SaleToPOIDeviceRequest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lvl | SaleToPOIDeviceRequest | Mult | Constraint | Rule | Usage |
| 1 | Header | [1..1] |  |  | <*::Header41<Hdr>::Header41* |
| 2 | MessageFunction | [1..1] |  | \* | v*::RetailerMessage1Code*value must be: SDDR <br/>*<MsgFctn>::RetailerMessage1Code* |
| 2 | ProtocolVersion | [1..1] |  | \* | V*::Max6Text*Value must be: 8.0 <br/>*<PrtcolVrsn>::Max6Text* |
| 2 | ExchangeIdentification | [1..1] |  |  | <*::Max35Text<XchgId>::Max35Text* |
| 2 | CreationDateTime | [1..1] |  |  | <*::ISODateTime<CreDtTm>::ISODateTime* |
| 2 | InitiatingParty | [1..1] |  |  | S[GenericIdentification177](#GenericIdentification177)*::GenericIdentification177*See MDR for sub elements and <a href="#GenericIdentification177">GenericIdentification177</a><br/>*<InitgPty>::GenericIdentification177* |
| 2 | RecipientParty | [0..1] |  |  | S[GenericIdentification177](#GenericIdentification177)*::GenericIdentification177*See MDR for sub elements and <a href="#GenericIdentification177">GenericIdentification177</a><br/>*<RcptPty>::GenericIdentification177* |
| 2 | Traceability | [0..\*] |  |  | S[Traceability8](#Traceability8)*::Traceability8*See MDR for sub elements and <a href="#Traceability8">Traceability8</a><br/>*<Tracblt>::Traceability8* |
| 1 | DeviceRequest | [1..1] |  |  | <*::DeviceRequest6<DvcReq>::DeviceRequest6* |
| 2 | Environment | [0..1] |  |  | S[CardPaymentEnvironment79](#CardPaymentEnvironment79)*::CardPaymentEnvironment79*See MDR for sub elements and <a href="#CardPaymentEnvironment79">CardPaymentEnvironment79</a><br/>*<Envt>::CardPaymentEnvironment79* |
| 2 | Context | [0..1] |  |  | S[CardPaymentContext30](#CardPaymentContext30)*::CardPaymentContext30*See MDR for sub elements and <a href="#CardPaymentContext30">CardPaymentContext30</a><br/>*<Cntxt>::CardPaymentContext30* |
| 2 | ServiceContent | [1..1] |  | \* | List of specific services for DeviceRequest.- **DDYQ: DeviceDisplayRequest** : *One System requests the other to display a message for cashier or customer.*- **DINQ: DeviceInputRequest** : *One system requests to the other System to get data input.* - **DPRQ: DevicePrintRequest** : *One system requests to the other System to print data.*- **DSOQ: DevicePlaySoundRequest** : *One system requests to the Other System to play a sound.*- **DSIQ: DeviceSecureInputRequest** : *One system requests to the Other System to securely get data input (e.g. for PIN).*- **DCIQ: DeviceInitialisationCardReaderRequest** : *Service to send parameters to use when card reader initializes a new communication with the card.*- **DCAQ: DeviceSendApplicationProtocolDataUnitCardReaderRequest** : *A service to send commands to a card.*- **DCPQ: DevicePowerOffCardReaderRequest** : *The Sale system requests to the POI System to power off the card reader.*- **DCOQ: DeviceTransmissionMessageRequest** : *The Sale system requests to the POI System to transmit a message (for instance to a mobile server).*- **DINO: DeviceInputNotification** : *One system sends a notification to the POI System to update a input request.::RetailerService8Code* <br/><br/>List of specific services for DeviceRequest.<br/>- <b>DDYQ: DeviceDisplayRequest</b> : <i>One System requests the other to display a message for cashier or customer.</i><br/>- <b>DINQ: DeviceInputRequest</b> : <i>One system requests to the other System to get data input. </i><br/>- <b>DPRQ: DevicePrintRequest</b> : <i>One system requests to the other System to print data.</i><br/>- <b>DSOQ: DevicePlaySoundRequest</b> : <i>One system requests to the Other System to play a sound.</i><br/>- <b>DSIQ: DeviceSecureInputRequest</b> : <i>One system requests to the Other System to securely get data input (e.g. for PIN).</i><br/>- <b>DCIQ: DeviceInitialisationCardReaderRequest</b> : <i>Service to send parameters to use when card reader initializes a new communication with the card.</i><br/>- <b>DCAQ: DeviceSendApplicationProtocolDataUnitCardReaderRequest</b> : <i>A service to send commands to a card.</i><br/>- <b>DCPQ: DevicePowerOffCardReaderRequest</b> : <i>The Sale system requests to the POI System to power off the card reader.</i><br/>- <b>DCOQ: DeviceTransmissionMessageRequest</b> : <i>The Sale system requests to the POI System to transmit a message (for instance to a mobile server).</i><br/>- <b>DINO: DeviceInputNotification</b> : <i>One system sends a notification to the POI System to update a input request.</i><br/>*<SvcCntt>::RetailerService8Code* |
| 2 | DisplayRequest | [0..1] |  | C1,C2 | <*::DeviceDisplayRequest5<DispReq>::DeviceDisplayRequest5* |
| 3 | DisplayOutput | [1..\*] |  |  | S[ActionMessage10](#ActionMessage10)*::ActionMessage10*See MDR for sub elements and <a href="#ActionMessage10">ActionMessage10</a><br/>*<DispOutpt>::ActionMessage10* |
| 2 | InputRequest | [0..1] |  | C1, C3 | <*::DeviceInputRequest5<InptReq>::DeviceInputRequest5* |
| 3 | DisplayOutput | [0..1] |  |  | S[ActionMessage10](#ActionMessage10)*::ActionMessage10*See MDR for sub elements and <a href="#ActionMessage10">ActionMessage10</a><br/>*<DispOutpt>::ActionMessage10* |
| 3 | InputData | [1..1] |  |  | <*::InputData5<InptData>::InputData5* |
| 4 | DeviceType | [1..1] |  |  | <Type of the Logical device located on a Sale Terminal or a POI Terminal, in term of class of information to output (display, print or store), or input (keyboard) for the Cashier   or the Customer.- **CHIN: CashierInput** : *Any kind of keyboard allowing all or part of the commands of the Input message request from the Sale System to the POI System (InputCommand data element). The output device attached to this input device is the CashierDisplay device.*- **CUIN: CustomerInput** : *Any kind of keyboard allowing all or part of the commands of the Input message request from the Sale System to the POI System (InputCommand data element).::SaleCapabilities2Code*<br/>Type of the Logical device located on a Sale Terminal or a POI Terminal, in term of class of information to output (display, print or store), or input (keyboard) for the Cashier   or the Customer.<br/>- <b>CHIN: CashierInput</b> : <i>Any kind of keyboard allowing all or part of the commands of the Input message request from the Sale System to the POI System (InputCommand data element). The output device attached to this input device is the CashierDisplay device.</i><br/>- <b>CUIN: CustomerInput</b> : <i>Any kind of keyboard allowing all or part of the commands of the Input message request from the Sale System to the POI System (InputCommand data element).</i><br/>*<DvcTp>::SaleCapabilities2Code* |
| 4 | InformationQualifier | [1..1] |  |  | <Qualification of the information to sent to an output logical device, to display or print to the Cashier or the Customer.- **CUSA: CustomerAssistance** : *Input of the Cardholder POI interface which can be entered by the Cashier to assist the Customer.*- **DISP: Display** : *Standard display interface.*- **DOCT: Document** : *When the POI System wants to print specific document (check, dynamic currency conversion ...). Used by the Sale System when the printer is not located on the Sale System.*- **ERRO: Error** : *The information is related to an error situation occurring on the message sender.*- **INPT: Input** : *Answer to a question or information to be entered by the Cashier or the Customer, at the request of the POI Terminal or the Sale Terminal.*- **POIR: POIReplication** : *Information displayed on the Cardholder POI interface, replicated on the Cashier interface.*- **RCPT: Receipt** : *Where you print the Payment receipt that could be located on the Sale System or in some cases a restricted Sale ticket on the POI Terminal.*- **SOND: Sound** : *Standard sound interface.*- **STAT: Status** : *The information is a new state on which the message sender is entering. For instance, during a payment, the POI could display to the Cashier that POI request an authorisation to the host acquirer.*- **VCHR: Voucher** : *Coupons, voucher or special ticket generated by the POI or the Sale System and to be printed.::InformationQualify1Code*<br/>Qualification of the information to sent to an output logical device, to display or print to the Cashier or the Customer.<br/>- <b>CUSA: CustomerAssistance</b> : <i>Input of the Cardholder POI interface which can be entered by the Cashier to assist the Customer.</i><br/>- <b>DISP: Display</b> : <i>Standard display interface.</i><br/>- <b>DOCT: Document</b> : <i>When the POI System wants to print specific document (check, dynamic currency conversion ...). Used by the Sale System when the printer is not located on the Sale System.</i><br/>- <b>ERRO: Error</b> : <i>The information is related to an error situation occurring on the message sender.</i><br/>- <b>INPT: Input</b> : <i>Answer to a question or information to be entered by the Cashier or the Customer, at the request of the POI Terminal or the Sale Terminal.</i><br/>- <b>POIR: POIReplication</b> : <i>Information displayed on the Cardholder POI interface, replicated on the Cashier interface.</i><br/>- <b>RCPT: Receipt</b> : <i>Where you print the Payment receipt that could be located on the Sale System or in some cases a restricted Sale ticket on the POI Terminal.</i><br/>- <b>SOND: Sound</b> : <i>Standard sound interface.</i><br/>- <b>STAT: Status</b> : <i>The information is a new state on which the message sender is entering. For instance, during a payment, the POI could display to the Cashier that POI request an authorisation to the host acquirer.</i><br/>- <b>VCHR: Voucher</b> : <i>Coupons, voucher or special ticket generated by the POI or the Sale System and to be printed.</i><br/>*<InfQlfr>::InformationQualify1Code* |
| 4 | InputCommand | [1..1] |  |  | <Type of requested input- **DCSG: DecimalString** : *Wait for a string of digit characters with a decimal point, the length range could be specified.*- **DGSG: DigitString** : *Wait for a string of digit characters.*- **GAKY: GetAnyKey** : *Wait for a key pressed on the Terminal, to be able to read the message displayed on the Terminal.*- **GCNF: GetConfirmation** : *Wait for a confirmation Yes (Y) or No (N) on the Sale System. Wait for a confirmation (Valid or Cancel button) on the POI Terminal. The result of the command is a Boolean: True or False.*- **GFKY: GetFunctionKey** : *Wait for a function key pressed on the Terminal: From POI, Valid, Clear, Correct, Generic Function key number. From Sale, Generic Function key.*- **GMNE: GetMenuEntry** : *To choose an entry among a list of entries (all of them are not necessary selectable). The OutputFormat has to be MenuEntry.*- **PSWD: Password** : *Request to enter a password with masked characters while typing the password.*- **SITE: SiteManager** : *Wait for a confirmation Yes (Y) or No (N) of the Site Manager on the Sale System.*- **TXSG: TextString** : *Wait for a string of alphanumeric characters.*- **HTML: XHTMLText** : *Wait for a XHTML data.*- **SIGN: Signature** : *Request to wait for signature.::InputCommand1Code*<br/>Type of requested input<br/>- <b>DCSG: DecimalString</b> : <i>Wait for a string of digit characters with a decimal point, the length range could be specified.</i><br/>- <b>DGSG: DigitString</b> : <i>Wait for a string of digit characters.</i><br/>- <b>GAKY: GetAnyKey</b> : <i>Wait for a key pressed on the Terminal, to be able to read the message displayed on the Terminal.</i><br/>- <b>GCNF: GetConfirmation</b> : <i>Wait for a confirmation Yes (Y) or No (N) on the Sale System. Wait for a confirmation (Valid or Cancel button) on the POI Terminal. The result of the command is a Boolean: True or False.</i><br/>- <b>GFKY: GetFunctionKey</b> : <i>Wait for a function key pressed on the Terminal: From POI, Valid, Clear, Correct, Generic Function key number. From Sale, Generic Function key.</i><br/>- <b>GMNE: GetMenuEntry</b> : <i>To choose an entry among a list of entries (all of them are not necessary selectable). The OutputFormat has to be MenuEntry.</i><br/>- <b>PSWD: Password</b> : <i>Request to enter a password with masked characters while typing the password.</i><br/>- <b>SITE: SiteManager</b> : <i>Wait for a confirmation Yes (Y) or No (N) of the Site Manager on the Sale System.</i><br/>- <b>TXSG: TextString</b> : <i>Wait for a string of alphanumeric characters.</i><br/>- <b>HTML: XHTMLText</b> : <i>Wait for a XHTML data.</i><br/>- <b>SIGN: Signature</b> : <i>Request to wait for signature.</i><br/>*<InptCmd>::InputCommand1Code* |
| 4 | NotifyCardInputFlag | [1..1] |  |  | F*::TrueFalseIndicator*Flag of notification of card to be entered in the POI card reader. <br/>*<NtfyCardInptFlg>::TrueFalseIndicator* |
| 4 | MaximumInputTime | [0..1] |  |  | <*::Number<MaxInptTm>::Number* |
| 4 | InputText | [0..1] |  |  | S[ActionMessage10](#ActionMessage10)*::ActionMessage10*See MDR for sub elements and <a href="#ActionMessage10">ActionMessage10</a><br/>*<InptTxt>::ActionMessage10* |
| 4 | ImmediateResponseFlag | [0..1] |  | C12 | F*::TrueFalseIndicator*Flag to request Immediate response without waiting for the completion of the command. default False. <br/>*<ImdtRspnFlg>::TrueFalseIndicator* |
| 4 | WaitUserValidationFlag | [0..1] |  |  | F*::TrueFalseIndicator*Flag to confirm by the user the entered characters, when the maximum allowed length is reached. default False. <br/>*<WaitUsrVldtnFlg>::TrueFalseIndicator* |
| 4 | BeepKeyFlag | [0..1] |  |  | F*::TrueFalseIndicator*Flag to indicate that when the user presses a key, a beep has to begenerated. default True. <br/>*<BeepKeyFlg>::TrueFalseIndicator* |
| 4 | GlobalCorrectionFlag | [0..1] |  | C17 | F*::TrueFalseIndicator*Flag to correct all characters (True) or just the last one (False). default False. <br/>*<GblCrrctnFlg>::TrueFalseIndicator* |
| 4 | DisableCancelFlag | [0..1] |  | C19 | F*::TrueFalseIndicator*Flag to deactivate the "Cancel" function key. default False. <br/>*<DsblCclFlg>::TrueFalseIndicator* |
| 4 | DisableCorrectFlag | [0..1] |  | C18 | F*::TrueFalseIndicator*Flag to deactivate the "Correct" function key. default False. <br/>*<DsblCrrctFlg>::TrueFalseIndicator* |
| 4 | DisableValidFlag | [0..1] |  | C20 | F*::TrueFalseIndicator*Flag to disable the "Valid" function key. default False. <br/>*<DsblVldFlg>::TrueFalseIndicator* |
| 4 | MenuBackFlag | [0..1] |  | C21 | F*::TrueFalseIndicator*Flag to enable the "Back" function key to go the upper level. default True. <br/>*<MenuBckFlg>::TrueFalseIndicator* |
| 2 | PrintRequest | [0..1] |  | C1, C4 | <*::DevicePrintRequest5<PrtReq>::DevicePrintRequest5* |
| 3 | DocumentQualifier | [1..1] |  |  | <Specifies a type of financial or commercial document.- **JNRL: Journal** : *When the POI or the Sale System wants to store a message on the journal printer or electronic journal of the Sale Terminal (it is sometimes a Sale Logging/Journal Printer).*- **CRCP: CustomerReceipt** : *When the Sale System requires the POI system to print the Customer receipt.*- **HRCP: CashierReceipt** : *When the Sale system print the Cashier copy of the Payment receipt.*- **SRCP: SaleReceipt** : *When the Sale System requires the POI system to print the Sale receipt.*- **RPIN: RelatedPaymentInstruction** : *Document is a linked payment instruction to which the current payment instruction is related, for example, in a cover scenario.*- **VCHR: Voucher** : *Document is an electronic payment document.::DocumentType7Code*<br/>Specifies a type of financial or commercial document.<br/>- <b>JNRL: Journal</b> : <i>When the POI or the Sale System wants to store a message on the journal printer or electronic journal of the Sale Terminal (it is sometimes a Sale Logging/Journal Printer).</i><br/>- <b>CRCP: CustomerReceipt</b> : <i>When the Sale System requires the POI system to print the Customer receipt.</i><br/>- <b>HRCP: CashierReceipt</b> : <i>When the Sale system print the Cashier copy of the Payment receipt.</i><br/>- <b>SRCP: SaleReceipt</b> : <i>When the Sale System requires the POI system to print the Sale receipt.</i><br/>- <b>RPIN: RelatedPaymentInstruction</b> : <i>Document is a linked payment instruction to which the current payment instruction is related, for example, in a cover scenario.</i><br/>- <b>VCHR: Voucher</b> : <i>Document is an electronic payment document.</i><br/>*<DocQlfr>::DocumentType7Code* |
| 3 | ResponseMode | [1..1] |  | \* | VMessage response awaited by the initiator of the Request.- **SEND: EndOfPlay** : *The Response is required at the end of play.*- **IMMD: Immediate** : *The Message Response is immediate, after taking into account the request.*- **NREQ: NotRequired** : *The Message Response is not required, except in case of error.*- **PEND: PrintEnd** : *The Print Response is required at the end of print.::ResponseMode2Code*Value of ResponseMode "SEND" (Sound End) is forbidden in case of PrintRequest. <br/><br/>Message response awaited by the initiator of the Request.<br/>- <b>SEND: EndOfPlay</b> : <i>The Response is required at the end of play.</i><br/>- <b>IMMD: Immediate</b> : <i>The Message Response is immediate, after taking into account the request.</i><br/>- <b>NREQ: NotRequired</b> : <i>The Message Response is not required, except in case of error.</i><br/>- <b>PEND: PrintEnd</b> : <i>The Print Response is required at the end of print.</i><br/>*<RspnMd>::ResponseMode2Code* |
| 3 | IntegratedPrintFlag | [0..1] |  | C22 | d*::TrueFalseIndicator*default False <br/>*<IntgrtdPrtFlg>::TrueFalseIndicator* |
| 3 | RequiredSignatureFlag | [0..1] |  |  | d*::TrueFalseIndicator*default False <br/>*<ReqrdSgntrFlg>::TrueFalseIndicator* |
| 3 | OutputContent | [1..1] |  |  | S[ActionMessage10](#ActionMessage10)*::ActionMessage10*See MDR for sub elements and <a href="#ActionMessage10">ActionMessage10</a><br/>*<OutptCntt>::ActionMessage10* |
| 2 | PlayResourceRequest | [0..1] |  | C1, C5 | <*::DevicePlayResourceRequest1<PlayRsrcReq>::DevicePlayResourceRequest1* |
| 3 | ResponseMode | [0..1] |  | \* | VMessage response awaited by the initiator of the Request.- **SEND: EndOfPlay** : *The Response is required at the end of play.*- **IMMD: Immediate** : *The Message Response is immediate, after taking into account the request.*- **NREQ: NotRequired** : *The Message Response is not required, except in case of error.*- **PEND: PrintEnd** : *The Print Response is required at the end of print.::ResponseMode2Code*Value of ResponseMode "SEND" (Sound End) is forbidden in case of PrintRequest. <br/><br/>Message response awaited by the initiator of the Request.<br/>- <b>SEND: EndOfPlay</b> : <i>The Response is required at the end of play.</i><br/>- <b>IMMD: Immediate</b> : <i>The Message Response is immediate, after taking into account the request.</i><br/>- <b>NREQ: NotRequired</b> : <i>The Message Response is not required, except in case of error.</i><br/>- <b>PEND: PrintEnd</b> : <i>The Print Response is required at the end of print.</i><br/>*<RspnMd>::ResponseMode2Code* |
| 3 | ResourceAction | [1..1] |  | C23, C24 | <Type of action to perform on a media resource.- **PAUS: Pause** : *Pause the media resource in progress as specified in the message.*- **STAS: Play** : *Start the media resource as specified in the message.*- **LOOP: PlayInLoop** : *Play in a loop the media resource as specified in the message.*- **RESU: Resume** : *Resume the progress of the media resource as specified in the message.*- **DVOL: SetDefaultVolume** : *Set the default volume of sounds.*- **STOS: Stop** : *Stop the media resource in progress.::ResourceAction1Code*<br/>Type of action to perform on a media resource.<br/>- <b>PAUS: Pause</b> : <i>Pause the media resource in progress as specified in the message.</i><br/>- <b>STAS: Play</b> : <i>Start the media resource as specified in the message.</i><br/>- <b>LOOP: PlayInLoop</b> : <i>Play in a loop the media resource as specified in the message.</i><br/>- <b>RESU: Resume</b> : <i>Resume the progress of the media resource as specified in the message.</i><br/>- <b>DVOL: SetDefaultVolume</b> : <i>Set the default volume of sounds.</i><br/>- <b>STOS: Stop</b> : <i>Stop the media resource in progress.</i><br/>*<RsrcActn>::ResourceAction1Code* |
| 3 | SoundVolume | [0..1] |  | C23 | <*::PercentageRate<SoundVol>::PercentageRate* |
| 3 | DisplayResolution | [0..1] |  |  | <*::Max35Text<DispRsltn>::Max35Text* |
| 3 | Resource | [0..1] |  | C24 | <*::ResourceContent1<Rsrc>::ResourceContent1* |
| 4 | ResourceType | [1..1] |  |  | <Type of resource.- **TEXT: TextToSpeech** : *Voice synthesis.*- **URLI: UniformResourceIdentifier** : *String of characters that unambiguously identifies a particular resource.::ResourceType1Code*<br/>Type of resource.<br/>- <b>TEXT: TextToSpeech</b> : <i>Voice synthesis.</i><br/>- <b>URLI: UniformResourceIdentifier</b> : <i>String of characters that unambiguously identifies a particular resource.</i><br/>*<RsrcTp>::ResourceType1Code* |
| 4 | ResourceFormat | [0..1] |  |  | <Type of sound to play.- **MSGR: MessageRef** : *Reference of a preloaded text to play.*- **SNDR: SoundRef** : *Preloaded sound File.*- **TEXT: Text** : *Text to play.::SoundFormat1Code*<br/>Type of sound to play.<br/>- <b>MSGR: MessageRef</b> : <i>Reference of a preloaded text to play.</i><br/>- <b>SNDR: SoundRef</b> : <i>Preloaded sound File.</i><br/>- <b>TEXT: Text</b> : <i>Text to play.</i><br/>*<RsrcFrmt>::SoundFormat1Code* |
| 4 | Language | [0..1] |  |  | <*::LanguageCode<Lang>::LanguageCode* |
| 4 | ResourceReference | [0..1] |  |  | <*::Max1025Text<RsrcRef>::Max1025Text* |
| 3 | TimingSlot | [0..1] |  |  | <Specifies the processing position.- **AFTE: After** : *Specifies that the transaction/instruction is to be executed after the linked transaction/instruction.*- **WITH: With** : *Specifies that the transaction/instruction is to be executed with the linked transaction/instruction.*- **BEFO: Before** : *Specifies that the transaction/instruction is to be executed before the linked transaction/instruction.*- **INFO: Information** : *Specifies that the transactions/instructions are linked for information purposes only.::ProcessingPosition2Code*<br/>Specifies the processing position.<br/>- <b>AFTE: After</b> : <i>Specifies that the transaction/instruction is to be executed after the linked transaction/instruction.</i><br/>- <b>WITH: With</b> : <i>Specifies that the transaction/instruction is to be executed with the linked transaction/instruction.</i><br/>- <b>BEFO: Before</b> : <i>Specifies that the transaction/instruction is to be executed before the linked transaction/instruction.</i><br/>- <b>INFO: Information</b> : <i>Specifies that the transactions/instructions are linked for information purposes only.</i><br/>*<TmgSlot>::ProcessingPosition2Code* |
| 2 | SecureInputRequest | [0..1] |  | C1, C6 | <*::DeviceSecureInputRequest5<ScrInptReq>::DeviceSecureInputRequest5* |
| 3 | PINRequestType | [1..1] |  |  | <Type of PIN Service.- **PIAE: PINAcquisitionEncryption** : *The cardholder enters the PIN, the POI enciphers the PIN Block and provides it as a result to the Sale System.*- **PIAV: PINAcquisitionVerification** : *The Cardholder enters the PIN and the POI verifies it.*- **PIVO: PINVerifyOnly** : *The Sale System send a previous keyed PIN and the POI verifies it.::PINRequestType1Code*<br/>Type of PIN Service.<br/>- <b>PIAE: PINAcquisitionEncryption</b> : <i>The cardholder enters the PIN, the POI enciphers the PIN Block and provides it as a result to the Sale System.</i><br/>- <b>PIAV: PINAcquisitionVerification</b> : <i>The Cardholder enters the PIN and the POI verifies it.</i><br/>- <b>PIVO: PINVerifyOnly</b> : <i>The Sale System send a previous keyed PIN and the POI verifies it.</i><br/>*<PINReqTp>::PINRequestType1Code* |
| 3 | PINVerificationMethod | [0..1] |  | C25 | <*::Max35Text<PINVrfctnMtd>::Max35Text* |
| 3 | MaximumWaitingTime | [0..1] |  | C26 | <*::Number<MaxWtgTm>::Number* |
| 3 | BeepKeyFlag | [0..1] |  |  | F*::TrueFalseIndicator*Flag to indicate that when the user presses a key, a beep has to begenerated. default True. <br/>*<BeepKeyFlg>::TrueFalseIndicator* |
| 3 | CardholderPIN | [0..1] |  | C27 | <*::OnLinePIN10<CrdhldrPIN>::OnLinePIN10* |
| 4 | EncryptedPINBlock | [1..1] |  |  | S[ContentInformationType35](#ContentInformationType35)*::ContentInformationType35*See MDR for sub elements and <a href="#ContentInformationType35">ContentInformationType35</a><br/>*<NcrptdPINBlck>::ContentInformationType35* |
| 4 | PINFormat | [1..1] |  |  | <PIN (Personal Identification Number) format used before encryption.- **ISO0: ISO0** : *PIN diversified with the card account number, conforming to the standard ISO 9564-2.*- **ISO1: ISO1** : *PIN completed with random padding characters, conforming to the standard ISO 9564-2.*- **ISO2: ISO2** : *PIN without diversification characters, conforming to the standard ISO 9564-2.*- **ISO3: ISO3** : *PIN diversified with the card account number and random characters, conforming to the standard ISO 9564-2.*- **ISO4: ISO4** : *PIN format used with AES encryption, conforming to the new ISO SC2 format.*- **ISO5: ISO5** : *Alternative PIN format used with AES encryption, conforming to the new ISO SC2 format.::PINFormat3Code*<br/>PIN (Personal Identification Number) format used before encryption.<br/>- <b>ISO0: ISO0</b> : <i>PIN diversified with the card account number, conforming to the standard ISO 9564-2.</i><br/>- <b>ISO1: ISO1</b> : <i>PIN completed with random padding characters, conforming to the standard ISO 9564-2.</i><br/>- <b>ISO2: ISO2</b> : <i>PIN without diversification characters, conforming to the standard ISO 9564-2.</i><br/>- <b>ISO3: ISO3</b> : <i>PIN diversified with the card account number and random characters, conforming to the standard ISO 9564-2.</i><br/>- <b>ISO4: ISO4</b> : <i>PIN format used with AES encryption, conforming to the new ISO SC2 format.</i><br/>- <b>ISO5: ISO5</b> : <i>Alternative PIN format used with AES encryption, conforming to the new ISO SC2 format.</i><br/>*<PINFrmt>::PINFormat3Code* |
| 4 | AdditionalInput | [0..1] |  |  | <*::Max35Text<AddtlInpt>::Max35Text* |
| 2 | InitialisationCardReaderRequest | [0..1] |  | C1, C7 | <*::DeviceInitialisationCardReaderRequest5<InitlstnCardRdrReq>::DeviceInitialisationCardReaderRequest5* |
| 3 | WarmResetFlag | [0..1] |  |  | <*::TrueFalseIndicator<WarmRstFlg>::TrueFalseIndicator* |
| 3 | ForceEntryMode | [0..\*] |  |  | <Type of reading of the card data.- **TAGC: Tag** : *Tag reading capabilities (RFID, etc.).*- **PHYS: Physical** : *Keyboard entry or OCR reading of embossing or printed data, either at time of transaction or after the event.*- **BRCD: BarCode** : *Bar code.*- **MGST: MagneticStripe** : *Magnetic stripe.*- **CICC: ICC** : *ICC (Integrated Circuit Card) with contact containing software applications conform to ISO 7816.*- **DFLE: AccountData** : *Account data on file.*- **CTLS: ProximityReader** : *Contactless proximity reader.*- **ECTL: EMVProximityReader** : *Contactless proximity reader, with application conform to the standard EMV (standard initiated by Europay, Mastercard and Visa).*- **CDFL: CardOnFile** : *Card information are stored on a file.*- **SICC: SynchronousIntegratedCircuitCard** : *Synchronous ICC - (Integrated Circuit Card) with contact.*- **UNKW: Unknown** : *Unknown card reading capability.*- **QRCD: QRCode** : *Quick response code.*- **OPTC: OpticalCode** : *Optical coded reading capabilities (e.g. barcode, QR code, etc.)::CardDataReading8Code*<br/>Type of reading of the card data.<br/>- <b>TAGC: Tag</b> : <i>Tag reading capabilities (RFID, etc.).</i><br/>- <b>PHYS: Physical</b> : <i>Keyboard entry or OCR reading of embossing or printed data, either at time of transaction or after the event.</i><br/>- <b>BRCD: BarCode</b> : <i>Bar code.</i><br/>- <b>MGST: MagneticStripe</b> : <i>Magnetic stripe.</i><br/>- <b>CICC: ICC</b> : <i>ICC (Integrated Circuit Card) with contact containing software applications conform to ISO 7816.</i><br/>- <b>DFLE: AccountData</b> : <i>Account data on file.</i><br/>- <b>CTLS: ProximityReader</b> : <i>Contactless proximity reader.</i><br/>- <b>ECTL: EMVProximityReader</b> : <i>Contactless proximity reader, with application conform to the standard EMV (standard initiated by Europay, Mastercard and Visa).</i><br/>- <b>CDFL: CardOnFile</b> : <i>Card information are stored on a file.</i><br/>- <b>SICC: SynchronousIntegratedCircuitCard</b> : <i>Synchronous ICC - (Integrated Circuit Card) with contact.</i><br/>- <b>UNKW: Unknown</b> : <i>Unknown card reading capability.</i><br/>- <b>QRCD: QRCode</b> : <i>Quick response code.</i><br/>- <b>OPTC: OpticalCode</b> : <i>Optical coded reading capabilities (e.g. barcode, QR code, etc.)</i><br/>*<ForceNtryMd>::CardDataReading8Code* |
| 3 | LeaveCardFlag | [0..1] |  |  | <*::TrueFalseIndicator<LeavCardFlg>::TrueFalseIndicator* |
| 3 | MaximumWaitingTime | [0..1] |  | C26 | <*::Number<MaxWtgTm>::Number* |
| 3 | DisplayOutput | [0..1] |  |  | S[ActionMessage10](#ActionMessage10)*::ActionMessage10*See MDR for sub elements and <a href="#ActionMessage10">ActionMessage10</a><br/>*<DispOutpt>::ActionMessage10* |
| 2 | CardReaderAPDURequest | [0..1] |  | C1, C8 | <*::DeviceSendApplicationProtocolDataUnitCardReaderRequest1<CardRdrAPDUReq>::DeviceSendApplicationProtocolDataUnitCardReaderRequest1* |
| 3 | Class | [1..1] |  |  | <*::Min1Max256Binary<Clss>::Min1Max256Binary* |
| 3 | Instruction | [1..1] |  |  | <*::Min1Max256Binary<Instr>::Min1Max256Binary* |
| 3 | Parameter | [1..1] |  |  | <*::Min1Max256Binary<Param1>::Min1Max256Binary* |
| 3 | Parameter | [1..1] |  |  | <*::Min1Max256Binary<Param2>::Min1Max256Binary* |
| 3 | Data | [0..1] |  | C28 | <*::Min1Max256Binary<Data>::Min1Max256Binary* |
| 3 | ExpectedLength | [0..1] |  |  | <*::Min1Max256Binary<XpctdLngth>::Min1Max256Binary* |
| 2 | PowerOffCardReaderRequest | [0..1] |  | C1, C9 | <*::DevicePoweroffCardReaderRequest5<PwrOffCardRdrReq>::DevicePoweroffCardReaderRequest5* |
| 3 | PowerOffMaximumWaitingTime | [0..1] |  |  | <*::Number<PwrOffMaxWtgTm>::Number* |
| 3 | DisplayOutput | [0..1] |  |  | S[ActionMessage10](#ActionMessage10)*::ActionMessage10*See MDR for sub elements and <a href="#ActionMessage10">ActionMessage10</a><br/>*<DispOutpt>::ActionMessage10* |
| 2 | TransmissionRequest | [0..1] |  | C1, C10 | <*::DeviceTransmitMessageRequest2<TrnsmssnReq>::DeviceTransmitMessageRequest2* |
| 3 | DestinationAddress | [1..1] |  |  | S[NetworkParameters7](#NetworkParameters7)*::NetworkParameters7*See MDR for sub elements and <a href="#NetworkParameters7">NetworkParameters7</a><br/>*<DstnAdr>::NetworkParameters7* |
| 3 | MaximumTransmissionTime | [1..1] |  |  | <*::Number<MaxTrnsmssnTm>::Number* |
| 3 | MaximumWaitingTime | [0..1] |  | C26 | <*::Number<MaxWtgTm>::Number* |
| 3 | MessageToSend | [1..1] |  |  | <*::Max100KBinary<MsgToSnd>::Max100KBinary* |
| 2 | InputNotification | [0..1] |  | C1, C11 | <*::DeviceInputNotification5<InptNtfctn>::DeviceInputNotification5* |
| 3 | ExchangeIdentification | [1..1] |  |  | <*::Max35Text<XchgId>::Max35Text* |
| 3 | OutputContent | [1..1] |  |  | S[ActionMessage10](#ActionMessage10)*::ActionMessage10*See MDR for sub elements and <a href="#ActionMessage10">ActionMessage10</a><br/>*<OutptCntt>::ActionMessage10* |
| 2 | SupplementaryData | [0..\*] |  |  | <*::SupplementaryData1<SplmtryData>::SupplementaryData1* |
| 3 | PlaceAndName | [0..1] |  |  | <*::Max350Text<PlcAndNm>::Max350Text* |
| 3 | Envelope | [1..1] |  |  | <*::SupplementaryDataEnvelope1<Envlp>::SupplementaryDataEnvelope1* |
| 1 | SecurityTrailer | [0..1] |  |  | S[ContentInformationType33](#ContentInformationType33)*::ContentInformationType33*See MDR for sub elements and <a href="#ContentInformationType33">ContentInformationType33</a><br/>*<SctyTrlr>::ContentInformationType33* |