Technical Documentation
Preapproved Payment Files

Contents

I. Document Revision Tracking

Version	Date	Name	Description
1.0.0	04.12.2020	BG	
1.0.1	14.12.2020	BG	Corrected type to Object in OrganizationIdentification
1.0.2	29.01.2021	BG	Removed unsupported IBAN. Moved description from Identification element.
1.0.3	02.02.2021	BG	Clarify codes for Organization Identification
1.0.4	22.11.2021	MH	Updated new emails
1.0.5	15.03.2023	MH	Updated new emails
1.0.6	16.07.2024	MH	Removed Telepay, not supported

1. Introduction

Direkte Remitting is a service provided by Mastercard Payment Services for Norwegian banks that gives the possibility to send one file containing several assignments related to payouts from different banks. It is a bank independent solution for pre-approved payment files. The banks sell this service to their Corporate Customers and the agreement for use of the service is set up between the company and the bank.

Anti Money Laundry act §13 states that the banks are obliged to introduce a stricter control of which individuals initiate payments from corporate accounts through file-based services. The banks are required to be AML compliant and as a result of Mastercard Payment Services providing Direkte Remittering to banks, Mastercard Payment Services Direkte Remittering service is updated to fulfil this requirement.

This document describes how a Corporate Payment Initiator (CPI) can prepare and send pre- approved payment files to the “Direkte Remittering” service. To use the service, all the procedures inthis document must be followed. The central part of the solution is to package the payment files in an Associated Signature Container (ASiC) along with approval data.

1.1. Target Group

CPIs that want to send preapproved payment files to “Direkte Remittering”, for accounts in banksthat require this approval.

1.2. Scope and Limitations

Scope of this version:

- Supported file formats:
  ISO20022 pain.001

- BBSFormat

- Approval data containing National Identity Numbers (NIN) must be included

- File sending channel: eNett/kundeportal: The
  ASiC must have file extension .asice
  ASiC name must start with AML.PAYMENT.*

- If authorisation fails for an approval in a file, only that corresponding assignment with transaction will be rejected, the approved Assignment with transactions will be processed. Approval rejections will be reported on existing receipts from the Direkte Remittering service.

- Receipts will be returned unsigned

1.3. References

ETSI ASiC standard specification

2. Requirements

Id	Requirement
R1	The CPI sending payment files into the value chain for pre-approved must adhere to thescope as defined in this document and appendices.
R2	The CPI sending payment order files into the value chain for pre-approved payments must establish the necessary agreements with the banks (Ref 3.1).
R3	The CPI sending payment files into the value chain for pre-approved payments must adhereto the technical flows as described in this document and appendices.
R4	The CPI sending payment files into the value chain for pre-approved payments must use the standards and file formats as described in this document and appendices.
R5	The CPI sending payment files into the value chain for pre-approved payments must use the protocols as described in 4.1.
R6	The CPI sending payment files into the value chain for pre-approved payments must implement the necessary user enrollment and user payment authentication mechanism inthe corporate system as defined in this document and appendices. BankID must be used in one of these ways: 1. Use BankID to authenticate the user during enrollment of user into the corporate system.The user must be re-authenticated at least once a month using BankID. Approval of payments can be done using a strong authentication method that are not BankID. 2. Use BankID as strong authentication or signing method every time when approving payments in the corporate system.
R7	The CPI sending payment files into the value chain for pre-approved payments must implement a way of storing authentication context and authentication details so that bank can retrieve a “proof package” based on the Authentication Reference provided in theapproval data XML file.
R8	Any deviations to a party’s implementation must be agreed upon with Mastercard Payment Services and bank.

3. How to get Started

1. Contact your bank to get information about the necessary preparations. Different banks canhave different requirements related to the AML procedure. Sign agreement(s) with relevant bank(s).

2. Order enterprise Certificate (Commfides or Buy pass).

3. Order BankID for approver authentication in the CPI system.

4. Implement in CPI system:

    Solution for strong authentication or signing of validated approvers. This could be e.g. approved multi-factor mobile or BankID authentication/signing. Strong authenticationfollows the PSD2 definition of strong customer authentication:

    > ’strong customer authentication’ means an authentication based on the use of two or more elements categorised as knowledge (something only the user knows), possession (something only the user possesses) and inheritance (something the useris) that are independent, in that the breach of one does not compromise the reliability of the others, and is designed in such a way as to protect the confidentiality of the authentication data

    Use an Associated Signature Container (ASiC) to package the payment files and approval information.

    Storing audit trail “package” of users approving payments according to requirements from banks. This information can later be requested by bank for use in customer/legal dialogue.

5. Contact Mastercard Payment Services to get a test-user and instructions on how to conductthe test: support.norway@mastercard.com.

4. Process Description

4.1. Mastercard Payment Services Payment File Reception

When Mastercard Payment Services receive the ASiC with the preapproved payment file(s) from the customer, this will happen:

1. Receive file via SFTP or Customer Portal and perform transport level controls.
2. Unpackage ASiC, validate structure and ASiC signature.

3. Validate approver data against bank registry.

4. If everything is OK: Payment files are sent to processing
   – otherwise error receipt is returned to CPI.

5. Store audit trail of files and process according to requirements from banks.

5. High Level Description

This chapter details what must be done to send preapproved payment files to Mastercard Payment Services.

5.1. Approval of Payments in CPI System

The end user will approve payments in the CPI system. The authentication or signing method must be a bank approved multi factor authentication. This will usually be some solution based on PKI – e.g.BankID authentication or signing.

The CPI system must store an information package that documents the approval performed by the user. This package will typically consist of log entries, results from authentication/signing and other context information that can be used as proof of user approval in a customer/legal dialogue. A reference to this package is put into the approver data file that is accompanying the payment file.

5.2. Creating the Preapproved Payment ASiC

The data container will contain two file categories (in addition to the container files):

1. Payment files:

- ISO20022 pain.001
- BBSFormat

2. Approver data files:

- There must be one approver data file for each payment file.
- Contains information about who has approved specific assignments.
- The first version will use national identification numbers as identification.
- The file also contains information about which signing/authentication methods that has been used in the approval process, and a reference to the information package in the customer system.

The payment files and approval data files will be packaged into an ASiC as described in the nextchapter.

5.3. Creating the Associated Signature Container

An Associated Signature Container (ASiC) will be used to package payment files and approval data.An ASiC is basically a compressed (zip) file that contains the data files together with a standard ASiC manifest file and a digital signature (PKCS#7) that ensures that no forgery or tampering of the ASiC content can happen during transport. The manifest file must contain hashes of all the payment and approval files. The hash method used must be SHA-256.

The digital signatures will be created using the private key in an enterprise certificate. The key mustbe RSA-2048 bit. The certificate must be obtained from a reputable issuer.

5.4. Sending the ASiC

The ASiC will be sent through the agreed channel to Mastercard Payment Services.

6. Appendix 1 ASiC Structure

To sign the payment files without affecting the files themselves, the data will be put in an ASiC.

ASiC example structure:

/META-INF/ASiCManifest.xml

/META-INF/signature.p7s

/Approval Data/ApprovalData1.xml

/Approval Data/ApprovalData2.xml

/Approval Data/ApprovalData3.xml

/mimetype

/paymentfile1

/paymentfile2

/paymentfile3

NB: Use Mastercard Payment Services naming convention for payment files inside the ASiC. Approval data files can have any name, but there must be an approval file corresponding to eachpayment file. Hashes of all payment and approval files in the container must be included in the ASiCManifest file.

7. Appendix 2: Approval Data File

Or	Tag	Level	Mult	Type	Comments
	<Message Reference format=”ISO20022” |format=”BBSFormat”>	1	1	String	Reference to file in ASiC-E container. Filename must follow Mastercard Payment Services naming convention
	<Message Identification>	1	1	String	Reference as assigned by the instruction part. XML ISO20022 = <MsgId>) / (BETFOR00) / BBSFormat reference
	<Initiating Party>	1	1	Object	Party that initiates the ASiC
	<Identification>	2	1	Object	
	<Organisation Identification>	3	1	Object	See “Organisation Identification”
	<Payment Information>	1	1..n	Object	
	<Debtor>	2	0..1	Object	
	<Identification>	3	0..1	Object	
	<Organisation Identification>	4	0..1	Object	See “Organisation Identification”
	<Debtor Account>	2	1..n	Object	
	<Identification>	3	1	Object	
	<Other>	4	1	Object	
	<Identification>	5	1	Object	
	<Scheme Name>	5	1	Object	
	<Code>	6	1	String	Used codes: • BAN See ExternalAccountIdentification1Code”at https://www.iso20022.org/ catalogue-messages/additional- content-messages/external-code-sets
	<Payment Type Information>	2	1	Object	
	<Category Purpose>	3	1	Object	

	<Code>	4	1	String	Used codes: • OTHR • SALA See “ExternalCategoryPurpose1Code”at https://www.iso20022.org/catalogue-messages/additional-content-messages/external-code-sets
	<Approver Identification>	2	1..2	Object	First or First and second approver
	<Identification>	3	1	Object	
	<Private Identification>	4	1	Object	
	<Other>	5	1	Object	
	<Identification>	6	1	String	NIN or file name reference
	<Scheme Name>	6	1	Object	
{Or	<Code>	7	1	String	Used codes: • SOSE See “ExternalPersonIdentification1Code”at https://www.iso20022.org/ catalogue-messages/additional- content-messages/external-code- sets, Only SOSE will be used
Or}	<Proprietary>	7	1	String	NORWEGIAN_BANKID
	<Authentication Information>	3	1	Object	
	<Authentication Method Vendor>	4	1	Object	
	<Name>	5	1	String	ERP name, BANKID
	<Authentication Method>	4	1	Object	
	<Name>	5	1	String	ERP method, example ERP_2FA_ mobile, BANKID_MOBILE
	<Authentication Reference>	4	1	String	Reference for authentication
	<Authentication Date Time>	4	0..1	String	Reference for date/time. Recommended field.

Organisation Identification

Tag	Level	Mult	Type	Comments
<Other>	1	0..n	Object	
<Identification>	2	1	String	Id registered in bank system. Usually the organisation number but can also be a bank customer id.
<Scheme Name>	2	1	Object	
<Code>	3	1	String	Type of identification for the initiating party. Code CUST for identification. Elements with code BANK is accepted but will be ignored.

8. Appendix 3: ASiCManifest

Message Item	Tag Name	Level	Comments
ASiCManifest	<ASiC Manifest>	1	
Signature Reference	<Sig Reference>	2	URI=”META-INF/signature.p7s” Mime Type=”application/x-pkcs7-signature”
Data Object Reference	<Data Object Reference>	2	
Digest Method	<Digest Method>	3	Digest method algorithm
Digest Value	<Digest Value>	3	Digest Value contains the Base64 encoded result of applying the hash algorithm to the transformed resource(s)defined in the Data Object Reference element attributes.

9. Appendix 4: Receipt

Message Item	Tag	Level	Comments
Customer Payment Status Report	<CstmrPmtStsRpt>	1	
Group Header	<GrpHdr>	2	
Message Identification	<MsgId>	3	
Creation Date Time	<CreDtTm>	3	
Initiating Party	<InitgPty>	3	
Identification	<Id>	4	
Organisation Identification	<OrgId>	5	
BICOrBEI	<BICOrBEI>	5	
Other	<Othr>	5	
Identification	<Id>	6	Subscription Number from group header
Scheme Name	<Schme Nm>	6	

Code	<Cd>	7	Type of identification for the initiating party - CUST
OriginalGroupInformationAndSta	<OrgnlGrpInfAndSts>	2	
Original Message File Name	<OrgnlMsgFileName>	3	File name
Group Status	<GrpSts>	3	
Status Reason Information	<StsRsnInf>	3	
Reason	<Rsn>	4	
Code	<Cd>	5	FF02 - SyntaxError. Approval Data rejected OK – Validation successful
Additional Information	< AddtlInf>	4	

10. Appendix 5: File Examples

Some example files are given in this chapter.

10.1. Approval Data File with One Debit Account and Division

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Document xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”>
    <Message Reference format=”ISO20022”> pain001_paymentfile1.xml</Message Reference>
    <Message Identification>908711</Message Identification>
    <Initiating Party>
        <Identification>
            <OrganisationIdentification>
                <Other>
                    <Identification>00099940204</Identification>
                    <Scheme Name>
                        <Code>CUST</Code>
                    </Scheme Name>
                </Other>
                <Other>
                    <Identification>TESTXML</Identification>
                    <Scheme Name>
                        <Code>BANK</Code>
                    </Scheme Name>
                </Other>
            </Organisation Identification>
        </Identification>
    </Initiating Party>
    <Payment Information>
```

```
        <Debtor Account>
            <Identification>
                <Other>
                    <Identification>70560525808</Identification>
                    <Scheme Name>
                        <Code>BBAN</Code>
                    </Scheme Name>
                </Other>
            </Identification>
        </Debtor Account>
    <Payment Type Information>
        <Category Purpose>
            <Code>OTHR</Code>
        </Category Purpose>
    </Payment Type Information>
    <Approver Identification>
        <Identification>
            <Private Identification>
                <Other>
                    <Identification>21026099999</Identification>
                    <Scheme Name>
                        <Code>SOSE</Code>
                    </Scheme Name>
                </Other>
            </Private Identification>
        </Identification>
        <Authentication Information>
            <Authentication Method Vendor>
                <Name>ERP</Name>
            </Authentication Method Vendor>
            <Authentication Method>
                <Name>ERP_2FA_mobile</Name>
            </Authentication Method>
            <Authentication Reference>
                1234-121212-1213_908712
            </Authentication Reference>
        </Authentication Information>
    </Approver Identification>
</Payment Information>
```

```
</Document>
```

10.2. Example with More Than One Division and One Debit Account

NB: This example is incomplete.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Document xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”>
    <Message Reference format=”ISO20022”>pain001_paymentfile1.xml</Message Reference>
    <Message Identification>908711</Message Identification>
    <Initiating Party>
        <Identification>
            <Organisation Identification>
                <Other>
                    <Identification>987654321</Identification>
                    <Scheme Name>
                        <Code>CUST</Code>
                    </Scheme Name>
                </Other>
                <Other>
                    <Identification>MYDIV1</Identification>
                    <Scheme Name>
                        <Code>BANK</Code>
                    </Scheme Name>
                </Other>
            </Organisation Identification>
        </Identification>
    </Initiating Party>
    <Payment Information>
    <Debtor>
        <Identification>
            <Organisation Identification>
                <Other>
                    <Identification>987654321</Identification>
                    <Scheme Name>
                        <Code>CUST</Code>
                    </Scheme Name>
                </Other>
```

```
                        <Other>
                                <Identification>MYDIV2</Identification>
                                <Scheme Name>
                                        <Code>BANK</Code>
                                </Scheme Name>
                        </Other>
                </Organisation Identification>
        </Identification>
</Debtor>
-->
<Debtor Account>
         <Identification>
                 <Other>
                        <Identification>5201010123</Identification>
                        <Scheme Name>
                                <Code>BBAN</Code>
                        </Scheme Name>
                 </Other>
         </Identification>
</Debtor Account>
<Payment Type Information>
        <Category Purpose>
                <Code>SALA</Code> <!-- OPT: OTHR -->
          </Category Purpose>
  </Payment Type Information>
<Approver Identification>
        <Identification>
                <Private Identification>
                <Other>
                <Identification>12345678901</Identification>
                <Scheme Name>
                <Code>SOSE</Code>
                </SchemeName>
                </Other>
                </PrivateIdentification>
        </Identification>
<AuthenticationInformation>
<AuthenticationMethodVendor>
```

```
        <Name>ERP</Name>

    </AuthenticationMethodVendor>
    <AuthenticationMethod>
        <Name>ERP_2FA_MOBILE</Name>
    </AuthenticationMethod>
        <AuthenticationReference>1234-121212-1213</AuthenticationReference>
    </AuthenticationInformation>
</ApproverIdentification>
<!-- OPTIONAL: Second approver with authentication info>
<ApproverIdentification>
...
</ApproverIdentification>
-->
</PaymentInformation>
<!-- OPTIONAL: Multiple of debit side>
<PaymentInformation>
...
</PaymentInformation>
-->
```

10.3. Example ASiCManifest

```
<?xml version=”1.0” encoding=”UTF-8” standalone=”yes”?>
<ASiCManifest xmlns=”http://uri.etsi.org/02918/v1.2.1#” xmlns:ns2=”http://www.w3.org/2000/09/xmldsig#”>
    <SigReference URI=”META-INF/signature.p7s”
    MimeType=”application/x-pkcs7-signature”/>
        <DataObjectReference URI=”pain001_paymentfile1.xml”  MimeType=”application/
xml”>
        <ns2:DigestMethod Algorithm=”http://www.w3.org/2001/04/xmlenc#sha256”/>
        <ns2:DigestValue>hashOfpain001_paymentfile1.xml</ns2:DigestValue>
    </DataObjectReference>
        <DataObjectReference URI=”ApprovalData/ApprovalData1.xml”
MimeType=”application/xml”>
        <ns2:DigestMethod Algorithm=”http://www.w3.org/2001/04/xmlenc#sha256”/>
        <ns2:DigestValue>hashOfApprovalData1.xmlTextFile</ns2:DigestValue>
    </DataObjectReference>
</ASiCManifest>
```

10.4. Example Mimetype File

```
Application/vnd.etsi.asic-e+zip
```