parameters
<b>ChangePersonCMSAbonent – changing messaging parameters for customer</b>			
FIMI/ChangePersonCMSAbonent/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer If not defined, the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/ChangePersonCMSAbonent/Rq/PersonId	Int(16)*	In	Customer ID

<b>Parameter</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
FIMI/ChangePersonCMSAbonent/Rq /AlternativeMessaging	ArrRec	In	<pre>{     Str(20)* Channel;     Str(500)* Address;     Str(20) Scheme;     Str(20) Name;     Str(150) Title;     Int(1) Disabled;     Str TestMessage;     Int(1) UseForDynAuth;     Int(1) IsDefault;     Int(1) Priority;     Int(1) Broadcast;     Str(20) NewChannel;     Str(500) NewAddress }</pre> <p>} [] – list of channels and addresses for delivery and reception of SMS commands by card:</p> <p>Channel, Address, Scheme – channel, address, scheme used to send/receive commands (list of available values for the channel and scheme is provided in the fields InitSession/Rp/CNSChannelList.Name and InitSession/Rp/CNSSchemeList.Name).</p> <p>Name – name of the messaging abonent to be used in the commands. If the Scheme field is not sent, the value will be reset to NULL.</p> <p>Title – any description.</p> <p>Disabled – “Abonent Disabled” attribute, if NULL, it=0.</p> <p>TestMessage – test message, if not NULL, the defined test message is sent to the customer via the defined channel and to the respective address.</p> <p>UseForDynAuth – use for dynamic authentication.</p> <p>IsDefault – “Default Abonent for Dynamic Authentication” attribute. It can be enabled only for one abonent for a certain card.</p> <p>Priority – priority of test message sending (0 – low; 1 – below average; 2 – average; 3 – above average; 4 – high). By default, 4 – high.</p> <p>Broadcast – allow the broadcast messages.</p> <p>NewChannel, NewAddress – new values of the channel and address.</p> <p>The NewChannel and NewAddress fields are filled only if the existing notification item is changed or filled in with new data.</p>
FIMI/ChangePersonCMSAbonent /Rq/NeedNotify	Int(1)	In	= 1 or NULL – notify a customer if notification on the customer is enabled = 0 – do not notify a customer even if notification on the customer is enabled
FIMI/ChangePersonCMSAbonent /Rq/ChangeReason	Str(1000)	In	Reason for setting the messaging parameters

The FIMI/ChangePersonCMSAbonent/Rq/AlternativeMessaging/Disabled field affects sending of the customer notifications only. If FIMI/ChangePersonCMSAbonent/Rq/AlternativeMessaging = 1, the notifications will not be generated but the commands incoming via sms from customers will be accepted and processed successfully. If FIMI/ChangePersonCMSAbonent/Rq/AlternativeMessaging/Scheme = NULL, the incoming customer commands will not be processed.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>ConfigurePersonCMSProfile – setting customer messaging parameters</b>			
FIMI/ConfigurePersonCMSProfile/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer  If not defined, the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/ConfigurePersonCMSProfile/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/ConfigurePersonCMSProfile/Rq/Channel	Str(20)*	In	Channel
FIMI/ConfigurePersonCMSProfile/Rq/Address	Str(500)*	In	Address
FIMI/ConfigurePersonCMSProfile/Rq/MessagingProfile	ArrRec	In	<pre>{     int* ItemType;     int* ItemCode;     num MinAmount;     int(1) Disabled; } [] – array of structures,</pre> <p>values of items responsible for sending SMS notifications. Existing items of the abonent are replaced with those of the array. If the array contains no items, all existing items are deleted.</p> <p>The description of fields is similar to that of the FIMI/GetPersonInfo/MessagingProfile field in the <i>GetPersonInfo</i> operation. The <i>ItemCode</i> field is extended with the following values:  -1 – all.  0 – any.</p>
FIMI/ConfigurePersonCMSProfile/Rq/NeedNotify	Int(1)	In	= 1 or NULL – notify a customer if the notification on the customer is enabled = 0 – do not notify a customer even if the notification on the customer is enabled
FIMI/ConfigurePersonCMSProfile/Rq/ChangeReason	Str(1000)	In	Reason for setting the messaging parameters
<b>ResetPersonBadDynAuthPwdTries – reset of counter of dynamic authentication password bad tries for card abonent</b>			
FIMI/ResetPersonBadDynAuthPwdTries/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer  If not defined, the institution owning FIMI terminal stands for the customer institution in <b>TWO</b> .
FIMI/ResetPersonBadDynAuthPwdTries/Rq/PersonId	Int(16)*	In	Customer ID

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ResetPersonBadDynAuthPwdTries/Rq/Channel	Str(20)*	In	Channel
FIMI/ResetPersonBadDynAuthPwdTries/Rq/Address	Str(500)*	In	Address
<b>ResetPersonAbonentAdditionalInfo – reset of additional data on abonent for customer abonent</b>			
FIMI/ResetPersonAbonentAdditionalInfo/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer  If not defined, the institution owning FIMI terminal stands for the customer institution in TWO.
FIMI/ResetPersonAbonentAdditionalInfo/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/ResetPersonAbonentAdditionalInfo/Rq/Channel	Str(20)*	In	Channel
FIMI/ResetPersonAbonentAdditionalInfo/Rq/Address	Str(500)*	In	Address
FIMI/ResetPersonAbonentAdditionalInfo/Rq/ChangeReason	Str(1000)	In	Reason for the abonent additional data reset
<b>GetPersonLimits – request for list of customer limits with their maximum and current values</b>			
FIMI/GetPersonLimits/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the customer  If not defined, the institution owning FIMI terminal stands for the customer institution in TWO.  For the list of available values, see the field <i>InitSession/Rp/AllowedFIID.Name</i> .
FIMI/GetPersonLimits/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/GetPersonLimits/Rq/GetCurrentCounters	Int(1)	In	if NULL then = 0  If = 1, the response will return current counter values in the <i>Limits[i].Current</i> fields. At that, the <i>AuthScheme</i> field must be defined in the request.
FIMI/GetPersonLimits/Rq/AuthScheme	Int	In	Authorization scheme ID. The field must be defined if <i>GetCurrentCounters</i> =1. The field is used to request the values of the group limit of the authorization scheme customer. If NULL, certain values of the group limit are not displayed.  For the list of available values, see the field <i>InitSession/Rp/AuthSchemes</i> .

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetPersonLimits/Rq/Limits	ArrRec	Out	<pre>{     int* LimitId;     num Max;     num* Current;     int(2) PeriodType;     int Period;     num TmpMax;     time TmpMaxExpiration;     num TmpMaxRange;     int(1) TmpMaxPeriodType;     time DateTimeForReset;     time LastResetTime; }</pre> <p>} [] – array of structures, List of limits with their maximum and current values:</p>
<p>LimitId – limit ID in <b>TWO</b> (see p. 3.3).</p> <p>Max – maximum limit value (0 – restricted, NULL – unlimited, &gt;0 – defined, -1 – as auth. scheme value).</p> <p>Current – current limit value.</p> <p>LastResetTime – date/time when the limit counter was last reset.</p> <p>PeriodType, Period – limit validity period (for the description, see the message <i>SetCardLimits</i>).</p> <p>TmpMax – temporary maximum limit value (0 – restricted, NULL – not defined, &gt;0 – defined).</p> <p>TmpMaxExpiration – temporary maximum validity period (NULL – until the next reset of the limit counter).</p> <p>TmpMaxRange – value restricting the range of the TmpMax parameter change (0 – restricted, NULL – not defined, &gt;0 – defined).</p> <p>TmpMaxPeriodType – type of the limit temporary maximum period (NULL – for a period; 1 – per a single operation).</p> <p>DateTimeForReset – calendar date and time of the current limit value reset.</p>			
FIMI/GetPersonLimits/Rp/LimitsRanges	ArrRec	Out	<pre>{     int* LimitId;     time* StartDate;     time EndDate;     num Max; }</pre> <p>} [] – array of structures, list of the limit ranges returned in the Limits structure:</p>
<p>LimitId – limit ID in <b>TWO</b> (see p. 3.3).</p> <p>StartDate – start date of the range.</p> <p>EndDate – end date of the range (NULL – range is unlimited).</p> <p>Max – maximum value of the limit in the range (NULL – as maximum limit value).</p>			
FIMI/GetPersonLimits/Rp/LimCurrency	Int*	Out	Currency of the financial customer limits
<b>SetPersonLimits – changing maximum values and validity periods of customer limits</b>			
FIMI/SetPersonLimits/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution that owns the account.</p> <p>If not defined, the institution owning FIMI terminal stands for the account institution in <b>TWO</b>.</p>
FIMI/SetPersonLimits/Rq/PersonId	Int(16)*	In	Customer ID

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetPersonLimits/Rq/RemoveMissingLimits	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing limits. 2 – remove received limits. If NULL, it is 0.
FIMI/SetPersonLimits/Rq/LimCurrency	Int	In	Code of the limits currency. It is possible to set the limit currency only if all limit values are set for the customer. Therefore, this parameter can be defined if RemoveMissingLimits = 1.
FIMI/SetPersonLimits/Rq/Limits	ArrRec	In	{ int* LimitId; num Max; int(2) PeriodType; int Period; time DateTimeForReset; } [] – array of structures, List of limits and their maximum values:
<p>LimitId - limit ID in TWO (see p. 3.3).</p> <p>Max –maximum limit value in the account currency (0 - restricted, NULL - unlimited, &gt;0 - defined, -1 - as auth. scheme value).</p> <p>PeriodType, Period, DateTimeForReset – limit validity period (see the description in the field <i>GetPersonLimits</i>).</p>			
FIMI/SetPersonLimits/Rq/ChangeReason	Str(1000)	In	Reason for changing maximum values and validity periods of the account limits
FIMI/SetPersonLimits/Rq/NeedNotify	Int(1)	In	= 1 or NULL – notify a customer if the notification is enabled = 0 – do not notify a customer even if the notification on cards is enabled
<b><i>SetPersonLimitsRanges – setting customer limit ranges</i></b>			
FIMI/SetPersonLimitsRanges/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the account. If not defined, the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/SetPersonLimitsRanges/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/SetPersonLimitsRanges/Rq/LimitId	Int*	In	Limit ID in TWO (see p. 3.3)
FIMI/SetPersonLimitsRanges/Rq/RemoveMissing	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing ranges. 2 – remove received ranges. If NULL, it is 0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetPersonLimitsRanges/Rq/LimitsRanges	ArrRec	In	<pre>{     time StartDate;     time EndDate;     num Max; }</pre> <p>[] – array of structures, list of limit ranges:</p> <p>StartDate – start date of the range (NULL – starting from the operation execution).      EndDate – end date of the range (NULL – range is unlimited).      Max – maximum value of the limit in the range (NULL – as maximum limit value).</p>
FIMI/SetPersonLimitsRanges/Rq/ChangeReason	Str(1000)	In	Reason for setting maximum values and validity periods of card limits
FIMI/SetPersonLimitsRanges/Rq/NeedNotify	Int(1)	In	= 1 or NULL – notify a customer if the notification is enabled = 0 – do not notify a customer even if the notification on cards is enabled
<b>GetPersonGroupLimits – request for limit values on customer of scheme level</b>			
FIMI/GetPersonGroupLimits/Rq/AuthScheme	Int*	In	Authorization scheme ID. For the list of available values, see the field InitSession/Rp/AuthSchemes
FIMI/GetPersonGroupLimits/Rp/Limits	ArrRec	Out	<pre>{     int* LimitId;     int* Currency;     num Max;     int(2) PeriodType;     int Period;     time DateTimeForReset; }</pre> <p>[] – array of structures,</p> <p>List of limits with their maximum values and validity periods (for the description of the limit validity periods, see the operation <i>GetPersonLimits</i>)</p>
<b>ResetPersonLimitCounters – reset of customer limits counter</b>			
FIMI/ResetPersonLimitCounters/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the account. If not defined, the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/ResetPersonLimitCounters/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/ResetPersonLimitCounters/Rq/Limits	Int[]*	In	List of identifier of limits for which the counter is reset.
FIMI/ResetPersonLimitCounters/Rq/ChangeReason	Str(1000)	In	Reason for the limits counter reset.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>SetTmpPersonLimits – setting temporary maximum values for customer limits</b>			
FIMI/SetTmpPersonLimits/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the account. If not defined, the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/SetTmpPersonLimits/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/SetTmpPersonLimits/Rq//Limits	ArrRec*	In	{ int* LimitId; num TmpMax; time TmpMaxExpiration; int TmpMaxPeriodType; } [] – array of structures, List of limits with their temporary maximum values:
LimitId – limit ID in TWO (see p. 3.3). TmpMax – temporary maximum limit value (it is reset when the limit validity period expires or upon a certain time period). TmpMaxExpiration – validity period of the temporary maximum (NULL – until the next reset of the limit counter). TmpMaxPeriodType – type of the limit temporary maximum period (NULL – for a period; 1 – per a single operation).			
FIMI/SetTmpPersonLimits/Rq/ChangeReason	Str(1000)	In	Reason for setting temporary maximum values for the account limit
FIMI/SetTmpPersonLimits/Rq/NeedNotify	Int(1)	In	= 1 or NULL – notify a customer if the notification is enabled = 0 – do not notify a customer even if the notification on cards is enabled
<b>GetPersonCounters – getting customer counters</b>			
FIMI/GetPersonCounters/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution that owns the account. If not defined, the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/GetPersonCounters/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/GetPersonCounters/Rq/AuthScheme	Int	In	Authorization scheme ID. If defined, the counter parameters are returned considering the authorization scheme parameters. See the list of available values in <i>InitSession/Rp/AuthSchemes</i> .

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetPersonCounters/Rp/Counters	ArrRec	Out	<pre>{     int* CounterId;     num* MaxValue;     num* Current;     int(2)* PeriodType;     time DateTimeForReset;     time LastResetTime;     int Period }</pre> <p>[]} – list of the customer counters.</p> <p>CounterId – counter ID in <b>TWO</b> (see the list of available values in <i>FIMI/InitSession/Rp/Counters</i>);      MaxValue – maximum counter value (available values &gt;=0, -1 – “as the scheme”, -2 – “unrestricted”);      Current – current counter value;      LastResetTime – time of the last reset;      PeriodType, LastTimeForReset, Period - period type and value (see the description in <i>GetCardCounters</i>).</p>
<b><i>SetPersonCounters – changing customer counters</i></b>			
FIMI/SetPersonCounters/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution that owns the account.</p> <p>If not defined, the institution owning FIMI terminal stands for the account institution in <b>TWO</b>.</p>
FIMI/SetPersonCounters/Rq/PersonId	Int(16)*	In	Customer ID
FIMI/SetPersonCounters/Rq/RemoveMissing	Int(1)	In	=1 – remove missing counters  By default = 0
FIMI/SetPersonCounters/Rq/Counters	ArrRec	In	<pre>{     int* CounterId;     num MaxValue;     num CurrentDelta;     int(2) PeriodType;     time DateTimeForReset;     int Period }</pre> <p>[]} – list of card counters:</p> <p>CounterId – counter ID in <b>TWO</b> (see the list of available values in <i>FIMI/InitSession/Rp/Counters</i>).      MaxValue – maximum counter value (available values &gt;=0, -1 – “as the scheme”, -2 – “unrestricted”).      CurrentDelta – positive or negative delta of the counter value change, if 0, the counter is reset.      PeriodType, DateTimeForReset, Period – period type and value (see the description in <i>GetCardCounters</i>).      If MaxValue, CurrentDelta, PeriodType = NULL, no impact on the respective attribute is performed.</p>
FIMI/SetPersonCounters/Rq/ChangeReason	Str(1000)	In	Reason for the customer counters change

## 2.3.3.5 Currency Exchange Rates Maintenance:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b><i>GetRateGroups – request for list of currency exchange rate groups</i></b>			
FIMI/GetRateGroups/Rp>List	ArrRec	Out	<pre>{     int* Id;     str(150) Title;     int* Reversible; }</pre> <p>[]} – currency exchange rate groups:      Reversible=1 – currency exchange rates in this group are reversible, = 0 – irreversible.      In the irreversible rates group, you should define the direct and reverse rates which can be different.      1st – from the first currency to the second one      2nd – from the second currency to the first one</p>
<b><i>GetRates – request for currency exchange rates</i></b>			
FIMI/GetRates/Rq/RateGroup	Int*	In	<p>Exchange rate group ID          For modes 1,2,3, the <i>GetRateGroups</i> request is used to get the list of available values.          For mode 4, the value is the same as ID in the external system, if no ID is present, the field is filled with value 0.</p>
FIMI/GetRates/Rq/Mode	Int*	In	<p>Request mode:          1-Current          2-Future          3-History          4-Host (request for the external system rates)</p>
FIMI/GetRates/Rq/InstName	Str(4)	In	<p>Financial institution name.          If NULL, the response contains the currency rates of all the trusted institutions of FIMI terminal and the rates of institution to which the FIMI terminal belongs in the specified rates group.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetRates/Rp>List	ArrRec	Out	{ time* ExchangeDate; int* FromCurrency; int* ToCurrency; str(3) FromCurrAlphaCode; str(3) ToCurrAlphaCode; num(12,6)* FromAmount; num(12,6)* ToAmount; str(4)* InstName; int(1)* Direction; time* SettingDate; str Description; } [] – currency rates.
ExchangeDate – business day. Direction = 1 – direct rate, = 2 – reverse rate. SettingDate – calendar date and time. Description – description.			
<b>AddRate – adding future exchange rate</b>			
FIMI/AddRate/Rq/RateGroup	Int	In	Exchange rate group ID
FIMI/AddRate/Rq/ExchangeDate	Time*	In	Business date when the future exchange rate is set (in TWO time zone)
FIMI/AddRate/Rq/FromCurrency	Int	In	1-st currency code
FIMI/AddRate/Rq/ToCurrency	Int	In	2-nd currency code
FIMI/AddRate/Rq/FromAmount	Num(12,6)	In	Amount in the FromCurrency for the direct rate. The value should be above zero
FIMI/AddRate/Rq/ToAmount	Num(12,6)	In	Amount in the ToCurrency for the direct rate. The value should be above zero
FIMI/AddRate/Rq/RevFromAmount	Num(12,6)	In	Amount in the ToCurrency for the reverse rate – it is defined if the rate is set in the irreversible group. The value should be above zero
FIMI/AddRate/Rq/RevToAmount	Num(12,6)	In	Amount in the FromCurrency for the reverse rate – it is defined if the rate is set in the irreversible group. The value should be above zero
FIMI/AddRate/Rq/InstName	Str(4)	In	Name of financial institution. If NULL, the institution of FIMI terminal is used
FIMI/AddRate/Rq/SettingDate	Time	In	Calendar date and time when the future rate is set (in TWO time zone). If not defined, the ExchangeDate value is used

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddRate/Rq/Rates	ArrRec	In	<pre>{     int* RateGroup;     int* FromCurrency;     int* ToCurrency;     num(12,6)* FromAmount;     num(12,6)* ToAmount;     num(12,6) RevFromAmount;     num(12,6) RevToAmount;     str(4) InstName;     time SettingDate; }</pre> <p>[] – currency rates being added.</p> <p>For the fields description, see the description of similar fields of the current request.</p> <p>If defined, the duplicated fields are ignored.</p>
FIMI/AddRate/Rp/MinExchangeDate	Time	Out	Min setting date of the new rate. It is defined together with Response=InvalidExchangeDate (see p. 3.1) if the request contains the invalid date of the future rate setting

**ModifyRate – modification of current or future exchange rate**

FIMI/ModifyRate/Rq/RateGroup	Int	In	Exchange rate group ID
FIMI/ModifyRate/Rq/ExchangeDate	Time*	In	Business date when the current or future exchange rate is active (in TWO time zone)
FIMI/ModifyRate/Rq/FromCurrency	Int	In	1st currency code
FIMI/ModifyRate/Rq/ToCurrency	Int	In	2nd currency code
FIMI/ModifyRate/Rq/FromAmount	Num(12,6)	In	Amount in the FromCurrency
FIMI/ModifyRate/Rq/ToAmount	Num(12,6)	In	Amount in the ToCurrency
FIMI/ModifyRate/Rq/RevFromAmount	Num(12,6)	In	The amount in the ToCurrency for the reverse rate is specified if the rate changes in the irreversible group. The specified value must be more than 0
FIMI/ModifyRate/Rq/RevToAmount	Num(12,6)	In	The amount in the FromCurrency for the reverse rate is specified if the rate changes in the irreversible group. If specified value must be more than 0
FIMI/ModifyRate/Rq/ClerkConfirmation	Int(1)	In	Confirmation of the current exchange rate change by the FIMI operator
FIMI/ModifyRate/Rq/InstName	Str(4)	In	Name of financial institution. If NULL, the institution of FIMI terminal is used
FIMI/ModifyRate/Rq/SettingDate	Time	In	Calendar date and time of the future or current rate (in TWO time zone). If not defined, the value won't be changed

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ModifyRate/Rq/Rates	ArrRec	In	<pre>{ int* RateGroup; int* FromCurrency; int* ToCurrency; num(12,6)* FromAmount; num(12,6)* ToAmount; num(12,6) RevFromAmount; num(12,6) RevToAmount; str(4) InstName; time SettingDate; }</pre> <p>[] – currency rates being changed.</p> <p>For the fields description, see the description of similar fields of the current request.</p> <p>If specified, the duplicated fields are ignored.</p>
FIMI/ModifyRate/Rq/ClerkConfirmation	Int(1)	In	FIMI operator confirmation to change the current rates.
FIMI/ModifyRate/Rp/MinExchangeDate	Time	Out	Minimum available date of the new exchange rate. It is defined together with Response= InvalidExchangeDate (see p. 3.1) if the request contained the invalid date of the future exchange rate
<b>DeleteRate – removing future exchange rate</b>			
FIMI/DeleteRate/Rq/RateGroup	Int	In	Exchange rates group ID
FIMI/DeleteRate/Rq/ExchangeDate	Time*	In	Business date when the future exchange rate is active ( <b>TWO</b> time zone)
FIMI/DeleteRate/Rq/FromCurrency	Int	In	1st currency code. Only the specified pair of currencies is deleted. To delete the reverse rate, repeat the operation, at that, specify the codes of currencies of the reverse rate.
FIMI/DeleteRate/Rq/ToCurrency	Int	In	2nd currency code
FIMI/DeleteRate/Rq/InstName	Str(4)	In	Name of financial institution. If NULL, the institution of FIMI terminal is used
FIMI/DeleteRate/Rq/SettingDate	Time	In	Calendar date and time of the future rate (in <b>TWO</b> time zone). If not defined, the ExchangeDate value is used

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/DeleteRate/Rq/Rates	ArrRec	In	<pre>{     int* RateGroup;     int* FromCurrency;     int* ToCurrency;     str(4) InstName;     time SettingDate; }</pre> <p>[] – currency rates being deleted.</p> <p>For the fields description, see the description of similar fields of the current request</p> <p>If specified, the duplicated fields are ignored.</p>
FIMI/DeleteRate/Rp/ MinExchangeDate	Time	Out	Minimum available date of the new exchange rate. It is defined together with Response= InvalidExchangeDate (see p. 3.1) if the request contains invalid date of the future exchange rate

### 2.3.3.6 Information from Back-Office:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetBackOfficeInfo – request for information from Back-Office</b>			
FIMI/GetBackOfficeInfo/Rq/IdentType	Int*	In	<p>Type of the object identifier:</p> <p>0 – contract number 1 – account number 2 – card “PAN-MBR” 3 – private customer ID 4 – customer ExtId (external Id).</p> <p>Identifiers are listed below. The other identifiers may transfer via TWO under the agreement between Back-Office and the FIMI terminal – such identifiers should not be equal to 0, 1, 2, 3, 4.</p>
FIMI/GetBackOfficeInfo/Rq/Ident	Str(30)*	In	Object identifier
FIMI/GetBackOfficeInfo/Rq/InfoType	Str(30)	In	<p>Requested information type.</p> <p>To obtain the list of supported requests, transfer Null. Then the list will be transferred in the response</p>
FIMI/GetBackOfficeInfo/Rq/Format	Int(1)*	In	<p>Response format:</p> <p>1 – XML 2 – Text</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetBackOfficeInfo/Rq/AddParams	Str(4000)	In	Request additional parameters  For SOAP XML: the field can contain the data in the XML format; therefore, the value must be screened by the protocol rules.
FIMI/GetBackOfficeInfo/Rp/Value	StrXML	Out	Obtained value

### 2.3.3.7 Messaging Service:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetCMSBuffer – request for log of messages to be sent to customer</b>			
FIMI/GetCMSBuffer/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/GetCMSBuffer/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/GetCMSBuffer/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/GetCMSBuffer/Rq/Address	Str(500)	In	Arranged address;  For e-mail:  the field should contain the data like: 'j.smith@domain.com';  For the mobile phone:  the field should contain the data like: '19012345678';
FIMI/GetCMSBuffer/Rq/FromTime	Time	In	Scheduled start time for sending messages
FIMI/GetCMSBuffer/Rq/ToTime	Time	In	Scheduled end time for sending messages
FIMI/GetCMSBuffer/Rq/Count	Int(4)	In	Number of the requested messages (NULL – all found under the specified conditions) but not more than 1000

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCMSBuffer/Rp/Buffer	ArrRec	Out	<pre>{     Int(1) Originator;     time* SendTime;     int(1)* Status;     int(3)* Priority;     str(500) CNSAddress;     str(20)* Channel;     str(500)* Address;     str(1000) Subject;     str Message;     str(19) PAN;     int(3) MBR;     int CardProfile;     str(20) Scheme;     int SchlItemCode;     str(64) CardUID;     int(16) PersonId.  } [] – list of messages sorted by the sending time descending order; Status – message status: 0 – prepared 1 – being sent For the description of the other fields, see GetCMSCommands.</pre>

**GetCMSArchive – request for archive of messages sent to customer**

FIMI/GetCMSArchive/Rq/Id	Int	In	Message ID
FIMI/GetCMSArchive/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/GetCMSArchive/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN field transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/GetCMSArchive/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/GetCMSArchive/Rq/Address	Str(500)	In	Arranged address For the parameter description, see the GetCMSBuffer operation.
FIMI/GetCMSArchive/Rq/FromTime	Time	In	Message sending start time
FIMI/GetCMSArchive/Rq/ToTime	Time	In	Message sending end time
FIMI/GetCMSArchive/Rq/LastId	Int	In	Identifier of the last received message; if it is defined in the request, all the messages satisfying the conditions (Id>LastId) are returned in the response

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCMSArchive/Rq/Count	Int(4)	In	Number of the requested messages (NULL – all found on the defined conditions) but not more than 1000
FIMI/GetCMSArchive/Rp/Archive	ArrRec	Out	<pre>{     int* Id;     int(1)* Status;     int(1) Originator;     time* SendTime;     int(3)* Priority;     time ScheduleTime;     str(500) CNSAddress;     str(20)* Channel;     str(500)* Address;     str(1000) Subject;     str Message;     str(19) PAN;     int(3) MBR;     int CardProfile;     str(20) Scheme;     int SchlItemCode;     int(2) MessageState;     time DeliveryTime;     str(64) CardUID;     str(3) NetworkErrorCode;     int(16) PersonId.  } [] – list of messages sorted by the sending time descending order; Status – message status: 0 – sent 1 – removed 2 – declined 3 – delivered 4 – expired ScheduleTime – scheduled delivery time NetworkErrorCode – network error code For the description of the other fields, see GetCMSCommands.</pre>

#### **GetCMSCommands – request for log of commands received from customer**

FIMI/GetCMSCommands/Rq/Id	Int	In	Message ID
FIMI/GetCMSCommands/Rq/PAN	Str(19)	In	Card number
FIMI/GetCMSCommands/Rq/MBR	Int(3)	In	Card member number
FIMI/GetCMSCommands/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetCMSCommands/Rq/Address	Str(500)	In	Composed address: See operation GetCMSBuffer for the parameter description
FIMI/GetCMSCommands/Rq/Archive FromTime	Time	In	Message receiving start time

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCMSCommands/Rq/ToTime	Time	In	Message receiving end time
FIMI/GetCMSCommands/Rq/LastId	Int	In	Identifier of the last received message; if it is defined in the request, all the messages satisfying the conditions (Id>LastId) are returned in the response
FIMI/GetCMSCommands/Rq/Count	Int(4)	In	Number of the requested messages (NULL – all found on the defined conditions) but not more than 1000
FIMI/GetCMSCommands/Rp/Commands	ArrRec	Out	<pre>{     int* Id;     int Code;     int(3)* Status;     int(3)* DeclineReason;     time OrigTime;     time ReceiveTime;     time SendTime;     str(255) MessageId;     int ProcessorType;     int ReplyId;     str(500) CNSAddress;     str(20)* Channel;     str(1500)* Address;     int ReplySrcPort;     int ReplyDstPort;     str(20) Scheme;     str(1000) SourceAddress;     int ReceiptSrcPort;     int ReceiptDstPort;     str(160) CmdName;     str(200) CmdText;     str(19) PAN;     int MBR;     int CardProfile;     str(64) CardUID; }</pre> <p>[]} – list of messages sorted by the sending time descending order:</p>

Code – command code (see p. 3.15).

Status – message status (0 – received; 1 – processed; 2 – executed; 3 - skipped).

DeclineReason – request decline reason (see p. 3.16).

OrigTime – original time.

ReceiveTime – receiving time.

MessageId – message ID assigned by the SMS Center.

ProcessorType – processor type (0 – template; 1 – algorithmix function).

ReplyId – identifier of the response to the command in the archive.

ReplySrcPort – host response source port.

ReplyDstPort – destination port of the abonent terminal response.

SourceAddress – sender address.

ReceiptSrcPort – source port of the abonent terminal command.

ReceiptDstPort – host command destination port.

CmdName – command name.

CmdText – text of the command.

Originator – message originator:

0 – generator

1 – authorizer

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description				
<p>2 – Refresh          3 – bank messaging unit          4 – broadcast          5 – test message          6 – response to the command          7 – resending a deleted message</p> <p>SendTime – sending time.          Priority – priority (0 – low; 1 – below normal; 2 – normal; 3 – above normal; 4 – high).          CNSAddress – abonent address.          Address – composed abonent address.          Subject – composed subject.          Message – composed message.          Channel – messaging channel identifier (for the list of available values for the channel, see field <i>InitSession/Rp/CNSChannelList.Name</i>).          Scheme – messaging scheme identifier (for the list of scheme available values, see field <i>InitSession/Rp/CNSSchemeList.Name</i>).          SchItemType – scheme item types (for the list of available values, see structure <i>GetSchemeItemTypes/Rp/SchemeItemTypes</i>).          SchItemCode – scheme item code (for the list of available values, see structure <i>GetSchemeItemTypes/Rp/SchemeItemTypes/ItemCode</i>).          CardProfile – card profile (for the list of available values, see field <i>FIMI/InitSession/Rp/CardProfileList</i>)          MessageState:          0 – message scheduled for delivery.          1 – message is being delivered.          2 – message is delivered.          3 – message validity period has expired.          4 – message has been deleted.          5 – message cannot be delivered.          6 – message is accepted for delivery.          7 – message is in the unknown state.          8 – message is in the rejected state.          9 – message was accepted but not transmitted.          DeliveryTime – message delivery time.          CardUID – unique card identifier.          PersonId – customer ID.</p> <p>Note:</p> <p>In the parameters PAN, CNSAddress, Address, the mask can be defined in the format of the SQL-operator 'like', i.e. using the following metacharacters: '_' (one character), '%' (any number of characters). E.g.: to request the cards with the rightmost characters '5678', it is necessary to pass '%5678' in the field.</p> <p style="text-align: center;"><b>GetSchemeItemTypes – request for list of available scheme items</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">FIMI/GetSchemeItemTypes/Rq/ InstName</td> <td style="padding: 5px;">Str(4)</td> <td style="padding: 5px;">In</td> <td style="padding: 5px;">Financial institution name. If NULL, items of schemes of all institutions allowed for the FIMI terminal are returned.</td> </tr> </table>	FIMI/GetSchemeItemTypes/Rq/ InstName	Str(4)	In	Financial institution name. If NULL, items of schemes of all institutions allowed for the FIMI terminal are returned.			
FIMI/GetSchemeItemTypes/Rq/ InstName	Str(4)	In	Financial institution name. If NULL, items of schemes of all institutions allowed for the FIMI terminal are returned.				

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetSchemeItemTypes/Rq/Scheme	Str(20)	In	Name of the messaging scheme for which available items are returned (see the list of available values in <i>InitSession/Rp/CNSSchemeList.Name</i> ). If NULL, all items of all institution schemes are returned or if InstName = NULL, items of schemes of all institutions allowed for the FIMI terminal are returned.
FIMI/GetSchemeItemTypes/Rq/SchemeItemTypes	ArrRec	Out	{ str* InstName; str* Scheme; int* ItemType; str(500) ItemTypeName; int* ItemCode; str(500) ItemCodeName; } [] – list of available scheme items
<p>InstName – institution of the scheme item          Scheme – scheme name (for the list of available values, see field <i>InitSession/Rp/CNSSchemeList.Name</i>)          ItemType – ID of the scheme item type, all scheme items with IDs less than 100 are system items, items with IDs exceeding 100 are user ones          ItemTypeName – name of the scheme item type          ItemCode – scheme item code (0 – any)          It contains the following data depending on ItemType for system types of items:</p> <ul style="list-style-type: none"> <li>- 1,2 – transaction code (see p.3.8)</li> <li>- 3 – number of days left before the expiration date</li> <li>- 4 – PeriodType*100+Period, where PeriodType: 0-Daily, 1-Weekly, 2-Monthly            Period – offset in the period (days) (it is used if PeriodType&gt;0)</li> <li>- 5 – type of the bank message (1 – Telebank; 2 – CMS; 3 – ATM; 4 – FIMI)</li> <li>- 6 – direction (1 – increase; 2 – decrease)</li> <li>- 7 – form ID</li> <li>- 8 – direction (1 – bonus increase; 2 – bonus decrease; 3 – debt increase; 4 – debt decrease)</li> <li>- 9 – ID of currency rates group (see the <i>GetRatesGroups</i> request)</li> <li>- 10 – subscription state (1 – subscription is enabled; 2 – subscription is disabled; 3 – activation; 4 – deactivation)</li> <li>- 12 – transaction code (see p.3.8 for transactions with code 3** or 621)</li> <li>- 13 – organization account number / organization code in Back-Office (vendor account)</li> <li>- 14 – overdraft change direction (1 – increase; 2 – decrease)</li> <li>- 17 – number of days (0–1000)</li> </ul> <p>11,15,16 – not used, has the default value 0.          Not used for the user types of items (ItemType &gt; 100), has the default value 0.          ItemCodeName – item code name</p>			
<b>SendMessage – sending message</b>			
FIMI/SendMessage/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.
FIMI/SendMessage/Rq/MBR	Int(3)	In	For the Telebank customer ID, the PAN fields transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/SendMessage/Rq/CardUID	Str(64)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SendMessage/Rq/InstName	Str(4)	In	Name of the financial institution owning the customer  If not defined, the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/SendMessage/Rq/PersonId	Int(16)	In	Customer ID
FIMI/SendMessage/Rq/Channel	Str(20)*	In	Channel  The list of available values is returned in the <i>FIMI/InitSession/Rp/CNSChannelList</i> structure.
FIMI/SendMessage/Rq/Address	Str(500)*	In	Address
FIMI/SendMessage/Rq/TextMessage	Str*	In	Test message
FIMI/SendMessage/Rq/Priority	Int(1)*	In	Message sending priority: 0 – low 1 – below average 2 – average 3 – above average 4 – high

### 2.3.3.8 Issuer History:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetIssuerHistory – request for issuer history log</b>			
FIMI/GetIssuerHistory/Rq/PAN	Str(19)	In	Number of the card and card owner or Telebank customer ID.  For the Telebank customer ID, the PAN field transfers the Telebank customer ID and MBR field – the value 0. If MBR is not transferred, it is 0.
FIMI/GetIssuerHistory/Rq/MBR	Int(3)	In	Unique ID of the card/Telebank customer. It can stand for the card number or Telebank customer ID.
FIMI/GetIssuerHistory/Rq/CardUID	Str(64)	In	Character ID (name) of the financial institution owning the account  If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .  <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name – it is allowed to request an account for the institution with AllowedFIID.AccessMask.Bit[1]=1 or AllowedFIID.AccessMask.Bit[3]=1</i>
FIMI/GetIssuerHistory/Rq/Account	Str(30)	In	Account number

<b>Parameter</b>	<b>Type</b> (* - mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetIssuerHistory/Rq/AccountUID	Str(32)	In	Unique card identifier within a session. It can stand for the card number.
FIMI/GetIssuerHistory/Rq/PersonId	Int(16)	In	Customer ID
FIMI/GetIssuerHistory/Rq/Count	Int(4)	In	Number of the requested messages (NULL – all found under the defined conditions) but not more than 1000
FIMI/GetIssuerHistory/Rq/FromTime	Time	In	Start time, NULL – from the start
FIMI/GetIssuerHistory/Rq/ToTime	Time	In	End time, NULL – until the current moment
FIMI/GetIssuerHistory/Rq/TableName	Str(100)	In	Object that has been changed (see section 3.17.1).  If NULL, it can be any.
FIMI/GetIssuerHistory/Rq/ArrestId	Str(32)	In	Arrest ID
FIMI/GetIssuerHistory/Rp/Log	ArrRec	Out	<pre>{   Int(12)* Id;   str(100)* TableName;   int* ModifyMask;   str(1)* OperType;   time* TimeStamp;   str(200)* UserName;   str(200)* OS_UserName;   str(200)* Terminal;   str(100) SessionClient;   str(30) FIMIClerk;   str(1000) Reason;   int(12) TranId;   str(19) PAN;   int(3) MBR;   str(4)* InstName;   str(30) Account;   int LimitId;   str(30) BackPaymentId;   str(20) CMSChannel;   str(500) CMSAddress;   int EMVTag;   int EMVScript;   str(64) CardUID;   str(32) AccountUID;   int FIID;   int(16) PersonId;   str(100) UserField;   str(120) ConfidentialWhat;   str(19) Token;   str(32) ArrestId; }</pre> <p>} [] – list of changes on the issuer object sorted by the time descending order;</p>
Id – event identifier. TableName – object which has been modified (see p. 3.17.1).			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
ModifyMask – bit mask of the modified attributes. The mask decryption depends on the contents of the TableName field. For the bit value, see p. 3.17.2.			
OperType – operation type (I – add; D – delete; U – update).			
TimeStamp – operation time.			
UserName – user name in Oracle.			
OS_UserName – user name in the operating system.			
Terminal – terminal of the operating system from which the operation was initiated.			
SessionClient – Oracle session client; Available values:			
‘unit #xxx’ – changes have been made by the <b>TWO</b> unit with the xxx identifier.			
‘instance #xxx’ – changes have been made by the <b>TWO</b> instance with the xxx identifier.			
NULL – changes have been made from the operators’ applications or via the direct access to the database.			
FIMIClerk – FIMI operator name.			
Reason – the reason of operation.			
TranId – transaction identifier.			
PAN – card number.			
MBR – card member number.			
InstName – name of the financial institution owning the modified object.			
FIID – ID of the financial institution owning the modified object.			
Account – account number.			
LimitId – limit identifier (see p. 3.3).			
BackPaymentId – payment identifier assigned to the card in Back-Office.			
CMSChannel – messaging channel identifier.			
CMSAddress – abonent CMS address.			
EMVTag – EMV tag (see p. 2.3.3.9).			
EMVScript – EMV-script type (see p. 2.3.3.9).			
CardUID – unique card identifier.			
AccountUID – unique account identifier.			
PersonId – customer ID.			
UserField – user-defined field name.			
ConfidentialWhat – personal question of the customer.			
Token – card token.			
ArrestId – arrest ID.			

#### **GetIssuerHistoryDetails – details of issuer history log**

FIMI//GetIssuerHistoryDetailsRq/Id	Int(12)	In	Message identifier
FIMI//RpGetIssuerHistoryDetails/ Details	ArrRec	Out	{ Int(12)* Id; str(60)* ColName; str(4000) OldValue; str(4000) NewValue; } [] – list of the attributes changed for the issuer object;

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
			ColName – attribute name (see p. 3.17.2); OldValue – attribute old value; newValue – attribute new value;

### 2.3.3.9 EMV Scripts:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetEMVScripts – request for card EMV-scripts</b>			
FIMI/GetEMVScripts/Rq/PAN	Str(19)	In	Card number
FIMI/GetEMVScripts/Rq/MBR	Int(3)	In	Card member number
FIMI/GetEMVScripts/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetEMVScript/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/GetEMVScripts/Rp/EMVScripts	ArrRec	Out	<pre>{     int* SeqNo;     int(12) SentTranId;     int(2)* Type;     int Address;     str(100) Value;     int Template; } [] – array of structures;</pre> <p>List of the EMV-scripts:</p>

SeqNo – script sequence number. 1 – when inserting to an empty queue, further - max+1; SeqNo is always increasing, but strict sequence is not required.

SentTranId – transaction ID starting from which script has been sent. Not NULL – script has been sent; the confirmation is required, NULL – a new script to be sent to the card.

Type – Script type:

- 1 - Application block
- 2 - Card block
- 3 - Pin unlock
- 4 - Put data
- 5 - Update record
- 6 - Application unlock
- 7 – PIN change

Template – address of the template on card to write the data to.

Available values:

- NULL, 0 – card field,
- 0xBF02 - Cumulative Counter Limits,
- 0xBF08 - Counter Limits,
- 0xBF0F - Limit Records;

Address - address of the field on card or address of the field in the card script template (available for *Put Data* script) to write the data to. It is transferred in decimal form. The format depends on the script type:

- For *Update Record* - the most significant byte is *SFI* (0-16), the least significant byte is *Record Number*,
- For *Put Data* - it is a list of available values of card or template addresses. If the template address is not defined (Template=NULL), it is the card field; otherwise – card template field.

Parameter		Type (* - mandatory parameter)	Direction (relative to TWO)	Description
Template (Hex)	Address (Hex)	Name		Field Value Format
<b>VISA</b>				
–	0x9F53	Consecutive Transaction Limit (International-Currency)		Decimal
–	0x9F54	Cumulative Total Transaction Amount Limit		Decimal
–	0x9F58	Lower Consecutive Offline Limit		Decimal
–	0x9F59	Upper Consecutive Offline Limit		Decimal
–	0xBF5B	Application Internal Data Template		Hex
–	0x9F5C	Cumulative Total Transaction Amount Upper Limit		Decimal
–	0x9F72	Consecutive Transaction Limit (International-Country)		Decimal
–	0x9F73	Currency Conversion Factor		Decimal
–	0x9F75	Cumulative Total Transaction Amount Limit (Dual Currency)		Decimal
–	0x9F77	VLP Funds Limit		Decimal
–	0x9F78	VLP Single Transaction Limit		Decimal
<b>Mastercard</b>				
–	0xC3	CIAC – Decline		Hex
–	0xC4	CIAC – Default		Hex
–	0xC5	CIAC – Online		Hex
–	0xC7	CDOL1 Related Data Length		Hex
–	0xC8	CRM Country Code		Decimal
–	0xC9	CRM Currency Code		Decimal
–	0xCA	Lower Cumulative Offline Transaction Amount		Decimal
–	0xCB	Upper Cumulative Offline Transaction Amount		Decimal
–	0xD1	Currency Conversion Table		Hex
–	0xD3	Additional Check Table		Hex
–	0xD5	Application Control		Hex
–	0xD7	Application Control (Contactless)		Hex
–	0xD6	Default ARPC Response Code		Hex
–	0xD9	Application File Locator (Contactless)		Hex
–	0x9F14	Lower Consecutive Offline Limit		Decimal
–	0x9F23	Upper Consecutive Offline Limit		Decimal
–	0xDF07	Linked Application Data		Hex
–	0xDF11	Advance Accumulator 1 Control (Contact)		Hex
–	0xDF1A	Advance Counter 1 Control (Contact)		Hex
–	0xDF30	Interface Enabling Switch		Hex
<b>NSPK</b>				
–	0xC3	Offline transaction days limit		Decimal
0xBF02	0xD1	Cumulative Counter 1 Limits		Decimal
0xBF02	0xD2	Cumulative Counter 2 Limits		Decimal
0xBF08	0xD1	Counter 1 Limits		Hex
0xBF08	0xD2	Counter 2 Limits		Hex
0xBF08	0xD3	Counter 3 Limits		Hex
0xBF0F	0xD1	Limit 1 Record		Decimal
0xBF0F	0xD2	Limit 2 Record		Decimal
0xBF0F	0xD3	Limit 3 Record		Decimal
0xBF0F	0xD4	Limit 4 Record		Decimal
0xBF0F	0xD5	Limit 5 Record		Decimal
0xBF0F	0xD6	Limit 6 Record		Decimal
0xBF0F	0xD7	Limit 7 Record		Decimal
0xBF0F	0xD8	Limit 8 Record		Decimal
0xBF0F	0xD9	Limit 9 Record		Decimal
0xBF0F	0xDA	Limit 10 Record		Decimal
0xBF0F	0xDB	Limit 11 Record		Decimal
0xBF0F	0xDC	Limit 12 Record		Decimal
0xBF0F	0xDD	Limit 13 Record		Decimal
0xBF0F	0xDE	Limit 14 Record		Decimal

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
Value – field value depends on the script type: For <i>Put Data</i> - an integer number in the decimal or hexadecimal form; the format corresponds to the field format for the certain address of the card field and address of the template field. For details, refer to specifications on EMV applications. For <i>Update Record</i> - a random hexadecimal string			
<b>AddEMVScript – adding EMV-script</b>			
FIMI/AddEMVScript/Rq/PAN	Str(19)	In	Card number
FIMI/AddEMVScript/Rq/MBR	Int(3)	In	Card member number
FIMI/AddEMVScript/Rq/CardUID	Str(64)	In	Unique card identifier within a session. It can stand for the card number.
FIMI/AddEMVScript/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/AddEMVScript/Rq/Type	Int(2)*	In	Script type
FIMI/AddEMVScript/Rq/Template	Int	In	Address of the template on the card to write the data to – can be defined for the <i>Put Data</i> scripts. It is transferred in decimal form.
FIMI/AddEMVScript/Rq/Address	Int	In	Address of the field on the card to write the data to – can be defined for the scripts <i>Put Data</i> and <i>Update Record</i> . It is transferred in decimal form.
FIMI/AddEMVScript/Rq/Value	Str(100)	In	Field value – can be specified for the scripts <i>Put Data</i> , <i>Update Record</i> and <i>PIN Change</i> . For the <i>PIN Change</i> script, it contains PIN Block in the <i>ANSI X9.8</i> format ( <i>ISO Format 0</i> ) encrypted under ZPK, working key or LMK
FIMI/AddEMVScript/Rq/KeyId	Int	In	ID of the working key It must be filled if the <i>Value</i> field contains the PIN Block encrypted under the working key; otherwise, NULL
FIMI/AddEMVScript/Rq/KeyType	Int(1)	In	Type of the PIN block encryption key: 0 or NULL – ZPK; 1 – FIMI driver working key; it can be left undefined as the <i>KeyId</i> field is filled; 2 – cryptoserver LMK.
For the description of format and a list of available values for the <i>Type</i> , <i>Template</i> , <i>Address</i> and <i>Value</i> fields, refer to the description of the <i>GetEMVScripts</i> operation.			
<b>UpdateEMVScript – updating EMV-script</b>			
FIMI/UpdateEMVScript/Rq/PAN	Str(19)	In	Card number
FIMI/UpdateEMVScript/Rq/MBR	Int(3)	In	Card member number
FIMI/UpdateEMVScript/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/UpdateEMVScript/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateEMVScript/Rq/SeqNo	Int*	In	Script sequence number
FIMI/AddEMVScript/Rq/Type	Int(2)*	In	Script type
FIMI/AddEMVScript/Rq/Template	Int	In	Address of the template on the card to write the data to – can be defined for the <i>Put Data</i> scripts. It is transferred in decimal form.
FIMI/UpdateEMVScript/Rq/Address	Int	In	Address of the field on the card to write the data to – can be defined for the scripts <i>Put Data</i> and <i>Update Record</i> . It is transferred in decimal form.
FIMI/UpdateEMVScript/Rq/Value	Str(100)	In	Field value – can be defined for the scripts <i>Put Data</i> and <i>Update Record</i>

For the description of format and a list of available values for the *Type*, *Template*, *Address* and *Value* fields, refer to the description of the *GetEMVScripts* operation.

#### **DeleteEMVScripts – removing EMV-script**

FIMI/DeleteEMVScripts/Rq/PAN	Str(19)	In	Card number
FIMI/DeleteEMVScripts/Rq/MBR	Int(3)	In	Card member number
FIMI/DeleteEMVScripts/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/DeleteEMVScript/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/DeleteEMVScript /Rq/SeqNo	Int*	In	Script sequence number

#### **SetEMVScriptSeqNo – setting EMV-script sequence number**

FIMI/SetEMVScriptSeqNo/Rq/PAN	Str(19)	In	Card number
FIMI/SetEMVScriptSeqNo/Rq/MBR	Int(3)	In	Card member number
FIMI/SetEMVScriptSeqNo/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/SetEMVScriptSeqNo/Rq/PersonId	Int(16)	In	Customer ID Reserved, not used for processing
FIMI/SetEMVScriptSeqNo/Rq/SeqNo1	Int*	In	Sequence numbers of the EMV-scripts to be exchanged
FIMI/SetEMVScriptSeqNo/Rq/SeqNo2	Int*	In	

#### **2.3.3.10 Bank Messages Offline Exchange**

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>BMSGetForms – request for list of forms and groups</b>			
FIMI/BMSGetForms/Rq/PAN	Str(19)	In	Card number

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/BMSGetForms/Rq/MBR	Int(3)	In	Card member number
FIMI/BMSGetForms/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/BMSGetForms/Rq/PersonId	Int(16)	In	Identifier of the private customer. The parameter can be specified instead of PAN. If PAN and Customer Identifier are not specified, the forms and groups are searched for by FIMI/Clerk. If personId is specified, the InstName must be also defined.
FIMI/BMSGetForms/Rq/InstName	Str(4)	In	Character identifier (name) of financial institution
FIMI/BMSGetForms/Rq/What	Int(1)	In	Request for: 1 – forms 2 – form groups 3 – all If not defined, it is = 3
FIMI/BMSGetForms/Rq/GroupId	Int	In	Group ID If defined, the forms of the current group and/or child group are returned; otherwise – all forms/groups.
FIMI/BMSGetForms/Rq/Direction	Int(1)	In	Direction of messages generated by the form: 1=to bank, 2=from bank. If defined, the forms with the defined direction are returned; otherwise – forms with all the directions.
FIMI/BMSGetForms/Rq/Language	Int	In	ID of language to return the headers of the forms and/or form groups: FIMI/BMSGetForms/Rp/Forms/Title, FIMI/BMSGetForms/Rp/Groups/Title. If the headers for the required language are not found, the standard headers will be returned. The language code is taken from the <b>TWO</b> dictionary. If the field is not defined, the standard headers will be returned.
FIMI/BMSGetForms/Rp/Forms	ArrRec	Out	{     int *Id;     str(100) *Name;     int(1)* Direction;     str(100) Title;     str Description;     int GroupId;   } [] – a list of forms: Id – form ID

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
			Name – form name (char ID) Direction: 1=to bank, 2=from bank Title – form title to be shown to customer Description – form description (detailed description) GroupId – ID of form group.
FIMI/BMSGetForms/Rp/Groups	ArrRec	Out	{ int *Id; int ParentId; str(600) Title; } } [] – form groups: Id – group ID ParentId – parent group ID (NULL-no parent) Title – group title
<b>BMSGetFormFields – request for form fields</b>			
FIMI/BMSGetFormFields/Rq/PAN	Str(19)	In	Card number
FIMI/BMSGetFormFields/Rq/MBR	Int(3)	In	Card member number
FIMI/BMSGetFormFields/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/BMSGetFormFields/Rq/PersonId	Int(16)	In	Identifier of the private customer. If personId is specified, the InstName must be also defined.
FIMI/BMSGetFormFields/Rq/InstName	Str(4)	In	Character identifier (name) of financial institution
FIMI/BMSGetFormFields/Rq/FormId	Int*	In	Form ID
FIMI/BMSGetFormFields/Rq/Language	Int	In	ID of language to return the field descriptions FIMI/BMSGetFormFields/Rp/Fields/Description. If the headers for the required language are not found, the standard field descriptions will be returned. The language code is taken from the TWO dictionary. If the field is not defined, the standard field descriptions will be returned.
FIMI/BMSGetFormFields/Rp/Fields	ArrRec	Out	{ str(100) *FieldName; str(255) Description; int *ValType; num *MinVal; num *MaxVal; int *AllowedChar; int *Mandatory; CustAcct DefaultValue;

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
			} [] – a list of form fields:
Description – field description (field purpose, filling rules, ...)			
ValType – value type: 1-Int 2-Num 3-Str, 4-Memo, 5-CheckBox, 6-ComboBox, 7-DateTime, 8-RadioButton, 9-File (CheckBox format: when transferring, the field takes on the value 0 or 1 ComboBox: the field is filled as follows: <string to be shown to customer>[=<digital value replacing the string>]<separator> separator – '0x0A'. The field is defined as a string with a set of values.			
Sample: US dollar=840			
RadioButton: the field is filled as follows - < string to be shown to customer >[=< digital value replacing the string >]< separator > separator – '0x0A'. The field is defined as a string with a set of values.			
Sample: Selection1=1 Selection2=2			
DateTime type – when transferring, the field is filled with the symbols from 1 to 9 in the ddmmmyyhhmiss format and is transferred as string. dd – day (2 symbols), mm – month (2 symbols), yy - year (2 symbols), hh – hours in twenty-four hour format (2 symbols), mi - minutes (2 symbols), ss – seconds (2 symbols), hh,mi,ss – optional. Sample – 220205161345 - February 22, 2005, 16:13:45			
MinVal, MaxVal – only for Form.Direction=1; for Int,Num – min/max value. For strings – min/max length			
AllowedChar – 1–all: 2–numbers only, 3–alphabetic characters.			
Mandatory – check whether the field is mandatory. The default value is 0 (do not check).			
DefaultValue – field default value			

#### **BMSGetMessages – request for customer messages**

FIMI/BMSGetMessages/Rq/PAN	Str(19)	In	Card number. The customer identifier can be specified instead of this parameter. If the card number and customer identifier are not specified, the messages are searched by the FIMI/Clerk parameter
FIMI/BMSGetMessages/Rq/MBR	Int(3)	In	Card member number
FIMI/BMSGetMessages/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/BMSGetMessages/Rq/PersonId	Int(16)	In	Identifier of the private customer. The parameter can be specified instead of the card number. If the card number and customer identifier are not specified, the messages are searched by the FIMI/Clerk parameter  If personId is specified, the InstName must be also defined.
FIMI/BMSGetMessages/Rq/InstName	Str(4)	In	Character identifier (name) of financial institution

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/BMSGetMessages/Rq/Id	Int	In	Message ID  If the parameter is not specified, other fields of the request are not analyzed.
FIMI/BMSGetMessages/Rq/Direction	Int(1)	In	1 = to bank  2 = from bank.  If not defined, all the messages are returned
FIMI/BMSGetMessages/Rq/Status	Int(1)	In	The status of the requested messages. Available values:  1 – Unread if Direction=2 2 – Read if Direction=2  If not defined, all the messages are returned.  (direction=2, status=empty - all the messages from bank) (direction=1 status=empty - all the messages to bank)
FIMI/BMSGetMessages/Rq/FromTime	Time	In	Message creation start time
FIMI/BMSGetMessages/Rq/ToTime	Time	In	Message creation end time
FIMI/BMSGetMessages/Rq/Title	Str(150)	In	Message subject or a part of it. It is possible to apply the “like” mask of SQL statement.
FIMI/BMSGetMessages/Rq/Count	Int	In	Number of requested messages.  If not defined, all the messages are selected. Count>0
FIMI/BMSGetMessages/Rq/FormId	Int	In	Form ID  If defined, all the messages of the current form are returned.
FIMI/BMSGetMessages/Rp/Messages	ArrRec	Out	{ int *Id; int ParentId; int *Status; int *Direction; int *FormId; str(100) FormTitle; time SentTime; time DeliverTime; str(150) Title; str (100) NextMessList; str(100) FormName; } } [] – a list of customer messages:

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
Id – message ID. Status – message status: 1= Delivered 2= Approved 4= Sent 5= Read FormId – form ID. FormTitle – message form title (it can stand for the message subject). SentTime – message creation time. DeliverTime – time needed to deliver the message from bank to <b>TWO</b> (for Direction=2). ParentId – parent message ID (request to response link). It is filled for Direction=2. Title – message subject. NextMessList – a list of response identifiers separated by comma. FormName – form name.			
<b>BMSGetMessageFields – request for message fields</b>			
FIMI/BMSGetMessageFields/Rq/PAN	Str(19)	In	Card Number. The customer identifier can be specified instead of this parameter. If the card number and customer identifier are not specified, the message fields are searched by the FIMI/Clerk parameter
FIMI/BMSGetMessageFields/Rq/MBR	Int(3)	In	Card member number
FIMI/BMSGetMessageFields/Rq/Card UID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/BMSGetMessageFields/Rq/PersonId	Int(16)	In	Identifier of the private customer. The parameter can be specified instead of the card number. If the card number and customer identifier are not specified, the message fields are searched by the FIMI/Clerk parameter  If personId is specified, the InstName must be also defined.
FIMI/BMSGetMessageFields/Rq/InstName	Str(4)	In	Character identifier (name) of financial institution
FIMI/BMSGetMessageFields/Rq/MessId	Int*	In	Message ID
FIMI/BMSGetMessageFields/Rp/Fields	ArrRec	Out	{ str(100) *Name; str Value; int *Type; str(9999999999) FileVal; str(100) FileExtension; } [] – list of message fields: Name – field name as defined in form

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<p>Value – field value in the message. It is defined for all fields except for the fields with Type=9 (File);</p> <p>Type – field type (see description of the <i>BMSGetFormFields</i> operation)</p> <p>FileVal – value of the field with the type 9 (File) in the base64 encoding</p> <p>FileExtension – extension (type) of file for the fields with Type=9 (File); it contains the string of the type “BMP”, “JPG”, “PDF”, etc.</p>
<b><i>BMSReadMessages</i> – marking bank messages as read</b>			
FIMI/BMSReadMessages/Rq/Id	Int[...]*	In	Array of bank message identifiers
<b><i>BMSSendMessage</i> – sending message to bank</b>			
FIMI/BMSSendMessage/Rq/PAN	Str(19)	In	Card number. The customer identifier can be specified instead of this parameter. If the card number and customer identifier are not specified, the sender from the FIMI/Clerk header parameter is used
FIMI/BMSSendMessage/Rq/MBR	Int(3)	In	Card member number
FIMI/BMSSendMessage/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/BMSSendMessage/Rq/PersonId	Int(16)	In	Identifier of the private customer. The parameter can be specified instead of the card number. If the card number and customer identifier are not specified, the sender from the FIMI/Clerk header parameter is used  If personId is specified, the InstName must be also defined.
FIMI/BMSSendMessage/Rq/InstName	Str(4)	In	Character identifier (name) of financial institution
FIMI/BMSSendMessage/Rq/FormId	Int	In	Form ID
FIMI/BMSSendMessage/Rq/FormName	Str(100)	In	Form name
FIMI/BMSSendMessage/Rq>Title	Str(150)	In	Message subject
FIMI/BMSSendMessage/Rq/MessField	ArrRec*	In	<pre>{     str(100)* Name;     str Value;     str(9999999999) FileVal;     str(100) FileExtension; }</pre> <p>} [] – a list of message fields: Name – field name as defined in form</p>

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
			<p>Value – field value in the message, for all the fields except for the fields of the File type.</p> <p>FileVal – value of the field of the type 9 (File) in the base64 encoding;</p> <p>FileExtension – extension (type) of file for the field with Type=9 (File); it contains the string of the type "BMP", "JPG", "PDF", etc.</p>
FIMI/BMSSendMessage/Rq/ParentId	Int	In	<p>Parent message ID. Optional.</p> <p>It is defined for responses to the messages from bank to customer.</p> <p>E.g., it is defined in case it is message repeated by customer in response to bank message where the customer previous application was not satisfied.</p>
FIMI/BMSSendMessage/Rp/MessId	Int	Out	Message ID
FIMI/BMSSendMessage/Rp/ DeclineCode	Int	Out	<p>Code of the message decline reason (negative value):</p> <ul style="list-style-type: none"> <li>-100-message status is invalid</li> <li>-101-invalid direction</li> <li>-102-number of fields in the message does not match the one defined in the template</li> <li>-103-error by checking the min or max value</li> <li>-104- error by checking the allowed symbols</li> <li>-105- error by checking the field by the algorithmix function</li> <li>-106- error by checking the entire message by the algorithmix function</li> <li>-107-message is not found</li> <li>-108-field is mandatory</li> </ul>
FIMI/BMSSendMessage/Rp/ DeclineField	Str(100)	Out	Name of field where the message check error occurred

## 2.3.3.11 Group Requests:

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>CreateIssuerObjects – issuer objects creation</b>			
FIMI/CreateIssuerObjects/Rq/InstName	Str(4)*	In	Institution name
FIMI/CreateIssuerObjects/Rq/PersonBranchId	Str(90)	In	Customer branch ID
FIMI/CreateIssuerObjects/Rq/PersonFIO	Str(300)	In	Full name of a customer
FIMI/CreateIssuerObjects/Rq/PersonVIP	Int(1)	In	“VIP” attribute The field value is mandatory if the customer is created within the operation.
FIMI/CreateIssuerObjects/Rq/PersonSex	Str(1)	In	Sex M – male F – female N – not specified U – unknown The field value is mandatory if the customer is created within the operation.
FIMI/CreateIssuerObjects/Rq/PersonIdentType	Int	In	Customer identification type See a list of available values in the GetPersonInfo/Rp/Info field.
FIMI/CreateIssuerObjects/Rq/PersonIdentity	Str(120)	In	Identifier, number of the identity document
FIMI/CreateIssuerObjects/Rq/PersonPassword	Str(60)	In	Secret word
FIMI/CreateIssuerObjects/Rq/PersonBirthDay	Time	In	Date of birth
FIMI/CreateIssuerObjects/Rq/PersonBirthPlace	Str(250)	In	Place of birth
FIMI/CreateIssuerObjects/Rq/PersonResidentCountry	Int	In	Resident country
FIMI/CreateIssuerObjects/Rq/PersonResidentCityInLatin	Str(100)	In	Resident city (in Latin characters)
FIMI/CreateIssuerObjects/Rq/PersonAddressInLatin	Str(200)	In	Address (in Latin characters)
FIMI/CreateIssuerObjects/Rq/PersonPostalCode	Str(30)	In	Postal code
FIMI/CreateIssuerObjects/Rq/PersonId	Int(16)*	In	ID of the created customer
FIMI/CreateIssuerObjects/Rq/PersonPhoneNumber	Str(20)	In	Customer phone number

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																					
FIMI/CreateIssuerObjects/Rq/PersonEmail	Str(300)	In	E-mail																					
FIMI/CreateIssuerObjects/Rq/PersonExtId	Str(50)	In	External customer ID																					
FIMI/CreateIssuerObjects/Rq/PersonUserFields	ArrRec	In	<pre>{   Str(100)* Name;   Str(3800) TextValue. }</pre> <p>[] – array of structures with the user fields:      Name – field name      TextValue – field text value      The total length of the <i>Name</i> and <i>TextValue</i> fields in all the records of the array <i>FIMI/CreateIssuerObjects/Rq/UserFields</i> must not exceed the 3900 symbols.</p>																					
FIMI/CreateIssuerObjects/Rq/Account	Str(30)*	In	Account number																					
FIMI/CreateIssuerObjects/Rq/AcctType	Int	In	<p>Account type in TWO:</p> <table style="margin-left: 20px;"> <tr><td>1</td><td>–</td><td>checking</td></tr> <tr><td>11</td><td>–</td><td>savings</td></tr> <tr><td>31</td><td>–</td><td>credit</td></tr> <tr><td>91</td><td>–</td><td>bonus</td></tr> </table> <p>The field value is mandatory if the account is created within the operation.</p>	1	–	checking	11	–	savings	31	–	credit	91	–	bonus									
1	–	checking																						
11	–	savings																						
31	–	credit																						
91	–	bonus																						
FIMI/CreateIssuerObjects/Rq/AcctStatus	Int	In	<p>Account status:</p> <table style="margin-left: 20px;"> <tr><td>0</td><td>–</td><td>Inactive account</td></tr> <tr><td>1</td><td>–</td><td>Open</td></tr> <tr><td>2</td><td>–</td><td>Deposit only</td></tr> <tr><td>3</td><td>–</td><td>Open primary account</td></tr> <tr><td>4</td><td>–</td><td>Deposit only primary account</td></tr> <tr><td>5</td><td>–</td><td>Information only</td></tr> <tr><td>9</td><td>–</td><td>Closed</td></tr> </table> <p>The field value is mandatory if the account is created within the operation.</p>	0	–	Inactive account	1	–	Open	2	–	Deposit only	3	–	Open primary account	4	–	Deposit only primary account	5	–	Information only	9	–	Closed
0	–	Inactive account																						
1	–	Open																						
2	–	Deposit only																						
3	–	Open primary account																						
4	–	Deposit only primary account																						
5	–	Information only																						
9	–	Closed																						
FIMI/CreateIssuerObjects/Rq/AcctCurrency	Int	In	<p>ISO code of account currency.</p> <p>If not defined, the <i>Card Profile ID</i> field must be specified. In this case, the account currency is the currency of the card profile limits.</p>																					

<b>Parameter Name</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/CreateIssuerObjects/Rq/AcctUserFields	ArrRec	In	<pre>{     Str(100)* Name;     Str(3800) TextValue. }</pre> <p>[] – array of structures</p> <p>The total length of the <i>Name</i> and <i>TextValue</i> fields in all the records of the array <i>FIMI/CreateIssuerObjects/Rq/UserFields</i> must not exceed the 3900 symbols.</p>
FIMI/CreateIssuerObjects/Rq/AcctBranchId	Str(90)	In	ID of account branch
FIMI/CreateIssuerObjects/Rq/CardExpiration	Time	In	Virtual card expiration time in <b>TWO</b> time zone
FIMI/CreateIssuerObjects/Rq/NameOnCard	Str(250)	In	Customer name on card in Latin characters
FIMI/CreateIssuerObjects/Rq/CardLimitPeriodType	Int	In	Limit period type (see the description in <i>CreateVCard</i> )
FIMI/CreateIssuerObjects/Rq/CardLimitPeriod	Int	In	Limit period (see the description in <i>CreateVCard</i> )
FIMI/CreateIssuerObjects/Rq/CardLimitCurrency	Int	In	Currency of the card financial limit (if not defined, the default currency of FIMI terminal is used)
FIMI/CreateIssuerObjects/Rq/CardLimitAmount	Num	In	Max amount of purchase for the specified period
FIMI/CreateIssuerObjects/Rq/CardLimitCount	Int	In	Max number of purchases for the specified period
FIMI/CreateIssuerObjects/Rq/CardProfile	Int	In	Card profile identifier
FIMI/CreateIssuerObjects/Rq/CardUserFields	ArrRec	In	<pre>{     Str(100)* Name;     Str(3800) TextValue. }</pre> <p>[] – array of structures</p> <p>The total length of the <i>Name</i> and <i>TextValue</i> fields in all the records of the array <i>FIMI/CreateIssuerObjects/Rq/UserFields</i> must not exceed the 3900 symbols.</p>
FIMI/CreateIssuerObjects/Rq/CardBranchId	Str(90)	In	ID of card branch
FIMI/CreateIssuerObjects/Rq/CardStatus	Int	In	Card new status (see a list of available values in the field <i>InitSesion/Rp/CardStatList.Id</i> or section 3.2)

<b>Parameter Name</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/CreateIssuerObjects/Rq/CardProductId	Int	In	Card product ID For details, see <i>FIMI/CreateVCard/Rq/CardProductId</i>
FIMI/CreateIssuerObjects/Rq/CardExternalObjectId	Str(64)	In	Unique external ID of a card
FIMI/CreateIssuerObjects/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/CreateIssuerObjects/Rq/MBR	Int(3)	In	They are transferred if it is required to create a card with the specified number. If the card PAN is not transferred, they are generated automatically according to the settings of the card product in use. If MBR is transferred, value 0 is set.
FIMI/CreateIssuerObjects/Rq/Amount	Num*	In	Operation amount in the account currency. Account will be credited with this amount in <b>TWO</b>
FIMI/CreateIssuerObjects/Rq/CorrespAcct	Str(30)	In	Correspondent account number
FIMI/CreateIssuerObjects/Rq/OrigAmount	Num	In	Operation original amount
FIMI/CreateIssuerObjects/Rq/OrigCurrency	Int	In	Operation original currency
FIMI/CreateIssuerObjects/Rq/IgnoreImpact	Int(1)	In	=1 – operation is allowed during the Impact period =0 or NULL – operation is not allowed in the Impact period
FIMI/CreateIssuerObjects/Rq/UseBonusDebt	Int(1)	In	NULL or not 0 – bonus/debt is considered when crediting the account 0 – bonus/debt is not considered when crediting the account
FIMI/CreateIssuerObjects/Rp/ApprovalCode	Str(6)	Out	Approval code for the credit operation
FIMI/CreateIssuerObjects/Rp/PAN	Str(19)*	Out	Virtual card PAN
FIMI/CreateIssuerObjects/Rp/MBR	Int(3)	Out	Virtual card MBR = 0
FIMI/CreateIssuerObjects/Rp/CardUID	Str(64)	Out	Unique card identifier. It can stand for the card number.
FIMI/CreateIssuerObjects/Rp/CardExpiration	Time*	Out	Virtual card expiration time in <b>TWO</b> time zone (it can be corrected by <b>TWO</b> authorizer)
FIMI/CreateIssuerObjects/Rp/CVV2	Int(3)	Out	Virtual card CVV2. It is returned if the CVV2 calculation is allowed in the settings of the <b>TWO</b> authorization scheme for virtual cards.

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateIssuerObjects/Rp/StrCVV2	Str(3)	Out	Virtual card CVV2 in the string format. It is returned if the CVV2 calculation is allowed in the settings of the <b>TWO</b> authorization scheme for virtual cards.
FIMI/CreateIssuerObjects/Rp/CardLimitCurrency	Int*	Out	Financial limit currency
FIMI/CreateIssuerObjects/Rp/CardLimitAmount	Num	Out	Max amount of purchase for the specified period (it can be corrected by <b>TWO</b> authorizer)
FIMI/CreateIssuerObjects/Rp/CardLimitCount	Int	Out	Max number of purchases for the specified period (it can be corrected by <b>TWO</b> authorizer)
<p>The Card product settings in Back-Office define whether the plastic card will be issued or the card will remain virtual.</p> <p>Either CardProfile field or CardLimitAmount and CardLimitCount fields must be filled In the CreateIssuerObjects operation. If they are absent in the operation at the same time, the Internet card creation transaction will be declined at the authorization stage.</p>			

### 2.3.3.12 Issuer Fees:

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b><i>GetIssuerFeesForOwnTerm – request for issuer fees for own terminals</i></b>			
FIMI/GetIssuerFeesForOwnTerm/Rq/InstName	Str(4)	In	Institution name If not defined, the fees of all allowed financial institutions (including the FIMI terminal institution) will be selected.
FIMI/GetIssuerFeesForOwnTerm/Rq/IssuerObjectType	Int(1)	In	Type of issuer object: 0 – «any» 1 – card profile 2 – card profile template 3 – cardless transactions If NULL, it=0.
FIMI/GetIssuerFeesForOwnTerm/Rq/IssuerObjectld	Int	In	Identifier of issuer object. The field value is used if the <i>IssuerObjectType</i> field is set to 1 or 2. If <i>IssuerObjectType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation). If <i>CardProfileType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<p><i>GroupTemplates</i> field of the <i>InitSession</i> operation).</p> <p>If the field is not defined, all allowed fees are selected.</p> <p>If the CardProfileType value is 3, the field transfers the value -1.</p>
FIMI/GetIssuerFeesForOwnTerm /Rp/Fees	ArrRec	Out	<pre>{     Str(4)* InstName;     Int(1)* IssuerObjectType;     Int* IssuerObjectId;     Int(1)* AcqObjectType;     Int* AcqObjectId;     Int(2)* AcctType;     Int(4)* AcctCurrency;     Int(1)* CurrencyExchange;     Int(4)* TranCode;     Int* SeqNo;     Num* Flat;     Num(3,2)* Percentage;     Num* Minimum;     Num* Maximum;     Num* TierAmount;     Num* TieredFeeAmount;     Int(1)* BoundAmountTier;     Int(4)* FeeCurrency;     Int RateGroup;     Str(20) BackOfficeId;     Int OptionMask;     Int CounterId;     Int(1)* UseCounter. }</pre> <p>[] – a list of issuer fees.</p>
FIMI/GetIssuerFeesForOwnTerm/ Rq/ UserFields	ArrRec	In	<pre>{     Str(4)* InstName;     Int(1)* IssuerObjectType;     Int* IssuerObjectId;     Int(1)* AcqObjectType;     Int* AcqObjectId;     Int(2)* AcctType;     Int(4)* AcctCurrency;     Int(1)* CurrencyExchange;     Int(4)* TranCode;     Int* SeqNo;     Str(100)* Name;     Str(4000) TextValue;     Str XMLValue. }</pre> <p>[] – list of the user fee fields</p>

<b>Parameter Name</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
InstName – institution name			
IssuerObjectType – type of issuer object (for details, see the description of the <i>IssuerObjectType</i> field)			
IssuerObjectld – identifier of issuer object (for details, see the description of the <i>IssuerObjectld</i> field)			
AcqObjectType – type of acquiring object (1–acquiring group, 2– acquiring group template)			
AcqObjectld – identifier of acquiring object; if <i>AcqObjectType</i> =1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation); if <i>AcqObjectType</i> =2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation); if <i>AcqObjectld</i> = 0, it contains «any»			
AcctType – account type (0–«any».; 1–checking; 11–savings; 31–credit; 91–bonus)			
AcctCurrency – account currency (0–«any».; a list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation)			
CurrencyExchange – currency conversion (1–yes; 2–no; 0– Indifferently)			
TranCode – transaction code (see a list of available values in section 3.8)			
SeqNo – sequence			
Flat – fixed fee amount			
Percentage – percent of amount			
Minimum – minimum available fee amount			
Maximum – maximum available fee amount (0–unlimited)			
TierAmount – tier amount			
TieredFeeAmount – tiered fee amount			
BoundAmountTier – bound amount tier (1– left (bottom); 2– right (upper))			
FeeCurrency – limit currency (the list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation)			
RateGroup – group of currency rates (for the list of available values, see <i>FIMI/GetRateGroups/Rp&gt;List</i> )			
BackOfficeId – fee ID in Back-Office			
OptionMask – additional fee options (bit mask, 1 bit – hidden fee)			
CounterId – counter ID (see the list of available values in <i>FIMI/InitSession/Rp/Counters</i> )			
UseCounter – counter use type: 0 – not used 1 – do not charge fee if the counter has not reached its maximum value 2 – transaction amount plus change delta 3 – transaction amount minus change delta			
<b><i>SetIssuerFeesForOwnTerm – changing issuer fees for own terminals</i></b>			
FIMI/SetIssuerFeesForOwnTerm /Rq/InstName	Str(4)*	In	Institution name
FIMI/SetIssuerFeesForOwnTerm /Rq/IssuerObjectType	Int(1)*	In	Type of issuer object: 1 – card profile 2 – card profile template 3 – cardless transactions
FIMI/SetIssuerFeesForOwnTerm /Rq/IssuerObjectld	Int*	In	Identifier of issuer object. If <i>IssuerObjectType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation). If <i>IssuerObjectType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<i>GroupTemplates</i> field of the <i>InitSession</i> operation). If the <i>CardProfileType</i> value is 3, the field transfers the value -1.
FIMI/SetIssuerFeesForOwnTerm /Rq/AcqObjectType	Int(1)*	In	Type of acquiring object: 1–acquiring group 2–acquiring group template
FIMI/SetIssuerFeesForOwnTerm /Rq/AcqObjectld	Int*	In	Identifier of acquiring object. If <i>AcqObjectType</i> =1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation). If <i>AcqObjectType</i> =2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation). If <i>AcqObjectld</i> = 0, it contains «any».
FIMI/SetIssuerFeesForOwnTerm /Rq/AcctType	Int(2)*	In	Account type: 0–«any» 1–checking 11–savings 31–credit 91–bonus
FIMI/SetIssuerFeesForOwnTerm /Rq/AcctCurrency	Int(4)*	In	Account currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0–«any»
FIMI/SetIssuerFeesForOwnTerm /Rq/CurrencyExchange	Int(1)*	In	Currency conversion: 0– Indifferently 1–yes 2–no
FIMI/SetIssuerFeesForOwnTerm /Rq/TranCode	Int(4)*	In	Transaction code. See a list of available values in section 3.8.
FIMI/SetIssuerFeesForOwnTerm /Rq/SeqNo	Int*	In	Sequence
FIMI/SetIssuerFeesForOwnTerm /Rq/Flat	Num	In	Fixed fee amount
FIMI/SetIssuerFeesForOwnTerm /Rq/Percentage	Num(3,2)	In	Percent of amount
FIMI/SetIssuerFeesForOwnTerm /Rq/Minimum	Num	In	Minimum available fee amount The field value is more or equal to 0.

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetIssuerFeesForOwnTerm /Rq/Maximum	Num	In	Maximum available fee amount (0–unlimited) The field value is more or equal to 0. 0–unlimited
FIMI/SetIssuerFeesForOwnTerm /Rq/TierAmount	Num	In	Tier amount The field value is more or equal to 0.
FIMI/SetIssuerFeesForOwnTerm /Rq/TieredFeeAmount	Num	In	Tiered fee amount
FIMI/SetIssuerFeesForOwnTerm /Rq/BoundAmountTier	Int(1)	In	Bound amount tier: 1 – left (bottom) 2 – right (upper)
FIMI/SetIssuerFeesForOwnTerm /Rq/CounterId	Int	In	Counter ID. See the list of available values in FIMI/InitSession/Rp/Counters.
FIMI/SetIssuerFeesForOwnTerm /Rq/UseCounter	Int(1)	In	Counter use type: 0 – not used (counter ID is reset) 1 – do not charge fee if the counter has not reached its maximum value 2 – transaction amount plus delta change 3 – transaction minus plus delta change If the field value is 1,2,3, the counter ID must be transferred in the CounterId field.
FIMI/SetIssuerFeesForOwnTerm /Rq/NewSeqNo	Int(1)	In	New sequence number. It is used to change the fee sequence by interchanging sequence numbers with the fee which already exists in the group.
FIMI/SetIssuerFeesForOwnTerm /Rq/RemoveMissingUserFields	Int(1)	In	Remove user fields which are absent in the UserFields structure. If NULL, it=0.
FIMI/SetIssuerFeesForOwnTerm/ Rq/UserFields	ArrRec	In	{ Str(100)* Name; Str(3800) TextValue; StrXML XMLValue. } [] – list of the user fee fields. If the TextValue and XMLValue fields are absent in the structure and card has the user field, this user field will not be changed. To reset the TextValue and XMLValue values, specify empty values of the TextValue and XMLValue fields.

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>AddIssuerFeesForOwnTerm – adding issuer fees for own terminals</b>			
FIMI/AddIssuerFeesForOwnTerm /Rq/InstName	Str(4)*	In	Institution name
FIMI/AddIssuerFeesForOwnTerm /Rq/IssuerObjectType	Int(1)*	In	Type of issuer object: 1 – card profile 2 – card profile template 3 – cardless transactions
FIMI/AddIssuerFeesForOwnTerm /Rq/IssuerObjectId	Int*	In	Identifier of issuer object. If <i>IssuerObjectType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation). If <i>IssuerObjectType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation). If the <i>CardProfileType</i> value is 3, the field transfers the value -1.
FIMI/AddIssuerFeesForOwnTerm/ Rq/AcqObjectType	Int(1)*	In	Type of acquiring object: 1–acquiring group 2–acquiring group template
FIMI/AddIssuerFeesForOwnTerm /Rq/AcqObjectId	Int*	In	Identifier of acquiring object. If <i>AcqObjectType</i> =1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation). If <i>AcqObjectType</i> =2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation). If <i>AcqObjectId</i> = 0, it contains «any».
FIMI/AddIssuerFeesForOwnTerm /Rq/AcctType	Int(2)*	In	Account type: 0–«any» 1–checking 11–savings 31–credit 91–bonus
FIMI/AddIssuerFeesForOwnTerm /Rq/AcctCurrency	Int(4)*	In	Account currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0–«any»

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddIssuerFeesForOwnTerm /Rq/CurrencyExchange	Int(1)*	In	Currency conversion: 0– indifferently 1–yes 2–no
FIMI/AddIssuerFeesForOwnTerm /Rq/TranCode	Int(4)*	In	Transaction code. See a list of available values in section 3.8.
FIMI/AddIssuerFeesForOwnTerm /Rq/Flat	Num	In	Fixed payment If NULL, it=0
FIMI/AddIssuerFeesForOwnTerm /Rq/Percentage	Num(3,2)	In	Percent of amount If NULL, it=0
FIMI/AddIssuerFeesForOwnTerm /Rq/Minimum	Num	In	Minimum available fee amount The field value more or equal to 0. If NULL, it=0
FIMI/AddIssuerFeesForOwnTerm /Rq/Maximum	Num	In	Maximum available fee amount (0–unlimited) The field value is more or equal to 0. 0–unlimited If NULL, it=0
FIMI/AddIssuerFeesForOwnTerm /Rq/TierAmount	Num	In	Tier amount The field value is more or equal to 0. If NULL, it=0
FIMI/AddIssuerFeesForOwnTerm /Rq/TieredFeeAmount	Num	In	Tiered fee amount If NULL, it=0
FIMI/AddIssuerFeesForOwnTerm /Rq/BoundAmountTier	Int(1)	In	Bound amount tier: 1– left (bottom) 2– right (upper) If NULL, it=1
FIMI/AddIssuerFeesForOwnTerm /Rq/FeeCurrency	Int(4)*	In	Limit currency The list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation.
FIMI/AddIssuerFeesForOwnTerm /Rq/RateGroup	Int	In	Group of currency rates For the list of available values, see <i>FIMI/GetRateGroups/Rp&gt;List</i>
FIMI/AddIssuerFeesForOwnTerm /Rq/BackOfficeId	Str(20)	In	Fee ID in Back-Office
FIMI/AddIssuerFeesForOwnTerm /Rq/OptionMask	Int	In	Additional fee options. Bit mask, bit 1 – hidden fee If NULL, it=0
FIMI/AddIssuerFeesForOwnTerm /Rq/CounterId	Int	In	Counter ID. See the list of available values in <i>FIMI/InitSession/Rp/Counters</i> .

<b>Parameter Name</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/AddIssuerFeesForOwnTerm /Rq/UseCounter	Int(1)	In	<p>Counter use type:</p> <ul style="list-style-type: none"> <li>0 – not used (counter ID is reset)</li> <li>1 – do not charge fee if the counter has not reached its maximum value</li> <li>2 – transaction amount plus change delta</li> <li>3 – transaction amount minus change delta</li> </ul> <p>If the field value is 1,2,3, the counter ID must be transferred in the CounterId field. If NULL,it=0</p>
<b><i>DeleteIssuerFeesForOwnTerm – deleting issuer fees for own terminals</i></b>			
FIMI/DeleteIssuerFeesForOwnTerm/Rq/InstName	Str(4)*	In	Institution name
FIMI/DeleteIssuerFeesForOwnTerm/Rq/IssuerObjectType	Int(1)*	In	<p>Type of issuer object:</p> <ul style="list-style-type: none"> <li>1 – card profile</li> <li>2 – card profile template</li> <li>3 – cardless transactions</li> </ul>
FIMI/DeleteIssuerFeesForOwnTerm/Rq/IssuerObjectId	Int*	In	<p>Identifier of issuer object.</p> <p>If <i>IssuerObjectType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation).</p> <p>If <i>IssuerObjectType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation).</p> <p>If the <i>CardProfileType</i> value is 3, the field transfers the value -1.</p>
FIMI/DeleteIssuerFeesForOwnTerm/Rq/AcqObjectType	Int(1)*	In	<p>Type of acquiring object:</p> <ul style="list-style-type: none"> <li>1–acquiring group</li> <li>2–acquiring group template</li> </ul>
FIMI/DeleteIssuerFeesForOwnTerm/Rq/AcqObject	Int*	In	<p>Identifier of acquiring object.</p> <p>If <i>AcqObjectType</i>=1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation).</p> <p>If <i>AcqObjectType</i>=2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation).</p> <p>If <i>AcqObject</i> = 0, it contains «any».</p>

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/DeleteIssuerFeesForOwnTerm/Rq/AcctType	Int(2)*	In	Account type: 0-«any» 1–checking 11–savings 31–credit 91–bonus
FIMI/DeleteIssuerFeesForOwnTerm/Rq/AcctCurrency	Int(4)*	In	Account currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0-«any»
FIMI/DeleteIssuerFeesForOwnTerm/Rq/CurrencyExchange	Int(1)*	In	Currency conversion: 0– indifferently 1–yes 2–no
FIMI/DeleteIssuerFeesForOwnTerm/Rq/TranCode	Int(4)*	In	Transaction code. See a list of available values in section 3.8.
FIMI/DeleteIssuerFeesForOwnTerm/Rq/SeqNo	Int	In	Sequence If NULL, it is any
<b>GetIssuerFeesForExternalTerm – request for issuer fees for external terminals</b>			
FIMI/GetIssuerFeesForExternalTerm/Rq/InstName	Str(4)	In	Institution name If not defined, the fees of all allowed financial institutions (including the FIMI terminal institution) will be selected.
FIMI/GetIssuerFeesForExternalTerm /Rq/IssuerObjectType	Int(1)	In	Type of issuer object: 0 – «any» 1 – card profile 2 – card profile template If NULL, 0.
FIMI/GetIssuerFeesForExternalTerm /Rq/IssuerObjectId	Int	In	Identifier of issuer object. The field value is used if the <i>CardProfileType</i> field is set to 1 or 2. If <i>CardProfileType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation). If <i>CardProfileType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation). If the field is not defined, all allowed fees are selected.
FIMI/GetIssuerFeesForExternalTerm /Rp/Fees	ArrRec	Out	{ Str(4)* InstName; Int(1)* IssuerObjectType; Int* IssuerObjectId;

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			Int(1)* AcqObjectType; Int* AcqObjectId; Int* PaymentSystem; Int* Country; Str(11)* AcqBIN; Int(4)* TranCurrency; Int(2)* AcctType; Int(4)* AcctCurrency; Int(1)* CurrencyExchange; Int(4)* TranCode; Int* SeqNo; Num* Flat; Num(3,2)* Percentage; Num* Minimum; Num* Maximum; Num* TierAmount; Num* TieredFeeAmount; Int(1)* BoundAmountTier; Int(4)* FeeCurrency; Int RateGroup; Str(20) BackOfficeId; Int OptionMask; Int CounterId; Int(1)* UseCounter. } [] – a list of issuer fees.
FIMI/GetIssuerFeesForExternalTerm/Rq/UserFields	ArrRec	In	{ Str(4)* InstName; Int(1)* IssuerObjectType; Int* IssuerObjectId; Int(1)* AcqObjectType; Int* AcqObjectId; Int* PaymentSystem; Int* Country; Str(11)* AcqBIN; Int(4)* TranCurrency; Int(2)* AcctType; Int(4)* AcctCurrency; Int(1)* CurrencyExchange; Int(4)* TranCode; Int* SeqNo; Str(100)* Name; Str(4000) TextValue; Str XMLValue. } [] – a list of user fee fields.
InstName – institution name IssuerObjectType – type of issuer object (for details, see the description of the <i>IssuerObjectType</i> field) IssuerObjectId – identifier of issuer object (for details, see the description of the <i>IssuerObjectId</i> field) AcqObjectType – type of acquiring object (1–acquiring group, 2– acquiring group template)			

<b>Parameter Name</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
AcqObjectId – identifier of acquiring object; if <i>AcqObjectType</i> =1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation); if <i>AcqObjectType</i> =2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation); if <i>AcqObjectId</i> = 0, it contains «any»			
PaymentSystem – payment system (a list of available values is transferred in the <i>PaymentSystems</i> field of the <i>InitSession</i> operation)			
Country – country (0–«any»; a list of available values is transferred in the <i>CountryList</i> field of the <i>InitSession</i> operation)			
AcqBIN – acquirer BIN ('.-any)			
TranCurrency – transaction currency (0–«any»; a list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation)			
AcctType – account type (0–«any».; 1–checking; 11–savings; 31–credit; 91–bonus)			
AcctCurrency – account currency (0–«any».; a list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation)			
CurrencyExchange – currency conversion (1–yes; 2–no; 0– Indifferently)			
TranCode – transaction code (see a list of available values in section 3.8)			
SeqNo – sequence			
Flat – fixed fee amount			
Percentage – percent of amount			
Minimum – minimum available fee amount			
Maximum – maximum available fee amount (0–unlimited)			
TierAmount – tier amount			
TieredFeeAmount – tiered fee amount			
BoundAmountTier – bound amount tier (1– left (bottom); 2– right (upper))			
FeeCurrency – limit currency (the list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation)			
RateGroup – group of currency rates (for the list of available values, see <i>FIMI/GetRateGroups/Rp&gt;List</i> )			
BackOfficeId – fee ID in Back-Office			
OptionMask – additional fee options (bit mask, 1 bit – hidden fee)			
CounterId – counter ID (see the list of available values in <i>FIMI/InitSession/Rp/Counters</i> )			
UseCounter – counter use type: 0 – not used 1 – do not charge fee if the counter has not reached its maximum value 2 – transaction amount plus change delta 3 – transaction amount minus change delta			

### **SetIssuerFeesForExternalTerm – changing issuer fees for external terminals**

FIMI/SetIssuerFeesForExternalTerm/Rq/InstName	Str(4)*	In	Institution name
FIMI/SetIssuerFeesForExternalTerm/Rq/IssuerObjectType	Int(1)*	In	Type of issuer object: 1 – card profile; 2 – card profile template.
FIMI/SetIssuerFeesForExternalTerm/Rq/IssuerObjectId	Int*	In	Identifier of issuer object. If <i>CardProfileType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation). If <i>CardProfileType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation).

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetIssuerFeesForExternalTerm/Rq/AcqObjectType	Int(1)*	In	Type of acquiring object: 1–acquiring group; 2– acquiring group template.
FIMI/SetIssuerFeesForExternalTerm/Rq/AcqObjectId	Int*	In	Identifier of acquiring object.  If <i>AcqObjectType</i> =1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation).  If <i>AcqObjectType</i> =2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation).  If <i>AcqObjectId</i> = 0, it contains «any».
FIMI/SetIssuerFeesForExternalTerm/Rq/PaymentSystem	Int*	In	Payment system A list of available values is transferred in the <i>PaymentSystems</i> field of the <i>InitSession</i> operation.
FIMI/SetIssuerFeesForExternalTerm/Rq/Country	Int*	In	Country A list of available values is transferred in the <i>CountryList</i> field of the <i>InitSession</i> operation. 0–«any»
FIMI/SetIssuerFeesForExternalTerm/Rq/AcqBIN	Str(11)*	In	Acquirer BIN ‘_’– any
FIMI/SetIssuerFeesForExternalTerm/Rq/TranCurrency	Int(4)*	In	Transaction currency: A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0–«any».
FIMI/SetIssuerFeesForExternalTerm/Rq/AcctType	Int(2)*	In	Account type: 0–«any» 1–checking 11–savings 31–credit 91–bonus
FIMI/SetIssuerFeesForExternalTerm/Rq/AcctCurrency	Int(4)*	In	Account currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0–«any»
FIMI/SetIssuerFeesForExternalTerm/Rq/CurrencyExchange	Int(1)*	In	Currency conversion: 0– Indifferently 1–yes 2–no

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetIssuerFeesForExternalTerm/Rq/TranCode	Int(4)*	In	Transaction code. See a list of available values in section 3.8.
FIMI/SetIssuerFeesForExternalTerm/Rq/SeqNo	Int*	In	Sequence
FIMI/SetIssuerFeesForExternalTerm/Rq/Flat	Num	In	Fixed fee amount
FIMI/SetIssuerFeesForExternalTerm/Rq/Persentage	Num(3,2)	In	Percent of amount
FIMI/SetIssuerFeesForExternalTerm/Rq/Minimum	Num	In	Minimum available fee amount. The field value is more or equal to 0.
FIMI/SetIssuerFeesForExternalTerm/Rq/Maximum	Num	In	Maximum available fee amount (0–unlimited) The field value is more or equal to 0. 0–unlimited.
FIMI/SetIssuerFeesForExternalTerm/Rq/TierAmount	Num	In	Tier amount The field value more or equal to 0.
FIMI/SetIssuerFeesForExternalTerm/Rq/TieredFeeAmount	Num	In	Tiered fee amount
FIMI/SetIssuerFeesForExternalTerm/Rq/BoundAmountTier	Int(1)	In	Bound amount tier: 1– left (bottom) 2– right (upper)
FIMI/SetIssuerFeesForExternalTerm/Rq/CounterId	Int	In	Counter ID. See the list of available values in FIMI/InitSession/Rp/Counters.
FIMI/SetIssuerFeesForExternalTerm/Rq/UseCounter	Int(1)	In	Counter use type: 0 – not used (counter ID is reset) 1 – do not charge fee if the counter has not reached its maximum value 2 – transaction amount plus change delta 3 – transaction amount minus change delta If the field value is 1,2,3, the counter ID must be transferred in the CounterId field.
FIMI/SetIssuerFeesForExternalTerm /Rq/NewSeqNo	Int(1)	In	New sequence number. It is used to change the fee sequence by interchanging sequence numbers with the fee which already exists in the group.
FIMI/SetIssuerFeesForExternalTerm/Rq/RemoveMissingUserFields	Int(1)	In	Remove user fields which are absent in the UserFields structure. If NULL, it=0.

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetIssuerFeesForExternalTerm/Rq/UserFields	ArrRec	In	<pre>{     Str(100)* Name;     Str(3800) TextValue;     StrXML XMLValue. }</pre> <p>[] – list of user fee fields.</p> <p>If the <i>TextValue</i> and <i>XML Value</i> fields are absent in the structure and card has the user field, this user field will not be changed. To reset the <i>TextValue</i> and <i>XML Value</i> values, specify empty values of the <i>TextValue</i> and <i>XML Value</i> fields.</p>
<b>AddIssuerFeesForExternalTerm – changing issuer fees for external terminals</b>			
FIMI/AddIssuerFeesForExternalTerm/Rq/InstName	Str(4)*	In	Institution name
FIMI/AddIssuerFeesForExternalTerm/Rq/IssuerObjectType	Int(1)*	In	Type of issuer object: 1 – card profile 2 – card profile template
FIMI/AddIssuerFeesForExternalTerm/Rq/IssuerObjectId	Int*	In	Identifier of issuer object. If <i>IssuerObjectType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation). If <i>IssuerObjectType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation).
FIMI/AddIssuerFeesForExternalTerm/Rq/AcqObjectType	Int(1)*	In	Type of acquiring object: 1–acquiring group 2–acquiring group template
FIMI/AddIssuerFeesForExternalTerm/Rq/AcqObjectId	Int*	In	Identifier of acquiring object. If <i>AcqObjectType</i> =1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation). If <i>AcqObjectType</i> =2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation). If <i>AcqObjectId</i> = 0, it contains «any».

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddIssuerFeesForExternalTerm/Rq/PaymentSystem	Int*	In	Payment system. The list of available values is transferred in the <i>PaymentSystems</i> field of the <i>InitSession</i> operation.
FIMI/AddIssuerFeesForExternalTerm/Rq/Country	Int*	In	Country. The list of available values is transferred in the <i>CountryList</i> field of the <i>InitSession</i> operation. 0-«any»
FIMI/AddIssuerFeesForExternalTerm/Rq/AcqBIN	Str(11)*	In	Acquirer BIN. ‘-’ – any
FIMI/AddIssuerFeesForExternalTerm/Rq/TranCurrency	Int(4)*	In	Transaction currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0-«any»
FIMI/AddIssuerFeesForExternalTerm/Rq/AcctType	Int(2)*	In	Account type: 0-«any» 1–checking 11–savings 31–credit 91–bonus
FIMI/AddIssuerFeesForExternalTerm/Rq/AcctCurrency	Int(4)*	In	Account currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0-«any»
FIMI/AddIssuerFeesForExternalTerm/Rq/CurrencyExchange	Int(1)*	In	Currency conversion: 0–indifferently 1–yes 2–no
FIMI/AddIssuerFeesForExternalTerm/Rq/TranCode	Int(4)*	In	Transaction code. See a list of available values in section 3.8.
FIMI/AddIssuerFeesForExternalTerm/Rq/Flat	Num	In	Fixed payment. If NULL, it=0
FIMI/AddIssuerFeesForExternalTerm/Rq/Percentage	Num(3,2)	In	Percent of amount. If NULL, it=0
FIMI/AddIssuerFeesForExternalTerm/Rq/Minimum	Num	In	Minimum available fee amount. The field is value more or equal to 0. If NULL, it=0
FIMI/AddIssuerFeesForExternalTerm/Rq/Maximum	Num	In	Maximum available fee amount. The field value is more or equal to 0. 0–unlimited. If NULL, it=0

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddIssuerFeesForExternalTerm/Rq/TierAmount	Num	In	Tier amount. The field value is more or equal to 0. If NULL, it=0
FIMI/AddIssuerFeesForExternalTerm/Rq/TieredFeeAmount	Num	In	Tiered fee amount. If NULL, it=0
FIMI/AddIssuerFeesForExternalTerm/Rq/BoundAmountTier	Int(1)	In	Bound amount tier: 1 – left (bottom) 2 – right (upper) If NULL, it=1
FIMI/AddIssuerFeesForExternalTerm/Rq/FeeCurrency	Int(4)*	In	Limit currency. The list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation.
FIMI/AddIssuerFeesForExternalTerm/Rq/RateGroup	Int	In	Group of currency rates. For the list of available values, see <i>FIMI/GetRateGroups/Rp&gt;List</i> .
FIMI/AddIssuerFeesForExternalTerm/Rq/BackOfficeId	Str(20)	In	Fee ID in Back-Office
FIMI/AddIssuerFeesForExternalTerm/Rq/OptionMask	Int	In	Additional fee options. Bit mask, bit 1 – hidden fee If NULL, it=0
FIMI/AddIssuerFeesForExternalTerm/Rq/CounterId	Int	In	Counter ID. See the list of available values in <i>FIMI/InitSession/Rp/Counters</i> .
FIMI/AddIssuerFeesForExternalTerm/Rq/UseCounter	Int(1)	In	Counter use type: 0 – not used (counter ID is reset); 1 – do not charge fee if the counter has not reached its maximum value; 2 – transaction amount plus change delta; 3 – transaction amount minus change delta. If the field value is 1,2,3, the counter ID must be transferred in the CounterId field. If NULL, it=0
<b>DeleteIssuerFeesForExternalTerm – deleting issuer fees for own terminals</b>			
FIMI/DeleteIssuerFeesForExternalTerm/Rq/InstName	Str(4)*	In	Institution name
FIMI/DeleteIssuerFeesForExternalTerm/Rq/IssuerObjectType	Int(1)*	In	Type of issuer object: 1 – card profile 2 – card profile template

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/DeleteIssuerFeesForExternalTerm/Rq/IssuerObjectId	Int*	In	Identifier of issuer object. If <i>CardProfileType</i> = 1, the field transfers the identifier of the card profile (a list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> operation). If <i>CardProfileType</i> = 2, the field transfers the identifier of the card profile template (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation).
FIMI/DeleteIssuerFeesForExternalTerm/Rq/AcqObjectType	Int(1)*	In	Type of acquiring object: 1—acquiring group 2—acquiring group template
FIMI/DeleteIssuerFeesForExternalTerm/Rq/AcqObjectById	Int*	In	Identifier of acquiring object. If <i>AcqObjectType</i> =1, it contains the identifier of the acquiring groups (a list of available values is transferred in the <i>AcquiringGroups</i> field of the <i>InitSession</i> operation). If <i>AcqObjectType</i> =2, it contains the identifier of the acquiring groups templates (a list of available values is transferred in the <i>GroupTemplates</i> field of the <i>InitSession</i> operation). If <i>AcqObjectId</i> = 0, it contains «any».
FIMI/DeleteIssuerFeesForExternalTerm/Rq/PaymentSystem	Int*	In	Payment system. A list of available values is transferred in the <i>PaymentSystems</i> field of the <i>InitSession</i> operation.
FIMI/DeleteIssuerFeesForExternalTerm/Rq/Country	Int*	In	Country. A list of available values is transferred in the <i>CountryList</i> field of the <i>InitSession</i> operation. 0—«any»
FIMI/DeleteIssuerFeesForExternalTerm/Rq/AcqBIN	Str(11)*	In	Acquirer BIN. ‘— any
FIMI/DeleteIssuerFeesForExternalTerm/Rq/TranCurrency	Int(4)*	In	Transaction currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0—«any»
FIMI/DeleteIssuerFeesForExternalTerm/Rq/AcctType	Int(2)*	In	Account type: 0—«any» 1—checking 11—savings 31—credit 91—bonus

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/DeleteIssuerFeesForExternalTerm/Rq/AcctCurrency	Int(4)*	In	Account currency. A list of available values is transferred in the <i>CurrencyList</i> field of the <i>InitSession</i> operation. 0-«any»
FIMI/DeleteIssuerFeesForExternalTerm/Rq/CurrencyExchange	Int(1)*	In	Currency conversion: 0–indifferently 1–yes 2–no
FIMI/DeleteIssuerFeesForExternalTerm/Rq/TranCode	Int(4)*	In	Transaction code. See a list of available values in section 3.8.
FIMI/DeleteIssuerFeesForExternalTerm/Rq/SeqNo	Int	In	Sequence. If NULL, it is any

### 2.3.3.13 On Device:

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetDevicesList – request for information on devices</b>			
FIMI/GetDevicesList/Rq/DeviceId	Str(32)	In	Device ID
FIMI/GetDevicesList/Rq/Customer	Str(19)	In	Telebank customer ID
FIMI/GetDevicesList/Rq/MBR	Int(3)	In	Number of the Telebank customer owner (for the Telebank customer, its value is 0)
FIMI/GetDevicesList/Rq/CardUID	Str(64)	In	Unique Telebank customer ID. It can stand for the Telebank customer ID.
FIMI/GetDevicesList/Rp/DeviceList	ArrRec	Out	<pre>{     str(4)* InstName;     str(32)* DeviceId;     str(250)* Name;     int(2)* Type;     str(50)* OS;     str(100)* SerialNumber;     str(250) OSVersion;     str(250) ApplicationVersion;     str(100) IMEI;     str(100) IMSI;     str(100) PhoneNumber;     str(50) BluetoothMAC;     str(50) WIFIMAC;     time LastActiveTime;     str(100) LastIPAddress;     int* Status;     time LastBlockTime; }</pre>

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
InstName – institution name DeviceId – device ID Name – device name Type – device type 0 – unknown 1 – mobile phone 2 – tab 3 – watch 4 – mobile phone or tab 5 – PC 6 – card 7 – mobile tag or key fob 8 – wristband 9 – mobile phone case 10 – sticker 11 – vehicle 12 – media/gaming device 13 – domestic appliance 14 – jewelry 15 – fashion accessory 16 – garment 17 – wearable device  OS – operating system of the device. SerialNumber – serial number of the device. OSVersion – OS version of the device. ApplicationVersion – application version. IMEI – device IMEI IMSI – device IMSI. PhoneNumber – phone number. BluetoothMAC – device Bluetooth MAC. WIFIMAC – device WIFI MAC. LastActiveTime – last time when the device was active. LastIPAddress – last IP address of the device. Status – device status: 1 – active, 2 – blocked, 3 – deleted. LastBlockTime – last time when the device was blocked. LastReasonBlock – reason for the last block of the device.			str(1000) LastReasonBlock; } [...] – array of the device data structures where:

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetDevicesList/Rp/UserFields	ArrRec	Out	<pre>{     str(100)* Name;     str(4000) TextValue;     strXML XMLValue; }</pre> <p>[] – array of structures, List of user-defined fields of the device is filled if only one device is found. Name, TextValue, XMLValue – name, text and XML values of user-defined fields.</p>
<b><i>GetDeviceHistory – request for device actions log</i></b>			
FIMI/GetDeviceHistory/Rq/DeviceId	Str(32)	In	Device ID
FIMI/GetDeviceHistory/Rq/Customer	Str(19)	In	Telebank customer ID
FIMI/GetDeviceHistory/Rq/MBR	Int(3)	In	Number of the Telebank customer owner (for the Telebank customer, its value is 0)
FIMI/GetDeviceHistory/Rq/CardUID	Str(64)	In	Unique Telebank customer ID. It can stand for the Telebank customer ID.
FIMI/GetDeviceHistory/Rq/LastId	Int	In	<p>ID of the last operation accepted by the terminal.</p> <p>If it is specified in the request, the response returns all operations which meet other conditions and have been executed later (their Id&gt;LastId)</p>
FIMI/GetDeviceHistory/Rq/FromTime	Time	In	<p>Start time</p> <p>If not defined, the operations over the last day are returned.</p>
FIMI/GetDeviceHistory/Rq/ToTime	Time	In	End time
FIMI/GetDeviceHistory/Rq/Count	Int(4)	In	<p>Number of requested operations, but not more than 1000.</p> <p>If NULL, it is 1000.</p>
FIMI/GetDeviceHistory/Rp/Log	ArrRec	Out	<pre>{     int(9)* Id;     str(19)* Customer;     int(3)* MBR;     str(32)* DeviceId;     str(1000)* DeviceTitle;     time* OperTime;     str(250)* OperName;     int(12) TranId;     str(100) IPAddress; }</pre> <p>[...] – array of the data structures: Id – operation ID</p>

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			CustomerID, MBR – Telebank customer ID DeviceID – device ID DeviceTitle – device name OperTime – operation time OperName – operation name TranId – ID of the transaction generated within the operation IPAddress – IP address of the operation
<b>ChangeDeviceStatus – device status change</b>			
TB/ChangeDeviceStatus/Rq/ DeviceID	Str(32)*	In	Device ID
TB/ChangeDeviceStatus/Rq/ Status	Int*	In	New device status. Available values: 1 – active 2 – blocked 3 – deleted
FIMI/ChangeDeviceStatus/Rq/ ChangeReason	Str(1000)	In	Status change reason
<b>SetDeviceAdditionalFields – setting user-defined fields of device</b>			
TB/SetDeviceAdditionalFields/Rq/ DeviceID	Str(32)*	In	Device ID
FIMI/SetDeviceAdditionalFields/ Rq/UserFields	ArrRec	In	<pre>{   str(100)* Name;   str(3800) TextValue;   strXML XMLValue; } [] – array of structures, list of additional terminal fields:</pre> <p>Name – name TextValue – text value XMLValue – XML value</p> <p>Total length of the Name and TextValue fields of all array records.</p> <p><i>FIMI/SetTerminalAdditionalFields/Rq/UserFields</i> must not exceed 3900 characters.</p>
FIMI/SetDeviceAdditionalFields/ Rq/RemoveMissingFields	Int	In	Indicates that missing fields must be removed. If NULL, it is 1.

### 2.3.3.14 On Alias:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetAliasInfo – request for information on aliases in TWO</b>			
FIMI/GetAliasInfo/Rq/Serviceld	Int(1)	In	<p>Service ID</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>1 – VISA ADS</li> <li>3 – NSPK FPS C2C</li> <li>4 – NSPK FPS C2B</li> <li>NULL – any</li> </ul>
FIMI/GetAliasInfo/Rq/Mode	Int(1)	In	<p>Operation execution mode:</p> <ul style="list-style-type: none"> <li>1 – request for the information from the external storage.</li> <li>2 – request for the information from the TWO storage of aliases.</li> <li>3 – request for the information from the external storage supplemented with data from the TWO storage.</li> </ul> <p>If NULL, it is 2. Modes 1 and 3 are allowed for Serviceld = 1(VISA ADS).</p>
FIMI/GetAliasInfo/Rq/Alias	Str(500)	In	Alias
FIMI/GetAliasInfo/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/GetAliasInfo/Rq/MBR	Int(3)	In	
FIMI/GetAliasInfo/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetAliasInfo/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution</p> <p>If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.</p>
FIMI/GetAliasInfo/Rq/Account	Str(30)	In	<p>Account number</p> <p>It is used if the card number for Serviceld 3, 4 is not defined.</p>
FIMI/GetAliasInfo/Rq/AccountUID	Str(32)	In	<p>Unique account ID in the session. It can stand for the account number.</p> <p>It is used if the card number for Serviceld 3, 4 is not defined.</p>
FIMI/GetAliasInfo/Rp/Status	Int(1)	In	<p>Alias status:</p> <ul style="list-style-type: none"> <li>-1 - all</li> <li>0 – only deleted</li> <li>1 – only active</li> </ul> <p>If NULL, it is 1. It is used for Mode = 1.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetAliasInfo/Rp/Aliases	ArrRec	Out	<pre>{     str(500)* Alias;     int(1)* ServiceId;     int(1)* AliasType;     time CreateTime;     time LastUpdateTime;     int(1)* Status;     time LastUpdateStatusTime;     int* Permission;     str(30) Account2;     str(32) Account2UID; }</pre> <p>} [] – list of the card or account aliases where</p> <p>Alias – alias.          ServiceId – service ID (for the list of available values, see <i>FIMI/GetAliasInfo/Rq/ServiceId</i>).          AliasType – alias type (1 – phone number, 2 – E-mail).          CreateTime – alias creation time.          LastUpdateTime – time of the last alias update.          Status – alias status (1 – active, 0 – deleted).          LastUpdateStatusTime – time of the last status update.          Permission – additional capabilities of the alias within the service for the member bank, bit mask:              Bit 1 – transfers using the alias by the request (C2C Me2Me Pull) are allowed.              Bit 2 – C2B transfers using the NSPK SBPay mobile application are allowed.          Bit values:              1 – allowed.              0 – prohibited.          Account2 – default account for debiting.          Account2UID – unique identifier of the default account for debiting.</p>
FIMI/GetAliasInfo/Rp/ AllowAliasMemberList	ArrRec	Out	<pre>{     str(500)* Alias;     str(20)* BIC;     int* Permission; }</pre> <p>} [] – list of member with additional permissions for the alias (transferred for aliases with the NSPK FPS C2C service) where</p> <p>Alias – alias.          BIC – member bank BIC (for the full list of the member banks, use the <i>GetServiceMembersList</i> request).          Permission – additional capabilities of the alias within the service for the member bank, bit mask:              1 bit – transfers using the alias by the request (C2C Me2Me Pull).          Bit values:              1 – allowed.              0 – prohibited.          It is returned if one alias is found for the NSPK FPS C2C service (ServiceId=3).</p>

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetAliasInfo/Rp/ AliasSubscriptions	ArrRec	Out	<pre>{     str(500)* Alias;     str(35)* RetailerName;     str(140)* RetailerTitle;     str(140)* SubscriptionPurpose;     str(32)* SubscriptionToken;     str(30)* Account;     str(32) AccountUID;     str(12)* TaxPayerNumber;     str(15) OGRN;     str(12)* MemberId;     str(32)* SubscriptionQRId;     int(12)* CreateTranId;     time* CreateDate;     time* UpdateDate;     int* Status;  } [] – list of the alias subscriptions (transferred for aliases with the NSPK FPS C2B service) where</pre>

Alias – alias.

RetailerTitle – merchant title.

RetailerName – merchant name.

SubscriptionPurpose – subscription purpose.

SubscriptionToken – unique identifier (token) of the subscription.

Account – account number.

TaxPayerNumber – tax payer number of a corporate customer or individual entrepreneur.

OGRN – primary state registration number of a corporate customer or individual entrepreneur.

MemberId – identifier of a participant subscribed to the service (the list of available values is requested within the *GetServiceMembersList* operation).

SubscriptionQRId – identifier of the QR code used for the subscription creation.

CreateTranId – identifier of the subscription creation transaction.

CreateDate – creation date.

UpdateDate – change date.

Status – subscription status (1 – active; 0 – deleted; 2 – suspended).

It is returned if one alias was found for the NSPK FPS C2B service (ServiceId = 4).

#### **CreateAlias – creating alias**

FIMI/CreateAlias/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/CreateAlias/Rq/MBR	Int(3)	In	
FIMI/CreateAlias/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/CreateAlias/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateAlias/Rq/Account	Str(30)	In	Account number. It is used if the card number for Serviceld 3, 4 is not defined.
FIMI/CreateAlias/Rq/AccountUID	Str(32)	In	Unique account ID in the session. It can stand for the account number. It is used if the card number for Serviceld 3, 4 is not defined.
FIMI/CreateAlias/Rq/Alias	Str(500)*	In	Alias
FIMI/CreateAlias/Rq/AliasType	Int(1)*	In	Alias type: 1 – phone number 2 – E-mail
FIMI/CreateAlias/Rq/Serviceld	Int(1)*	In	Service ID. Available values: 1 - VISA ADS 3 – NSPK FPS C2C 4 – NSPK FPS C2B
FIMI/CreateAlias/Rq/ContactEmail	Str(100)	In	Contact E-mail. It is used for VISA ADS.
FIMI/CreateAlias/Rq/ContactPhone	Str(30)	In	Contact phone number. It is used for VISA ADS.
FIMI/CreateAlias/Rq/Mode	Int(1)	In	Operation execution mode: 1 – operation is executed in the external system. 2 – operation is executed in TWO. 3 – operation is executed in both external system and TWO.  If NULL, it is 3. For VISA ADS, the transferred Mode value is ignored and it is always 3. For NSPK FPS, all operation modes are available. When the operation is executed in the external system, the default bank is set (the operation is two-pass with the OTP request (FIMI/Response=52)).
FIMI/CreateAlias/Rq/PrevTranId	Int(12)	In	Id of the transaction generated within the first request for which OTP is required (FIMI/Response 52 – DynamicPasswordRequired was returned) to continue the operation execution.  It is used for NSPK FPS with Mode = 1,3.
FIMI/CreateAlias/Rq/ExtOTP	Str(30)	In	External OTP  It is used for NSPK FPS with Mode = 1,3 and defined PrevTranId field.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateAlias/Rq/Account2	Str(30)	In	Default account for debiting
FIMI/CreateAlias/Rq/Account2UID	Str(32)	In	Unique identifier of the account within the session. It can stand for the number of the default account for debiting.
FIMI/CreateAlias/Rq/Permission	Int	In	Additional capabilities of the alias within the service for the member bank, bit mask: 1 bit – transfers using the alias by the request (C2C Me2Me Pull) are allowed. It is used for the NSPK FPS service.
<b>DeleteAlias – deleting alias</b>			
FIMI/DeleteAlias/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/DeleteAlias/Rq/MBR	Int(3)	In	It used for VISA ADS.
FIMI/DeleteAlias/Rq/CardUID	Str(64)	In	Unique card ID It can stand for the card number. It is used for VISA ADS.
FIMI/DeleteAlias/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the account If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO. It is used for NSPK FPS.
FIMI/DeleteAlias/Rq/Alias	Str(500)*	In	Alias
FIMI/DeleteAlias/Rq/Serviceld	Int(1)*	In	Alias ID Available values: 1 - VISA ADS 3 – NSPK FPS C2C 4 – NSPK FPS C2B
FIMI/DeleteAlias/Rq/OnlyChangeStatus	Int(1)	In	Attribute of changing the alias status: 0 – delete alias from the TWO storage. 1 – change the alias status in the TWO storage to "deleted". If NULL, it is 1. It is used for the NSPK FPS C2C and NSPK FPS C2B services.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>UpdateAlias – changing alias</b>			
FIMI/UpdateAlias/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/UpdateAlias/Rq/PAN	Str(19)	In	Card number and card member number.
FIMI/UpdateAlias/Rq/MBR	Int(3)	In	It is used for the VISA ADS service.
FIMI/UpdateAlias/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.  It is used for the VISA ADS service.
FIMI/UpdateAlias/Rq/Alias	Str(500)*	In	Alias
FIMI/UpdateAlias/Rq/AliasType	Int(1)	In	Alias type: 1 – phone number. 2 – E-mail.  If NULL, the alias attribute is not changed.
FIMI/UpdateAlias/Rq/Serviceld	Int(1)*	In	Service ID  Available values: 1 – VISA ADS  3 – NSPK FPS C2C
FIMI/UpdateAlias/Rq/ContactEmail	Str(100)	In	Contact E-mail  If NULL, the alias attribute is not changed.  It is used for the VISA ADS service.
FIMI/UpdateAlias/Rq/ContactPhone	Str(30)	In	Contact phone number  If NULL, the alias attribute is not changed.  It is used for the VISA ADS service.
FIMI/UpdateAlias/Rq/Permission	Int	In	Additional capabilities of the alias within the service for the member bank, bit mask:  1 bit – transfers using the alias by the request (C2C Me2Me Pull) are allowed.  It is used for the NSPK FPS service.  If NULL, the account to be debited is not changed.
FIMI/UpdateAlias/Rq/Account2	Str(30)	In	Default account for debiting  If NULL, the account to be debited is not changed.
FIMI/UpdateAlias/Rq/Account2UID	Str(32)	In	Unique identifier of the account within the session. It can stand for the number of the default account for debiting.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateAlias/Rq/Account	Str(30)	In	Account number. It is used if the card number is not defined for NSPK FPS C2C. If NULL, the account is not changed.
FIMI/UpdateAlias/Rq/AccountUID	Str(32)	In	Unique identifier of the account within the session. It can stand for the account number. It is used if the card number is not defined for NSPK FPS C2C.
<b>CheckAlias – checking alias</b>			
FIMI/CheckAlias/Rq/Alias	Str(500)*	In	Alias
FIMI/CheckAlias/Rq/Serviceld	Int(1)*	In	Service ID Available values: 1 - VISA ADS 3 – NSPK FPS C2C 4 – NSPK FPS C2B
FIMI/CheckAlias/Rq/Mode	Int(1)	In	Operation execution mode: 1 – request for the information in the external system. 2 – request for the information in TWO.  For NSPK FPS, when executing the operation in the external system, the request for the default member (bank) ID is executed.
FIMI/CheckAlias/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the customer.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the customer institution in TWO.
FIMI/CheckAlias/Rq/PersonId	Int(16)	In	ID of the customer who initiated the alias check. It must be specified when the alias is checked in the external system within the NSPK FPS service.
FIMI/CheckAlias/Rq/PersonData	CustAcct	In	Additional customer data

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
The following subfields can be transferred:			
Name	Type		Description
Identity/Type	Int(1)		Type of the identity document: 1 – passport. 2 – driving licence. 3 – social insurance number. 4 – TPN. 5 – national ID. 6 – international passport. 7 – military identity card. 8 – other document type.
Identity/Number	Str		Document number
Name/First	Str		First name
Name/Last	Str		Last name
Name/Middle	Str		Middle name
TaxPayerNumber	Str		TPN
PhoneNumber	Str		Phone number
It can be specified instead of PersonId when the alias is checked in the external system within the NSPK FPS service if the customer data are not stored in TWO.			
FIMI/CheckAlias/Rp/MemberId	Str(12)	Out	Member ID in the service associated with the alias. It is returned if NSPK FPS is used and Mode=1, it is the default bank and list of possible options is requested in the <i>GetServiceMembersList</i> operation.
FIMI/CheckAlias/Rp/Cards	ArrRec	Out	{ str(19)* PAN; int(3) MBR; int* Status; str(64) CardUID; time* ExpDate; int(1)* Type; str(300) FIO; str IssuerName; str CardBrandName; } [] – cards associated with the alias:
Status – card status (see <i>InitSession/Rp/CardStatList/Id</i> or refer to section 3.2). ExpDate – card expiration date. Type – card type (see the list of available values in the <i>FIMI/GetCardInfo/Rq/Type</i> field). FIO – card owner name. IssuerName – card issuer name. CardBrandName – card program name.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CheckAlias/Rp/Accounts	ArrRec	Out	{ str(30)* Account; str(32) AccountUID; } [] – accounts associated with the alias It is returned if NSPK FPS is used.
<b><i>Alias Verification – checking recipient alias and sender data before transfer</i></b>			
FIMI/AliasVerification/Rq/ServiceId	Int(1)*	In	Service identifier: 3 – NSPK FPS C2C.
FIMI/AliasVerification/Rq/MemberType	Int(1)*	In	Recipient member type: 1 – international.
FIMI/AliasVerification/Rq/MemberId	Str(12)*	In	Recipient member identifier within the service. The list of available variants is requested within the <i>GetServiceMembersList</i> operation.
FIMI/AliasVerification/Rq/Alias	Str(500)*	In	Recipient alias
FIMI/AliasVerification/Rq/CreditorAmount	Num	In	Amount to be credited to the recipient. It is specified if the DebtorAmount field is absent.
FIMI/AliasVerification/Rq/DebtorAmount	Num	In	Amount to be debited from the sender. It is specified if the CreditorAmount field is absent.
FIMI/AliasVerification/Rq/Currency	Int(3)*	In	ISO code of the debit amount or credit amount currency
FIMI/AliasVerification/Rq/PaymentDetailsCategoryCode	Str(10)	In	Transfer category code
FIMI/AliasVerification/Rq/PaymentPurpose	Str(140)	In	Transfer purpose
FIMI/AliasVerification/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/AliasVerification/Rq/FromAccount	Str(30)*	In	Sender account number
FIMI/AliasVerification/Rp/CreditorAmount	Num*	Out	Amount to be credited to the recipient
FIMI/AliasVerification/Rp/CreditorAmountCurrency	Int(3)*	Out	ISO code of the credit amount currency
FIMI/AliasVerification/Rp/DebtorAmount	Num*	Out	Amount to be debited from the sender

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/AliasVerification/Rp/ DebtorAmountCurrency	Int(3)*	Out	ISO code of the debit amount currency
FIMI/AliasVerification/Rp/Amount	Num*	Out	Operation amount
FIMI/AliasVerification/Rp/Currency	Int(3)*	Out	ISO code of the currency for the operation amount
FIMI/AliasVerification/Rp/ ExchangeRate	Num*	Out	Currency conversion rate
FIMI/AliasVerification/Rp/ Fee	Num*	Out	Operation fee in the debit currency
FIMI/AliasVerification/Rp/ RecipientData	CustAcct*	Out	Recipient PAM

The following subfields can be transferred:

<b>Name</b>	<b>Type</b>	<b>Description</b>
Identity/Type	Int(1)	Identity document type: 1 – passport. 2 – driving license. 3 – social insurance number. 4 – TPN. 5 – national ID. 6 – international passport. 7 – military identity card. 8 – other document type.
Identity/Number	Str	Document number
Name/First	Str	First name
Name/Last	Str	Last name
Name/Middle	Str	Middle name
TaxPayerNumber	Str	TPN
PhoneNumber	Str	Phone number
Location/Address	Str	Address
Account/Number	Str	Bank account number
Account/Category	Str	Account funds category
PAM	Str*	Personal Assurance Message(PAM)
CountryOfResidence	Int	Country ISO code

#### **SetAliasPermission – setting additional permissions for alias**

FIMI/SetAliasPermission/Rq/ InstName	Str(4)	In	Character ID (name) of the financial institution  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/SetAliasPermission/Rq/ ServiceId	Int(1)*	In	Service identifier.  Available values: 3 – NSPK FPS C2C.
FIMI/SetAliasPermission/Rq/Alias	Str(500)*	In	Alias

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetAliasPermission/Rq/Permission	Int(1)*	In	Additional capabilities of the alias within the service for the member banks (list of the member banks is specified in the <i>BICMemberList</i> array): 1 – transfers using the alias by the request (C2C Me2Me Pull).
FIMI/SetAliasPermission/Rq/Action	Int(1)*	In	Action: 0 – prohibit. 1 – allow.
FIMI/SetAliasPermission/Rq/BICMemberList	Str(20)[]	In	Array of BICs of the member bank (for the full list of the member banks, use the <i>GetServiceMembersList</i> request) to which the permission is granted. The value BIC ‘999999999’ indicates any member bank.
<b><i>SetAliasSubscription – managing alias subscriptions</i></b>			
FIMI/SetAliasSubscription/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/SetAliasSubscription/Rq/Serviceld	Int(1)*	In	Service identifier Available values: 4 – NSPK FPS C2B.
FIMI/SetAliasSubscription/Rq/Alias	Str(500)*	In	Alias
FIMI/SetAliasSubscription/Rq/Action	Int(1)*	In	Action: 1 – create. 2 – update data (account/status). 3 – delete/cancel.
FIMI/SetAliasSubscription/Rq/NewStatus	Int(1)	In	Subscription status Available values: NULL – do not change. 1 – active. 2 – suspended. The field is used for Action=2.
FIMI/SetAliasSubscription/Rq/QRCodeInfoTranId	Int(12)	In	ID of the transaction with data on QR code with the account subscription. The field transfers identifier of the transaction generated as part of the <i>GetQRCodeInfo</i> operation with the QR code check and receiving the code data. The field is used and mandatory for Action = 1.

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/SetAliasSubscription/Rq/Account	Str(30)	In	Account number.  The field is used for Action = 2, if the account number is not specified, the account is not changed. The field is used for Action = 1, if the account number is absent, a customer refused to create a subscription.
FIMI/SetAliasSubscription/Rq/AccountUID	Str(32)	In	Unique account identifier within the session. It can stand for the account number.
FIMI/SetAliasSubscription/Rq/SubscriptionToken	Str(32)	In	Unique identifier (token) of a subscription  The field is used and mandatory for Action = 2/3.
FIMI/SetAliasSubscription/Rp/SubscriptionToken	Str(32)	Out	Unique identifier (token) of a subscription

**2.3.3.15 On Arrest:**

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetArrests – request for list of arrests</b>			
FIMI/GetArrests/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/GetArrests/Rp/ArrestId	Str(32)	In	Arrest ID
FIMI/GetArrests/Rp/ArrestExtId	Str(250)	In	Arrest ID in the external system
FIMI/GetArrests/Rq/Account	Str(30)	In	Number of the account associated with the arrest
FIMI/GetArrests/Rq/AccountUID	Str(32)	In	Unique ID of the account associated with the arrest within the session. It can stand for the account number.
To search for the information on arrests, it is required to transfer one (or more) of key fields in the request: ArrestId, ArrestExtId, Account (or AccountUID). Otherwise, the request will be declined with code 62-InvalidParameterValue.			
FIMI/GetArrests/Rp/ArrestInfo	ArrRec	Out	{ str(4)* InstName; str(32)* ArrestId; str(250)* ArrestExtId; int* Priority; num* WithheldAmount; num* AmountHold; int* Currency; str Description; time StartDate; time EndDate; } } [] – information on arrests:
<p>InstName – institution name.</p> <p>ArrestId – arrest ID.</p> <p>ArrestExtId – arrest ID in the external system.</p> <p>Priority – priority.</p> <p>WithheldAmount – hold amount in the arrest currency.</p> <p>AmountHold – arrested amount.</p> <p>Currency – ISO code of the arrest currency.</p> <p>Description – description.</p> <p>StartDate – start time.</p> <p>EndDate – end time.</p>			

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetArrests/Rp/Accounts	ArrRec	Out	{           str(32)* ArrestId;           str(30)* Account;           str(32) AccountUID;           int(2)* Type;           int(1)* Status;           str(40) AggregateId;           int AggregatePriority;           num* WithheldAmount;           int* Currency;         } [] – arrest accounts:

ArrestId – arrest ID.

Account, AccountUID – account associated with the arrest.

Type – account type (1–checking; 11–savings; 31–credit; 91–bonus).

Status – account status:

- 0 – Inactive account.
- 1 – Open.
- 2 – Deposit only.
- 3 – Open primary account.
- 4 – Deposit only primary account.
- 5 – Information only.
- 9 – Closed.

AggregateId – accounts aggregate ID.

AggregatePriority – account priority in the aggregate.

WithheldAmount – arrested amount in the account currency.

Currency – ISO code of the account currency.

#### AddArrest – adding arrest

FIMI/AddArrest/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.
FIMI/AddArrest/Rq/ArrestId	Str(32)	In	Arrest ID. If NULL, ID is generated in TWO.
FIMI/AddArrest/Rq/ArrestExtId	Str(250)	In	Arrest ID in the external system
FIMI/AddArrest/Rq/Priority	Int*	In	Arrest priority
FIMI/AddArrest/Rq/AmountHold	Num*	In	Arrested amount
FIMI/AddArrest/Rq/Currency	Int*	In	ISO code of the arrest currency
FIMI/AddArrest/Rq/Description	Str(2000)	In	Description
FIMI/AddArrest/Rq/StartDate	Time	In	Start time. If NULL, time since the creation.
FIMI/AddArrest/Rq/EndDate	Time	In	End time. If NULL, it is unlimited.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddArrest/Rq/Accounts	ArrRec*	In	<pre>{     str(30) Account;     str(32) AccountUID; }</pre> <p>[]} – arrest accounts: Account – account number; AccountUID – unique account ID, it can stand for the account number.</p>
FIMI/AddArrest/Rq/ ArrestCategoryId	Int	In	Arrest category ID
FIMI/AddArrest/Rq/ ExecutiveDocumentCode	Int	In	Code of the enforcement document type
FIMI/AddArrest/Rq/ ExecutiveDocumentNumber	Str(60)	In	Number of the enforcement document
FIMI/AddArrest/Rq/ ExecutiveDocumentDate	Time	In	Judgement date
FIMI/AddArrest/Rq/ ExecutiveDocumentFIO	Str(100)	In	Full name of the official
Date in the ArrestCategoryId, ExecutiveDocumentCode, ExecutiveDocumentNumber, ExecutiveDocumentDate, ExecutiveDocumentFIO fields are not stored in <b>TWO</b> , it is transferred in <b>TWCMS</b> , the fields are mandatory if the arrest creation transaction is transferred to <b>TWCMS</b> .			
FIMI/AddArrest/Rp/ArrestId	Str(32)*	Out	Arrest ID
<b>DeleteArrest – deleting (releasing) arrest</b>			
FIMI/DeleteArrest/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b>.</p>
FIMI/DeleteArrest/Rq/ArrestId	Str(32)	In	Arrest ID
FIMI/DeleteArrest/Rq/ArrestExtId	Str(250)	In	Arrest ID in the external system
FIMI/DeleteArrest/Rq/ NeedReimbursReserves	Int(1)*	In	<p>Attribute of reimbursing reserves when releasing the amount from arrest: 1 – when deleting the arrest, return the reserved amount of the arrest to the account. 0 – when deleting the arrest, do not return the reserved amount of the arrest to the account.</p>
FIMI/DeleteArrest/Rq/ ChangeReason	Str(1000)	In	Reason for deleting
To execute the operation, the ArrestId field (or ArrestExtId) must be transferred in the request. Otherwise, the request will be declined with code 62- InvalidParameterValue.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>ChangeArrest – changing arrest</b>			
FIMI/ChangeArrest/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/ChangeArrest/Rq/ArrestId	Str(32)	In	Arrest ID
FIMI/ChangeArrest/Rq/ArrestExtId	Str(250)	In	Arrest ID in the external system
FIMI/ChangeArrest/Rq/AmountHold	Num	In	Arrested amount.  If NULL, it is not changed.
FIMI/ChangeArrest/Rq/StartDate	Time	In	Start time.  If NULL, it is not changed.
FIMI/ChangeArrest/Rq/EndDate	Time	In	End time.  If NULL, it is not changed. If the date is 31.12.9999, the unlimited value is set.
FIMI/ChangeArrest/Rq/ChangeReason	Str(1000)	In	Change reason
To execute the operation, the ArrestId field (or ArrestExtId) must be transferred in the request. Otherwise, the request will be declined with code 62- InvalidParameterValue.			

**2.3.3.16 On E-Commerce Merchant:**

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetECMerchant – request for list of E-Commerce merchants</b>			
FIMI/GetECMerchant/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/GetECMerchant/Rp/ECMerchantId	Str(32)	In	E-Commerce merchant ID
FIMI/GetECMerchant/Rq/RequestorId	Str(35)	In	Request ID
FIMI/GetECMerchant/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/GetECMerchant/Rq/MBR	Int(3)	In	
FIMI/GetECMerchant/Rq/CardUID	Str(64)	In	Unique card ID.  It can stand for the card number.
FIMI/GetECMerchant/Rq/Count	Int	In	Size of the requested list of merchants (no more than 1000).  If NULL, it is 1000.
FIMI/GetECMerchant/Rq/StartRowNumber	Int	In	Number of the row starting from which the list is requested.  If NULL, it is 1.
To search for the information on E-Commerce merchants, it is required to transfer one (or more) of the key fields in the request: ECMerchantId, RequestorId, PAN (or CardUID). Otherwise, the request will be declined with code 62- InvalidParameterValue.			
FIMI/GetECMerchant/Rp/ECMerchants	ArrRec	Out	<pre>{     str(4)* InstName;     str(32)* ECMerchantId;     str(35) RequestorId;     str(40) RequestorName;     str(2048) RequestorURL;     str(11) AcquirerBIN;     str(35) AcquirerMerchantId;     str(40) Name;     int MCC;     int Country;     time* ExpirationDate;     int(1)* Status;     str Commentary;     time* CreateDate;  } [] – information on E-Commerce merchants:</pre>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
InstName – institution name. ECMerchantId – E-Commerce merchant ID (in TWO). RequestorId – external identifier of the 3DSecure merchant. RequestorName – 3DSecure merchant name. RequestorURL – 3DSecure merchant URL. AcquirerBIN – acquirer BIN. AcquirerMerchantId – merchant ID. Name – merchant name. MCC – merchant category code. Country – ISO code of the country. ExpirationDate – validity period (if 31.12.9999, it is unlimited). Status – status (0 – not active; 1 – active; 2 – compromised). Commentary – comment. CreateDate – date of creating a merchant in the system.			
FIMI/GetECMerchant/Rp/ UserFields	ArrRec	Out	<pre>{     str(100)* Name;     str(4000) TextValue;     strXML XMLValue; }</pre> <p>} [] – array of structures, List of user-defined fields of the merchant is filled if only one E-Commerce merchant is found.</p> <p>Name, TextValue, XMLValue – name, text and XML values of user-defined fields.</p>
FIMI/GetECMerchant/Rp/ TrustedMerchants	ArrRec	Out	<pre>{     str(32)* ECMerchantId;     str(19)* PAN;     int(3)* MBR;     str(64) CardUID;     int* Status;     time LastAuthTime; }</pre> <p>} [] – information on the trusted E-Commerce merchants of the card, it is returned if the card is specified in the request (PAN/CardUID): PAN/MBR/CardUID – card.</p> <p>ECMerchantId – E-Commerce merchant ID (in TWO).</p> <p>Status – status of the card E-Commerce merchant (0 – inactive; 1 – active; 2 – compromised).</p> <p>LastAuthTime – time of the last merchant authorization.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>ChangeECMerchantStatus – changing E-Commerce merchant status</b>			
FIMI/ChangeECMerchantStatus/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/ChangeECMerchantStatus/Rq/ECMerchantId	Str(32)*	In	E-Commerce merchant ID
FIMI/ChangeECMerchantStatus/Rq/NewStatus	Int(1)*	In	Merchant or card-to-merchant link status to be set: 0 – inactive. 1 – active. 2 – compromised.
FIMI/ChangeECMerchantStatus/Rq/ExpirationDate	Time	In	Merchant lifetime If NULL, do not change. If 31.12.9999, it is unlimited. It is used if the merchant status is changed (request does not transfer the PAN/CardUID card data).
FIMI/ChangeECMerchantStatus/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/ChangeECMerchantStatus/Rq/MBR	Int(3)	In	
FIMI/ChangeECMerchantStatus/Rq/CardUID	Str(64)	In	Unique card ID. It can stand for the card number.
If the card data is not specified in the request (in the PAN/CardUID fields), the E-Commerce merchant status is changed. If the card data is specified, the E-Commerce merchant status is changed for the specified card.			
FIMI/ChangeECMerchantStatus/Rq/ChangeReason	Str(1000)	In	Status change reason
<b>DetachECMerchant – deleting card linked to E-Commerce merchant</b>			
FIMI/DetachECMerchant/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/DetachECMerchant/Rq/ECMerchantId	Str(32)*	In	E-Commerce merchant ID
FIMI/DetachECMerchant/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/DetachECMerchant/Rq/MBR	Int(3)	In	

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/DetachECMerchant/Rq/CardUID	Str(64)	In	Unique card ID. It can stand for the card number.
FIMI/DetachECMerchant/Rq/ChangeReason	Str(1000)	In	Reason for deleting the link between the card and E-Commerce merchant

#### 2.3.3.17 Invoice:

<b>GetInvoice – request for list of invoices on card/account</b>			
Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetInvoice/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .
FIMI/GetInvoice/Rq/InvoiceType	Int(1)*	In	Invoice type: 0 – decoupled 3DS authentication. 1 – outgoing request for funds within the C2C Me2Me Pull FPS service. 2 – incoming request for funds within the C2C Me2Me Pull FPS service. 3 – outgoing request within the C2G FPS service. 4 – outgoing request within the C2C FPS service. 5 – outgoing request within the C2B FPS service. 6 – incoming request for payment from the linked account within the C2B FPS service. 7 – outgoing request within the C2C International FPS service. 8 – outgoing request within the BQR2B FPS service.
FIMI/GetInvoice/Rq/InvoiceId	Str(32)	In	Invoice ID
FIMI/GetInvoice/Rq/PAN	Str(19)	In	Card number and card member number
FIMI/GetInvoice/Rq/MBR	Int(3)	In	
FIMI/GetInvoice/Rq/CardUID	Str(64)	In	Unique card ID. It can stand for the card number.
FIMI/GetInvoice/Rq/Account	Str(30)	In	Account number

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetInvoice/Rq/AccountUID	Str(32)	In	Unique account identifier within the session. It can stand for the account number.
FIMI/GetInvoice/Rq/Alias	Str(500)	In	Alias. It is used for InvoiceType = 1,2.
FIMI/GetInvoice/Rq/Mode	Int(1)	In	Request mode: 1 – invoices to be confirmed or rejected. 2 – confirmed or earlier rejected invoices. 3 – all invoices. If NULL, it is 1.
To search for the information on invoices, it is required to transfer one (or more) of key fields in the request: Invoiceld, PAN (or CardUID), Account(AccountUID), Alias. Otherwise, the request will be declined with code 62 – InvalidParameterValue.			
FIMI/GetInvoice/Rp/Invoices	ArrRec	Out	{ str(4)* InstName; str(32)* Invoiceld; str(19) PAN; int(3) MBR; str(64) CardUID; int(1)* Status; time* ExpirationDate; time* CreateDate; num* Amount; int* Currency; int* RequestTranId; str(150) RetailerName; str Description; str(30) Account; str(32) AccountUID; str(500) Alias; str(20) BIC; int PaymentTranId; int FinalAuthRespCode; int(1) MWLRequested; } [] – information on invoices:
InstName – institution name. Invoiceld – invoice ID (in TWO). PAN, MBR, CardUID – card number and card UID associated with the invoice. Status – status of the invoice confirmation by customer: 0 – rejected 1 – confirmed 2 – waiting for confirmation 3 – expired 4 – confirmed automatically (allowed for InvoiceType=2) 5 – complete (allowed for InvoiceType=2) 6 – confirmed by institution (allowed for InvoiceType=0) ExpirationDate – validity period (if 31.12.9999, it is unlimited). CreateDate – creation date. Amount – amount.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
Currency – ISO code of the currency. RequestTranId – ID of the transaction within which the invoice was submitted. RetailerName – merchant name. Description – description. Account – account number. AccountUID – unique account identifier. Alias – alias. BIC – bank BIC. PaymentTranId – identifier of a financial transaction of payment/transfer generated by the invoice. FinalAuthRespCode – final authorizer response code for a financial transaction of payment/transfer. MWLRequested – indicates that the cardholder confirmation is required to add the Ecommerce merchant to the white list (for InvoiceType=0).			
<b>ChangeInvoiceStatus – changing invoice status</b>			
FIMI/ChangeInvoiceStatus/Rq/ InstName			
Str(4) If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .			
FIMI/ChangeInvoiceStatus/Rq/ InvoiceType			
Int(1)* Invoice type: 0 - decoupled 3DS authentication 2 - incoming request for funds within the FPS C2C Me2Me Pull service.			
FIMI/ChangeInvoiceStatus/Rq/ Invoiceld			
Str(32)* Invoice ID			
FIMI/ChangeInvoiceStatus/Rq/ NewStatus			
Int(1)* Invoice status to be set: 0 – rejected. 1 – confirmed. 3 – expired. 6 – confirmed by institution (allowed for InvoiceType=0).			
Available status transitions: "waiting for confirmation" -> "rejected" "waiting for confirmation" -> "confirmed" "waiting for confirmation" -> "expired" "waiting for confirmation" -> "confirmed by institution" The "rejected", "expired", "confirmed", and "confirmed by institution" statuses are final, it is prohibited to change the status for the invoice with the current status "rejected", "expired", or "confirmed".			
FIMI/ChangeInvoiceStatus/Rq/ Account			
Str(30) Account number to be set. It must be associated with the same customer to which the current invoice account is linked. The field is used for InvoiceType = 2.			
FIMI/ChangeInvoiceStatus/Rq/ AccountUID			
Str(32) Unique (within the session) identifier of the account being set.			
FIMI/ChangeInvoiceStatus/ Rq/ChangeReason			
Str(1000) Status change reason			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ChangeInvoiceStatus/Rq/ MWLAccepted	Int(1)	In	Indicates the cardholder confirmation of adding the Ecommerce merchant to the white list.  The field is used for InvoiceType = 0.

### 2.3.3.18 On Corporate Customer:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetLegalEntityInfo – request for information on corporate customer</b>			
FIMI/GetLegalEntityInfo/Rq/ InstName	Str(4)	In	Character ID (name) of the financial institution where a corporate customer is created.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the corporate customer institution in <b>TWO</b> .
FIMI/GetLegalEntityInfo/Rq/ LegalEntityId	Int(16)	In	Corporate customer identifier
FIMI/GetLegalEntityInfo/Rq/ LegalEntityName	Str(150)	In	Corporate customer name  <i>The mask can be defined in the format of SQL-operator 'like', i.e. applying the metacharacters '_' (any single character), '%' (any number of any characters).</i>
FIMI/GetLegalEntityInfo/Rq/ LegalEntityExtId	Str(150)	In	External identifier of a corporate customer
It is allowed to search for a corporate customer using the identifier, name and external identifier. If all key attributes are absent in the request, the request will be declined.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetLegalEntityInfo/Rp/ LegalEntityInfo	ArrRec	Out	<pre>{     int(16)* LegalEntityId;     str(150)* LegalEntityName;     str(30) ShortName;     str(100) ShortNameTransliteration;     str(50) LegalEntityExtId;     str(20) RegistrationNumber;     time RegistrationDate;     str(80) RegisteringOrganization;     str(20) TaxPayerNumber;     int OKFS;     int OKOPF;     str(9) TaxReasonRegCode;     str(15) OGRN;     int EnterpriseCode;     str(2000) AdditionalInformation;     time ClosingDate; }</pre> <p>} [] – information on corporate customers:</p> <p>InstName – institution name.      LegalEntityId – corporate customer identifier.      LegalEntityName – corporate customer name.      ShortName – short name of a corporate customer.      ShortNameTransliteration – transliteration of the short name of a corporate customer.      LegalEntityExtId – external identifier.      RegistrationNumber – registration number.      RegistrationDate – registration date.      RegisteringOrganization – name of the organization that registered a corporate customer.      TaxPayerNumber – tax payer number.      OKFS – classifier of forms of ownership.      OKOPF – classifier of forms of incorporation.      TaxReasonRegCode – tax registration reason code.      OGRN – primary state registration number.      EnterpriseCode – enterprise code.      AdditionalInformation – additional information on a corporate customer.      ClosingDate – closing date.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetLegalEntityInfo/Rp/ LegalEntityAddress	ArrRec	Out	<pre>{     str(10)* Type;     int   Country;     str(240) CountryName;     str(20)  Region;     str(20)  City;     str(20)  Street;     str(20)  House;     str(20)  Building;     str(20)  Frame;     str(20)  Flat;     str(30)  ZIP;     str(200) Address;     str(20)  Phone;     str(20)  Fax;     str(50)  EMail; }</pre> <p>} [] – array of structures, list of the corporate customer addresses, it is filled if LegalEntityInfo transfers the information on one corporate customer.</p> <p>Type – address type (available values: legal – legal address, contact – contact address, postal – postal address).  Country – country ISO code.  CountryName – country name.  Region – region.  City – city.  Street – street.  House – house.  Building – building.  Frame – frame.  Flat – flat.  ZIP – index.  Address – address.  Phone – phone number.  Fax – fax number.  Email – E-mail address.</p>
FIMI/GetLegalEntityInfo/Rp/ UserFields	ArrRec	Out	<pre>{     str(100)* Name;     str(4000) TextValue;     strXML XMLValue; }</pre> <p>} [] – array of structures, list of user-defined fields of a corporate customer, it is filled if LegalEntityInfo transfers the information on one corporate customer.</p> <p>Name, TextValue, XMLValue – name, text and XML values of user-defined fields.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>CreateLegalEntity – creating corporate customer</b>			
FIMI/CreateLegalEntity/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution where a corporate customer is created.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the corporate customer institution in <b>TWO</b> .
FIMI/CreateLegalEntity/Rq/LegalEntityId	Int(16)	In	Corporate customer identifier  If not transferred, it is calculated automatically and returned in the response.
FIMI/CreateLegalEntity/Rq/LegalEntityName	Str(150)*	In	Corporate customer name
FIMI/CreateLegalEntity/Rq/ShortName	Str(30)	In	Short name of corporate customer
FIMI/CreateLegalEntity/Rq/ShortNameTransliteration	Str(100)	In	Transliteration of the short name of corporate customer
FIMI/CreateLegalEntity/Rq/LegalEntityExtId	Str(50)	In	External identifier
FIMI/CreateLegalEntity/Rq/RegistrationNumber	Str(20)	In	Registration number
FIMI/CreateLegalEntity/Rq/RegistrationDate	Time	In	Registration date
FIMI/CreateLegalEntity/Rq/RegisteringOrganization	Str(80)	In	Name of the organization that registered a corporate customer
FIMI/CreateLegalEntity/Rq/TaxPayerNumber	Str(20)	In	Tax payer number
FIMI/CreateLegalEntity/Rq/OKFS	Int	In	Classifier of forms of ownership
FIMI/CreateLegalEntity/Rq/OKOPF	Int	In	Classifier of forms of incorporation
FIMI/CreateLegalEntity/Rq/TaxReasonRegCode	Str(9)	In	Tax registration reason code
FIMI/CreateLegalEntity/Rq/OGRN	Str(15)	In	Primary state registration number
FIMI/CreateLegalEntity/Rq/EnterpriseCode	Int	In	Enterprise code
FIMI/CreateLegalEntity/Rq/AdditionalInformation	Str(2000)	In	Additional information on a corporate customer
FIMI/CreateLegalEntity/Rq/ClosingDate	Time	In	Closing date

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateLegalEntity/Rq/ LegalEntityAddress	ArrRec	In	<pre>{     str(10)* Type;     int   Country;     str(20) Region;     str(20) City;     str(20) Street;     str(20) House;     str(20) Building;     str(20) Frame;     str(20) Flat;     str(30) ZIP;     str(200) Address;     str(20) Phone;     str(20) Fax;     str(50) EMail; }</pre> <p>} [] – array of structures, list of the corporate customer addresses, it is filled if LegalEntityInfo transfers the information on one corporate customer.</p> <p><i>See the detailed description of fields in the FIMI/GetLegalEntityInfo/Rp/LegalEntityAddress structure.</i></p>
FIMI/CreateLegalEntity/Rp/ LegalEntityId	Int(16)*	Out	Corporate customer identifier
<b>DeleteLegalEntity – deleting corporate customer</b>			
FIMI/DeleteLegalEntity/Rq/ InstName	Str(4)	In	<p>Character ID (name) of the financial institution where a corporate customer is created.</p> <p>If the field is not defined (=NULL), the institution owning FIMI terminal stands for the corporate customer institution in <b>TWO</b>.</p>
FIMI/DeleteLegalEntity/Rq/ LegalEntityId	Int(16)*	In	Corporate customer identifier
<b>UpdateLegalEntity – updating information on corporate customer</b>			
FIMI/UpdateLegalEntity/Rq/ InstName	Str(4)	In	<p>Character ID (name) of the financial institution where a corporate customer is created.</p> <p>If the field is not defined (=NULL), the institution owning FIMI terminal stands for the corporate customer institution in <b>TWO</b>.</p>
FIMI/UpdateLegalEntity/Rq/ LegalEntityId	Int(16)*	In	Corporate customer identifier
FIMI/UpdateLegalEntity/Rq/ LegalEntityName	Str(150)	In	Corporate customer name

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateLegalEntity/Rq/ShortName	Str(30)	In	Short name of corporate customer
FIMI/UpdateLegalEntity/Rq/ShortNameTransliteration	Str(100)	In	Transliteration of the short name of corporate customer
FIMI/UpdateLegalEntity/Rq/LegalEntityExtId	Str(50)	In	External identifier
FIMI/UpdateLegalEntity/Rq/RegistrationNumber	Str(20)	In	Registration number
FIMI/UpdateLegalEntity/Rq/RegistrationDate	Time	In	Registration date
FIMI/UpdateLegalEntity/Rq/RegisteringOrganization	Str(80)	In	Name of the organization that registered a corporate customer
FIMI/UpdateLegalEntity/Rq/TaxPayerNumber	Str(20)	In	Tax payer number
FIMI/UpdateLegalEntity/Rq/OKFS	Int	In	Classifier of forms of ownership
FIMI/UpdateLegalEntity/Rq/OKOPF	Int	In	Classifier of forms of incorporation
FIMI/UpdateLegalEntity/Rq/TaxReasonRegCode	Str(9)	In	Tax registration reason code
FIMI/UpdateLegalEntity/Rq/OGRN	Str(15)	In	Primary state registration number
FIMI/UpdateLegalEntity/Rq/EnterpriseCode	Int	In	Enterprise code
FIMI/UpdateLegalEntity/Rq/AdditionalInformation	Str(2000)	In	Additional information on a corporate customer
FIMI/UpdateLegalEntity/Rq/ClosingDate	Time	In	Closing date
FIMI/UpdateLegalEntity/Rq/UpdateFieldsMethod	Int(1)	In	Mode of changing fields. If = 0 or NULL – only transferred fields will be updated. If = 1 – optional fields which are not sent will be set to NULL. The LegalEntityName field cannot be set to NULL, it will not be changed.

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/UpdateLegalEntity/Rq/ LegalEntityAddress	ArrRec	In	<pre>{     str(10)* Type;     int    Country;     str(20) Region;     str(20) City;     str(20) Street;     str(20) House;     str(20) Building;     str(20) Frame;     str(20) Flat;     str(30) ZIP;     str(200) Address;     str(20) Phone;     str(20) Fax;     str(50) EMail; }</pre> <p>} [] – array of structures, list of the corporate customer addresses, it is filled if LegalEntityInfo transfers the information on one corporate customer.</p> <p>If the subfield is not transferred in the structure, the attribute value is reset. If a certain address type is not transferred in the structure, the attributes on this type of address are not changed.</p> <p><i>See the detailed description of fields in the FIMI/GetLegalEntityInfo/Rp/LegalEntityAddress structure.</i></p>

#### ***SetLegalEntityUserFields – setting user-defined fields of corporate customer***

FIMI/SetLegalEntityUserFields/Rq/ InstName	Str(4)	In	<p>Character ID (name) of the financial institution where a corporate customer is created.</p> <p>If the field is not defined (=NULL), the institution owning FIMI terminal stands for the corporate customer institution in <b>TWO</b>.</p>
FIMI/SetLegalEntityUserFields/Rq/ LegalEntityId	Int(16)*	In	Corporate customer identifier
FIMI/SetLegalEntityUserFields/Rq/ UserFields	ArrRec	In	<pre>{     Str(100)* Name;     Str(3800) TextValue;     StrXML XMLValue. }</pre> <p>[] – array of structures with user-defined fields:</p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<p>Name – field name.          TextValue – text field value.          XMLValue – XML value of the field.</p> <p>The total length of the Name and TextValue fields of all records of the UserFields array must not exceed 3900 characters.</p> <p>If the TextValue and XMLValue fields are absent in the structure and user-defined field is present for a corporate customer, this user-defined field will not be changed. To reset TextValue and XMLValue, specify empty values for the TextValue and XMLValue fields.</p>			
<p>FIMI/SetLegalEntityUserFields/Rq/ RemoveMissingFields</p>			
	Int(1)	In	<p>1 – Remove missing fields.          By default, 0.</p>

## 2.3.4 Acquiring Requests

### 2.3.4.1 On Terminals:

#### 2.3.4.1.1 Terminal and Retailer Lists:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetRetailerList – request for retailers list</b>			
FIMI/GetRetailerList/Rq/RetailerId	Int	In	ID of the requested retailer, if none – ALL.
FIMI/GetRetailerList/Rq/ RetailerName	Str(150)	In	Name of the requested retailer; effective if RetailerId=NULL; if not defined, all the institution retailers are requested.
FIMI/GetRetailerList/Rq/ ArrayRetailerId	Int	In[]	Array of retailers ID.  If defined, it overrides the value of the <i>RetailerId</i> field.
FIMI/GetRetailerList/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the account  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the account institution in <b>TWO</b> .  <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name – it is allowed to request the retailers for the institution with AllowedFIID.AccessMask.Bit[1]=1 or AllowedFIID.AccessMask.Bit[3]=1</i>
FIMI/GetRetailerList/Rq/Count	Int(4)	In	Max number of the requested retailers, but not more than 1000 (NULL – all the found retailers satisfying the search conditions).  If = 0, the <i>List</i> of retailers is not filled, only the <i>FIMI/GetRetailerList/Rp/Count</i> field is specified

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetRetailerList/Rq/ StartRetailerNumber	Int	In	Serial number to sort the retailers by name. The number of the first returned retailer is as follows: StartRetailerNumber + 1
FIMI/GetRetailerList/Rp/List	ArrRec	Out	<pre>{     int* Id;     str(150)* Name;     str(150) Title;     int SIC;     str(240) SICName;     int Country;     str(240) CountryName;     str(80) StateName;     str(80) CountyName;     str(60) Region;     str(60) City;     str(100) Address;     str(30) ZIP;     str(30) Phone;     int* PurchaseDetention;     str(16) DefaultTerminalName;     int BackOfficeId;     int RiskLevel;     str(20) TaxId;     str(4)* InstName;     str(90) BranchId;     Int(1)* IsBlocked;     Time BlockedTime;     str(240) Branch; } [] – array of structures,</pre> <p>List of retailers:</p> <p>Id – ID.</p> <p>Name – name.</p> <p>Title – title.</p> <p>SIC – merchant category.</p> <p>SICName – MCC description.</p> <p>Country – country ISO code.</p> <p>CountryName – name of the country.</p> <p>StateName – name of the state.</p> <p>CountyName – name of the region.</p> <p>PurchaseDetention – purchase confirmation delay period.</p> <p>DefaultTerminalName – default terminal.</p> <p>BackOfficeId – retailer ID in Back-Office.</p> <p>RiskLevel – risk level.</p> <p>TaxId – tax ID.</p> <p>InstName – retailer institution.</p> <p>BranchId – retailer branch identifier.</p> <p>Branch – retailer branch name.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																
IsBlocked – "Access blocked" attribute. BlockedTime – block time.																			
FIMI/GetRetailerList/Rp/Count	Int	Out	Number of the found retailers																
<b>CreateRetailer – creating retailer</b>																			
FIMI/CreateRetailer/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution owning a retailer.</p> <p>If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer institution in <b>TWO</b>.</p> <p><i>For the list of available values, see the field InitSession/Rp/AllowedFI/D.Name.</i></p>																
FIMI/CreateRetailer/Rq/BranchId	Str(30)	In	Branch																
FIMI/CreateRetailer/Rq/RetailerPrototypeName	Str(150)	In	<p>Name of a retailer used as a prototype of the settings for the object to be created.</p> <p>If it is not specified, all the retailer settings will be determined by the prototype and re-defined with those transferred in the request.</p>																
FIMI/CreateRetailer/Rq/NeedCopyFromPrototype	Int	In	<p>Indicates that it is required to transfer additional data from the prototype terminal.</p> <p>The field is a bit mask:</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit Number</th><th>Description</th></tr> </thead> <tbody> <tr><td>1</td><td>External PS attributes</td></tr> <tr><td>2</td><td>Accounts</td></tr> <tr><td>3</td><td>Additional fields</td></tr> <tr><td>4</td><td>Multilingual fields</td></tr> <tr><td>5</td><td>Password</td></tr> <tr><td>6</td><td>Abonents</td></tr> <tr><td>7</td><td>Limits</td></tr> </tbody> </table> <p>It is used if RetailerPrototypeName is transferred.</p>	Bit Number	Description	1	External PS attributes	2	Accounts	3	Additional fields	4	Multilingual fields	5	Password	6	Abonents	7	Limits
Bit Number	Description																		
1	External PS attributes																		
2	Accounts																		
3	Additional fields																		
4	Multilingual fields																		
5	Password																		
6	Abonents																		
7	Limits																		
FIMI/CreateRetailer/Rq/BackOfficeId	Int	In	Back-Office ID																
FIMI/CreateRetailer/Rq/Name	Str(150)*	In	Retailer name																
FIMI/CreateRetailer/Rq>Title	Str(150)	In	Retailer title																
FIMI/CreateRetailer/Rq/SIC	Int(4)*	In	Merchant category code																
FIMI/CreateRetailer/Rq/CountryId	Int(3)	In	Country ISO code																
FIMI/CreateRetailer/Rq/Region	Str(60)	In	Region																
FIMI/CreateRetailer/Rq/City	Str(60)	In	City																

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateRetailer/Rq/Address	Str(100)	In	Address
FIMI/CreateRetailer/Rq/ZIP	Str(30)	In	Postal code
FIMI/CreateRetailer/Rq/Phone	Str(30)	In	Phone number
FIMI/CreateRetailer/Rq/ServicePhone	Str(16)	In	Phone number of the support service
FIMI/CreateRetailer/Rq/URL	Str(255)	In	Online store URL
FIMI/CreateRetailer/Rq/RiskLevel	Int(3)	In	Retailer risk level (from 0 to 999)
FIMI/CreateRetailer/Rq/CutoverStart	Int(4)	In	Start of the POS terminals cutover, min (from 0 to 1439) If NULL, it is 1200.
FIMI/CreateRetailer/Rq/CutoverEnd	Int(4)	In	End of the POS terminals cutover, min (from 0 to 1439) If NULL, it is 1260.
FIMI/CreateRetailer/Rq/TaxId	Str(20)	In	Terminal owner TPN
FIMI/CreateRetailer/Rq/TaxReasonRegCode	Str(10)	In	Tax registration reason code of a terminal owner
FIMI/CreateRetailer/Rq/ParentRetailerName	Str(150)	In	Parent retailer name
FIMI/CreateRetailer/Rq/ABURegistration	Int(1)	In	ABU participation status Available values: 0 – <i>Undefined</i> 1 – <i>Registration required</i> 2 – <i>Registered</i> 3 – <i>Unregistration required</i> 4 – <i>Unregistered</i> If NULL, it is 0.
FIMI/CreateRetailer/Rq/AnyPOSRefund	Int(1)	In	Indicates that refunds/adjustments/user reversals are allowed at various POS terminals of a retailer. Available values: 0 – <i>Prohibited</i> . 1 – <i>Allowed from any POS terminals of a retailer</i> . 2 – <i>Allowed from any POS terminals of a retailer and any retailer with the same parent</i> . If NULL, it is 0.
FIMI/CreateRetailer/Rq/IsBlocked	Int(1)	In	Blocked retailer indicator
FIMI/CreateRetailer/Rq/RetailerGroupName	Str(30)	In	Name of a retailer group <i>See the list of available values in InitSession in the RetailerGroups structure</i>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateRetailer/Rq/ ExtPSNames	ArrRec	In	<pre>{     Int(2)* Network;     Int(2)* FieldType;     Str(100)* ExtValue; } [] – external retailer attributes for the online transfer to the external payment system:</pre> <p>Network – network for which the external online attributes of a retailer are created in Back Office (see the list of available values in section 3.23).</p> <p>FieldType – retailer attribute being translated. Available values:</p> <ul style="list-style-type: none"> <li>1 – Name – retailer name transferred in the Retailer.Name(D) field.</li> <li>2 – Single Merchant Identifier – identifier of the retailer registered in VISA Cross-Border Domestic Interchange Programme.</li> <li>3 – Payment Facilitator ID. For details, refer to MasterCard Customer Interface Specification, DE 48 Subelement 37 Subfield 1; VISA BASE I Technical Specifications, DE104, Usage 2, Dataset ID 56 Tag 01; Technical Specifications on Bankcard Interoperability, DE 117, Usage PF.</li> <li>4 – Independent Sales Organization ID. For details, refer to MasterCard Customer Interface Specification, DE 48 Subelement 37 Subfield 2.</li> <li>5 – Sub-Merchant ID. For details, refer to MasterCard Customer Interface Specification, DE 48 Subelement 37 Subfield 3.</li> <li>6 – list of VISA Fee Program Indicator values separated by ':'.</li> <li>7 – MasterCard Assigned ID, for details, refer to MasterCard Customer Interface Specification, DE 48 Subelement 32; MIR Assigned Id (NSPK DE 48.16).</li> <li>8 – Digital Wallet Operator – indicates whether the retailer is a digital wallets operator. Available values: 'Y' and 'N'.</li> <li>9 – VISA Merchant Verification Value. For details, refer to VISA SMS/BASE I Technical Specifications, Volume 1, Field 62.20.</li> <li>10 – MasterCard Merchant Capability. For details, refer to MasterCard IPM Clearing Formats, PDS 0042.</li> <li>11 – MasterCard Domestic Card Acceptor TAX ID, PDS0180, string format up to 20 characters.</li> <li>12 – Legal Corporate Name. For details, refer to MasterCard IPM Clearing Formats, PDS 0173 - Legal Corporate Name.</li> <li>13 – Tier numbers of minimum annual volume of transactions needed to qualify for the interchange programs.</li> <li>14 – Merchant Tax ID Type for Visa / Card Acceptor Type for Mastercard.</li> <li>15 – Visa Merchant Identifier, for details, refer to VISA SMS/BASE I Technical Specifications, Volume 1, Field 126.5.</li> <li>16 – Merchant Account Information for QR service.</li> <li>17 – Foreign Retailer Indicator. Available values: 1 and 0. It is used for Visa.</li> <li>18 – Government-controlled – indicates whether or not the retailer is government-controlled, available values – 1 and 0. It is used for Mastercard.</li> <li>19 – Merchant Country of Origin – code of the retailer country of origin. It is used for Mastercard.</li> <li>20 – INTES (International Establishment Code). It is used for Diners Club.</li> <li>21 – Microenterprise Indicator. Available values: 1 and 0. It is used for Mastercard.</li> <li>22 – MCC.</li> <li>23 – Region code/name – code or name of the retailer region.</li> </ul> <p>ExtValue – value of the attribute being translated. It is meant for the online transfer to the payment system.</p>
FIMI/CreateRetailer/Rq/Accounts	ArrRec	In	<pre>{     Int(3)* Currency;     Str(100)* Account; } [] – retailer accounts:</pre> <p>Currency – currency (ISO code). Account – account.</p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateRetailer/Rq/SubMerchantCheck	Int(1)	In	Indicates the check of the retailer sub-merchants, it is specified for the facilitator retailer.  Available values: NULL, 0 – do not check sub-merchant. 1 – check sub-merchant.
FIMI/CreateRetailer/Rq/SubMerchantData	Int(1)	In	Sub-merchant data in the transaction, it is specified for the facilitator retailer.  Available values: NULL, 1 – use data from the transaction originator. 2 – use data from the retailer-sub-merchant settings.
FIMI/CreateRetailer/Rq/MCCCheckList	Str(1000)	In	Lists of allowed and prohibited MCC values, it is specified for the facilitator retailer or sub-merchant retailer.  The field format is a structure of the following type: 0-9999, !5962, !5966, !6051-7994 where: <ul style="list-style-type: none"><li>- Values and ranges are separated by comma.</li><li>- Start and end values in the range are separated by dash (0x2D).</li><li>- Prohibited values and ranges are preceded with character '!'. </li><li>- Spaces between values are allowed but optional.</li></ul>
FIMI/CreateRetailer/Rq/Facilitators	ArrRec	In	{ Str(150)* RetailerName; Str(32)* SubMerchantId; Str(150) SubMerchantName; } [] – list of the facilitator retailers for which the created retailer is a submerchant.
 RetailerName – name of the facilitator retailer. SubMerchantId – sub-merchant identifier within the facilitator retailer. SubMerchantName – unique sub-merchant name within the facilitator retailer.			
FIMI/CreateRetailer/Rp/RetailerId	Int*	Out	Retailer ID
<b>UpdateRetailer – request for updated information on retailer</b>			
FIMI/UpdateRetailer/Rq/Id	Int	In	Retailer ID in TWO.  It is mandatory if the BackOfficeId field is not transferred.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateRetailer/Rq/BackOfficeId	Int	In	Retailer ID in Back-Office. It can be transferred instead of the Id field.
FIMI/UpdateRetailer/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning a retailer. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer institution in TWO. <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name.</i>
FIMI/UpdateRetailer/Rq/Name	Str(150)	In	Retailer name
FIMI/UpdateRetailer/Rq>Title	Str(150)	In	Retailer title
FIMI/UpdateRetailer/Rq/SIC	Int	In	MCC
FIMI/UpdateRetailer/Rq/Country	Int	In	Country ISO code
FIMI/UpdateRetailer/Rq/Region	Str(60)	In	Region
FIMI/UpdateRetailer/Rq/RegionExternalId	Str(20)	In	Region external ID
FIMI/UpdateRetailer/Rq/City	Str(60)	In	City
FIMI/UpdateRetailer/Rq/CityExternalId	Str(20)	In	City external ID
FIMI/UpdateRetailer/Rq/Address	Str(100)	In	Address
FIMI/UpdateRetailer/Rq/ZIP	Str(30)	In	Postal code
FIMI/UpdateRetailer/Rq/Phone	Str(30)	In	Phone number
FIMI/UpdateRetailer/Rq/ServicePhone	Str(16)	In	Phone number of the support service
FIMI/UpdateRetailer/Rq/URL	Str(255)	In	Online store URL
FIMI/UpdateRetailer/Rq/PurchaseDetention	Int	In	Purchase confirmation delay period (in days)
FIMI/UpdateRetailer/Rq/RemoveMissingFields	Int(1)	In	Indicates whether it is required to remove the fields that are not sent. The fields are removed if =1. The default value is 0
FIMI/UpdateRetailer/Rq/ChangeReason	Str(1000)	In	Information update reason
If the optional field is not specified or =NULL, it is not updated.			
<i>RegionExternalId</i> and <i>CityExternalId</i> are filled if the <i>Region</i> and <i>City</i> fields are defined and the transaction is to be unloaded in the offline exchange file. <i>RegionExternalId</i> and <i>CityExternalId</i> are not saved in TWO, they are used only to transfer the identifiers in the file for transactions offline exchange between TWO and TWCMS.			
The <i>Name</i> and <i>PurchaseDetention</i> fields cannot be deleted. If <i>RemoveMissingFields</i> =1 and these fields are not filled, the information will remain unchanged.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetRetailerLimits – request for list of retailer limits</b>			
FIMI/GetRetailerLimits/Rq/InstName	Str(4)	In	<p>Character ID (name) of the financial institution owning a retailer.</p> <p>If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer institution in <b>TWO</b>.</p> <p><i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name.</i></p>
FIMI/GetRetailerLimits/Rq/RetailerId	Int	In	<p>Retailer ID</p> <p><i>For the list of available values, see GetRetailerList.</i></p>
FIMI/GetRetailerLimits/Rq/RetailerName	Str(150)	In	<p>Retailer name</p> <p>It can be specified instead of the retailer ID (RetailerId).</p>
FIMI/GetRetailerLimits/Rq/GetCurrentCounters	Int(1)	In	<p>If NULL then = 0.</p> <p>If = 1, the <i>Limits[i].Current</i> fields of the response will return current values of counters; the <i>AuthScheme</i> field must be defined in the request.</p>
FIMI/GetRetailerLimits/Rq/AuthScheme	Int	In	<p>Authorization scheme ID. The field must be defined if <i>GetCurrentCounters</i>=1. The field is used to request values of the group limit of a retailer of the authorization scheme. If NULL, certain values of the group limit are not displayed.</p> <p><i>For the list of available values, see the field InitSession/Rp/AuthSchemes.</i></p>
FIMI/GetRetailerLimits/Rp/Limits	ArrRec	Out	<p>{</p> <p>int* LimitId; num Max; num* Current; int(2) PeriodType; int Period; time DateTimeForReset; time LastResetTime;</p> <p>} [] – array of structures, list of limits with their maximum and current values:</p>
<p>LimitId – limit ID in <b>TWO</b> (see p.3.3).</p> <p>Max – maximum value of the limit (0 – restricted, NULL – unlimited, &gt;0 – defined, -1 – as auth. scheme value).</p> <p>Current – current value of the limit.</p> <p>LastResetTime – date/time of the last limit counter reset.</p> <p>PeriodType – period type:</p> <ul style="list-style-type: none"> <li>- NULL – period type and limit reset time within the period are unknown.</li> <li>- 0 – Absolute (Daily) – absolute period; it starts from the beginning of the day (1440 – it is reset every day at 0:00, 2880+420 – at 7:00 every second day).</li> <li>- 1 – Weekly – calendar weekly period; it starts from the beginning of the week (2880+420 – it is reset every Wednesday at 7:00).</li> <li>- 2 – Monthly – calendar monthly period; it starts from the beginning of the month (2880+420 – it is reset on the third day of every month at 7:00).</li> </ul>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description						
<ul style="list-style-type: none"> <li>- 3 – Quarterly – calendar quarterly period; it starts from the beginning of the quarter (2880+420 – it is reset on the third day of every quarter at 7:00).</li> <li>- 4 – Infinite – limit counter will never be reset; the period is optional.</li> <li>- 5 – Single operation – one-time limit (limit for a single operation); limit counter is always reset before the limit check; the period is optional.</li> <li>- 6 – Reset in Refresh – limit counter is reset with Refresh; the period is optional.</li> <li>- 7 – Yearly – the calendar year limit, the period starts from the beginning of the year.</li> </ul> <p>(Period=0 - the counter is reset on the 1st of January at 00:00;    Period=2880+420 – the counter is reset each year on the 3<sup>rd</sup> of January at 7:00)</p> <ul style="list-style-type: none"> <li>- - 11 - Calendar – reset on the calendar date and time (for a particular type, the DateTimeForReset field is used instead of the Period field).</li> </ul> <p>Period – limit reset time within the calendar period in minutes:    Period/1440 - number of day (starts from 0); Mod(Period,1440) - reset time within a day (min)</p> <p>Restrictions:</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">PeriodType=0 → 1440&lt;=Period&lt;=1440000</td> <td style="width: 50%;">PeriodType=1 → 0&lt;=Period&lt;1440*7</td> </tr> <tr> <td>PeriodType=2 → 0&lt;=Period&lt;1440*31</td> <td>PeriodType=3 → 0&lt;=Period&lt;1440*90</td> </tr> <tr> <td>PeriodType=7 → 0&lt;=Period&lt;1440*360</td> <td></td> </tr> </table> <p>DateTimeForReset – calendar date and time of the current limit value reset.</p>				PeriodType=0 → 1440<=Period<=1440000	PeriodType=1 → 0<=Period<1440*7	PeriodType=2 → 0<=Period<1440*31	PeriodType=3 → 0<=Period<1440*90	PeriodType=7 → 0<=Period<1440*360	
PeriodType=0 → 1440<=Period<=1440000	PeriodType=1 → 0<=Period<1440*7								
PeriodType=2 → 0<=Period<1440*31	PeriodType=3 → 0<=Period<1440*90								
PeriodType=7 → 0<=Period<1440*360									
FIMI/GetRetailerLimits/Rp/ LimCurrency	Int*	Out	Currency of financial limits						
<b>SetRetailerLimits – changing maximum values and validity periods of retailer limits</b>									
FIMI/SetRetailerLimits/Rq/ InstName	Str(4)	In	<p>Name of the financial institution owning a retailer.</p> <p>If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer institution in <b>TWO</b>.</p>						
FIMI/SetRetailerLimits/Rq/ RetailerId	Int	In	<p>Retailer ID</p> <p><i>For the list of available values, see GetRetailerList.</i></p>						
FIMI/SetRetailerLimits /Rq/ RetailerName	Str(150)	In	<p>Retailer name</p> <p>It can be specified instead of the retailer ID (RetailerId).</p>						
FIMI/SetRetailerLimits/Rq/ RemoveMissingLimits	Int(1)	In	<p>0 – do not remove, add/update only.</p> <p>1 – remove missing limits.</p> <p>2 – remove received limits.</p> <p>If NULL, it is 0.</p>						
FIMI/SetRetailerLimits/Rq/Limits	ArrRec	In	<pre>{   int* LimitId;   num Max;   int(2) PeriodType;   int Period;   time DateTimeForReset; }</pre> <p>[]} – array of structures, list of limits with their maximum values:</p>						
<p>LimitId – limit ID in <b>TWO</b> (refer to section 3.3).</p> <p>Max – maximum limit value in the account currency (0 - restricted, NULL - unlimited, &gt;0 - defined, -1 - as auth. scheme value).</p> <p>PeriodType, Period, DateTimeForReset – limit validity period (see the description in <i>GetRetailerLimits</i>).</p>									

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetRetailerLimits/Rq/ChangeReason	Str(1000)	In	Change reason
<b>GetRetailerSchemeLimits – request for values of limits by scheme-level retailers</b>			
FIMI/GetRetailerSchemeLimits/Rq/AuthScheme	Int*	In	Authorization scheme ID. <i>See the list of available values in the field InitSession/Rp/AuthSchemes.</i>
FIMI/GetRetailerSchemeLimits/Rp/Limits	ArrRec	Out	{ int* LimitId; num Max; int(2) PeriodType; int Period; time DateTimeForReset; } [] – array of structures, list of limits with their maximum values and validity periods (see the description of the limit validity periods in the GetRetailerLimits operation)
<b>ResetRetailerLimitCounters – reset of retailer limits counter</b>			
FIMI/ResetRetailerLimitCounters/Rq/InstName	Str(4)	In	Name of the financial institution owning a retailer.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer institution in <b>TWO</b> .
FIMI/ResetRetailerLimitCounters/Rq/RetailerId	Int	In	Retailer ID <i>For the list of available values, see GetRetailerList.</i>
FIMI/ResetRetailerLimitCounters/Rq/RetailerName	Str(150)	In	Retailer name  It can be specified instead of the retailer ID (RetailerId).
FIMI/ResetRetailerLimitCounters/Rq/Limits	Int[]*	In	List of identifiers of the limits for which the counter is reset
FIMI/ResetRetailerLimitCounters/Rq/ChangeReason	Str(1000)	In	Reason for the limit counter reset
<b>GetRetailerGroupLimits – request for list of retailer group limits</b>			
FIMI/GetRetailerGroupLimits/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning a group of retailers.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer group institution in <b>TWO</b> .  <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name.</i>
FIMI/GetRetailerGroupLimits/Rq/RetailerGroupId	Int	In	ID of the retailer group.  <i>See the list of available values in InitSession in the RetailerGroups structure.</i>
FIMI/GetRetailerGroupLimits/Rq/	Str(30)	In	Name of the group of retailers.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
RetailerGroupName			It can be specified instead of the retailer group ID (RetailerGroupId).
FIMI/GetRetailerGroupLimits/Rq/ GetCurrentCounters	Int(1)	In	If NULL then = 0. If = 1, the <i>Limits[i].Current</i> fields of the response will return current values of counters; the <i>AuthScheme</i> field must be defined in the request.
FIMI/GetRetailerGroupLimits/Rq/ AuthScheme	Int	In	Authorization scheme ID; the field must be defined if <i>GetCurrentCounters</i> =1; the field is used to request values of the group limit of a customer of the authorization scheme, if NULL, certain values of the group limit are not displayed. <i>For the list of available values, see the field InitSession/Rp/AuthSchemes.</i>
FIMI/GetRetailerGroupLimits/Rp/ Limits	ArrRec	Out	{ int* LimitId; num Max; num* Current; int(2) PeriodType; int Period; time DateTimeForReset; time LastResetTime; } [] – array of structures, list of limits with their maximum and current values:

LimitId – limit ID in **TWO** (refer to section 3.3).

Max – maximum limit value (0 – restricted, NULL – unlimited, >0 – defined, -1 – as auth. scheme value).

Current – current limit value.

LastResetTime – date/time of the last limit counter reset.

PeriodType, Period – limit validity period (see the description in the *GetRetailerLimits* message).

DateTimeForReset – calendar date and time of the current limit value reset.

FIMI/GetRetailerGroupLimits/Rp/ LimCurrency	Int*	Out	Currency of financial limits
--	------	-----	------------------------------

#### **SetRetailerGroupLimits – changing maximum values and validity periods of retailer group limits**

FIMI/SetRetailerGroupLimits/Rq/ InstName	Str(4)	In	Name of the financial institution owning a group of retailers. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer group institution in <b>TWO</b> .
FIMI/SetRetailerGroupLimits/Rq/ RetailerGroupId	Int	In	Retailer group ID. <i>See the list of available values in InitSession in the RetailerGroups structure.</i>
FIMI/SetRetailerGroupLimits/Rq/ RetailerGroupName	Str(30)	In	Retailer group name. It can be specified instead of the retailer group ID (RetailerGroupId).

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/SetRetailerGroupLimits/Rq/RemoveMissingLimits	Int(1)	In	0 – do not remove, add/update only. 1 – remove missing limits. 2 – remove received limits. If NULL, it is 0.
FIMI/SetRetailerGroupLimits/Rq/Limits	ArrRec	In	{ int* LimitId; num Max; int(2) PeriodType; int Period; time DateTimeForReset; } [] – array of structures, list of limits with their maximum values:  LimitId – limit ID in TWO (refer to section 3.3). Max – maximum limit value in the account currency (0 - restricted, NULL - unlimited, >0 - defined, -1 - as auth. scheme value). PeriodType, Period, DateTimeForReset – limit validity period (see the description in GetRetailerLimits).
<b>GetRetailerGroupSchemeLimits – request for values of limits by scheme-level retailer group</b>			
FIMI/GetRetailerGroupSchemeLimits/Rq/AuthScheme	Int*	In	Authorization scheme ID <i>See the list of available values in the field InitSession/Rp/AuthSchemes.</i>
FIMI/GetRetailerGroupSchemeLimits/Rp/Limits	ArrRec	Out	{ int* LimitId; num Max; int(2) PeriodType; int Period; time DateTimeForReset; } [] – array of structures, List of limits with their maximum values and validity periods (for the description of the limit validity periods, see the GetRetailerLimits operation)
<b>ResetRetailerGroupLimitCounters – reset of retailer group limit counter</b>			
FIMI/ResetRetailerGroupLimitCounters/Rq/InstName	Str(4)	In	Name of the financial institution owning a group of retailers. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the retailer group institution in TWO.
FIMI/ResetRetailerGroupLimitCounters/Rq/RetailerGroupId	Int	In	ID of the retailer group. <i>See the list of available values in InitSession in the RetailerGroups structure.</i>
FIMI/ResetRetailerGroupLimitCounters/R	Str(30)	In	Name of the retailer group.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
q/RetailerGroupName			It can be specified instead of the retailer group ID (RetailerGroupId).
FIMI/ResetRetailerGroupLimitCounters/Rq/Limits	Int[]*	In	List of identifiers of limits for which the counter is reset
FIMI/ResetRetailerGroupLimitCounters/Rq/ChangeReason	Str(1000)	In	Reason for the limit counter reset
<b>GetTermList – request for terminals list</b>			
FIMI/GetTermList/Rq/TermId	Int	In	Terminal ID
FIMI/GetTermList/Rq/ArrayTermId	Int[...]	In	Array of terminal IDs. If set, it overlays the value of the TermId, TermName field.
FIMI/GetTermList/Rq/TermName	Str(16)	In	Terminal name
FIMI/GetTermList/Rq/Retailer	Int	In	Retailer ID. NULL (the field is not defined) – all -1 – terminals without retailer (bank terminals)
FIMI/GetTermList/Rq/Class	Int[...]	In	Array – terminal classes to be returned to the list: 1 – ATM 2 – POS 3 – CRT 4 – VTBI  Class[] = NULL – all the terminal classes are requested
FIMI/GetTermList/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning the account If not defined (=NULL), the institution owning FIMI terminal stands for the account institution in TWO.  For the list of available values, see the field <i>InitSession/Rp/AllowedFIID.Name</i> – it is allowed to request the terminals of the institution with <i>AllowedFIID.AccessMask.Bit[1]=1</i> or <i>AllowedFIID.AccessMask.Bit[3]=1</i>
FIMI/GetTermList/Rq/Count	Int(4)	In	Max number of the requested terminals (NULL – all terminals satisfying the defined conditions, but not more than 1000). If = 0, the List of terminals is not filled, only the <i>FIMI/GetTermList/Rp/Count</i> field is defined.
FIMI/GetTermList/Rq/StartTerminalNumber	Int	In	Serial number for sorting the terminals by the terminal name. The number of the first returned terminal will be StartTerminalNumber + 1

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetTermList/Rq/Use	Int(1)	In	<p>Terminals selection parameter.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>- 0, NULL – all terminals, additional filtering is disabled</li> <li>- 1 – return ATM terminals whose units are in use (Class is refilled with the value 1 – ATM)</li> <li>- 2 – return deactivated POS terminals (Class is refilled with the value 2 – POS)</li> <li>- 3 – return active POS terminals (Class is refilled with the value 2 – POS)</li> </ul>
FIMI/GetTermList/Rq/ SerialNumber	Str(50)	In	Serial number of the terminal
FIMI/GetTermList/Rp/List	ArrRec	Out	<pre>{     int* Id;     int* Class;     int* Type;     str(16)* Name;     str(250) Title;     int Retailer;     int SIC;     str(240) SICName;     int* Currency;     str(240) CurrencyName;     int* Country;     str(240) CountryName;     str(80) StateName;     str(80) CountyName;     str(90) Owner;     str(90) Region;     str(90) Branch;     str(90) City;     str(300) Location;     str(90) ZIP;     int TimeOffset;     str(90) Phone;     str(750) Description;     int(3) RiskLevel;     int(1)* PerformRiskControl;     int(1)* AcqControlDisabled;     str(4)* InstName;     str(50) SerialNumber;     str(240) BranchName; }</pre> <p>} [] – array of structures, list of the terminals:</p>

Class – class (1-ATM, 2-POS, 3-CRT, 4-VTBI).  
Type – terminal type (see p. 3.3).  
Name – terminal name.  
Title – terminal title.  
Retailer – retailer ID.  
SIC – terminal MCC.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description												
SICName – MCC description (if present). TimeOffset – time offset (min) between the terminal time zone and the processing center time zone. RiskLevel – terminal risk level (from 0 up to 999 or NULL). PerformRiskControl – flag ‘Perform Risk Control’. AcqControlDisabled – flag ‘Acquiring Risk Control Disabled’; if it =1, one transaction will be conducted without risk control. After that the flag will be automatically unset. SerialNumber – serial number of the terminal.															
FIMI/GetTermList/Rp/Count	Int	Out	Number of the found terminals												
<b>TerminalRiskControl – setting terminal risk level</b>															
FIMI/TerminalRiskControl/Rq/TermId	Int	In	Terminal ID												
FIMI/TerminalRiskControl/Rq/TermName	Str(16)	In	Terminal name. It can be specified instead of the terminal ID												
FIMI/TerminalRiskControl/Rq/RiskLevel	Int(3)	In	Terminal risk level. Available values: from 0 to 999 or NULL												
FIMI/TerminalRiskControl/Rq/ChangeReason	Str(1000)	In	Reason for the terminal risk level change												
<b>CreateTerminal – creating terminal</b>															
FIMI/CreateTerminal/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning a terminal. If the field is not defined (=NULL), the institution owning FIMI terminal stands for the terminal institution in <b>TWO</b> . <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name.</i>												
FIMI/CreateTerminal/Rq/Branch	Str(30)	In	Branch												
FIMI/CreateTerminal/Rq/TerminalPrototypeName	Str(16)	In	Name of a terminal used as a prototype of the settings for the object to be created. If it is not specified, all the terminal settings will be determined by the prototype and re-defined with those transferred in the request.												
FIMI/CreateTerminal/Rq/NeedCopyFromPrototype	Int	In	Indicates that it is required to transfer additional data from the prototype terminal. The field is a bit mask: <table border="1" style="margin-left: 20px;"> <tr> <th>Bit Number</th> <th>Description</th> </tr> <tr> <td>1</td> <td>External PS attributes</td> </tr> <tr> <td>2</td> <td>Accounts</td> </tr> <tr> <td>3</td> <td>Additional fields</td> </tr> <tr> <td>4</td> <td>Multilingual fields</td> </tr> <tr> <td>5</td> <td>Terminal keys (for the ATM/POS terminals)</td> </tr> </table>	Bit Number	Description	1	External PS attributes	2	Accounts	3	Additional fields	4	Multilingual fields	5	Terminal keys (for the ATM/POS terminals)
Bit Number	Description														
1	External PS attributes														
2	Accounts														
3	Additional fields														
4	Multilingual fields														
5	Terminal keys (for the ATM/POS terminals)														

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			It is used if TerminalPrototypeName is transferred.
FIMI/CreateTerminal/Rq/BackOfficeId	Int	In	Back-Office ID
FIMI/CreateTerminal/Rq/Type	Int(3)	In	Terminal type. See the list of available values in section 3.4.  The field is mandatory if the prototype terminal name (TerminalPrototypeName) is not transferred. If the prototype terminal name is transferred, it is allowed not to transfer the field, in this case the terminal type is identified by the prototype terminal.
FIMI/CreateTerminal/Rq/SubType	Int(1)	In	Terminal subtype. Available values: 0 – unknown; 1 – NCR 56xx(for ATM NDC+); 2 – Operating point (for FIMI); 5 – 3-D Secure 1.0 (for ECommerce); 6 – 3-D Secure 2.x (for ECommerce); 7 – 3-D Secure 2.x CB (for ECommerce).  If NULL, it is 0.
FIMI/CreateTerminal/Rq/ProtocolVersion	Int	In	Protocol version. If NULL, it is 0.
FIMI/CreateTerminal/Rq/RouteingScheme	Int	In	Routing scheme ID If NULL, it is 0 (Default).
FIMI/CreateTerminal/Rq/AcqGroupId	Int	In	Acquiring group ID
FIMI/CreateTerminal/Rq/CfgGroup	Int	In	Configuration group ID. It is used for the ATMs and POS terminals in TWO.  If NULL, it is 1 (Default).
FIMI/CreateTerminal/Rq/SIC	Int(4)	In	Merchant category code  It is used and mandatory for the POS terminals and imprinter
FIMI/CreateTerminal/Rq/Name	Str(16)*	In	Terminal name in Latin letters
FIMI/CreateTerminal/Rq>Title	Str(250)	In	Terminal title
FIMI/CreateTerminal/Rq/Description	Str(750)	In	Detailed description of a terminal that can include its location, owner, etc.
FIMI/CreateTerminal/Rq/RetailerName	Str(150)	In	Terminal retailer name  It is used for the POS terminals and imprinter
FIMI/CreateTerminal/Rq/Currency	Int(3)*	In	Default terminal currency (ISO code)

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateTerminal/Rq/CountryId	Int(3)	In	Country ISO code
FIMI/CreateTerminal/Rq/County	Int	In	County
FIMI/CreateTerminal/Rq/Owner	Str(90)	In	Terminal owner (in Latin letters)
FIMI/CreateTerminal/Rq/Region	Str(90)	In	Region (depending on the payment system where the terminal is used, the field can be transferred in Latin letters)
FIMI/CreateTerminal/Rq/City	Str(90)	In	City (in Latin letters)
FIMI/CreateTerminal/Rq/Location	Str(300)	In	Location (in Latin letters)
FIMI/CreateTerminal/Rq/Longitude	Num(18,9)	In	Longitude
FIMI/CreateTerminal/Rq/Latitude	Num(18,9)	In	Latitude
FIMI/CreateTerminal/Rq/ZIP	Str(90)	In	Postal index
FIMI/CreateTerminal/Rq/Phone	Str(90)	In	Phone number
FIMI/CreateTerminal/Rq/SerialNumber	Str(50)	In	Serial number
FIMI/CreateTerminal/Rq/TimeOffset	Int(4)	In	Terminal time offset relative to the Processing Center time, in minutes If NULL, it is 0.
FIMI/CreateTerminal/Rq/RiskLevel	Int(3)	In	Terminal risk level. Available values: from 0 to 999. If NULL, it is 0.
FIMI/CreateTerminal/Rq/PerformRiskControl	Int(1)	In	"Perform acquiring risk control for terminal" attribute. If NULL, it is 1.
FIMI/CreateTerminal/Rq/Active	Int(1)	In	Indicates that the terminal is active. If NULL, it is 1.
FIMI/CreateTerminal/Rq/ParentRetailerName	Str(50)	In	Parent retailer name It is used if the ParentTerminalName field is specified for the POS terminals and imprinter
FIMI/CreateTerminal/Rq/ParentTerminalName	Str(50)	In	Parent terminal name It is used for the POS terminals, voice authorization terminal and imprinter
FIMI/CreateTerminal/Rq/SelfService	Int(1)	In	"Self-service terminal" attribute, i.e. Customer Activated Terminal. If NULL, it is 0.
FIMI/CreateTerminal/Rq/POSPINEEntryCapability	Int(1)	In	"POS terminal supports the PIN entry" attribute If NULL, it is 1.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																																							
FIMI/CreateTerminal/Rq/POSCategory	Int(1)	In	<p>POS terminal category</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>NULL, 0 – common terminal;</li> <li>1 – Electronic cash register;</li> <li>3 – Limited-amount terminal;</li> <li>4 – In-flight commerce;</li> <li>7 – Transponder device;</li> <li>8 – MPOS Acceptance Device / Dedicated mPOS Terminal with PCI compliant dongle;</li> <li>9 – MPOS Acceptance Device / Off the Shelf Mobile Device</li> </ul>																																							
FIMI/CreateTerminal/Rq/EntryCapsExtended	Int	In	<p>Terminal card read capabilities.</p> <p>The field is a bit mask:</p> <table border="1"> <thead> <tr> <th>Bit Number</th><th>Description</th><th>Allowed</th></tr> </thead> <tbody> <tr><td>NULL</td><td>None</td><td>All</td></tr> <tr><td>1</td><td>Unknown</td><td>All</td></tr> <tr><td>2</td><td>NoTerminal</td><td>All</td></tr> <tr><td>3</td><td>MagStripe</td><td>ATM, POS</td></tr> <tr><td>4</td><td>BarCode</td><td>ATM, POS</td></tr> <tr><td>5</td><td>OCR</td><td>ATM, POS</td></tr> <tr><td>6</td><td>Chip</td><td>ATM, POS</td></tr> <tr><td>7</td><td>Key</td><td>ATM, POS, VTBI</td></tr> <tr><td>8</td><td>Contactless</td><td>ATM, POS</td></tr> <tr><td>9</td><td>ThreeDSWOAcqr</td><td>POS</td></tr> <tr><td>10</td><td>ThreeDSFullGrade</td><td>POS</td></tr> <tr><td>11</td><td>SingleTap</td><td>POS</td></tr> </tbody> </table>	Bit Number	Description	Allowed	NULL	None	All	1	Unknown	All	2	NoTerminal	All	3	MagStripe	ATM, POS	4	BarCode	ATM, POS	5	OCR	ATM, POS	6	Chip	ATM, POS	7	Key	ATM, POS, VTBI	8	Contactless	ATM, POS	9	ThreeDSWOAcqr	POS	10	ThreeDSFullGrade	POS	11	SingleTap	POS
Bit Number	Description	Allowed																																								
NULL	None	All																																								
1	Unknown	All																																								
2	NoTerminal	All																																								
3	MagStripe	ATM, POS																																								
4	BarCode	ATM, POS																																								
5	OCR	ATM, POS																																								
6	Chip	ATM, POS																																								
7	Key	ATM, POS, VTBI																																								
8	Contactless	ATM, POS																																								
9	ThreeDSWOAcqr	POS																																								
10	ThreeDSFullGrade	POS																																								
11	SingleTap	POS																																								
FIMI/CreateTerminal/Rq/ProcessingOptions	ArrRec	In	<p>{</p> <p>Int(2)* Network, Int(2)* Type, Str(100)* Value</p> <p>} [] – terminal options for processing transactions.</p> <p>It is used for the POS terminals only.</p>																																							
<p>Network. Available values:</p> <ul style="list-style-type: none"> <li>11 – VISA;</li> <li>22 – MasterCard.</li> </ul> <p>Type – option type. Available values:</p> <ul style="list-style-type: none"> <li>1 – Merchandise return processing.</li> <li>2 – Allowed preauthorization completion overdraft.</li> </ul> <p>Value – option value. Available variants depending on Type:</p>																																										

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<i>Type=1:</i> 0 – Authorization prohibited. 1 – Authorization allowed. 2 – Authorization is mandatory.			
<i>Type=2:</i> <i>It can take value -1 (unlimited) and 0...N (N – some large number, for example 999999999)</i>			
FIMI/CreateTerminal/Rq/ NormId	Int	In	ID of the norms for eliminating failures It is used only for the DDC ATMs.
FIMI/CreateTerminal/Rq/ Denomination	ArrRec	In	{ Str(2)* Denom; Num(15,2)* Amount; Int* Currency; } [] – data on standard initial amounts.  Denom – denomination; available values: 'A','B','C','D','E','F','G','H'. Amount – total amount on the denomination. Currency – denomination currency. It is used only for the DDC ATMs.
FIMI/CreateTerminal/Rq/ ExtPSNames	ArrRec	In	{ Int(2)* Network; Int(2)* FieldType; Str(100)* ExtValue; } [] – external terminal attributes for the online transfer to the external payment system:
Network – network for which external online attributes of a terminal are generated in Back Office (see the list of available values in section 3.23).  FieldType – terminal attribute being translated; available values: 1 – Name – terminal name. 2 – MCC. 3 – Owner – terminal owner name. 4 – Partner Id Code. 5 – Merchant Account Information for QR service. 6 – Region code/name – code or name of the terminal region.  ExtValue – value of the attribute being translated. It is meant for the online transfer to the payment system.			
FIMI/CreateTerminal/Rq/Accounts	ArrRec	In	{ Int(9) * Currency; Str(100)* Account; } [] – terminal accounts:  Currency – terminal currency (ISO code). Account – terminal account.
FIMI/CreateTerminal/Rp/TerminalId	Int*	Out	Created terminal ID
<b>UpdateTerminal – updating terminal information</b>			
FIMI/UpdateTerminal/Rq/ TerminalId	Int	In	Terminal ID in TWO.  It is mandatory if the BackOfficeId field is not transferred.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateTerminal/Rq/BackOfficeId	Int	In	Terminal ID in Back-Office. It can be transferred instead of the TerminalId field.
FIMI/UpdateTerminal/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution owning a terminal.  If the field is not defined (=NULL), the institution owning FIMI terminal stands for the terminal institution in TWO.  <i>For the list of available values, see the field InitSession/Rp/AllowedFIID.Name.</i>
FIMI/UpdateTerminal/Rq/Name	Str(16)	In	Terminal name
FIMI/UpdateTerminal/Rq>Title	Str(250)	In	Terminal title
FIMI/UpdateTerminal/Rq/Description	Str(750)	In	Terminal detailed description
FIMI/UpdateTerminal/Rq/SIC	Int	In	Terminal MCC
FIMI/UpdateTerminal/Rq/RetailerId	Int	In	Terminal retailer ID
FIMI/UpdateTerminal/Rq/CurrencyId	Int	In	Terminal Currency ID
FIMI/UpdateTerminal/Rq/CountryId	Int	In	Terminal country ID
FIMI/UpdateTerminal/Rq/Region	Str(90)	In	Terminal Region
FIMI/UpdateTerminal/Rq/RegionExternalId	Str(20)	In	Terminal region external ID
FIMI/UpdateTerminal/Rq/City	Str(90)	In	Terminal city
FIMI/UpdateTerminal/Rq/CityExternalId	Int	In	Terminal city external ID
FIMI/UpdateTerminal/Rq/Owner	Str(90)	In	Terminal owner
FIMI/UpdateTerminal/Rq/Location	Str(300)	In	Terminal location
FIMI/UpdateTerminal/Rq/ZIP	Str(90)	In	Terminal postal code
FIMI/UpdateTerminal/Rq/TimeOffset	Int	In	Terminal time offset (number of minutes)
FIMI/UpdateTerminal/Rq/Phone	Str(90)	In	Phone number
FIMI/UpdateTerminal/Rq/RemoveMissingFields	Int(1)	In	Indicates whether to delete the fields which are not sent. The fields are deleted if =1. The default value is 0
FIMI/UpdateTerminal/Rq/ChangeReason	Str(1000)	In	Information update reason
This operation can be applied only to the POS terminals. Terminal ID is not updated. If any optional field is not defined or =NULL, it is not updated. <i>RegionExternalId</i> and <i>CityExternalId</i> are filled if the <i>Region</i> and <i>City</i> fields are defined and the transaction is to be unloaded in the offline exchange file. <i>RegionExternalId</i> and <i>CityExternalId</i> are not saved in TWO, they are used only to transfer the identifiers in the file for transactions offline exchange between TWO and TWCMS. The <i>Name</i> , <i>CurrencyId</i> , <i>CountryId</i> and <i>TimeOffset</i> fields cannot be deleted. If <i>RemoveMissingFields=1</i> and these fields are not filled, the information will remain unchanged.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetTerminalAdditionalFields – request for terminal additional fields</b>			
FIMI/GetTerminalAdditionalFields/ Rq/TerminalId	Int	In	Terminal ID
FIMI/GetTerminalAdditionalFields/ Rq/TerminalName	Str(16)	In	Terminal name. It can be specified instead of the terminal ID.
FIMI/GetTerminalAdditionalFields/ Rp/Fields	ArrRec	Out	<pre>{   str(100)* Name;   str(4000) TextValue;   strXML XMLValue; } [] – array of structures,</pre> <p>List of the terminal additional fields:</p> <p>Name – name TextValue – text value XMLValue – XML value</p>
<b>SetTerminalAdditionalFields – setting terminal additional fields</b>			
FIMI/SetTerminalAdditionalFields/ Rq/TerminalId	Int	In	Terminal ID
FIMI/SetTerminalAdditionalFields/ Rq/TerminalName	Str(16)	In	Terminal name. It can be specified instead of the terminal ID
FIMI/SetTerminalAdditionalFields/ Rp/Fields	ArrRec	In	<pre>{   str(100)* Name;   str(4000) TextValue;   strXML XMLValue; } [] – array of structures,</pre> <p>List of the terminal additional fields:</p> <p>Name – name TextValue – text value XMLValue – XML value</p> <p>The total length of the <i>Name</i> and <i>TextValue</i> fields in all the records of the array <i>FIMI/SetTerminalAdditionalFields/Rq/UserFields</i> must not exceed 3900 symbols.</p>
FIMI/SetTerminalAdditionalFields/ Rq/RemoveMissingFields	Int	In	1 – remove missing fields 0 – by default

## 2.3.4.1.2 ATM Status:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetATMStatus – request for ATM state</b>			
FIMI/GetATMStatus/Rq/TermId	Int[]	In	ATM ID array. If it = NULL, states of all the ATMs belonging to the institution or those defined by the following parameters are returned. The following parameters manage the list of ATMs: <i>FIMI/GetATMStatus/Rq/StartTermId</i> and <i>FIMI/GetATMStatus/Rq/Count</i>
FIMI/GetATMStatus/Rq/ StartTermId	Int	In	ID of ATM after which starts the list of ATMs of size <i>FIMI/GetATMStatus/Rq/Count</i>
FIMI/GetATMStatus/Rq/Count	Int	In	Size of the requested list of ATMs, it should not exceed 1000
FIMI/GetATMStatus/Rq/ SortingColumn	Int(1)	In	The number indicating the column and column sorting type. If it = NULL, sorting by the TermId field in the ascending order applies. Sorting is effective within the range of the requested data only.  Available values: 1 – TermId (in the descending order) 2 – Connected (in the descending order) 3 – Connected (in the ascending order) 4 – Opened (in the descending order) 5 – Opened (in the ascending order) 6 – LastTranTime (in the descending order) 7 – LastTranTime (in the ascending order)
FIMI/GetATMStatus/Rq/ AskExhaustPrognosis	Int(1)	In	"Forecast of Notes Exhaustion Time by ATM Currencies Requested" attribute  If the field value is 1, the time of the notes exhaustion by ATM currencies is determined and result is returned in the <i>FIMI/GetATMStatus/Rp/ExhaustPrognosis</i> structure. If the field value is 0 (or NULL), the time is not determined and the respective structure is not filled.
FIMI/GetATMStatus/Rp/ ExhaustPrognosis	ArrRec	Out	{ int* TermId; int Currency; num Total; str ExhaustTime; num Available } [] – array of structures,  forecast of notes exhaustion time by ATM currencies:
Total – total end amount in the specified currency by all ATM hoppers.			
ExhaustTime – approximate time when the cash in the specified currency exhausts (in TWO time zone):			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<p>-1 – more than 10 days left till exhaustion.</p> <p>0 – no cash in the specified currency left in the ATM.</p> <p>NULL – cash exhaustion forecast is unknown (no statistics for the forecast).</p> <p>&gt;0 – cash exhaustion time.</p> <p>The ExhaustTime field returns the Int or Time value presented as a string.</p> <p>Available – amount in the specified currency available for dispensing from all ATM cassettes (it can be less than <i>Total</i>). It participates in <i>ExhaustTime</i> calculation.</p>			<pre>{     int* TermId;     int(1)* Connected;     int(1)* Opened;     int(12) LastTranId;     time LastTranTime;     int LastTranCode;     time LastDisconnectTime;     int(1)* CfgGroupType;     int(1)* RecyclingType;     int(1)* OverallSupplyStatus;     int(1)* OverallFitnessStatus;     int(1)* OverallSensorStatus     Int(1)* TR34BindingStatus; } [] – array of structures:</pre>

TermId – terminal ID

Connected – terminal attribute:  
0 – disconnected; 1 – connected.

Opened – terminal attribute:  
0 – closed; 1 – opened.

LastTranId, LastTranTime, LastTranCode – identifier, time and code of the last financial transaction.

LastDisconnectTime – time when the connection was broken.

CfgGroupType – type of ATM configuration group (or the protocol used by the ATM):  
1 – NDC, 2 – DDC, 3 - Wincor NDC, 4 – Wincor DDC.

RecyclingType – Cash Recycling function type:  
0 – None, 1 – NCR, 2 – Wincor, 3 – Diebold, 4 – Diebold NDC (NDC protocol simulation by the Diebold ATMs).

OverallSupplyStatus – level of supplies and cash as well as free space in the ATM containers:  
-1 – level is unknown (the devices are not configured), 0 – Normal, nomal level of supplies/cash or no conditions for containers overfilling, 1 – warning, low level of supplies/cash; 2 – error, the supplies/cash came to an end or the container is overfilled.

OverallFitnessStatus – ATM device fitness:  
1 – Level ii unknown as the devices are not configured.  
0 – Normal, all devices function correctly.  
1 – Warning, fault occurred in some devices, but the device continues functioning.  
2 – Error, fatal fault occurred in some devices, device cannot function.

OverallSensorStatus – state of sensors:  
0 – Normal, 3 – Alarm.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
TR34BindingStatus – status of binding the ATM to the host to perform RKL via the ASC X9 TR-34 protocol: -1 - Unknown 0 - Unbound 1 - Bound 2 - Bound to Other			
For the <i>OverallSupplyStatus</i> , <i>OverallFitnessStatus</i> and <i>OverallSensorStatus</i> fields, the status is calculated as maximum among all devices. For example, the <i>OverallSupplyStatus</i> =1 indicates that one or more devices have the status 1.			
FIMI/GetATMStatus/Rp/ Cassettes	ArrRec	Out	<pre>{ int* TermId; int* Number; int(1)* Type; int* Currency; str(240) CurrencyName; num* BillValue; num* EndCash; num* InReject; } [] – array of structures:</pre>
TermId – terminal ID Number – cassette number (for NDC, cassette type) Type – cassette type, 1 – only for dispense (notes), 2 – only for dispense (coins), 3 – Recycling cassette (notes) Currency – currency code CurrencyName – currency name BillValue – denomination EndCash – end amount InReject – retained amount			

#### 2.3.4.1.3 ATM Equipment (Hardware, Fitness, Media, Sensors):

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetATMHardware – request for ATM hardware configuration</b>			
FIMI/GetATMHardware/Rq/ID	Int*	In	ATM ID Notes: for the devices atypical for the ATM model, the response contains the field values = NULL
FIMI/GetATMHardware/Rp/Product	Int	Out	Terminal type (Diebold only): 1 – 911 model 2 – 912 mode (international) 3 – 912 mode (domestic)
FIMI/GetATMHardware/Rp/ TerminalModel	Int	Out	Terminal model (Diebold only)

<b>Parameter</b>	<b>Type</b> (* - mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetATMHardware/Rp/Disk	Int	Out	Hard disk installed (NDC only)
FIMI/GetATMHardware/Rp/CardReader	Int	Out	CardReader installed
FIMI/GetATMHardware/Rp/CashHandler	Int	Out	CashHandler is installed
FIMI/GetATMHardware/Rp/Depository	Int	Out	Depository installed
FIMI/GetATMHardware/Rp/ReceiptPrinter	Int	Out	Receipt printer installed
FIMI/GetATMHardware/Rp/JournalPrinter	Int	Out	Journal printer (e-journal) installed
FIMI/GetATMHardware/Rp/NightDepository	Int	Out	Night depository installed
FIMI/GetATMHardware/Rp/Encryptor	Int	Out	Encryptor (DES PIN Hardware for Diebold) installed
FIMI/GetATMHardware/Rp/Camera	Int	Out	Camera is installed
FIMI/GetATMHardware/Rp/DoorAccess	Int	Out	Door access device installed (NDC only)
FIMI/GetATMHardware/Rp/FlexDisk	Int	Out	Flexible disk is installed (NDC only)
FIMI/GetATMHardware/Rp/TamperBins	Int	Out	Tamper Indicating Bins installed (NDC only)
FIMI/GetATMHardware/Rp/CardholderKbd	Int	Out	Function Keys Configuration:  0 – 4 keys: right ABC; 1 – 4 keys: left ABCD 2 – 8 keys: right ABCD, left FGHI 3 – 12 keys: right ABCD, left FGHI, color keys JKLM
FIMI/GetATMHardware/Rp/OperatorKbd	Int	Out	Operator's keyboard installed (NDC only)
FIMI/GetATMHardware/Rp/CardholderDispVoice	Int	Out	Customer display installed
FIMI/GetATMHardware/Rp/StatementPrinter	Int	Out	Statement printer installed
FIMI/GetATMHardware/Rp/NDCplusRelease	Int	Out	NDC release (NDC only)
FIMI/GetATMHardware/Rp/NDCplusId	Str	Out	NDC identifier (NDC only)
FIMI/GetATMHardware/Rp/ConsumerDisplayGrid	Int	Out	Customer display grid (Diebold only): 0 – 32X16; 1 – 40X20
FIMI/GetATMHardware/Rp/ConsumerPrinterType	Int	Out	Receipt printer type (Diebold only): 0 – 40 column; 1 – 80 column

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMHardware/Rp/ConsumerPrinterColumnWidth	Int	Out	Column width of the receipt printer (Diebold only)
FIMI/GetATMHardware/Rp/CheckDepositor	Int	Out	Check Depositor installed (Diebold only)
FIMI/GetATMHardware/Rp/CoinDispenser	Int	Out	Coin Dispenser installed
FIMI/GetATMHardware/Rp/ColorGraphics	Int	Out	Color graphics feature configured (Diebold only)
FIMI/GetATMHardware/Rp/Track12CardWriter	Int	Out	Track1, Track2 card writer configured (Diebold only)
FIMI/GetATMHardware/Rp/ElectronicDataCapture	Int	Out	Electronic Data Capture configured (Diebold only)
FIMI/GetATMHardware/Rp/TouchScreen	Int	Out	Touch screen configured (Diebold only)
FIMI/GetATMHardware/Rp/LTIndicators	Int	Out	Lead-through indicators configured (Diebold only)
FIMI/GetATMHardware/Rp/EncryptedPINPad	Int	Out	Encrypted PIN Pad configured (Diebold only)
FIMI/GetATMHardware/Rp/WithdrawalDoor	Int	Out	Withdrawal door installed (Diebold only)
FIMI/GetATMHardware/Rp/Presenter	Int	Out	Presenter configured (Diebold only)
FIMI/GetATMHardware/Rp/ICCCComponentRelease	Int	Out	ICC Processing Core Application Component Release Number (NDC only)
FIMI/GetATMHardware/Rp/CAMComponentRelease	Int	Out	CAM ICC Application Component Release Number (NDC only)
FIMI/GetATMHardware/Rp/BNA	Int	Out	Cash-In unit
FIMI/GetATMHardware/Rp/EPPVendor	Int	Out	EPP (Encrypting PIN-Pad) vendor ID
FIMI/GetATMHardware/Rp/EPPRKLSupport	Int	Out	RKL method supported by EPP. Bit mask: - bit 1 – signature - bit 2 – certificate - bit 3 – extended signature (NDC only) - bit 4 – TR-34 (NDC only) If no bit is enabled or field is NULL, RKL is not supported.
FIMI/GetATMHardware/Rp/BarcodeReader	Int	Out	Barcode reader (NDC only)
FIMI/GetATMHardware/Rp/DualCashHandler	Int	Out	Attribute indicating that two cash dispense devices are present: CDM and GBRU (NDC only).

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMHardware/Rp/ChequeProcessor	Int	Out	Cheque acceptance system of the ATM (NDC only)
FIMI/GetATMHardware/Rp/ChequeBinCount	Int	Out	Number of cassettes for deposited cheques (NDC only)
FIMI/GetATMHardware/Rp/CashinBoxCount	Int	Out	Number of cassettes for deposited notes
FIMI/GetATMHardware/Rp/CoininBoxCount	Int	Out	Number of cassettes for deposited coins
FIMI/GetATMHardware/Rp/CardReader2	Int	Out	Contactless card reader (NDC only)
FIMI/GetATMHardware/Rp/EPPHostCertThumbprint	Str(128)	Out	Host certificate thumbprint to perform RKL via the ASC X9 TR-34 protocol, format – ASCII-HEX, thumbprint (hash) calculation algorithm – SHA-1. (NDC only).
FIMI/GetATMHardware/Rp/EPPTR31KBVersionSupport	Int(1)	Out	TR-31 Key Block versions supported by the ATM (NDC only): 0 – does not support 1 – VersionA 2 - VersionB 4 - VersionC
FIMI/GetATMHardware/Rp/EPPVarSNSupport	Int(1)	Out	Attribute indicating whether or not the ATM supports EPP with the serial number of the variable length (NDC only).
FIMI/GetATMHardware/Rp/EPPTR31KBPHeader	Str(250)	Out	Header of TR-31 Key Block Protection Key (TMK in the TR-31 Key Block format). If set, the key is loaded to the ATM and it is possible to load working keys in the TR-31 Key Block format (NDC only).
<b>GetATMFitness – request for ATM fitness</b>			
FIMI/GetATMFitness/Rq/Id	Int	In	ATM ID. If defined, the IdList field is not analyzed.  Notes: for the devices atypical for the ATM model, the response contains the field values = NULL
FIMI/GetATMFitness/Rq/IdList	Int[...]	In	List of the ATM identifiers. It is analyzed, if the ID field is not defined.
FIMI/GetATMFitness/Rq/Level	Int(1)	In	ATM state level starting from which the device states are requested (for the list of available values, see p. 3.5). E.g.:  Request of the states on the devices with failures and fatal errors - Level=1.  If Level=NULL, states of all the levels are requested

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMFitness/Rp/Clock	Int	Out	Clock (NDC)
FIMI/GetATMFitness/Rp/Communication	Int	Out	Communication (NDC)
FIMI/GetATMFitness/Rp/Disk	Int	Out	Hard disk (NDC)
FIMI/GetATMFitness/Rp/CardReader	Int	Out	Card reader The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/CashHandler	Int	Out	Cash handler The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/Depository	Int	Out	Depository The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/ReceiptPrinter	Int	Out	Receipt printer The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/JournalPrinter	Int	Out	Journal printer The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/NightDepository	Int	Out	Night Safe Depository The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/Encryptor	Int	Out	Encryptor (NDC)
FIMI/GetATMFitness/Rp/Camera	Int	Out	Camera (NDC)
FIMI/GetATMFitness/Rp/DoorAccess	Int	Out	Door access (NDC)
FIMI/GetATMFitness/Rp/FlexDisk	Int	Out	Flexible disk (NDC)
FIMI/GetATMFitness/Rp/Cassete1	Int	Out	Cassettes The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/Cassete2	Int	Out	-/-
FIMI/GetATMFitness/Rp/Cassete3	Int	Out	-/-
FIMI/GetATMFitness/Rp/Cassete4	Int	Out	-/-
FIMI/GetATMFitness/Rp/Cassete5	Int	Out	-/-
FIMI/GetATMFitness/Rp/Cassete6	Int	Out	-/-

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMFitness/Rp/Cassete7	Int	Out	-/-
FIMI/GetATMFitness/Rp/Cassete8	Int	Out	-/-
FIMI/GetATMFitness/Rp/Drum1	Int	Out	Drums (Recycling cassettes). The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/Drum2	Int	Out	-/-
FIMI/GetATMFitness/Rp/Drum3	Int	Out	-/-
FIMI/GetATMFitness/Rp/Drum4	Int	Out	-/-
FIMI/GetATMFitness/Rp/Drum5	Int	Out	-/-
FIMI/GetATMFitness/Rp/Drum6	Int	Out	-/-
FIMI/GetATMFitness/Rp/Drum7	Int	Out	-/-
FIMI/GetATMFitness/Rp/Drum8	Int	Out	-/-
FIMI/GetATMFitness/Rp/StmtPrinter	Int	Out	Statement printer The field must be filled in the response when requesting the information by the Id field
FIMI/GetATMFitness/Rp/EncryptedPINPad	Int	Out	PIN PAD (Diebold)
FIMI/GetATMFitness/Rp/TouchScreen	Int	Out	Touch screen (Diebold)
FIMI/GetATMFitness/Rp/LTIndicators	Int	Out	Lead Through Lamps (Diebold)
FIMI/GetATMFitness/Rp/CardWriter	Int	Out	Card writer (Diebold)
FIMI/GetATMFitness/Rp/VandalShield	Int	Out	Vandal shield (Diebold)
FIMI/GetATMFitness/Rp/Drawer	Int	Out	Withdrawal Door (Diebold)
FIMI/GetATMFitness/Rp/CoinDispenser	Int	Out	Coin dispenser (Diebold)
FIMI/GetATMFitness/Rp/BNA	Int	Out	BNA (Cash In)
FIMI/GetATMFitness/Rp/BNACassette1	Int	Out	BNA cassettes of the ATM. The field is filled in the response for NDC ATMs.
FIMI/GetATMFitness/Rp/BNACassette2	Int	Out	-/-
FIMI/GetATMFitness/Rp/BNACassette3	Int	Out	-/-
FIMI/GetATMFitness/Rp/BNACassette4	Int	Out	-/-
FIMI/GetATMFitness/Rp/BNACassette5	Int	Out	-/-

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMFitness/Rp/ BNACassette6	Int	Out	-/-
FIMI/GetATMFitness/Rp/ BNACassette7	Int	Out	-/-
FIMI/GetATMFitness/Rp/ CoinCassete1	Int	Out	Cassette for dispensing coins (NDC only)
FIMI/GetATMFitness/Rp/ CoinCassete2	Int	Out	-/-
FIMI/GetATMFitness/Rp/ CoinCassete3	Int	Out	-/-
FIMI/GetATMFitness/Rp/ CoinCassete4	Int	Out	-/-
FIMI/GetATMFitness/Rp/ EDC	Int	Out	Electronic journal
FIMI/GetATMFitness/Rp/ GBRU	Int	Out	Combined device for dispensing and accepting cash (Recycling unit) (NDC only)
FIMI/GetATMFitness/Rp/ BarcodeReader	Int	Out	Barcode reader (NDC only)
FIMI/GetATMFitness/Rp/ CDM	Int	Out	Additional device for dispensing notes only
FIMI/GetATMFitness/Rp/ ChequeProcessor	Int	Out	Cheque acceptance system of the ATM (NDC only)
FIMI/GetATMFitness/Rp/ EndorsePrinter	Int	Out	Endorsement printer (NDC only)
FIMI/GetATMFitness/Rp/ Stamper	Int	Out	Device for printing the predefined text in the receipt (NDC only)
FIMI/GetATMFitness/Rp/ ChequeEscRow	Int	Out	Temporary storage (NDC only) that keeps cheques deposited by a customer until the host posts a certain command to the ATM (return or retain).
FIMI/GetATMFitness/Rp/ CoinAcceptor	Int	Out	Coin acceptor
FIMI/GetATMFitness/Rp/ CardReader2	Int	Out	Contactless card reader (NDC only)
FIMI/GetATMFitness/Rp/ ChequeBin	ArrRec*	Out	<p>{          int* Id;          int* Number;          int Level;        } [] – list of states of the ATM cheque bins (NDC only):</p> <p>Id – terminal ID.          Number – bin number (from 1 to 15).          Level – bin level.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMFitness/Rp/FitnessList	ArrRec	Out	<pre>{     int* Id;     int Clock;     int Communication;     int Disk;     int CardReader;     int CashHandler;     int Depository;     int ReceiptPrinter;     int JournalPrinter;     int NightDepository;     int Encryptor;     int Camera;     int DoorAccess;     int FlexDisk;     int* Cassete1;     int* Cassete2;     int* Cassete3;     int* Cassete4;     int* StmtPrinter;     int EncryptedPINPad;     int TouchScreen;     int LTindicators;     int CardWriter;     int VandalShield;     int Drawer;     int CoinDispenser;     int BNA;     int Cassete5;     int Cassete6;     int Cassete7;     int Cassete8;     int Drum1;     int Drum2;     int Drum3;     int Drum4;     int Drum5;     int Drum6;     int Drum7;     int Drum8;     int BNACassette1;     int BNACassette2;     int BNACassette3;     int BNACassette4;     int BNACassette5;     int BNACassette6;     int BNACassette7;     int CoinCassete1;     int CoinCassete2;     int CoinCassete3;     int CoinCassete4;     int EDC;     int GBRU;     int CDM;     int ChequeProcessor;     int EndorsePrinter;     int Stamper;     int ChequeEscRow;     int CoinAcceptor;     int CardReader2;     int BarcodeReader; }</pre>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			} [] – list of ATM fitness. It is defined if the request contains the IdList field
<b>GetATMMedia – request for ATM media</b>			
FIMI/GetATMMedia/Rq/Id	Int*	In	ATM ID.  Notes: for the devices atypical for the ATM model, the response contains the field values = NULL
FIMI/GetATMMedia/Rq/Level	Int(1)	In	Level of the ATM media states starting from which the media states are requested (for the list of available values, see p. 3.6). E.g.:  Request of the media states on the devices with 'low', 'empty', 'overfill' - Level=2.  if Level=NULL, media states on all the devices are requested
FIMI/GetATMMedia/Rp/CardCaptureBin	Int	Out	Bin for the captured cards (NDC)
FIMI/GetATMMedia/Rp/CashRejectBin	Int	Out	Bin for the rejected cash (NDC)
FIMI/GetATMMedia/Rp/DepositBin	Int	Out	Deposit bin
FIMI/GetATMMedia/Rp/ReceiptPaper	Int	Out	Paper for the receipt printer
FIMI/GetATMMedia/Rp/JournalPaper	Int	Out	Paper for the journal printer
FIMI/GetATMMedia/Rp/NightSafe	Int	Out	Night depository safe (NDC)
FIMI/GetATMMedia/Rp/Cassete1	Int	Out	Cassette for coins
FIMI/GetATMMedia/Rp/Cassete2	Int	Out	-/-
FIMI/GetATMMedia/Rp/Cassete3	Int	Out	-/-
FIMI/GetATMMedia/Rp/Cassete4	Int	Out	-/-
FIMI/GetATMMedia/Rp/Cassete5	Int	Out	-/-
FIMI/GetATMMedia/Rp/Cassete6	Int	Out	-/-
FIMI/GetATMMedia/Rp/Cassete7	Int	Out	-/-
FIMI/GetATMMedia/Rp/Cassete8	Int	Out	-/-
FIMI/GetATMMedia/Rp/StatementPaper	Int	Out	Paper for the statement printer
FIMI/GetATMMedia/Rp/StatementRibbon	Int	Out	Ribbon for the statement printer (NDC)
FIMI/GetATMMedia/Rp/CameraFilm	Int	Out	Camera film (NDC)
FIMI/GetATMMedia/Rp/BNACashBin	Int	Out	BNA cash bin
FIMI/GetATMMedia/Rp/EDC	Int	Out	Electronic journal

<b>Parameter</b>	<b>Type</b> (* - mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetATMMedia/Rp/EndorsePrinter	Int	Out	Endorsement printer (NDC)
FIMI/GetATMMedia/Rp/Stamper	Int	Out	Device for printing the predefined text in the receipt (NDC)
FIMI/GetATMMedia/Rp/CoinAcceptor	Int	Out	Coin acceptor
FIMI/GetATMMedia/Rp/ChequeBin	ArrRec*	Out	{ int* Number; int State; } [] – list of states of ATM cheque bins: Number – bin number (from 1 to 15). State – supplies state (NDC).
FIMI/GetATMMedia/Rp/BNAcassettes	ArrRec*	Out	{ int* Number; int* State; int* Notes; } [] – list of ATM BNA-cassettes Number – BNA-cassette number (from 1 to 7). State – BNA-cassette state. Notes – number of notes in a cassette;
FIMI/GetATMMedia/Rp/CashInRetractBox	Int	Out	Cash In Retract cassette
FIMI/GetATMMedia/Rp/CoinRetractBox	Int	Out	Coin Retract cassette
FIMI/GetATMMedia/Rp/CoinInCassettes	ArrRec*	Out	{ int* Number; int* State; int* Coins; } [] – list of ATM Coin In cassettes Number – Coin In cassette number (from 1 to 7). State – Coin In cassette state. Coins – number of coins in a cassette.
FIMI/GetATMMedia/Rp/Drums	ArrRec*	Out	{ int* Number; int* State; } [] – list of the ATM drums Number – drum number (from 1 to 7). State – drum state.
<b>GetATMSensors – request for ATM sensors states</b>			
FIMI/GetATMSensors/Rq/Id	Int*	In	ATM ID

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMSensors/Rp/SupervisorMode	Int	Out	Flag 'Supervisor Mode' (1 or 0)
FIMI/GetATMSensors/Rp/Vibration	Int	Out	Flag 'Vibration/Heat Sensor Activated', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Door	Int	Out	Flag 'Door Sensor Activated', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/SilentSignal	Int	Out	Flag 'Silent Signal Activated', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/ElectronicEnclosure	Int	Out	Flag 'Electronic Enclosure Sensor Activated', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/DepositBin	Int	Out	Flag 'Deposit Bin Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/CardBin	Int	Out	Flag 'Card Bin Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/CurrencyRejectBin	Int	Out	Flag 'Currency Reject Bin Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete1	Int	Out	Flag 'Cassette 1 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete2	Int	Out	Flag 'Cassette 2 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete3	Int	Out	Flag 'Cassette 3 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete4	Int	Out	Flag 'Cassette 4 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete5	Int	Out	Flag 'Cassette 5 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete6	Int	Out	Flag 'Cassette 6 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete7	Int	Out	Flag 'Cassette 7 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Cassete8	Int	Out	Flag 'Cassette 8 Inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Drum1	Int	Out	Flag 'Drum (Recycling cassette) 1 inserted, NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Drum2	Int	Out	Flag 'Drum (Recycling cassette) 2 inserted, NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Drum3	Int	Out	Flag 'Drum (Recycling cassette) 3 inserted, NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Drum4	Int	Out	Flag 'Drum (Recycling cassette) 4 inserted, NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Drum5	Int	Out	Flag 'Drum (Recycling cassette) 5 inserted, NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Drum6	Int	Out	Flag 'Drum (Recycling cassette) 6 inserted, NDC, (1,0,NULL)

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMSensors/Rp/Drum7	Int	Out	Flag 'Drum (Recycling cassette) 7 inserted, NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/Drum8	Int	Out	Flag 'Drum (Recycling cassette) 8 inserted, NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/GBRUCashRejectBin	Int	Out	Flag 'Reject cassette GBRU inserted', NDC, (1,0,NULL)
FIMI/GetATMSensors/Rp/SupplyMode	Int	Out	Flag 'Supply Mode', Diebold, (1,0,NULL)
FIMI/GetATMSensors/Rp/AntiAmbush	Int	Out	Flag 'AntiAmbush Sensor Activated', Diebold, (1,0,NULL)
FIMI/GetATMSensors/Rp/Burglary	Int	Out	Flag 'Burglary', Diebold, (1,0,NULL)
FIMI/GetATMSensors/Rp/ChestDoor	Int	Out	Flag 'Chest Door', Diebold, (1,0,NULL)
FIMI/GetATMSensors/Rp/CoinDoor	Int	Out	Flag 'Coin Door', Diebold, (1,0,NULL)
FIMI/GetATMSensors/Rp/OtherDoor	Int	Out	Flag 'Other Doors', Diebold, (1,0,NULL)
FIMI/GetATMSensors/Rp/WithdrawalDoorTampering	Int	Out	Flag 'Withdrawal Door Tampering', Diebold, (1,0,NULL)
FIMI/GetATMSensors/Rp/Sensor	Int	Out	Flag 'Extra Sensor Activated', Diebold (1,0,NULL)
FIMI/GetATMSensors/Rp/CoinDispenser	Int	Out	Flag 'Coin Dispenser Detected' (1,0,NULL)
FIMI/GetATMSensors/Rp/CoinCassete1	Int	Out	Flag 'Coin Dispenser Cassette 1 inserted', NDC only, (1,0,NULL)
FIMI/GetATMSensors/Rp/CoinCassete2	Int	Out	Flag 'Coin Dispenser Cassette 2 inserted', NDC only, (1,0,NULL)
FIMI/GetATMSensors/Rp/CoinCassete3	Int	Out	Flag 'Coin Dispenser Cassette 3 inserted', NDC only, (1,0,NULL)
FIMI/GetATMSensors/Rp/CoinCassete4	Int	Out	Flag 'Coin Dispenser Cassette 4 inserted', NDC only, (1,0,NULL)
FIMI/GetATMSensors/Rp/DepositCassettes	ArrRec	Out	{ int* Number; int* State; } [] – list of the BNA cassette sensor states (NDC only): Number – cassette number (from 1 to 8). State – cassette state (1,0).
FIMI/GetATMSensors/Rp/DepositRetractCassettes	ArrRec	Out	{ int* Number; int* State; } [] – list of the BNA Retract cassette sensor states (NDC only): Number – cassette number (from 1 to 8). State – Retract cassette state (1,0).

## 2.3.4.1.4 ATM Hoppers:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetATMHoppers – request for ATM hoppers state</b>			
FIMI/GetATMHoppers/Rq/Id	Int*	In	ATM ID
FIMI/GetATMHoppers/Rq/CoinsHoppers	Int(1)	In	<p>Indicates that it is required to transfer data on hoppers with coins.</p> <p>If 1, the <i>List</i> or <i>Cassettes</i> array also returns hoppers with coins. If 0 or NULL, not data is transferred.</p> <p>The parameter does not affect filling of the <i>ExhaustPrognosis</i> array of structures – cassettes with coins are always considered</p>
FIMI/GetATMHoppers/Rq/Mode	Int(1)	In	<p>Request execution mode.</p> <p>If 0 or NULL, the <i>List</i> and <i>Drums</i> structure arrays are filled (the outdated mode, it is recommended to use Mode=1).</p> <p>If 1, the <i>Cassettes</i> structure array is filled</p>
FIMI/GetATMHoppers/Rp/List	ArrRec*	Out	<pre>{     int* Number;     int* Currency;     str(240) CurrencyName;     int(1)* ForCoins;     num* BillVal;     str(1) Denomination;     num* StdBegCash;     num* BegCash;     num* IncCash;     num* DecCash;     num* CashOut;     num* EndCash;     num* InReject;     int(1)* Blocked; }</pre> <p>} [] – array of structures, list of the ATM hoppers:</p> <p>Number – hopper number.      Currency – currency code.      CurrencyName – currency name.      ForCoins – indicator, for coins.      BillVal – denomination.      Denomination - denomination # .      StdBegCash – standard start amount.      BegCash – start cash.      IncCash – loaded amount.      DecCash – removed amount.      CashOut – dispensed amount.      EndCash – end cash.      InReject – rejected amount.      Blocked – blocking.</p>

Parameter	Type (*- mandatory parameter)	Dirac- tion (relative to TWO)	Description
FIMI/GetATMHoppers/Rp/Drums	ArrRec	Out	<pre>{     int* Number;     str(1) Denomination;     num* CashIn;     num* CashOut;     num* EndCash;     num* BillVal;     int* Currency;     str(240)* CurrencyName; }</pre> <p>} [] – array of structures, list of the ATM drums (Recycling cassettes):</p> <p>DrumNumber – drum number.      Number – hopper type.      Denomination – denomination #.      CashIn – deposited.      CashOut – dispensed amount.      EndCash – end cash.      BillVal – denomination.      Currency – currency code.      CurrencyName – currency name.</p>
FIMI/GetATMHoppers/Rp/Cassettes	ArrRec	Out	<pre>{     int* Number;     int PhysicalNumber;     int(1)* Type;     int(1)* Blocked;     int* Currency;     str(240) CurrencyName;     str(1)* Denomination;     num* BillVal;     num* StdBegCash;     num* BegCash;     num* IncCash;     num* DecCash;     num* CashOut;     num* EndCash;     num* InReject;     num CustDepositCash;     int* FitnessStatus;     num* ReservedCash;     num* AvailableCash; }</pre> <p>} [] – array of structures, list of ATM cassettes inserted during the last cash replenishment procedure:</p> <p>Number – cassette number (cassette type in the NDC notation).      PhysicalNumber – cassette physical number (it is filled for NDC only).      Type – cassette type, 1 – for dispense only (notes), 2 – for dispense only (coins), 3 – Recycling cassette (notes.)      Blocked – blocking.      Currency – currency code.      CurrencyName – currency name.      Denomination – denomination #.      BillVal – denomination.      StdBegCash – standard initial amount.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
BegCash – initial amount. IncCash – loaded amount. DecCash – removed amount. CashOut – dispensed amount. EndCash – end amount. InReject – rejected amount. CustDepositCash – amount deposited by customer (it is filled only for Recycling cassettes and included in IncCash). FitnessStatus – cassette fitness status (see the list of available values in section 3.5). ReservedCash – reserved amount. AvailableCash – cash available for dispense.			
FIMI/GetATMHoppers/Rp/ ExhaustPrognosis	ArrRec*	Out	<pre>{     int Currency;     num Total;     str ExhaustTime;     num Available }</pre> [] – array of structures, forecast of the bills exhaustion time by the ATM currencies:
Total – total end amount in the specified currency on all the ATM hoppers. ExhaustTime – approximate time when ATM cash in the specified currency will be exhausted (in TWO time zone): -1 – 10 days before the cash end. 0 – no cash in the ATM in the specified currency. NULL – the cash end forecast is unknown (no statistics). >0 – cash exhaust time. The ExhaustTime field returns the Int or Time value presented as a string. Available – amount in the specified currency available for dispensing from all ATM cassettes (it can be less than Total). It participates in ExhaustTime calculation. When filling this array of structures, cassettes with coins are always considered (even if FIMI/GetATMHoppers/Rq/CoinsHoppers = 0).			
FIMI/GetATMHoppers/Rp/Connected	Int(1)*	Out	Flag 'ATM Connected'
FIMI/GetATMHoppers/Rp/Opened	Int(1)*	Out	Flag 'ATM Opened'

#### 2.3.4.1.5 ATM Activity:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetATMActivity – request for ATM last activities</b>			
FIMI/GetATMActivity/Rq/Id	Int*	In	ATM ID
FIMI/GetATMActivity/Rp/ CardsRetained	Int*	Out	Number of the cards retained since the last balancing

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMActivity/Rp/BusinessDate	Time	Out	Current business day of the terminal
FIMI/GetATMActivity/Rp/BalancingDate	Time	Out	The last balancing time
FIMI/GetATMActivity/Rp/BalanceType	Int(1)*	Out	Balancing type: 0 – from ATM through the administrative card 1 – by operator 2 – automatically
FIMI/GetATMActivity/Rp/LastTranId	Int(12)	Out	ID of the last financial transaction
FIMI/GetATMActivity/Rp/LastTranTime	Time	Out	Time of the last financial transaction
FIMI/GetATMActivity/Rp/LastTranCode	Int	Out	Code of the last financial transaction (for the list of available values, see p. 3.7)
FIMI/GetATMActivity/Rp/DepositValue	ArrRec	Out	{ int* Currency; num* Amount; num* BillValue; } [] – information on the deposited amounts is defined for ATM with depository: Currency – deposit currency Amount – deposit amount BillValue – deposit denomination – it is defined only for the CashIn depository (with an option of bills automatic identification when depositing); it is 0 if the ATM doesn't identify the bills.

#### 2.3.4.1.6 Commands to ATM:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>ATMCommand – ATM driver commands</b>			
FIMI/ATMCommand/Rq/Id	Int	In	ATM ID
FIMI/ATMCommand/Rq/Name	Str(16)	In	ATM name. It can be specified instead of the ATM ID
FIMI/ATMCommand/Rq/Command	Int(3)*	In	Command code (for the available values, see p. 3.9)
FIMI/ATMCommand/Rq/HopperNumber	Int	In	Hopper number for the commands AdjustHopper, EnableCassete and DisableCassete

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ATMCommand/Rq/Sign	Int(1)	In	1 – add; 0 – subtract; (for the AdjustHopper command)
FIMI/ATMCommand/Rq/BillCount	Int	In	Number of bills for the AdjustHopper command

#### 2.3.4.1.7 ATM Holds

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetATMWithdrawalHold – request for information on ATM holds</b>			
FIMI/GetATMWithdrawalHold/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution of the ATM. If not defined (=NULL), the institution owning FIMI terminal stands for the ATM institution in <b>TWO</b> .
FIMI/GetATMWithdrawalHold/Rq/TermId	Int	In	ATM ID
FIMI/GetATMWithdrawalHold/Rq/TermName	Str(16)	In	ATM name. It can stand for TermId.
FIMI/GetATMWithdrawalHold/Rq/HoldId	Int	In	Hold ID
FIMI/GetATMWithdrawalHold/Rq/ArrayStatus	Int[...]	In	Array of the hold statuses. If set, the response returns the holds with the statuses specified in the array. If not transferred, the filtering of results by statuses is not performed.  Available values: 1 – waits for confirmation. 2 – confirmed. 11 – declined. 12 – purged. 21 – reserved cash is dispensed. 22 – canceled. 23 – expired.
FIMI/GetATMWithdrawalHold/Rq/FromTime	Time	In	Start time of the creation date.  If specified, the response returns the holds whose creation date exceeds or equals the specified one. If the field is not transferred, the filtering of results by the creation date is not performed.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetATMWithdrawalHold/Rq/ToTime	Time	In	End time of the creation date. If specified, the response returns the holds whose creation date is less than or equal to the specified one. If the field is not transferred, the filtering of results by the creation date is not performed.
FIMI/GetATMWithdrawalHold/Rq/Count	Int(4)	In	Maximum number of requested hold records (NULL – all records found by the set conditions) but not more than 1000.
At least one of the key search fields must be transferred to search for the holds: TermId, TermName, HoldId. Otherwise, the request will be declined.			
FIMI/GetATMWithdrawalHold/Rp/HoldInfo	ArrRec	Out	{ int* TermId; str(16)* TermName; int* HoldId; int(3)* Currency; num* Amount; int(3)* Status; int* ExpirationPeriod; time* CreateDate; time ConfirmDate; time CloseDate; } [] – information on the ATM holds:
<p>TermId – terminal ID.          TermName – terminal name.          HoldId – hold ID.          Currency – currency ISO code.          Amount – hold amount.          Status – hold status (see the list of available values in the FIMI/GetATMWithdrawalHold/Rq/ArrayStatus field).          ExpirationPeriod – hold validity period.          CreateDate – hold creation time.          ConfirmDate – time when the ATM driver confirmed the hold.          CloseDate – hold close time.</p>			
<b>CreateATMWithdrawalHold – creating ATM hold</b>			
FIMI/CreateATMWithdrawalHold/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution of the ATM. If not defined (=NULL), the institution owning FIMI terminal stands for the ATM institution in TWO.
FIMI/CreateATMWithdrawalHold/Rq/TermId	Int	In	ATM ID
FIMI/CreateATMWithdrawalHold/Rq/TermName	Str(16)	In	ATM name. It can stand for TermId.
FIMI/CreateATMWithdrawalHold/Rq/Currency	Int(3)*	In	Currency ISO code

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CreateATMWithdrawalHold/ Rq/Amount	Num*	In	Hold amount
FIMI/CreateATMWithdrawalHold/ Rq/ExpirationPeriod	Int*	In	Hold validity period (in minutes). Available values: 1<= ExpirationPeriod<=1440
FIMI/CreateATMWithdrawalHold/ Rp/HoldId	Int	Out	Hold ID
<b>ChangeStatusATMWithdrawalHold – changing ATM hold status</b>			
FIMI/ ChangeStatusATMWithdrawalHold /Rq/InstName	Str(4)	In	Character ID (name) of the financial institution of the ATM. If not defined (=NULL), the institution owning FIMI terminal stands for the ATM institution in <b>TWO</b> .
FIMI/ ChangeStatusATMWithdrawalHold /Rq/TermId	Int	In	ATM ID
FIMI/ ChangeStatusATMWithdrawalHold /Rq/TermName	Str(16)	In	ATM name. It can stand for TermId.
FIMI/ ChangeStatusATMWithdrawalHold /Rq/HoldId	Int*	In	Hold ID
FIMI/ ChangeStatusATMWithdrawalHold /Rq/NewStatus	Int(3)*	In	New hold status. Available values: 12 – purged 22 – canceled  Available status transitions: “waits for confirmation” -> “purged” “confirmed” -> “canceled”
FIMI/UpdateTerminal/Rq/ ChangeReason	Str(1000)	In	Status change reason

## 2.3.4.1.8 POS Terminal State:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>POSState – request for POS terminal state</b>			
FIMI/POSState/Rq/Id	Int*	In	POS terminal ID
FIMI/POSState/Rp/ShouldBeActive	Int(1)*	Out	State of the last request for activation/deactivation of the POS terminal: 0 – deactivation requested 1 – activation requested
FIMI/POSState/Rp/Deactivation	Int(1)*	Out	POS-terminal condition: 0 – activated 1 – deactivated 2 – should be deactivated
FIMI/POSState/Rp/NeedLoad	Int(1)*	Out	0 – configuration was updated from the host (loading is not needed) 1 – terminal needs to be loaded
FIMI/POSState/Rp/LastTranId	Int(12)	Out	Last transaction ID
FIMI/POSState/Rp/LastTranCode	Int	Out	Last transaction code (for the available values, see p. 3.8)
FIMI/POSState/Rp/LastRespCode	Int	Out	Authorization response for the last transaction (see p. 3.8)
FIMI/POSState/Rp/Clerk	Str(6)	Out	Clerk ID
FIMI/POSState/Rp/RespTime	Time	Out	Time of response to the terminal
FIMI/POSState/Rp/SeqNumber	Int*	Out	Transaction number in the current batch
FIMI/POSState/Rp/BatchNumber	Int*	Out	Batch number in the current shift
FIMI/POSState/Rp/ShiftNumber	Int*	Out	Shift number in the current day
FIMI/POSState/Rp/TransmitNumber	Int*	Out	Number of request from the terminal
FIMI/POSState/Rp/CardsRetained	Int*	Out	Number of the retained cards
FIMI/POSState/Rp/BatchTranCnt	Int*	Out	Number of the <i>close batch</i> transactions since the last Cutover
FIMI/POSState/Rp/ShiftTranCnt	Int*	Out	Number of the <i>close shift</i> transactions since the last Cutover
FIMI/POSState/Rp/BusinessDate	Time	Out	Terminal business day
FIMI/POSState/Rp/CutoverTime	Time	Out	Last Cutover time
FIMI/POSState/Rp/CutoverMode	Int(1)	Out	Last Cutover Mode: 1 – TerminalGood 2 – TerminalWithError 3 – ForcedWithoutClear 4 – ForcedWithClear
FIMI/POSState/Rp/MACErrorCount	Int*	Out	Number of the serial MAC errors
FIMI/POSState/Rp/HotListVersion	Int	Out	Loaded Hot List version

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSState/Rp/ HotListNewVersion	Int	Out	New Hot List version
FIMI/POSState/Rp/ HotListNeedReload	Int*	Out	Command 'Hot List Needs Reloading'
FIMI/POSState/Rp/HotListLoadTime	Time	Out	Time when the Hot List was last successfully loaded
FIMI/POSState/Rp/HotCardsInserted	Int*	Out	Number of cards added to the Hot List when downloading
FIMI/POSState/Rp/HotCardsDeleted	Int*	Out	Number of cards removed from the Hot List when downloading
FIMI/POSState/Rp/ HotCardTermCount	Int	Out	Number of 'hot' cards at the terminal
FIMI/POSState/Rp/ HotCardHostCount	Int	Out	Number of 'hot' cards at host
FIMI/POSState/Rp/TranTotals	ArrRec	Out	<pre>{     int(1)* TranKind;     num* BatchAmt;     int* BatchCnt;     num* ShiftAmt;     int* ShiftCnt;     num* DayAmt;     int* DayCnt; }</pre> <p>} [] – array of structures:  TranKind – transaction types:  1 – Debit; 2 – Credit; 3 – Adjust;  4 – DCDebit; 5 – DCCredit; 6 – DCAdjust.</p>

#### 2.3.4.1.9 Commands to POS Terminal:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>POSCommand – command to POS terminal</b>			
FIMI/POSCommand/Rq/Id	Int*	In	POS terminal ID. It is defined for all the commands to the exclusion of POSActivateFIID(203), POSDeactivateFIID(204), POSActivateRetl(205), POSDeactivateRetl(206),
FIMI/POSCommand/Rq/Name	Str(16)	In	POS terminal name. It can be specified instead of the POS terminal ID.
FIMI/POSCommand/Rq/Command	Int(3)*	In	Command code (for the list of available values, see p. 3.11)

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSCommand/Rq/Retailer	Int	In	Retailer ID for the commands: POSActivateRetl (205), POSDeactivateRetl (206)
FIMI/POSCommand/Rq/RetailerName	Str(150)	In	For the commands POSActivateRetl(205), POSDeactivateRetl(206). It can stand for the retailer ID (the field FIMI/POSCommand/Rq/Retailer) or can be used to search for the POS terminal by its name.
FIMI/POSCommand/Rq/InstName	Str(4)	In	Character identifier (name) of the financial institution owning the POS terminal
Only for the POS terminal commands: it is recommended to fill the Name, RetailerName and InstName fields when the POS terminal name is used (the Name field) in this operation instead of the POS terminal identifier. Otherwise, the terminal may not be found.			

#### 2.3.4.1.10 ATM Audit Journal:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetNDCAuditJournal – request for NDC e-journal</b>			
FIMI/GetNDCAuditJournal/Rq/TermId	Int*	In	ATM ID
FIMI/GetNDCAuditJournal/Rq/FromTime	Time	In	Start time and date when the message appeared in the audit journal
FIMI/GetNDCAuditJournal/Rq/ToTime	Time	In	End time and date when the message appeared in the audit journal
FIMI/GetNDCAuditJournal/Rp/Journal	Str(999999999)	Out	ATM journal
<b>GetDDCAuditJournal – request for DDC e-journal</b>			
FIMI/GetDDCAuditJournal/Rq/TermId	Int*	In	ATM ID
FIMI/GetDDCAuditJournal/Rq/FromTime	Time	In	System start date and time when the message appeared in the audit journal
FIMI/GetDDCAuditJournal/Rq/ToTime	Time	In	System end date and time when the message appeared in the audit journal
FIMI/GetDDCAuditJournal/Rq/FromTermTime	Time	In	Terminal start date and time when the message appeared in the audit journal
FIMI/GetDDCAuditJournal/Rq/ToTermTime	Time	In	Terminal end date and time of the message

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description												
FIMI/GetDDCAuditJournal/Rq/Type	Int(1)	In	<p>Message type. Available values:</p> <p>0 – Status 1 – Transaction 2 – Device Counter 3 – Device Log 4 – Diagnostic1 5 – Continuation 6 – DIT 7 – Diagnostic2</p> <p>If not defined, messages of any specified type will return.</p>												
FIMI/GetDDCAuditJournal/Rq/Count	Int(4)	In	Maximum number of the log requested records (NULL – all records found according to the set conditions) but not more than 1000.												
FIMI/GetDDCAuditJournal/Rq/StartRecordNumber	Int	In	<p>Sequence number when sorting the log records in the system date and time descending order.</p> <p>The first returned record has the number StartRecordNumber + 1</p> <p>If NULL, it = 0.</p>												
FIMI/GetDDCAuditJournal/Rq/Journal	ArrRec	Out	<pre>{     time* Time;     time* TermTime;     int(1)* Type;     str(240) Device;     str(2000) Message;     str(2000) MessComment'</pre> <p>} [] – array of structures:</p> <table> <tr> <td>Time</td> <td>– system date and time;</td> </tr> <tr> <td>TermTime</td> <td>– terminal date and time;</td> </tr> <tr> <td>Type</td> <td>– message type;</td> </tr> <tr> <td>Device</td> <td>– device type;</td> </tr> <tr> <td>Message</td> <td>– message;</td> </tr> <tr> <td>MessComment</td> <td>– comment on the message.</td> </tr> </table>	Time	– system date and time;	TermTime	– terminal date and time;	Type	– message type;	Device	– device type;	Message	– message;	MessComment	– comment on the message.
Time	– system date and time;														
TermTime	– terminal date and time;														
Type	– message type;														
Device	– device type;														
Message	– message;														
MessComment	– comment on the message.														

#### 2.3.4.2 On Events:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetATMEvents – request for ATM events</b>			
FIMI/GetATMEvents/Rq/TermId	Int	In	ATM ID
FIMI/GetATMEvents/Rq/TermName	Str(16)	In	Terminal Name
<p>Note: If the TermId and TermName are not defined, events on all the institution ATMs are returned</p>			

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetATMEvents/Rq/LastId	Int	In	ID of the last event accepted by the terminal; if defined in the request, all the events meeting the other conditions are returned in the response (with Id>LastId)
FIMI/GetATMEvents/Rq/Severity	Int(1)	In	Severity starting from which the events are requested (NULL-all): 1 – Event 2 – Warning 3 – Error 4 – Alarm
FIMI/GetATMEvents/Rq/FromTime	Time	In	Event start time
FIMI/GetATMEvents/Rq/ToTime	Time	In	Event end time
FIMI/GetATMEvents/Rq/Count	Int(4)	In	Number of the requested events (NULL-all found on the defined conditions), but not more than 1000
FIMI/GetATMEvents/Rq/EventCodes	ArrRec	In	{ Int Code } [] – list of the event codes (for the list of available values, see field <i>InitSession/Rp/EventCodes</i> )
FIMI/GetATMEvents/Rp/EventList	ArrRec	Out	{ Int(1)* Severity; time* RaiseTime; str(500) Mess; str(750) Cause; str(750) Remedy; str(750) Comment; int* EventId; str Source; int(12) TranId; str(16) TermName; } [] – list of the events sorted by the time descending order
<b><i>PutEvent – writing messages to Event log</i></b>			
FIMI/PutEvent/Rq/EventCode	Int*	In	Event code from the Dictionary of System Messages
FIMI/PutEvent/Rq/Word1	Str	In	First parameter
FIMI/PutEvent/Rq/Word2	Str	In	Second parameter
FIMI/PutEvent/Rq/Word3	Str	In	Third parameter
FIMI/PutEvent/Rq/Word4	Str	In	Fourth parameter
FIMI/PutEvent/Rq/Word5	Str	In	Fifth parameter
FIMI/PutEvent/Rq/Word6	Str	In	Sixth parameter
FIMI/PutEvent/Rq/TranId	Int(12)	In	Transaction ID

### 2.3.4.3 On Correspondent Accounts:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GetCorrespAcctInfo – request for information on institution correspondent account</b>			
FIMI/GetCorrespAcctInfo/Rp/ AgentList	ArrRec	Out	<pre>{   Str(4)* AgentInstName;   str(30)* Account;   str(150) Title;   num* Balance;   int* Currency;   str(3) CurrAlphaCode;   int* RateGroup;   str(256) LastRefreshFileName;   time LastRefreshFileCreationTime;   int LastRefreshFileSize;   time LastRefreshTime;   num LastRefreshBalance;   int LastRefreshPackNo;   int(12) LastFIMITranId;   time LastFIMITranTime;   num LastFIMIDeltaTranAmount;   num LastFIMIBalanceTranAmount;   int LastFIMIBalanceTranPackNo; } [] – list of the agent institution accounts (the accounts can be changed on the ModifyCorrespAcct request):</pre> <p>AgentInstName – name of the agent institution owning the account;    Account – correspondent account number;    Title – account description;    Balance – correspondent account balance;    Currency – correspondent account balance currency;    CurrAlphaCode – currency symbolic indication;    RateGroup – currency rate group for converting on the correspondent account;    LastRefreshFileName – Refresh file short name (without path);    LastRefreshFileCreationTime – OS date and time of the Refresh file creation on <b>TWO</b> host;    LastRefreshFileSize – Refresh file size in bytes, according to OS, on <b>TWO</b> host;    LastRefreshTime – date and time of the correspondent account update from the Refresh file, according to the <b>TWO</b> host time (in the <b>TWO</b> time zone);    LastRefreshBalance – correspondent account balance specified in the Refresh file (regardless of the Impact);    LastRefreshPackNo – Refresh file batch number specified in its header;    LastFIMITranId – ID of the last <i>ModifyCorrespAcct</i> FIMI transaction (correspondent account attributes modification);    LastFIMITranTime – date and time of the last transaction of correspondent account update, according to the <b>TWO</b> host time (in the <b>TWO</b> time zone);    LastFIMIDeltaTranAmount – value of the correspondent account balance change in the last <i>ModifyCorrespAcct</i> FIMI transaction;    LastFIMIBalanceTranAmount – correspondent account balance value specified in the last transaction setting the balance with regard to Impact;    LastFIMIBalanceTranPackNo – Impact batch number specified in the last <i>ModifyCorrespAcct</i> transaction setting the correspondent account balance with regard to Impact</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCorrespAcctInfo/Rp/SponsortList	ArrRec	Out	{           Str(4)* SponsorInstName;           str(30)* Account;           str(150) Title;           num* Balance;           int* Currency;           str(3) CurrAlphaCode;           int* RateGroup;           str(256) LastRefreshFileName;           time LastRefreshFileCreationTime;           int LastRefreshFileSize;           time LastRefreshTime;           num LastRefreshBalance;           int LastRefreshPackNo;           int(12) LastFIMITranId;           time LastFIMITranTime;           num LastFIMIDeltaTranAmount;           num LastFIMIBalanceTranAmount;           int LastFIMIBalanceTranPackNo;         } [] – list of the sponsor institution accounts: SponsorInstName – name of the sponsor institution owning the account

**ModifyCorrespAcct – changing balance of agent correspondent account**

FIMI/ModifyCorrespAcct/Rq/Account	Str(30)*	In	Number of the agent correspondent account
FIMI/ModifyCorrespAcct/Rq/Delta	Num	In	Balance increment (>0 or <0)
FIMI/ModifyCorrespAcct/Rq/Balance	Num	In	Absolute value of the correspondent account balance. It can be specified instead of Delta.  If Balance is specified, PackNo must be also defined
FIMI/ModifyCorrespAcct/Rq/PackNo	Int(9)	In	Package number used to consider the changes made by Impact

### 2.3.4.4 On Acquiring Limits:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b><i>GetCumulativeLimits – getting cumulative limits</i></b>			
FIMI/GetCumulativeLimits/Rq/ AcqObjType	Int(1)	In	Acquiring object type: NULL – any 1 – terminal 2 – host 3 – acquiring group 4 – retailer 5 – institution 6 – acquiring group template 7 – Retailers group
FIMI/GetCumulativeLimits/Rq/ AcqObjId	Int(9)	In	Acquiring object ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/ AcqObjId</i>
FIMI/GetCumulativeLimits/Rq/ AcqObjName	Str(150)	In	Acquiring object name. The field is used if the AcqObjType field is received and AcqObjId field is not transferred, it can have the following values: if AcqObjType=1, the field transfers the terminal name; if AcqObjType=4, the field transfers the retailer name; if AcqObjType=5, the field transfers the institution name; if AcqObjType=7, the field transfers the name of a group of retailers; if AcqObjType != (1, 4, 5, 7), the field value is ignored
FIMI/GetCumulativeLimits/Rq/ AcqObjInstName	Str(4)	In	Character ID (name) of the financial institution of the acquiring object. If NULL, the institution owning the terminal is used. The field is used if the AcqObjType, AcqObjName fields are received and AcqObjId field is not transferred, it can have the following values: if AcqObjType=1, the field transfers the terminal institution name; if AcqObjType=4, the field transfers the retailer institution name; if AcqObjType=7, the field transfers the name of the institution of a group of retailers; if AcqObjType != (1, 4, 7), the field value is ignored

Parameter	Type (*- mandatory parameter)	Dirac- tion (relative to TWO)	Description
FIMI/GetCumulativeLimits/Rq/ CumulativeLimits	ArrRec	Out	<pre>{ Int(1)* AcqObjType; Int(9)* AcqObjId; Int(1)* IssObjType; Int(9)* IssObjId; Int(9)* LimitId; Int(9)* BonusProgramId; Int(1)* CalcForTerminals; Int(9)* CALCScheme; Int(9)* Currency; Int(9)* RateGroup; Int(2)* PeriodType; Int(9) Period; Num* MaxValue; Num(3,2)* WarningPercent; Int(4)* WarningInterval; Num* CurValueCounter; Time LastResetTime; Time LastWarningTime; } [] – list of cumulative limits AcqObjType, AcqObjId, IssObjType, IssObjId, LimitId, BonusProgramId, CalcForTerminals, CALCScheme, Currency, RateGroup, PeriodType, Period, MaxValue, WarningPercent, CurValue – see the description in the <i>AddCumulativeLimit</i> request; CurValueCounter – current value of the limit counter; LastResetTime – time of the counter last reset; LastWarningTime – time of the last notification.</pre>
FIMI/GetCumulativeLimits/Rq/ TerminalCumulativeLimitCounters	ArrRec	Out	<pre>{ Int(1)* AcqObjType; Int(9)* AcqObjId; Int(1)* IssObjType; Int(9)* IssObjId; Int(9)* LimitId; Int(9)* BonusProgramId; Int(9)* TermId; Num* CurValueCounter; } [] – list of current values of counters on the cumulative limit for each terminal. Terminal counters on the cumulative limit are returned if FIMI/AddCumulativeLimit/Rq/ CumulativeLimits/CalcForTerminals=1. AcqObjType, AcqObjId, IssObjType, IssObjId, LimitId, BonusProgramId – see the description in the <i>AddCumulativeLimit</i> request; TermId – terminal ID, see the list of available values in the <i>GetTermList</i> request;</pre>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			CurValueCounter – current value of the terminal limit counter;
<b>AddCumulativeLimit – adding cumulative limit</b>			
FIMI/AddCumulativeLimit/Rq/ AcqObjType	Int(1)*	In	Acquiring object type: 1 – terminal 2 – host 3 – acquiring group 4 – retailer 5 – institution 6 – acquiring group template 7 – Retailers group
FIMI/AddCumulativeLimit/Rq/ AcqObjId	Int(9)	In	Acquiring object ID
It contains the acquiring object ID depending on the AcqObjType field: – if AcqObjType=1, the field transfers the terminal ID (see the list of available values in the <i>GetTermList</i> request) – if AcqObjType=2, the field transfers the host ID (the list of available values is transferred in field <i>FIMI/InitSession/Rp/Hosts</i> ) – if AcqObjType=3, the field transfers the acquiring group ID (the list of available values is transferred in field <i>FIMI/InitSession/Rp/AcquiringGroups</i> ) – if AcqObjType=4, the field transfers the retailer ID (see the list of available values in the <i>GetRetailerList</i> request) – if AcqObjType=5, the field transfers the institution ID (the list of available values is transferred in field <i>FIMI/InitSession/Rp/AllowedFID</i> ) – if AcqObjType=6, the field transfers the identifier of the acquiring group template (the list of available values is transferred in field <i>FIMI/InitSession/Rp/GroupTemplates</i> with Type=2) – if AcqObjType=7, the field transfers the identifier of the retailers group (the list of available values is transferred in field <i>FIMI/InitSession/Rp/RetailerGroups</i> )			
FIMI/AddCumulativeLimit/Rq/ AcqObjName	Str(150)	In	Acquiring object name. It can be used instead of the AcqObjId field if AcqObjType = (1, 4, 5, 7), The field is used if the AcqObjId field is not transferred, it can have the following values: if AcqObjType=1, the field transfers the terminal name; if AcqObjType=4, the field transfers the retailer name; if AcqObjType=5, the field transfers the institution name; if AcqObjType=7, the field transfers the name of a group of retailers.
FIMI/AddCumulativeLimit/Rq/ AcqObjInstName	Str(4)	In	Character ID (name) of the financial institution of the acquiring object. If NULL, the institution owning the terminal is used. The field is used if the AcqObjName field is received and AcqObjId field is not

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			transferred, it can have the following values: if AcqObjType=1, the field transfers the terminal institution name; if AcqObjType=4, the field transfers the retailer institution name; if AcqObjType=7, the field transfers the name of the institution of a group of retailers; if AcqObjType != (1, 4, 7), the field value is ignored.
FIMI/AddCumulativeLimit/Rq/ IssObjType	Int(1)*	In	Issuer object type: 1 – card profile 2 – card profile template
FIMI/AddCumulativeLimit/Rq/ IssObjId	Int(9)*	In	Issuer object ID It contains ID of the card profile or card profile template depending on the CardProfileType field: – if IssObjType=1, the field transfers the card profile ID (the list of available values is transferred in the <i>CardProfileList</i> field of the <i>InitSession</i> request). – if CardProfileType=2, the field transfers the card profile template ID (the list of available values is transferred in the field <i>FIMI/InitSession/Rp/GroupTemplates</i> with Type=1). -1 – any card profile/template
FIMI/AddCumulativeLimit/Rq/ LimitId	Int(9)*	In	Limit ID It contains the limit ID, see the list of available values in <i>FIMI/InitSession/Rp/Limits</i> for the card limits (the Type field with the value in the 1 <sup>st</sup> bit) except for the system limits with the identifiers 1, 201, 301.
FIMI/AddCumulativeLimit/Rq/ BonusProgramId	Int(9)*	In	Bonus program ID. It contains the bonus program ID, see the list of available values in <i>FIMI/InitSession/Rp/BonusPrograms</i> If -1, it is not defined.
FIMI/AddCumulativeLimit/Rq/ CalcForTerminals	Int(1)*	In	Indicates that the counters must be calculated by terminals. The field value can be 1, if AcqObjType=3, 4 or 6. If not, it is 0.
FIMI/AddCumulativeLimit/Rq/ CALCScheme	Int(9)*	In	CALC scheme. It contains the authorization scheme ID, see the list of available values in <i>FIMI/InitSession/Rp/AuthSchemes</i> with Type = 21
FIMI/AddCumulativeLimit/Rq/ Currency	Int(9)*	In	Currency. It contains the currency ID, see the list of available values in <i>FIMI/InitSession/Rp/CurrencyList</i>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AddCumulativeLimit/Rq/RateGroup	Int(9)*	In	Group of currency rates. It contains ID of the currency rates group, see the list of available values in <i>FIMI/GetRateGroups/Rp&gt;List</i>
FIMI/AddCumulativeLimit/Rq/PeriodType	Int(2)*	In	Period type
Available values:			
<ul style="list-style-type: none"> <li>-1 – as the scheme; the period is optional.</li> <li>-0 – Absolute (Daily) – absolute period; it starts from the beginning of the day (1440 – it is reset every day at 0:00, 2880+420 – at 7:00 every second day).</li> <li>-1 – Weekly – calendar weekly period; it starts from the beginning of the week (2880+420 – it is reset every Wednesday at 7:00).</li> <li>-2 – Monthly – calendar monthly period; it starts from the beginning of the month (2880+420 – it is reset on the third day of every month at 7:00).</li> <li>-3 – Quarterly – calendar quarterly period; it starts from the beginning of the quarter (2880+420 – it is reset on the third day of every quarter at 7:00).</li> <li>-4 – Infinite – limit counter will never be reset; the period is optional.</li> <li>-5 – Single operation – one-time limit; limit counter is always reset before the limit check; the period is optional.</li> <li>-7 – Yearly – calendar yearly period, it starts from the beginning of the year (Period=0 indicates that the counter is reset on the 1<sup>st</sup> of January at 00:00; Period=2880+420 – it is reset each year on the 3<sup>rd</sup> of January at 7:00).</li> <li>-8 – Sliding – sliding limit.</li> </ul>			
The limit real value is calculated during the transaction acquiring checks and considers all transactions executed over the time period starting from the moment in the past (it is the difference between the time current value and the specified period value) till the current moment.			
FIMI/AddCumulativeLimit/Rq/Period	Int(9)	In	<p>Limit reset time in minutes within the calendar period: Period/1440 – number of day (starts from 0); Mod(Period,1440) – reset time within a day (min)</p> <p>Restrictions:</p> <p>PeriodType=0 → 1440&lt;=Period&lt;=1440000 PeriodType=2 → 0&lt;=Period&lt;1440*31 PeriodType=7 → 0&lt;=Period&lt;1440*360 PeriodType=8 → 1&lt;=Period&lt;1440*7</p> <p>The value depends on the PeriodType field.</p>
FIMI/AddCumulativeLimit/Rq/MaxValue	Num*	In	Maximum limit value The value can exceed 0.
FIMI/AddCumulativeLimit/Rq/WarningPercent	Num(3,2)*	In	Threshold (in percents) of the notification generation
FIMI/AddCumulativeLimit/Rq/WarningInterval	Int(4)*	In	Notification generation interval (in minutes). The range of available values is from 0 to 1440

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b><i>UpdateCumulativeLimit – changing cumulative limit</i></b>			
FIMI/UpdateCumulativeLimit/Rq/AcqObjType	Int(1)*	In	Acquiring object type. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjType</i>
FIMI/UpdateCumulativeLimit/Rq/AcqObjId	Int(9)	In	Acquiring object ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjId</i>
FIMI/UpdateCumulativeLimit/Rq/AcqObjName	Str(150)	In	Acquiring object name. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjName</i>
FIMI/UpdateCumulativeLimit/Rq/AcqObjInstName	Str(4)	In	Character ID (name) of the financial institution of the acquiring object. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjInstName</i>
FIMI/UpdateCumulativeLimit/Rq/IssObjType	Int(1)*	In	Issuer object type. See the field description in <i>FIMI/AddCumulativeLimit/Rq/IssObjType</i>
FIMI/UpdateCumulativeLimit/Rq/IssObjId	Int(9)*	In	Issuer object ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/IssObjId</i>
FIMI/UpdateCumulativeLimit/Rq/LimitId	Int(9)*	In	Limit ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/LimitId</i>
FIMI/UpdateCumulativeLimit/Rq/BonusProgramId	Int(9)*	In	Bonus program ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/BonusProgramId</i>
FIMI/UpdateCumulativeLimit/Rq/CalcForTerminals	Int(1)	In	Indicates that counters must be calculated by terminals. If NULL, the value is not changed. The field value can be 1 if AcqObjType=3, 4 or 6; if not, it is 0.
FIMI/UpdateCumulativeLimit/Rq/WarningPercent	Num(3,2)	In	Threshold (in percents) of the notification generation. If NULL, the value is not changed.
FIMI/UpdateCumulativeLimit/Rq/WarningInterval	Int(4)	In	Notification generation interval (in minutes). The range of available values is from 0 to 1440 If NULL, the value is not changed.
FIMI/UpdateCumulativeLimit/Rq/MaxValue	Num	In	Maximum limit value. The value can exceed 0. If NULL, the value is not changed.
FIMI/UpdateCumulativeLimit/Rq/PeriodType	Int(2)	In	Period type. See the field description in <i>FIMI/AddCumulativeLimit/Rq/PeriodType</i>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/UpdateCumulativeLimit/Rq/Period	Int(9)	In	Limit reset time. See the field description in <i>FIMI/AddCumulativeLimit/Rq/Period</i>
<b>DeleteCumulativeLimit – deleting cumulative limit</b>			
FIMI/DeleteCumulativeLimit/Rq/AcqObjType	Int(1)*	In	Acquiring object type. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjType</i>
FIMI/DeleteCumulativeLimit/Rq/AcqObjId	Int(9)	In	Acquiring object ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjId</i>
FIMI/DeleteCumulativeLimit/Rq/AcqObjName	Str(150)	In	Acquiring object name. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjName</i>
FIMI/DeleteCumulativeLimit/Rq/AcqObjInstName	Str(4)	In	Character ID (name) of the financial institution of the acquiring object. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjInstName</i>
FIMI/DeleteCumulativeLimit/Rq/IssObjType	Int(1)*	In	Issuer object type. See the field description in <i>FIMI/AddCumulativeLimit/Rq/IssObjType</i>
FIMI/DeleteCumulativeLimit/Rq/IssObjId	Int(9)*	In	Issuer object ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/IssObjId</i>
FIMI/DeleteCumulativeLimit/Rq/LimitId	Int(9)*	In	Limit ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/LimitId</i>
FIMI/DeleteCumulativeLimit/Rq/BonusProgramId	Int(9)*	In	Bonus program ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/BonusProgramId</i>
<b>ResetCumulativeLimit – reset of current counters on cumulative limit</b>			
FIMI/ResetCumulativeLimit/Rq/AcqObjType	Int(1)*	In	Acquiring object type. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjType</i>
FIMI/ResetCumulativeLimit/Rq/AcqObjId	Int(9)	In	Acquiring object ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjId</i>
FIMI/ResetCumulativeLimit/Rq/AcqObjName	Str(150)	In	Acquiring object name. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjName</i>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ResetCumulativeLimit/Rq/AcqObjInstName	Str(4)	In	Character ID (name) of the financial institution of the acquiring object. See the field description in <i>FIMI/AddCumulativeLimit/Rq/AcqObjInstName</i>
FIMI/ResetCumulativeLimit/Rq/IssObjType	Int(1)*	In	Issuer object type. See the field description in <i>FIMI/AddCumulativeLimit/Rq/IssObjType</i>
FIMI/ResetCumulativeLimit/Rq/IssObjId	Int(9)*	In	Issuer object ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/IssObjId</i>
FIMI/ResetCumulativeLimit/Rq/LimitId	Int(9)*	In	Limit ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/LimitId</i>
FIMI/ResetCumulativeLimit/Rq/BonusProgramId	Int(9)*	In	Bonus program ID. See the field description in <i>FIMI/AddCumulativeLimit/Rq/BonusProgramId</i>
FIMI/ResetCumulativeLimit/Rq/TermId	Int(9)	In	Terminal ID. If NULL, the counter is reset on the limit as a whole. If not NULL, the counter of a certain terminal is reset within the limit. The terminal counter is reset if the limit is allowed to keep counters on the terminal ( <i>FIMI/GetCumulativeLimits/Rq/CumulativeLimits/CalcForTerminals = 1</i> ) See the list of available terminal IDs in <i>FIMI/GetCumulativeLimits/Rq/TerminalCumulativeLimitCounters/TermId</i>

#### 2.3.4.5 On QR Codes:

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>GenerateQRCode – generating merchant-presented QR code in the system</b>			
FIMI/GenerateQRCode/Rq/Serviceld	Int(1)*	In	Service ID. Available values: 1 - EMVCo QR 3 – NSPK FPS C2B 4 – NSPK FPS B2B
FIMI/GenerateQRCode/Rq/RetailerName	Str(150)*	In	Retailer name

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GenerateQRCode/Rq/ RetailerAccount	Str(150)	In	Retailer account. It is used for ServiceId=3, 4.
FIMI/GenerateQRCode/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution. If it is not filled, FIMI terminal institution will be used.
FIMI/GenerateQRCode/Rq/TermId	Int	In	POS terminal ID. It is used for ServiceId=1.
FIMI/GenerateQRCode/Rq/ TermName	Str(16)	In	POS terminal name. It can be specified instead of TermId for ServiceId=1.
FIMI/GenerateQRCode/Rq/ QRCodeType	Int(1)*	In	QR code type: 1 - QR-Static (QR sticker) 2 - QR-Dynamic (QR at the cashier desk) 3 - QR-CashBox (QR with the amount activation). 5 - QR-Subscription (account subscription QR). Types 3, 5 are allowed for ServiceId=3.
FIMI/GenerateQRCode/Rq/Amount	Num	In	Transaction amount. If NULL, it is 0 for ServiceId=3, 4.
FIMI/GenerateQRCode/Rq/TipType	Int(1)	In	Tip/fee type: 0 – not used. 1 – entered by customer in the application during payment. 2 – fixed amount in the TipValue field 3 – percent of the transaction amount in the TipValue field. If NULL, it is 0. Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/TipValue	Num	In	Value of tip/fee in percents or fixed amount, the type depends on the TipType field. If NULL, it is 0. Used for ServiceId=1 and TipType 1 or 2.
FIMI/GenerateQRCode/Rq/TaxType	Int(1)	In	VAT type: 0, NULL – not subject to VAT. 1 – subject to VAT, amount is determined by the merchant in TaxValue. 2 – subject to VAT, amount is determined by the payer. Used for ServiceId=4.
FIMI/GenerateQRCode/Rq/TaxValue	Num	In	VAT amount. Used for ServiceId=4 and TaxType =1.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GenerateQRCode/Rq/Currency	Int	In	ISO code of the transaction currency. If NULL, it is the terminal currency.
FIMI/GenerateQRCode/Rq/ExpirationDate	Time	In	Code expiration date. If NULL, it is unlimited.
FIMI/GenerateQRCode/Rq/PaymentPurpose	Str(140)	In	Payment purpose. For ServiceId=1(EMVCo QR ID 62.09) maximum field size is Str(25)
FIMI/GenerateQRCode/Rq/SubscriptionPurpose	Str(140)	In	Information from the merchant with the account subscription details. It is used for ServiceId=3, optionally for the QR-Dynamic type if the account subscription after the payment by QR code is required and mandatory for the QR-Subscription type.
FIMI/GenerateQRCode/Rq/MobileNumber	Str(25)	In	Phone number (EMVCo QR ID 62.02). Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/StoreLabel	Str(25)	In	Store label (EMVCo QR ID 62.03). Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/LoyaltyNumber	Str(25)	In	Loyalty card number (EMVCo QR ID 62.04). Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/ReferenceLabel	Str(25)	In	Operation label (EMVCo QR ID 62.05). Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/CustomerLabel	Str(25)	In	Customer label (EMVCo QR ID 62.06). Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/TerminalLabel	Str(25)	In	Terminal label (EMVCo QR ID 62.07). Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/QRLanguageld	Int	In	ISO code of the QR code language. It is used to fill in fields of the QR code in the alternative (national) language (EMVCo QR ID 64). See the list of available values in the <i>LanguageList</i> structure of the <i>InitSession</i> operation. Used for ServiceId=1.
FIMI/GenerateQRCode/Rq/ImageType	Int(1)	In	QR code image format: 0 – image is not required. 1 – PNG 2 – SVG. If NULL, it is 0.
FIMI/GenerateQRCode/Rq/ImageWidth	Int(4)	In	Image width in pixels. Minimum value is 200.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GenerateQRCode/Rq/ImageHeight	Int(4)	In	Image height in pixels. Minimum value is 200.
FIMI/GenerateQRCode/Rq/RedirectURL	Str(1024)	In	Link to return to the application or webstore. It is used for ServiceId=3, 4.
FIMI/GenerateQRCode/Rp/QRId	Str(32)*	Out	QR code ID
FIMI/GenerateQRCode/Rp/QRPayload	Str*	Out	Payload of the registered QR code in the Base64 encoding.
FIMI/GenerateQRCode/Rp/QRImage	Str	Out	QR code image. It is transferred in the Base64 format if the value of ImageType is 1 or 2.
<b>GetQRCodeInfo – getting information on QR code and merchant</b>			
FIMI/GetQRCodeInfo/Rq/ServiceId	Int(1)*	In	Service ID. Available values: 3 – NSPK FPS C2B 4 – NSPK FPS B2B
FIMI/GetQRCodeInfo/Rq/QRId	Str(32)*	In	QR code ID
FIMI/GetQRCodeInfo/Rp/QRCodeList	ArrRec	Out	{         str(32)* QRId;         int(1)* QRCodeType;         str(4) QRPayloadCRC;         num Amount;         int Currency;         str(140) PaymentPurpose;         str(140) RetailerTitle;         str(140) RetailerName;         str(140) RetailerAddress;         str(12) MemberId;         int MCC;         str(1024) RedirectURL;         str(12) AgentId;         str(140) SubscriptionPurpose;         str(12) TaxPayerNumber;         str(15) OGRN;         str(12) LegalEntityExtId;         int(1) TaxType;         num TaxValue;         str(35) UIP;       } [] – information on QR codes:

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<p>QRId – QR code ID.</p> <p>QRCodeType – QR code type, available values: 1 - QR-Static, 2 - QR-Dynamic, 5 - QR-Subscription.</p> <p>QRPayloadCRC – value of the payload checksum.</p> <p>Amount – transaction amount.</p> <p>Currency – ISO code of the transaction currency.</p> <p>PaymentPurpose – payment purpose.</p> <p>RetailerTitle – title of the merchant that issued the QR code.</p> <p>RetailerName – name of the merchant that issued the QR code.</p> <p>ReatailerAddress – address of the merchant that issued the QR code.</p> <p>MemberId – member ID within the service (list of available values is requested within the GetServiceMembersList operation).</p> <p>MCC – merchant category code.</p> <p>RedirectURL – link to return to the application or webstore.</p> <p>AgentId – identifier of the merchant agent that registered the payment link.</p> <p>TaxPayerNumber – tax payer number.</p> <p>OGRN – primary state registration number.</p> <p>LegalEntityExtId – external identifier of a corporate customer.</p> <p>TaxType – VAT type (the field is allowed for Serviceld=4), available values:</p> <ul style="list-style-type: none"> <li>• 0 – not subject to VAT.</li> <li>• 1 – subject to VAT, amount is determined in TaxValue.</li> <li>• 2 – subject to VAT, amount is determined by the payer.</li> </ul> <p>TaxValue – VAT amount (the field is allowed for Serviceld=4).</p> <p>UIP – unique payment identifier (the field is allowed for Serviceld=4).</p>			

#### **GetQRCodePayload – getting payload by QR code**

FIMI/GetQRCodePayload/Rq/ Serviceld	Int(1)*	In	Service ID. Available values: 1 - EMVCo QR 3 – NSPK FPS C2B 4 – NSPK FPS B2B
FIMI/GetQRCodePayload/Rq/ QRCodeType	Int(1)	In	QR code type: 1 - QR-Static (QR sticker) 2 - QR-Dynamic (QR at the cashier desk) 3 - QR-CashBox (QR with the amount activation) 5 - QR-Subscription (account subscription QR). Types 3, 5 are allowed for Serviceld=3. If NULL, it is any.
FIMI/GetQRCodePayload/Rq/QRId	Str(32)	In	QR code ID

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetQRCodePayload/Rq/ QRActivationId	Str(32)	In	QR code activation ID  It is used for QR with the amount activation (QRCodeType=3), it is mandatory if QRId is transferred and optional if RetailerName is not transferred instead of QRId.
FIMI/GetQRCodePayload/Rq/ RetailerName	Str(150)	In	Retailer name.  It is used if QRId =NULL.
FIMI/GetQRCodePayload/Rq/ InstName	Str(4)	In	Character ID (name) of the financial institution.  If it is not filled, FIMI terminal institution will be used.  It is used if the search for QR codes by retailer is performed (QRId =NULL).
FIMI/GetQRCodePayload/Rq/ NeedQRImage	Int(1)	In	Attribute indicating that the QR code image is required.  If NULL, it is 0. If QR code are requested by the retailer (QRId =NULL), it is 0.
FIMI/GetQRCodePayload/Rp/ QRPayloads	ArrRec	Out	{ str(32)* QRId; int(1)* QRCodeType; str* QR.Payload; time* CreateDate; num Amount; int Currency; str QRImage; str(32) QRActivationId;         } [] – information on QR codes:
QRId – QR code ID. QRCodeType – QR code type (see the list of available values in <i>FIMI/GetQRCodePayload/Rq/QRCodeType</i> ). QR.Payload – payload of the registered QR codes in the Base64 encoding. CreateDate – code creation date/time. Amount – transaction amount. Currency – ISO code of the transaction currency. QRImage – QR code image. It is transferred in the Base64 format if it was created when generating the QR code and the field NeedQRImage=1. QRActivationId – QR code activation ID.			
<b>CheckQRCodeStatus – checking status by QR code</b>			
FIMI/CheckQRCodeStatus/Rq/QRId	Str(32)*	In	QR code ID
FIMI/CheckQRCodeStatus/Rq/ QRActivationId	Str(32)	In	QR code activation ID  It is used and mandatory for QRCodeType=3 (QR with the amount activation).

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CheckQRCodeStatus/Rq/ ServiceId	Int(1)*	In	Service ID. Available values: 1 - EMVCo QR 3 – NSPK FPS C2B 4 – NSPK FPS B2B
FIMI/CheckQRCodeStatus/Rp/ QRStatus	Int(1)*	Out	QR code status: 0 – InProgress 1 – Accepted 2 – Rejected 3 – Error 4 – Purged 5 – Canceled 6 – Suspended
FIMI/CheckQRCodeStatus/Rp/ ExternalTranId	Str(32)	Out	External ID of the payment transaction. It can be returned in case of completing the process of payment by QR code for the Accepted, Rejected, Error statuses.
FIMI/CheckQRCodeStatus/Rp/ KZO	Str(128)	Out	Check value of the NSPK FPS operation. It can be returned in case of completing the process of the QR code payment for the Accepted statuses.
FIMI/CheckQRCodeStatus/Rp/ Currency	Int	Out	ISO code of currency of the transaction by QR code
FIMI/CheckQRCodeStatus/Rp/ Amount	Num	Out	Amount of transaction by QR code
FIMI/CheckQRCodeStatus/Rp/ QRPayload	Str*	Out	QR code of Payload in the Base64 encoding
FIMI/CheckQRCodeStatus/Rp/ QRPurchaseNetworkId	Str(100)	Out	ID of the payment network used for executing the payment. Available values: AMEX, VISA, UPI, MC, BANGLA, JCB, DISCOVER. It is returned for ServiceId = 1.
FIMI/CheckQRCodeStatus/Rp/ SubscriptionToken	Str(32)	Out	Subscription identifier created during the account subscription process. It is returned for the QR-Dynamic type (if the account subscription is performed after the payment) and QR-Subscription of the NSPK FPS service for SubscriptionStatus=1.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CheckQRCodeStatus/Rp/MemberId	Str(12)	Out	<p>Identifier of the service participant who performs the account subscription.</p> <p>It is returned for the QR-Dynamic type (if the account subscription is performed after the payment) and QR-Subscription of the NSPK FPS service.</p> <p><i>The list of available variants is requested within the GetServiceMembersList operation.</i></p>
FIMI/CheckQRCodeStatus/Rp/SubscriptionStatus	Int(1)	Out	<p>Subscription status, payer account subscription status</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>NULL – account was not subscribed or subscription status is unknown.</li> <li>0 – payer account subscription is declined.</li> <li>1 – payer account is subscribed.</li> <li>2 – not supported by the payer bank.</li> </ul> <p>It is allowed for the QR-Dynamic type (if the account subscription is performed after the payment by QR code) and QR-Subscription of the NSPK FPS service.</p>
FIMI/CheckQRCodeStatus/Rp/PaymentRequestStatus	Int(1)	Out	<p>Status of the request for payment by QR code using the subscription.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>NULL – request was not executed.</li> <li>0 – waiting for payer bank agreement.</li> <li>1 – payer bank agreed to perform the payment.</li> <li>2 – payer bank refusal, account subscription is not found.</li> <li>3 – payer bank refused to perform the payment.</li> <li>4 – failed to request the payer bank agreement to perform the payment.</li> </ul> <p>It is allowed for the QR-Dynamic type (if the request for payment by subscription is executed) of the NSPK FPS service.</p>
<b>ModifyQRCode – changing QR code status/availability</b>			
FIMI/ModifyQRCode/Rq/Serviceld	Int(1)*	In	<p>Service ID</p> <p>Available value:</p> <ul style="list-style-type: none"> <li>3 – NSPK FPS C2B</li> </ul>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ModifyQRCode/Rq/Action	Int(1)*	In	Action: 0 – deactivate 1 – activate 2 – resume 3 – suspend 4 – purge  Actions 0 and 1 are available for QR with the amount activation (QRCodeType=3). Actions 2, 3, 4 are available for the QR sticker (QRCodeType=1). Action 4 is available for QR at cash register (QRCodeType=2).
FIMI/ModifyQRCode/Rq/QRId	Str(32)*	In	QR code ID
FIMI/ModifyQRCode/Rq/QRActivationId	Str(32)	In	QR code activation ID  It is used and mandatory for deactivation (Action=0).
FIMI/ModifyQRCode/Rq/Amount	Num	In	Transaction amount  It is used and mandatory for activation (Action=1).
FIMI/ModifyQRCode/Rq/Currency	Int	In	ISO code of the transaction currency If NULL, it is the terminal currency.  The field is used for activation (Action=1).
FIMI/ModifyQRCode/Rq/ActivationExpirationDate	Time	In	Expiration date of the QR code activation  The field is used for activation (Action=1).
FIMI/ModifyQRCode/Rq/PaymentPurpose	Str(140)	In	Payment purpose  The field is used for activation (Action=1).
FIMI/ModifyQRCode/Rp/QRId	Str(32)*	Out	QR code ID
FIMI/ModifyQRCode/Rp/QRActivationId	Str(32)	Out	QR code activation ID  The field is returned during the activation (Action=1).
<b><i>MerchantRefund – refund generation by merchant</i></b>			
FIMI/MerchantRefund/Rq/Serviceld	Int(1)*	In	Service ID.  Available values: 1 - EMVCo QR 3 – NSPK FPS C2B
FIMI/MerchantRefund/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution.  If it is not filled, FIMI terminal institution will be used.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/MerchantRefund/Rq/ RetailerName	Str(150)*	In	Retailer name
FIMI/MerchantRefund/Rq/ Currency	Int	In	ISO code of the transaction currency. If NULL, the terminal currency is used.
FIMI/MerchantRefund/Rq/ Amount	Num*	In	Transaction amount
FIMI/MerchantRefund/Rq/ PaymentPurpose	Str(140)	In	Payment purpose
FIMI/MerchantRefund/Rq/ Alias	Str (500)	In	Alias of the refund recipient. It is used for Serviceld=3, mandatory parameter for the NSPK FPS service if QRId or QRPurchaseExternalTranId are not transferred. If one of the fields is transferred, the field can be transferred optionally, in this case the alias is determined basing on the original transaction executed with the QR code for which the refund is generated.
FIMI/MerchantRefund/Rq/ AliasMemberId	Str(12)	In	Member ID within the service specified in the Serviceld field. The refund is performed for this member. It is used for Serviceld=3. The list of available options is requested within the GetServiceMembersList operation. If NULL, the refund is performed for the originator of the transaction of purchase by QR code.
FIMI/MerchantRefund/Rq/QRId	Str(32)	In	QR code ID. It can be specified instead of QRPurchaseExternalTranId and PurchaseNetworkId to cancel the operation by the dynamic QR code.
FIMI/MerchantRefund/Rq/ QRPurchaseExternalTranId	Str(32)	In	External ID of the payment transaction (TID)
FIMI/MerchantRefund/Rq/ QRPurchaseNetworkId	Str(100)	In	ID of the payment network used for the payment. Available values: AMEX, VISA, UPI, MC, BANGLA, JCB, DISCOVER. The parameter is mandatory for Serviceld = 1. It is returned in the CheckQRCodeStatus operation response.
FIMI/MerchantRefund/Rq/ ThisTranId	Int(12)	In	ID of the current transaction that is repeatedly sent for authorization after contacting the operator. This field is specified: - If response code (FIMI/Response) 32-NeedClerkConfirmation was received for the transaction in advance and it is required to continue the transaction after contacting the operator.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description																																				
			<ul style="list-style-type: none"> <li>- If response code (FIMI/Response) 65-PartialAuthorizationExecuted was received for the transaction in advance to clarify the refund operation status and receive the AuthRespCode and DeclineReason fields.</li> </ul> <p>It is used for ServiceId=3.</p>																																				
FIMI/MerchantRefund/Rq/ MessageExtentions	ArrRec	In	<pre>{   str(100)* Name;   str(100)* Id;   str* Data; } [] – array of structures</pre> <p>List of extensions with data on the PS risks:</p>																																				
<p>Name – extension name. Available values: "FPSScoring" – for NSPK FPS. Id – extension ID. Available values: "FPSScoring" – for NSPK FPS. Data – extension value in the native format, it is transferred in the Base64 encoding: <b>for FPSScoring (NSPK FPS)</b> It contains indicators of the suspicious operation in the UAMP format with the separator PS(0x10): str(16) NSPK_FPS_107 – indicator of the suspicious operation of the sender bank. str(16) NSPK_FPS_108 – indicator of the suspicious operation of the recipient bank. str(16) NSPK_FPS_109 – indicator of the suspicious operation of NSPK OPCC. str(9) NSPK_FPS_125 – category of the sender account funds.</p>																																							
FIMI/MerchantRefund/Rq/ MerchantData	CustAcct	In	<p>Additional merchant data. It is used for ServiceId=3.</p>																																				
<p>The following fields can be transferred:</p> <table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th colspan="2">Description</th></tr> </thead> <tbody> <tr> <td>MerchantId</td><td>Str</td><td colspan="2">Merchant ID</td></tr> <tr> <td>LegalId</td><td>Str</td><td colspan="2">Legal entity ID</td></tr> <tr> <td>TaxPayerNumber</td><td>Str</td><td colspan="2">Tax payer number</td></tr> <tr> <td>Title</td><td>Str</td><td colspan="2">Legal entity title</td></tr> <tr> <td>Name</td><td>Str</td><td colspan="2">Merchant name</td></tr> <tr> <td>Account/IncomeCode</td><td>Str</td><td colspan="2">Code of the bank income type</td></tr> <tr> <td>Account/Number</td><td>Str</td><td colspan="2">Bank account number</td></tr> <tr> <td>Account/Category</td><td>Str</td><td colspan="2">Account funds category</td></tr> </tbody> </table> <p>It can be specified within the NSPK FPS service if the merchant data is not kept in TWO.</p>				Name	Type	Description		MerchantId	Str	Merchant ID		LegalId	Str	Legal entity ID		TaxPayerNumber	Str	Tax payer number		Title	Str	Legal entity title		Name	Str	Merchant name		Account/IncomeCode	Str	Code of the bank income type		Account/Number	Str	Bank account number		Account/Category	Str	Account funds category	
Name	Type	Description																																					
MerchantId	Str	Merchant ID																																					
LegalId	Str	Legal entity ID																																					
TaxPayerNumber	Str	Tax payer number																																					
Title	Str	Legal entity title																																					
Name	Str	Merchant name																																					
Account/IncomeCode	Str	Code of the bank income type																																					
Account/Number	Str	Bank account number																																					
Account/Category	Str	Account funds category																																					
FIMI/MerchantRefund/Rp/ AuthRespCode	Int	Out	<p>Authorizer response code – for the list of available values, see p. 3.9. It is returned for ServiceId=3.</p>																																				

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/MerchantRefund/Rp/ DeclineReason	Str	Out	Transaction decline reason – it is filled for the declined transaction. For standard checks, it is in English.  It is returned for ServiceId=3.
FIMI/MerchantRefund/Rp/ ToAcctDescr	Str(250)	Out	Destination account description
FIMI/MerchantRefund/Rp/ RefundExternalTranId	Str(32)	Out	External ID of the refund transaction (TID)
FIMI/MerchantRefund/Rp/FPSData	Str(8000)	Out	Additional data of the B2C transfer transaction. UAMP string with the parameters is transferred in the Base64 encoding (encoding of data prior to the Cp866 conversion), for the format and set of fields, refer to <i>TW International Container Fields Specification</i> , section 4.1.14.1 (FPSData).
FIMI/MerchantRefund/Rp/ PersonData	CustAcct	Out	Additional data of the refund recipient.  It is returned for ServiceId=3.

The following subfields can be transferred:

Name	Type	Description
Identity/Type	Int(1)	Type of the identity document: 1 – passport. 2 – driving licence. 3 – insurance number. 4 – TPN. 5 – national ID. 6 – international passport. 7 – military identity card. 8 – other document type.
Identity/Number	Str	Document number
Name/First	Str	First name
Name/Last	Str	Last name
Name/Middle	Str	Middle name
TaxPayerNumber	Str	TPN
PhoneNumber	Str	Phone number
Location/Address	Str	Address
Account/Number	Str	Bank account number
Account/Category	Str	Account funds category
PAM	Str	Personal Assurance Message(PAM)

#### **PullQRRequest – request for generation of payment by QR code**

FIMI/PullQRRequest/Rq/ ServiceId	Int(1)*	In	Service identifier.  Available values: 3 – NSPK FPS C2B
-------------------------------------	---------	----	--

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/PullQRRequest/Rq/InstName	Str(4)	In	Character ID (name) of the financial institution.  If it is not filled, FIMI terminal institution will be used.
FIMI/PullQRRequest/Rq/RetailerName	Str(150)*	In	Retailer name
FIMI/PullQRRequest/Rq/QRId	Str(32)*	In	QR code identifier
FIMI/PullQRRequest/Rq/SubscriptionToken	Str(32)*	In	Subscription identifier created during the account subscription procedure.  <i>It is returned in the response of the CheckQRCodeStatus operation in the SubscriptionToken field.</i>
FIMI/PullQRRequest/Rq/MemberId	Str(12)*	In	Identifier of the service participant who performs the account subscription  <i>It is returned in the response of the CheckQRCodeStatus operation in the MemberId field.</i>
<b>NotificationQRSubscription – notification on created subscription by QR code</b>			
FIMI/NotificationQRSubscription/Rq/Serviceld	Int(1)*	In	Service ID.  Available values: 3 – NSPK FPS C2B
FIMI/NotificationQRSubscription/Rq/QRId	Str(32)*	In	QR code ID
FIMI/NotificationQRSubscription/Rq/SubscriptionToken	Str(32)	In	Subscription ID generated in the course of linking the account.  The field is mandatory if Status=1 (subscription created successfully).
FIMI/NotificationQRSubscription/Rq/Status	Int(1)*	In	Subscription creation status: 0 – subscription creation failed. 1 – subscription is created successfully.

### 2.3.5 Requests on Transactions

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetTLADir – request for TLA dictionary of partitions</b>			
FIMI/GetTLADir/Rq/OnlyLastPartitions	Int(1)	In	<p>Indicates that only last partitions are requested.</p> <p>If = 1, the response returns the information only on partitions for the previous, current and next business day.</p> <p>If = NULL, it = 0</p>
<b>GetTransInfo – request for transaction information</b>			
FIMI/GetTransInfo/Rq/Id	Int(12)	In	Transaction ID
FIMI/GetTransInfo/Rq/TranNumber	Str(50)	In	<p>Transaction number assigned by initiator (or external payment system). The system searches for the transaction by TranNumber in TLG and TLA.</p>
FIMI/GetTransInfo/Rq/PartitionName	Str(40)	In	<p>Name of the partition in TLA where the search is performed; if not defined and BusinessDay field value is not defined, then:</p> <ul style="list-style-type: none"> <li>- if the FromTime field is defined, the search is performed starting from the specified time and the list of partitions id defined automatically;</li> <li>- if the FromTime field is not defined and the search is performed by the terminal (TermId(ArrayTermId)!=NULL or TermName!=NULL) or retailer (TermRetailerName!=NULL), the</li> </ul>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			search is performed in the current partition in TLA (in the current business day); - if the FromTime field is not defined and the search is performed by the card (PAN!=NULL), the search is performed through the WHOLE TLA table.
FIMI/GetTransInfo/Rq/Type	Int[]	In	Array of the requested transaction types (see p. 3.11). If it = NULL, all the transaction types are requested
FIMI/GetTransInfo/Rq/OrigType	Int(1)	In	Transaction originator type: NULL – all terminals (in case of the search by terminal) 0 – any 1 – own terminals (device drivers) 2 – external hosts (host interfaces) 8 – automatically 9 – manually
FIMI/GetTransInfo/Rq/PAN	Str(19)	In	Card number. It is possible to specify the card number as SQL statement 'like', that is, using the following metasymbols: '_' (any symbol), '%' (any number of any symbols). E.g., to request the cards whose number ends with '5678', the field must contain '%5678'
FIMI/GetTransInfo/Rq/MBR	Int(3)	In	Card member number NULL - any
FIMI/GetTransInfo/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetTransInfo/Rq/TermId	Int	In	Terminal ID
FIMI/GetTransInfo/Rq/TermName	Str(16)	In	Terminal name
FIMI/GetTransInfo/Rq/ArrayTermId	Int[]	In	Terminal IDs array. If defined, it overlays the values of the TermId, TermName fields
FIMI/GetTransInfo/Rq/TermRetailerName	Str(150)	In	Terminal retailer name
FIMI/GetTransInfo/Rq/FromTime	Time	In	Transaction start date and time
FIMI/GetTransInfo/Rq/ToTime	Time	In	Transaction end date and time
FIMI/GetTransInfo/Rq/AcqInstName	Str(4)	In	Name of the acquirer institution
FIMI/GetTransInfo/Rq/ArrayAcqInstName	Str[]	In	Array of the acquirer institution names. If defined, it overlays the AcqInstName field values.
FIMI/GetTransInfo/Rq/IssInstName	Str(4)	In	Name of the issuer institution

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description								
FIMI/GetTransInfo/Rq/ArrayInstName	Str[]	In	Array of the issuer institution names. If defined, it overlays the IssInstName field values.								
FIMI/GetTransInfo/Rq/TranCode	Int[]	In	Array of transaction codes to be returned in the response								
FIMI/GetTransInfo/Rq/AuthRespCode	Int[]	In	Array of the authorizer response codes for transactions to be returned in the response. <i>For the list of available values, refer to section 3.9.</i>								
FIMI/GetTransInfo/Rq/Count	Int(4)	In	Max number of the requested transactions (NULL- all found on the defined conditions) but not more than 1000								
FIMI/GetTransInfo/Rq/ResponseFields	Int(12)	In	<p>Bit mask of selecting the returned response. If NULL, the standard response is returned, matches value 6 (with the enabled bit 2 and 3).</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Bit Number</th><th>Returned Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>If the bit is enabled, the <i>TranList</i> structure of the last transaction returns only Id and RespCode, the <i>ATMCashValue</i> and <i>TranAcctBalances</i> structures are not returned.  The values of other field bits and <i>ArrayFieldTranList</i> structure are ignored.  The setting is considered if the search is performed by Id or TranNumber.</td></tr> <tr> <td>2</td><td>If the bit is enabled, the <i>ATMCashValue</i> structure is returned.</td></tr> <tr> <td>3</td><td>If the bit is enabled, the <i>TranAcctBalances</i> structure is returned.</td></tr> </tbody> </table>	Bit Number	Returned Response	1	If the bit is enabled, the <i>TranList</i> structure of the last transaction returns only Id and RespCode, the <i>ATMCashValue</i> and <i>TranAcctBalances</i> structures are not returned.  The values of other field bits and <i>ArrayFieldTranList</i> structure are ignored.  The setting is considered if the search is performed by Id or TranNumber.	2	If the bit is enabled, the <i>ATMCashValue</i> structure is returned.	3	If the bit is enabled, the <i>TranAcctBalances</i> structure is returned.
Bit Number	Returned Response										
1	If the bit is enabled, the <i>TranList</i> structure of the last transaction returns only Id and RespCode, the <i>ATMCashValue</i> and <i>TranAcctBalances</i> structures are not returned.  The values of other field bits and <i>ArrayFieldTranList</i> structure are ignored.  The setting is considered if the search is performed by Id or TranNumber.										
2	If the bit is enabled, the <i>ATMCashValue</i> structure is returned.										
3	If the bit is enabled, the <i>TranAcctBalances</i> structure is returned.										
FIMI/GetTransInfo/Rq/ArrayFieldTranList	Str(100)[]	In	Array of names of the <i>TranList</i> response structure fields for which data must be returned in the response. If the array is not transferred or does not contain the field names, all fields in the <i>TranList</i> structure are returned.								

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetTransInfo/Rq/BusinessDay	Time	In	Business day. It can stand for PartitionName. If the PartitionName parameter is set, the BusinessDay value is ignored. If the BusinessDay parameter is set, the search for the transaction is performed in all partitions of the set business day.
FIMI/GetTransInfo/Rq/NewerTran	Int(1)	In	Transactions selection type: 0 – when creating the list of transactions, older transactions have the higher priority if the maximum number of requested transactions is exceeded (the Count parameter); 1 – when creating the list of transactions, newer transactions have the higher priority if the maximum number of requested transactions is exceeded (the Count parameter).  If = NULL, it = 0. It is used if the FromTime parameter is set and Id and TranNumber parameters are not set.
FIMI/GetTransInfo/Rq/ExtRRN	Str(32)	In	External transaction RRN (Retrieval Reference Number – transaction number as it is sent via the host-interface: ISO 8583, field 37)
FIMI/GetTransInfo/Rp/MaskBalances	Int	Out	=1 — mask the ledger and available balance
FIMI/GetTransInfo/Rp/ATMCashValue	ArrRec	Out	<pre>{     Int(12)* Id;     num* Value;     num* Amount;     int* Count;     Int* Currency;     int* Direction;     int* Type; }</pre> <p>[]} – information on the dispensed/deposited bills/coins for the cash withdrawal/dispense transaction at ATM;</p> <p>Id – transaction ID (identical to the one in the <i>TranList</i> field);      Currency – currency;      Value –denomination;      Amount –amount;      Count – number of bills/coins;      Direction – direction:      1-dispense;      2-deposit;      4-cashback      8 – administrative dispense/deposit;</p>

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
			16 – counterfeit deposit 32 – suspicious deposit. Type – type (1-bills; 2-coins)
FIMI/GetTransInfo/Rp/TranAcctBalances	ArrRec	Out	<pre>{     Int(12)* Id;     str(30)* Account;     num AvailBalanceBefore;     num AvailBalanceAfter;     num LedgerBalanceBefore;     num LedgerBalanceAfter;     num DebitHoldBefore;     num DebitHoldAfter;     num CreditHoldBefore;     num CreditHoldAfter;     num BonusBefore;     num BonusAfter;     num OverdraftLimit;     int Currency; }</pre> <p>} [] – information on balances of accounts participating in the transaction:</p> <p>Id – transaction ID (the same as in the <i>TranList</i> field);      Account – account;      AvailBalanceBefore – account available balance before the transaction authorization;      AvailBalanceAfter – account available balance after the transaction authorization;      LedgerBalanceBefore – account ledger balance before the transaction authorization;      LedgerBalanceAfter – account ledger balance after the transaction authorization;      DebitHoldBefore – debit hold before the transaction authorization;      DebitHoldAfter – debit hold after the transaction authorization;      CreditHoldBefore – credit hold before the transaction authorization;      CreditHoldAfter – credit hold after the transaction authorization;      BonusBefore – bonus/debt before the transaction authorization;      BonusAfter – bonus/debt after the transaction authorization;      OverdraftLimit – account overdraft;      Currency – currency code.</p> <p>The structure is returned if the transaction was authorized by the accounts aggregate.</p>
FIMI/GetTransInfo/Rp/TranList	ArrRec	Out	<pre>{     int(12) Id;     int(4) Type;     time Time;     int(3) Phase;     int(3) TermClass;     int TermType;     str(16) TermName;     time TermDate;     int TranCode;     int(1) DraftCapture;     str(30) FromAcct;     str(30) ToAcct;     num Amount;     num Amount2;     num Fee;     num IssuerFee;     int Currency;     str(30) PAN;     int(3) CardMember;     int RespCode;     int(1) RetainCard;</pre>

Parameter	Type (*- mandatory parameter)	Direc- tion (relative to TWO)	Description
			str(8) ApprovalCode; num LedgerBalance; num AvailBalance; int(1) BalanceCurrencyAcct; int CurrencyAcct; int CurrencyAcctTo; num AmountAcct; num AmountAcctTo; num ExchangeRateAcct; num ExchangeRateAcctTo; int(12) RevRequestId; int Error; str(11) ExtAID; str(11) ExtFID; str(32) ExtRRN; int ExtSTAN; int(1)* OrigType; int CurrencyOrig; num AmountOrig; str(4) TermFName; str(11) TermlnslID; str(150) TermRetailerName; int TermSIC; str(240) TermSICName; int TermCountry; str(240) TermCountryName; str(90) TermCity; str(300) TermLocation; str(90) TermOwner; str(39) Track2; str(4) AuthFName; num RevAcualAmount; str(16) InvoiceNum; str(9) SeqNum; str(50) TranNumber; int POSCondition; int(3) POSEntryMode; int(1) PINOK; int(1) CVVOK; int(2) FromAcctType; int(2) ToAcctType; int CNSent; int IssuerCountry; str(250) IssuerCountryName; str(250) IssuerCardBrand; strUAMP(1000) ExtTranAttr; int Network; str(30) HostNetID; str(30) ToAcct2; num OverdraftLimit; num TmpOverdraft; str(4000) DeclineReason; int OrigId; str(2000) ChangeReason; int(12) NextTran; int(12) PrevTran; time OrigTime; int CashOutCycle; int CashInCycle; str(80) Track1; num DebitHold; num CreditHold; num Bonus; num LedgerBalanceBefore; num AvailBalanceBefore; num DebitHoldBefore; num CreditHoldBefore; num BonusBefore; int Reason; str(34) PAN2;

Parameter	Type (*- mandatory parameter)	Direc-tion (relative to TWO)	Description
			int(3) CardMemeber2; int(9) CVV2; str(64) CardUID; int(9) ICC_TermCaps; hex(5) ICC_TVR; str(8) ICC_TermSN; hex(32) ICC_IssuerData; str(10) ICC_AppTranCount; str(8) ICC_TermTranCount; int(9) ICC_AppProfile; int(1) ICC_IssuerScript1; int(1) ICC_IssuerScript2; int(9) ICC_TranType; int(9) ICC_TermType; int(9) ICC_CryptinformData; int(9) ICC_CVMRes; int(1) ICC_CAOK; str(10) ICC_PrevAppTranCount; hex(8) ICC_Cryptogram; hex(16) ICC_IAD; int(1) EMVSchemeType; int(9) IncSTAN; str(48) Clerk; int(1) PINEntry; str(4000) TextMess; int(9) Host; int(9) CNSId; time ICC_TranDate; num LedgerBalance2; num AvailBalance2; num LedgerBalance2Before; num AvailBalance2Before; strUAMP InstallmentData; int POSBatchNumber; int(1) ReversalAllowed; int(1) ReceiptFlag; strUAMP TokenData; int(1) IsContainCAVV; int TermlInstCountry; str(250) TermlInstCountryName; strUAMP MerchantData; strUAMP(4000) P2PSenderData; time TranTime; num RevActualAmountAcct;  } [] – transactions listed in the time descending order (see the fields description below):

Type – transaction type (see p.3.12).

Time – transaction date and time in **TWO** time zone.

Phase – transaction phase (21-Completed), 22-Purged).

TermClass – terminal class: 1=ATM, 2=POS, 3=CRT, 4=VTBI.

TermType – terminal type, see p.3.4.

TermName – terminal name (terminal character identifier; it is unique for the acquirer).

TermDate – transaction business day.

TranCode – transaction code, see p. 3.7.

Amount, Amount2 – transaction amounts in the Currency, where: Amount – transaction main amount used for all the transactions.

Amount2 – actual amount of the original transaction for the adjustment transactions.

Fee – amount of the acquiring fee in the transaction currency (the Currency field)

IssuerFee – amount of the issue fee in the account currency (AmountAcct,AmountAcctTo).

RespCode – authorization response code (for the list of available values, see p.3.9).

RevRequestId – for the transactions Request/AuthRequest/Advice/AuthAdvice, it is reference to the reversal transaction, for the reversal – reference to the original transaction.

Error – error by processing transactions not related to authorization (0 – no error, >0 – error); for the reversal, an error is of higher priority

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
than the authorization response (reversal with error != 0 is considered to be unsuccessfully processed).			
OrigType – transaction originator type (1-terminal, 2-external host, 8-automatically, 9-manually).			
OrigTime – transaction time in the originator time zone.			
OrigId – transaction initiator ID: for OrigType=1 – terminal ID; for OrigType=2 – host ID			
CurrencyOrig, AmountOrig – original currency and amount of the incoming transaction.			
RevActualAmount – actual amount of the original transaction (for reversal).			
InvoiceNum – POS terminal invoice number.			
SeqNum – request sequence number in the format DDDSSSBBB - Day(3), Shift(3), Batch(3).			
PINOK, CVVOK – PIN/CVV verification results – available values: 0-BAD, 1-OK, NULL- not checked.			
DraftCapture=CaptureMode (mode of receiving transaction from the terminal).			
0 – Not Capture transaction doesn't debit the account, but only holds the amounts – it is characteristic of the original transaction of voice authorization.			
1 – DraftCapture – transaction is the basis for posting an entry to the account			
2 – Slip – financial transaction of slip reconciliation and transaction of confirmation manually entered at POS terminal (Force-Post transaction); DraftCapture can be equal to 2, if Type=220 (Advice).			
3 – Fast Funds – the transaction <i>ATM/POS P2P Credit</i> corresponding to the operation <i>VISA Original Credit Enhanced Format (Fast Funds)</i> .			
4 – Verification Only – financial transaction of checking the destination card before executing the transfer.			
5 – G2C – financial transaction of the G2C transfer.			
6 – OTP – transaction with the OTP generation.			
7 – Verification Limit – generating the additional transaction for checking the limits.			
8 – Discontinuous – indicator of the discontinuous P2P transfer, it is used in the POS P2P Debit(132) transaction.			
FromAcctType, ToAcctType – type of the source and destination account: 0-Default, 1-Checking, 11-Savings, 31-Credit, 91 – Bonus.			
FromAcct, ToAcct – number of the card accounts used during the authorization: FromAcct – source account for all the POS/ATM-transactions to the exclusion of the ATM Deposit; ToAcct – destination account used for the payments, transfers and in the ATM Deposit transaction.			
ToAcct2 – payment personal information (personal account).			
LedgerBalance after the transaction authorization in the transaction or account currency.			
AvailBalance after the transaction authorization in the transaction or account currency.			
OverdraftLimit – permissible overdraft for the transaction in the transaction or account currency.			
TmpOverdraft – account temporary overdraft during the transaction authorization in the transaction or account currency.			
BalanceCurrencyAcct – flag: 1-balance in the account currency; 0-balance in the transaction currency.			
CNSent >0 – number of the customer notifications on the transaction, =NULL,0 – no notifications on the transaction.			
IssuerCountry, IssuerCountryName – code and name of the issuer country (can be defined for the 'external' cards only).			
IssuerCardBrand – name of the card program (for the 'external' cards only).			
ExtRRN – Retrieval Reference Number (transaction number as it is transferred via the host interface: ISO 8583, field №37).			
ExtSTAN – System Trace Audit Number (additional transaction ID in the external network: ISO 8583, field №11).			
ExtTranAttr – additional transaction attributes for the external network.			
Network – network whose rules are applied for the transaction execution (for the list of available values, refer to section 3.23).			
HostNetID – VISA host network ID: '0002' – VISA, '0004' – PLUS			
DeclineReason – transaction decline reason (for the standard checks - in English).			
NextTran, PrevTran – next and previous transaction linked to the current one.			
POSCondition – transaction condition – for the list of available values, see section 3.18.			
POSEntryMode – card entry mode – for the list of available values, see section 3.20.			
CashOutCycle – counter of the ATM replenishment cycles. Dispensed bills.			
CashInCycle – counter of the ATM replenishment cycles. Accepted bills.			
DebitHold – debit hold after the authorization in the account currency.			
CreditHold – credit hold after the authorization in the account currency.			
Bonus – bonus\debt after the authorization in the transaction or account currency.			
LedgerBalanceBefore – ledger balance prior to the transaction authorization in the transaction or account currency.			
AvailBalanceBefore – available balance prior to the transaction authorization in the transaction or account currency.			

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
DebitHoldBefore – debit hold prior to the authorization in the account currency.			
CreditHoldBefore – credit hold prior to the authorization in the account currency.			
BonusBefore – bonus\debt prior to the authorization in the transaction or account currency.			
Reason – transaction reason codes. For the list of available values, see p. 3.22.			
PAN2, CardMember2 – number of the Telebank customer card (), virtual card number, the destination card when performing transfer.			
CVV2 – virtual card CVV2.			
CardUID – unique card ID.			
ICC_TermCaps – terminal entry capability (bit mask).			
ICC_TVR – terminal verification results (bit mask).			
ICC_TermSN – serial number.			
ICC_IssuerData – issuer discretionary data.			
ICC_AppTranCount – application transaction counter (ATC).			
ICC_TermTranCount – terminal transactions counter.			
ICC_AppProfile – application interchange profile (bit mask).			
ICC_IssuerScript1 – flag indicating the presence of the issuer script 1.			
ICC_IssuerScript2 – flag indicating the presence of the issuer script 2.			
ICC_TranType – transaction type.			
ICC_TermType – terminal type.			
ICC_CryptInformData – Cryptogram Information Data.			
ICC_CVMRes – CVM results.			
ICC_CAOK – cryptogram verification result.			
ICC_PrevAppTranCount – value of the <i>Application Transaction Counter</i> in the previous online transaction.			
ICC_Cryptogram – application cryptogram.			
ICC_IAD – authorization response code and cryptogram.			
EMVSchemeType – EMV scheme type: 0 – VISA; 1 – MasterCard; 3 – Common Core Definitions (CCD); NULL – unknown (card is not chip).			
IncSTAN – STAN of incoming transaction.			
Clerk – terminal user name.			
PINEntry – PIN entry indicator: 0 – not entered, 1 – entered.			
Host – ID of authorizing host.			
CNSId – identifier of the SMS message generated for the transaction.			
ICC_TranDate – terminal transaction date.			
LedgerBalance2 – ledger balance after authorization in another currency (see the description of the <i>BalanceCurrencyAcct</i> field; if <i>BalanceCurrencyAcct</i> =1, the <i>LedgerBalance</i> field contains the ledger balance in the account currency (the <i>CurrencyAcct</i> field; if this field is NULL, then <i>CurrencyAcctTo</i> ), and the <i>LedgerBalance2</i> field contains the ledger balance in the transaction currency (the <i>Currency</i> field); if <i>BalanceCurrencyAcct</i> =0, then vice versa).			
AvailBalance2 – available balance after authorization in another currency.			
LedgerBalance2Before – ledger balance prior to authorization in another currency.			
AvailBalance2Before – available balance prior to authorization in another currency.			
InstallmentData – Installment data can be filled only for POS transactions. For details on the field format, refer to "TW International Container Fields Specification".			
POSBatchNumber – number of the POS batch the transaction belongs to (batch number in TWO).			
ReversalAllowed – indicates that the operator can perform the reversal for a certain transaction (it is sent only for operators not included in <i>Unlimited Access Group</i> ).			
ReceiptFlag – indicates that the receipt was requested for the particular transaction.			
TokenData – contains data on tokens in the UAMP format, for details on the field format, refer to "TW International Container Fields Specification".			
IsContainCAVV – indicates that CAVV/AAV/AEVV is present in the transaction.			
TermInstCountry, TermInstCountryName – code and name of the acquirer country.			
MerchantData – data on the real merchant in the UAMP format, for details on the field format, refer to "TW International Container Fields Specification".			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
P2PSenderData – information on the participants of the international P2P/C2C funds transfer transaction, for details on the field format, refer to "TW International Container Fields Specification".			
TranTime – transaction date and time considering the terminal offset.			
RevActualAmountAcct – real amount of a reversed transaction in the account currency.			
<b>GetFIMITrans – request for FIMI transactions on terminal</b>			
FIMI/GetFIMITrans/Rq/PartitionName	Str(40)	In	Partition name in TLA. If not defined, the current operative archive is used
FIMI/GetFIMITrans/Rq/TermId	Int	In	FIMI terminal ID
FIMI/GetFIMITrans/Rq/Clerk	Str(30)	In	FIMI terminal user name; if not defined, the terminal transactions conducted by all the users are returned
FIMI/GetFIMITrans/Rq/Account	Str(30)	In	Account number
FIMI/GetFIMITrans/Rq/AccountUID	Str(32)	In	Account unique identifier within a session. It can stand for the account number
FIMI/GetFIMITrans/Rq/PAN	Str(19)	In	Card number
FIMI/GetFIMITrans/Rq/MBR	Int(3)	In	
FIMI/GetFIMITrans/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetFIMITrans/Rq/TranCode	Int[]	In	FIMI transaction codes (see p. 3.7); if not defined, all the terminal transactions are returned

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description												
FIMI/GetFIMITrans/Rp/TranList	ArrRec	Out	<pre>{     int* TranCode;     time* Time;     str(30) Clerk;     str(300) RetAddress;     str(19) PAN;     int(3) CardMember;     str(30) Account;     num LedgerDelta;     num AvailDelta;     num OverdraftDelta;     int(12) AdditionalInfo;     int Command;     int(12)* TranId;     str(64) CardUID;     str(32) AccountUID; }</pre> <p>} [] – list of the FIMI transactions:</p> <p>Command – command ID (it is defined for the transactions ATMCommand, POSCommand)</p> <p>Interpretation of the AdditionalInfo field:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left; padding: 2px;">Transaction</th><th style="text-align: left; padding: 2px;">AdditionalInfo Value</th></tr> <tr> <td style="padding: 2px;">512 – Set Card Status</td><td style="padding: 2px;">Set card status</td></tr> <tr> <td style="padding: 2px;">522 - Set Account Status</td><td style="padding: 2px;">Set account status</td></tr> <tr> <td style="padding: 2px;">572 - Reverse Transaction</td><td style="padding: 2px;">ID of the reversed transaction</td></tr> <tr> <td style="padding: 2px;">547 - ATM Command</td><td style="padding: 2px;">ID of the ATM for which the command was issued</td></tr> <tr> <td style="padding: 2px;">552 - POS Command</td><td style="padding: 2px;">ID of the POS for which the command was issued</td></tr> </table> <p>For the <i>Set Card Status (512)</i> transaction, the <i>Account</i> field transfers the previous status of the card</p>	Transaction	AdditionalInfo Value	512 – Set Card Status	Set card status	522 - Set Account Status	Set account status	572 - Reverse Transaction	ID of the reversed transaction	547 - ATM Command	ID of the ATM for which the command was issued	552 - POS Command	ID of the POS for which the command was issued
Transaction	AdditionalInfo Value														
512 – Set Card Status	Set card status														
522 - Set Account Status	Set account status														
572 - Reverse Transaction	ID of the reversed transaction														
547 - ATM Command	ID of the ATM for which the command was issued														
552 - POS Command	ID of the POS for which the command was issued														
<b>ReverseTransaction – transaction reversal</b>															
FIMI/ReverseTransaction/Rq/Id	Int(12)	In	ID of the transaction being reversed												
FIMI/ReverseTransaction/Rq/RevActualAmount	Num	In	<p>Actual amount of the transaction being reversed with the acquiring fee.</p> <p>If it = NULL or 0, it is a full reversal</p> <p>The value that exceeds or is equal to 0 is allowed, if NULL or 0, it is a full reversal.</p>												
FIMI/ReverseTransaction/Rq/RevActualFee	Num	In	Actual amount of acquiring fee in the transaction being reversed – can be defined for the partial reversal												

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ReverseTransaction/Rq/TranNumber	Str(50)	In	ID of the transaction being reversed. The field is used for the POSRequest or AcctDebit/AcctCredit transaction reversal. The FIMI/ReverseTransaction/Rq/Id field is optional.
FIMI/ReverseTransaction/Rq/PAN	Str(19)	In	Card number. The field is mandatory for the POSRequest or AcctDebit/AcctCredit transaction reversal.  It is mandatory if the TranNumber field is specified.
FIMI/ReverseTransaction/Rq/MBR	Int(3)	In	Card member number. The field is mandatory for the POSRequest or AcctDebit/AcctCredit transaction reversal.  If NULL, it is 0. If -1, the MBR is not considered when searching for the original transaction.
FIMI/ReverseTransaction/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/ReverseTransaction/Rq/TermName	Str(16)	In	Name of POS terminal described in TWO.  It is used to search for a transaction by TranNumber. If not specified, it can be any.
FIMI/ReverseTransaction/Rq/Reason	Int	In	Reversal reason code. See section 3.22 "Transaction Reason Codes"   "Online Reversal".  If =NULL, the value 8 (Cancel) will be used
FIMI/ReverseTransaction/Rq/ExtPSFields	StrUAMP (1000)	In	Information on the fields of ISO message for the external payment system. For details on the field format, refer to "TW International Container Fields Specification".
FIMI/ReverseTransaction/Rq/SettleDate	Time	In	Settlement date
FIMI/ReverseTransaction/Rq/ReverseType	Int(3)	In	Type of processing the transaction reversal: 400 – reversal with authorization 420 - reversal without authorization. If NULL, then 420.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ReverseTransaction/Rq/ RefreshSeqNo	Int(12)	In	<p>Number of the last Refresh batch that contains the transaction of the account operation cancellation in Back-Office. If the field is specified, it is used for the impact on the account balances for a certain operation when performing the account Refresh.</p> <p>The field is specified if the online and offline exchange with Back-Office is present.</p>
FIMI/ReverseTransaction/Rq/ OrigTermName	Str(16)	In	<p>Name of the POS terminal that originated the reversal described in <b>TWO</b>.</p> <p>If not filled, the reversal is generated on behalf of the FIMI operator.</p>
FIMI/ReverseTransaction/Rq/ OrigTermInstName	Str(4)	In	<p>Name of the institution of the POS terminal that originated the reversal.</p> <p>This filled must be filled to explicitly identify the POS terminal if the terminal name (the TermName field) is not unique in <b>TWO</b>.</p> <p>If not filled, the FIMI terminal institution will be used.</p> <p>The field is used of the OrigTermName field is transferred.</p>
FIMI/ReverseTransaction/Rq/ OrigRetailerName	Str(150)	In	<p>Name of the retailer of the POS terminal that originated the reversal.</p> <p>This filled must be filled to explicitly identify the POS terminal if the terminal name (the TermName field) is not unique in <b>TWO</b>.</p> <p>This filed can be used only if the retailer is set for the POS terminal.</p> <p>The field is used of the OrigTermName field is transferred.</p>
FIMI/ReverseTransaction/Rp/ ThisTranId	Int(12)	Out	ID of the reversal transaction sent for authorization
FIMI/ReverseTransaction/Rp/ AuthRespCode	Int	Out	Authorizer response code. For the list of available values, refer to p. 3.9.
FIMI/ReverseTransaction/Rp/ AuthRespText	Str	Out	Text field in the free format. It can be filled in by the <b>Post-authorization</b> algorithmic function.
FIMI/ReverseTransaction/Rp/ ExtRespCode	Str(10)	Out	Original response of IPS or external system
FIMI/ReverseTransaction/Rp/ DeclineReason	Str	Out	Transaction declination reason. It is filled for the declined reversal. For standard checks, it is in English.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
The Request/AuthRequest/Advice/AuthAdvice/Admin (only for 528 - Account Debit, 529 – Account Credit) transaction can be reversed.			
Partial reversals are prohibited for the adjustment transactions (codes 121,122,123, 141), transfer transactions (codes 40, 351, 354) and for the ATM Deposit(20) transaction.			
ReverseTransaction allows cancellation of the approved transaction by the transaction identifier (Id) and transaction number (TranNumber).			
If the cancellation is performed by the transaction identifier, transfer the transaction identifier in the Id field, the value can be received in the <i>POSRequest</i> operation response (the FIMI/POSRequest/Rp/ThisTranId field) or GetTransInfo operation (FIMI/GetTransInfo/Rp/TranList.Id), it is not required to fill in the TranNumber, PAN, MBR, TermName request fields of the <i>ReverseTransaction</i> operation.			
If the transaction number was transferred for the transaction being cancelled during the generation (for example, the POSRequest/Rq/TranNumber field was transferred in the <i>POSRequest</i> operation), the operation cancellation can be performed by the transaction number generated by the terminal (originator). For this purpose, transfer TranNumber, PAN, MBR, and, optionally, TermName.			
To reverse the transaction by <i>TranNumber</i> , it is mandatory to define the card. If several transactions with the same <i>TranNumber</i> are found, the latest transaction is selected (that is the transactions with the greater identifier).			
The <i>AuthRespCode</i> , <i>AuthRespText</i> , <i>ExtRespCode</i> , <i>DeclineReason</i> fields are returned for reversals with authorization (ReverseType=400).			
<b>GetATMAdminTrans – request for information on ATM administrative transactions</b>			
FIMI/GetATMAdminTrans/Rq/TermId	Int	In	ID of the terminal that originated the transaction
FIMI/GetATMAdminTrans/Rq/TermName	Str(16)	In	Name of the terminal that initiated the transaction
FIMI/GetATMAdminTrans/Rq/PartitionName	Str(40)	In	Name of the partition in TLA where the transactions search has been performed; if not defined, the search in the TLA current partition is performed (in the current business day)
FIMI/GetATMAdminTrans/Rq/ReqTranCode	Int[]	In	Requested transaction codes, if it =NULL, all the terminal administrative transactions on the specified ATM are returned
FIMI/GetATMAdminTrans/Rp/TranList	ArrRec	Out	{ int(12)* Id; int* TranCode; int* AdminInitiator; str(19) PAN; time* Time; time* TermDate; int* CardsRetained; int* TermId; str(64) CardUID; } [] – transactions list

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
Id – administrative transaction ID. TranCode – transaction code. Available values: 91, 92, 93, 96, 97 – for the decryption, see p. 3.8; AdminInitiator – transaction was initiated: (0-from the terminal through the administrative card; 1-by the Front-Office operator; 2-automatically) PAN – number of the administrative card if the transaction was initiated from the terminal, otherwise = NULL; Time – transaction date and time; TermDate – terminal business day in TWO time zone; CardsRetained – number of the retained cards; Termld – terminal ID			
FIMI/GetATMAdminTrans/Rp/Hoppers	ArrRec	Out	<pre>{     int(12)* Id;     int* Numbr;     num* BegCash;     num* IncCash;     num* DecCash;     num* CashOut;     num* EndCash;     int Currency;     num BillValue; }</pre> } [] – hopper states:
Id – administrative transaction ID; Numbr – hopper number; BegCash – starting amount in the hopper (amount since the last balancing); IncCash – amount added by the cash messenger; DecCash – amount withdrawn by the cash messenger; CashOut – amount dispensed to the customers EndCash – amount remained in the hopper; Currency – hopper currency; BillValue – bill denomination.  All the amounts are defined for the moment of the administrative transaction, starting from the previous balancing.			
FIMI/GetATMAdminTrans/Rp/Drums	ArrRec	Out	<pre>{     int(12)* Id;     int* Numbr;     num* BegCash;     num* IncCash;     num* DecCash;     num* CashOut;     num* EndCash;     int Currency;     num BillValue; }</pre> } [] – drums (Recycling cassettes) state:
Id – administrative transaction ID Numbr – cassette number BegCash – starting amount in the cassette (amount since the last balancing) IncCash – amount added by the cash messenger DecCash – amount withdrawn by the cash messenger CashOut – amount dispensed to the customers EndCash – end amount Currency – cassette currency			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
BillValue – bill denomination  All the amounts are defined for the moment of the administrative transaction, starting from the previous balancing.			
FIMI/GetATMAdminTrans/Rp/ DepositHopper	ArrRec	Out	<pre>{     int(12)* Id;     int Currency;     num BillValue;     num Amount; }</pre> <p>{} – hopper states  Id – administrative transaction ID  Currency – currency  BillValue – denomination  Amount – amount</p> <p>All the amounts are defined for the moment of the administrative transaction, starting from the previous balancing</p>
FIMI/GetATMAdminTrans/Rp/DayTotals	ArrRec	Out	<pre>{     int(12)* Id;     int* TranCode;     int* Currency;     str(3) CurrAlphaCode;     num* Amount;     int* Count; }</pre> <p>{} – ATM statistics for the business day on the Withdrawal/Deposit transactions in the ATM currencies:</p> <p> Id – administrative transaction ID  TranCode – 10-Withdrawal; 20-Deposit  Currency – currency ISO code  CurrAlphaCode – 3-char currency code  Amount – amount of the Withdrawal/Deposit transactions in the specified currency for the business day  Count – number of the Withdrawal/Deposit transactions in the specified currency for the business day. </p>
<b>POSRequest – generating POS transactions</b>			
FIMI/POSRequest/Rq/KeyId	Int	In	ID of the working key obtained by means of the <i>GetWorkingKey</i> request. If not defined, PIN will be checked by the TPK key specified at FIMI terminal.
FIMI/POSRequest/Rq/TranType	Int(3)	In	<p>Transaction type:  100=AuthRequest (online-authorization);  120=AuthAdvice (receiving and approving a notification on the transaction);  200=Request (online authorization);  220=Advice (receiving and approving a notification on the transaction).  999=Admin (administrative request used only for the <i>CloseDay</i> (162) transaction).</p>

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/POSRequest/Rq/TranCode	Int(3)*	In	<p>Transaction code. Available values:</p> <ul style="list-style-type: none"> <li>- Purchase (110)</li> <li>- Pre-purchase(111)</li> <li>- Pre-purchase Complete (112)</li> <li>- Mail/Phone Order(113)</li> <li>- Merchandise Return(114)</li> <li>- Cash Advance(115)</li> <li>- Card Verification(116)</li> <li>- POS Balance Inquiry(117)</li> <li>- Purchase With Cashback(118)</li> <li>- Purchase Adjustment(121)</li> <li>- Merchandise Return Adjustment(122)</li> <li>- Cash Advance Adjustment(123)</li> <li>- Pre-purchase Increment(124)</li> <li>- Purchase Cancellation(125)</li> <li>- POS Message to Financial Institution(126)</li> <li>- Quasi-Cash(130)</li> <li>- POS P2P Debit(132)</li> <li>- POS P2P Credit(133)</li> <li>- POS P2P Calc Fee(134)</li> <li>- POS P2P Pass(135)</li> <li>- POS Installment Details(136)</li> <li>- POS PIN Change(139)</li> <li>- POS Deposit(140)</li> <li>- POS Deposit Adjustment(141)</li> <li>- POS Transfer Pass(149)</li> <li>- Close Batch(160)</li> <li>- Close Day(162)</li> <li>- POS Prepaid Pass(171)</li> <li>- POS Prepaid Debit(172)</li> <li>- POS Prepaid Credit(173)</li> <li>- POS Payment Pass(175)</li> <li>- POS Accumulate Bonus(181)</li> <li>- POS Redeem Bonus(182)</li> <li>- POS Cash Payment Pass(184)</li> <li>- POS Calc Fee(185)</li> <li>- POS Exchange Rate Inquiry(194)</li> <li>- PAR Inquiry(234)</li> <li>- C2C Pass(802)</li> <li>- C2B Pass(822)</li> <li>- C2C Pull Pass(835)</li> <li>- C2G Pass(860)</li> <li>- C2CI Pass(872)</li> <li>- B2B Pass(805)</li> <li>- POS ATC Update(827)</li> <li>- POS QR Purchase Debit(838)</li> <li>- POS QR Purchase Credit(839)</li> <li>- POS QR Purchase Pass(840)</li> <li>- POS ECPurchase (856)</li> <li>- POS ECRefund (857)</li> </ul> <p>(for the description of the above mentioned available operations, see p. 3.8, section <i>POS-transactions</i>)</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/TermName	Str(16)	In	Name of the POS terminal described in <b>TWO</b> .  If not filled, the POS terminal set by default for a certain FIMI terminal is used.
FIMI/POSRequest/Rq/TermInstName	Str(4)	In	Name of the POS terminal institution.  If the terminal name (the TermName field) is not unique in <b>TWO</b> , this field must be filled to determine the POS terminal definitely.  If not filled, the FIMI terminal institution will be used.
FIMI/POSRequest/Rq/RetailerName	Str(150)	In	Retailer name.  If the terminal name (the TermName field) is not unique in <b>TWO</b> , this field must be filled to determine the POS terminal definitely.  This field can be used only if the retailer is defined for the POS terminal.
FIMI/POSRequest/Rq/RetailerPassword	Str(20)	In	Retailer password.  If this field is filled, the password of the retailer to which the POS terminal (the TermName field) belongs is checked.
FIMI/POSRequest/Rq/PAN	Str(19)	In	Card number (or Telebank customer ID)  The fields are not defined only for cardless transactions (for the list of available transaction codes, see the description of the <i>FIMI/POSRequest/Rq/Track2</i> field). In all other cases the fields are mandatory.
FIMI/POSRequest/Rq/MBR	Int(3)	In	It is not required to fill in the MBR field if the "Calculate MBR Using Card Expiration Date" setting is enabled for the card prefixes. For the MBR calculation, the data from the Track2 field are used.  For Telebank customer ID, MBR=0.
FIMI/POSRequest/Rq/CardUID	Str(64)	In	Unique card identifier (or Unique ID of the Telebank customer). It can stand for the card number or Telebank customer ID (instead of the PAN and MBR fields)

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/Track2	Str(39)	In	Magnetic stripe
<p>The contents of the magnetic stripe in the following format:</p> <p>PAN=YYMM[SRV]... or PANDYYMM[SRV]...</p> <p>where:</p> <p>PAN – value transferred in the PAN field.</p> <p>YYMM – card expiration date (two characters – year, two characters – month).</p> <p>SRV – three-character value of the card service code.</p> <p>... – other information from the magnetic stripe.</p> <p>'PAN=YYMM' – mandatory part of the stripe.</p> <p>Other data of the magnetic stripe starting from the service code can be absent.</p> <p>The field is not defined only for cardless transactions. In all other cases it is mandatory. Cardless transactions can be generated for the following transactions:</p> <ul style="list-style-type: none"> <li>- POS P2P Credit(133)</li> <li>- POS Cash Payment Pass(184)</li> <li>- C2C Pass(802)</li> <li>- C2B Pass(822)</li> <li>- C2B Pull(847)</li> <li>- C2C Pull Pass(835)</li> <li>- C2G Pass(860)</li> <li>- C2CI Pass(872)</li> <li>- B2B Pass(805)</li> <li>- POS QR Purchase Debit(838)</li> <li>- POS QR Purchase Pass(840)</li> <li>- Pre-purchase(111)</li> <li>- POS P2P Debit(132)</li> <li>- POS P2P Pass(135)</li> <li>- POS Transfer Pass(149)</li> <li>- POS Payment Pass(175)</li> <li>- POS Exchange Rate Inquiry(194)</li> <li>- POS Calculation Fee (185)</li> <li>- POS P2P Calc Fee (134)</li> </ul> <p>In this case, the <i>FIMI/POSRequest/Rq/PAN</i> and <i>FIMI/POSRequest/Rq/CardUID</i> fields are not transferred in the request.</p> <p>If the CardUID field is filled (instead of the PAN and MBR fields), it is not required to send the PAN value in the field and the allowed contents of the magnetic stripe have the following format: =YYMM[SRV]...</p>			
FIMI/POSRequest/Rq/PAN2	Str(34)	In	Source/destination card number or account

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<i>Transfer destination card number</i>			
It can be specified for <i>POS P2P Debit</i> (132), <i>POS P2P Calc Fee</i> (134), <i>POS P2P Pass</i> (135), the field length does not exceed 19 characters. For <i>POS P2P Pass</i> (135) it is mandatory, it is allowed not to transfer it only if the <i>P2PAlias</i> field is used.			
<i>Card number or merchant account number</i>			
It can be specified for <i>POS QR Purchase Debit</i> (838), <i>POS QR Purchase Pass</i> (840), the field length does not exceed 19 characters.			
<i>Source card/account number</i>			
It can be specified for <i>POS P2P Credit</i> (133), <i>POS QR Purchase Credit</i> (839), the field length does not exceed 32 characters.			
<i>Waiter card number</i>			
It can be specified for <i>POS Purchase</i> (110) with tips (TIPAmount is transferred), the field length does not exceed 19 characters.			
FIMI/POSRequest/Rq/MBR2	Int(3)	In	<p>Number of the transfer destination cardholder.</p> <p>The field is specified for transactions described in the PAN2 field. If=NULL, it is 0.</p>
FIMI/POSRequest/Rq/CardUID2	Str(64)	In	Unique card identifier. It can stand for the card number (PAN2 and MBR2 fields).
FIMI/POSRequest/Rq/P2PSenderData	StrUAMP (4000)	In	<p>Information on the participants of the international P2P funds transfer transaction (132, 133, 135).</p> <p>Data on the C2C transfer sender within the M2M NSPK FPS service (if BusinessApplicationIdent is 'CC' or 'MP' in the transaction).</p> <p>Data on the C2B transfer sender within the NSPK FPS service.</p> <p>Data on the C2G transfer sender within the NSPK FPS service.</p> <p>Data on the originator of the QR code payment: <i>POS QR Purchase Debit</i> (838), <i>POS QR Purchase Credit</i> (839), <i>POS QR Purchase Pass</i> (840).</p> <p>For details on the field format, refer to <i>TW International Container Fields Specification</i>.</p> <p><b><i>The field is obsolete, it is recommended to use the P2PData structure to transfer the sender/recipient data.</i></b></p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/P2PData	ArrRec	In	<pre>{     str(1)* Type;     int(3) IssCardCountry;     str(100) ResidentCity;     str(50) ResidentState;     int(3) ResidentCountry;     str(30) PostalCode;     str(200) Address;     int(1) IdentificationType;     str(120) IdentityNumber;     int(3) IdentificationCountry;     time IdentificationExpDate;     str(250) FullName;     str(100) FirstName;     str(100) LastName;     str(100) MiddleName;     str(100) FirstNameNat;     str(100) LastNameNat;     str(100) MiddleNameNat;     str(20) TaxPayerNumber;     str(20) PhoneNumber;     str(300) Email;     int(1) Subject;     time DateOfBirth;     int(3) CountryOfBirth;     int(3) Nationality;     int(1) AccountNumberType;     str(12) TransferCode;     int(12) TransferAmount;     int(3) TransferCurrencyCode;     str(20) TaxId;     str(9) TaxReasonRegCode;     str(20) OperatorTaxId;     str(9) OperatorTaxReasonRegCode; }</pre> <p>} [] – array of structures,  Data on the P2P/C2C/C2B/C2G/QR transfer sender/recipient:</p> <p>Type – type of data: S – sender data, R – recipient data.  IssCardCountry – numeric ISO code of the card issuer country (available for sender only, Type=S).  ResidentCity – residence city (in Latin characters).  ResidentState – code of the residence state/province (in Latin characters).  ResidentCountry – numeric ISO code of the resident country.  PostalCode – postal code.  Address – address (in Latin characters).  IdentificationType – type of the identity document:  1 – passport.  2 – driving license.  3 – social insurance number.  4 – TPN.  5 – national identifier.  6 – international passport.  7 – military identity card.  8 – other document type.  IdentityNumber – identity document number.  IdentificationCountry – numeric ISO code of the country that issued the identity document.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<p>IdentificationExpDate – expiration date of the identity document.</p> <p>FullName – full name (in Latin characters).</p> <p>FirstName – first name (in Latin characters).</p> <p>LastName – last name (in Latin characters).</p> <p>MiddleName – middle name (in Latin characters).</p> <p>FirstNameNat – first name (available for sender only, Type=S).</p> <p>LastNameNat – last name (available for sender only, Type=S).</p> <p>MiddleNameNat – middle name (available for sender only, Type=S).</p> <p>TaxPayerNumber – TPN (available for sender only, Type=S).</p> <p>PhoneNumber – phone number.</p> <p>Email – E-mail address (available for sender only, Type=S).</p> <p>Subject – object: 0 – private customer, 1 – corporate customer.</p> <p>DateOfBirth – birth date.</p> <p>CountryOfBirth – numeric ISO code of the country of birth.</p> <p>Nationality – nationality, numeric country code.</p> <p>AccountNumberType – account type:</p> <ul style="list-style-type: none"> <li>0 – Other</li> <li>1 – RTN + Bank Account</li> <li>2 – IBAN</li> <li>3 – Card Account</li> <li>4 – Email</li> <li>5 – Phone Number</li> <li>6 – Bank account number (BAN) + Bank Identification Code (BIC)</li> <li>7 – Wallet ID</li> <li>8 – Social Network ID</li> </ul> <p>TransferCode – transfer destination code (available for sender only, Type=S).</p> <p>TransferAmount – transfer amount in the sender currency (currency before the conversion), the value must be in minimum currency units (available for sender only, Type=S).</p> <p>TransferCurrencyCode – numeric code of the sender currency (available for sender only, Type=S).</p> <p>TaxId – TPN of a corporate customer (available for recipient only, Type=R).</p> <p>TaxReasonRegCode – tax registration reason code of a corporate customer (available for recipient only, Type=R).</p> <p>OperatorTaxId – TPN of the digital wallet operator (available for recipient only, Type=R).</p> <p>OperatorTaxReasonRegCode – tax registration reason code of the digital wallet operator (available for recipient only, Type=R).</p> <p>It is used in the following transactions:</p> <ul style="list-style-type: none"> <li>• P2P transfer of funds (132, 133, 135)</li> <li>• C2C transfer (802) within the M2M NSPK FPS service (BusinessApplicationIdent ‘CC’ or ‘MP’)</li> <li>• International C2C transfer (872)</li> <li>• C2B transfer (822)</li> <li>• C2C Pull transfer (835)</li> <li>• B2B transfer (805)</li> <li>• QR code payments (838, 839, 840)</li> </ul>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/P2PAlias	Str (500)	In	<p>Alias of the destination card.</p> <p>It can be transferred in the <i>POS P2P Pass</i> (135) transaction instead of the <i>PAN2</i> field for P2P transfers executed within the VISA ADS and MC Transfer Hub services.</p> <p>It can be transferred in the <i>C2C Pass(802)</i>, <i>C2C Pull Pass(835)</i>, <i>C2CI Pass(872)</i> transaction instead of the <i>PAN2</i> field for C2C transfers executed within the NSPK FPS service.</p>
FIMI/POSRequest/Rq/ P2PAliasServiceId	Int(1)	In	<p>Identifier of the service of P2P transfers by card alias or C2B transfers by QR code:</p> <ul style="list-style-type: none"> <li>1 - VISA ADS</li> <li>2 - MC Transfer Hub</li> <li>3 – NSPK FPS</li> </ul>
FIMI/POSRequest/Rq/ P2PAliasMemberId	Str(12)	In	<p>Identifier of the service member within the service specified in the <i>P2PAliasServiceId</i> field.</p> <p>The list of available options is requested within the <i>GetServiceMembersList</i> operation.</p> <p>It is used in the NSPK FPS service and <i>C2C Pass(802)</i>, <i>C2C Pull Pass(835)</i>, <i>C2CI Pass(872)</i> transaction.</p>
FIMI/POSRequest/Rq/ P2PSourceOfFunds	Int(2)	In	<p>P2P transfer funds source:</p> <ul style="list-style-type: none"> <li>1 – VISA Credit</li> <li>2 – VISA Debit</li> <li>3 – VISA Prepaid</li> <li>4 – Cash</li> <li>5 – Other Debit</li> <li>6 – Other Credit</li> <li>7 – Prepaid</li> <li>8 – Deposit Account</li> <li>9 – Mobile Money Account</li> <li>10 – Tip</li> </ul> <p>The field is specified for POS P2P Credit (133), POS QR Purchase Credit (839). If it is not specified, then it is 4 - Cash.</p>
FIMI/POSRequest/Rq/ MessageToRecipient	Str (1000)	In	Message to the recipient of the P2P/C2C/QR transfer.
FIMI/POSRequest/Rq/ QRPayload	Str	In	<p>Payload of the QR code in the Base64 encoding.</p> <p>It is used within the NSPK FPS service – C2B Pass (822), B2B Pass (805) transactions, and EMV QR service – <i>POS QR Purchase Debit</i> (838), <i>POS QR Purchase Credit</i> (839), <i>POS QR Purchase Pass</i> (840) transactions.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/ QRPurchaseNetworkId	Str(100)	In	<p>ID of the payment network used for executing the payment. Available values: AMEX, VISA, UPI, MC, BANGLA, JCB, DISCOVER.</p> <p>It is used within the EMV QR service – <i>POS QR Purchase Debit (838)</i>, <i>POS QR Purchase Credit (839)</i>, <i>POS QR Purchase Pass (840)</i> transactions.</p>
FIMI/POSRequest/Rq/PINBlock	Hex(16)	In	<p>PIN block.</p> <p>PIN block is in the ANSI X9.8 (ISO Format 0) format and transferred by the encrypted working key (if the KeyId field is defined) or under TPK of the FIMI terminal.</p> <p>These fields are not used for cardless transactions (see the list of available transaction codes in the description of the <i>FIMI/POSRequest/Rq/Track2</i> field).</p>
FIMI/POSRequest/Rq/NewPINBlock	Hex(16)	In	<p>New PIN block.</p> <p>PIN block is in the ANSI X9.8 (ISO Format 0) format and transferred by the encrypted working key (if the KeyId field is defined) or under TPK of the FIMI terminal. The field must be transferred for the <i>POS PIN Change (139)</i> transaction.</p>
FIMI/POSRequest/Rq/FromAcctType	Int(2)	In	<p>Type of the source account. It is defined for all the transactions.</p> <p>Available values:</p> <p>0-Default, 1-Checking, 11-Savings, 31-Credit, 91-Bonus.</p> <p>The field is not defined for the POS Cash Payment Pass transaction only. In all the other cases the field is mandatory.</p>
FIMI/POSRequest/Rq/ToAcctType	Int(2)	In	<p>Type of the destination account. It is defined for all the transactions.</p> <p>Available values:</p> <p>0-Default, 1-Checking, 11-Savings, 31-Credit, 91-Bonus.</p> <p>If not defined, it = 0.</p>
FIMI/POSRequest/Rq/FromAccount	Str(30)	In	Source account. If defined, type of this account should match the one defined in the FromAcctType field or FromAcctType should be equal to 0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/ToAccount	Str(30)	In	Destination account for the transfer or vendor account for the payment. It should be defined for the transactions POS Payment Pass(175) and POS Cash Payment Pass(184). For the transactions, the ToAccount field should contain the vendor account number defined in TWO. It can be defined for the POS Transfer Pass(149) transaction. If defined for the transfer transaction, type of this account should match the one defined in the fields ToAcctType or ToAcctType and be equal to 0.
FIMI/POSRequest/Rq/Amount	Num	In	Transaction amount For the Purchase Cancellation(125) transaction for Pre-purchase, it is required to transfer the total amount including the Pre-purchase amount and all Pre-purchase increments.
FIMI/POSRequest/Rq/Amount2	Num	In	Transaction actual amount – it is defined for the adjustment transactions (codes: 121,122,123,141) only. For the transaction <i>PurchaseWithCashback(118)</i> – change amount in cash
FIMI/POSRequest/Rq/Fee	Num	In	Fee amount. If defined, it is considered that the transaction amount already includes the fee amount.
FIMI/POSRequest/Rq/Currency	Int	In	Transaction currency. If not defined, it is default POS terminal currency specified in TWO.
FIMI/POSRequest/Rq/Invoice	Str(16)	In	Invoice number
FIMI/POSRequest/Rq/OrigInvoice	Str(16)	In	Invoice number of the original transaction – it can be defined for the pre-purchase complete transaction (code 112) and for the adjustment transactions (code: 121, 122, 123, 141)
FIMI/POSRequest/Rq/ApprovalCode	Str(8)	In	Approval code. If defined, it will be the same in response. It is defined for the pre-purchase complete transaction (code 112). This field is mandatory for the 112 <i>Pre-Purchase Complete</i> and 124 <i>Pre-Purchase Increment</i> Standalone transactions – it must contain the information from the field of the original 111 <i>Pre-Purchase</i> Standalone transaction.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/CVV2	Str(4)	In	CVV2
If CVV2 is negative value, it is treated as follows: if CVV2=-99, it is replaced with 00; in other cases, the sign '-' is dismissed. See the results in the following table:			
<b>Position 1</b> 0 – CVV2 Value is intentionally not provided 2 – CVV2 Value is Present but illegible 9 – Cardholder states that no CVV2 value is imprinted			
<b>Position 2</b> 0 – Only the normal Response Code should be returned 1 – Both the Response Code and the CVV2 Result Code should be returned			
FIMI/POSRequest/Rq/Condition	Int	In	Transaction condition. For the list of available values, see Section 3.19.  This field is mandatory for all transactions, except for the <i>CloseDay</i> (162) transaction
FIMI/POSRequest/Rq/EntryMode	Int	In	Card entry mode. For the list of available values, see Section 3.20.  This field is mandatory for all transactions, except for the <i>CloseDay</i> (162) transaction
FIMI/POSRequest/Rq/DraftCapture	Int(1)	In	Mode of receiving transaction from the terminal
Available values: <ul style="list-style-type: none"> <li>- 0 – Not Capture – financial approval (slip) is required; it is characteristic of the original transaction of voice authorization and Pre-Purchase transaction (111).</li> <li>- 1 – Draft Capture – transaction is the basis for posting an entry to the retailer account.</li> <li>- 2 – Slip – financial representation that must be processed; financial transaction of slip reconciliation (in this case transaction is of the Advice type) or transaction of Pre-purchase Complete (112).</li> <li>- 4 – Verification Only – generation of the additional transaction to check the destination card before the transfer execution.</li> <li>- 8 – Discontinuous – indicates the discontinuous P2P transfer, it is used in the POS P2P Debit(132) transaction.</li> </ul>			
If not defined, the value is calculated automatically basing on the transaction parameters.			
FIMI/POSRequest/Rq/ThisTranId	Int(12)	In	Identifier of the current transaction
For the <i>POS Payment Pass</i> (175), <i>POS CashPaymentPass</i> (184) transactions:  ID of the current transaction which is resent for the authorization after the dialog with a customer.  The field is defined if the following authorization responses have been received for this transaction: AuthRespCode = Select Bill (13) or Confirm Payment Precheck (12).			
For the <i>Purchase Cancellation</i> (125) transaction:  ID of the original transaction that must be canceled.			
For the <i>POS P2P Pass</i> (135) / <i>C2C Pass</i> (802) transaction:  ID of the current transaction which is resent for the authorization after the dialog with a customer.  The field is defined if authorizer responses AuthRespCode = Customer confirmation requested (14) were received for this transaction.			

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/POSRequest/Rq/BSPPAN	Str(20)	In	Billing Service Provider PAN and PIN.
FIMI/POSRequest/Rq/BSPPIN	Str(10)	In	The fields are specified in the primary request for the POS Payment Pass(175) transaction, if the transaction is for the vendor of the Billing Service Provider.  E.g.: for payment to the «Gorod» system:  BSPPAN – card number of the «Gorod» system; BSPPIN – card PIN as a plain text
FIMI/POSRequest/Rq/ToAccount2	CustAcct (4000)	In	Payment personal attributes or bill identifier.  Bill identifier is defined in the repeated requests for the POS Payment Pass(175) transaction, if the transaction ) if it is the transaction to the Billing Service Provider
FIMI/POSRequest/Rq/TranNumber	Str(50)	In	ID used to search for or reverse the transaction.  This field is mandatory for <i>112 Pre-Purchase Complete</i> and <i>124 Pre-Purchase Increment</i> Standalone transactions – it must contain the information from the field of the original <i>111 Pre-Purchase</i> Standalone transaction.
FIMI/POSRequest/Rq/OrigTranId	Int(12)	In	Original transaction identifier
<p>The field is used:</p> <p><i>For the Merchandise Return(114) transaction,</i></p> <p><i>For the transaction with the Slip(2) mode:</i></p> <p>Used to search for the current transaction and link it with the original transaction.</p> <p><i>For the POS ECPurchase (856) and POS ECRefund (857) transactions:</i></p> <p>Used to search for the current transaction and link it with <i>Purchase(110)</i>, <i>POS ECPurchase (856)</i> and <i>Merchandise Return(114)</i>.</p> <p><i>For the POS P2P Credit(133) transaction:</i></p> <p>Used to search for and link the <i>POS P2P Credit(133)</i> transaction with <i>POS Purchase(110)</i> when using the service of a payment with tips.</p> <p>Used to search for and link the <i>POS P2P Credit(133)</i> transaction with the <i>POS P2P Debit(132)</i> <i>DraftCapture=8(Discontinuous)</i> transaction in case of using the service of discontinuous NSPK P2P transfers.</p> <p><i>For the POS P2P Debit (132) transaction:</i></p> <p>Used to search for and link the <i>POS P2P Debit (132)</i> <i>DraftCapture=8(Discontinuous)</i> <i>AuthAdvice(110)</i> transaction with <i>AuthRequest(100)</i> in case of using the service of discontinuous NSPK P2P transfers.</p>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<i>For the C2C Pass(802) transaction:</i>			
If BusinessApplicationIdent=CC(C2C NSPK FPS) in the transaction, the field transfers ID of the transaction generated within the <i>CheckAlias</i> operation implying the alias check in the external system:			
If BusinessApplicationIdent=MP(C2C Pull NSPK FPS) in the transaction, the field transfers the identifier of the transaction generated within the C2C transfer generation request (837-C2C Pull), the transaction identifier is returned within the <i>GetInvoice</i> operation with InvoiceType=2 in the <i>FIMI/GetInvoice/Rp/Invoices.RequestTranId</i> field.			
<i>For the POS Installment Details(136) transaction:</i>			
ID of the original Installment transaction.			
<i>For the C2B Pass(822)/B2B Pass(805) transaction:</i>			
The field transfers ID of the transaction generated in the <i>GetQRCodeInfo</i> operation, QR code is checked and code data are received. It is required for the transfer of data on the QR code to the <i>C2B Pass(822)/B2B Pass(805)</i> transaction.			
<i>For the C2CI Pass(872) transaction:</i>			
The field transfers ID of the <i>FPS Verification Pass(870)</i> transaction generated within the <i>AliasVerification</i> operation. It is required for the transfer of the recipient data to the <i>C2CI Pass(872)</i> transaction.			
FIMI/POSRequest/Rq/OrigTime	Time	In	Date and time of the original transaction.  The field is mandatory for transaction <i>Purchase Cancellation(125)</i> .
FIMI/POSRequest/Rq/EC3DSVersion	Int(1)	In	3D Secure version  Available values: 1 – 3D Secure Version 1.0 (3DS 1.0) 2 – 3D Secure Version 2.1 (3DS 2.1) 3 – 3D Secure Version 2.2 (3DS 2.2) 4 – 3D Secure Version 2.3 (3DS 2.2) If NULL, it is 1.
FIMI/POSRequest/Rq/ECI	Str(2)	In	ECommerce indicator
For the detailed description and list of available values, refer to the following specifications:			
<b>For NSPK cards</b>			
- Participant Interface Specification. NSPK OPCC			
<b>For UPI cards</b>			
- Technical Specifications on Bankcard Interoperability Part II Online Message F60.2.8: E-commerce Identifier (ECI)			
<b>For MC cards</b>			
- Mastercard Customer Interface Specification DE 48—Additional Data—Private Use, Subelement 42—Electronic Commerce Indicators, Subfield 1—Electronic Commerce Security Level Indicator and UCAF Collection Indicator, Position 3 (UCAF Collection Indicator)			
<b>For Visa cards</b>			
- Visa POS SMS / BASE I Technical Specifications, Volume 1 Field 60.8/Positions 9–10: Mail/Phone/Electronic Commerce and Payment Indicator			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/ECTranId	Str(40)	In	Transaction ID on merchant side
It is used in 3D Secure VISA, 3D Secure MC 2.0 and 3D Secure UPI 2.0:			
<b>For Visa</b>			
The string format corresponds to the description from BASE-I F126.8 – 40 symbols in Hex.			
<b>For Mastercard</b>			
The string format, the length is 36 characters (if the value contains symbol '-' and is transferred in the 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx' format) or 32 characters (the the value does not contain symbol '-'). The example for 3DS 2.0 MC: "f38e6948-5388-41a6-bca4-b49723c19437".			
<b>For UPI</b>			
The string format corresponds to Technical Specifications on Bankcard Interoperability Field 125 – string with the length of 36 characters.			
FIMI/POSRequest/Rq/EC3DSTranCondition	Str (1000)	In	Information on the 3D Secure transaction conditions.  Consists of the "Parameter" = "Value" subfields separated by ';' where "Parameter" – MPI setting os transaction execution condition.
FIMI/POSRequest/Rq/TransactionStatus	Str(1)	In	ECommerce Transaction Status, for details, refer to <i>Participant Interface Specification. NSPK OPCC</i> .
FIMI/POSRequest/Rq/TransactionStatusReason	Str(2)	In	ECommerce Transaction Status Reason, for details, refer to <i>Participant Interface Specification. NSPK OPCC</i> .
FIMI/POSRequest/Rq/EC3DSNetwork	Int	In	Network whose rules were applied to execute the ECommerce transaction.  <i>For the list of available values, refer to section 3.23.</i>
FIMI/POSRequest/Rq/ECAAV	Str(40)	In	MS UCAF or MC Static AVV.  It is used in MC 3D Secure. The format of string corresponds to the description from CIS 48.43 (40 symbols in Base 64).  It is used in the DSRP transactions (CardholderTerminal=1). The format of string corresponds to the description from CIS 104.1 (40 symbols in Base 64).
FIMI/POSRequest/Rq/CAVV	Str(40)	In	CAVV/AEVV/Authentication Value
<b>VISA CAVV</b>			
It is used in VISA 3D Secure. The format of string corresponds to the description from BASE-I F126.9 (40 characters in Hex).			
<b>UPI VCODE</b>			
It is used in UPI SecurePlus. The format of string corresponds to the description from Technical Specifications on Bankcard Interoperability Part II Online Message chapter "Verification Code" (20 characters).			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>UPI Authentication Value</b>			
It is used in 3D Secure UPI 2.0. The format of string corresponds to the description from Technical Specifications on Bankcard Interoperability Field 61.6 Usage AM (28 characters in base64).			
<b>JCB CAVV</b>			
It is used in 3D Secure JCB. The format of string corresponds to the description from JCB Regulatory Publications System Specifications – Online Interface Guide chapter "Data Elements of J-Link8" (40 characters in Hex).			
<b>American Express AEVV</b>			
It is used in 3D Secure American Express. The format of string corresponds to the description from "POS Authorization Request (1100) Bit 61 National Use Data" ("6 American Express Verification Value (AEVV)") (40 characters in Hex).			
FIMI/POSRequest/Rq/TAVV	Str(40)	In	<p><b>For Visa</b> VISA TAVV/DTVV cryptogram The format of string is 40 characters in Hex. Also refer to <i>BASE I Technical Specifications, Field 126.8 - Transaction ID (XID)</i>.</p> <p><b>For Mastercard</b> DSRP Cryptogram The format of string is 28 characters in Base 64. Also refer to <i>Customer Instarface Specification, DE 104 - Digital Payment Data, SE 001 - Digital Payment Cryptogram</i>.</p>
FIMI/POSRequest/Rq/CAVVOK	Str(2)	In	CAVV verification result <i>For details, see FIMI/POSRequest/Rp/CAVVOK.</i>
FIMI/POSRequest/Rq/CustomerAddress	Str(40)	In	Customer address. It is used in the AVS service
FIMI/POSRequest/Rq/CustomerPostalCode	Str(9)	In	Customer postal code. It is used in the AVS service.
FIMI/POSRequest/Rq/AVCode	Str(1)	In	Address verification result <i>See the details in the FIMI/POSRequest/Rq/AVCode field.</i>
FIMI/POSRequest/Rq/RecipientLastName	Str(35)	In	Recipient name
FIMI/POSRequest/Rq/RecipientPostalCode	Str(10)	In	Recipient postal code
FIMI/POSRequest/Rq/RecipientDateBirth	Str(8)	In	Recipient birth date (it is transferred in the YYYYMMDD format)
FIMI/POSRequest/Rq/RecipientAccountNumber	Str(20)	In	Recipient account

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/DetailAddenda	Str	In	<p>Information on the additional programs, services and other peculiarities concerning the financial transaction.</p> <p>For Network = 32 – Amex, it corresponds to the message <i>AEGNS Industry Specific Detail Addenda</i> (9240/9340).</p> <p>For Network != 32 – Amex, it is a string in UAMP format presented as a set of pairs &lt;field name&gt;=&lt;value&gt; separated by 0x10. Available names of fields:</p> <p>For Network = 11 – VISA:</p> <ul style="list-style-type: none"> <li>- «VISA AVIATIC KET» – contains the data on the air ticket number</li> </ul>
FIMI/POSRequest/Rq/DetailAddendaExt	StrXML	In	<p>Information in XML format on the additional programs, services and other peculiarities concerning the financial transaction</p> <p>If defined, it overrides the data from <i>DetailAddenda</i> field.</p> <p>For details on parameters, refer to XSD scheme <i>tranAddendum.xsd</i> (the files do not include the FIMI WSDL/XSD schemes)</p>
FIMI/POSRequest/Rq/AmountOrigDCC	Num	In	<p>Original amount of the incoming external transaction</p> <p>The field is filled if the Dynamic Currency Conversion was performed during the transaction execution</p>
FIMI/POSRequest/Rq/CurrencyOrigDCC	Int	In	<p><b>For POS Exchange Rate Inquiry (194):</b></p> <p>Currency to convert the transaction amount to.</p> <p><b>For other POS transactions:</b></p> <p>Original currency of the incoming external transaction</p> <p>The field is filled if the Dynamic Currency Conversion was performed during the transaction execution</p>
FIMI/POSRequest/Rq/DCCFee	Num	In	<p>Dynamic Currency Conversion fee</p> <p>The field is filled if Dynamic Currency Conversion was performed during the transaction execution.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/DCCRequest	Int(1)	In	DCC request indicator. Available values: 1 – DCC conversion request. 0 or NULL – common transaction.
FIMI/POSRequest/Rq/DCCTranId	Int(12)	In	ID of the transaction within which the DCC request was executed, it is used to link the DCC request transaction and common transaction.
FIMI/POSRequest/Rq/ICC_TermCaps	Int	In	EMV Tag – 9F33. Terminal Capabilities (bit mask)
FIMI/POSRequest/Rq/ICC_TVР	Hex(5)	In	EMV Tag – 95. Terminal Verification Results (bit mask)
FIMI/POSRequest/Rq/ICC_Random	Hex(4)	In	EMV Tag – 9F37. Unpredictable Number generated by the terminal.
FIMI/POSRequest/Rq/ICC_TermSN	Str(8)	In	EMV Tag – 9F1e. Serial Number
FIMI/POSRequest/Rq/ICC_TermType	Int	In	EMV Tag – 9F35. Terminal Type
FIMI/POSRequest/Rq/ICC_IssuerData	Hex(32)	In	EMV Tag – 9F10. Issuer Discretionary Data
FIMI/POSRequest/Rq/ICC_Cryptogram	Hex(8)	In	EMV Tag – 9F26. Application Cryptogram
FIMI/POSRequest/Rq/ICC_CryptInformData	Int	In	EMV Tag – 9F27. Cryptogram Information Data
FIMI/POSRequest/Rq/ICC_AppTranCount	Str(8)	In	EMV Tag – 9F36. Application Transaction Counter (ATC). It is transferred in the decimal format.
FIMI/POSRequest/Rq/ICC_TermTranCount	Str(8)	In	EMV Tag – 9F41. Transaction Sequence Counter
FIMI/POSRequest/Rq/ICC_AppProfile	Int	In	EMV Tag – 82. Application Interchange Profile (bit mask)
FIMI/POSRequest/Rq/ICC_TranType	Int	In	EMV Tag – 9C. Transaction Type
FIMI/POSRequest/Rq/ICC_TermCountry	Int	In	EMV Tag – 9F1A. Terminal Country Code (indicates the country of the terminal, represented according to ISO 3166)
FIMI/POSRequest/Rq/ICC_TranDate	Str(6)	In	EMV Tag – 9A. Transaction Date (YYMMDD)
FIMI/POSRequest/Rq/ICC_Amount	Str(12)	In	EMV Tag – 9F02. Amount
FIMI/POSRequest/Rq/ICC_Currency	Int	In	EMV Tag – 5F2a. It indicates the currency code of the transaction according to ISO 4217
FIMI/POSRequest/Rq/ICC_CBAmount	Str(12)	In	EMV Tag – 9F03. Amount of cash back transaction
FIMI/POSRequest/Rq/ICC_CVMRes	Int	In	EMV Tag – 9F34. CVM Results
FIMI/POSRequest/Rq/ICC_CardMember	Int	In	EMV Tag – 5F34. Card member number

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/ ICC_FormFactorIndicator	Hex(32)	In	<p>EMV Tag – 9F6E</p> <p>This field can transfer one of the following values:</p> <ul style="list-style-type: none"> <li>- Form Factor Indicator (FFI). This tag is personalized on the card or device and transfers additional information about the contactless device, its security features, and the technology used to acquire the transaction. The fixed length is 4 bytes.</li> <li>- Third Party Data (TPD). For detailed description of the tag format, refer to <i>MasterCard MChip Requirements for Contact and Contactless</i>. The variable length is of 5-32 bytes.</li> </ul>
FIMI/POSRequest/Rq/ ICC_CustomerExclusiveData	Hex(32)	In	EMV Tag – 9F7C. Customer Exclusive Data. This tag is personalized on the card or device. This tag is carried in U.S. contactless transactions and contains issuer proprietary information in TLV format. If present in an interregional transaction, the tag transfers additional data.
FIMI/POSRequest/Rq/ ICC_DedicatedFileName	Hex(32)	In	EMV Tag – 84. Dedicated File Name The field is transferred in the hexadecimal format with the variable length of 10 to 32 characters.
FIMI/POSRequest/Rq/ ICC_CardholderName	Hex(26)	In	EMV Tag – 5F20. Cardholder Name The field is transferred in the hexadecimal format with the variable length of 2 to 26 characters.
FIMI/POSRequest/Rq/ ICC_CardholderNameExt	Hex(45)	In	EMV Tag – 9F0B. Cardholder Name Extended The field is transferred in the hexadecimal format with the variable length of 27 to 45 characters.
FIMI/POSRequest/Rq/ ICC_CardProductIdentificationInfo	Hex(16)	In	EMV Tag – 9F63. Card Product Identification Information
FIMI/POSRequest/Rq/ ICC_AppCurrencyCode	Int	In	EMV Tag – 9F42. Application Currency Code

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/ExtPSFields	StrUAMP (1000)	In	<p>Information on the fields of the ISO message for the external payment system. For details on the field format, refer to <i>TW International Container Fields Specification</i>.</p> <p>This field is mandatory for the 112 <i>Pre-Purchase Complete</i> and 124 <i>Pre-Purchase Increment</i> Standalone transactions – it must contain the information from the field of the original 111 <i>Pre-Purchase</i> Standalone transaction.</p>
FIMI/POSRequest/Rq/ExtTranAttr	StrUAMP (1000)	In	Additional transaction attributes for the external network. For the format description, refer to specification <b>TIP(E).pdf</b> .
FIMI/POSRequest/Rq/ExtRRN	Str(32)	In	Transaction number as it was sent to the external payment system via host interface
FIMI/POSRequest/Rq/ExtSTAN	Int(12)	In	STAN (ISO field 11) as it was sent to the external payment system via the host interface
FIMI/POSRequest/Rq/SettleDate	Time	In	Settlement date
FIMI/POSRequest/Rq/Subcode	Int(12)	In	<p>It is used to transfer the additional subcode in the <i>POS Message to Financial Institution</i> transaction (126)</p> <p>It is used to transfer the transaction code for the fee calculation in the <i>POS Calculation Fee</i> transaction (185)</p>
FIMI/POSRequest/Rq/CalcIssuerFee	Int(1)	In	<p>Indicates that the issuer fee must be calculated.</p> <p>It is transferred in the <i>POS Calculation Fee</i> (185), <i>POS P2P Calc Fee</i> (134) transactions and the financial transaction following the preliminary fee calculation.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>1 – issuer fee calculation is required (for the financial transaction, it means that the preliminary fee calculation was performed).</li> <li>0, NULL – common transaction of the acquiring fee calculation.</li> </ul>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/CalcFeeTranId	Int(12)	In	<p>ID of the POS Calculation Fee (185) or POS P2P Calc Fee (134) transaction that involved the preliminary calculation of the issuer fee.</p> <p>It is used if the value of the CalclssuerFee field is 1.</p>
FIMI/POSRequest/Rq/BatchNumber	Int(12)	In	<p>Batch number.</p> <p>It is used for the <i>CloseDay</i> (162) transaction</p>
FIMI/POSRequest/Rq/WalletData	Str(3)	In	MasterCard Wallet Program Data
FIMI/POSRequest/Rq/StandAlone	Int(1)	In	<p>Indicates the Standalone transaction.</p> <p>It can be filled for the 110 – Purchase, 111 Pre-Purchase, 112 Pre-Purchase Complete, 124 Pre-Purchase Increment, and 125 – Purchase Cancellation POS transactions, for other transactions, the field is ignored.</p> <p>If NULL, it is 0</p> <p><b><i>The field is considered obsolete, it is recommended to use the TranCategory field with value 52(Standalone PreAuth) instead.</i></b></p>
FIMI/POSRequest/Rq/TranCategory	Int(2)	In	<p>Transaction category</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>0, NULL – Normal.</li> <li>52 – Standalone PreAuth.</li> <li>60 – Detokenization.</li> <li>61 – Detokenization Only.</li> <li>62 – Detokenization History Advice.</li> </ul> <p>The Standalone PreAuth(52) value is allowed for the following transactions (TranCode):</p> <ul style="list-style-type: none"> <li>110 – Purchase,</li> <li>111 – Pre Purchase,</li> <li>112 – Pre Purchase Complete,</li> <li>124 – Pre Purchase Increment,</li> <li>125 – Purchase Cancellation.</li> </ul>
FIMI/POSRequest/Rq/AUAR	Str(11)	In	Agent Unique Account Result. This data field is used to identify transactions that originate from a particular service provider.
FIMI/POSRequest/Rq/ICC_FormFactorIndicator	Str(64)	In	<p>EMV Tag – 9f6e. It is transferred in the hexadecimal format.</p> <p>This field can transfer one of the following values:</p> <ul style="list-style-type: none"> <li>- Form Factor Indicator (FFI). This tag is personalized on the card or device and transfers additional information about the contactless</li> </ul>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			<p>device, its security features, and the technology used to acquire the transaction. The fixed length is 4 bytes.</p> <p>- Third Party Data (TPD). For detailed description of the tag format, refer to <i>MasterCard MChip Requirements for Contact and Contactless</i>. The variable length is of 5-32 bytes.</p>
FIMI/POSRequest/Rq/ ICC_CustomerExclusiveData	Str(64)	In	<p>EMV Tag – 9f7c. Customer Exclusive Data. This tag is personalized on the card or device. This tag is carried in U.S. contactless transactions and contains issuer proprietary information in TLV format. If present in an interregional transaction, the tag transfers additional data.</p> <p>The field is of variable length (32 bytes maximum). It is transferred in hexadecimal format.</p>
FIMI/POSRequest/Rq/ ICC_DedicatedFileName	Str(32)	In	<p>EMV Tag – 84. Dedicated File Name</p> <p>The field is transferred in the hexadecimal format with the variable length of 10 to 32 characters.</p>
FIMI/POSRequest/Rq/ PaymentFacilitatorId	Str(11)	In	Payment Facilitator ID
For details, refer to the following specifications:			
<ul style="list-style-type: none"> <li>- <b>For Mastercard cards</b> MasterCard Customer Interface Specification, DE 48 Subelement 37 Subfield 1</li> <li>- <b>For UPI cards</b> UPI Technical Specifications on Bankcard Interoperability, DE 117, Usage PF</li> <li>- <b>For Visa cards</b> VISA BASE I Technical Specifications, DE104, Usage 2, Dataset Value Hex 65</li> </ul>			
FIMI/POSRequest/Rq/ IndependentSalesOrgId	Str(11)	In	<p>Independent Sales Organization ID.</p> <p>For details, refer to <i>MasterCard Customer Interface Specification, DE 48 Subelement 37 Subfield 2</i>.</p>
FIMI/POSRequest/Rq/SubMerchantId	Str(15)	In	<p>Sub-merchant ID.</p> <p>For details, refer to <i>MasterCard Customer Interface Specification, DE 48 Subelement 37 Subfield 3</i>.</p>
FIMI/POSRequest/Rq/SubMerchantName	Str(18)	In	<p>Sub-merchant name</p> <p>It corresponds to the 2<sup>nd</sup> part of the field (after '*') in DE 43 Subfield 1: see the description in <i>MasterCard Customer Interface Specification</i>.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/SubMerchantCity	Str(13)	In	Sub-merchant city, see the description in MasterCard Customer Interface Specification.
FIMI/POSRequest/Rq/SubMerchantRegion	Str(3)	In	Sub-merchant region
FIMI/POSRequest/Rq/SubMerchantCountryCode	Int(3)	In	Sub-merchant country code, see the description in MasterCard Customer Interface Specification.
FIMI/POSRequest/Rq/SubMerchantPostalCode	Str(10)	In	Sub-merchant postal code, see the description in MasterCard Customer Interface Specification.
FIMI/POSRequest/Rq/SubMerchantStreetAddress	Str(48)	In	Sub-merchant street address, see the description in the specification MasterCard IPM Clearing Formats, DE 43 Subfield 2.
FIMI/POSRequest/Rq/SubMerchantCategoryCode	Int(4)	In	Sub-merchant category code
FIMI/POSRequest/Rq/SubMerchantTaxId	Str(20)	In	Sub-merchant TPN
FIMI/POSRequest/Rq/SubMerchantTaxReasonRegCode	Str(10)	In	Sub-merchant tax registration reason code
FIMI/POSRequest/Rq/SubMerchantTermId	Str(15)	In	Identifier of the sub-merchant terminal
FIMI/POSRequest/Rq/MerchantCapability	Str(1)	In	MasterCard Merchant Capability: see the description in the specification MasterCard IPM Clearing Formats, PDS 0042
FIMI/POSRequest/Rq/MerchantInitiatedTran	Int	In	Additional attributes on the transaction initiated by entering the card data (or their analog) from the merchant data storage.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
Bit mask:			
- bit 1 – Merchant Initiated Transaction, Credential on file entry mode. Attribute of the transaction initiated by merchant by entering the card data (or their analog) from the merchant data storage.			
- bit 2 – Customer Initiated Transaction, Credential on file entry mode. Attribute of the transaction initiated by cardholder by entering the card data (or their analog) from the merchant data storage.			
- bit 3 – Credential on file Initial transaction.  Attribute of the initial transaction for which the merchant saves the card data (or their analog) for possible subsequent transactions.			
- bit 4 - Deferred transaction. Attribute of the delayed transaction.			
- bit 5 – Terminated.  Canceled subscription attribute.			
- Bit 6 – Deposit Only. Attribute indicating that the card data (or its analog) is saved for the next card/account deposit transaction.			
- bit 7 – Partial/Split Shipment. Attribute indicating that items of the same order are sent separately.			
<i>For details on the field format, refer to "TW International Container Fields Specification", the TermAdditionalData container, the MIT field.</i>			
FIMI/POSRequest/Rq/ MerchantVerificationValue	Hex(5)	In	VISA Merchant Verification Value  Refer to <i>BASE I Technical Specifications, Field 62.20 - Merchant Verification Value</i>
FIMI/POSRequest/Rq/ AssignedId	Str(6)	In	MasterCard Assigned ID  Refer to <i>MasterCard Customer Interface Specification, DE 48 Subelement 32</i>
FIMI/POSRequest/Rq/RespCode	Int	In	Authorizer response code. See the list of available values in section 3.9.  The field must be filled in transactions with the Advice type (TranType = 120 and 220)  If =NULL (in transactions with the Advice type if TranType = 120 and 220), the default value 1 (Approved) will be used
FIMI/POSRequest/Rq/ExtRespCode	Str(10)	In	Original response of IPS or external system.  It is allowed to fill in the field in transactions with the Advice type (TranType = 120 and 220).

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/ BonusProgramName	Str(20)	In	Bonus program name. It is filled for the POS Redem Bonus (182) transaction.
FIMI/POSRequest/Rq/ BonusAccumulation	Str(2000)	In	<p>Array with the information on accumulated or redeemed bonuses. The string in the UAMP format provides the number of the bonus account and amount that impacts it in the array of records:</p> <pre>{   1) ProgramName Str(20) – bonus program name (ID);   2) Account Str(30) – bonus account number;   3) Amount Num(15,2) – accumulated/redeemed amount in the bonus account currency; }</pre> <p>Amount&gt;0 – bonuses are accumulated, Amount&lt;0 – bonuses are redeemed.</p>
FIMI/POSRequest/Rq/ TokenRequestorId	Str(11)	In	<p>The field contains ID of the object that initiated tokenization process. It is transferred in case of the transaction by token.</p>
FIMI/POSRequest/Rq/ CardholderTerminal	Int(1)	In	<p>Flag indicating that transaction was generated using either MasterCard DSRP or Visa Application-based E-Commerce technologies.</p> <p>1 – “DSRP transaction”. 0 – “non-DSRP transaction”</p>
FIMI/POSRequest/Rq/ PremisesTermLocation	Int(1)	In	<p>Attribute that shows whether the terminal is on or off premises of the sales/service organization.</p> <p>1 – “On premises”. 0 – “Off premises”</p>
FIMI/POSRequest/Rq/ SubsequentRemote	Int(1)	In	<p>Flag indicating that the previous transaction was generated using either MasterCard DSRP or Visa Application-based E-Commerce technologies and completely authenticated, contained either the cryptogram or UCAF.</p> <p>1 – “Subsequent transaction”. 0 – “not Subsequent transaction”.</p>
FIMI/POSRequest/Rq/PrizeID	Str(40)	In	Identifier of prize to exchange the bonuses for. It is used for the <i>POS Redeem Bonus (182)</i> transaction

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/PrizeQuantity	Num	In	Number of prizes to exchange the bonuses for. It is used for the <i>POS Redeem Bonus (182)</i> transaction.
FIMI/POSRequest/Rq/TokenAssuranceLevel	Str(2)	In	Token Assurance Level – assigned token assurance level. For details, refer to <i>MasterCard Customer Interface Specification, DE 48, subelement 33, Subfield 5</i> .
FIMI/POSRequest/Rq/SubMerchantCity	Str(13)	In	Sub-Merchant City – merchant organization city. For details, refer to <i>MasterCard Customer Interface Specification</i> .
FIMI/POSRequest/Rq/SubMerchantCountryCode	Int(3)	In	Sub-Merchant Country Code – merchant organization country code. For details, refer to <i>MasterCard Customer Interface Specification</i> .
FIMI/POSRequest/Rq/SubMerchantPostalCode	Str(10)	In	Sub-Merchant Postal Code – merchant organization postal code. For details, refer to <i>MasterCard Customer Interface Specification</i> .
FIMI/POSRequest/Rq/SenderRRN	Str(32)	In	Sender Reference Number
FIMI/POSRequest/Rq/InstallmentData	StrUAMP	In	Contains the Installment transaction data in the UAMP format. It is also transferred in the <i>POS Installment Details(136)</i> transaction. For details on the field format, refer to <b>ContainerFields(E).pdf</b> .
FIMI/POSRequest/Rq/Purchaseldentifier	Str(26)	In	Purchase Identifier. For details, refer to <i>SMS POS (Visa &amp; Visa Electron) Technical Specifications, Volume 1, Field 62.7</i>
FIMI/POSRequest/Rq/Reason	Int(9)	In	Transaction reason code. It can be specified for the merchant-initiated transactions. Refer to <i>VISA BASE I Technical Specifications, Volume 1, Field 63.3 - Message Reason Code</i>
FIMI/POSRequest/Rq/TransactionIdentifier	Str(15)	In	Unique transaction ID assigned by the external payment system. Also refer to: - <i>VISA BASE I Technical Specifications, Volume 1 SMS Technical Specifications, Volume 1 Field 62.2 - Transaction Identifier</i> - <i>MasterCard Customer Interface Specification Single Message System Specifications DE 48 Subelement 63 - Trace ID</i>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rq/ EIDRequestCode	Int	In	E-ID Request Code  For details, refer to <i>Customer Interface Specification DE 48 SE 53</i> .
FIMI/POSRequest/Rq/ BusinessApplicationIdent	Str(2)	In	<b>For VISA</b> Business Application Identifier, See VISA Field 104 Dataset 57 Tag 01 <b>For UPI</b> Business Indication Refer to <i>Technical Specifications on Bankcard Interoperability Part II Online Message F104 Usage: BI</i> . <b>For NSPK FPS</b> Within the C2C Pass(802) transaction, it is required to transfer one of the following values: CC – for C2C NSPK FPS MM – for M2M NSPK FPS MP – for C2C Pull NSPK FPS
FIMI/POSRequest/Rq/FPTTI	Str(3)	In	Funding/Payment Transaction Type Indicator, see the description in: <ul style="list-style-type: none"> <li>• MC CIS - DE 48 Subelement 77</li> <li>• NSPK PIS – field 48 element 14</li> </ul>
FIMI/POSRequest/Rq/ SpecialConditionIndicator	Str(1)	In	Special Condition Indicator  For details, refer to <i>VISA Field 60.4</i> .
FIMI/POSRequest/Rq/ TIPAmount	Num	In	Tip amount It is used in the <i>mVisa Merchant Payment</i> and <i>POS Purchase(110)</i> P2P transfers with tips. If specified for <i>POS Purchase(110)</i> , it is considered that tips are already included in the transaction amount (Amount).

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/POSRequest/Rq/MerchantData	StrUAMP	In	<p>It contains data of the merchant organization that is the destination of:</p> <ul style="list-style-type: none"> <li>- P2P transfer</li> <li>POS P2P Credit (133)</li> <li>POS P2P Pass (135)</li> <li>- QR code purchase</li> <li>POS QR Purchase Debit (838)</li> <li>POS QR Purchase Credit (839)</li> <li>POS QR Purchase Pass (840)</li> </ul> <p><i>For details on the field format, refer to "TW International Container Fields Specification".</i></p> <p><i>It is used in the mVisa Merchant Payment, Masterpass QR Payment transfers.</i></p>
FIMI/POSRequest/Rq/SenderPhoneNumber	Str(16)	In	<p>Customer phone number</p> <p>It is used in the mVisa Merchant Payment transfers. Also refer to VISA SMS POS Technical Specifications, Field 56, Dataset ID 02, Tag 83.</p> <p><b>The field is obsolete, it is recommended to send data in the P2PData structure.</b></p>
FIMI/POSRequest/Rq/SenderEmailAddress	Str(99)	In	<p>Customer E-mail</p> <p>It is used in the mVisa Merchant Payment transfers. Also refer to VISA SMS POS Technical Specifications, Field 56, Dataset ID 02, Tag 86.</p> <p><b>The field is obsolete, it is recommended to send data in the P2PData structure.</b></p>
FIMI/POSRequest/Rq/SecondaryId	Str(28)	In	<p>Secondary Id</p> <p>It is used in the mVisa Merchant Payment transfers. Also refer to mVisa Technical Specifications, Additional Data - Value 2.</p>
FIMI/POSRequest/Rq/ConversionDate	Time	In	Date of converting currencies by the external payment system
FIMI/POSRequest/Rq/MessageExtentions	ArrRec	In	<pre>{   str(100)* Name;   str(100)* Id;   str* Data; } [] – array of structures,</pre> <p>List of extensions with data on the PS risks:</p>
<p>Name - extension name.</p> <p>Available values: "rbaScoring" – for NSPK.</p> <p>"FPSScoring" – for NSPK FPS.</p> <p>"C2GRequisites" – for NSPK FPS.</p>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<p>Id – extension ID. Available values: "A00000065801" – for NSPK. "FPSScoring" – for NSPK FPS. "C2GRequisites" – for NSPK FPS.</p> <p>Data – extension value in the native format, it is transferred in the Base64 encoding, the data is re-coded in UTF8 in advance:</p> <p><b>for rbaScoring (NSPK)</b> For the format, refer to <i>TW International Container Fields Specification</i>, section 4.1.15.3.</p> <p><b>for FPSScoring (NSPK FPS)</b> It contains indicators of the suspicious operation in the UAMP format with the separator PS(0x10): str(16) NSPK_FPS_107 – indicator of the suspicious operation of the sender bank. str(16) NSPK_FPS_108 – indicator of the suspicious operation of the recipient bank. str(16) NSPK_FPS_109 – indicator of the suspicious operation of NSPK OPCC. Str(9) NSPK_FPS_125 – category of the sender account funds.</p> <p>Available for the C2C Pass(802), C2B Pass(822), B2B Pass(805) C2G Pass(860) transactions.</p>			

FIMI/POSRequest/Rq/ TerminalAdditionalData	CustAcct	In	Additional terminal data. Structure with additional data on the terminal.
---	----------	----	--

For the complete list of available subfields and their description, refer to *TW International Container Fields Specification*, description of the *TermAdditionalData* structure.

List of subfields:

Name	Type	Description
AC	Int(1)	Additional conditions. Bit mask: Bit 1 – transaction contains the repeated ATC value, following the previous single tap transaction. Bit 2 – online authorization is required.
VMI	Str(8)	Visa Merchant Identifier
RCAID	Str (150)	MC Remote Commerce Acceptor Identifier
SCA/DA	Str(1)	Delegated Authentication
SCA/LVEI	Str(1)	Low Value Exemption Indicator
SCA/TRAEI	Str(1)	Transaction Risk Analysis (TRA) Exemption Indicator
SCA/TMEI	Str(1)	Trusted Merchant Exemption Indicator
SCA/SCPI	Str(1)	Visa Secure Corporate Payment (SCP) Indicator
SCA/LRMI	Str(2)	Low-Risk Merchant Indicator
SCA/TPEI	Str(1)	Transport Fares and Parking Fees Indicator
SCA/SID	Str(32)	Session ID
SCA/AOI	Str(1)	Authentication Outage Indicator
BSKTID	Str(24)	Basket ID Identifier of the basket to be used for the NSPK service of the purchase with the use of the electronic certificate (FD-048.EL-38)
VTAXID	Str(20)	Sub-merchant TPN
VTRR	Str(10)	Tax registration reason code of a sub-merchant
AFTCDK	Int(1)	AFT Customer data key

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
OCTCDK	Int(1)		OCT Customer data key
MEI	Int(1)		Microenterprise Indicator
FIMI/POSRequest/Rq/RecurringData	CustAcct	In	Data on the Recurring Payment transactions. For the list of fields and their description, refer to document <b>ContainerFields(E).pdf</b> , description of the RecurringData structure.
FIMI/POSRequest/Rp/ThisTranId	Int	Out	ID of the transaction the system generates to the POSRequest transaction, that is ID of POS transaction sent for authorization
FIMI/POSRequest/Rp/AuthRespCode	Int*	Out	Authorization response code – for the list of available values, see p. 3.9.
FIMI/POSRequest/Rp/DetailedAuthRespCode	Int	Out	Additional authorizer response code that can be returned for some AuthRespCode to detail the decline reason. For the list of available values, see p. 3.9.
FIMI/POSRequest/Rp/AuthRespText	Str	Out	Text field in free format. It can be filled by the <b>Post-processing</b> algorithmix function.
FIMI/POSRequest/Rp/ExtRespCode	Str(10)	Out	Original response of IPS or external system
FIMI/POSRequest/Rp/AuthRespCodeCategory	Int(2)*	Out	Response code category: 0 – no additional response conditions. 1 – it is prohibited to repeat the transaction. 2 – transaction cannot be approved now, it is allowed to execute it later. 3 – new information on card is available, originator is allowed to attempt to update the card data and repeat the operation. 4 – token requirements are not met for this token type. 5 – payment cancellation. 6 – retailer does not have rights to process transactions on cards of this product code. 7 – repeat in an hour. 8 – repeat in 24 hours. 9 – repeat in 2 days. 10 – repeat in 4 days. 11 – repeat in 6 days. 12 – repeat in 8 days. 13 – repeat in 10 days.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rp/Network	Int	Out	Network whose rules are applied for the transaction execution. <i>See the list of available values in section 3.23.</i>
FIMI/POSRequest/Rp/ApprovalCode	Str(8)	Out	Approval code – it is present for the approved transactions.
FIMI/POSRequest/Rp/CVxOK	Int	Out	CVx\CVx2 check result. -1 – not checked, 0 – negative, 1 – positive
FIMI/POSRequest/Rp/AVCode	Str(1)	Out	Address verification result: The following values are filled by the Stand-In authorizer: A – address matches, but postal code does not match N – both address and postal code do not match U – no data for address verification W – postal code matches, but address does not match X – both address and postal code match  The value returned from the payment system is transferred unchanged.
FIMI/POSRequest/Rp/CAVVOK	Str(2)	Out	CAVV verification result <b>For VISA</b> <i>Base I, table 4 25 field 44.13 CAVV results codes</i> <b>For MasterCard</b> <i>Integer value or NULL.</i> <i>UCAF MasterCard verification result:</i> -1 - Static AAV, do not check NULL - AAV not checked 1 - AAV failed validation 2 - AAV passed validation
FIMI/POSRequest/Rp/DeclineReason	Str	Out	Transaction decline reason – defined for the declined transactions. It is in the English language for the standard verifications.
FIMI/POSRequest/Rp/FromAcct	Str(30)	Out	Number of the source account used for the authorization.
FIMI/POSRequest/Rp/ToAcct	Str(30)	Out	Number of the destination account used for the authorization.

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/POSRequest/Rp/AvailBalance	Num	Out	Available balance of the source account right after the transaction authorization. The currency code is specified in the BalanceCurrency field.
FIMI/POSRequest/Rp/LedgerBalance	Num	Out	Ledger balance of the source account right after the transaction authorization. The currency code is specified in the BalanceCurrency field.
FIMI/POSRequest/Rp/BonusDebt	Num	Out	Bonus/debt of the source account right after the transaction authorization. The currency code is specified in the BalanceCurrency field.
FIMI/POSRequest/Rp/BalanceCurrency	Int	Out	Balance currency code. The balance currency is specified by the settings of POS terminal (described in TWO) where the transaction is performed
FIMI/POSRequest/Rp/MaskBalances	Int(1)	Out	=1 — account balance must be masked
FIMI/POSRequest/Rp/Bills	ArrRec	Out	<pre>{     int* BillId;     str(100) VendorTitle;     str(200) BillingAcct;     time StartPeriodDate;     time EndPeriodDate;     time EffectiveDate;     num Amount;     int Currency;     str(40) ShortDescription;     str(100) Description;     str(4000) ExtDescription;     str(200) BillAlias;     int ParentBillId;     str(200) SubBillId;     str(4000) VendorExtPaymentParam;     str(200) CustomerId;     str(20) CustomerDDAgentId;     int(1) VendorExactPayment; }</pre> <p>[]} — the list of bills:</p>
<p>BillId – bill ID.          VendorTitle – vendor title.          BillingAcct – vendor personal account.          StartPeriodDate – start date of the period to be paid.          EndPeriodDate - start date of the period to be paid.          EffectiveDate – bill presentment start;          Amount – bill amount.          Currency – bill currency.          ShortDescription – brief description.          Description – full description.          ExtDescription – extended description.          BillAlias – code of the service.          ParentBillId – parent bill ID.</p>			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
SubBillId – sub-service ID. VendorExtPaymentParam – external parameter of the payment. CustomerId – customer ID in TW PostPay. CustomerDDAgentId – ID of the direct debit agent in TW PostPay; VendorExactPayment – attribute indicating that the vendor supports the full payment only.			
<i>It is defined in the response to the generation of payment (code 175) to vendor of the Bill Service Provider. E.g.: for payment to TW PostPay or to the "Gorod" system. The field is defined together with the value RespCode=13 (SelectBill).</i>			
FIMI/POSRequest/Rp/TotalAmount	Num	Out	Total amount of the payment transaction with all the fees (for AuthRespCode=12) - <b>TWO</b> acquiring fee and fees of the Bill Service Provider. E.g.: of the "Gorod" system.  Partially authorized amount of the transaction (for AuthRespCode=2 or AuthRespCode=3).  The field is defined together with the value AuthRespCode=2 (ApprovedPartial), AuthRespCode=3 (ApprovedPurchaseOnly) or AuthRespCode=12 (ConfirmPaymentPrecheck).
FIMI/POSRequest/Rp/Fee	Num	Out	Acquiring fee amount in the transaction currency (Currency).
FIMI/POSRequest/Rp/IssuerFee	Num	Out	Issuer fee amount in the account currency (CurrencyAcct).  It is returned for the POS Calculation Fee (185), POS P2P Calc Fee (134) transactions if CalclssuerFee = 1 in the request.
FIMI/POSRequest/Rp/Currency	Int	Out	ISO code of the transaction currency
FIMI/POSRequest/Rp/AccountCurrency	Int	Out	ISO code of the account currency
FIMI/POSRequest/Rp/ICC_IAD	Hex(16)	Out	EMV Tag – 91. Authorizer response code and cryptogram: ARPC + ResponseCode
FIMI/POSRequest/Rp/ICC_IssuerScript1	Hex (1000)	Out	Value of the issuer script 1
FIMI/POSRequest/Rp/ICC_IssuerScript2	Hex (1000)	Out	Value of the issuer script 2
FIMI/POSRequest/Rp/ExtPSFields	StrUAMP (1000)	Out	Information on the fields of the ISO message for the external payment system. For details on the field format, refer to <b>ContainerFields(E).pdf</b> .
FIMI/POSRequest/Rp/ExtTranAttr	StrUAMP (1000)	Out	Additional transaction attributes for the external network. For the format description, refer to specification TIP.pdf.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rp/ExtRRN	Str(32)	Out	Transaction number as it was sent to the external payment system via host interface
FIMI/POSRequest/Rp/ExtSTAN	Int	Out	Transactions STAN as it is sent to the external payment system via host interface
FIMI/POSRequest/Rp/SettleDate	Time	Out	Settlement date
FIMI/POSRequest/Rp/TextInfo	Str	Out	<p>Text information to be shown on the receipt.</p> <p>The field is filled in the POS Response Control algorithmix function</p>
FIMI/POSRequest/Rp/TransmissionTime	Time	Out	<p>Time of the Interchange message transmission.</p> <p>It is filled in the response to the <i>POS P2P Pass (135), POS QR Purchase Pass (840)</i> transactions.</p>
FIMI/POSRequest/Rp/TransmissionTime2	Time	Out	<p>Time of the Interchange message transmission.</p> <p>It is filled in the response to the <i>POS P2P Pass (135), POS QR Purchase Pass (840)</i> transactions.</p>
FIMI/POSRequest/Rp/RelatedTran	ArrRec	Out	<pre>{     int* RelatedTranID;     int* RelatedTranCode;     int* RelatedAuthRespCode;     time TransmissionTime;     str(32) ExtRRN; }</pre> <p>[]} – list of linked transactions:</p> <p>RelatedTranID – transaction ID  RelatedTranCode – transaction code  RelatedAuthRespCode – authorizer response code (see the list of available values in section 3.9).  TransmissionTime – time of the Interchange message transmission.  ExtRRN – transaction number as it is sent to the external payment system via host interface.  It is filled in response to transactions with the code 135, 149, 171, 175, 184, 802, 822, 805, 835, 840 (see the description in section 3.8 – <i>POS Transactions</i>).</p>
FIMI/POSRequest/Rp/ExtPAN	Str(30)	Out	<p>Masked value of the cardholder PAN.</p> <p>It is returned in case of the token transaction and detokenization transaction (TranCategory=60).</p>
FIMI/POSRequest/Rp/ExtExpDate	Time	Out	<p>Card expiration date.</p> <p>It is returned with the ExtPAN field for the detokenization transaction (TranCategory=60).</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rp/ToPAN	Str(30)	Out	Masked value of the transfer destination cardholder PAN. It is returned for the POS P2P Pass (135) and P2P transfers within the VISA ADS service.
FIMI/POSRequest/Rp/ToAcctDescr	Str(250)	Out	Destination account description. For the POS P2P Pass (135) and P2P transfers by the card alias within the VISA ADS service, it contains PAN of the transfer destination card.
FIMI/POSRequest/Rp/ExtPaymentFields	StrUAMP (1000)	Out	Additional payment attributes received from the external payment gateway
FIMI/POSRequest/Rp/InstallmentData	StrUAMP	Out	Data on the Installment transactions in the UAMP format. For details on the field format, refer to <b>ContainerFields(E).pdf</b> .
FIMI/POSRequest/Rp/TransactionIdentifier	Str(15)	Out	Unique transaction ID assigned by the external payment system  Refer to: <ul style="list-style-type: none"><li>- <i>VISA BASE I Technical Specifications, Volume 1 SMS Technical Specifications, Volume 1 Field 62.2 - Transaction Identifier</i></li><li>- <i>MasterCard Customer Interface Specification Single Message System Specifications DE 48 Subelement 63 - Trace ID</i></li></ul>
FIMI/POSRequest/Rp/MerchantInitiatedTran	Int	Out	Additional attributes on the transaction originated using card data (or their analog) from the merchant data storage. For details, see <i>FIMI/POSRequest/Rq/MerchantInitiate dTran</i> .
FIMI/POSRequest/Rp/PrepaidInfo	ArrRec	Out	{ str(20)* Code1; str(20) Code2; str(300) Info1; str(300) Info2; str(300) Info3; str(300) Info4; str(300) Info5; str(300) Info6; str(300) Info7; str(300) Info8; str(300) Info9; str(100) OrigBatchId; str(100) OrigId; } [] – information on the Prepaid code:

<b>Parameter</b>	<b>Type</b> (*- mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
Code1 – code 1. Code2 – code 2. Info1- Info9 – string providing the additional information. OrigBatchId – codes batch ID. OrigId – code ID. <i>It is returned for the approved transaction POS Prepaid Pass (171).</i>			
FIMI/POSRequest/Rp/MerchantData			
FIMI/POSRequest/Rp/ConversionDate			
FIMI/POSRequest/Rp/PAR			
FIMI/POSRequest/Rp/PARCreationDate			
FIMI/POSRequest/Rp/DCCRates			
Amount – amount. Currency – three-character numeric currency code (ISO 4217). Rate – rate of the original currency to the converted currency. DCCFee – conversion fee. Description – additional information.			
FIMI/POSRequest/Rp/FPSData	Str(8000)	Out	Additional data of the C2C/C2B/C2G transfer transaction.  UAMP string with the parameters is transferred in the Base64 encoding (data encoding prior to the Cp866 conversion), for the format and set of fields, refer to <i>TW International Container Fields Specification</i> , section 4.1.14.1 (FPSData).  <i>It is filled in the response for the C2C Pass(802), C2B Pass(822), B2B Pass(805), C2C Pull Pass(835), C2G Pass(860), C2CI Pass(872) transaction.</i>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/POSRequest/Rp/InvoiceId	Str(32)	In	<p>ID of Invoice created while processing the transaction (see the <i>GetInvoice</i> operation).</p> <p><i>It is returned in the response for the approved C2C Pass(802), C2B Pass(822), B2B Pass(805), C2C Pull Pass(835), C2G Pass(860), C2CI Pass(872) transaction.</i></p>
FIMI/POSRequest/Rp/LegalEntityData	CustAcct	Out	Additional data of a corporate customer owning the card/account on which a transaction was performed.

The following fields can be transferred:

Name	Type	Description
TaxId	Str(20)	Tax payer number (TPN) of a corporate customer.
TaxReasonRegCode	Str(10)	Tax registration reason code of a corporate customer.

FIMI/POSRequest/Rp/SCAIndicators      CustAcct      Out      SCA indicators

The following fields can be transferred:

Name	Type	Description
DA	Str(1)	Delegated Authentication
LVEI	Str(1)	Low Value Exemption Indicator
TRAEI	Str(1)	Transaction Risk Analysis (TRA) Exemption Indicator
TMEI	Str(1)	Trusted Merchant Exemption Indicator
SCPI	Str(1)	Secure Corporate Payment (SCP) Indicator
LRMI	Str(2)	Low-Risk Merchant Indicator
TPEI	Str(1)	Transport Fares and Parking Fees Indicator
SID	Str(32)	Session ID
AOI	Str(1)	Authentication Outage Indicator

FIMI/POSRequest/Rp/RecipientData	CustAcct	Out	<p>Additional data of the C2C/C2B transfer recipient.</p> <p><i>It is filled in the response for the approved C2C Pass(802), C2B Pass(822), B2B Pass(805), C2C Pull Pass(835), C2CI Pass(872) transaction.</i></p>
----------------------------------	----------	-----	--

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
-----------	----------------------------------	--------------------------------	-------------

The following fields can be transferred:

Name	Type	Description
Identity/Type	Int(1)	Type of the identity document: 1 – passport. 2 – driving licence. 3 – social insurance number. 4 – TPN. 5 – national ID. 6 – international passport. 7 – military identity card. 8 – other document type.
Identity/Number	Str	Document numer
Name/First	Str	First name
Name/Last	Str	Last name
Name/Middle	Str	Middle name
TaxPayerNumber	Str	TPN
PhoneNumber	Str	Phone number
Location/Address	Str	Address
Account/Number	Str	Bank account number
Account/Category	Str	Account funds category
PAM	Str	Personal Assurance Message(PAM)
CountryOfResidence	Int	Country ISO code

#### **StandAloneReversal – transaction Stand Alone reversal**

FIMI/StandAloneReversal/Rq/TranCode	Int(3)*	In	Transaction code. Available values: - Purchase (110) - Pre-purchase(111) - Pre-purchase Complete (112) - Mail/Phone Order(113) - Merchandise Return(114) - Cash Advance(115) - Card Verification(116) - POS Balance Inquiry(117) - Purchase With Cashback(118) - Purchase Adjustment(121) - Merchandise Return Adjustment(122) - Cash Advance Adjustment(123) - Pre-purchase Increment (124) - POS Message to Financial Institution(126) - Quasi-Cash(130) - POS P2P Debit(132) - POS P2P Credit(133) - POS P2P Calc Fee(134) - POS P2P Pass(135) - POS Deposit(140) - POS Deposit Adjustment(141) - Close Day(162) - POS Prepaid Debit(172) - POS Prepaid Credit(173) - POS Accumulate Bonus(181) - POS Redeem Bonus(182)
-------------------------------------	---------	----	--

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			- POS Calc Fee(185) (for the description of the above mentioned available operations, see p.3.8, section <i>POS-transactions</i> )
FIMI/StandAloneReversal/Rq/TermName	Str(16)	In	Name of the POS terminal described in <b>TWO</b> . If not filled, the POS terminal set by default for a certain FIMI terminal is used
FIMI/StandAloneReversal/Rq/TermInstName	Str(4)	In	Name of the POS terminal institution. If the terminal name (the TermName field) is not unique in <b>TWO</b> , this field must be filled to determine the POS terminal definitely. If not filled, the FIMI terminal institution will be used.
FIMI/StandAloneReversal/Rq/RetailerName	Str(150)	In	Retailer name. If the terminal name (the TermName field) is not unique in <b>TWO</b> , this field must be filled to determine the POS terminal definitely.
FIMI/StandAloneReversal/Rq/PAN	Str(19)	In	Card number
FIMI/StandAloneReversal/Rq/MBR	Int(3)	In	Card member number
FIMI/StandAloneReversal/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number (PAN and MBR fields).
FIMI/StandAloneReversal/Rq/PAN2	Str(34)	In	Number of the transfer destination card.  The field is specified for the <i>POS P2P Pass</i> (135) transaction. It can be specified for the <i>POS P2P Debit</i> (132), <i>POS P2P Credit</i> (133), <i>POS P2P Calc Fee</i> (134) transactions.  For all transactions, the field does not exceed 19 characters except for the <i>POS P2P Credit</i> (133) operation for which the destination account number with the length not exceeding 30 characters can be transferred.
FIMI/StandAloneReversal/Rq/MBR2	Int(3)	In	Number of the transfer destination cardholder.  The field is specified for transactions described in the PAN2 field.
FIMI/StandAloneReversal/Rq/CardUID2	Str(64)	In	Unique identifier of the card 2. It can be specified instead of the card number (PAN2 and MBR2 fields)

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/StandAloneReversal/Rq/FromAcctType	Int(2)*	In	Type of the source account. It is defined for all the transactions. Available values: 0-Default, 1-Checking, 11-Savings, 31-Credit, 91-Bonus
FIMI/StandAloneReversal/Rq/ToAcctType	Int(2)	In	Type of the destination account, it is used for the <i>POS Transfer Pass(149)</i> transaction. Available values: 0-Default, 1-Checking, 11-Savings, 31-Credit, 91-Bonus. If not defined, it= 0.
FIMI/StandAloneReversal/Rq/FromAccount	Str(30)	In	Source account. If defined, type of this account should match the one defined in the FromAcctType field or FromAcctType should be equal to 0.
FIMI/StandAloneReversal/Rq/Track2	Str(39)	In	Mag stripe contents in the format:  PAN=YYMM[SRV]... or PANDYYMM[SRV]...,  where: PAN – value passed in the PAN field; YYMM – the card validity period (two characters – year, two characters – month); SRV – 3-char value of the card service code; ... – other information from the mag stripe.  'PAN=YYMM' – mandatory part of the mag stripe.  Mag stripe other data starting with the service code can be absent.  If the CardUID field is filled instead of PAN and MBR fields, this field can contain no PAN and the mag stripe contents will have the following format: =YYMM[SRV]...
FIMI/StandAloneReversal/Rq/Amount	Num	In	Transaction amount
FIMI/StandAloneReversal/Rq/Amount2	Num	In	For the <i>Purchase Adjustment(121)</i> , <i>Merchandise Return Adjustment(122)</i> , <i>Cash Advance Adjustment(123)</i> , <i>Deposit Adjustment(141)</i> transactions – actual transaction amount.  For the <i>Purchase With Cashback(118)</i> transaction – change amount in cash.  For the <i>B2B Pass(805)</i> transaction – VAT amount if VAT is determined by the payer.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/StandAloneReversal/Rq/Fee	Num	In	Fee amount. If defined, it is considered that the transaction amount already includes the fee amount.
FIMI/StandAloneReversal/Rq/Currency	Int	In	Transaction currency. If not defined, it is default POS terminal currency specified in TWO
FIMI/StandAloneReversal/Rq/InvoiceNumber	Str(16)	In	Invoice number
FIMI/StandAloneReversal/Rq/OrigInvoiceNumber	Str(16)	In	Invoice number of the original transaction – it can be defined for the pre-purchase complete transaction (code 112) and for the adjustment transactions (codes 121,122,123,141)
FIMI/StandAloneReversal/Rq/ApprovalCode	Str(8)	In	Approval code. If defined, it will be the same in response. It is defined for the pre-purchase complete transaction (code 112).
FIMI/StandAloneReversal/Rq/CVV2	Str(4)	In	CVV2. If CVV2 is negative value, it is treated as follows: if CVV2=-99, it is replaced with 00; in other cases, the sign '-' is dismissed. See the results in the following table:  Position 1 0 – CVV2 Value is intentionally not provided 2 – CVV2 Value is Present but illegible 9 – Cardholder states that no CVV2 value is imprinted  Position 2 0 – Only the normal Response Code should be returned 1 – Both the Response Code and the CVV2 Result Code should be returned
FIMI/StandAloneReversal/Rq/Condition	Int	In	Transaction condition. For the list of available values, see section 3.19.
FIMI/StandAloneReversal/Rq/EntryMode	Int	In	Card entry mode. For the list of available values, see section 3.20. This field is mandatory for all transactions, except for the <i>CloseDay</i> (162) transaction.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/StandAloneReversal/Rq/ DraftCapture	Int(1)	In	<p>Mode of receiving transaction from the terminal:</p> <p>0 – Not Capture – financial approval (slip) is required; it is characteristic of the original transaction of voice authorization and Pre-Purchase transaction (111);            1 – Draft Capture – transaction is the basis for posting an entry to the retailer account;            2 – Slip – financial representation that must be processed; financial transaction of slip reconciliation (in this case transaction is of the Advice type) or transaction of Pre-purchase Complete (112).</p> <p>If not defined, the value is calculated automatically basing on the transaction parameters.</p>
FIMI/StandAloneReversal/Rq/ ToAccount	Str(30)	In	Destination account for the transfer or vendor account for the payment.
FIMI/StandAloneReversal/Rq/ ToAccount2	CustAcct (4000)	In	<p>Payment personal attributes or bill identifier.</p> <p>Bill identifier is defined in the repeated requests for the POS Payment Pass(175) transaction if it is the transaction to the Billing Service Provider.</p>
FIMI/StandAloneReversal/Rq/OrigTime	Time	In	Date and time of the original transaction
FIMI/StandAloneReversal/Rq/ EC3DSVersion	Int(1)	In	<p>3D Secure version.</p> <p>Available values:</p> <p>1 – 3D Secure Version 1.0 (3DS 1.0)            2 – 3D Secure Version 2.1 (3DS 2.1)            3 – 3D Secure Version 2.2 (3DS 2.2)            4 – 3D Secure Version 2.3 (3DS 2.2)</p> <p>If NULL, it is 1.</p>
FIMI/StandAloneReversal/Rq/ECI	Str(2)	In	<p>ECommerce indicator.</p> <p><i>For the detailed description and list of available values, see FIMI/POSRequest/Rq/ECI.</i></p>
FIMI/StandAloneReversal/Rq/ECTranId	Str(40)	In	<p>Transaction ID on merchant side.</p> <p><i>For the detailed description and list of available values, see FIMI/POSRequest/Rq/ECTranId.</i></p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/StandAloneReversal/Rq/ECAAV	Str(40)	In	MS UCAF or MC Static AVV. It is used in MC 3D Secure. The format of string corresponds to the description from CIS 48.43 (40 characters in Base 64).
FIMI/StandAloneReversal/Rq/CAVV	Str(40)	In	VISA CAVV <i>For the detailed description and list of available values, see FIMI/POSRequest/Rq/CAVV.</i>
FIMI/StandAloneReversal/Rq/P2PSenderData	StrUAMP (4000)	In	Information on the participants of the international transaction <i>P2P Funds Transfer</i> (132, 133, 135). For details on the field format, refer to <b>ContainerFields(E).pdf</b> .
FIMI/StandAloneReversal/Rq/CustomerAddress	Str(40)	In	Customer address. It is used in the AVS service.
FIMI/StandAloneReversal/Rq/CustomerPostalCode	Str(9)	In	Customer postal code. It is used in the AVS service.
FIMI/StandAloneReversal/Rq/DetailAddenda	Str	In	Information on the additional programs, services and other peculiarities concerning the financial transaction.  For Network = 32 – Amex, it corresponds to the message <i>AEGNS Industry Specific Detail Addenda</i> (9240/9340).  For Network != 32 – Amex, it is a string in UAMP format presented as a set of pairs <field name>=<value> separated by 0x10. Available names of fields:  For Network = 11 – VISA: - «VISA AVIATIC TICKET» – contains the data on the air ticket number
FIMI/StandAloneReversal/Rq/DetailAddendaExt	StrXML	In	Information in XML format on the additional programs, services and other peculiarities concerning the financial transaction. If defined, it overrides the data from <i>DetailAddenda</i> field.  For details on parameters, refer to XSD scheme <i>tranAddendum.xsd</i> (the files do not include the FIMI WSDL/XSD schemes).
FIMI/StandAloneReversal/Rq/AmountOrigDCC	Num	In	Original amount of the incoming external transaction.  The field is filled if the Dynamic Currency Conversion was performed during the transaction execution.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/StandAloneReversal/Rq/CurrencyOrigDCC	Int	In	Original currency of the incoming external transaction.  The field is filled if the Dynamic Currency Conversion was performed during the transaction execution.
FIMI/StandAloneReversal/Rq/ExtPSFields	StrUAMP (1000)	In	Information on the fields of ISO message for the external payment system. For details on the field format, refer to <b>ContainerFields(E).pdf</b> .
FIMI/StandAloneReversal/Rq/ExtTranAttr	StrUAMP (1000)	In	Additional transaction attributes for the external network. For the format description, refer to <b>TIP(E).pdf</b> .
FIMI/StandAloneReversal/Rq/ExtRRN	Str(32)	In	Transaction number as it was sent to the external payment system via host interface
FIMI/StandAloneReversal/Rq/WalletData	Str	In	Mastercard Wallet Program Data
FIMI/StandAloneReversal/Rq/SettleDate	Time	In	Settlement date
FIMI/StandAloneReversal/Rq/Subcode	Int(12)	In	It is used to transfer the additional subcode in the <i>POS Message to Financial Institution</i> transaction (126)
FIMI/StandAloneReversal/Rq/ExtSTAN	Int(12)	In	STAN (ISO field 11) as it is sent to the external host
FIMI/StandAloneReversal/Rq/ExtRespCode	Str(10)	In	Original response of IPS or external system received after the original transaction authorization
FIMI/StandAloneReversal/Rq/RevActualAmount	Num	In	Actual amount of the transaction being reversed with the acquiring fee.  If it=NULL or 0, it is a full reversal
FIMI/StandAloneReversal/Rq/RevActualFee	Num	In	Actual amount of acquiring fee in the transaction being reversed – can be defined for the partial reversal
FIMI/StandAloneReversal/Rq/RevActualAmountAcct	Num	In	Actual amount of the reversed transaction in the account currency.
FIMI/StandAloneReversal/Rq/RevReason	Int	In	Reversal reason code. See p.3.24 Transaction Reason Codes, section Online Reversal.  If =NULL, the value 8 (Cancel) will be used
FIMI/StandAloneReversal/Rq/ICC_TermCaps	Int	In	EMV Tag – 9F33. Terminal Capabilities (bit mask)
FIMI/StandAloneReversal/Rq/ICC_TVR	Hex(5)	In	EMV Tag – 95. Terminal Verification Results (bit mask)

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/StandAloneReversal/Rq/ICC_Random	Hex(4)	In	EMV Tag – 9f37. Unpredictable Number generated by the terminal
FIMI/StandAloneReversal/Rq/ICC_TermSN	Str(8)	In	EMV Tag – 9f1e. Serial Number
FIMI/StandAloneReversal/Rq/ICC_TermType	Int	In	EMV Tag – 9f35. Terminal Type
FIMI/StandAloneReversal/Rq/ICC_IssuerData	Hex(32)	In	EMV Tag – 9f10. Issuer Discretionary Data
FIMI/StandAloneReversal/Rq/ICC_Cryptogram	Hex(8)	In	EMV Tag – 9f26. Application Cryptogram
FIMI/StandAloneReversal/Rq/ICC_CryptInformData	Int	In	EMV Tag – 9f27. Cryptogram Information Data
FIMI/StandAloneReversal/Rq/ICC_AppTranCount	Str(8)	In	EMV Tag – 9f36. Application Transaction Counter (ATC). It is transferred in the decimal format.
FIMI/StandAloneReversal/Rq/ICC_TermTranCount	Str(8)	In	EMV Tag – 9f41. Transaction Sequence Counter
FIMI/StandAloneReversal/Rq/ICC_AppProfile	Int	In	EMV Tag – 82. Application Interchange Profile (bit mask)
FIMI/StandAloneReversal/Rq/ICC_TranType	Int	In	EMV Tag – 9c. Transaction Type
FIMI/StandAloneReversal/Rq/ICC_TermCountry	Int	In	EMV Tag – 9f1a. Terminal Country Code according to ISO 3166
FIMI/StandAloneReversal/Rq/ICC_TranDate	Str(6)	In	EMV Tag – 9a. Transaction Date (YYMMDD)
FIMI/StandAloneReversal/Rq/ICC_Amount	Str(12)	In	EMV Tag – 9f02. Amount
FIMI/StandAloneReversal/Rq/ICC_Currency	Int	In	EMV Tag – 5f2a. Currency according to ISO 4217
FIMI/StandAloneReversal/Rq/ICC_CBAmount	Str(12)	In	EMV Tag – 9f03. Amount of <i>cash back</i> transaction
FIMI/StandAloneReversal/Rq/ICC_CVMRes	Int	In	EMV Tag – 9f34. CVM Results
FIMI/StandAloneReversal/Rq/ICC_CardMember	Int	In	EMV Tag – 5f34. Card member number
FIMI/StandAloneReversal/Rq/ICC_IssuerScriptResults	Str(40)	In	EMV Tag 9f5b, Issuer Script Results
FIMI/StandAloneReversal/Rq/TransmissionTime	Time	In	Time of transmitting the Interchange message to the original transaction
FIMI/StandAloneReversal/Rq/TransmissionTime2	Time	In	Time of transmitting the Interchange message to the original transaction. It is filled for the <i>POS P2P Pass</i> transaction

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/StandAloneReversal/Rq/ ReverseType	Int(3)	In	Type of processing the transaction reversal: 400 – reversal with authorization 420 – reversal without authorization If NULL, it=420.  If ReverseType=400, TranCode = 110, 111, 113, RevReason=8(Cancel) and the operation is authorized at the China Union Pay payment gateway, the driver automatically converts TranCode to the value 125 (Purchase Cancellation).
FIMI/StandAloneReversal/Rq/ MerchantInitiatedTran	Int	In	Additional attributes on the transaction initiated by entering the card data (or their analog) from the merchant data storage.  For details, refer to the description of the <i>FIMI/POSRequest/Rq/MerchantInitiatedTran</i> field.
FIMI/StandAloneReversal/Rp/ ThisTranId	Int(12)	Out	ID of the reversal transaction sent for authorization
FIMI/StandAloneReversal/Rp/ AuthRespCode	Int	Out	Authorizer response code. For the list of available values, refer to p. 3.9.
FIMI/StandAloneReversal/Rp/ AuthRespText	Str	Out	Text field in the free format. It can be filled in by the <b>Post-processing</b> algorithmix function.
FIMI/StandAloneReversal/Rp/ ExtRespCode	Str(10)	Out	Original response of IPS or external system
FIMI/StandAloneReversal/Rp/ DeclineReason	Str	Out	Transaction declination reason. It is filled for the declined reversal. For standard checks, it is in English.
The <i>AuthRespCode</i> , <i>AuthRespText</i> , <i>ExtRespCode</i> , <i>DeclineReason</i> fields are returned for reversals with authorization (ReverseType=400).			
<b>ServiceOperationCancelation – cancelling service operation</b>			
FIMI/ServiceOperationCancelation/Rq/ ServiceId	Int(1)*	In	Service ID.  Available value: 3 – NSPK FPS
FIMI/ServiceOperationCancelation/Rq/ TranId	Int(12)	In	Transaction ID  ID of the transaction that is executed within the service and requires the cancellation.  Within the service 3 – NSPK FPS, it is possible to transfer ID of transaction 804-C2CCredit.

### 2.3.6 Reports

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b><i>GetReportList – request for reports list</i></b>			
FIMI/GetReportList/Rp/List	ArrRec	Out	<pre>{     int* ReportId;     str(300) Title;     str(200) Description;     int ReportGroupId; } [] – reports list: Report – report ID; Title – report title; Description – report description; ReportGroupId – reports group ID (see FIMI/InitSession/Rp/ReportGroups).</pre>
FIMI/GetReportList/Rp/GenerationPossibility	Int(1)*	Out	Flag indicating that the report generation units that can process FIMI requests for reports creation are available
<b><i>ExecuteReport – sending command for report execution</i></b>			
FIMI/ExecuteReport/Rq/ReportId	Int*	In	Number of report to be executed.
FIMI/ExecuteReport/Rq/Params	ArrRec	In	<pre>{     str(30)* Name;     str(250) Value;     int SeqNo } [] – parameters list SeqNo is filled when transferring the multiple values of the same parameter</pre>
FIMI/ExecuteReport/Rq/Commentary	Str(1000)	In	Comment
FIMI/ExecuteReport/Rq/FileFormat	Int(1)*	In	Report file format: 1 – PDF 2 – TXT 3 - DBF 4 - CSV 5 – HTML 6 - MHTML 7 - XLSX
FIMI/ExecuteReport/Rq/Priority	Int(1)	In	Report execution priority. Available values: 1 – low 2 – average 3 – high If =NULL, the priority is average (2).

<b>Parameter</b>	<b>Type</b> (* - mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/ExecuteReport/Rq/StartTime	Time	In	Report execution start time.
FIMI/ExecuteReport/Rp/Params	ArrRec	Out	<pre>{     str(30)* Name;     str(500) Title;     int(1)* Type;     str EditMask;     int(1)* IsMulti;     int(1)* IsMandatory }</pre> <p>} [] – a list of report parameters:</p> <p>Name – parameter name;      Title – parameter title;      Type – parameter type:          1-Num:          2-Int:          3-Str:          4-Time.      EditMask – editing mask. NULL is not restricted.      IsMulti – indicates whether the multiple values are allowed for this parameter.      IsMandatory – indicates that the field is mandatory.</p>
FIMI/ExecuteReport/Rp/CustomValuesList	ArrRec	Out	<pre>{     str(30)* Name;     str(500)* Title;     str(250)* Value }</pre> <p>} [] – a list of possible values of the random parameter:</p> <p>Name – name of parameter from the <i>FIMI/ExecuteReport/Rp/Params</i> structure;      Title – value title;      Value – value.</p>
FIMI/ExecuteReport/Rp/RequestId	Int	Out	ID of report request.
If the list of parameters to be transferred in the request is unknown (FIMI/ExecuteReport/Rq/Params), the request must be sent without this list. In this case, the response will return the list of available parameters (FIMI/ExecuteReport/Rp/Params) for this report.			
<b>GetReportsRequests – obtaining list of requests for executed/scheduled reports</b>			
FIMI/GetReportRequests/Rp/List	ArrRec	Out	<pre>{     int* RequestId;     int* ReportId;     time* RaiseTime;     int(1)* Status;     int(1)* Priority;     time StartTime;     time EndTime;     str(500) ErrorMess;     str(1000) Commentary;     int(12) ReportSize;     int(1)* FileFormat }</pre> <p>} [] – a list of report requests:</p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<p>RequestId – request ID  ReportId – report ID (NULL-report template is deleted)  RaiseTime – request time  Status – report status:  1-waits for execution  2-being executed  3-Ready  4-Execution error  Priority – execution priority:  1-low  2-average  3-high  StartTime – scheduled execution time;  EndTime – report creation time;  ErrorMess – error occurred when executing a report;  Comment – comment;  ReportSize – report size in bytes;  FileFormat – report file format:  1 - PDF  2 - TXT  3 – DBF  4 - CSV  5 - HTML;  6 - MHTML;  7 - XLSX.</p>			

**GetReport – executed report request**

FIMI/GetReport/Rq/RequestId	Int*	In	ID of report request
FIMI/GetReport/Rq/BlockSize	Int(12)*	In	Max size of requested data block (in bytes)
FIMI/GetReport/Rq/StartPosition	Int(12)*	In	Start position of data block in bytes  Initial position of the report file is 1. If StartPosition < 1, then StartPosition = 1
FIMI/GetReport/Rp/Data	Str( 9999999999 )*	Out	Report data block in Base64 encoding  From StartPosition to StartPosition+BlockSize-1
FIMI/GetReport/Rp/CRC32	Str(8)*	Out	Cyclic redundancy code

**DeleteReportRequest – deleting report/command for report execution from list of requests**

FIMI/DeleteReportRequest/Rq/ RequestId	Int*	In	ID of report request
---	------	----	----------------------

### 2.3.7 Stoplists Maintenance

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>VISASToplist – processing VISA Stoplist</b>			
FIMI/VISASToplist/Rq/PAN	Str(19)	In	Card number Define 'own' card to put on to the VISA Stoplist
FIMI/VISASToplist/Rq/MBR	Int(3)	In	Card member number If NULL, it = 0
FIMI/VISASToplist/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/VISASToplist/Rq/Action	Int(3)*	In	Add Card to Stoplist=220 Delete Card from Stoplist=221 Modify Stoplist=222 Inquiry Stoplist=223
FIMI/VISASToplist/Rq/Reason	Int	In	Card status in the VISA Stoplist: Specify the field for Action=220,222 The list of available values is given below:
Lost=3000 Stolen=3001 VIP=11 Deny=5 Referral=1 Pickup=4 Pick up Card, Special attention=7			VIP High value activity level 1 = 12 VIP High value activity level 2 = 13 VIP High value activity level 3 = 14 VIP High value activity level 4 = 15 VIP High value activity level 5 = 16 VIP High value activity level 6 = 17 VIP High value activity level 7 = 18 VIP High value activity level 8 = 19 VIP High value activity level 9 = 20
FIMI/VISASToplist/Rq/PurgeDate	Time	In	Date in TWO time zone before which the card is put on the Stoplist. Specify the field for Action=220,222
FIMI/VISASToplist/Rq/RegionsList	ArrRec	In	{ Str(1)* Code; } [] – list of the VISA regions. For the regions available codes, see p. 3.12. Specify the field for Action=220,222
FIMI/VISASToplist/Rp/CurReason	Int(4)	Out	Card current status in the VISA Stoplist. The available values are the same as for the field FIMI/VISASToplist/Rq/Reason.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/VISAStoplist/Rp/CurPurgeDate	Time	Out	Current date before which the card is put on the Visa stop list (in TWO time zone).
FIMI/VISAStoplist/Rp/CurRegionsList	ArrRec	Out	{ Str(1) Code; } [] – regions current list for the card in the Visa Stoplist. For the region available codes, see p. 3.12.
<b>MCStoplist – processing MasterCard-Europe Stoplist</b>			
FIMI/MCStoplist/Rq/PAN	Str(19)	In	Card number; Define 'own' card to put on to the Visa stop list.
FIMI/MCStoplist/Rq/MBR	In(3)	In	Card member number If NULL, it = 0
FIMI/MCStoplist/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/MCStoplist/Rq/Action	Int(3)*	In	Add Card to Stoplist=220 Add Region to Stoplist=224 Delete Card from Stoplist=221 Delete Region from Stoplist=226 Modify Stoplist=222 Modify Region in Stoplist=225 Inquiry Stoplist=223
FIMI/MCStoplist/Rq/Reason	Int(4)	In	Card status in the Mastercard-Europe Stoplist: Lost=3000 Stolen=3001 Undeliver=3002 Fraudulent=3701 Counterfeit=3003 Pickup=4 Specify this field for Action=220,224,222,225
FIMI/MCStoplist/Rq/ExpDate	Time*	In	Card expiration date

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/MCStoplist/Rq/RegionsList	ArrRec	In	{ Str(1)* Code; time* PurgeDate; } [] – list of the MasterCard Europe regions with the Purge-dates. For the region available codes, see p. 3.14. Specify the field for Action=220,224,226,222,225 PurgeDate – date in TWO time zone before which the card is put on the Stoplist in the specified region.
FIMI/MCStoplist/Rp/CurReason	Int(4)	Out	Card current card status in the MasterCard-Europe Stoplist. The available values are the same as for the field FIMI/VISASToplist/Rq/Reason.
FIMI/MCStoplist/Rp/CurRegionsList	ArrRec	Out	{ Str(1)* Code; time* PurgeDate; } [] – regions current list with the Purge-dates for the card in the MasterCard Europe Stoplist. For the region available codes, see p. 3.14.

***MCGlobalStoplist – processing Mastercard-Global Stoplist***

FIMI/MCGlobalStoplist/Rq/PAN	Str(19)	In	Card number; Define 'own' card to put on to the Mastercard-Global Stoplist.
FIMI/MCGlobalStoplist/Rq/MBR	Int(3)	In	Card member number If NULL, it = 0
FIMI/MCGlobalStoplist/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/MCGlobalStoplist/Rq/ExpDate	Time	In	Card expiration date. Define for FIMI/MCGlobalStoplist/Rq/OperCode=220, 222.
FIMI/MCGlobalStoplist/Rq/FileFormat	Int(1)*	In	File format: 0 – CIS 1 – MDS

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/MCGlobalStoplist/Rq/FileName	Str(6)*	In	File type.  For all formats: <i>MCC102 - Account File;</i> <i>MCC103 - Account Management File;</i>  MDS: <i>MCCNEG - MDS Stand-In Negative File;</i>  CIS: <i>MCC104 - Account Management File (AMS region, country, and subcountry)</i>  <i>MCC105 - Recurring Payment Cancellation Service.</i>
FIMI/MCGlobalStoplist/Rq/OperCode	Int(3)*	In	Operation code.  MCC102, MCCNEG, MCC105: 220 – Add Card to Stoplist; 221 – Delete Card from Stoplist; 222 – Modify Stoplist; 223 – Inquire Stoplist.  MCC103, MCC104 220 – Add Card to Stoplist; 221 – Delete Card from Stoplist; 223 – Inquire Stoplist.
FIMI/MCGlobalStoplist/Rq/ToDate	Time	In	Ending date of all the changes. Define the field for FIMI/MCGlobalStoplist/Rq/OperCode=220, 222.
FIMI/MCGlobalStoplist/Rq/MCCustomerID	Int(6)*	In	MC Customer ID. It is used if FIMI/MCGlobalStoplist/Rq/FileName=MCC 102, MCC103 or MCCNEG
FIMI/MCGlobalStoplist/Rq/Reason	Int(4)	In	Card status in the MasterCard-Global Stoplist. It is used with FIMI/MCGlobalStoplist/Rq/OperCode=220, 222 only.  MCC102, MCC103, MCC 104, MCCNEG: 3003 – Counterfeit; 30 – External; 3701 – Fraud;  MCC102, MCCNEG: 4 – Capture card; 3001 – Stolen card; 3000 – Lost; 34 – Unauthorized use;  MCC102, MCC103, MCC104: 31 – Credit;  MCC102: 33 – Gold; 11 – VIP; MCC103: 31 – Credit;  MCC105: 35 – Recurring payment cancellation service.

<b>Parameter</b>	<b>Type (*- mandatory parameter)</b>	<b>Direction (relative to TWO)</b>	<b>Description</b>
FIMI/MCGlobalStoplist/Rq/VIPAccumulativeLimit	Num	In	Cumulative limit of VIP card. It is used for FIMI/MCGlobalStoplist/Rq/Reason=11 only.
FIMI/MCGlobalStoplist/Rq/VIPLimitCurrency	Int	In	Limit currency of VIP card. It is used for FIMI/MCGlobalStoplist/Rq/Reason=11 only.
FIMI/MCGlobalStoplist/Rq/CardProgram	Str(3)	In	ID of the MC card program. It is used if FIMI/MCGlobalStoplist/Rq/FileName=MCC 103.
FIMI/MCGlobalStoplist/Rq/RegionsList	ArrRec	In	{ Str(1)* Code; time* PurgeDate; Int Country; int SubCountry; } [] – regions current list with the Purge-dates for the card in the MasterCard-Global Stoplist. It is used if FIMI/MCGlobalStoplist/Rq/FileName=MCC 103 or MCC104. The fields Country and SubCountry are used only for FIMI/MCGlobalStoplist/Rq/FileName=MCC 104. The codes order in the ascending order: 1, A, ..., E. For the available region codes, see p. 3.14
FIMI/MCGlobalStoplist/Rq/AcquirerID	Int(6)	In	Acquirer ID. It is used only if FIMI/MCGlobalStoplist/Rq/FileName=MCC 105
FIMI/MCGlobalStoplist/Rq/CardAcceptordId	Str(15)	In	Card Acceptor ID Code. It is used only if FIMI/MCGlobalStoplist/Rq/FileName=MCC 105
FIMI/MCGlobalStoplist/Rq/TranCurrency	Int	In	Transaction currency. It is used only if FIMI/MCGlobalStoplist/Rq/FileName=MCC 105
FIMI/MCGlobalStoplist/Rq/TranAmountLow	Num	In	Transaction min amount. It is used only if the field FIMI/MCGlobalStoplist/Rq/TranCurrency is defined.
FIMI/MCGlobalStoplist/Rq/TranAmountHigh	Num	In	Transaction max amount. It is used only if the field FIMI/MCGlobalStoplist/Rq/TranCurrency is defined.
FIMI/MCGlobalStoplist/Rp/CurReason	Int	Out	Card current status in the MasterCard-Global Stoplist. The available values are the same as for the field FIMI/MCGlobalStoplist/Rq/Reason.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/MCGlobalStoplist/Rp/CurRegionsList	ArrRec	Out	{ Str(1)* Code; time* PurgeDate; } [] – regions current list with the Purge-dates for the card in the MasterCard-Global Stoplist. For the region available codes, see p. 3.14.
FIMI/MCGlobalStoplist/Rp/ExtRespCode	Int	Out	Response code of the payment system.
<b>LocalStoplist – processing local Stoplist</b>			
FIMI/LocalStoplist/Rq/PAN	Str(19)	In	Card number; Define 'external' card to put on to the local Stoplist
FIMI/LocalStoplist/Rq/MBR	In(3)	In	Card member number If NULL, it = 0 If IsPrefix = 1, the field is ignored
FIMI/LocalStoplist/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/LocalStoplist/Rq/IsPrefix	Int(1)	In	Indicates that the card prefix (= 1) is transferred, not the card PAN. If NULL, it = 0
FIMI/LocalStoplist/Rq/Action	Int(3)*	In	Inquiry Local StopList = 410 Add Card to Local Stoplist = 411 Modify Local Stoplist = 412 Delete Card from Local Stoplist = 413
FIMI/LocalStoplist/Rq/Reason	Int	In	Card status in TWO; for the available values, see the field InitSession/Rp/CardStatList.Id or p. 3.1. The field should be defined for Action = 411, 412.
FIMI/LocalStoplist/Rq/PurgeDate	Time	In	Date before which the card is put on the local Stoplist. NULL – time during which the card is on the local Stoplist is unlimited. The field can be defined for Action=411,412.
FIMI/LocalStoplist/Rq/Response Code	Int(2)	In	Authorizer response code (for the list of available values, refer to p. 3.9). If Approved, the card is considered to be out of the stoplist. If =NULL, it is declined with the code specific for a certain host.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/LocalStoplist/Rq/ChangeReason	Str(1000)	In	Description of the reason for placing the object to the local stoplist.
FIMI/LocalStoplist/Rq/RemoveMissingRules	Int(1)	In	If =1 – delete the non-transferred rules. By default = 0
FIMI/LocalStoplist/Rq/RulesList	ArrRec	In	<pre>{     Int* SequenceNumber;     Str(4) AcquirerInstitutionName;     Int(3) AcquirerCountry;     Int(3) IssuerCountry;     Str(160) TerminalRetailerName;     Str(16) TerminalName;     Int(4) FromMCC;     Int(4) ToMCC;     Int(2) Condition;     Int(2) EntryMode;     Int Currency;     Num AmountMin;     Num AmountMax;     Int(2) ResponseCode; }</pre> <p>} [] – array of records with rules:</p> <p>SequenceNumber – rule sequence number;      AcquirerInstitutionName – name of the acquirer financial institution;      AcquirerCountry – acquirer country ISO code;      IssuerCountry – issuer country ISO code;      TerminalRetailerName – terminal retailer name;      TerminalName – terminal name;      FromMCC – first MCC. If =NULL, it =0;      ToMCC – last MCC. If =NULL, it =9999;      Condition – transaction execution condition (for the list of available values, refer to p. 3.19);      EntryMode – card PAN entry mode (for the list of available values, refer to p.3.22);      Currency – currency code;      AmountMin – transaction minimum amount;      AmountMax – transaction maximum amount;      ResponseCode – authorizer response code (for the list of available values, refer to p. 3.9). If Approved, the card is considered to be out of the stoplist. If =NULL, it is declined with the code specific for a certain code.      For all fields, except SequenceNumber, NULL = Any</p>
FIMI/LocalStoplist/Rp/CurReason	Int	Out	<p>Card current status in the local Stoplist.</p> <p>For the available values, see the field InitSession/Rp/CardStatList.Id or p. 3.2.</p>
FIMI/LocalStoplist/Rp/CurPurgeDate	Time	Out	<p>Date before which the card will be on the local Stoplist.</p> <p>NULL – time of the card presence in the local Stoplist is unlimited.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/LocalStoplist/Rp/ResponseCode	Int(2)	Out	<p>Authorizer response code (for the list of available values, refer to p. 3.9).</p> <p>If Approved, the card is considered to be out of the stoplist.</p> <p>If =NULL, it is declined with the code specific for a certain host</p>
FIMI/LocalStoplist/Rp/ChangeReason	Str(1000)	Out	Description of the reason for placing the object to the local stoplist
FIMI/LocalStoplist/Rp/RulesList	ArrRec	Out	<pre>{     Int(1)* SequenceNumber;     Str(4) AcquirerInstitutionName;     Int(3) AcquirerCountry;     Int(3) IssuerCountry;     Str(150) TerminalRetailerName;     Str(16) TerminalName;     Int(4) FromMCC;     Int(4) ToMCC;     Int(2) Condition;     Int(2) EntryMode;     Int Currency;     Num AmountMin;     Num AmountMax;     Int(2) ResponseCode; }</pre> <p>} [] – array of records with rules.</p> <p>For the description of fields, see above.</p>

#### ***AmExStoplist – processing American Express stoplist***

FIMI/AmExStoplist/Rq/PAN	Str(19)	In	Card number; Define 'own' card to put on to the American Express stoplist.
FIMI/AmExStoplist/Rq/MBR	Int(3)	In	Card member number If NULL, it = 0
FIMI/AmExStoplist/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AmExStoplist/Rq/Action	Int*	In	Add Card to Stoplist=220 Delete Card from Stoplist=221 Modify Stoplist=222

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AmExStoplist/Rq/Reason	Int(4)	In	Status to be set to card in the American Express stoplist:  Lost=3000 Stolen=3001 Undeliver=3002 Fraudulent=3701 Pickup=4 VIP=11 Other=30  The field is mandatory for Action=220,222
FIMI/AmExStoplist/Rq/PurgeDate	Time	In	Date in <b>TWO</b> time zone (accurate to within a day) before which the card will be on the stoplist.  The field is mandatory for Action=220,222
<b><i>DiscoverStoplist – processing Discover stoplist</i></b>			
FIMI/DiscoverStoplist/Rq/PAN	Str(19)	In	Card number;  Define 'own' card to put on to the Discover stoplist.
FIMI/DiscoverStoplist/Rq/MBR	Int(3)	In	Card member number  If NULL, it = 0
FIMI/DiscoverStoplist/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/DiscoverStoplist/Rq/ExpirationDate	Time	In	Card Expiration Date.  This field is mandatory for Action=220, 222
FIMI/DiscoverStoplist/Rq/Action	Int(3)*	In	Add Card to Stoplist=220  Modify Stoplist=222  Delete Card from Stoplist=221
FIMI/DiscoverStoplist/Rq/Reason	Int(4)	In	Status to be set to card in the Discover stoplist:  Lost/Stolen=3000 Fraud=3701 Bankruptcy=10 Closed=8 Revoked=9 Frozen=6 Interest Prohibited=36 Charged Off=3 Unauthorized=34

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			VIP=11 The field is mandatory for Action =220, 222
FIMI/DiscoverStoplist/Rq/CardAction	Int(1)	In	Card operations: Decline=1 Pick Up Card=2 Approve=3 The field is mandatory for Action =220, 222
FIMI/DiscoverStoplist/Rq/PurgeDate	Time	In	Date in TWO timezone (accurate to within a day) before which the card will be on the stoplist. The field is mandatory for Action =220, 222

### 2.3.8 Visa Administrative Transactions

Parameter	Type (* mandatory parameter)	Direction (relative to TWO)	Description
<b>AdjustToVISA – adjustment of Visa acquiring SMS transaction</b>			
FIMI/AdjustToVISA/Rq/OrigTranId	Int(12)*	In	Original transaction ID. It is the SMS transaction, any financial transaction can be used as an original transaction for adjustment – Network=11(VISA), Type=200.
FIMI/AdjustToVISA/Rq/TranCode	Int(3)*	In	Code of the adjustment transaction; Available values: 201 – Debit Adjustment 202 – Credit Adjustment
FIMI/AdjustToVISA/Rq/Reason	Int*	In	Reason of the adjustment transaction <i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PSType = 1 (VISA) and bit 1(Adjustment) in the Usage field.</i>
FIMI/AdjustToVISA/Rq/Amount	Num*	In	Amount of the adjustment transaction
FIMI/AdjustToVISA/Rq/Fee	Num*	In	Acquiring fee
FIMI/AdjustToVISA/Rq/OrigId	Int*	In	Transaction originator ID (terminal ID) <i>This and the following fields of the request take their initial values from the original transaction but they can be changed manually.</i>
FIMI/AdjustToVISA/Rq/PAN	Str(19)	In	Card number “External” card on which SMS-transaction was conducted in the “own” acquiring network
FIMI/AdjustToVISA/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/AdjustToVISA/Rq/ExtRRN	Str(32)*	In	Original transaction RRN
FIMI/AdjustToVISA/Rq/ExtSTAN	Int*	In	Original transaction STAN
FIMI/AdjustToVISA/Rq/ExtTranAttr	Str(1000)	In	Extra attributes of the original transaction
FIMI/AdjustToVISA/Rq/HostNetID	Str(4)*	In	Host network ID. Available values: ‘0002’ – VISA, ‘0004’ – PLUS
FIMI/AdjustToVISA/Rq/Currency	Int*	In	Currency for the fields Amount and Fee
FIMI/AdjustToVISA/Rq/FromAcctType	Int(2)*	In	Source account type: 0 - None; 1=checking; 11=savings; 31=credit
FIMI/AdjustToVISA/Rq/ToAcctType	Int(2)*	In	Destination account type: 0 - None; 1=checking; 11=savings; 31=credit

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description																		
<b>RepresentToVISA – repeated representment of Visa acquiring SMS-transaction</b>																					
FIMI/RepresentToVISA/Rq/ChargebackTranId	Int(12)*	In	Chargeback transaction ID: – it should be the IssuerRequest (422) transaction																		
FIMI/RepresentToVISA/Rq/HostNetID	Str(4)*	In	Host network ID. Available values: '0002' – VISA, '0004' – PLUS <i>Initial value to be edited can be obtained from the Chargeback transaction</i>																		
FIMI/RepresentToVISA/Rq/Amount	Num*	In	Original transaction amount <i>Initial value to be edited can be obtained from the Chargeback transaction</i>																		
FIMI/RepresentToVISA/Rq/Fee	Num*	In	Original transaction fee <i>Initial value to be edited can be obtained from the Chargeback transaction</i>																		
FIMI/RepresentToVISA/Rq/Currency	Int*	In	Currency for the fields Amount, Fee <i>Initial value to be edited can be obtained from the Chargeback transaction</i>																		
FIMI/RepresentToVISA/Rq/Reason	Int*	In	Representment reason; <i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PSType = 1 (VISA) and bit 4(Represent) in the Usage field.</i>																		
FIMI/RepresentToVISA/Rq/TextMess	Str(255)	In	Text message in the format: <table border="1"> <thead> <tr> <th>Position</th> <th>Length</th> <th>Subfield Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>'V' char – Field Identifier</td> </tr> <tr> <td>2</td> <td>1</td> <td>Usage Code: '2' – First representment; '3' – Second representment</td> </tr> <tr> <td>3</td> <td>1</td> <td>Document Indicator: '-' – Not applicable; '0' – No documentation provided '1' – Mailed documentation to follow '2' – Invalid acquirer's reference number used in prior record, no supporting documentation was required or received '3' – Invalid acquirer's reference number used in prior record and supporting documentation was received '4' – No supporting documentation received for prior chargeback</td> </tr> <tr> <td>4</td> <td>6</td> <td>Chargeback reference number, '000000' by default</td> </tr> <tr> <td>10</td> <td>...</td> <td>Message text</td> </tr> </tbody> </table>	Position	Length	Subfield Description	1	1	'V' char – Field Identifier	2	1	Usage Code: '2' – First representment; '3' – Second representment	3	1	Document Indicator: '-' – Not applicable; '0' – No documentation provided '1' – Mailed documentation to follow '2' – Invalid acquirer's reference number used in prior record, no supporting documentation was required or received '3' – Invalid acquirer's reference number used in prior record and supporting documentation was received '4' – No supporting documentation received for prior chargeback	4	6	Chargeback reference number, '000000' by default	10	...	Message text
Position	Length	Subfield Description																			
1	1	'V' char – Field Identifier																			
2	1	Usage Code: '2' – First representment; '3' – Second representment																			
3	1	Document Indicator: '-' – Not applicable; '0' – No documentation provided '1' – Mailed documentation to follow '2' – Invalid acquirer's reference number used in prior record, no supporting documentation was required or received '3' – Invalid acquirer's reference number used in prior record and supporting documentation was received '4' – No supporting documentation received for prior chargeback																			
4	6	Chargeback reference number, '000000' by default																			
10	...	Message text																			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b>ConfirmCopyRequestToVISA – confirmation of sending receipt copy on issuer request</b>			
FIMI/ConfirmCopyRequestToVISA/Rq/OrigTranId	Int(12)*	In	ID of the Copy Request(210) original transaction.
FIMI/ConfirmCopyRequestToVISA/Rq/TextMess	Str(255)	In	Text message
<b>FeeToVISA – request for fees from Visa or cash back to Visa</b>			
FIMI/FeeToVISA/Rq/HostNetID	Str(4)*	In	Host network ID, available values: '0002' – VISA, '0004' – PLUS
FIMI/FeeToVISA/Rq/TranCode	Int(3)*	In	205 = Fee Collection 206 = Funds Disbursement
FIMI/FeeToVISA/Rq/Reason	Int*	In	Outbound Fee reason; <i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PSType = 1 (VISA) and bit 5(Fee Collectio /Funds Disbursement) in the Usage field.</i>
FIMI/FeeToVISA/Rq/Amount	Num*	In	Transaction amount – fee value transmitted to the payment system
FIMI/FeeToVISA/Rq/Fee	Num*	In	Transaction acquiring fee
FIMI/FeeToVISA/Rq/Currency	Int*	In	Currency for the fields Amount, Fee
FIMI/FeeToVISA/Rq/AcqInstID	Str(11)	In	Acquiring institution BIN
FIMI/FeeToVISA/Rq/IssInstID	Str(11)*	In	Issuing institution BIN
FIMI/FeeToVISA/Rq/PAN	Str(19)	In	Fee related card
FIMI/FeeToVISA/Rq/TextMess	Str(255)	In	Text message
FIMI/FeeToVISA/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
<b>TextToVISA – sending text message to Visa</b>			
FIMI/TextToVISA/Rq/HostNetID	Str(4)*	In	Host network ID, available values: '0002' – VISA, '0004' – PLUS
FIMI/TextToVISA/Rq/SrcInstID	Str(11)	In	Source institution BIN
FIMI/TextToVISA/Rq/DestInstID	Str(11)*	In	Destination institution BIN
FIMI/TextToVISA/Rq/PAN	Str(19)	In	Message related card
FIMI/TextToVISA/Rq/TextMess	Str(255)	In	Text message
FIMI/TextToVISA/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.

### 2.3.9 MasterCard Global Administrative Transactions

Parameter	Type (* mandatory parameter)	Direction (relative to TWO)	Description
<b>AdjustToMC – adjustment of Mastercard Global acquiring MDS-transaction</b>			
FIMI/AdjustToMC/Rq/OrigTranId	Int(12)*	In	<p>Original transaction ID.</p> <p>It is the MDS transaction, any financial transaction can be used as an original transaction for adjustment – Network=22(MC Global), Type=200.</p>
FIMI/AdjustToMC/Rq/TranCode	Int*	In	<p>Adjustment transaction code.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>201 – Debit Adjustment</li> <li>202 – Credit Adjustment</li> </ul>
FIMI/AdjustToMC/Rq/Reason	Int(3)*	In	<p>Adjustment transaction reason.</p> <p><i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PType = 2 (MasterCard) and bit 1(Adjustment) in the Usage field.</i></p>
FIMI/AdjustToMC/Rq/Amount	Num*	In	Adjustment amount
FIMI/AdjustToVISA/Rq/HostNetID	Str(2)*	In	<p>Host network identifier.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>MC -- MasterCard</li> <li>CI -- Cirrus</li> <li>MS -- Maestro</li> <li>MD -- MasterCard debit card</li> <li>PL -- Plus</li> <li>VI -- VISA</li> </ul>
FIMI/AdjustToMC/Rq/Currency	Int*	In	Adjustment transaction currency
FIMI/AdjustToMC/Rq/TextMess	Str(255)	In	Text message
<b>RepresentToMC – repeated representment of Mastercard Global acquiring MDS-transaction</b>			
FIMI/RepresentToMC/Rq/ChargebackTranId	Int(12)*	In	<p>Chargeback transaction ID:</p> <p>– the transaction must be of the IssuerRequest type (Type=422)</p>
FIMI/RepresentToMC/Rq/HostNetID	Str(2)*	In	<p>Host network identifier.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>MC -- MasterCard</li> <li>CI -- Cirrus</li> <li>MS -- Maestro</li> <li>MD -- MasterCard debit card</li> <li>PL -- Plus</li> <li>VI -- VISA</li> </ul> <p><i>Initial value to be edited can be obtained from the Chargeback transaction</i></p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/RepresentToMC/Rq/Amount	Num*	In	Transaction amount <i>Initial value to be edited can be obtained from the Chargeback transaction</i>
FIMI/RepresentToMC/Rq/Currency	Int*	In	Transaction currency <i>Initial value to be edited can be obtained from the Chargeback transaction</i>
FIMI/RepresentToMC/Rq/Reason	Int*	In	Representment reason. <i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PSType = 2 (MasterCard) and bit 4(Representment) in the Usage field.</i>
FIMI/RepresentToMC/Rq/TextMess	Str(255)	In	Text message
<b>ChargebackToMC – issuer reversal of Mastercard Global MDS-transaction</b>			
FIMI/ChargebackToMC/Rq/OrigTranId	Int(12)*	In	ID of original financial transaction (Type=200)
FIMI/ChargebackToMC/Rq/HostNetID	Str(2)*	In	Host network identifier. Available values: MC -- MasterCard CI -- Cirrus MS -- Maestro MD -- MasterCard debit card PL -- Plus VI -- VISA <i>Initial value to be edited can be obtained from the Chargeback transaction</i>
FIMI/ChargebackToMC/Rq/Amount	Num*	In	Transaction amount <i>Initial value to be edited can be obtained from the Chargeback transaction</i>
FIMI/ChargebackToMC/Rq/Reason	Int*	In	Chargeback reason. <i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PSType = 2 (MasterCard) and bit 2(Chargeback) in the Usage field.</i>
FIMI/ChargebackToMC/Rq/TextMess	Str(255)	In	Text message
<b>ChargebackReversalToMC – reversal of issuer Chargeback of MasterCard Global MDS transaction</b>			
FIMI/ChargebackReversalToMC/Rq/ChargebackTranId	Int(12)*	In	ID of chargeback transaction with the IssuerRequest type (Type=422)

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/ChargebackReversalToMC/Rq/ HostNetID	Str(2)*	In	<p>Host network identifier.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>MC -- MasterCard</li> <li>CI -- Cirrus</li> <li>MS -- Maestro</li> <li>MD -- MasterCard debit card</li> <li>PL -- Plus</li> <li>VI -- VISA</li> </ul> <p><i>Initial value to be edited can be obtained from the Chargeback transaction</i></p>
FIMI/ChargebackReversalToMC/Rq/ Amount	Num*	In	<p>Original transaction amount</p> <p><i>Initial value to be edited can be obtained from the Chargeback transaction</i></p>
FIMI/ChargebackReversalToMC/Rq/ Reason	Int*	In	<p>Chargeback reversal reason.</p> <p><i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PSType = 2 (MasterCard) and bit 3(Chargeback Reversal) in the Usage field.</i></p>
FIMI/ChargebackReversalToMC/Rq/ TextMess	Str(255)	In	Text message

### 2.3.10 Administrative Dispute Transactions in the Local Payment System

Parameter Name	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b><i>LocalChargeback – issuer reversal of Request transaction</i></b>			
FIMI/LocalChargeback/Rq/OrigTranId	Int(12)*	In	ID of original finance transaction (Type=200)
FIMI/LocalChargeback/Rq/Amount	Num*	In	Transaction amount <i>Initial value to be edited can be obtained from the original transaction</i>
FIMI/LocalChargeback/Rq/Reason	Int*	In	Chargeback reason <i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PType = 3 (Local) and bit 2(Chargeback) in the Usage field.</i>
FIMI/LocalChargeback/Rq/TextMess	Str(255)	In	Text Message
<b><i>LocalChargebackReversal – Chargeback Reversal</i></b>			
FIMI/LocalChargebackReversal/Rq/ChargebackTranId	Int(12)*	In	ID of Chargeback transaction of the IssuerRequest type (Type=422)
FIMI/LocalChargebackReversal/Rq/Amount	Num*	In	Original transaction amount <i>Initial value to be edited can be obtained from the original transaction</i>
FIMI/LocalChargebackReversal/Rq/Reason	Int*	In	Chargeback reversal reason <i>See the list of available values in the InitSession/Rp/VISATranReasonList field with PType = 3 (Local) and bit 3(Chargeback Reversal) in the Usage field.</i>
FIMI/LocalChargebackReversal/Rq/TextMess	Str(255)	In	Text Message
<b><i>LocalAdjustment – adjustment of acquirer transaction</i></b>			
FIMI/LocalAdjustment/Rq/OrigTranId	Int(12)*	In	Original transaction identifier
FIMI/LocalAdjustment/Rq/TranCode	Int(3)*	In	Adjustment transaction code Available values: 201 – Debit Adjustment 202 – Credit Adjustment
FIMI/LocalAdjustment/Rq/Reason	Int*	In	Adjustment transaction reason <i>For the list of available values, see the InitSession/Rp/VISATranReasonList field with PType = 3 (Local) and bit 1(Adjustment) in the Usage field.</i>
FIMI/LocalAdjustment/Rq/Amount	Num*	In	Adjustment transaction amount
FIMI/LocalAdjustment/Rq/Fee	Num*	In	Acquirer fee

Parameter Name	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/LocalAdjustment/Rq/Currency	Int*	In	ISO code of the currency for the Amount and Fee fields
FIMI/LocalAdjustment/Rq/PAN	Str(19)	In	Card number
FIMI/LocalAdjustment/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.

### 2.3.11 System Units

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetUnitState – request for TWO units and their statuses</b>			
FIMI/GetUnitState/Rq/UnitId	Int	In	Unit ID
FIMI/GetUnitState/Rq/Instanceld	Int[]	In	<p>Array of the instance identifiers. If not defined, the list of units of all the instances belonging to the system will return in request.</p> <p><i>for the list of available values, see the field InitSession/Rp/Instances</i></p>
FIMI/GetUnitState/Rq/UnitType	ArrRec	In	<p>{</p> <p>    Int* Type;     int SubType;</p> <p>} [] – array of structures, list of the unit types to be returned; If not defined, list of all the instance units will be returned in the request.</p> <p>    Type – unit type (see p. 3.17.1).</p> <p>    NULL – all the unit types are requested;</p> <p>    SubType – unit subtype (see p. 3.17.1).                 it is effective together with the Type field only;</p>
FIMI/GetUnitState/Rp/Units	ArrRec	Out	<p>{</p> <p>    int* UnitId;     int* Instanceld;     int* Type;     int SubType;     str* Description;     int State;</p> <p>} [] – array of structures, list of units and their states:</p> <p>    UnitId – unit identifier;     Instanceld – instance identifier;     Type – unit type (see p. 3.18.1);     SubType – unit subtype (see p. 3.18.1);</p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
			Description – unit description; State – unit state. It depends on the unit type (see p. 3.18.2);
FIMI/GetUnitState/Rp/SubUnits	ArrRec	Out	<pre>{ int* UnitId; int* SubUnitId; str(240) Title; Time LastExecuted; } [] – list of sub-units. It is returned if Units has units with Type = 45 (see p.3.18.1) and contains the list of jobs. UnitId – unit ID matches UnitId of the <i>Units</i> structure. SubUnitId – sub-unit ID Title – sub-unit name LastExecuted – time of the sub-unit last start</pre>

#### ***UnitCommand – sending command to unit***

FIMI/UnitCommand/Rq/UnitId	Int	In	Unit ID. The field value is ignored if the ArrayUnitId field with the list of the unit identifiers is transferred.
FIMI/UnitCommand/Rq/ArrayUnitId	Int[]	In	List of the unit identifiers
FIMI/UnitCommand/Rq/Command	Int*	In	Command ID (see p. 3.18.3)
FIMI/UnitCommand/Rq/Param1	Int(12)	In	First parameter of the command
FIMI/UnitCommand/Rq/Param2	Int	In	Second parameter of the command

For the operation execution, the UnitId field with the unit identifier or ArrayUnitId field with the list of the unit identifiers must be transferred, otherwise, the operation will be declined.

#### ***GetUnitEvents – request for unit events***

FIMI/GetUnitEvents/Rq/UnitId	Int	In	Unit ID
FIMI/GetUnitEvents/Rq/TermId	Int	In	Terminal ID
FIMI/GetUnitEvents/Rq/HostId	Int	In	Host ID
FIMI/GetUnitEvents/Rq/LastId	Int	In	ID of the last event received by the unit/terminal; if defined in the request, all the events which occurred later and satisfy the other conditions are returned in the response (with Id>LastId)

<b>Parameter</b>	<b>Type</b> (* - mandatory parameter)	<b>Direction</b> (relative to TWO)	<b>Description</b>
FIMI/GetUnitEvents/Rq/Severity	Int(1)	In	Severity starting from which events are requested (NULL – all): 1 – Event 2 – Warning 3 – Error 4 – Alarm
FIMI/GetUnitEvents/Rq/FromTime	Time	In	Event occurrence start time
FIMI/GetUnitEvents/Rq/ToTime	Time	In	Event occurrence end time
FIMI/GetUnitEvents/Rq/Count	Int(4)	In	Number of the requested events (NULL – all found under the defined conditions) but not more than 1000
FIMI/GetUnitEvents/Rq/EventList	ArrRec	Out	{ Int(1)* Severity; time* RaiseTime; str(500) Mess; str(750) Cause; str(750) Remedy; str(750) Comment; int* EventId; str(250) UnitName; int(12) TranId; str(16) TermName; } [] – list of events sorted by the time in the descending order
<b><i>GetIAConfiguration – request for configuration of TIC host interface connected to Interface Application</i></b>			
FIMI/GetIAConfiguration/Rq/Id	Int*	In	ID of TIC host interface connected to IA.
FIMI/GetIAConfiguration/Rp/ExternalPortAddress	Str(150)	Out	Address of the external IA port
FIMI/GetIAConfiguration/Rp/ExternalPortIsServer	Int(1)	Out	Attribute indicating that the external IA port is server port
FIMI/GetIAConfiguration/Rp/ExternalPortFrameType	Int	Out	Type of the external IA port frame: 0 – <####> 1 – X25: <####>; Others: STX L:ASCII[6] <####> ETX 2 – STX L:ASCII[6] <####> ETX 3 – STX <####> ETX 4 – <####> ETX [CR] 5 – L1 L2 <####> 6 – L2 L1 <####> 7 – L2 L1 0 0 <####> 8 – STX L:BCD[2] <####> ETX LRC(L-ETX) 9 – L2 L1 0 0 0 0 <####> 10 – STX L2 L1 <####> LRC(####) ETX 11 – 0xFF L:ASCII[6] <####> 12 – L4 L3 L2 L1 <####>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description												
			13 – L2 L1 0 <####> 14 – < #### > 15 – L:ASCII[4] <####> 16 – STX <####> ETX LRC(<####>,ETX) 17 – L:ASCII[6] <####> 18 – 0xD7 0x4A L2 L1 0x0M 0x00 <####> 19 – IncLen L1 L2 <####> 20 – IncLen L2 L1 <####> 21 – IncLen L4 L3 L2 L1 <####> 22 – 0xFF L:ASCII[5] <####> 23 – <####> CR 24 – L:ASCII[4] <####> ETX 25 – IncLen L2 L1 <####> ETX 26 – L2 L1 <####> ETX 27 – (Len+1) L2 L1 <####> ETX 28 – (Len-4) L4 L3 L2 L1 <####> 29 – L:BCD2 <####> 30 – L2 L1 'SCA\$ISO' 0x1C <####> 31 – IncLen L:ASCII[4] <####>												
FIMI/GetIAConfiguration/Rp/ ExternalPortKeepAlive	Int(1)	Out	TCP option of IA external port												
FIMI/GetIAConfiguration/Rp/ ExternalPortTxTimeout	Num(6,3)	Out	Transfer timeout of IA external port (seconds)												
FIMI/GetIAConfiguration/Rp/ ExternalPortRxTimeout	Num(6,3)	Out	Receipt timeout of IA external port (seconds)												
FIMI/GetIAConfiguration/Rp/ ExternalPortUseSSL	Int	Out	Attribute indicating whether or not SSL is used by the external IA port: = 0 – not used. > 0 – used if the value is a bit mask of available SSL versions: <table border="1"> <thead> <tr> <th>Bit Number</th> <th>SSL Version</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>SSL3</td> </tr> <tr> <td>3</td> <td>TLS1</td> </tr> <tr> <td>4</td> <td>TLS1.1</td> </tr> <tr> <td>5</td> <td>TLS1.2</td> </tr> <tr> <td>6</td> <td>TLS1.3</td> </tr> </tbody> </table>	Bit Number	SSL Version	2	SSL3	3	TLS1	4	TLS1.1	5	TLS1.2	6	TLS1.3
Bit Number	SSL Version														
2	SSL3														
3	TLS1														
4	TLS1.1														
5	TLS1.2														
6	TLS1.3														
FIMI/GetIAConfiguration/Rp/ ExternalPortSSLAAuthRequired	Int	Out	Attribute indicating that the customer authentication is required in case of the SSL connection with the external IA port.												
FIMI/GetIAConfiguration/Rp/ ExternalPortSSLPrivateKeyAndCertificate	Str	Out	Private key and certificate (or chain of certificates) in the PKCS12 format with the password in the Base64 encoding for the external IA port. The FIMI operator hash is used as a password for the private key encryption.												
FIMI/GetIAConfiguration/Rp/ ExternalPortSSLCACertificates	Str	Out	List of the CA certificates in the PKCS7 format for the external IA port, it is encoded in PEM.												

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetIAConfiguration/Rp/InternalPortAddress	Str(150)	Out	Address of the internal IA port
FIMI/GetIAConfiguration/Rp/InternalPortIsServer	Int(1)	Out	Attribute indicating that the internal IA port is server port
FIMI/GetIAConfiguration/Rp/InternalPortFrameType	Int	Out	Frame type of the internal IA port (see a list of available values in the field FIMI/GetIAConfiguration/Rp/ExternalPortFrameType)
FIMI/GetIAConfiguration/Rp/InternalPortKeepAlive	Int(1)	Out	TCP option of the internal IA port
FIMI/GetIAConfiguration/Rp/InternalPortTxTimeout	Num(6,3)	Out	Transfer timeout of IA internal port (seconds)
FIMI/GetIAConfiguration/Rp/InternalPortRxTimeout	Num(6,3)	Out	Receipt timeout of the internal IA port (seconds)
FIMI/GetIAConfiguration/Rp/InternalPortUseSSL	Int	Out	Attribute indicating whether or not SSL is used by the internal IA port (see the description of the ExternalPortUseSSL field)
FIMI/GetIAConfiguration/Rp/InternalPortSSLAAuthRequired	Int	Out	Attribute indicating that the customer authentication is required in case of the SSL connection with the internal IA port.
FIMI/GetIAConfiguration/Rp/InternalPortSSLPrivateKeyAndCertificate	Str	Out	Private key and certificate (or chain of certificates) in the PKCS12 format with the password in the Base64 encoding for the internal IA port. The FIMI operator hash is used as a password for the private key encryption.
FIMI/GetIAConfiguration/Rp/InternalPortSSLCACertificates	Str	Out	List of the CA certificates in the PKCS7 format for the internal IA port, it is encoded in PEM.
FIMI/GetIAConfiguration/Rp/Configuration	ArrRec	Out	{           Str(100)* Parameter;           Str Value;         } [] – Array of IA configuration parameters: Parameter Value
<b>SetIAConfiguration – request for changing configuration of TIC host interface connected to Interface Application</b>			
FIMI/SetIAConfiguration/Rq/Id	Int*	In	ID of TIC host interface connected to IA
FIMI/SetIAConfiguration/Rq/Configuration	ArrRec	In	{           Str(100)* Parameter;           Str Value;         } [] – Array of IA configuration parameters: Parameter Value
FIMI/SetIAConfiguration/Rq/RemoveMissingFields	Int(1)	In	1 – Delete the transferred fields. The default value is 0

### 2.3.12 Cryptographic Operations

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetWorkingKey – generating working key</b>			
FIMI/GetWorkingKey/Rq/ KeyType	Int(1)	In	<p>Working key type.</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>1 – TPK</li> <li>2 - RSA Sign Key Pair (RSA-SK)</li> </ul> <p>If NULL, it is 1.</p>
FIMI/GetWorkingKey/Rq/ PINEntryDevice	Str(16)	In	<p>Name of the PIN entry device. If not defined, the working key will be encrypted with the FIMI terminal master key (used in POSRequest). If defined, master key of the PIN entry device will apply.</p> <p>It is used for KeyType = 1(TPK).</p>
FIMI/GetWorkingKey/Rp/KeyId	Int*	Out	Working key identifier
FIMI/GetWorkingKey/Rp/Key	Str(258)*	Out	<p>Working key.</p> <p>The format for transferring the key and its allowed length depend on the cryptoserver settings:</p> <ul style="list-style-type: none"> <li>• For the DES algorithm, the allowed length is 16, the key is transferred in the hexadecimal format.</li> <li>• For the Triple DES algorithm, the allowed length is 32, the key is transferred in the hexadecimal format.</li> <li>• For the key in the GISKE format, the allowed length is 120, the key is transferred in the GISKE block format.</li> <li>• For the key in the KeyBlock Thales format, the allowed length is 258, the key is transferred in the hexadecimal format.</li> </ul> <p>It is used for KeyType = 1(TPK).</p>
FIMI/GetWorkingKey/Rp/PublicKey	Str	Out	<p>Module and exponent of the working RSA key in the DER ASN.1 format (for details, see PKCS#1 (RFC 8017)), it is transferred in the Base64 encoding.</p> <p>It is returned for KeyType = 2 (RSA Sign Key Pair).</p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b>GetPVV_PINOffset – generating PVV/Internet PVV or PIN offset/Internet PIN offset</b>			
FIMI/GetPVV_PINOffset/Rq/KeyId	Int	In	Working key ID (obtained using the <i>GetWorkingKey</i> operation)  If not defined, it is considered that received PIN blocks are under cryptoserver LMK
FIMI/GetPVV_PINOffset/Rq/ZPK	Str	In	ZPK key encrypted by the public key of the working key of the RSA type, it is transferred in the Base64 encoding.  It is transferred and mandatory if the type of the working key is RSA Sign Key Pair.
FIMI/GetPVV_PINOffset/Rq/IsInternet	Int(1)	In	If defined and it = 1, Internet PVV/Internet Pin Offset will be generated.  If not defined or it = 0, PVV/ PIN Offset will be generated.
FIMI/GetPVV_PINOffset/Rq/IsPVV	Int(1)*	In	Available values:  1 – PVV 0 – PIN Offset -1 – define by prefix
FIMI/GetPVV_PINOffset/Rq/OldPINBlock	Hex(33)	In	Old PIN block
<p>The PIN block is generated in the format specified in PINBlockFormat and encrypted by the working key or cryptoserver LMK.</p> <p>If the working key was generated with the TPK(1) type, the PIN block is encrypted by the clear value of the working key (the cryptogram obtained in the <i>GetWorkingKey</i> operation must be decrypted in advance using the TMK key).</p> <p>If the working key was generated with the RSA Sign Key Pair (2) type, the PIN block is encrypted by a random ZPK key and ZPK key is encrypted by the public key of the working key and encrypted key is transferred in the request in the ZPK key.</p> <p>If the PIN block key is transferred under the AES LMK key, the encrypted PIN block must be preceded with the 'J' character, the field value is transferred in the HEX format and its length is 66 characters.</p>			
FIMI/GetPVV/Rq/OldPVV PINOffset	Int(12)	In	Old PVV/PIN Offset (Internet PVV/Internet PIN Offset). For PVV, the value length should be equal to 4. For PIN Offset, the value length should be equal to 12.
FIMI/GetPVV/Rq/PAN	Str(19)	In	Card number
FIMI/GetPVV/Rq/MBR	Int(3)	In	Card member number. If NULL, it = 0.
FIMI/GetPVV_PINOffset/Rq/ CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetPVV_PINOffset/Rq/ PINBlockFormat	Int(1)	In	PIN block format in the NewPINBlock and OldPINBlock fields. Available values: 0 or NULL – ANSI X9.8 (ISO Format 0); 1 – ANSI X9.8 (ISO Format 1).
FIMI/GetPVV_PINOffset/Rq/ NewPINBlock	Hex(33)*	In	New PIN block

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
The PIN block is generated in the format specified in PINBlockFormat and encrypted by the working key or cryptoserver LMK.			
If the working key was generated with the TPK(1) type, the PIN block is encrypted by the clear value of the working key (the cryptogram obtained in the <i>GetWorkingKey</i> operation must be decrypted in advance using the TMK key).			
If the working key was generated with the RSA Sign Key Pair (2) type, the PIN block is encrypted by a random ZPK key and ZPK key is encrypted by the public key of the working key and encrypted key is transferred in the request in the ZPK key.			
If the PIN block key is transferred under the AES LMK key, the encrypted PIN block must be preceded with the 'J' character, the field value is transferred in the HEX format and its length is 66 characters.			
FIMI/GetPVV_PINOffset/Rp/ NewPVV_PINOffset	Int(12)*	Out	<p>New PVV/PIN Offset (Internet PVV/Internet PIN Offset).</p> <p>For PVV (FIMI/GetPVV_PINOffset/Rp/IsPVV=1), the value has the [PVKI][PVV] format (contains PVKI in the first leftmost character) and fixed length of 5 characters.</p> <p>For PIN Offset (FIMI/GetPVV_PINOffset/Rp/IsPVV=0), PIN Offset is transferred with the variable length of 12 characters at the most.</p>
FIMI/GetPVV_PINOffset/Rp/IsPVV	Int(1)*	Out	<p>PVV type.</p> <p>Available values:</p> <p>1 – PVV</p> <p>0 – PIN Offset</p>
FIMI/GetPVV_PINOffset/Rp/ NewPINBlock	Hex(16)	Out	New PIN block under ZPK to be stored in Back-Office. It is used when reissuing a card with the same PIN. It is defined in the format ANSI X9.8 (ISO Format 0).
To set the generated PVV to card, use the <i>SetCardPVV</i> request. To set PVV/IPVV, transfer only PVV (i.e. without PVKI that is returned in the <i>FIMI/GetPVV_PINOffset/Rp/NewPVV_PINOffset</i> field) in the <i>SetCardPVV</i> request.			
<b>GetCVV – generating CVV or CVV2</b>			
FIMI/GetCVV/Rq/PAN	Str(19)	In	Card number
FIMI/GetCVV/Rq/MBR	Int(3)	In	Card member number
FIMI/GetCVV/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetCVV/Rq/IsCVV2	Int(1)	In	<p>If defined and equal to 1, Internet CVV2 will be generated.</p> <p>If not defined or equal to 0, CVV will be generated.</p>
FIMI/GetCVV/Rq/ExpDate	Str(4)	In	<p>Card expiration date (YYMM)</p> <p>If not defined, FIMI terminal will determine the card expiration date by itself.</p>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetCVV/Rp/CSCLen	Int(1)	In	CVV length. Available values: 3,4,5. If NULL, it is 3. It is used if the CSC card verification type is set for the card prefix.
FIMI/GetCVV/Rq/CardVerificationType	Int(1)	In	Card verification type Available values: -1 – identify by the card prefix 1 – CVV 2 – CSC If NULL, it is 1.
FIMI/GetCVV/Rp/CVV	Int(5)*	Out	CVV or CVV2
FIMI/GetCVV/Rp/StrCVV	Str(5)*	Out	CVV or CVV2 in the string format
FIMI/GetCVV/Rp/CardVerificationType	Int(1)	Out	Card verification type Available values: 1 – CVV 2 – CSC
<b>Check PVV_PINOffset – PVV/Internet PVV or PIN Offset/Internet PIN Offset check</b>			
FIMI/CheckPVV_PINOffset/Rp/KeyId	Int*	In	Working key ID (obtained using the <i>GetWorkingKey</i> operation)
FIMI/CheckPVV_PINOffset/Rq/ZPK	Str	In	ZPK key encrypted by the public key of the working key of the RSA type, it is transferred in the Base64 encoding. It is transferred and mandatory if the type of the working key is RSA Sign Key Pair.
FIMI/CheckPVV_PINOffset/Rq/IsPVV	Int(1)*	In	Available values: 1 – PVV 0 – PIN Offset -1 – define by prefix
FIMI/CheckPVV_PINOffset/Rq/PINBlock	Hex(16)*	In	PIN block
<p>The PIN block is generated in the format specified in PINBlockFormat and encrypted by the working key.</p> <p>If the working key was generated with the TPK(1) type, the PIN block is encrypted by the clear value of the working key (the cryptogram obtained in the <i>GetWorkingKey</i> operation must be decrypted in advance using the TMK key).</p> <p>If the working key was generated with the RSA Sign Key Pair (2) type, the PIN block is encrypted by a random ZPK key and ZPK key is encrypted by the public key of the working key and encrypted key is transferred in the request in the ZPK key.</p>			
FIMI/CheckPVV_PINOffset/Rq/PINBlockFormat	Int(1)	In	Format of the PIN block in the field. Available values: 0 or NULL – ANSI X9.8 (ISO Format 0); 1 – ANSI X9.8 (ISO Format 1).

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CheckPVV_PINOffset/Rq/PVV_PINOffset	Int(12)	In	PVV/PIN Offset (Internet PVV/Internet PIN Offset) value  For PVV, the value length should be equal to 4. For PIN Offset, the value length should be equal to 12. If the PVV_PINOffset field is not sent, the search for the card in the database is performed and PVV and PIN Offset values are determined by the card data in DB.
FIMI/CheckPVV_PINOffset/Rq/PAN	Str(19)	In	Card number
FIMI/CheckPVV_PINOffset/Rq/MBR	Int(3)	In	Card member number. If NULL, it =0.
FIMI/CheckPVV_PINOffset/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
<b>GetPIN – request for PIN block using card PINOffset</b>			
FIMI/GetPIN/Rq/PAN	Str(19)	In	Card number
FIMI/GetPIN/Rq/MBR	Int(3)	In	Card member number. If NULL, it =0.
FIMI/GetPIN/Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/GetPIN/Rp/PINBlock	Hex(16)	Out	PIN block under ZPK (PIN encryption key set for the card prefix). It is set in the ANSI X9.8 format (ISO Format 0).
<b>CryptoFunction – calling Security Module</b>			
FIMI/CryptoFunction/Rq/FuncName	Str(100)*	In	Name of the function being invoked. See p. 3.21.  Define the respective parameters for each function.
FIMI/CryptoFunction/Rq/CSVK1	Hex(129)	In	CVK cryptogram under LMK. <i>It is used in CalcCVKCheck, CalculateCVV</i>
FIMI/CryptoFunction/Rq/PVK1	Hex(129)	In	PVK cryptogram under LMK. <i>It is used in CalcPVKCheck, CalculatePINOffset, CalculatePVV.</i>
FIMI/CryptoFunction/Rq/ZPK	Hex(129)	In	ZPK cryptogram under LMK. <i>It is used in CalcZonePinKeyCheck, ExportZonePinBlock, ImportZonePinBlock</i>
FIMI/CryptoFunction/Rq/ZMK	Hex(129)	In	ZMK cryptogram under LMK. <i>It is used in CalcZoneMasterKeyCheck, CalcZoneMasterKey, GenerateKey.</i>
FIMI/CryptoFunction/Rq/ZEK	Hex(129)	In	ZEK cryptogram under LMK. <i>It is used in EncryptZoneData, DecryptZoneData.</i>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CryptoFunction/Rq/PINBlock	Hex(16)	In	PIN-block under ZPK in case it is used in ExportZonePinBlock, ImportZonePinBlock. PIN-block under PVK in case it is used in CalculatePVV, CalculatePINOffset. PIN-block under LMK in case it is used in PrintPIN.
FIMI/CryptoFunction/Rq/PAN	Str(19)	In	Card number. <i>It is used in ExportZonePinBlock, ImportZonePinBlock, ImportClearPinBlock, CalculateCVV, CalculatePINOffset, CalculatePVV, GeneratePIN, CheckCAPToken, CalculateARPC.</i>
FIMI/CryptoFunction/Rq/MBR	Int(3)	In	Card member number. If NULL, then =0. <i>It is used in CheckARQC, CheckCAPToken, CalculateARPC.</i>
FIMI/CryptoFunction/Rq/CardUID	Str(64)	In	Unique card ID. It can stand for the card number.
FIMI/CryptoFunction/Rq/PINFormat	Int(1)	In	PIN block format. Available values: 1 – ANSI X9.8 (ISO Format 0) 2 – Docutel 3 – Diebold 4 – PLUS 5 – ISO Format 1 <i>It is used in ExportZonePinBlock and ImportZonePinBlock.</i>
FIMI/CryptoFunction/Rq/PIN	Str(12)	In	PIN as a plain text. It is used in ImportClearPinBlock <b>for test only!</b> <i>ImportClearPinBlock returns the PIN-block in the ANSI X9.8 format (ISO Format 0) encrypted with LMK</i>
FIMI/CryptoFunction/Rq/UseCBC	Int(1)	In	Attribute indicating that the CBC encryption mode is used. If NULL, it=1. <i>It is used in EncryptZoneData, DecryptZoneData.</i>
FIMI/CryptoFunction/Rq/PrintFields	Str	In	Fields for print
FIMI/CryptoFunction/Rq/CVK2	Hex(129)	In	CVK2 cryptogram under LMK <i>It is used in CalculateCVV.</i>
FIMI/CryptoFunction/Rq/Expiration	Str(4)	In	Card expiration date
FIMI/CryptoFunction/Rq/Service	Str(3)	In	Card service code <i>It is used in CalculateCVV.</i>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CryptoFunction/Rq/Decimalizer	Str	In	IBM PIN Offset Decimalization table <i>It is used in CalculatePINOffset.</i>
FIMI/CryptoFunction/Rq/ValDataFormat	Int(1)	In	IBM PIN Offset Validation data format
FIMI/CryptoFunction/Rq/PVK2	Hex(129)	In	PVK2 cryptogram under LMK <i>It is used in CalculatePVV.</i>
FIMI/CryptoFunction/Rq/PVKI	Int(1)	In	Index of the PVK pair of keys <i>It is used in CalculatePVV.</i>
FIMI/CryptoFunction/Rq/Format	Str		Data formatting string <i>It is used in LoadFormattingData.</i>
FIMI/CryptoFunction/Rq/Len	Int(2)	In	PIN length <i>It is used in GeneratePIN and GenerateKey.</i> <i>For GeneratePIN, PIN length is transferred.</i> <i>For GenerateKey, the key length is transferred: 1 - single; 2 - double</i>
FIMI/CryptoFunction/Rq/KeyType	Int(1)	In	Key type Available values: 1 – ZPK 2 – ZAK <i>It is used in ImportKey, ExportKey, CalculateMAC, CheckMAC and GenerateKey.</i>
FIMI/CryptoFunction/Rq/KeyZMK	Hex(129)	In	Cryptogram of the KeyType key under ZMK <i>It is used in ImportKey.</i>
FIMI/CryptoFunction/Rq/KeyLocal	Hex(129)	In	Cryptogram of the KeyType key under LMK <i>It is used in ExportKey, CalculateMAC, CheckMAC, CalculateHMAC, CheckHMAC.</i>
FIMI/CryptoFunction/Rq/AtallaVariant	Int(2)	In	ZMK key variant Available values: 0 .. 31. <i>It is used in ImportKey and ExportKey.</i>
FIMI/CryptoFunction/Rq/Data	Str	In	Data for MAC calculation, signature, encryption, decryption (in base64 encoding). <i>They are used in CalculateMAC, CheckMAC, CalculateSignature, EncryptZoneData, DecryptZoneData, CalculateHMAC, CheckHMAC.</i>
FIMI/CryptoFunction/Rq/MAC	Hex	In	MAC/HMAC on data. <i>It is used in CheckMAC, CheckHMAC.</i>
FIMI/CryptoFunction/Rq/RSASK	Hex (2000)	In	Cryptogram of private RSA key under LMK. <i>It is used in CalculateSignature.</i>

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CryptoFunction/Rq/HashAlgorithm	Int(1)	Int	<p>Hashing algorithm type</p> <p>Available values:</p> <ul style="list-style-type: none"> <li>0 – no hashing, PKCS#1 padding</li> <li>1 – SHA-1 hash, PCKS#1 padding</li> <li>2 – MD5 hash, PKCS#1 padding</li> <li>3 – no hashing, no padding</li> <li>4 – SHA-224 hash, PCKS#1 padding</li> <li>5 – SHA-256 hash, PCKS#1 padding</li> <li>6 – SHA-384 hash, PCKS#1 padding</li> <li>7 – SHA-512 hash, PCKS#1 padding</li> </ul> <p><i>It is used in CalculateSignature, CalculateHMAC, CheckHMAC.</i></p>
FIMI/CryptoFunction/Rq/CAPToken	Str(40)	In	<p>CAP token of the chip card</p> <p><i>It is used in CheckARQC, CheckCAPToken.</i></p>
FIMI/CryptoFunction/Rq/Challenge	Str(8)	In	<p>Random encryption component</p> <p><i>It is used in CheckARQC, CheckCAPToken.</i></p>
FIMI/CryptoFunction/Rq/ARQC	Hex(8)	In	<p>EMV Tag – 9f26. Application Cryptogram</p> <p><i>It is used in CalculateARPC, the field is mandatory.</i></p>
FIMI/CryptoFunction/Rq/RespCode	Hex(2)	In	<p>Response code</p> <p><i>It is used in CalculateARPC</i></p>
FIMI/CryptoFunction/Rq/CSU	Hex(4)	In	<p>Card Status Update</p> <p>The field is mandatory for the CalculateARPC function if the CPA/CCD EMV scheme is used.</p> <p><i>It is used in CalculateARPC</i></p>
FIMI/CryptoFunction/Rq/ICC_Amount	Str(12)	In	<p>EMV Tag – 9f02. Amount</p> <p><i>It is used in CalculateARPC</i></p>
FIMI/CryptoFunction/Rq/ICC_CBAmount	Str(12)	In	<p>EMV Tag – 9f03. Cashback transaction amount</p> <p><i>It is used in CalculateARPC</i></p>
FIMI/CryptoFunction/Rq/ICC_TermCountry	Int	In	<p>EMV Tag – 9f1a. Terminal country code according to ISO 3166</p> <p><i>It is used in CalculateARPC</i></p>
FIMI/CryptoFunction/Rq/ICC_TVР	Hex(5)	In	<p>EMV Tag – 95. Results of the card/transaction state verification by the terminal (bit mask).</p> <p><i>It is used in CalculateARPC</i></p>
FIMI/CryptoFunction/Rq/ICC_Currency	Int	In	<p>EMV Tag – 5f2a. Currency according to ISO 4217</p> <p><i>It is used in CalculateARPC</i></p>
FIMI/CryptoFunction/Rq/ICC_TranDate	Str(6)	In	<p>EMV Tag – 9a. Transaction date (YYMMDD)</p> <p><i>It is used in CalculateARPC</i></p>

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
FIMI/CryptoFunction/Rq/ ICC_TranType	Int	In	EMV Tag – 9c. Transaction type <i>It is used in CalculateARPC</i>
FIMI/CryptoFunction/Rq/ ICC_Random	Hex(4)	In	EMV Tag – 9f37. Unpredictable number generated by the terminal <i>It is used in CalculateARPC</i>
FIMI/CryptoFunction/Rq/ ICC_AppProfile	Int	In	EMV Tag – 82. Application Interchange Profile (bit mask) <i>It is used in CalculateARPC</i>
FIMI/CryptoFunction/Rq/ ICC_AppTranCount	Str(8)	In	EMV Tag – 9f36. Application Transaction Counter (ATC). It is transferred in the decimal format. <i>It is used in CalculateARPC</i>
FIMI/CryptoFunction/Rq/ ICC_CardMember	Int(3)	In	Card member number If NULL, the MBR field is used. <i>It is used in CalculateARPC.</i>
FIMI/CryptoFunction/Rq/ ICC_IssuerData	Hex(32)	In	EMV Tag 9F10. Issuer Discretionary Data <i>It is used in CalculateARPC, the field is mandatory.</i>
FIMI/CryptoFunction/Rq/ KeyCheckLen	Int(1)	In	Size of the check value in bytes (it is transferred in the KeyCheck response field): 2 – two bytes. 3 – three bytes. If NULL, it is 2. <i>It is used in ImportKey, ExportKey and GenerateKey.</i>
FIMI/CryptoFunction/Rp/Result	Str	Out	Function activity result
FIMI/CryptoFunction/Rq/KeyZMK	Hex(129)	Out	Cryptogram of the KeyType key under ZMK <i>It is used in ExportKey and GenerateKey.</i>
FIMI/CryptoFunction/Rp/KeyLocal	Hex(129)	Out	Cryptogram of the KeyType key under LMK. <i>It is used in ImportKey and GenerateKey.</i>
FIMI/CryptoFunction/Rp/KeyCheck	Int	Out	Key check value. The size (two or three bytes) depends on the value in the KeyCheckLen request field. <i>It is used in ImportKey, ExportKey and GenerateKey.</i>
<b>PrepareProvisioningData – generating data for Push Provisioning (card registration in wallet)</b>			
FIMI/PrepareProvisioningData/ Rq/PAN	Str(19)	In	Card number
FIMI/PrepareProvisioningData/ Rq/MBR	Int(3)	In	Card member number. If NULL, it =0.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/PrepareProvisioningData/ Rq/CardUID	Str(64)	In	Unique card identifier. It can stand for the card number.
FIMI/PrepareProvisioningData/ Rq/WalletType	Int(1)*	In	Wallet type: 1 – Apple Pay 2 – Samsung Pay 3 – Google Pay 4 – MDES Issuer Wallet 5 – MIR Pay 6 – Visa Issuer Wallet
FIMI/PrepareProvisioningData/ Rq/ExpirationDate	Time	In	Card expiration date. If it is not transferred but required for the generation, card data in TWO is used.
FIMI/PrepareProvisioningData/ Rq/Time	Time	In	Generation time. It is used for the Apple Pay wallet. If it is not transferred but required for the generation, it is calculated automatically in TWO in compliance with the card prefix settings.
FIMI/PrepareProvisioningData/ Rq/Nonce	Str	In	Random number. It is used for the Apple Pay wallet for cards with the Visa, NSPK card program.
FIMI/PrepareProvisioningData/ Rq/CVV2	Int(3)	In	Card CVV2. It is used for the Samsung Pay and Google Pay wallets.
FIMI/PrepareProvisioningData/ Rq/NameOnCard	Str(250)	In	Name on card. It is used for the Samsung Pay and Google Pay wallets.
FIMI/PrepareProvisioningData/ Rq/Provider	CustAcct (1000)	In	Payment Instrument Provider. It is used for the Samsung Pay and Google Pay wallets, Visa Issuer Wallet. For the list of available fields and their formats, refer to the <i>VTS Push Provisioning</i> specification.
FIMI/PrepareProvisioningData/ Rq/BillingAddress	CustAcct (1000)	In	Billing Address. It is used for the Samsung Pay and Google Pay wallets. For the list of available fields and their formats, refer to the <i>VTS Push Provisioning</i> specification.
FIMI/PrepareProvisioningData/ Rq/WalletId	Str(250)	In	Unique wallet ID on the device. It is used for the MIR Pay, Samsung Pay and Google Pay wallet types in case cards have the NSPK MIR card program.

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/PrepareProvisioningData/Rq/DeviceId	Str(250)	In	Device ID. It is used for the MIR Pay, Samsung Pay and Google Pay wallet types in case cards have the NSPK MIR card program.
FIMI/PrepareProvisioningData/Rq/TokenRequestorId	Int(11)	In	Unique ID of the object that initiated the tokenization process. It is used for MDES Issuer Wallet.
FIMI/PrepareProvisioningData/NonceSignature	Hex	In	Signature. It is used for the Apple Pay wallet.
FIMI/PrepareProvisioningData/Certificates	Str[]	In	X.509 certificates in the Base64 format. It is used for the Apple Pay wallet.
FIMI/PrepareProvisioningData/Rp/Data	Str(4000)*	Out	Encryption result: For the Samsung Pay and Google Pay wallets, the string in the JWE format is returned (for details, refer to the <i>VTS Push Provisioning</i> specification). For the Apple Pay wallets for cards with the Visa card program, the string with encrypted data is returned. For the Apple Pay wallet for cards with the MC card program, the string with encrypted data in the Base64 format is returned. For MDES Issuer Wallet, the string with encrypted data in the Base64 format is returned. For the Mir Pay, Samsung Pay and Google Pay wallet types for cards with the NSPK MIR card program, the string with encrypted data in the Base64 format is returned (for details, refer to <i>Mir Payment System Standard. Specification for Issuers on App2App Provisioning Implementation for Mir Pay, Google Pay and Samsung Pay</i> ).
FIMI/PrepareProvisioningData/Rp/FundingAccountInfo	Str(4000)	Out	Structure with card data. It is returned for MDES Issuer Wallet in the Base64 format.
FIMI/PrepareProvisioningData/EncryptedPassData	Str(4000)	Out	Encrypted data to be transferred. It is returned for the Apple Pay wallet in the Base64 format.
FIMI/PrepareProvisioningData/EphemeralPublicKey	Str(4000)	Out	Ephemeral public key. It is returned for the Apple Pay wallet in the Base64 format.

Parameter	Type (* - mandatory parameter)	Direction (relative to TWO)	Description
<b><i>GetExcludedPINList – getting list of “weak” PINs</i></b>			
FIMI/GetExcludedPINList/Rq/NotUse	Int(1)	In	If =0 or NULL, the response will return only PINs that must be loaded to HSM. If =1, the response must contain all PINs (both the ones that must be loaded to HSM and those that must not be loaded)
FIMI/GetExcludedPINList/Rq/List	Str[]	Out	Array of weak PINs Maximum length of one array element is 12 symbols
<b><i>GetRepositoryKey – obtaining key value from Key Management System</i></b>			
FIMI/GetRepositoryKey/Rq/KeyId	Int	In	Key ID in the TWO repository.
FIMI/GetRepositoryKey/Rq/ExternalKeyId	Int	In	External ID of the key. It is used if KeyId is not defined.
FIMI/GetRepositoryKey/Rq/KeyType	Int(2)	In	Key type: 1 – TMK 2 – TPK 3 – TAK 4 – ZMK 5 – ZPK 6 – ZAK 7 – PVK 8 – CVK 9 – ICC Auth 10 – ICC Encr 11 – ICC MAC 12 – BDK 13 – RSA-SK 14 – HMAC 15 – ICC CVC3 16 – CSCK 17 – ZEK NULL - any
FIMI/GetRepositoryKey/Rq/TermName	Str(16)	In	Name of the key owner terminal It is used if KeyId and ExternalKeyId are not specified.
FIMI/GetRepositoryKey/Rq>Status	Int(1)	In	Key status: 1 – New 2 – In transit 3 – Ready 4 – Loaded 6 – Expired 7 – Compromised 9 – Cancelled NULL – any
To search for the key, the request must transfer at least one of the KeyId, ExternalKeyId, TermName fields. Otherwise, the request will be declined.			

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/GetRepositoryKey/Rq/ KeyCheckLen	Int(1)	In	Size of the check value in bytes (it is transferred in the KeyCheck response field): 2 – two bytes. 3 – three bytes. If NULL, it is 2. It is used if a key was found according to the search conditions, otherwise – it is always 2.
FIMI/GetRepositoryKey/Rp/ KeyLocal	Hex(1280)	Out	Key cryptogram under LMK It is returned if one key has been found according to the search conditions. Otherwise, the information is returned in the KeyInfo structure.
FIMI/GetRepositoryKey/Rp/ KeyCheck	Int	Out	Key check value. The size (two or three bytes) depends on the value in the KeyCheckLen request field. It is returned if one key has been found according to the search conditions. Otherwise, the information is returned in the KeyInfo structure.
FIMI/GetRepositoryKey/Rp/KeyInfo	ArrRec	Out	{ int* KeyId; int ExternalKeyId; Hex(1280)* KeyLocal; int* KeyCheck; str(250) Title; int(2)* Type; int(1)* Status; time ExpDate; str OwnerTitle; } [] – array of structures, list of keys:
<p>KeyId – key ID            ExternalKeyId – external ID of the key            KeyLocal – cryptogram of the key under LMK            KeyCheck – key check value            Title – title            Type – key type (for the list of values, see FIMI/GetRepositoryKey/Rq/KeyType)            Status – key status (for the list of values, see FIMI/GetRepositoryKey/Rq/Status)            ExpDate – key validity period (NULL - unlimited)            OwnerTitle – name of the key owner</p>			
<b>AssignKeyByTerm – linking key from Key Management System to terminal</b>			
FIMI/AssignKeyByTerm/Rq/KeyId	Int	In	Key ID in the TWO repository
FIMI/AssignKeyByTerm/Rq/ ExternalKeyId	Int	In	Key external ID  The field value is used if the KeyId field is not defined. The search is performed by the "Key Index" key attribute (see the <b>Key Management System</b> application).

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
FIMI/AssignKeyByTerm/Rq/TermId	Int*	In	Terminal ID It is allowed to transfer ID of the ATM/POS terminals.
FIMI/AssignKeyByTerm/Rq/KeyType	Int(1)*	In	Key type: 1 – POS PIN 2 – POS MAC 3 – POS EMK 4 – ATM TMK Types 1,2,3 are allowed for the POS terminals only, type 4 is allowed for ATMs only.
FIMI/AssignKeyByTerm/Rq/KeyExpDate	Time	In	Key validity period If the field is not transferred, the unlimited validity period is set for the key.

The comment to the operation is defined in the FIMI/Comment field, its length is 250 characters, the comment will display in the key states history.

#### **ChangeKeyStatus – changing status of key from Key Management System**

FIMI/ChangeKeyStatus/Rq/KeyId	Int	In	Key ID in TWO repository
FIMI/ChangeKeyStatus/Rq/ExternalKeyId	Int	In	Key external ID The field value is used if the KeyId field is not defined
FIMI/ChangeKeyStatus/Rq>Status	Int(1)*	In	Key status: 1 – New 2 – In transit 3 – Ready 4 – Loaded 6 – Expired 7 – Compromised 9 – Cancelled  Allowed transitions: New -> In transit New -> Ready New -> Expired New -> Compromised In transit -> Expired In transit -> Compromised Ready -> Expired Ready -> Compromised Loaded -> Compromised; Loaded -> Expired  The key transition to the Loaded status is prohibited. The transition from the Expired, Compromised, Cancelled statuses is prohibited. The key transition from the Loaded status is available for the keys linked to the POS terminals and ATMs. When changing the status, the key is unlinked from the terminal to which it was linked before. To restore the

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
			terminal operability, it is required to link a new key using the <i>AssignKeyByTerm</i> operation.

The comment to the operation is defined in the FIMI/Comment field, its length is 250 characters, the comment will display in the key states history.

### 2.3.13 User Operations

Parameter	Type (*- mandatory parameter)	Direction (relative to TWO)	Description
<b><i>UserDefined – calling algorithmix user function</i></b>			
FIMI/UserDefined/Rq/OperName	Str(100)	In	Name of user operation (according to the applied logic of FIMI terminal). The response is created in <b>TWO</b> by the algorithm depending on operation name and input parameters transferred in the <i>Parameters</i> field.
FIMI/UserDefined/Rq/UserData	Str	In	<p>Input data required for the execution on <b>TWO</b> side. The data are created according to the applied logic of FIMI terminal. It is recommended to use the structured format of field, e.g., transfer the parameters as XML document. To describe the XML document structure, use the <i>UserData</i> type of the scheme <i>fimi_userdefined.xsd</i>.</p> <p>If the SOAP XML protocol is used, the <i>UserData</i> tag with the attributes and namespace is transferred to algorithmix.</p>
FIMI/UserDefined/Rp/Result	Str	Out	Results of the request execution in <b>TWO</b> . It is recommended to use the structured format of field, e.g., transfer the response as XML document. To describe the XML document structure, use the <i>UserData</i> type of the scheme <i>fimi_userdefined.xsd</i>

## 3 Reference Data

### 3.1 Available Values of Response Field

Request is approved if FIMI/Response=1. If FIMI/Response=23, 29, 52, 65, the request is partially approved or declined but the additional information is sent in the response (see the description of operations *AddRate*, *ModifyRate*, *DeleteRate*, *Logon*, *AcctDebit*). Any other values of the *FIMI/Response* field mean that the transaction has been declined. The list of all available codes is the following:

Value	Name	Description
1	Approved	OK
2	FormatError	Format error of the incoming message
3	InvalidPINBlockFormat	Invalid format of PIN block
4	InvalidKeyId	Invalid key identifier
5	InvalidClerk	Invalid (unauthorized) FIMI terminal user
6	InvalidPassword	Invalid password of the FIMI terminal user
7	ClerkExpired	The user account is expired
8	SystemMalfunction	System error
9	ClerkBlocked	The user is blocked in <b>TWO</b>
10	InvalidTerminal	Invalid terminal
11	InvalidTransaction	Invalid transaction (not allowed for the FIMI terminal)
12	InvalidCard	Invalid card (not found in the database)
13	InvalidAccount	Invalid account
14	InsufficientFunds	Account funds are insufficient
15	AccountAlreadyExists	Account already exists
16	Operation not allowed in Impact period	Account operation is prohibited in the Impact period
17	UnableExecute Command	Command execution is impossible
18	InvalidSession	Invalid session identifier
19	PVVVerificationFailed	PVV/PIN Offset verification has failed
20	DuplicatedTransaction	Transaction has been already conducted
21	DuplicatedRate	Currency rate with such parameters already exists (possible Response to the AddRate message)
22	RateNotFound	Rate with the defined attributes is not found (possible Response to the messages: ModifyRate, DeleteRate)

<b>Value</b>	<b>Name</b>	<b>Description</b>
23	InvalidExchangeDate	Rate date is invalid (possible Response to the messages: AddRate, ModifyRate, DeleteRate)
24	InvalidCurrency	Currency code is invalid (possible Response to the AddRate message)
25	UnableAcctCredit	Account credit is prohibited
26	UnallowedEMVScriptType	Prohibited type of EMV script
27	UniqueSessionClerkViolated	FIMI operator session uniqueness is violated
28	InvalidTelebankCustomer	Invalid identifier of the Telebank customer
29	CAPAuthRequired	CAP-authentication is required
30	InvalidCAPToken	CAP-Token check value is invalid
31	UnableLocate StoplistRecord	The card is absent in the Stoplist
32	NeedClerkConfirmation	FIMI operator confirmation is needed
33	RestrictedLimitValue	The limit value is restricted
35	RequiredCardStatusNotAllowed	The card status is prohibited by the FIMI operator profile
36	InvalidPersonId	Invalid customer ID
37	InvalidInstitutionName	Invalid name of the financial institution
38	InvalidCardPassword	Invalid password of the card
39	DuplicatedRecord	The record already exists
41	RepresentNoOrig Found	No original transaction is found when generating the Representent
42	RepresentTooMany OrigsFound	Several original transactions are found when generating the Representent
43	LimitExceeded	Limit is exceeded
44	ReportHasInvalidParameters	The report does not contain atomic parameters. It can not be executed.
45	ErrorWhileReportExecution	Report execution error
46	ReportRequestScheduled	Request for execution waits for processing
47	ReportRequestInProgress	Request for report execution in the course of processing
48	InvalidReport	Report is not found
49	InvalidBMSMessage	Invalid BMS message
50	BMSSendMessageDeclined	The BMS message is declined
51	UnknownSavepoint	Unknown savepoint is defined in the RollBack request
52	DynamicPasswordRequired	Dynamic authentication is required
53	InvalidDynamicPassword	The dynamic password is invalid

Value	Name	Description
54	AbonentDoesntExist	Abonent does not exist
55	AbonentIsBlocked	Abonent is blocked for dynamic authentication
56	DynamicPaswordLifetimeExpired	Dynamic password lifetime is expired
57	BadDynamicPasswordLimitExceeded	Bad dynamic password tries limit is exceeded
58	AbonentAlreadyExist	Abonent already exists
59	OriginalTransactionIsNotMDS	Original transaction is not MasterCard Global MDS-transaction
60	InvalidRetailer	Retailer invalid code
61	TerminalDeactivated	POS terminal is deactivated
62	InvalidParameterValue	Invalid value of parameter
63	ObjectUIDError	Error of object search by UID: the object does not exist or not found
64	NotAddEMVScriptToNon	EMV script was not added to the non-EMV card
65	PartialAuthorizationExecuted	Partial authorization executed
66	DeletingDeclined	Deleting of the card, Telebank customer, account is declined
67	Card Not Defined For Dynamic Authentication	Card for the dynamic authentication is not defined
68	InvalidSignature	Invalid signature in the request
69	OperationNotAllowedOnEPMode	The operation is not allowed in the Emergency Processing mode
70	UpdatePasswordFormat	The operation cannot be performed as the operator uses the password hashing algorithm with the version lower than that set on the terminal. It is required to change the password for the FIMI operator in TWO or using the ChangePassword operation. If the password is changed, the version of the operator password hashing algorithm will be brought to the current version specified on the terminal.
71	InvalidLegalEntity	Invalid identifier of a corporate customer

### 3.1.1 Translation of Authorizer Response Codes to Response Field Values

The authorizer response codes (refer to 3.9) are translated to the Response field values for the operations (*SetCardPVV*, *AddEMVScript*, *ChangeAcctBalances*, *SetAccountOverdraft*, *AcctDebit*, *AcctCredit*, *CNSCardConfig*) calling the **CST and FIMI Operation Control** algorithmix functions. If the algorithmix function declines the request, the translated authorizer code is added to the response (to the Response field).

Response Field Value	Response Field Value Name	Authorizer Response Code	Authorizer Response Code Name
1	Approved	1	Approved
2	FormatError	74	Format error
8	SystemMalfunction	0	None
		2	Approved Partial
		3	Approved Purchase Only
		11	Should change PVV
		12	Confirm Payment Precheck
		13	Select Bill
		16	Slip already received
		40	Lost card
		41	Stolen card
		50	Unauthorized usage
		51	Expired card
		54	System error
		66	Statement not available
		65	No statement information
		71	Contact card issuer
		72	Destination not available
		73	Routing error
		75	External decline special condition
10	InvalidTerminal	69	No sharing
11	InvalidTransaction	55	Ineligible transaction
		57	Transaction not supported
		58	Restricted Card
		68	External decline
		73	Routing error
		82	Invalid transaction
12	InvalidCard	52	Invalid card
		30	Blocked
13	InvalidAccount	49	Ineligible vendor account
		56	Ineligible account
		29	Closed account

Response Field Value	Response Field Value Name	Authorizer Response Code	Authorizer Response Code Name
14	InsufficientFunds	21	Corresponding account exhausted
		59	Insufficient funds
19	PVVVerificationFailed	53	Invalid PIN
		80	Bad CVV
		81	Bad CVV2
		84	Bad CAVV
		85	Bad ARQC
41	RepresentNoOrigFound	15	Original transaction not found
43	LimitExceeded	22	Acquirer limit exceeded
		60	Uses limit exceeded
		61	Withdrawal limit would be exceeded
		62	PIN tries limit was reached
		63	Withdrawal limit already reached
		64	Credit amount limit
		83	PIN tries limit was exceeded
62	InvalidParameterValue	17	Personal information input error
		67	Invalid amount

### 3.2 Card Status

**TWO** standard card statuses are listed below. These statuses are present in the **TWO** at the initial setup. In addition, the other statuses can be created, at that, their IDs and titles should be transmitted in the field InitSession/Rp/CardStatList.

Card Status	TWO	Description
0	Not Active	The card is not active
1	Open	The card is operable
2	Lost	The card is lost
3	Stolen	The card is stolen
4	Restricted	The card is restricted – only some transactions are allowed for the card
5	VIP	The card is owned by VIP-client
6	Open Domestic	The card is opened for the domestic use
8	Compromised	The card is compromised (forged, suspected and etc.)
9	Closed	The card is closed, the use is prohibited
10	Referral	The additional request to the issuer is required
12	Declared	The card is not issued
15	Expired	The card is expired

### 3.3 Card Limits

**TWO system** limits are listed below. These limits are always present in **TWO**. Moreover, the system allows **user** limits of the card, account or account-card link whose IDs are defined independently by **TWO** administrator.

ID	Limit Name	Limit Title	Application	Counter
1	BAD_PIN_TRIES	Invalid PIN	TOTAL	Yes
2	TTL_WDL_LMT	Withdrawal and Purchase total limit	TOTAL	
3	TTL_AGGR_LMT	Expenses Total Limit	TOTAL	
101	ATM_USE_LMT	Card Use Limit	ATM	Yes
102	ATM_WDL_LMT	Cash Withdrawal Limit	ATM	
103	ATM_DEP_LMT	Deposit Limit	ATM	
201	POS_USE_LMT	Limit on the Card Use at POS terminal	POS(+CRT)	Yes
202	POS_WDL_LMT	Limit on Purchase and POS terminal Withdrawal	POS(+CRT)	
203	POS_RFND_LMT	Purchase Return via POS terminal	POS(+CRT)	
204	POS_DEP_LMT	Deposit via POS terminal	POS(+CRT)	
211	INTERNET_BAD_PIN_TRIES	Invalid Internet-PIN	POS(+CRT)	Yes
301	EMV_ATC_INCREMENT	Limit on the Number of Offline-Transactions on EMV-card	TOTAL	Yes

### 3.4 Terminal Types

Type	Class	Title in TWO
1	ATM	ATM NDC+
2	ATM	ATM DDC
101	POS	POS TPTP Lite
102	POS	POS TPTP Full Option
103	POS	POS TITP Hypercom
105	POS	POS Sudapan
106	POS	POS IBTP
201	VOICE	CRT Terminal
202	VOICE	Imprinter
301	VTBI	TeleBank Internet
302	VTBI	TeleBank Phone
303	VTBI	TeleBank Voice
305	VTBI	FIMI
306	VTBI	E-Commerce

### 3.5 ATM Device State

Value	State
0	OK
1	Routing error
2	Warning
3	Suspend
4	Fatal error

452(554)

### 3.6 ATM Media State

Value	State
0	Not configured
1	OK
2	Low
3	Out
4	Overfill

### 3.7 Account Operation Codes

The system account operation codes are listed below. They are always predefined and present in **TWO**. Besides the codes, the user operation codes with the identifiers exceeding 500 can be created. For the full dictionary of the operation codes, see the AcctOperations field, response to the InitSession request.

Operation Code	FO Only	Title in TWO
1		Cash Receipts
2		Cash Withdrawals
3		Payment by Order
4	*	POS Cash Advance
5	*	ATM Deposit
6	*	ATM Withdrawal
7		Place to Account
8		Account Debit
9		Fee from Customer
10		Payment under Insurance Contract
11		Capitalization
12	*	ATM Balance Inquiry
13	*	ATM MiniStatement
14	*	ATM Statement
15	*	ATM Message to Institution
16	*	ATM P2P Debit
17	*	ATM P2P Credit
18	*	ATM Card Statement
19	*	Telebank Outside Transfer
20	*	Telebank Transfer Credit
21	*	ATM Transfer
22	*	ATM Payment
23	*	ATM PrepaidCode
24	*	POS PrepaidCode
25	*	Telebank PrepaidCode
26	*	Telebank Outside Transfer
27	*	POS Deposit
28	*	POS Deposit Adjustment
30	*	POS Quasi-Cash
31	*	POS Purchase

<b>Operation Code</b>	<b>FO Only</b>	<b>Title in TWO</b>
32	*	POS Merchandise Return
33	*	POS Purchase Adjustment
34	*	POS Merchandise Return Adjustment
35	*	POS Cash Advance Adjustment
36	*	POS Pre-Purchase
37	*	POS Pre-Purchase Complete
38	*	POS Balance Inquiry
39	*	POS Mail/Phone Order
40	*	Card Verification
41	*	POS P2P Debit
42	*	POS P2P Credit
43	*	Telebank Transfer
44	*	Telebank Payment
45	*	Telebank Buy Balance
46	*	Telebank Buy MiniStatement
47	*	Telebank P2P Debit
48	*	Telebank P2P Credit
49	*	Telebank Message to Institution
51		Account Statement Fee
52		Customer Payments Remittance
53		Capitalization with Hold
54		Card Status Change Fee
55		Card Maintenance Fee
56		Close Account
57		Close Credit Line
58		Fix Credit Line
59		Use Credit Line
60		Breach of Credit Terms
61		Repay Credit Debt
62		Credit
63		Currency Purchase
64		Currency Sale
65		Denomination
66		Balancing of Agreement Accounts

<b>Operation Code</b>	<b>FO Only</b>	<b>Title in TWO</b>
67		Premium to Customer
68		Card Issuing Fee
69		Account Opening Fee
70		Repay Credit
71		Place Interest on Account
72		Pay off Outstanding Interest
73		Write off Interest
74		Recover Balance Due
75		Repay Debt by Balance Due
76		Hold Calculation
77		Write Off Hold
78		Single Payment
79		Fine
80		Salary
81	*	FIMI Account Debit
82	*	FIMI Account Credit
83	*	POS PIN Change
84		Change PIN
85		Write Card
86	*	POS Transfer
87	*	POS Mini Statement
88	*	CMS P2P Debit
89	*	CMS P2P Credit
91	*	CMS Payment
92	*	POS Payment
93	*	Bonus Accumulation
94	*	POS Redeem Bonus
98		Decline of a Purchase Authorization
99		Other Operation on Account
100		Telebank Balance Inquiry
101		C2C Debit
102		C2C Credit
106		C2B Debit
107		C2B Credit
108		B2C Credit

### 3.8 Transaction Codes

Transaction Code	Title
<b>ATM Transactions</b>	
10	Withdrawal
15	ATM Currency Exchange
20	Deposit
21	Cardless Deposit
22	ATM Change EC Status
23	ATM Add Abonent
26	ATM Get Account
27	ATM Cheque Deposit
30	Balance Inquiry
31	Transfer Data Inquiry
40	Transfer
41	ATM Transfer Pass
42	ATM Transfer Debit
43	ATM Transfer Credit
46	ATM P2P Pass
47	ATM P2P Credit Pass
48	ATM Transfer Credit Pass
50	Payment
51	ATM Payment Pass
52	ATM Payment Debit
53	ATM Payment Credit
54	ATM Cash Payment
55	ATM Bills Request
56	ATM Bill Payment
57	ATM Bill Reject
58	Payment return
59	ATM Cash Payment Pass
60	Message to Financial Institution
61	User Defined
62	ATM PrepaidCode
63	ATM P2P Debit
64	ATM P2P Credit
65	ATM P2P Transfer
66	ATM Precheck
67	ATM Confirmation
68	ATM Payment Request

Transaction Code	Title
69	ATM P2P Credit Confirmation
70	Statement Print
71	Mini-Statement Print
72	Card Statement
73	ATM Prepaid Pass
74	ATM Prepaid Debit
75	ATM Prepaid Credit
76	ATM Card Cash Payment Pass
77	ATM Card Cash Payment
80	ATM Create Virtual Card Pass
81	PIN Change
82	Internet PIN Change via ATM
83	Link Authentication Card to Telebank Customer
84	Link Card to Telebank Customer
85	ATM Create Virtual Card
86	Telebank PIN Change via ATM
87	PIN Unblock
88	ATM Create Telebank Customer Pass
89	ATM Create Telebank Customer
90	Check PIN
91	ATM Balancing
92	ATM Cutover
93	ATM Cutover and Balancing
95	Adjust ATM Hopper
96	Decrease ATM Hopper
97	Increase ATM Hopper
98	Decrease ATM CashIn Hopper
99	Adjust ATM Coins
100	Decrease ATM Coins
101	Increase ATM Coins
102	ATM Send BMS Message
103	Internet Payment System Registry Confirmation
104	ATM P2P Internet Payment System Transfer Pass
105	ATM Internet Payment System Contract Request
106	Internet Payment System Registry Cancellation
107	Internet Payment System Registration
108	ATM Service Auto Payment
109	ATM Service Auto Payment Pass

Transaction Code	Title
<b>POS Transactions</b>	
110	Purchase
111	Pre-Purchase
112	Pre-Purchase Complete
113	POS Mail/Phone Order
114	Merchandise Return
115	Cash Advance
116	Card Verification
117	POS Balance Inquiry
118	Purchase With Cashback
119	Check Verification
120	Check Guarantee
121	Purchase Adjustment
122	Merchandise Return Adjustment
123	Cash Advance Adjustment
124	Pre-Purchase Increment
125	Purchase Cancellation
126	POS Message to Financial Institution
127	Cardless Cash Advance
130	Quasi-Cash
131	POS PrepaidCode
132	POS P2P Debit
133	POS P2P Credit
134	POS P2P Calc Fee
135	POS P2P Pass
136	POS Installment Details
137	POS Transfer
138	POS Mini-statement Print
139	POS PIN Change
140	POS Deposit
141	POS Deposit Adjustment
142	POS Payment
143	POS Precheck
144	POS Confirmation
145	POS Payment Request
146	POS Bills Request
147	POS Transfer Debit
148	POS Transfer Credit
149	POS Transfer Pass

Transaction Code	Title
150	Logon
151	Logoff
152	POS Create Telebank Customer Pass
153	POS Create Telebank Customer
154	POS Link Ccard to Telebank Ccustomer
155	POS Link Authentication Card to Telebank Customer
156	POS Add Abonent
157	POS Update Abonent
158	POS Delete Abonent
159	Telebank PIN Change via POS
160	Close Batch
161	Close Shift
162	Close Day
164	Clerk Subtotals
165	Batch Subtotals
166	Shift Subtotals
167	Day Subtotals
168	POS Debit Adjustment
169	POS Credit Adjustment
170	POS Cardless Get Rates
171	POS Prepaid Pass
172	POS Prepaid Debit
173	POS Prepaid Credit
175	POS Payment Pass
176	POS Payment Debit
177	POS Payment Credit
178	Tokenization
179	Send Dynamic Password
180	POS Transfer Inquiry
181	POS Accumulate Bonus
182	POS Redeem Bonus
183	POS Cash Payment
184	POS Cash Payment Pass
185	POS Calc Fee
186	POS Create Virtual Card Pass
187	POS Create Virtual Card
188	Tokenization Completion
189	Tokenization Event
190	Download

Transaction Code	Title
191	Generate IPEK
193	Download file
194	POS Exchange Rate Inquiry
195	Handshake
196	POS TC Advice
197	Internet PIN Change via POS
198	Get Card by Alias
199	Alias Action
801	Alias Action Pass
802	C2C Pass
803	C2C Debit
804	C2C Credit
820	QR Code Data Request
821	QR Code Status Change Notification
822	C2B Pass
823	C2B Debit
824	C2B Credit
825	QR Code Status Request
826	C2B Payment Data Request
827	ATC Update
828	B2C Pass
829	B2C Debit
830	B2C Credit
831	B2C Status
832	POS QR Purchase
833	POS QR Refund
834	POS Get PIN
835	C2C Pull Pass
836	C2C Pull Check
837	C2C Pull
838	POS QR Purchase Debit
839	POS QR Purchase Credit
840	POS QR Purchase Pass
841	POS QR Actions
850	Hold Debit
851	Hold Credit

Transaction Code	Title
852	Change Hold Debit
853	Change Hold Credit
854	Clearing Hold Debit
855	Clearing Hold Credit
856	POS EC Purchase
857	POS EC Refund
860	C2G Pass
861	C2G Debit
862	C2G Credit
<b>Interchange Transactions</b>	
201	Debit Adjustment
202	Credit Adjustment
205	Fee Collection
206	Funds disbursement
209	Text Message
210	Copy Request
211	Confirmation of Copy Request
220	Add Card to Stoplist
221	Delete Card from Stoplist
222	Modify Stoplist
223	Inquiry Stoplist
224	Add Region to Stoplist
225	Modify Region in Stoplist
226	Delete Region from Stoplist
230	Host Logon
231	Host Logon
232	Set Card Token Status
233	Token Confirmation
234	PAR Inquiry
235	Get Card by Token
236	Token Inquiry
237	Update Card Metadata for Tokens
250	Change Invoice Status
290	Acquirer Totals
291	Issuer Totals
292	Acquirer Batch Totals
293	Issuer Batch Totals
294	Acquirer Day Totals
295	Issuer Day Totals

Transaction Code	Title
296	Acquirer STIP Batch Totals
297	Issuer STIP Batch Totals
298	Acquirer STIP Day Totals
299	Issuer STIP Day Totals
<b>Telebank Transactions</b>	
301	Telebank Logon
302	Account List
303	Operation History
304	Change Account Description
305	Telebank Balance Inquiry
306	Change Customer CMS Address
307	Change Card CMS Address
308	Change Card to Account Description
309	Set Card Limits
310	Full Vendor List
311	Customer Vendor List
312	Attach Vendor
313	Detach Vendor
314	Card Vendor List
315	Attach Vendor to Card
316	Detach Vendor from Card
317	Change Text Login
318	Change Numeric Login
319	Update Vendor Parameters
320	Get Payment/Transfer Schedule
321	Explain Payment/Transfer Schedule
322	Payment/Transfer History
323	Dynamic Authentication Address List
324	Set Default Dynamic Authentication Address
325	Generate Dynamic Password
326	Transfer Templates List
327	Add Transfer Template
328	Update Transfer Template
329	Delete Transfer Template
330	Telebank PIN Change
331	Card List
332	Card Lock
333	Card Activation
334	Change Card Description

Transaction Code	Title
335	Cards on Account
336	Create Virtual Card
337	Internet PIN Change via Telebank
338	Card Limits
339	Change Temporary Card Limits
340	Payment/Transfer Schedule
341	Cancel Payment/Transfer Schedule
342	Update Payment/Transfer Schedule
343	Telebank Bill Subamount Payment
344	Telebank Bill Subamount Reject
345	Telebank Bill Payment
346	Telebank Bill Reject
347	Telebank Get Back Office Information
348	Telebank Get Service Point
349	Transfer to Cardless Recipient
350	Service Auto Payment
351	Telebank Transfer
352	Telebank Payment
353	Telebank PrepaidCode
354	Telebank Outside Transfer
355	Telebank P2P Debit
356	Telebank P2P Credit
357	Telebank Precheck
358	Telebank Confirmation
359	Telebank Payment Request
360	Reverse Telebank Transaction
361	Buy Balance
362	Buy MiniStatement
363	Telebank Card Statement
364	Telebank Change EC Status
365	Get Currency Rates
366	Set Extra Authentication Level
367	Set Extra Authentication Address
368	Telebank Transfer Pass
369	Telebank Transfer Debit
370	Telebank Transfer Credit
371	Telebank Send Message to Bank
372	Telebank Read Messages from Bank
373	Telebank Get Customer User Fields

Transaction Code	Title
374	Telebank Set Customer User Fields
375	Telebank Message to Financial Institution
376	Telebank Change Alphanumeric Password
377	Telebank Change Card Alphanumeric Password
378	Calculate CVV2 for Virtual Card
379	Telebank Payment by Template
380	Telebank Payment Return
381	Telebank P2P Pass
382	Telebank Payment Pass
383	Telebank Payment Debit
384	Telebank Payment Credit
385	Telebank Prepaid Pass
386	Telebank Prepaid Debit
387	Telebank Prepaid Credit
388	Telebank Add Abonent
389	Telebank Update Abonent
390	Telebank Delete Abonent
391	Telebank Card Payment by Template
392	Telebank Transfer by Template
393	IPS Registration
394	IPS Registration Confirmation
395	Get IPS Limit
396	IPS Registration Cancellation
397	P2P IPS Transfer
398	Get CAP Challenge
399	Check CAP Token
900	Telebank Transfer Data Inquiry
1301	Get Transaction History
1302	Card PIN Change via Telebank
1303	Create Telebank Customer via Telebank Pass
1304	Create Telebank Customer via Telebank
1305	Link Card to Telebank Customer via Telebank
1306	Link Authentication Card to Telebank Customer via Telebank
1307	Get Virtual Card Products
1308	Get PAN
1309	Account Limits
1310	Set Account Limits
1311	Set Temporary Account Limits
1312	Telebank Bills Request

Transaction Code	Title
1313	Attach Foreign Card To Customer
1314	Detach Foreign Card From Customer
1315	Telebank Get Autopayment Subscription List
1316	Telebank Create Autopayment Subscription
1317	Telebank Edit Autopayment Subscription
1318	Telebank Delete Autopayment Subscription
1319	Telebank Get Person Info
1320	Telebank Set Person Info
1321	Get Card Allowed 3-D Secure Authentication Rule List via Telebank
1322	Set Card 3-D Secure Authentication Rule via Telebank
1323	Telebank Set Avatar
1324	Telebank Update Card Vendor
1325	Customer Messaging Profile Configuration
1326	Get Abonent Information
1327	Customer Messaging Profiles
1328	Get Subscription History
1329	Get Card Tokens
1330	Set Card Token Status
1331	Person Limits
1332	Set Person Limits
1333	Set Temporary Person
1334	Telebank Set Card Contactless State
1335	Telebank Get Avatar
1336	Telebank Change Temporary EC Status
1337	Telebank Get Alias Info
1338	Telebank Create Alias
1339	Telebank Update Alias
1340	Telebank Delete Alias
1341	Telebank Check Secret Word
1342	Telebank Get Devices List
1343	Telebank Get Device History
1344	Telebank Change Device Status
1345	Telebank Set Device User Fields
1346	Telebank Fee Calculation
1347	Telebank Prepare Provisioning Data
1348	Telebank Check Alias
1349	Telebank Get Service Members List
1351	Telebank Set User Defined CVV2

Transaction Code	Title
1353	Telebank Get Allowed Aliases
1354	Telebank Get Card Token Need Confirm
1355	Telebank Card Token Confirmation
1356	Telebank Get ECMerchant
1357	Telebank Detach ECMerchant
1358	Telebank Get Invoice
1359	Telebank Change Invoice Status
1360	Telebank Get Counters
1361	Telebank Set Counters
1362	Telebank Set Alias Permission
1363	Telebank Authorization Check
1364	Telebank Operate DevPIN
1365	Telebank Update Card Metadata
1366	Telebank Token Inquiry
<b>Administrative Transactions</b>	
401	Card Risk Control Disable
402	Change Card Status
403	Clear Card PVV
404	Change Prepaid Code Status
405	Customer Notification on Accounts Balance
406	Customer Notification on Card Expiration Date
407	Account Operation History Request
408	Reset Limit Counter
409	Transactions History Request
410	Inquiry Local Stoplist
411	Add Card to Local Stoplist
412	Modify Local Stoplist
413	Delete Card from Local Stoplist
414	Set Temporary Overdraft on Account
415	Account Balance Request
416	Change Account Status
417	Change Account to Card Status
418	Reset Card EMV ATC
419	Issuer History Request
420	ATM Cardless Message
421	Abonent Delivery Log Request
422	Abonent Command Log Request
423	Change Abonent Details
424	Change Abonent Challenge

Transaction Code	Title
425	Abonent Profile Management
426	Broadcast Message Send
427	Test Message Send
428	Abonent Issuer History
429	Abonent Transaction History
430	Change Temporary Account Limit
431	Get BMS Message Forms
432	Get BMS Message Fields
433	Send BMS Message
434	Get Card User Fields
435	Personal Payments List
436	Payments and Transfers Schedule and History
437	Get Back Office Information
438	Clear Card Password
439	Dynamic Authentication Control
440	Abonent Profile Ddeleting
441	Receive Prepaid Code
442	Abonent Profile Updating
443	Change EC Status
444	Change Card Limit
445	Get Back Office Information by Card
446	VISA Stoplist Processing Back Office
447	MasterCard Stoplist Processing via CST
448	Set Card User Fields
449	Set Account User Fields
450	Set Customer User Fields
451	Get Concomitant Cards
452	Get Account User Fields
453	Get Customer User Fields
454	Add User Field
455	Delete User Field
456	Get Confidential Data
457	Outer Internet Payment System Registration
458	Outer Internet Payment System Registration Confirmation
459	Outer Internet Payment System Registration Cancellation
460	Clear Card IPVV
461	Change Card PVV
462	Change Card IPVV
463	Change Customer Address

Transaction Code	Title
464	CST Service Auto Payment
465	DCC Inquiry
466	Incoming Message
467	Change 3-D Secure Authentication
468	Get Auto Payment History Log
469	Reset Abonent Additional Info
470	Get Card by UID
471	Get UID by Card
472	Create Card UID
473	Delete UID
474	Change Temporary Customer Limit
475	Get Devices List
476	Get Device History
477	Change Device Status
478	Set Device Additional Fields
479	Get Device Additional Fields
480	Get Card Token Need Confirm
481	Card Token Confirmation
482	Set Account Permissible Excess
483	Get Invoice
485	Token Request
486	Change Counter
487	Set Card Contactless State
488	Get CoF Subscriptions
489	Set CoF Subscription
490	Get Aliases
491	Void Transaction
492	Rekey Terminal
493	Update Card Metadata
<b>FIMI Transactions</b>	
501	Logon
502	Change Password
503	Set Reissued Card Variant
504	Set Card to Account Description
505	Get Card Statement
506	Configure CNS Parameters by Card
507	Configure Risk Control Parameters by Card
508	Create Card to Account Link
509	Set Card Profile

Transaction Code	Title
510	Set EMV Card Parameters
511	Get Card Information
512	Set Card Status
513	Get Card Limits
514	Set Card Limits
515	Set Card PVV
516	Reset Bad PIN Tries
517	Set Card Expiration
518	Reset InfoBuyer PIN
519	Reset InfoBuyer Bad PIN tries
520	Set Temporary Limit Values by Card
521	Get Account Information
522	Set Account Status
523	Change Account Balances
524	Get Account Statement
525	Create Account
526	Get Person Information
527	Set Account to Card Status
528	Account Debit
529	Account Credit
530	Reset Card to Account Link
531	Get Retailer List
532	Get Terminal List
533	Get Telebank Customer History
534	Get Account to Card Limits
535	Set Account to Card Limits
536	Create Virtual Card
537	Get Account Limits
538	Set Account Limits
539	Set Temporary Account Limits
540	Get ATM Hardware
541	Get ATM Status
542	Get ATM Fitness
543	Get ATM Media
544	Get ATM Sensors
545	Get ATM Hoppers
546	Get ATM Activity
547	ATM Command
548	Get ATM Audit Journal

Transaction Code	Title
550	POS Request
551	POS State
552	POS Command
553	Get EMV Scripts
554	Add EMV Script
555	Update EMV Script
556	Delete EMV Script
557	Set EMV Script Sequence Number
558	Get Card Profile Limits
559	Set Card Profile Limits
560	Set Telebank Customer Extra Authentication Level
561	Get ATM Events
562	Get Working Key
563	Get PVV/PIN Offset
564	Get CVV
565	Check PVV/PIN Offset
566	Get Correspondent Account Information
567	Modify Correspondent Account
568	FIMI Attach Vendor to Card
569	FIMI Detach Vendor to Card
570	FIMI Reset Card EMV ATC
571	Get Transactions Information
572	Reverse Transaction
573	Get ATM Admin Transactions
574	Get FIMI Transactions
575	Confirm Copy Request to VISA
576	Fee to VISA
577	Text to VISA
578	Get Unit State
579	Unit Command
580	Set User Fields
581	Get Currency Rates
582	Add Currency Rate
583	Modify Currency Rate
584	Delete Currency Rate
585	MasterCard-Global Stoplist Processing
586	VISA Stoplist Processing
587	MasterCard Stoplist Processing
588	Local Stoplist Processing

Transaction Code	Title
589	Adjustment to VISA
590	Representment to VISA
591	Get Back Office Information
592	Create Telebank Customer
593	Get Customer Messaging Service buffer
594	Get Customer Messaging Service archive
595	Get Customer Messaging Service commands
596	Get Issuer History Log
597	Get Issuer History Details
598	Configure CMS Profile
599	Call Cryptographic Function
800	Balance Inquiry
1501	Create Card
1502	Create Person
1503	Get Unit Events
1504	Add Person Confidential Info
1505	Del Person Confidential Info
1506	Change Person Confidential Info
1507	ATM Request
1508	Reset Card Limits Counter
1509	Reset Account Limit Counters
1510	Reset Account to Card Limits Counter
1511	FIMI Change Alphanumeric Password
1512	Reset Bad Dynamic Authentication Password Tries
1513	Get Reports List
1514	Get Reports Requests
1515	Get Report
1516	Delete Report Request
1517	Execute Report
1518	FIMI Change EC Status
1519	BMS Read Message
1520	BMS Send Message
1521	Generate Dynamic Password
1522	Get Prefix Information
1523	Add Telebank Transfer Template
1524	Del Telebank Transfer Template
1525	Adjustment to MasterCard Global
1526	Representment to MasterCard Global
1527	Chargeback to MasterCard Global

Transaction Code	Title
1528	Chargeback Reversal to MasterCard Global
1529	Update Retailer
1530	Add Telebank Payment Template
1531	Del Telebank Payment Template
1532	Local Chargeback
1533	Local Chargeback Reversal
1534	Set Dynamic PVV PIN Offset
1535	Reset Alpha Numeric Password
1536	Create Issuer Objects
1537	Put Event
1538	Get Repository Key
1539	Get IA Configuration
1540	User Defined
1541	Set IA Configuration
1542	Update Person
1543	Get Acct Group Limits
1544	Copy Card Limit Counters
1545	Set Need Change Password
1546	Delete Operator Sessions
1547	Terminal Risk Control
1548	Get FIMI Clerk Addresses
1549	Get Transaction Identifier
1550	American Express Stoplist
1551	Set Card Person
1552	Set Account Person
1553	Set Account Overdraft
1554	Add Telebank Retailer
1555	Delete Telebank Retailer
1556	Get FIMI Clerks
1557	Get VC Products
1558	Get Terminal Additional Fields
1559	Set Terminal Additional Fields
1560	Discover Stoplist
1561	Update Terminal
1562	Get Excluded PIN List
1563	Get Persons Hierarchy
1564	Attach Card To TB Customer
1565	Detach Card From TB Customer
1566	Attach Foreign Card To TB Customer

Transaction Code	Title
1567	Detach Foreign Card From TB Customer
1568	Service Auto Pay
1569	Delete Card
1570	Delete Telebank Customer
1571	Delete Account
1572	Delete Person
1573	Get Card Autopayments
1574	Attach Autopayment To Card
1575	Detach Autopayment From Card
1576	Edit Card Autopayment
1577	Add CMS Abonent
1578	Remove CMS Abonent
1579	Change CMS Abonent
1580	Set Card 3-D Secure Authentication
1589	Stand Alone Reversal
1590	Get Customer Avatar
1591	Set Customer Avatar
1592	Get Issuer Fees for Own Terminals
1593	Set Issuer Fees for Own Terminals
1594	Get Issuer Fees for External Terminals
1595	Set Issuer Fees for External Terminals
1596	Delete Card To Account Link
1597	Assign Key by Terminal
1598	Change Key Status
1599	Add Person CMS Abonent
2500	Remove Person CMS Abonent
2501	Change Person CMS Abonent
2502	Configure Person CMS Profile
2503	Reset Person Bad Dynamic Authentication Password Tries
2504	Get Cumulative Limits
2505	Add Cumulative Limit
2506	Update Cumulative Limit
2507	Delete Cumulative Limit
2508	Reset Cumulative Limit
2509	Get Scheme Item Types
2510	Reset Card Abonent Additional Info
2511	Reset Person Abonent Additional Info
2512	Get Card Tokens Info
2513	Get Card Autopayment History

Transaction Code	Title
2514	Set TB Customer Login
2515	Set Card Token Status
2516	Add Issuer Fees for Own Terminals
2517	Delete Issuer Fees for Own Terminals
2518	Add Issuer Fees for External Terminals
2519	Delete Issuer Fees for External Terminals
2520	Get Person Limits
2521	Set Person Limits
2522	Set Temporary Person Limits
2523	Reset Person Limits Counter
2524	Get Person Group Limits
2525	Get Person Counters
2526	Set Person Counters
2527	Get Account Counters
2528	Set Account Counters
2529	Get Card Counters
2530	Set Card Counters
2531	Attach Cards
2532	Set Card Contactless Interface
2533	Reassign Card Tokens
2534	Create Card Token
2535	Delete Card Token
2536	Update Card Token
2537	Get Alias Info
2538	Create Alias
2539	Delete Alias
2540	Update Alias
2541	Get Devices List
2542	Get Device History Log
2543	Change Device Status
2544	Set Device Additional Fields
2545	Prepare Provisioning Data
2546	Check Alias
2547	Get Service Members List
2549	Send Message by Address
2550	Set Card User Defined CVV2(UdCVV2)
2551	Generate QR Code
2552	Get QR Code Info
2553	Check QR Code Status

Transaction Code	Title
2554	Get QR Code Payload
2555	Merchant Refund
2556	Get Card Token Need Confirm
2557	Card Token Confirmation
2558	Set Card Limits Ranges
2559	Set Account Limits Ranges
2560	Set Person Limits Ranges
2561	Get Arrests
2562	Add Arrest
2563	Delete Arrest
2564	Change Arrest Amount
2565	Set Account Permissible Excess
2566	Create FIMI Clerk
2567	Set FIMI Clerk Activity Status
2568	Get EC Merchant
2569	Change EC Merchant Status
2570	Get Invoice
2571	Change Invoice Status
2572	Get Card/Token Payment Account Reference
2573	Detach EC Merchant from Card
2574	Get PIN
2575	Local Adjustment
2576	Set Account Protected Amount
2577	Set Alias Permission
2578	Get Retailer Limits
2579	Set Retailer Limits
2580	Reset Retailer Limit Counters
2581	Get Retailer Scheme Limits
2582	Get Retailer Group Limits
2583	Set Retailer Group Limits
2584	Reset Retailer Group Limit Counters
2585	Get Retailer Group Scheme Limits
2586	Token Inquiry
2587	Update Card Metadata
2588	Modify QR Code
<b>E-Commerce Transactions</b>	
601	3-D Secure Verify Enrollment
602	3-D Secure Payer Authentication
603	3-D Secure Check Proxy PAN

Transaction Code	Title
604	3-D Secure Get Enrolled
605	3-D Secure Get Dynamic Authentication Address List
606	3-D Secure Add Abonent
607	3-D Secure Update Abonent
611	3D Secure Change Authentication Data
621	3-D Secure Generate Dynamic Password
622	3D-Secure Change EC Status
623	3-D Secure Prepare Authorization
624	3-D Secure Process Authorization
625	3-D Secure Request For External Authentication
626	3-D Secure Check External Authentication
<b>CMS Transactions</b>	
701	CMS Payment
702	Customer change card status
703	Customer messaging on accounts balance
704	Customer messaging on card expiration date
705	Customer messaging on commands list
706	Customer messaging on challenge request
707	Customer messaging on switch notification
708	Customer messaging on forwarding
709	Customer notification on approved transaction
710	Customer notification on declined transaction
711	Customer notification on changing balance
712	Customer messaging profile modification
713	Customer messaging on bank message receipt
714	Customer messaging on bank message delivery
715	Customer messaging on bank message generation
716	Customer messaging set card limit temporary maximum
717	Customer messaging on personal payments list
718	Customer messaging on card limits list
719	Customer notification on changing bonus/dept field in refresh
720	Customer messaging on currency rates
721	Customer messaging on currency list
722	Customer messaging on abonent sign
723	Customer messaging on operation history
724	Customer messaging on operation list
725	Customer messaging on payment order
750	CMS Payment Pass
751	CMS Payment Debit

Transaction Code	Title
752	CMS Payment Credit
753	CMS Payment Precheck
754	CMS Payment Confirmation
755	CMS P2P Pass
756	CMS P2P Debit
758	CMS P2P Credit
758	CMS Payment Request
759	CMS Get Abonent Information
799	Customer Messaging Traffic

### 3.9 Authorizer Response Codes

*Main Authorizer Response Codes (RespCode):*

Response Code	Title	Description
0	None	–
1	Approved	OK
2	Approved Partial	Transaction is approved for the partial amount
3	Approved Purchase Only	Transaction is approved for the <i>Purchase</i> amount; the <i>Cashback</i> amount is not approved
4	Postponed	Transaction is postponed, it will be processed later
6	Strong customer authentication required	Strong customer authentication is required for the transaction execution.
7	Need Checker's confirmation	The checker confirmation is required
8	Telebank customer already exists	Telebank customer already exists
9	Should select virtual card product	Virtual card product should be selected
10	Should select account number	Account number should be selected
11	Should change PVV	PIN should be changed
12	Confirm payment precheck	The results of payment verification in the payment online acceptance system should be confirmed
13	Select bill	Select the bill to be paid
14	Customer confirmation requested	Customer confirmation is requested
15	Original transaction not found	Original transaction is not found (for example, while receiving an electronic slip from POS terminal; or an original transaction for the reversal)
16	Slip already received	Slip has already been received
17	Personal information input error	Error by entering payment attributes
18	SMS/EMail dynamic password requested	SMS/Email dynamic password is requested
19	DPA/CAP dynamic password requested	DPA/CAP dynamic password is requested
20	Prepaid code not found	Prepaid code is not found
21	Corresponding account exhausted	Agent bank correspondent account is exhausted

Response Code	Title	Description
22	Acquirer limit exceeded	Merchant acquiring limit has already been reached or exceeded
23	Cutover in process	Cutover is being performed
24	Dynamic PVV Expired	Dynamic PVV expired
25	Weak PIN	Weak PIN
26	External authentication required	External authentication is requested
27	Additional data required	Additional data is required
29	Closed account	The account is closed
30	Blocked	Blocked
40	Lost card	The card is lost
41	Stolen card	The card is stolen
49	Ineligible vendor account	Ineligible vendor account
50	Unauthorized usage	The card unauthorized usage
51	Expired card	The card is expired
52	Invalid card	The card is invalid
53	Invalid PIN	Invalid PIN
54	System error	System error
55	Ineligible transaction	Ineligible transaction
56	Ineligible account	Ineligible account
57	Transaction not supported	The transaction is not supported
58	Restricted card	The card is restricted (the operation is prohibited)
59	Insufficient funds	There are not enough money on the account
60	Uses limit exceeded	The card usage limit is exceeded
61	Withdrawal limit would be exceeded	Withdrawal limit will be exceeded
62	PIN tries limit was reached	Invalid PIN tries limit is reached
63	Withdrawal limit already reached	Withdrawal limit is already reached
64	Credit amount limit	Deposit limit is reached
65	No statement information	No statement information provided
66	Statement not available	Statement is not available (prohibited)
67	Invalid amount	Invalid amount
68	External decline	Transaction is declined by the external host
69	No sharing	The card is not serviced at the terminal

Response Code	Title	Description
71	Contact card issuer	Contact the issuer
72	Destination not available	The authorizer is not available
73	Routing error	Routing error
74	Format error	Format error
75	External decline special condition	The transaction is declined by the external host by special condition (the card owner is suspected of fraud)
80	Bad CVV	Invalid CVV
81	Bad CVV2	Invalid CVV2
82	Invalid transaction	The transaction with such attributes is prohibited
83	PIN tries limit was exceeded	Bad PIN tries limit is already exceeded
84	Bad CAVV	Invalid 3D Secure Cardholder Authentication Verification Value
85	Bad ARQC	Invalid cryptogram of EMC card application
90	Approve administrative card operation inside window	Operation by the administrative card in the window is approved
91	Approve administrative card operation outside of window	Operation by the administrative card out of the window is approved
92	Approve administrative card operation	Operation by the administrative card is approved
93	Should select card	Card should be selected
94	Confirm Issuer Fee	Issuer fee confirmation is required
95	Insufficient cash	Insufficient funds
96	Approved frictionless	Approved as Frictionless
98	Invalid merchant	Invalid merchant

**Detailed Authorizer Response Codes (DetailedRespCode):**

Response Code	Description
1	PIN is absent but it is required according to the prefix settings.
2	PIN is absent but it is required for the PIN change transactions.
3	PIN is absent but it is required according to the authorization scheme settings.
4	PIN is present but it matches the previous value.
5	Pre-Purchase Complete amount exceeds the maximum allowed value of the Pre-Purchase amount.
6	Transaction duplicate

Response Code	Description
7	Transaction data did not match those in DB.
8	A card has no subscribers.
9	Card expiration dates on Track and in DB differ.
10	The impact on the account with requirements defined in the algorithmix function is impossible.
11	Invalid CVV2 tries limit is exceeded.
12	Invalid birth date.
13	Frictionless process.
14	Dynamic password ID does not match password ID during the generation.
15	Transaction is declined for the Fallback decline reason.
16	Accepted/waits for processing.
17	Subscriber for OTP is not found during the token activation process.
18	Customer cannot be created as the active Telebank user is present.
19	Invalid secret word.
20	PIN is required.
21	Invalid number of the recipient document.
22	Invalid MCC.
23	A card is not allowed for participating in the tokenization process.
24	QR Code lifetime has expired.
25	Low violation
26	Alias is deleted.
27	Alias is not found.
28	Cardholder is dead.
29	Text login has been changed.
30	Card status is other than Open, VIP, Open Domestic.
31	Sender data does not correspond to the recipient data.
32	Text login already exists.
33	Merchant account is not found.
34	Merchant account is blocked.
35	Account token is not found.
36	Number of the OTP send attempts is expired.
37	Time before the repeated OTP sending is not expired.
38	Invalid token status
39	OTP has expired.
40	Number of the OTP entry attempts is expired.

Response Code	Description
41	Invalid OTP
42	Invalid TVV
43	RiskData parameters differ.
44	No response from the external system is received.

### 3.10 ATM Control Commands

Value	Name	Description
11	Open	Open ATM
12	Close	Close ATM
13	Reopen	Reopen ATM
21	UpdateCfg	Update configuration
22	UploadCfg	Upload configuration
24	DownloadEJ	Download EJ
26	UpdateRoutingCache	Update routing table
30	SelfTest	Perform self-test, only for ATMs with the type 2 (ATM DDC)
31	FetchConfig	Fetch ATM configuration
32	FetchSupplies	Fetch counters
33	FetchAll	Fetch all
34	ResetStatuses	Reset the devices statuses, only for ATMs with the type 2 (ATM DDC)
40	ResetRetainedCards	Reset the cards counter
41	BalanceCurCash	Balancing with current amounts against Central counters
42	BalanceStdCash	Cash replenishment and balancing with standard amounts
43	BalanceAtmCash	Cash replenishment and balancing against ATM counters
45	AdjustHopper	ATM hopper to which an increase/decrease is made
46	BalanceEmpty	Complete cash removal and balancing
48	DisableCassete	Block cassette
49	EnableCassete	Unblock cassette
231	Terminal Acquiring Control Disable	Acquiring control disabling
500	Release Deposit	Clear deposit cassettes
501	Change TMK	Change TMK on the ATM (RKL)
606	TR34 Bind	Bind the terminal to the host to perform RKL via the ASC X9 TR-34 protocol
607	TR34 Unbind	Unbind the terminal from the host to perform RKL via the ASC X9 TR-34 protocol

### 3.11 POS Terminal Control Commands

Value	Name	Description
201	POSActivate	Activate POS terminal
202	POSDeactivate	Deactivate POS terminal
203	POSActivateFIID	Activate all the institution terminals
204	POSDeactivateFIID	Deactivate all the institution terminals
205	POSActivateRetl	Activate all the retailer terminals
206	POSDeactivateRetl	Deactivate all the retailer terminals
211	POSUpdateCfg	Update terminal configuration
212	POSUpdateCfgGroup	Update configuration of all the terminals belonging to the configuration group of the terminal for which a command was initiated
221	POSReloadHotList	Reload Hot List to the terminal
231	Terminal acquiring control disable	Disable the acquiring control

### 3.12 Transaction Types

Transaction Type	Title	Description
100	AuthRequest	Authorization transaction
120	AuthAdvice	Authorization advice
200	Request	Financial transaction
220	Advice	Financial advice
420	Reversal	Reverse – reversal of the previously completed transaction AuthRequest/AuthAdvice/Request/Advice
422	IssuerRequest	1) Dispute transaction received from the external payment system (Chargeback) (if transaction code!={410, 411, 412, 413, 585, 586, 587, 588) 2) Transaction with stoplists (if transaction code ={410, 411, 412, 413, 585, 586, 587, 588})
442	IssuerReversal	Chargeback reversal
600	AdminAdvice	Administrative advice
999	Admin	Administrative transaction

### 3.13 Visa Stoplist Regions

Region Code	Title
0	Do not list in any Card Recovery Bulletin
1	U.S. (California, Hawaii, Nevada)
2	U.S. (Alaska, Arizona, Idaho, Oregon, Utah, Washington)
3	U.S. (Colorado, Iowa, Kansas, Minnesota, Montana, Nebraska, New Mexico, North Dakota, South Dakota, Wyoming)
4	U.S. (Oklahoma, Texas)
5	U.S. (Illinois, Indiana, Kentucky, Michigan, Missouri, Ohio, West Virginia, Wisconsin)
6	U.S. (Alabama, Arkansas, Louisiana, Maryland, Mississippi, North Carolina, South Carolina, Tennessee, Virginia, Washington, D.C.)
7	U.S. (Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont)
8	U.S. (Delaware, New Jersey, New York, Pennsylvania)
9	U.S. (Florida, Georgia)
A	Included all countries in the Asia-Pacific region
B	Africa and part of the Middle East (included countries that are part of Visa subregion 3 and of the Europe, Middle East, Africa region)
C	All Canadian CRB regions (regions 1 through 3)
D	National CRB indicator
E	Europe and of the Middle East (includes countries in the Europe, Middle East, Africa region not classified as part of CRB region B)
F	Includes all countries in the Latin America Region
X	All U.S. CRB regions (regions 1 through 9)
Y	All non-U.S. CRB regions (regions A,B,C,E,F)
Z	All CRB regions

### 3.14 Mastercard Europe/Global Stoplist Regions

MasterCard	Region Code	Title
Europe	0	Mastercard's centralized stand-in account file
Europe/Global	1	USA
Europe/Global	A	Canada
Europe	B	Latin America
Global	B	Latin America/Caribbean
Europe/Global	C	Asia/Pacific
Europe/Global	D	Europe
Europe	E	Middle East/Africa
Global	E	South Asia/Middle East/Africa

### 3.15 Messaging Service Command Codes

Command Code	Title	Description
0	CommandList	List of commands
1	BalanceRequest	Balance request
2	ChallengeRequest	Challenge request
11	DisableNotifications	Disabling notifications
12	EnableNotifications	Enabling notifications
13	CardDeactivation	Card deactivation
14	CardActivation	Card activation
15	ItemNotificationOn	Switch on item notification
16	ItemNotificationOff	Switch off item notification
17	PersonalPaymentList	List of personal payments
18	PersonalPayment	Personal payment
19	ItemMinAmountChange	Changing the item minimum amount
20	BankMessageInquiry	Bank message request
21	BankMessageCreation	Creating a message to the bank
22	LimitChanging	Changing limit
23	LimitList	List of limits

### 3.16 Messaging Service Command Decline Reason Codes

Command Code	Title	Description
0	Ok	Command is processed
1	InvalidAddressTemplate	Invalid address template
2	EmptyCommand	Empty command
3	CardNotFound	Card is not found
4	NeedPANParam	PAN parameter is required
5	InvalidPANParam	Invalid PAN parameter
6	Insufficient PAN param	Insufficient digits in the PAN parameter
7	UnknownCommand	Unknown command
8	NeedChallenge	Challenge is required
9	RequestChallenge	Challenge request is required
10	ChallengeExpired	Challenge has expired
11	InvalidChallenge	Invalid challenge
12	InvalidCardStatus	Invalid card status
13	TranGenerationError	Error by generating a transaction
14	TranDeclined	Transaction is declined
15	ReplyArchError	Error by archiving a response
16	NoValidAccts	Valid accounts are not found
17	InvalidAmountParam	Invalid parameter of the amount
18	InadmissibleAmount	Inadmissible amount
21	NeedPaymentAmount	Payment amount is not defined
22	NeedPersonalPaymentName	Personal payment name is required
23	UnknownPersonalPaymentName	Personal payment name is unknown
24	NeedMBRParam	MBR parameter is required
25	InvalidMBRParam	Invalid MBR parameter
26	NeedLimitName	Limit name is required
27	UnknownCardLimit	Unknown limit
28	NeedBankMessageld	Bank message identifier is required
29	InvalidBankMessageld	Invalid identifier of the bank message
30	RemovedBankMessageld	Bank requested message is removed
31	InvalidPortNumber	Invalid port number
100	PaymentFormatError	Invalid format of the payment personal information
101	AuthorizerNotAvailable	Authorizer is not available

Command Code	Title	Description
102	AuthorisationFailed	Authorization error
103	AuthorisationTimeout	Authorization timeout
999	SystemError	System error

## 3.17 Issuer History Objects and Attributes

### 3.17.1 Objects Description

Object	Description
CRD_CARD	Table of cards
CRD_LIMIT	Card/Telebank customer limits
CRD_CARDLIMITSRANGES	Card/Telebank customer limit ranges
CRD_EMVCARDTAG	Card EMV-tags
CRD_EMVSCRIPT	Card EMV-scripts
CRD_MESSAGING	Messaging service abonents of card/Telebank customer
CRD_ACCOUNT	Table of accounts
CRD_ACCOUNT2CARD	Card-to-account links
CRD_ACCT2CARDLIMIT	Account-card link limits
CRD_ACCTLIMIT	Account limits
CRD_ACCTLIMITSRANGES	Account limit ranges
TLB_CUSTOMER	Telebank customer
CRD_CARDPASSWORD	Alphanumeric password of card/Telebank customer
CRD_PERSON	Table of customers
CRD_PERSONINFO	Customer personal data
CNS_PERSONABONENT	Customer messaging service abonents
CRD_PERSONLIMIT	Customer limits
CRD_PERSONLIMITSRANGES	Customer limit ranges
CRD_PERSONUSERFIELDS	User-defined customer fields
CRD_ACCTUSERFIELDS	User-defined account fields
CRD_CARDUSERFIELDS	User-defined card/Telebank customer fields
CRD_TOKENLIMIT	Token limits
CRD_ARRESTS	Arrests
CRD_TOKEN	Card token

## 3.17.2 Attributes Description

Object	Attribute	Bit	Description
CRD_CARD	STATUS	1	Status
	EXPDATE	2	Expiration date
		3	<b><i>PIN verification parameters</i></b>
	PVV	3	PIN verification value
	PREVPVV	3	Previous PIN verification value
	IPVV	3	Internet PIN verification value
	PVKI	3	PIN Verification Key Index
	PREVPVKI	3	Previous PIN verification key index
		4	<b>Last EMV ATC value</b>
	LASTATC	4	Last EMV ATC value
	PREVLASTATC	4	Previous last EMV ATC value
	CARDPROFILE	5	Card profile
		6	<b><i>Telebank customer authentication</i></b>
	AUTHCUSTPAN	6	Card number of the Telebank customer
	AUTHCUSTMBR	6	Telebank card owner
		7	<b><i>EMV Offline spending control</i></b>
	OSC_ALGO	7	Algorithm of EMV Offline spending control
	OFFLINEPENDING	7	EMV Offline Pending
	OFFLINELIMIT	7	Offline spending limit
	ECSTATUS	8	Status of participation in E-Commerce
		10	<b>E-Commerce authentication settings</b>
	ECUSECARDSETTINGS	10	Using card settings for E-Commerce authentication
	ECNEEDSTATICAUTH	10	Using static authentication for E-Commerce
	ECNEEDDYNPWDAUTH	10	Using dynamic authentication for E-Commerce
	ECNEEDCAPAUTH	10	Using CAP authentication for E-Commerce
	ECNEEDTOKENAUTH	10	Using authentication by token for E-Commerce
CRD_LIMIT	MAXVALUE	1	Maximum limit value
		2	<b><i>Limit temporary maximum</i></b>

Object	Attribute	Bit	Description
	TMPMAXVALUE	2	Limit temporary maximum value
	TMPMAXSTARTTIME	2	Temporary maximum start time
	TMPMAXEXPIRATION	2	Temporary maximum expiry date
	TMPMAXPERIODTYPE	2	Temporary maximum period type
		3	<b>Period</b>
	PERIOD	3	Period
	PERIODTYPE	3	Limit period type
	DATETIME4LIMRESET	3	Date and time of the limit counter reset
		4	<b>Limit counter reset</b>
	CURVALUE	4	Limit counter
CRD_CARDLIMITSRANGES		5	<b>Limit counter</b>
	CURVALUE	5	Limit counter
		6	<b>Limit range</b>
	STARTDATE	6	Range start time
CRD_EMVCARDTAG	ENDDATE	6	Range end time
	MAXVALUE	6	Maximum value in the range
CRD_EMVSCRIPT	TAG	1	Tag
	REQUESTEDVALUE	2	Tag requested value
CRD_MESSAGING	EMVTYPET	1	EMV-script type
	EMVADDRESS	2	Address
	EMVVALUE	3	Value
	EMVTEMPLATE	4	Template
CRD_ACCOUNT	CHANNEL	1	Messaging channel
	ADDRESS	2	Abonent address
	SCHEME	3	Messaging scheme
	STATUS	1	Status
		2	<b>Balance (manually changed)</b>
		3	<b>Balance (manually or automatically changed)</b>
	AVAILBALANCE	2,3	Available balance
	LEDGERBALANCE	2,3	Ledger balance
	OVERDRAFTLIMIT	4	Overdraft limit
		5	<b>Temporary overdraft</b>
	TMPOVERDRAFTEXPIRATION	5	Expiration date of the temporary overdraft

Object	Attribute	Bit	Description
	TMPOVERDRAFT	5	Temporary overdraft value
	TMPOVERDRAFTPERIOD TYPE	5	Temporary overdraft period type
	DROPTMPOVERONREFRESH	5	Reset temporary overdraft in Refresh
	BONUS	6	Bonus/Debt
CRD_ACCOUNT2CARD	ACCTSTATUS	1	'Card-Account' link status
	ACCTDESCR	2	'Card-Account' link description
CRD_ACCT2CARDLIMIT	MAXVALUE	1	Limit maximum value
		2	<b>Period</b>
	PERIOD	2	Period
	PERIODTYPE	2	Limit period type
		3	<b>Limit counter</b>
	CURVALUE	3	Limit counter value
	DATETIME4LIMRESET	3	Date and time of the limit counter reset
CRD_ACCTLIMIT	MAXVALUE	1	Limit maximum value
		2	<b>Period</b>
	PERIOD	2	Period
	PERIODTYPE	2	Limit period type
		3	<b>Limit counter reset</b>
	CURVALUE	3	Limit counter
	DATETIME4LIMRESET	3	Date and time of the limit counter reset
		4	<b>Limit counter change</b>
	CURVALUE	4	Limit counter
		5	<b>Limit temporary maximum</b>
	TMPMAXVALUE	5	Limit temporary maximum value
CRD_ACCTLIMITSRANGES	TMPMAXPERIODTYPE	5	Limit temporary maximum type
	TMPMAXEXPIRATION	5	Temporary maximum period
		6	<b>Limit range</b>
	STARTDATE	6	Range start time
TLB_CUSTOMER	ENDDATE	6	Range end time
	MAXVALUE	6	Maximum value in the range
	STATUS	1	Status
	EXPDATE	2	Expiration date
		3	<b>PIN verification parameters</b>

Object	Attribute	Bit	Description
	PVV	3	PIN verification value
	PREVPVV	3	Previous PIN verification value
	IPVV	3	Internet PIN verification value
	PVKI	3	PIN verification key index
	PREVPVKI	3	Previous PIN verification key index
	PVV	3	PIN verification value
	IPVV	3	Internet PIN verification value
	NUMERICLOGIN	4	Numeric Login
	TEXTLOGIN	5	Text Login
	EXTRAAUTHLEVEL	6	Extra authentication type
CRD_CARDPASSWORD	PASSWORDTOKEN	9	Alphanumeric password
CRD_PERSON	PASSWORD	1	Customer secret word
CRD_PERSONINFO		2	<b><i>Customer personal data</i></b>
	VALUE	2	Answer to the control question
	ISALLOWEDCST	2	Question can be used in CST
	ISALLOWEDADS	2	Question can be used in E-Commerce
	ISALLOWEDTB	2	Question can be used in Telebank
CNS_PERSONABONENT	CHANNEL	1	Messaging channel
	ADDRESS	2	Abonent address
	SCHEME	3	Messaging scheme
CRD_PERSONLIMIT	MAXVALUE	1	Maximum limit value
		2	<b><i>Period</i></b>
	PERIOD	2	Period
	PERIODTYPE	2	Limit period type
		3	<b><i>Limit counter reset</i></b>
	CURVALUE	3	Limit counter value
	DATETIME4LIMRESET	3	Date and time of the limit counter reset
		4	<b><i>Limit counter change</i></b>
	CURVALUE	4	Limit counter
		5	<b><i>Limit temporary maximum</i></b>
	TMPMAXVALUE	5	Limit temporary maximum value
	TMPMAXPERIODTYPE	5	Limit temporary maximum type
	TMPMAXEXPIRATION	5	Temporary maximum period

Object	Attribute	Bit	Description
CRD_PERSONLIMITS RANGES		6	<i>Limit range</i>
	STARTDATE	6	Range start time
	ENDDATE	6	Range end time
	MAXVALUE	6	Maximum value in the range
CRD_PERSONUSER FIELDS	TEXTVALUE	3	Text value of the user-defined customer field
CRD_ACCTUSERFIELDS	TEXTVALUE	7	Text value of the user-defined account field
CRD_CARDUSERFIELDS	TEXTVALUE	7	Text value of the user-defined Telebank customer field
	TEXTVALUE	11	Text value of the user-defined card field
CRD_ARRESTS	AMOUNT	1	Hold amount in the arrest currency
	STARTDATE	2	Start time
	ENDDATE	3	End time
	AMOUNTHOLD	4	Arrest amount
CRD_TOKEN	STATUS	1	Token status

## 3.18 System Unit Dictionaries

### 3.18.1 Unit, Terminal Types and Unit Subtypes

Unit Type		Terminal Type/Unit Subtype		Group
Value	Description	Value	Description	
1	ATM	1	NDC+	Device Drivers
		2	DDC	
2	POS	101	TPTP	Device Drivers
		103	TITP	
		106	NAC	
3	CRT	—	—	
4	VTBI	301	TeleBank Internet	Hosts and Authorizers
		302	TeleBank Phone	
		303	TeleBank Voice	
		304	TeleBank Info Seller	
		305	FIMI	
		306	E-Commerce	
6	BSP	—	—	
7	American Express	—	—	
8	DinaCard	—	—	
9	W4	—	—	
10	BancNet	—	—	
11	Transmaster	—	—	
12	BASE 24 ISO	—	—	
13	VISA	—	—	
14	DINERS	—	—	
15	MC-Europe	—	—	
16	TRANS24	—	—	
17	TranzWare Inter Connect (TIC )	—	—	
18	DHI	—	—	
19	Stand-In Authorizer	—	—	
20	Core Banking Authorizer	—	—	
25	Splitter	—	—	
50	E-Port	—	—	
51	M-Kassa	—	—	

Unit Type		Terminal Type/Unit Subtype		Group
Value	Description	Value	Description	
52	Postilion	—	—	
53	ArkSys	—	—	
54	American Express ISO	—	—	
55	City	—	—	
56	BASE24es	—	—	
57	China Union Pay	—	—	
58	CyberPlat	—	—	
59	BanknetVN	—	—	
60	HTV	—	—	
61	123	—	—	
62	NSL	—	—	
63	SBERCART	—	—	
64	Pulse	—	—	
65	TPII	—	—	
66	MegaFon	—	—	
67	EVN	—	—	
68	JCB	—	—	
69	VTU	—	—	
70	NEO	—	—	
71	OST24	—	—	
72	OASIS	—	—	
73	Electra Prepaid	—	—	
74	COP	—	—	
75	UDPI	—	—	
76	SUDAPAN	—	—	
77	NAPS	—	—	
78	FIS	—	—	
79	SPAN	—	—	
80	SmartLink	—	—	
81	SYB	—	—	
82	Bersama	—	—	
83	TecBan	—	—	
84	WebMoney	—	—	
85	Cortex	—	—	
86	eKassir	—	—	

Unit Type		Terminal Type/Unit Subtype		Group
Value	Description	Value	Description	
87	Discover	—	—	
88	PowerCard	—	—	
89	UEC	—	—	
90	GIS GMP	—	—	
91	Beeline	—	—	
92	MTS	—	—	
93	QIWI	—	—	
94	Rapida	—	—	
71	OST24	—	—	
101	Elecsnet	—	—	
102	MC-Global	—	—	
103	PayFair	—	—	
104	NSPK	—	—	
105	NBC	—	—	
106	VTS	—	—	
107	NYCE	—	—	
108	Visa ADS	—	—	
109	NETC	—	—	
110	Mastercard Transfers Hub	—	—	
111	BORQ	—	—	
112	NSPK FPS	—	—	
113	NSPK NIPF	—	—	
114	NSPK FPS TSC Agent	—	—	
201	TWFA OFP	—	—	
21	Batch Processor	1	TLG Archiving	Service Units
		2	Refresh Batch Processor	
		3	Hot Lists Batch Processor	
		4	Telebank Transaction Generation	
		5	Report Generation	
		6	Event Notification Service	

Unit Type		Terminal Type/Unit Subtype		Group
Value	Description	Value	Description	
		7	Prepaid Codes Processing	
		8	Messaging Service	
		9	Batch File Authorization	
		10	ATM Audit Journal Loading	
		11	Data Downloading	
		12	Bank Messaging	
		13	Customer Messaging Generator	
		14	Extract Data Unloader	
31	Cryptoserver	—	—	
41	Bridges	1	Connects Commutation Bridge	
		2	Packets Commutation Bridge	
42	XOT Bridge	—	—	
43	Monitor	—	—	
44	System Maintenance	—	—	
45	Job Scheduler	—	—	
46	ISO Bridge	—	—	
47	Database Maintenance	—	—	

### 3.18.2 Unit States Bit Mask

The unit state is identified by the transferred value. It is a set of bits, the state is decrypted using bits depending on the unit type. If the unit state value is NULL, the unit has been unloaded from the instance.

Type/ Subtype	Bit	Description
All (except for 42)	NULL	The unit is unloaded from the instance
	1	Started
	2-3	Required unit state:  0 – stop 1 – start 2 – restart
1/1; 1/2; 2/101; 2/103; 3; 4/301; 4/303; 4/304; 4/305; 4/306; 6; 7; 12; 14; 15; 17; 18; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 101; 102; 21/8; 84; 85; 86; 87; 88; 89; 103; 104; 105; 107; 109; 111; 113	4	Connected
1/1; 1/2	5	Opened
	6-7	Required ATM state:  0 – close 1 – open 2 – reopen
	8-9	Configuration. Requested:  1 – refresh 2 – full upload
	10-11	Requested information on:  1 – supplies 2 – configuration 3 – all
	12-24	Full upload. The range is 1 - 1000

Type/ Subtype	Bit	Description
8; 9; 10; 11; 14; 15; 17; 18; 52; 53; 54; 56; 57; 59; 60; 61; 101; 102; 85; 87; 88; 89; 103; 104; 105; 107; 109; 111; 113	5	Sign On
8; 9; 11; 14; 15; 17; 18; 52; 53; 54; 56; 57; 59; 60; 61; 101; 102; 85; 87; 88; 89; 103; 104; 105; 107; 109; 111; 113	6	Required state of Sign On
8	7	Advice On
	8	Required state of Advice On
12; 16	5	The channel is operable
	6	Sign On
	8	Required state of Sign On
13	4	SMS/Common – connected
	5	BASE I – connected
	6-7	SMS/Common Sign On: 1 – SMS 2 – BASE1 3 – BOTH
	8-9	Required state of SMS/Common Sign On: 1 – SMS 2 – BASE1 3 – BOTH
	10-11	SMS/Common Advice On: 1 – SMS 2 – BASE1 3 – BOTH
	12-13	Required state of SMS/Common Advice On: 1 – SMS 2 – BASE1 3 – BOTH
	14	Totals should be requested.

Type/ Subtype	Bit	Description
	15-16	BASE I Sign On: 1 – SMS 2 – BASE1 3 – BOTH
	18-19	Required state of BASE I Sign On: 1 – SMS 2 – BASE1 3 – BOTH
	20-21	BASE I Advice On: 1 – SMS 2 – BASE1 3 – BOTH
	22-23	Required state of BASE I Advice On: 1 – SMS 2 – BASE1 3 – BOTH
	24	The Suppress Inquiry mode
	25	Required the Suppress Inquiry mode
17; 52;59	7	PIN key change is required
17;52	8	PIN key change
56	7	PIN key change for the incoming requests is required
	8	PIN key change for the incoming requests
	9	PIN key change for the outgoing requests is required
	10	PIN key change for the outgoing requests
61	7	PIN key change for outgoing requests
	8	PIN key change for the incoming requests is required
	9	PIN key change is required
	10	MAC key change is required
57	7	PIN key change is required
	8	PIN key change
	9	MAC key change is required
	10	MAC key change
19; 20	4	The record of transaction decline reason is enabled
51	5	The list of operators is requested
21/xx	9-21	The procedure progress. The range is 1 – 1000
21/1	4	Executing the Cutover
21/14	5	Executing the Extract
21/2	4	Executing the Refresh

Type/ Subtype	Bit	Description
21/3	4	Loading Hot Lists
	5	Comparing the Hot List versions
21/4	4	Generating Telebank transactions
21/5	4	Generating reports
21/7	4	Downloading Prepaid-codes
	5	Downloading confirmation from the provider
	6	Unloading files from the provider
	7	Downloading the files of revoking codes from provider
21/6; 21/8	5	Logical connection is established
21/9	4	Authorizing the batch file
21/10	4	Loading the ATM audit journal
21/11	4	Unloading data
21/12	4	Receiving messages
	5	Transferring messages
31	4	Security Module #1
	5	Security Module #2
	6	Security Module #3
	7	Connected to Security Module #1
	8	Connected to Security Module #2
	9	Connected to Security Module #3
	10	Switched to backup
41/1	3	'Bridge' is organized
	5	Connection to the external port is established
	6	Connection to the internal port #1 is established
	7	Connection to the internal port #2 is established
	8	Connection to the internal port #3 is established
41/2	5	Connection to the external port is established
42	1-2	Required unit state: 0 – stop 1 – start 2 – restart
	3	Connection is established
43	4	Updating
	5-17	Update progress
44	4	Daily cleaning
	5	Periodical cleaning

Type/ Subtype	Bit	Description
102	7	PIN key change for outgoing requests is required
	8	PIN key change for outgoing requests
	9	Advice On
	10	Required state of Advice On

## 3.18.3 Commands for Unit

Type/ Subtype	Parameters	Command Codes	Description
All	—	1	Start
	—	2	Stop
	—	191	Load
	—	192	Unload
	Parameter 1: 1-2 bit – Trace Level: 0 – No tracing; 1 – Errors only; 2 – Errors and warnings; 3 – Trace all; 3 bit – Write Trace to File; 4 bit – Tracing parameters for: 0 – unit; 1 – port; (for the type 13 only: If 4 bit = 1, then 5 bit set in 1 means that tracing parameters are defined for the port #2.)	5	Set tracing parameters
	—	3	Restart
	—	11	Open
	—	12	Close
	—	13	Reopen
	—	31	Fetch configuration
1/1; 1/2	—	32	Fetch supplies
	—	33	Fetch all
	—	21	Update configuration
	—	22	Configuration full upload
	Parameter 1: 1 – Change business day when balancing; 0 – Do not change business day when balancing;	43	Adjustment and balancing against ATM counters
		42	Adjustment and balancing with standard amounts
		41	Balancing with current amounts against Central counters
		46	Cash removal and balancing
	Parameter 1: 1-27 bit – Number of notes (amount); 28-30 bit – Hopper number (1-6); 31 bit: 1 – Subtract notes 0 – Add notes	45	Change the number of notes in the hopper

Type/ Subtype	Parameters	Command Codes	Description
	Parameter 1: 28-30 bit – Hopper number (7);	45	Cash replenishment of deposit hopper
	Parameter 1: 1-3 bit – Hopper number (1-4);	48 49	Block hopper Unlock hopper
	Parameter 1: 1-8 bit – bill denomination status: 0 – Do not use 1 – For deposit 2 – For deposit and dispense from 9 bit – bill denomination ID	47	Setting bill denomination status
	—	501	Change TMK
8; 9; 11; 12; 13; 14; 15; 16; 17; 18; 52; 53; 54; 56; 57; 101; 102	Parameter 1 (for the type 13 only): 1-2 bit – Sign: 1 – SMS 2 – BASE I 3 – BOTH 3 bit – channel: 0 – SMS/Common Station 1 – BASE I	51 52 53 54	Sign On Sign Off Advice On Advice Off
8; 13	—	58	Switching the Core Banking authorizer to the Inquiry mode
20	—	61	Key exchange (type 17, 52, 56 – PIN change); type 57 – PIN and MAC change)
12; 13; 17; 52; 56; 57	—	62 63	Repeat the key exchange Key verification
12; 13	—	56 57	Enter the Suppress Inquiry mode Exit from the Suppress Inquiry mode
13	Parameter 1: 1 bit – Totals type: 0 – Current 1 – Previous 2 bit – Network: 0 – VISA 1 – Plus	55	Request totals
19; 20	—	91 92	Enable the record of transaction decline reason Disable the record of transaction decline reason
51	—	55	Request operators list
21/1	—	71 72	Perform the archiving Execute the Cutover

Type/ Subtype	Parameters	Command Codes	Description
21/14	Parameter 1: Transactions batch number.  Parameter 2: Financial institution ID. If NULL, Extract is performed for all the institutions defined for the unit.	77	Execute the Extract
21/2	Parameter 1: Financial institution ID. NOT NULL.	75	Execute the Refresh
21/3	Parameter 1: Upload source ID	73	Download Hot Lists
	—	74	Compare the Hot List versions
21/4	—	76	Generate the currently scheduled Telebank-transactions
21/5; 21/11	Parameter 1: Identifier of the reports generation schedule (for the type 21/11:identifier of the data unload schedule)	78	Generate report (for the type 21/11: unload data)
21/7	Parameter 1: Provider ID	101	Download Prepaid-codes
		102	Download provider confirmations
		104	Download the files of revoking codes from provider
21/9	—	103	Download and authorize the files of batch authorization
21/10	ATM ID	111	Unload Audit journal
21/12	—	80	Obtain bank messages
	—	79	Send messages to a bank
31	—	83	Switch to the backup cryptoserver
	—	84	Switch to the general cryptoserver
	Parameter 1: Security Unit number (1-3)	82	Allow the Security Module
		81	Prohibit the Security Module
41/1	—	15	Reconnect
44	—	120	Start daily cleaning
	—	121	Start periodical cleaning
4/305	—	354	Reread unit settings
4/305	—	356	Scan the folder with the incoming files
1/1, 1/2, 2/101, 2/103, 2/106	—	26	Update routing table

Type/ Subtype	Parameters	Command Codes	Description
102	—	61	Change keys
	—	65	Activate Session
	—	66	Deactivate Session
	—	70	SAF (Store and Forward) request
45	Parameter 1: Job ID (see the list of available values in <i>FIMI/GetUnitState/Rp/SubUnits</i> )	85	Execute the Scheduler job
		86	Interrupt the Scheduler job

### 3.19 Transaction Conditions

Value	Name	Description
0	Normal	Normal transaction
1	Customer not present	Transaction is conducted without the customer presence
2	Unattended customer operated terminal (CAT)	Transaction is conducted at the self-service terminal (e.g.: at the automatic petrol station without the cashier)
3	Merchant suspicious	Merchant has defined the customer as suspicious
5	Customer present, card not present	Transaction is conducted in the customer presence, but without a card
8	Mail/phone order	Mail or phone order
10	Customer identify verified	Customer presents the identification document
11	Normal – with Bonuses	Normal transaction with bonuses
51	Verification only	Card/account verification only (to verify the customer paying capability)
52	Recurring payment	Recurring payment performed under the agreement with the merchant
53	Installment payment	Deferred payment purchase, recurring payment to repay debt
59	VSEC request (public network, unknown encryption)	Internet transaction passed through the public networks with unknown encryption
60	Recurring payment - Secure Attempt	Recurring payment via internet with the payment system check
61	Recurring payment - Secure VSEC	Internet recurring payment via the encrypted channels
62	Recurring payment - Non-secure VSEC	Internet recurring payment via the public networks without the encryption
63	Recurring payment - EMV	Internet recurring payment executed by a chip EMV card
64	Installment payment - Secure Attempt	Purchase by installment conducted with the payment system check
65	Installment payment - Secure VSEC	Purchase by installment. Operation is conducted via the encrypted channels
66	Installment payment - Non-secure VSEC	Deferred payment purchase. Operation is conducted via the public networks
67	Installment payment - EMV chip transaction	Installment payment. EMV chip transaction
71	Referral	Transaction is conducted via the payment system Referral service (pre-addressing the issuer)
72	Hardware cryptographic customer authentication	Telebank-transaction (Internet/phone/mobile-banking) with the cryptographic pre-check at the cryptohardware
73	Software cryptographic customer authentication	Telebank-transaction (Internet/phone/mobile-banking) with the software cryptographic pre-check
78	Merchant Risk Based Decisioning	Internet purchase transaction executed via MasterPass provided that the merchant does not support 3D-Secure.

Value	Name	Description
79	Authentication by Issuer Risk Based Decisioning	Internet purchase transaction executed via MasterPass with authentication in the issuer risk system.
81	3D-Secure supported only by acquirer	Internet purchase transaction on condition that 3D-Secure is supported by the acquirer (the issuer doesn't support the 3D-Secure) only
82	3D-Secure Authentication	Internet purchase transaction with the prior authentication in the 3D-Secure
83	Nonauthenticated SET without cardholder certificate, chip cryptogram used	SET transaction conditions – defined for the transactions incoming from Visa and Mastercard IPSs for the Internet purchase modes
84	Secure SET with cardholder certificate, chip cryptogram used	
85	Secure SET with cardholder certificate	
86	Nonauthenticated SET without cardholder certificate	
87	Channel encrypted VSEC	VSEC transaction conditions - defined for the transactions incoming from Visa and Mastercard IPSs for the Internet purchase modes
88	Non secure VSEC	
89	Channel encrypted VSEC, chip cryptogram used	
91	EMV chip transaction	EMV chip transaction
92	EMV chip transaction - with Bonuses	EMV chip transaction with bonuses

### 3.20 Card Entry Modes

Card entry mode is an integer value with the length up to 3 characters. If the length is less than 3 characters, the value of POSEntryMode should be zero-filled to the left so that the length of the obtained value was equal to three.

The first two characters of the obtained value represent the card entry mode, whereas the third character represents PIN entry mode in a transaction.

The list of available values for the POSEntryMode (PAN entry mode) 1-st and 2-nd characters:

- '00' – unknown or no terminal (unknown or the terminal is absent);
- '01' – manual key entry (card number manual entry);
- '02' – magnetic stripe read, CVV not reliable (magnetic stripe has been read, CVV is not available);
- '03' – consumer-presented QR (EMV transaction, EMV data is read from the QR code generated on the mobile device of a cardholder);
- '04' – optical character reader (optical reader has been used);
- '05' – ICC, CVV reliable (chip EMV transaction, CVV is available);
- '07' – contactless EMV (EMV card contactless reading);
- '09' – PAN entry via electronic commerce, including remote chip;
- '90' – mag stripe read, CVV reliable (magnetic stripe has been read, CVV is available);
- '91' – contactless magnetic stripe data (contactless reading of the EMV card magnetic stripe data);
- '95' – ICC, CVV unreliable (chip EMV transaction, CVV is not available).

The list of available values for the POSEntryMode (PIN entry mode) 3-rd character:

- '0' – unknown (unknown if PIN has been entered);
- '1' – PIN accepted (PIN has been entered);
- '2' – cannot accept PIN (PIN has not been entered).

### 3.21 Crypto Functions

Function Name	Function Profile
CalcCVKCheck	RInt CalcCVKCheck(RStr CVK) Calculates the check value for CVK.
CalcPVKCheck	RInt CalcPVKCheck(RStr PVK) Calculates the check value for PVK.
CalcZonePinKeyCheck	RInt CalcZonePinKeyCheck(RStr ZPK) Calculates the check value for ZPK.
CalcZoneMasterKeyCheck	RInt CalcZoneMasterKeyCheck(RStr ZMK) Calculates the check value for ZMK.
ExportZonePinBlock	RStr ExportZonePinBlock(RStr ZPK, RStr PINBlock, RStr PAN, RInt PINFormat) Obtains the PIN block under ZPK.
ImportClearPinBlock	RStr ImportClearPinBlock(RStr Pin, RStr PAN) Obtains the clear PIN block of the card.
ImportZonePinBlock	RStr ImportZonePinBlock(RStr ZPK, RStr PINBlock, RStr PAN, RInt PINFormat) Obtains the clear PIN block from the PIN block under ZPK
LoadFormattingData	void LoadFormattingData(RStr format) Loads the print template to the cryptoserver used for the keys printing.
CalculateCVV	RInt CalculateCVV(RStr CVK1, RStr CVK2, RStr PAN, RStr expiration, RStr service) Calculates PVV by card data.
CalculatePINOffset	RInt CalculatePINOffset(RStr PVK, RStr decimalizer, RStr PAN, RInt ValDateFormat, RStr PinBlock) Calculates PINOffset by card data.
CalculatePVV	RInt CalculatePVV(RStr PVK1, RStr PVK2, RInt PVK1, RStr PAN, RStr PinBlock) Calculates PVV by card data.
GeneratePIN	RStr GeneratePIN(RStr PAN, RInt Len) Generates PIN
ExportKey	void ExportKey(RStr ZMK, RInt KeyType, LStr KeyZMK, RStr KeyLocal, LInt KeyCheck, RInt AtallaVariant = NULL, RInt KeyCheckLen = 2) Exports key with the specified type. Available values: KeyType = 1 (ZPK), 2 (ZAK)

Function Name	Function Profile
ImportKey	<pre>void ImportKey(RStr ZMK, RInt KeyType, RStr KeyZMK, LStr KeyLocal, LInt KeyCheck, RInt AtallaVariant = NULL, RInt KeyCheckLen = 2)</pre> <p>Imports key with the specified type. Available values: KeyType = 1 (ZPK), 2 (ZAK)</p>
CalculateMAC	<pre>RStr CalculateMAC(RStr KeyLocal, RInt KeyType, RStr Data)</pre> <p>Calculates MAC Available values: KeyType = 2 (ZAK)</p>
CheckMAC	<pre>RInt CheckMAC(RStr KeyLocal, RInt KeyType, RStr Data, RStr MAC)</pre> <p>Checks MAC Available values: KeyType = 2 (ZAK)</p>
CalculateHMAC	<pre>RStr CalculateHMAC(RStr KeyLocal, RStr Data, RInt HashAlgorithm)</pre> <p>Calculates HMAC.</p>
CheckHMAC	<pre>RInt CheckHMAC(RStr KeyLocal, RStr Data, RStr MAC, RInt HashAlgorithm)</pre> <p>Checks HMAC.</p>
CalculateSignature	<pre>RStr CalculateSignature(RStr RSASK, RInt HashAlgorithm, RStr Data)</pre> <p>Calculates signature by data</p>
GenerateKey	<pre>void GenerateKey(RStr ZMK, RInt KeyType, RInt Len, LStr KeyZMK, LStr KeyLocal, LInt KeyCheck, RInt KeyCheckLen = 2)</pre> <p>Generates the key of the specified type with the specified length. Available values of KeyType = 1 (ZPK), 2 (ZAK). Available values of Len = 1, 2</p>
CheckARQC CheckCAPToken	<p>RInt CheckARQC(RStr PAN, RInt MBR, RStr CAPToken, RStr Challege) – the function is outdated; it is recommended to use the <i>CheckCAPToken</i> function.</p> <p>RInt CheckCAPToken (RStr PAN, RInt MBR, RStr CAPToken, RStr Challege)</p> <p>Check CAP token of the card.</p> <p>It is possible to check only CAPToken that is created using the VISA algorithm calculated by the default cryptogram values.</p>

Function Name	Function Profile
CalculateARPC	<p>RStr CalculateARPC (RStr PAN, RInt MBR, RStr ARQC, RStr RespCode, RStr CSU, RStr ICC_Amount, RStr ICC_CBAmount, RInt ICC_TermCountry, RStr ICC_TVR, RInt ICC_Currency, RStr ICC_TranDate, RInt ICC_TranType, RStr ICC_Random, RInt ICC_AppProfile, RStr ICC_AppTranCount, RInt ICC_CardMember, RStr ICC_IssuerData)</p> <p>Calculates ARPC by card data and ICC fields of the transaction.</p> <p>The function returns ARPC in the hexadecimal format.</p>
EncryptZoneData	<p>RStr EncryptZoneData (RStr Data, RStr ZEK, RInt UseCBC=1)</p> <p>Encrypts data by ZEK.</p> <p>It returns data in the Base64 format.</p>
DecryptZoneData	<p>RStr DecryptZoneData (RStr Data, RStr ZEK, RInt UseCBC=1)</p> <p>Decrypts data by ZEK.</p> <p>It returns data in the Base64 format.</p>

### 3.22 Transaction Reason Codes

Code	Description
<b>Telebank Card Blocking</b> (transaction code 332)	
0	Not active
2	Lost
3	Stolen
<b>Online Reversal</b> (if Type=420 and OrigId!=NULL)	
1	Timeout
2	Invalid response
3	Destination not available
8	Cancel
10	Hardware error
11	Completed partially
20	Suspect
21	MAC failure
22	MAC key error
23	Reply error
24	Invalid MAC
30	Visa BASE-I advice
40	Corresponding account exhausted
41	Cumulative acquiring limit exceeded
50	Related transaction declined
51	Confirmation transaction declined
<b>Online Representation, Online Fee collection</b>	
The values from the dictionary InitSession/VISATranReasonList are used.	
<b>Visa Stoplist Processing via FIMI</b> (transaction 586)	
The card status in the VISA Stoplist. The field is defined, if Message=220 or 222.	
Available values:	
Lost=3000, Stolen=3001, VIP=11, Deny=5, Referral=1, Pickup=4, Pick up Card, Special attention=7	
VIP High value activity level 1 = 12, VIP High value activity level 2 = 13, VIP High value activity level 3 = 14,	
VIP High value activity level 4 = 15, VIP High value activity level 5 = 16, VIP High value activity level 6 = 17,	
VIP High value activity level 7 = 18, VIP High value activity level 8 = 19, VIP High value activity level 9 = 20	

Code	Description
<b>Mastercard Stoplist Processing via FIMI</b> (transaction code 587)	
The card status in the MasterCard-Europe Stoplist. The field is defined, if Message=220,224,222,225.	
Available values:	
Lost=3000, Stolen=3001, Undeliver=3002, Fraudulent=3701, Counterfeit=3003, Pickup=4	
<b>Local Stoplist Processing via FIMI</b> (transaction code 588)	
Card status value 3.2	

### 3.23 Networks

Value	Name
1	TranzWare
11	Visa
21	Mastercard Europe
22	Mastercard Global
31	DinersClub
32	Amex
33	JCB
34	Discover
41	BanknetVN
42	UnionPay
43	SPAN
44	NSPK
45	EBC_NPS
46	Bangla
51	Sbercart
81	SmartVista

## 4 Password and OTP Encryption Algorithm

### 4.1 Transfer of Password in Clear or as Hash

In the protocol version 1.0 and to version 3.5, the operator password is transferred as a plain text. If *FIMI/Ver<=3.5* or *NULL*, the *FIMI/Password* field transfers the clear operator password or password hash, the behavior depends on the settings at the terminal level. If the password hash is transferred (for the transfer, the password hash must be converted to ASCII HEX), it is allowed to use two hashing algorithms.

#### ***3DES ESB, without padding***

Password hash with the length of 8 bytes that is calculated as the 3DES cryptogram of first 8 characters (bytes) of the password (it is truncated to 8 characters or right-padded with spaces) encrypted with the password that is truncated to or right-padded with spaces up to 16 characters (bytes). The 3DES algorithm with the key of the double length is used (ANSI X9.52 keying option 2). Prior to the encryption, the password is converted to the upper case.

The password encryption algorithm can be presented with the following pseudo code:

```
HashPassword = Encrypt(Algorithm=>"3des2Key-ecb-nopad",
                      Data=>PadTrail( SubStr(UpperCase(ClearPassword),1,8),8) ,
                      Key=>PadTrail(SubStr(UpperCase(ClearPassword),1,16),16)),
```

where:

HashPassword – operator password hash.

ClearPassword – clear password.

Example:

```
Operator password 'FIMIProd'
HashPassword = 3DES2KEY('FIMIPROD','FIMIPROD_____') = 0xA216B5BEF1577D1A
```

#### ***SHA3-256***

Password hash with the length of 32 bytes that is calculated as a hash function result by the SHA3-256 algorithm of the concatenated string with the operator name converted to the upper case and with the password.

The password encryption algorithm can be presented with the following pseudo code:

```
HashPassword = Hash( Algorithm=>"sha3-256",
                      UpperCase(Clerk) + ClearPassword)
where:
HashPassword – operator password hash.
ClearPassword – clear password.
Clerk – operator name.
```

Example:

```
Operator password 'FIMIProd', operator name 'User'
HashPassword = SHA3-256('USER' + 'FIMIProd') =
0xE4ADD5A841C7214FF0C736EC8344D62839FE3BB18E526717EB8B2D7D287A14C1
```

The algorithm is determined depending on the operator and terminal settings in TWO, the password hashing algorithm version used by the operator is returned in the *PasswordVersion* response field

of the *InitSession* operation. The terminal should transfer the hash according to the specified version in the PasswordVersion response field of the *InitSession* operation.

## 4.2 Transfer of Password Using Encryption

Starting from verison 3.6, the operator password is transferred encrypted between the terminal and driver. The driver considers password as encrypted if the *FIMI/Ver* field is not NULL and *FIMI/Ver>=3.6*. The allowed password encryption is determined on the basis of the current operator password hashing algorithm, the version of the password hashing algorithm currently used by a certain operator is returned in the PasswordVersion response field of the *InitSession* operation or NewPasswordVersion response field of the *ChangePassword* operation.

If the encrypted password is transferred (for the transfer, the encrypted password must be converted to ASCII HEX), it is allowed to use two encryption algorithms depending on the current password hashing algorithm defined for the operator.

### **3DES ESB, without padding**

1. The terminal creates the password hash with the length of 8 bytes that is calculated as the 3DES cryptogram of first 8 characters (bytes) of the password (truncated to 8 characters or right-padded up with spaces) encrypted with the password that is truncated to or right-padded with spaces up to 16 characters (bytes). The 3DES algorithm with the key of the double length is used (ANSI X9.52 keying option 2). Prior to the encryption, the password is converted to the upper case.
2. The terminal receives the initial Challenge (encryption component with the length of 8 bytes) and session ID when initializing the FIMI session by the *InitSession* operation in the *FIMI/NextChallenge* and *FIMI/InitSession/Rp/Id* fields, respectively.
3. The terminal sends the session ID in the *FIMI/SessionId* header field in each next operation request. The *FIMI/Password* header field transfers the DES-encrypted Challenge value (received in the response to the previous operation within the session in the *FIMI/NextChallenge* response field), the encryption key is the hash. The encrypted Challenge is transferred in all requests to the FIMI driver, except for *InitSession*. When the new value is received in the *FIMI/NextChallenge* field of the FIMI driver response, the password encryption component should be changed starting from the next request. If the *FIMI/NextChallenge* field is absent in the response, the password encryption component has not been changed and it is required to use the current Challenge, it should not be updated for the next request.
4. When the password is changed, the *NewPassword* request field of the *ChangePassword* operation transfers the new password encrypted with the hash of the old password according to the des-ecb-nopad algorithm. If the password length is not multiple of 8, the password must be padded with spaces up to 8, 16 characters. The encrypted password must be converted to ASCII HEX.

The password encryption algorithm can be presented with the following pseudo code:

```
HashPassword = Encrypt(Algorithm=>"3des2Key-ecb-nopad",
                      Data=>PadTrail( SubStr(UpperCase(ClearPassword),1,8),8),
                      Key=>PadTrail(SubStr(UpperCase(ClearPassword),1,16),16))
```

```
ChallengeByHashPassword = Encrypt(Algorithm=>"des-ecb-nopad",
                                   Data=>FIMI/NextChallenge,
                                   Key=>HashPassword)
```

where:

HashPassword – operator password hash.

ClearPassword – clear password.

FIMI/NextChallenge – Challenge value received in the response to the previous operation within the session.

ChallengeByHashPassword – Challenge encrypted with the password hash, it is transferred in the request.

Example:

Operator password 'FIMIProd', Challenge '06253765'.

```
HashPassword = 3DES2KEY('FIMIPROD','FIMIPROD_____') = 0xA216B5BEF1577D1A
ChallengeByHashPassword = DES-ECB('06253765', 0xA216B5BEF1577D1A) =
0x9CC5530A38A7B667
```

Calculating encrypted password in Oracle 12g DBMS for this example:

```
declare
    passwordStr varchar2(16);
    passwordKeyStr varchar2(16);
    passwordRaw raw(128);
    passwordKeyRaw raw(128);
    TDES_ECB_NONE constant pls_integer := dbms_crypto.ENCRYPT_3DES_2KEY +
dbms_crypto.CHAIN_ECB + dbms_crypto.PAD_NONE;
    DES_ECB_NONE constant pls_integer := dbms_crypto.ENCRYPT_DES +
dbms_crypto.CHAIN_ECB + dbms_crypto.PAD_NONE;
    passwordHash raw(1024);
    nextChallengeStr varchar2(8);
    nextChallengeRaw raw(128);
    cryptPassword raw(1024);
    password varchar2(50);
begin
    password := 'FIMIPROD';
    nextChallengeStr := '06253765';
    dbms_output.put_line('> Password clear: ' || password||'
challenge:'||nextChallengeStr);
    passwordStr := rpad(substr(upper(password), 1,8), 8, chr(32));
    passwordKeyStr := rpad(substr(upper(password), 1,16), 16, chr(32));
    passwordRaw := utl_raw.cast_to_raw(passwordStr);
    passwordKeyRaw := utl_raw.cast_to_raw(passwordKeyStr);
    passwordHash := dbms_crypto.encrypt(src => passwordRaw, typ =>
TDES_ECB_NONE, key => passwordKeyRaw);
    dbms_output.put_line('> Password hash (encrypted DES3 hex value): ' ||
rawtohex(passwordHash));
    nextChallengeRaw := utl_raw.cast_to_raw(nextChallengeStr);
    cryptPassword:= dbms_crypto.encrypt(src => nextChallengeRaw, typ =>
DES_ECB_NONE, key => passwordHash);
    dbms_output.put_line('> Crypt password (encrypted DES hex value): ' ||
rawtohex(cryptPassword));
end;
/
```

Execution result:

```
> Password hash (encrypted DES3 hex value): A216B5BEF1577D1A
> Crypt password (encrypted DES hex value): 9CC5530A38A7B667
```

### **SHA3-256**

#### **Allowed starting from protocol version 16.13 (FIMI/Ver>=16.13)**

1. The terminal generates the password hash with the length of 32 bytes that is calculated as a hash function result by the SHA3-256 algorithm of the concatenated string with the operator name converted to the upper case and with the password.
2. The terminal receives the initial Challenge (encryption component with the length of 32 bytes) and session ID when initializing the FIMI session by the *InitSession* operation in the FIMI/NextChallenge and FIMI/InitSession/Rp/Id fields, respectively.

3. The terminal sends the session ID in the FIMI/SessionId header field in each next operation request. The FIMI/Password header field transfers the AES-encrypted Challenge value (received in the response to the previous operation within the session in the FIMI/NextChallenge response field), the encryption key is the hash. The encrypted Challenge is transferred in all requests to the FIMI driver, except for *InitSession*. When the new value is received in the FIMI/NextChallenge field of the FIMI driver response, the password encryption component should be changed starting from the next request. If the FIMI/NextChallenge field is absent in the response, the password encryption component has not been changed and it is required to use the current Challenge, it should not be updated for the next request.
4. When the password is changed, the NewPassword request field of the *ChangePassword* operation transfers the new password encrypted with the hash of the old password according to the aes-256-ecb-nopad algorithm. If the password length is not multiple of 16, the password must be padded with spaces up to 16, 32, 48 characters. The encrypted password must be converted to ASCII HEX.

The password encryption algorithm can be presented with the following pseudo code:

```
HashPassword = Hash(Algorithm=>"sha3-256",
                    Uppercase(Clerk) + ClearPassword)
ChallengeByHashPassword = Encrypt(Algorithm=>"aes-256-ecb-nopad",
                                   Data=>FIMI/NextChallenge,
                                   Key=>HashPassword)
```

where:

HashPassword – operator password hash.

ClearPassword – clear password.

Clerk – operator name.

FIMI/NextChallenge – Challenge value received in the response to the previous operation within the session.

ChallengeByHashPassword – Challenge encrypted with the password hash, it is transferred in the request.

Example:

```
Operator password 'FIMIProd', operator name 'User',
Challenge '64C89F99C6A988403D57004A71D947CA'
```

```
HashPassword = SHA3-256('USER' + 'FIMIProd') =
               0xE4ADD5A841C7214FF0C736EC8344D62839FE3BB18E526717EB8B2D7D287A14C1
ChallengeByHashPassword = AES-256-ECB('64C89F99C6A988403D57004A71D947CA',
                                         0xE4ADD5A841C7214FF0C736EC8344D62839FE3BB18E526717EB8B2D7D287A14C1) =
               0x210589AC6D0C671DEA056A4B35F54F0165677F7EE84EBF59BAC5618711190B01
```

## 4.3 OTP Hash Calculation Algorithm

The hash of OTP (dynamic password) has the length of 8 bytes. It is calculated as the 3DES cryptogram of first 8 characters (bytes) of OTP (it is truncated to 8 characters or right-padded with spaces) encrypted with OTP that is truncated or right-padded with spaces up to the length of 16 characters (bytes). The 3DES algorithm with the double-length key is used (ANSI X9.52 keying option 2). The OTP hash is transferred in the *Logon*, *ResetPassword* requests.

The OTP hash calculation algorithm can be presented with the following pseudo code:

```
HashOTP = Encrypt(Algorithm=>"3des2Key-ecb-nopad",
                   Data=>PadTrail( SubStr(ClearOTP,1,8),8),
                   Key=>PadTrail(SubStr(ClearOTP,1,16),16)),
```

where:

HashOTP – OTP hash.

ClearOTP– clear OTP.

Example:

```
Dynamic password(OTP) '3456'
HashOTP = 3DES2KEY('3456_____','3456_____') = 0x602EEFC5F4FF7BC5
```

## 5 DS Algorithm for SOAP Messages

For the SOAP XML protocol, the digital signature (hereinafter - DS) can be created for the SOAP message body within the request and response. The signature is created according to the XML Digital Signature standard (<https://www.w3.org/TR/xmlsig-core1/>), the Web Services Security SOAP Message Security (WS-Security) extension of the SOAP protocol is used for the DS transfer. The signature in the message can be applied to one XML element only.

### 5.1 DS Generation

After the request is generated within the FIMI operation, the terminal can sign the SOAP body of the message. The signature is created according to the following algorithm:

1. SOAP Body is generated in compliance with the operation description. After that, the wsu:Id attribute (`xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"`) is added to the SOAP Body element. It is the identifier of the data to be signed.
2. The SOAP Body node is converted to the canonical form. To convert the SOAP Body to the canonical form, one of the following algorithms can be used (Transform):
  - *Canonical XML Version 1.0* (<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>)
  - *Exclusive XML Canonicalization XML Version 1.0\** (<http://www.w3.org/2001/10/xml-exc-c14n#>)If the fields with the StrUAMP type (for the *GetPrefixInfo*, *ReverseTransaction*, *StandAloneReversal* operations) and XML version 1.1 are transferred, replace with spaces all ASCII characters with code from 01 to 31 (0x01 to 0x1F including) transferred as the escape sequence of the "`&#x<Hex code>`," or "`&#<decimal code>`;" format and, thereby, bring XML to version 1.0 prior to converting the SOAP Body node to the canonical form, and perform the actions for canonicalization according to the selected algorithm.
3. For the sequence of bytes obtained at step 2, the hash (DigestValue) is calculated using one of the following available hash functions RFC3874 (DigestMethod):
  - *SHA-224* (<http://www.w3.org/2001/04/xmldsig-more#sha224>)
  - *SHA-256\** (<http://www.w3.org/2001/04/xmlenc#sha256>)
  - *SHA-384* (<http://www.w3.org/2001/04/xmldsig-more#sha384>)
  - *SHA-512* (<http://www.w3.org/2001/04/xmlenc#sha512>)
4. One of the canonicalization algorithms for the element with data for signing (CanonicalizationMethod) `ds:SignedInfo(xmlns:ds="http://www.w3.org/2000/09/xmldsig#")` is defined:
  - *Canonical XML Version 1.0* (<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>)
  - *Exclusive XML Canonicalization XML Version 1.0* (<http://www.w3.org/2001/10/xml-exc-c14n#>) – recommended algorithm.
5. One of the available RFC 4051 signing algorithms (SignatureMethod) is defined:
  - *RSA SHA-224* (<http://www.w3.org/2001/04/xmldsig-more#rsa-sha224>)
  - *RSA SHA-256\** (<http://www.w3.org/2001/04/xmldsig-more#rsa-sha256>)
  - *RSA SHA-384* (<http://www.w3.org/2001/04/xmldsig-more#rsa-sha384>)
  - *RSA SHA-512* (<http://www.w3.org/2001/04/xmldsig-more#rsa-sha512>)

6. In compliance with the XML Digital Signature standard, the ds:Signature element with the sub-element ds:SignedInfo(xmlns:ds="http://www.w3.org/2000/09/xmldsig#") is generated. It contains the following:
  - CanonicalizationMethod@Algorithm (defined at step 4)
  - SignatureMethod@Algorithm (defined at step 5)
  - Transform@Algorithm (defined at step 2)
  - DigestMethod@Algorithm (defined at step 3)
  - DigestValue (calculated at step 3 and encoded in Base64)
  - Reference@URI (defined at step 1)
7. For the obtained ds:SignedInfo element, the canonicalization algorithm defined at step 4 is applied. For the obtained sequence after canonicalization, the signature is calculated using the algorithm defined at step 5. The result is encoded to Base64 and placed to the ds:SignatureValue sub-element of the ds:Signature element (xmlns:ds="http://www.w3.org/2000/09/xmldsig#").
8. The ds:KeyInfo sub-element is generated additionally and added to the ds:Signature element (xmlns:ds="http://www.w3.org/2000/09/xmldsig#") with data on the public key or certificate.
9. The obtained ds:Signature element is placed to SOAP Header of the request within the element wsse:Security(xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd").

\* - recommended algorithm.

After the response to the terminal request is generated, the driver can sign the SOAP body of the message if the signing of messages for the operation is activated at the FIMI terminal of **TWO**. The driver always applies the *Exclusive XML Canonicalization XML Version 1.0* algorithm for the canonicalization of ds:SignInfo and SOAPBody, the SHA-256 algorithm for the DigestValue calculation, and the RSA SHA-256 algorithm for the SignatureValue calculation.

## 5.2 Example of DS Creation

For the *AcctCredit* operation request. The following algorithms were selected: *Exclusive XML Canonicalization XML Version 1.0* for the canonicalization of ds:SignInfo and SOAPBody, the SHA-256 algorithm for the DigestValue calculation, the RSA SHA-256 algorithm for the SignatureValue calculation.

SOAP Body is created according to the operation description and the identifier of data to be signed is added:

```
<soap:Envelope xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
  xmlns:fimi1="http://schemas.compassplus.com/two/1.0/fimi_types.xsd" xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body wsu:Id="body" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <fimi:AcctCreditRq>
      <fimi:Request Clerk="USER" Password="USER" Product="FIMI" Ver="3.5">
        <fimi1:InstName>DEMO</fimi1:InstName>
        <fimi1:Account>888</fimi1:Account>
        <fimi1:Amount>100</fimi1:Amount>
        <fimi1:IgnoreImpact>1</fimi1:IgnoreImpact>
        <fimi1:NeedNotify>1</fimi1:NeedNotify>
      </fimi:Request>
    </fimi:AcctCreditRq>
  </soap:Body>
</soap:Envelope>
```

The SOAP Body node is converted to the canonical form, the conversion result is as follows:

```
<soap:Body xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wsseecurity-utilty-1.0.xsd" wsu:id="body">
<fimi:AcctCreditRq xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd">
<fimi:Request Clerk="USER" Password="USER" Product="FIMI" Ver="3.5">
<fimi1:InstName xmlns:fimi1="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">DEMO</fimi1:InstName>
<fimi1:Account xmlns:fimi1="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">888</fimi1:Account>
<fimi1:Amount xmlns:fimi1="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">100</fimi1:Amount>
<fimi1:IgnoreImpact xmlns:fimi1="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">1</fimi1:IgnoreImpact>
<fimi1:NeedNotify xmlns:fimi1="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">1</fimi1:NeedNotify>
</fimi:Request>
</fimi:AcctCreditRq>
</soap:Body>
```

DigestValue is calculated using the SHA-256 algorithm, the result is converted to Base64:

```
5u11a/GBLBypSQR4abNV+SqA+hKadDmKCh6Z3+b3+g=
```

ds:Signature with the ds:SignedInfo sub-element is generated and the obtained DigestValue is added to it:

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
<ds:Reference URI="#body">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
<ds:DigestValue>5u11a/GBLBypSQR4abNV+SqA+hKadDmKCh6Z3+b3+g=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue/>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509IssuerSerial>
<ds:X509IssuerName>C=RU, ST=Some-State, L=Mgn, O=C+, OU=FIMI, CN=FIMIRP</ds:X509IssuerName>
<ds:X509SerialNumber>11000510014721093786</ds:X509SerialNumber>
</ds:X509IssuerSerial>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
```

The canonicalization is performed for the obtained ds:SignedInfo element:

```
<ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /></ds:CanonicalizationMethod>
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" /></ds:SignatureMethod>
<ds:Reference URI="#body">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /></ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" /></ds:DigestMethod>
<ds:DigestValue>5u11a/GBLBypSQR4abNV+SqA+hKadDmKCh6Z3+b3+g=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
```

For the canonicalized ds:SignedInfo, SignatureValue is calculated and the obtained value is encoded to Base64:

```
qzxjbVEHmxW79q2woPW/ildAOBINzNW6RrfDzPxfFCyp8MRkWCyclTpRUjYqrqL9KZNoDQdavhMiVYiWnm6s6rlV6uGD
xizFvm3QvEFDCAski1fKGYc/k0EMQN9ylHljfQ8ZeTbzb2yxcq+F5oOJzIYDj9CTMCK+qtc2J4DxKlnwP/YVliQPFIIMnuhxP
Ts/6F3/eHJXqrUG2ZNmMP9DZ/WK+/e16NmBWARk2GIIoZy/jGIIHlleAoPX8+dIFfwzrdSvxNDMGYJX8UJF2Pkwa8CASM
Q+7ly5aFjAJLwHazFWjvKezSXJ+QKo4qr2pnu0EWNi22cf9lpOSTuhwR8+A==
```

The ds:SignatureValue sub-element is added to the ds:Signature element, the obtained ds:Signature element is placed to SOAP Header:

```
<soap:Envelope xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
    xmlns:fimi1="http://schemas.compassplus.com/two/1.0/fimi_types.xsd" xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
    <soap:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
            xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
                    <ds:Reference URI="#body">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                        <ds:DigestValue>5u11a/GLBLBFypSQR4abNV+SqA+hKadDmKCh6Z3+b3+g=</ds:DigestValue>
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue>qzxjbVEHmxW79q2woPW/ildAOBINzNW6RrfDzPxfFCyp8MRkWCyc
ITpRUjYqrqL9KZNoDQdavhMiVYiWnm6s6rlV6uGDxizFvm3QvEFDCAski1fKGYc/k0EMQN9ylHlj
JfQ8ZeTbzb2yxcq+F5oOJzIYDj9CTMCK+qtc2J4DxKlnwP/YVliQPFIIMnuhxPTs/6F3/eHJXqrUG2
ZNmMP9DZ/WK+/e16NmBWARk2GIIoZy/jGIIHlleAoPX8+dIFfwzrdSvxNDMGYJX8UJF2Pkwa8
CASMQ+7ly5aFjAJLwHazFWjvKezSXJ+QKo4qr2pnu0EWNi22cf9lpOSTuhwR8+A==</ds:SignatureValue>
                <ds:KeyInfo>
                    <ds:X509Data>
                        <ds:X509IssuerSerial>
                            <ds:X509IssuerName>C=RU, ST=Some-State, L=Mgn, O=C+, OU=FIMI, CN=FIMIRq</ds:X509IssuerName>
                            <ds:X509SerialNumber>11000510014721093786</ds:X509SerialNumber>
                        </ds:X509IssuerSerial>
                    </ds:X509Data>
                </ds:KeyInfo>
            </ds:Signature>
        </wsse:Security>
    </soap:Header>
    <soap:Body wsu:Id="body" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <fimi:AcctCreditRq>
            <fimi:Request Clerk="USER" Password="USER" Product="FIMI" Ver="3.5">
                <fimi1:InstName>DEMO</fimi1:InstName>
                <fimi1:Account>888</fimi1:Account>
                <fimi1:Amount>100</fimi1:Amount>
                <fimi1:IgnoreImpact>1</fimi1:IgnoreImpact>
                <fimi1:NeedNotify>1</fimi1:NeedNotify>
            </fimi:Request>
        </fimi:AcctCreditRq>
    </soap:Body>
</soap:Envelope>
```

## 5.3 DS Verification

If the signature verification for the operation requests is enabled in the **TWO** terminal settings, the FIMI driver verifies the signature. If the verification is successful, the driver continues the operation on the basis of the request. If the verification fails, the driver declines the operation with the special response code (*FIMI/Response*) InvalidSignature(68). The terminal can verify the signature in responses on the operation prior to the response analysis.

The signature is verified according to the generation algorithm in the reverse order:

1. The correctness of the signature structure and ds:SignedInfo attributes (`xmlns:ds="http://www.w3.org/2000/09/xmldsig#"`) are analyzed in the message:

- CanonicalizationMethod@Algorithm
- SignatureMethod@Algorithm
- Transform@Algorithm
- DigestMethod@Algorithm

The algorithms for the canonicalization of ds:SignInfo and SOAP Body, calculation of the DigestValue hash, verification of the SignatureValue signature are defined.

2. The ds:SignedInfo canonicalization according to the algorithm specified in ds:CanonicalizationMethod and signature verification according to the algorithm specified in ds:SignatureMethod using the public key and signature data from the ds:SignatureValue element are performed.
3. The canonicalization of the SOAP Body element is performed according to the algorithm specified in ds:Transform. If the fields with the StrUAMP type (for the *GetPrefixInfo*, *ReverseTransaction*, *StandAloneReversal* operations) and XML version 1.1 are transferred, replace with spaces all ASCII characters with code from 01 to 31 (0x01 to 0x1F including) transferred as the escape sequence of the "&#x<Hex code>;" or "&#<decimal code>;" format and, thereby, bring XML to version 1.0 prior to converting the SOAP Body node to the canonical form, and perform the actions for canonicalization according to the algorithm. For the obtained sequence, the hash is calculated according to the algorithm specified in ds:DigestMethod and compared with data in the ds:DigestValue element. If they differ, the signature is considered invalid.
4. The attributes wsu:Id (`xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"`) from SOAP Body and URI from the ds:Reference element are compared.

## 6 Card PIN Setting Technology

### 6.1 Working Key is Used

#### 6.1.1 Working Key TPK

##### Example 1

(with the use of *GetPVV\_PINOffset*, *SetCardPVV* and *AddEMVScript* operations)

To set up or change the card PIN, perform the following actions:

1. Initialize the session – InitSession FIMI-request. The obtained session ID is transmitted in all the subsequent requests.
2. Authenticate the operator (optionally) - Logon FIMI-request.
3. Search for the card - GetCardInfo FIMI-request. For that purpose, specify the card PAN (in part or in full). The response to the request will contain CardUID that can be sent in all requests instead of PAN. If the “Use Additional Card ID” flag is enabled in the FIMI terminal settings and CardUID can be calculated by the algorithm, the calculated CardUID can be sent in the request. The response to the request will contain the *IssueTechnology* parameter that must be considered when setting Offline PIN for the chip card.
4. Generate a working session key TPK - GetWorkingKey FIMI request (KeyType=1). The response will contain a working key - TPK (TPK under TMK) and its identifier. In this case, the TPK key is a working key for encrypting the PIN block, it is generated by the cryptoserver and linked to the session. The TPK key defined in **TWO** (FIMI terminal settings) is designed for other purposes.
5. Get TPK value. For that purpose, the party responsible for sending the request must store TMK (e.g.: in HSM Eracom).
6. Get PIN. PIN can be entered by the user or generated as a random number (e.g.: in HSM Eracom).
7. PIN block is created in the ISO-0 or ISO-1 format. The clear PIN block is encrypted under the key that was obtained in p. 5.
8. Calculate PVV/PIN offset - GetPVV\_PINOffset FIMI-request. The request contains the PIN block encrypted under TPK with a new PIN and identifier of key obtained at step 4. It is also possible to send the old PIN for verification in the particular request (the old PIN is sent as a PIN block encrypted under TPK and PVV/PIN Offset value for the old PIN is sent). If the “PIN Block Encryption” key is defined for the card prefix, the response will contain a PIN block encrypted under ZPK that can be used to set Offline PIN for the chip card.
9. Set PVV - SetCardPVV FIMI-request.
10. To set Offline PIN for the chip card, go to this paragraph. To define the card issue technology (mag stripe or EMV), use the *IssueTechnology* parameter of the GetCardInfo FIMI-request (see p.3). For the chip card, add EMV script to the card – AddEMVScript FIMI-request and send PIN block encrypted under ZPK that was obtained in p. 7 in the request (the *AddEMVScript* request can also contain the PIN block under the cryptoserver LMK or working key; refer to the request description).
11. End operator session – Logoff FIMI-request.

**Example 2****(with the use of the POSRequest operation)**

To set up or change the card PIN, perform the following actions:

1. Initialize the session – InitSession FIMI-request. The obtained session ID is transmitted in all the subsequent requests.
2. Authenticate the operator (optionally) - Logon FIMI-request.
3. Generate a working session key TPK - GetWorkingKey FIMI request (KeyType=1). The response will contain a working key - TPK (TPK under TMK) and its identifier. In this case, the TPK key is a working key for encrypting the PIN block, it is generated by the cryptoserver and linked to the session.
4. Get TPK clear value. For that purpose, the party responsible for sending the request must store TMK (e.g.: in HSM Eracom).
5. Get PIN. PIN can be entered by the user or generated as a random number (e.g.: in HSM Eracom).
6. PIN block for the new PIN is created. The clear PIN block is encrypted under TPK obtained at step 4.
7. Create PIN block for the old PIN. The clear PIN block is encrypted under TPK obtained at step 4. It is permitted to change PIN without checking the old one if no PIN is set for a card or if it has been reset, at that, the new PIN and its PIN block are used instead of the old PIN. The parameters are set up according to the rules of PIN check at TWO level.
8. Change card PIN using the *POSRequest* operation with the *POS PIN Change (139)* transaction of the *Request (200)* type. The request contains the following mandatory fields: PAN, MBR, Track2, Condition, EntryMode, FromAcctType. It also contains PIN block with the new PIN (encrypted under TPK), PIN block with the old PIN (encrypted under TPK) and identifier of key obtained at step 3.
9. End operator session – Logoff FIMI request.

### 6.1.2 Working key RSA-SK

**Example 1****(with the use of GetPVV\_PINOffset, SetCardPVV, AddEMVScript operations)**

To set up or change the card PIN, perform the following actions:

1. Initialize the session – InitSession FIMI request. The obtained session ID is transmitted in all the subsequent requests.
2. Authenticate the operator (optionally) - Logon FIMI request.
3. Search for the card - GetCardInfo FIMI-request. For that purpose, specify the card PAN (in part or in full). The response to the request will contain CardUID that can be sent in all requests instead of PAN. If the “Use Additional Card ID” flag is enabled in the FIMI terminal settings and CardUID can be calculated by the algorithm, the calculated CardUID can be sent in the request. The response to the request will contain the *IssueTechology* parameter that must be considered when setting Offline PIN for the chip card.
4. Generate a working session key RSA-SK – GetWorkingKey FIMI request (KeyType=2). The response returns the public key (unit and component in the DER ANS.1 format) for the key encryption and ID. The private key for encryption is saved within the FIMI operator session.

5. Generate ZPK.
6. Get PIN. PIN can be entered by the user or generated as a random number (e.g.: in HSM Eram).
7. Create PIN block in the ISO-0 or ISO-1 format. The clear PIN block is encrypted under the ZPK key that was obtained in p. 5.
8. Encrypt ZPK key by the RSA/ECB/PKCS1Padding algorithm using the public key (module and exponent) obtained in p. 4.  
Example of encrypting the ZPK key using the public working key:

```

import javax.crypto.Cipher;
import java.security.*;
import java.security.spec.*;
import org.bouncycastle.asn1.*;
import java.util.Base64;

def publicKey =
"MIJJAoGBAL4/sGWkGO9ipD2OahSelTQ6FE75mEtt2CDT7tLutmVcBZrO7YpvvGLH3xYmXaGo
A3zyeWnu88KMvjqEVs1DwjZmvqzh8g49Cp4438Ej1B15f25Q11BXB8SyzsquZWVIO07vtWmav
/zL1IjhKw6cYc3eoH1k3FT7Z64liHhIo+7ZAgMBAAE="

ASN1Sequence sequence =
ASN1Sequence.getInstance(Base64.getDecoder().decode(publicKey));
ASN1Integer modulus = ASN1Integer.getInstance(sequence.getObjectAt(0));
ASN1Integer exponent = ASN1Integer.getInstance(sequence.getObjectAt(1));

RSAPublicKeySpec keySpec = new
RSAPublicKeySpec(modulus.getPositiveValue(), exponent.getPositiveValue());
KeyFactory factory = KeyFactory.getInstance("RSA");
PublicKey pk = factory.generatePublic(keySpec);

def clearZPK = "0123456789ABCDEF0123456789ABCDEF";
byte[] messageToBytes = clearZPK.decodeHex();
Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
cipher.init(Cipher.ENCRYPT_MODE, pk);
def encryptZPK =
Base64.getEncoder().encodeToString(cipher.doFinal(messageToBytes));
//encryptZPK =
"DSQxadTxQdw1xiCYs1GYq+QWva0tmg51b0Ao9Vo8a1WCm9GST1BTwSVyD054D2uDxDkkm5Up
2L5fkfkuou9jqyOTwkgiKQaW+0pGAdgJ29T0pngoa9IHWOoirKj+jARh2lN1MaCg1fRWRX+a
7PNMz83cwOAJRwELZiMV+NXzEU="

```

9. Calculate PVV/PIN offset - GetPVV\_PINOffset FIMI request. The request contains the PIN block encrypted under ZPK with new PIN (obtained at step 6), ZPK encrypted by the public working key (obtained at step 8) and working key identifier (obtained at step 4). It is also possible to send the old PIN for verification in the particular request (the old PIN is sent as a PIN block encrypted under ZPK and PVV/PIN Offset value for the old PIN is sent). If the "PIN Block Encryption" key is defined for the card prefix, the response will contain the PIN block encrypted under ZPK that can be used to set Offline PIN for the chip card.
10. Set PVV – SetCardPVV FIMI-request.
11. Execute this step if it is required to set Offline PIN for the chip card. Identify the card issue technology (magnetic or EMV) using the IssueTechnology parameter of the GetCardInfo FIMI request (see p.3). If the card is chip, add the EMV script for the card – AddEMVScript FIMI request that should contain the PIN block under ZPK obtained at

step 9. (also the AddEMVScript request can contain the PIN block under the cryptoserver LMK or working key, see the request description).

12. End operator session – Logoff FIMI request.

For details, refer to the requests description.

## 6.2 Working Key is not Used

### Example 1

(With the use of the *GetPVV\_PINOffset*, *SetCardPVV* and *AddEMVScript* operations)

To set the card PIN, perform the following actions:

1. Initialize session – InitSession FIMI-request. The obtained session ID is transmitted in all the requests. In this case, the session is required to securely transfer the FIMI operator password and use the session CardUID. Thus, usage of the session is not mandatory condition.
2. Authenticate operator (optionally) – Logon FIMI request.
3. Search for the card – GetCardInfo FIMI-request. For that purpose, specify the card PAN (in part or in full). The response to the request will contain CardUID that can be sent in all requests instead of PAN. If the “Use Additional Card ID into FIMI Session” flag is enabled in the FIMI terminal settings and CardUID can be calculated by the algorithm, the calculated CardUID can be sent in the request. The response to the request will contain the *IssueTechnology* parameter that must be considered when setting Offline PIN for the chip card.
4. Generate PIN – CryptoFunction FIMI-request (the *GeneratePIN* function). The response will contain a PIN block encrypted under the cryptoserver LMK.
5. Calculate PVV/PIN offset – GetPVV\_PINOffset FIMI-request. The request will contain a PIN block encrypted under LMK with a new PIN that was obtained in p. 4 (if Thales cryptoserver is used, PIN block must be preliminarily converted to the hexadecimal format as the PIN block obtained in p. 4 is returned as a string). If the “PIN Block Encryption” key is defined for the card prefix, the response will contain PIN block encrypted under ZPK.
6. Set PVV – SetCardPVV FIMI-request.
7. To set Offline PIN for the chip card, go to this paragraph. To define the card issue technology (mag stripe or EMV), use the *IssueTechnology* parameter of the GetCardInfo FIMI-request (see p.3). For the chip card, add EMV script to the card – AddEMVScript FIMI-request and send PIN block encrypted under ZPK that was obtained in p. 4 in the request.
8. The party that sends requests to the FIMI driver must store ZPK (e.g.: in HSM Eracom). The clear PIN is calculated from PIN block encrypted under ZPK that was obtained in p.5.
9. End operator session – Logoff FIMI-request.

### Example 2

(with the use of the *POSRequest* operation)

To set the card PIN, perform the following actions:

1. Initialize the session – InitSession FIMI-request. The obtained session ID is transmitted in all the subsequent requests.

2. Authenticate the operator (optionally) - Logon FIMI request.
3. Get PIN. PIN can be entered by the user or generated as a random number (e.g.: in HSM Eracom).
4. PIN block for new PIN is created. The clear PIN block is encrypted under the TPK key defined at the FIMI terminal. For that purpose, the party responsible for sending the request must store TPK (e.g.: in HSM Eracom).
5. Create PIN block for the old PIN. The clear PIN block is encrypted under TPK defined at FIMI terminal. It is permitted to change PIN without checking the old one if no PIN is set for a card or if it has been reset, at that, the new PIN is used instead of the old PIN and its PIN block. The parameters are set up according to the rules of PIN check at TWO level.
6. Change card PIN using the *POSRequest* operation with the *POS PIN Change (139)* transaction of the *Request (200)* type. The request contains the following mandatory fields: PAN, MBR, Track2, Condition, EntryMode, FromAcctType. It also contains PIN block with the new PIN (encrypted under TPK) and PIN block with the old PIN (encrypted under TPK).
7. End operator session – Logoff FIMI-request.

For details, refer to the requests description.

## 6.3 Transactional Mode Usage

The FIMI driver transactional mode is used when setting or changing the chip card PIN and it is planned to set new Offline PIN value when setting new PVV/PIN Offset value. The following FIMI requests must be used: BeginTransaction, Commit and Rollback:

1. BeginTransaction.
2. SetCardPVV. If the request has been successfully executed, the system executes the other requests. Otherwise, rollback is executed and the procedure of setting new PIN is terminated.
3. AddEMVScript.
4. If the previous request has been successfully executed, the system executes the Commit request. Otherwise, rollback is executed.

## 7 CardUID and AccountUID Fields Usage

CardUID is the card or Telebank customer unique identifier. It can be used in requests instead of the PAN and MBR fields. The field can be returned in responses to the *GetCardInfo*, *CreateVCard*, *GetAcctInfo*, *GetAcctStatement*, *GetPersonInfo*, *GetCMSBuffer*, *GetCMSArchive*, *GetCMSCommands*, *GetIssuerHistory*, *GetFIMITrans*, *GetATMAdminTrans* requests.

Depending on the “Use Additional Card Id.” setting, the FIMI driver can interact with various unique identifiers. If the flag is set, the FIMI driver interacts with unique card identifiers received by **TWO** from **TWCMS** at the Refresh processing. In case this setting is applied and the card has no unique identifier in **TWO**, the FIMI driver responses will not contain this field and FIMI operations for this card will be performed using PAN and MBR.

If the “Use Additional Card Id.” flag is unset, the session unique card identifiers are returned in the FIMI driver responses to the requests containing the “FIMI/Session” parameter. These card identifiers are valid only within the FIMI session they were received for. In case of interacting with session card identifiers if the card identifier from another session is used, the card will not be found and a new card identifier will be created for the same card for each new session.

AccountUID is the session unique account identifier within the FIMI session. It can be used in requests instead of the Account and FIID fields. The field can be returned in responses to the *GetCardInfo*, *GetAcctInfo*, *GetPersonInfo*, *GetIssuerHistory*, *GetFIMITrans* requests. The principle of interacting with the account identifier is similar to that of interacting with session card identifiers.

## 8 Request Examples

### 8.1 Request for Working via UAMP

Examples of the terminal requests and **TWO** responses are listed below. The “>In” direction stands for **TWO** input stream, whereas the “<Out” direction stands for **TWO** output stream.

#### 1. Session Initialization, Request for Password Encryption Component:

```
10:16:10.889 >In
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/RetAddress=192.166.22.33
PSFIMI/Clerk=fimiprodPSFIMI/Operation=InitSessionPSFIMI/Ver=9PSFIMI
/InitSession/Rq/NeedDicts=0

10:16:10.953 <Out
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/NextChallenge=06253765PSF
IMI/Response=1PSFIMI/InitSession/Rp/Id=29868
# 10:16:11.000 Disconnected
```

#### 2. Operator Authentication, Request for Allowed Operations List:

```
11:16:32.250 >In
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/RetAddress=192.166.22.33
PSFIMI/Clerk=fimiprodPSFIMI/Password=9CC5530A38A7B667PSFIMI/Operat
ion=LogonPSFIMI/Ver=9PSFIMI/Session=29868

11:16:32.628 <Out
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Response=1PSFIMI/TranId=6
51925PSFIMI/Logon/Rp/ClerkExpiration=3319488000PSFIMI/Logon/Rp/Oper
ations=LogonISSetCard2AcctDescrISGetCardStatementISCNSSCardConfigISC
ardRiskControlISUpdateCard2AcctLinkISSetCardProfileISSetEMVCardPara
msISGetCardInfoISSetCardStatusISGetCardLimitsISSetCardLimitsISSetCar
dPVVISResetBadPINTriesISSetCardExpirationISSetTmpCardLimitsISGetAcc
tInfoISSetAcctStatusISChangeAcctBalancesISGetAcctStatementISCreateA
ccountISGetPersonInfoISSetAcct2CardStatusISAcctDebitISAcctCreditISRe
setCard2AcctLinkISGetRetailerListISGetTermListISGetAcct2CardLimitsIS
SetAcct2CardLimitsISCreateVCardISGetAcctLimitsISSetAcctLimits
```

#### 3. Data Request (Account Details):

```
10:17:10.436 >In
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/RetAddress=192.166.22.33
PSFIMI/Clerk=fimiprodPSFIMI/Password=9CC5530A38A7B667PSFIMI/Operat
ion=GetAcctInfoPSFIMI/GetAcctInfo/Rq/Account=010010PSFIMI/GetAcctI
```

nfo/Rq/InstName=DEMO~~PSFIMI~~/Ver=9~~PSFIMI~~/Session=29868  
10:17:10.453 <Out  
Encoding=Cp866~~PS~~Product=FIMI~~PS~~Ver=1~~PSFIMI~~/Response=1~~PSFIMI~~/TranId=6  
51929~~PSFIMI~~/GetAcctInfo/Rp/Avail=668434~~PSFIMI~~/GetAcctInfo/Rp/Card  
s=4555550000000096~~FS~~0~~FSNIFS~~FE4CEE54386740889EEB9C7BB78E06FD~~PSFIMI~~/  
GetAcctInfo/Rp/Currency=840~~PSFIMI~~/GetAcctInfo/Rp/DropTmpOverOnRef  
resh=0~~PSFIMI~~/GetAcctInfo/Rp/FoundAccount=010010~~PSFIMI~~/GetAcctInfo/  
Rp/LastDepAmount=1~~PSFIMI~~/GetAcctInfo/Rp/LastDepTime=3251628258~~PSFI~~  
MI/GetAcctInfo/Rp/LastTranId=535015~~PSFIMI~~/GetAcctInfo/Rp/LastWdlAm  
ount=14.87~~PSFIMI~~/GetAcctInfo/Rp/LastWdlTime=3253424611~~PSFIMI~~/GetAc  
ctInfo/Rp/Ledger=-746.47~~PSFIMI~~/GetAcctInfo/Rp/MaskBalances=0~~PSFIMI~~/  
GetAcctInfo/Rp/Overdraft=0~~PSFIMI~~/GetAcctInfo/Rp/ParentAcct=010008~~PS~~  
~~FIMI~~/GetAcctInfo/Rp>Status=1~~PSFIMI~~/GetAcctInfo/Rp/TmpOverdraft=-80  
~~PSFIMI~~/GetAcctInfo/Rp/TmpOverdraftExpiration=3301228220~~PSFIMI~~/GetAc  
ctInfo/Rp/Type=1

**PS MS FS NI IS** – special symbols of UAMP protocol (see UAMP specification)

## 8.2 Request for Working via SOAP XML

Interaction is implemented via the SOAP protocol in the ‘request-response’ mode. If an error occurred when processing a request, the server sends a response – Fault.

***It is recommended to set the FIMI terminal version to not lower than 2.0 in the FIMI terminal settings for working via the SOAP XML protocol.***

Interaction order:

### 1. Session Initialization, Request for Password Encryption Component:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <env:Body>
    <m1:InitSessionRq>
      <m1:Request Ver="9.1" Product="FIMI"
      RetAddress="192.166.22.33" Clerk="FIMIXML">
        <m0:NeedDicts>0</m0:NeedDicts>
        <m0:AllVendors>0</m0:AllVendors>
        <m0:AvoidSession>0</m0:AvoidSession>
      </m1:Request>
    </m1:InitSessionRq>
  </env:Body>
</env:Envelope>
```

### Response to Request:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd"
  xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
  <env:Body>
    <m1:InitSessionRp>
      <m1:Response NextChallenge="06253765" Response="1"
      Product="FIMI">
        <m0:Id>29868</m0:Id>
        <m0:NeedCAPAuth>0</m0:NeedCAPAuth>
      </m1:Response>
    </m1:InitSessionRp>
  </env:Body>
</env:Envelope>
```

### 2. Operator Authentication, Request for Allowed Operations List:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <env:Body>
    <m1:LogonRq>
      <m1:Request Ver="9.1" Product="FIMI"
      RetAddress="192.166.22.33" Clerk="FIMIXML" Password="9CC5530A38A7B667"
      Session="29868"/>
    </m1:LogonRq>
  </env:Body>
</env:Envelope>
```

### Response to Request:

```

<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd"
  xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
  <env:Body>
    <m1:LogonRp>
      <m1:Response Response="1" TranId="705180" Product="FIMI">
        <m0:Operations>
          <m0:Row>
            <m0:Value>Logon</m0:Value>
          </m0:Row>
          <m0:Row>
            <m0:Value>GetReport</m0:Value>
          </m0:Row>
          ...
          <m0:Row>
            <m0:Value>POSRequest</m0:Value>
          </m0:Row>
        </m0:Operations>
      </m1:Response>
    </m1:LogonRp>
  </env:Body>
</env:Envelope>

```

### 3. Data Request (account details):

```

<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <env:Body>
    <m1:GetAcctInfoRq>
      <m1:Request Ver="9.1" Product="FIMI" RetAddress="192.166.22.33" Password="9CC5530A38A7B667" Session="29868" Clerk="FIMIXML">
        <m0:Account>010010</m0:Account>
        <m0:InstName>DEMO</m0:InstName>
      </m1:Request>
    </m1:GetAcctInfoRq>
  </env:Body>
</env:Envelope>

```

**Response to Request:**

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd"
  xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
  <env:Body>
    <m1:GetAcctInfoRp>
      <m1:Response Response="1" TranId="705243">
        Product="FIMI">
          <m0:Avail>1336799</m0:Avail>
          <m0:Cards>
            <m0:Row>
              <m0:PAN>4555550000000096</m0:PAN>
              <m0:MBR>0</m0:MBR>
              <m0:Status>1</m0:Status>
            </m0:Row>
            <m0:Row>
              <m0:PAN>912</m0:PAN>
              <m0:MBR>0</m0:MBR>
              <m0:Status>1</m0:Status>
            </m0:Row>
          </m0:Cards>
          <m0:Currency>840</m0:Currency>

          <m0:DropTmpOverOnRefresh>0</m0:DropTmpOverOnRefresh>
          <m0:FoundAccount>010010</m0:FoundAccount>
          <m0:LastDepAmount>9</m0:LastDepAmount>
          <m0:LastDepTime>06.04.2006
          17:29:45</m0:LastDepTime>
          <m0:LastTranId>704344</m0:LastTranId>
          <m0:LastWdlAmount>100</m0:LastWdlAmount>
          <m0:LastWdlTime>06.04.2006
          17:10:14</m0:LastWdlTime>
          <m0:Ledger>272.53</m0:Ledger>
          <m0:MaskBalances>0</m0:MaskBalances>
          <m0:Overdraft>2000</m0:Overdraft>
          <m0:ParentAcct>010008</m0:ParentAcct>
          <m0:Status>3</m0:Status>
          <m0:TmpOverdraft>-80</m0:TmpOverdraft>
          <m0:TmpOverdraftExpiration>11.08.2005
          15:50:20</m0:TmpOverdraftExpiration>
          <m0>Type>1</m0>Type>
        </m1:Response>
      </m1:GetAcctInfoRp>
    </env:Body>
  </env:Envelope>
```

## Response for Declined Transactions:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <env:Body>
        <env:Fault>
            <env:Code>
                <env:Value>env:Receiver</env:Value>
            </env:Code>
            <env:Reason>
                <env:Text xml:lang="en">System error</env:Text>
            </env:Reason>
            <env:Detail>
                <m:DeclineRp
                xmlns:m="http://schemas.compassplus.com/two/1.0/fimi.xsd">
                    <m:Response Product="FIMI" Response="54">
NextChallenge="06253765" Ver="14.3"/>
                    </m:DeclineRp>
                </env:Detail>
            </env:Fault>
        </env:Body>
    </env:Envelope>
```

Where value of the tag *Fault/Detail/DeclineRp/Response/Response* is error code (for the error codes, see the **Response Field Available Values** table, p. 3.1).

Fault/Reason/Text – decline reason.

Fault/Detail/DeclineRp/Response/NextChallenge – value of the password encryptpom component. The terminal is offered to the work with this component at the next request. It is transferred starting from v. 3.6.

Fault/Reason/Ver – protocol maximum version used for the FIMI driver operation.

## 8.3 Request for Working via JMS XML

When working via JMS XML, the structure of requests and responses is similar to that when working via SOAP XML. The difference is that the SOAP envelope is not used. For details, refer to section 8.2.

Example of the *AcctCredit* operation request:

```
<fimi:AcctCreditRq
  xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
  xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
    Password="USER_OFF">
    <fimil:InstName>DEMO</fimil:InstName>
    <fimil:Account>888</fimil:Account>
    <fimil:Amount>100</fimil:Amount>
    <fimil:IgnoreImpact>1</fimil:IgnoreImpact>
    <fimil:NeedNotify>1</fimil:NeedNotify>
  </fimi:Request>
</fimi:AcctCreditRq>
```

Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<m1:AcctCreditRp
  xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <m1:Response Product="FIMI" Response="1" TranId="4048061644"
    Ver="14.9">
    <m0:ApprovalCode>137486</m0:ApprovalCode>
  </m1:Response>
</m1:AcctCreditRp>
```

Fault response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<DeclineRp xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
  xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <Response Product="FIMI" Response="8" DeclineReason="Account
    '88889' for institution DEMO(1) not found or doesn't belong to
    'USER_OFF' clerk access group (0) or belong to restricted
    institution by terminal #170040 or disabled by VIP access
    configuration"/>
</DeclineRp>
```

## 9 Examples of Files Generation for Offline Mode

### 9.1 Examples for Working via UAMP

In this mode the input/output file consists of the UAMP requests/responses (for details, refer to section 8.1) separated by 0x0A (LF). It is allowed to use the FileHeader and FileTrailer optional operations (for details, refer to section 1.2).

Example of input file:

```
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Operation=FileHeaderPSFIMI/FileHeader/Rq/Counters=63LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Clerk=USER_OFFPSFIMI/Language=0PSFIMI/Operation=GetCardInfoPSFIMI/Password=USER_OFFPSFIMI/RetAddress=10.10.98.250FIMI/GetCardInfo/Rq/MBR=0PSFIMI/GetCardInfo/Rq/PAN=4555000088888888LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Clerk=USER_OFFPSFIMI/Password=USER_OFFPSFIMI/Operation=SetAcctStatusPSFIMI/SetAcctStatus/Rq/Account=888PSFIMI/SetAcctStatus/Rq/InstName=DEMOPSFIMI/SetAcctStatus/Rq>Status=1PSFIMI/SetAcctStatus/Rq/ChangeReason=TestLF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Clerk=USER_OFFPSFIMI/Password=USER_OFFPSFIMI/Operation=AcctCreditPSFIMI/AcctCredit/Rq/Account=888PSFIMI/AcctCredit/Rq/InstName=DEMOPSFIMI/AcctCredit/Rq/Amount=100PSFIMI/AcctCredit/Rq/IgnoreImpact=1PSFIMI/AcctCredit/Rq/NeedNotify=1LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Clerk=USER_OFFPSFIMI/Password=USER_OFFPSFIMI/Operation=AcctCreditPSFIMI/AcctCredit/Rq/Account=88889PSFIMI/AcctCredit/Rq/InstName=DEMOPSFIMI/AcctCredit/Rq/Amount=100PSFIMI/AcctCredit/Rq/IgnoreImpact=1PSFIMI/AcctCredit/Rq/NeedNotify=1LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Operation=FileTrailerPSFIMI/FileTrailer/Rq/Counters=63PSFIMI/FileTrailer/Rq/CalculateCRC=1LF
```

Example of output file:

```
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Response=1PSFIMI/FileHeader/Rp/ApprovedAmount=00000000100.000PSFIMI/FileHeader/Rp/ApprovedRequests=00000003PSFIMI/FileHeader/Rp/DeclinedAmount=00000000100.000PSFIMI/FileHeader/Rp/DeclinedRequests=00000001PSFIMI/FileHeader/Rp/TotalAmount=00000000200.000PSFIMI/FileHeader/Rp/TotalRequests=00000004LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Response=1PSFIMI/Ver=14.9PSFIMI/GetCardInfo/Rp/Accounts=888FS3FSNoneFSFSFS2886925.8FS2418414.7FS840...LF
Encoding=Cp866Product=FIMIVer=1FIMI/Response=1FIMI/TranId=4048061607FIMI/Ver=14.9LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Response=1PSFIMI/TranId=4048061608PSFIMI/Ver=14.9PSFIMI/AcctCredit/Rp/ApprovalCode=196900LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/DeclineReason='Account '88889' for institution DEMO(1) not found or doesn't belong to 'USER_OFF' clerk access group (0) or belong to restricted institution by terminal #170040 or disabled by VIP access configuration'PSFIMI/Response=13PSFIMI/Ver=14.9LF
Encoding=Cp866PSProduct=FIMIPSVer=1PSFIMI/Response=1PSFIMI/FileTrailer/Rp/ApprovedAmount=00000000100.000PSFIMI/FileTrailer/Rp/ApprovedRequests=00000003PSFIMI/FileTrailer/Rp/CRC=88ed4bf3PSFIMI/FileTrailer/Rp/DeclinedAmount=00000000100.000PSFIMI/FileTrailer/Rp/DeclinedRequests=00000001PSFIMI/FileTrailer/Rp/TotalAmount=0000000200.000PSFIMI/FileTrailer/Rp/TotalRequests=00000004LF
```

**PS, MS, FS, NI, IS**—special symbols of UAMP protocol (see UAMP specification)

## 9.2 Examples for Working via SOAP

When generating files in the SOAP format, two modes are available:

- The input/output file consists of the SOAP requests/responses with the HTTP header (***the mode is available for the FIMI terminal version 2.0***). It is mandatory to specify the length of the message body in the header. The header and body of the request/response are separated by the sequence of characters CR(0x0D)LF(0x0A) CR(0x0D)LF(0x0A).
- The input/output file consists of the SOAP requests/responses separated by 0x10 (DLE) without the HTTP header (***the mode is available for the FIMI terminal version 3.0 and later versions***).

It is allowed to use the FileHeader and FileTrailer optional operations (for details, refer to section 1.2). It is allowed to use the "space" 0x20, "tabulation" 0x09 and "line feed" 0x0D(0x0A) characters in the incoming request and outgoing response for formatting. When processing the input file, these characters will be ignored.

Example of input file with HTTP header:

```
POST / HTTP/1.0CRLF
Content-Type: text/xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 460CRLF
CRLF
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:FileHeaderRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:Counters>63</fimil:Counters>
            </fimi:Request>
        </fimi:FileHeaderRq>
    </soap:Body>
</soap:Envelope>
POST / HTTP/1.0CRLF
Content-Type: text/xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 491CRLF
CRLF
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:GetCardInfoRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:MBR>0</fimil:MBR>
                <fimil:PAN>4555000088888888</fimil:PAN>
            </fimi:Request>
        </fimi:GetCardInfoRq>
    </soap:Body>
</soap:Envelope>
POST / HTTP/1.0CRLF
Content-Type: text/xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 588CRLF
CRLF
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:SetAcctStatusRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:Account>888</fimil:Account>
                <fimil:InstName>DEMO</fimil:InstName>
                <fimil>Status>1</fimil>Status>
                <fimil:ChangeReason>Test</fimil:ChangeReason>
            </fimi:Request>
        </fimi:SetAcctStatusRq>
    </soap:Body>
</soap:Envelope>
POST / HTTP/1.0CRLF
Content-Type: text/xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 622CRLF
CRLF
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:AcctCreditRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:InstName>DEMO</fimil:InstName>
                <fimil:Account>888</fimil:Account>
                <fimil:Amount>100</fimil:Amount>
                <fimil:IgnoreImpact>1</fimil:IgnoreImpact>
                <fimil:NeedNotify>1</fimil:NeedNotify>
            </fimi:Request>
        </fimi:AcctCreditRq>
    </soap:Body>
</soap:Envelope>
POST / HTTP/1.0CRLF
Content-Type: text/xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 624CRLF
CRLF
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:AcctCreditRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:InstName>DEMO</fimil:InstName>
                <fimil:Account>88889</fimil:Account>
                <fimil:Amount>100</fimil:Amount>
                <fimil:IgnoreImpact>1</fimil:IgnoreImpact>
                <fimil:NeedNotify>1</fimil:NeedNotify>
            </fimi:Request>
        </fimi:AcctCreditRq>
    </soap:Body>
</soap:Envelope>
POST / HTTP/1.0CRLF
```

```
Content-Type: text/xml; charset="utf-8" CRLF
Connection: close CRLF
Content-Length: 502 CRLF
CRLF
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
    xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
    xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:FileTrailerRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
                Password="USER_OFF">
                <fimil:Counters>63</fimil:Counters>
                <fimil:CalculateCRC>1</fimil:CalculateCRC>
            </fimi:Request>
        </fimi:FileTrailerRq>
    </soap:Body>
</soap:Envelope>
```

**Example of output file with HTTP header:**

```
HTTP/1.0 200 OK CRLF
Content-Type: application/soap+xml; charset="utf-8" CRLF
Connection: close CRLF
Content-Length: 677 CRLF
CRLF
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
    xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd"
    xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
    <env:Body>
        <m1:FileHeaderRp>
            <m1:Response Response="1">
                Product="FIMI"><m0:ApprovedAmount>00000000100.000</m0:ApprovedAmount>
                <m0:ApprovedRequests>00000003</m0:ApprovedRequests>
                <m0:DeclinedAmount>00000000100.000</m0:DeclinedAmount>
                <m0:DeclinedRequests>00000001</m0:DeclinedRequests>
                <m0:TotalAmount>00000000200.000</m0:TotalAmount>
                <m0:TotalRequests>00000004</m0:TotalRequests>
            </m1:Response>
        </m1:FileHeaderRp>
    </env:Body>
</env:Envelope>
HTTP/1.0 200 OK CRLF
Content-Type: application/soap+xml; charset="utf-8" CRLF
Connection: close CRLF
Content-Length: 7011 CRLF
CRLF
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
    xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd"
    xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
    <env:Body>
        <m1:GetCardInfoRp>
            <m1:Response Response="1" Ver="14.9" Product="FIMI"><m0:Accounts><m0:Row>
                <m0:AcctNo>888</m0:AcctNo>
                <m0>Status>3</m0>Status>
                <m0:Descr>None</m0:Descr>
                <m0:LedgerBalance>2887325.8</m0:LedgerBalance>
                <m0:AvailBalance>2418814.7</m0:AvailBalance>
                <m0:Currency>840</m0:Currency>
            </m0:Accounts>
        </m1:Response>
    </env:Body>
</env:Envelope>
```

```
<m0:Type>1</m0:Type>
<m0:AccountStatus>1</m0:AccountStatus>
</m0:Row>
...
</m1:GetCardInfoRp>
</env:Body>
</env:Envelope>
HTTP/1.0 200 OKCRLF
Content-Type: application/soap+xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 399CRLF
CRLF
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
<env:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
<env:Body>
<m1:SetAcctStatusRp >
<m1:Response Response="1" TranId="4048061617" Ver="14.9"
Product="FIMI"></m1:Response>
</m1:SetAcctStatusRp>
</env:Body>
</env:Envelope>
HTTP/1.0 200 OKCRLF
Content-Type: application/soap+xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 435CRLF
CRLF
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
<env:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
<env:Body>
<m1:AcctCreditRp >
<m1:Response Response="1" TranId="4048061618" Ver="14.9"
Product="FIMI"><m0:ApprovalCode>483141</m0:ApprovalCode>
</m1:Response>
</m1:AcctCreditRp>
</env:Body>
</env:Envelope>
HTTP/1.0 500 Internal Server ErrorCRLF
Content-Type: application/soap+xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 677CRLF
CRLF
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:Body>
<env:Fault>
<env:Code>
<env:Value>env:Receiver</env:Value>
</env:Code>
<env:Reason>
<env:Text xml:lang="en">Account '88889' for institution DEMO(1) not
found or doesn't belong to 'USER_OFF' clerk access group (0) or
belong to restricted institution by terminal #170040 or disabled by VIP access
configuration</env:Text>
</env:Reason>
<env:Detail><m:DeclineRp
xmlns:m="http://schemas.compassplus.com/two/1.0/fimi.xsd"><m:Response
```

```

Product="FIMI" Response="13"
Ver="14.9"/></m:DeclineRp></env:Detail></env:Fault></env:Body></env:Envelope>
HTTP/1.0 200 OKCRLF
Content-Type: application/soap+xml; charset="utf-8"CRLF
Connection: closeCRLF
Content-Length: 705CRLF
CRLF
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd"
xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd">
<env:Body>
<m1:FileTrailerRp >
<m1:Response Response="1">
Product="FIMI"><m0:ApprovedAmount>00000000100.000</m0:ApprovedAmount>
<m0:ApprovedRequests>000000003</m0:ApprovedRequests>
<m0:CRC>fcbe52e4</m0:CRC>
<m0:DeclinedAmount>00000000100.000</m0:DeclinedAmount>
<m0:DeclinedRequests>000000001</m0:DeclinedRequests>
<m0:TotalAmount>00000000200.000</m0:TotalAmount>
<m0:TotalRequests>000000004</m0:TotalRequests>
</m1:Response>
</m1:FileTrailerRp>
</env:Body>
</env:Envelope>

```

#### Example of input file without HTTP header:

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:FileHeaderRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:Counters>63</fimil:Counters>
            </fimi:Request>
        </fimi:FileHeaderRq>
    </soap:Body>
</soap:Envelope>DLE
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:GetCardInfoRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:MBR>0</fimil:MBR>
                <fimil:PAN>4555000088888888</fimil:PAN>
            </fimi:Request>
        </fimi:GetCardInfoRq>
    </soap:Body>
</soap:Envelope>DLE
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:SetAcctStatusRq>

```

```

<fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
    <fimil:Account>888</fimil:Account>
    <fimil:InstName>DEMO</fimil:InstName>
    <fimil>Status>1</fimil>Status>
    <fimil:ChangeReason>Test</fimil:ChangeReason>
</fimi:Request>
</fimi:SetAcctStatusRq>
</soap:Body>
</soap:Envelope>DLE
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:AcctCreditRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:InstName>DEMO</fimil:InstName>
                <fimil:Account>888</fimil:Account>
                <fimil:Amount>100</fimil:Amount>
                <fimil:IgnoreImpact>1</fimil:IgnoreImpact>
                <fimil:NeedNotify>1</fimil:NeedNotify>
            </fimi:Request>
        </fimi:AcctCreditRq>
    </soap:Body>
</soap:Envelope>DLE
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:AcctCreditRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:InstName>DEMO</fimil:InstName>
                <fimil:Account>88889</fimil:Account>
                <fimil:Amount>100</fimil:Amount>
                <fimil:IgnoreImpact>1</fimil:IgnoreImpact>
                <fimil:NeedNotify>1</fimil:NeedNotify>
            </fimi:Request>
        </fimi:AcctCreditRq>
    </soap:Body>
</soap:Envelope>DLE
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <soap:Header/>
    <soap:Body>
        <fimi:FileTrailerRq>
            <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
                <fimil:Counters>63</fimil:Counters>
                <fimil:CalculateCRC>1</fimil:CalculateCRC>
            </fimi:Request>
        </fimi:FileTrailerRq>
    </soap:Body>
</soap:Envelope>DLE

```

Example of output file without HTTP header:

```

<?xml version="1.0" encoding="UTF-8" ?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <s:Body>
    <m1:FileHeaderRp>
      <m1:Response Product="FIMI" Response="1" Ver="14.9">
        <m0:ApprovedAmount>000000000100.000</m0:ApprovedAmount>
        <m0:ApprovedRequests>00000003</m0:ApprovedRequests>
        <m0:DeclinedAmount>000000000100.000</m0:DeclinedAmount>
        <m0:DeclinedRequests>00000001</m0:DeclinedRequests>
        <m0:TotalAmount>000000000200.000</m0:TotalAmount>
        <m0:TotalRequests>00000004</m0:TotalRequests>
      </m1:Response>
    </m1:FileHeaderRp>
  </s:Body>
</s:Envelope>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <s:Body>
    <m1:GetCardInfoRp>
      <m1:Response Product="FIMI" Response="1" Ver="14.9">
        <m0:Accounts>
          <m0:Row>
            <m0:AcctNo>888</m0:AcctNo>
            <m0:Status>3</m0:Status>
            <m0:Descr>None</m0:Descr>
            <m0:LedgerBalance>2888025.8</m0:LedgerBalance>
            <m0:AvailBalance>2419514.7</m0:AvailBalance>
            <m0:Currency>840</m0:Currency>
            <m0:Type>1</m0:Type>
            <m0:AccountStatus>1</m0:AccountStatus>
          </m0:Row>
        ...
        <m1:Response>
      </m1:Response>
    </m1:GetCardInfoRp>
  </s:Body>
</s:Envelope>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <s:Body>
    <m1:SetAcctStatusRp>
      <m1:Response Product="FIMI" Response="1" TranId="4048061645" Ver="14.9"/>
    </m1:SetAcctStatusRp>
  </s:Body>
</s:Envelope>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
  <s:Body>
    <m1:AcctCreditRp>
      <m1:Response Product="FIMI" Response="1" TranId="4048061646" Ver="14.9">
        <m0:ApprovalCode>815746</m0:ApprovalCode>
      </m1:Response>
    </m1:AcctCreditRp>
  </s:Body>
</s:Envelope>DLE

```

```
</m1:AcctCreditRp>
</s:Body>
</s:Envelope>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
<s:Body>
<s:Fault>
<s:Code>
<s:Value>s:Receiver</s:Value>
</s:Code>
<s:Reason>
<s:Text xml:lang="en">Account '88889' for institution DEMO(1) not found or
doesn't belong to 'USER_OFF' clerk access group (0) or belong to restricted
institution by terminal #170040 or disabled by VIP access configuration</s:Text>
</s:Reason>
<s:Detail>
<m1:DeclineRp>
<m1:Response Product="FIMI" Response="13" Ver="14.9"/>
</m1:DeclineRp>
</s:Detail>
</s:Fault>
</s:Body>
</s:Envelope>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
<s:Body>
<m1:FileTrailerRp>
<m1:Response Product="FIMI" Response="1" Ver="14.9">
<m0:ApprovedAmount>00000000100.000</m0:ApprovedAmount>
<m0:ApprovedRequests>000000003</m0:ApprovedRequests>
<m0:CRC>11a26a9e</m0:CRC>
<m0:DeclinedAmount>000000000100.000</m0:DeclinedAmount>
<m0:DeclinedRequests>000000001</m0:DeclinedRequests>
<m0:TotalAmount>00000000200.000</m0:TotalAmount>
<m0:TotalRequests>000000004</m0:TotalRequests>
</m1:Response>
</m1:FileTrailerRp>
</s:Body>
</s:Envelope>DLE
```

## 9.3 Examples for Working via XML

*The mode is available if the FIMI terminal version is later than 3.0.*

In this mode the input/output file consists of the XML requests/responses (for details, refer to section 8.3) separated by 0x10 (DLE). It is allowed to use the FileHeader and FileTrailer optional operations (for details, refer to section 1.2). It is allowed to use the "space" 0x20, "tabulation" 0x09 and "line feed" 0x0D(0x0A) characters in the incoming request and outgoing response for formatting. When processing the input file, these characters will be ignored.

Example of input file:

```
<fimi:FileHeaderRq xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
        <fimil:Counters>63</fimil:Counters>
    </fimi:Request>
</fimi:FileHeaderRq>DLE
<fimi:GetCardInfoRq xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
        <fimil:MBR>0</fimil:MBR>
        <fimil:PAN>4555000088888888</fimil:PAN>
    </fimi:Request>
</fimi:GetCardInfoRq>DLE
<fimi:SetAcctStatusRq
xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
        <fimil:Account>888</fimil:Account>
        <fimil:InstName>DEMO</fimil:InstName>
        <fimil>Status>1</fimil>Status>
        <fimil:ChangeReason>Test</fimil:ChangeReason>
    </fimi:Request>
</fimi:SetAcctStatusRq>DLE
<fimi:AcctCreditRq xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
        <fimil:InstName>DEMO</fimil:InstName>
        <fimil:Account>888</fimil:Account>
        <fimil:Amount>100</fimil:Amount>
        <fimil:IgnoreImpact>1</fimil:IgnoreImpact>
        <fimil:NeedNotify>1</fimil:NeedNotify>
    </fimi:Request>
</fimi:AcctCreditRq>DLE
<fimi:AcctCreditRq xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
        <fimil:InstName>DEMO</fimil:InstName>
        <fimil:Account>88889</fimil:Account>
        <fimil:Amount>100</fimil:Amount>
        <fimil:IgnoreImpact>1</fimil:IgnoreImpact>
        <fimil:NeedNotify>1</fimil:NeedNotify>
    </fimi:Request>
</fimi:AcctCreditRq>DLE
```

```
<fimi:FileTrailerRq xmlns:fimi="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:fimil="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <fimi:Request Ver="3.5" Product="FIMI" Clerk="USER_OFF"
Password="USER_OFF">
        <fimil:Counters>63</fimil:Counters>
        <fimil:CalculateCRC>1</fimil:CalculateCRC>
    </fimi:Request>
</fimi:FileTrailerRq>
```

**Example of output file:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<m1:FileHeaderRp xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <m1:Response Product="FIMI" Response="1" Ver="14.9">
        <m0:ApprovedAmount>000000000100.000</m0:ApprovedAmount>
        <m0:ApprovedRequests>000000003</m0:ApprovedRequests>
        <m0:DeclinedAmount>000000000100.000</m0:DeclinedAmount>
        <m0:DeclinedRequests>000000001</m0:DeclinedRequests>
        <m0:TotalAmount>000000000200.000</m0:TotalAmount>
        <m0:TotalRequests>000000004</m0:TotalRequests>
    </m1:Response>
</m1:FileHeaderRp>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<m1:GetCardInfoRp xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <m1:Response Product="FIMI" Response="1" Ver="14.9">
        <m0:Accounts>
            <m0:Row>
                <m0:AcctNo>888</m0:AcctNo>
                <m0:Status>3</m0:Status>
                <m0:Descr>None</m0:Descr>
                <m0:LedgerBalance>2887925.8</m0:LedgerBalance>
                <m0:AvailBalance>2419414.7</m0:AvailBalance>
            </m0:Row>
        ...
    </m1:GetCardInfoRp>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<m1:SetAcctStatusRp xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <m1:Response Product="FIMI" Response="1" TranId="4048061643"
Ver="14.9"/>
</m1:SetAcctStatusRp>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<m1:AcctCreditRp xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <m1:Response Product="FIMI" Response="1" TranId="4048061644" Ver="14.9">
        <m0:ApprovalCode>137486</m0:ApprovalCode>
    </m1:Response>
</m1:AcctCreditRp>DLE
<?xml version="1.0" encoding="UTF-8" ?>
<DeclineRp xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
<Response Product="FIMI" Response="8" DeclineReason="Account '88889' for
institution DEMO(1) not found or doesn't belong to 'USER_OFF' clerk access group
(0) or belong to restricted institution by terminal #170040 or disabled by VIP
access configuration"/>
</DeclineRp>DLE
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<m1:FileTrailerRp xmlns:m1="http://schemas.compassplus.com/two/1.0/fimi.xsd"
xmlns:m0="http://schemas.compassplus.com/two/1.0/fimi_types.xsd">
    <m1:Response Product="FIMI" Response="1" Ver="14.9">
        <m0:ApprovedAmount>00000000100.000</m0:ApprovedAmount>
        <m0:ApprovedRequests>00000003</m0:ApprovedRequests>
        <m0:CRC>7cf5d26b</m0:CRC>
        <m0:DeclinedAmount>00000000100.000</m0:DeclinedAmount>
        <m0:DeclinedRequests>00000001</m0:DeclinedRequests>
        <m0:TotalAmount>00000000200.000</m0:TotalAmount>
        <m0:TotalRequests>00000004</m0:TotalRequests>
    </m1:Response>
</m1:FileTrailerRp>DLE
```