EBICS - Implementation Guide in France
Version 2.1.4
This version is consistent with the V2.4.2 of the specifications

NOTICE TO THE READER

The documentation relating to the EBICS protocol, designed in its original version by the ZKA (German equivalent of CFONB) was written in German and then translated into English.

Release 2.4 is the first joint French-German release. An update V2.4.1 has been published in September 2009.
However the implementation of EBICS in France must be adapted to the national context (ETEBAC 3&5 migration, national payment instruments...).

The latest version of the specifications EBICS v2.4.2, published in February 2010, supplements or clarifies the version 2.4.1 for the implementation of the personal signature attached. An exhaustive list of clarifications is available in the specifications, this guide corresponds to version v2.4.2.

The CFONB has developed this implementation guide to EBICS in France, for the following reasons:

1. the French practices in customer to bank files transfers had to be taken into account to ensure the smooth replacement of ETEBAC 3 (phase 1) and 5 (phase 2) protocols as well as the maximum flexibility in the migration,
2. to replace ETEBAC3, adaptations were needed in relation to the securised functionalities offered by the transport protocol,
3. to replace ETEBAC 5, adaptations were needed to process execution orders electronically signed and transported with the orders to execute.
4. the list of orders (Order types) in the protocol includes the orders used in the German and French contexts. This guide identifies the precise sub-set recommended for the orders used in France,
5. EBICS protocol is to be used for the remittance of SEPA payments but also for domestic payments (VO, LCR...) and customer reporting (account statements...). The description of the national settings was necessary.

EBICS protocol can also be used for other banking purposes.

Note:
To facilitate the updating and the loading by editors, the following lists are not attached to this document but are available as separate documents on CFONB website: www.cfonb.org with this Implementation Guide (Documentation> ETEBAC Migration to SWIFTNet and EBICS):

- Annex A2 : FileFormat/Request Type - file naming
- Annex A3.1: List of error messages related to certificates
- Annex A5 : Payment Status Report format
- Annex A6 : Example of Hash calculation

Amendment history

Version	Date	Chapter	Type[1]	Description
VO 1.1	24/4/2009			First English version. In phase with French V1.1
VO 1.2	9/2009	All		Consistency with the version 2.4.1 of the specifications
		1.1	C	Complementarities of the migration phases
		1.2.5	A	ASCII / EBCDIC Coding
		All	C	PSR is at disposal and not sent
		2.1.2.2	C	Remarks on signature computing
		2.1.4	C	Case of the service providers
		A2	A	File naming : Addition of the protocol level of the PSR (type pain.002.001.02.ack)
		A.5	A / M	Format of the Payment Status Report
VO 2.0	04/2010	All	A / M	Personal signature management Replacement of the phase (1&2) concept for designation of EBICS Profile T and EBICS profile TS
		2.1.5	A	Multi users management within the same customer
		A.5	M	Payment Status Report Format: annex in a dedicated document
VO 2.1	05/2010	All	C	Clarifications on V2.0
		1.2.5	M	No rejection of remittance when ES quantity = 1 with 2 signatures received Rejection when more than 2 signatures are received
VO2.1.1	01/02/2011	A 3.2	M	KeyUsage – pg 25,26,27
VO2.1.2	21/06/2011	2.1.2.1	C	These clarifications allow to respect the security rules for personal certificates
VO2.1.3	13/10/2011	1.2.9	C	Line feed delimiter management
VO2.1.4	24/02/2012	1.2.8	C	Add in the second paragraph : "…..only in exchange for the bank to customer….."

[1] E : Error ; M : Modification ; C : Clarification ; S : suppression ; A : Addition/Extension

Summary - Implementation Guide

1. INTRODUCTION

1.1. Purpose and scope of the guide

This implementation guide is intended for developers of EBICS customer workstations and bank servers. This is not a user manual for customers. Its main purpose is to specify the methods of implementation and the settings.

It is the responsibility of software suppliers to provide a user manual. This user manual will have to contain the list of EBICS return codes as well as the return codes specific to the application with an explicit narrative for each.
This implementation guide is a supplement to the EBICS documentation:

- The general specifications - Release 2.4.2 (in German or translated into English) and annexes OrderTypes et Return codes
- The Implementation Guide - Release 1.7
  in German http://www.ebics.de/index.php?id=93
  or translated into English http://www.ebics.org/index.php?id=93

Reading this documentation is a prerequisite for a proper understanding of this guide.

This guide is based on EBICS release 2.4.2 common to France and Germany - codified H003 in the messages - which has been implemented in both countries from autumn 2009.

The target is to use identically the EBICS protocol in France and Germany. However, considering the existing solutions in the two countries, the mode of implementation of this release will be slightly different from one country to another.
In addition, the use of some optional features being in the competition area, it is the user responsibility to check the service level offered by its banking institutions and to parameterize the software accordingly.

The scope of this CFONB guide is limited to file transfers between customer workstations located in France and bank servers located in France. The choice to extend it to cross-border communications is the responsibility of software vendors or banks[2].

This guide is specifically designed to describe the set-up of customer workstations but it also gives recommendations for configuring bank servers.

The implementation should respect the existing recommendations and regulations, including those relating to banking and/or financial services on Internet (SBFI)
(refer to website: www.ssi.gouv.fr/site_documents/pp/ppcr0401.pdf).

This guide describes the procedures for EBICS Protocol implementation:

- as a successor of ETEBAC3 with a confirmation, by an other channel than EBICS, of the execution order. Later in the document, this terms will be described as EBICS profile T (transport)
- or as a successor of ETEBAC5 with the customer electronic signature of the execution order attached to the order. Later in the document, this mode will be described as EBICS profile TS (transport and personal signature).

Note: The EBICS protocol allows a disjointed signature of the orders via EBICS, this function is not addressed in this version of the implementation guide, but will be included in a later "DS : Distributed Signature " profile.

[2] The term "bank" used in this document should be understood as a Payment Service Provider (PSP) under 2007/064/CE Payment Services Directive dated November 17th 2007. The French transposition is the « Ordonnance 2009-866 » dated July 15th 2009.

1.2. Common implementation rules

1.2.1. Interoperability between customer workstations and bank servers

Each customer workstation must be able to connect to various bank servers which are in conformity with the recommendations of this guide.
It should be possible to add or remove a bank connection to an already installed workstation.

1.2.2. The ways to implement EBICS in France

They can be of two kinds:

- either related to French interbank recommendations
- or related to the service offered by one or more banks.

➢ The French interbank recommendations are related mainly to:

- The Order types (see Annex A1)
- The naming of files (FileFormat / Request type - see Annex A2)
- The type of certificates (Certificates - see Annex A4) depending on the profile (T or TS)

To limit the list of order types, there are not as many order types as file types (unlike in the German implementation) but the file type is a value of one set-up of the order (FileFormat / request type).

➢ Specific Bank services may include:

- Others controls to be carried out by the protocol on the banking data (e.g. monitoring of the amount or account)
- specific levels of security,
- proprietary file types
- proprietary types of information,
- the bank choice to make specific settings mandatory or optional
- the period of availability of the files on the server
- the procedures for retrieving those files by the customer.

The specific rules related to specific bank services are not covered by this guide.
In order to configure the installation, each bank must provide in advance to his customer the information needed to describe its subscription.

1.2.3. Securing the file transfers

The EBICS Protocol provides security for the file transfers at two levels:

- ➢ on the network with the https protocol based on the bank server certificate
- ➢ At the level of the application protocol by using certificates dedicated to authentication, confidentiality, and integrity/signature usages.

At the network level: confidentiality of exchanges on the internet network is established through a https (TLS Transport Layer Security layer) connection. This is based on using a server certificate at the Bank Information System (IS). This certificate previously checked and accepted by the client before any Exchange Protocol will allow a first level of Bank Server authentication by the Customer workstation. After establishing the https channel, all data (EBICS orders, client data and signatures) will be carried in this TLS secure network.

At the application level, the EBICS Protocol provides:

1. mutual authentication of the Customer workstation and server using client and server certificates dedicated to this authentication service.
2. Confidentiality, in addition to that provided by https, by data encryption using client and server certificates dedicated to this encryption service
3. EBICS Profile T: for the integrity of data transmitted by the client to the Bank server, the data are sealed by electronic signature by using a certificate dedicated for this feature. The execution order is transmitted by another communications channel than EBICS.

4. EBICS Profile TS: for integrity and non-repudiation of the data transmitted by the client to the Bank server, data are signed electronically using a certificate for this feature. The electronically signed execution order, attached with the data, has execution order value.

1.2.4. Implementation of a contract

It is recommended in a contract to always define a "UserId" (physical person or service) for Transport. This transport "UserId", has in principle, all transport rights for all the files of the contract, including all reporting files. This transport "UserId", as its name implies, has, of course, no signature rights on the orders transmitted.

All the users (UserId) allowed to personal signature (Signatory) must be named in the contract. For each FileFormat are listed the rights of each signatory on this FileFormat (simple signature, dual signature, dual optional signature, possibly maximum amount by order or by file).

In EBICS TS, each signatory shall have a personal signature certificate on physical device whose structure is consistent with that shown in Annex A3. This personal signature certificate is mandatory issued by a CA that is recognized by the Bank.

Data associated with the signatory are:
- his first and last name,
- the name of the CA used by signatory
- a data in the certificate ensuring the uniqueness of the certificate within a CA. Depending on the s CA this data can be : DN, SAN,.., or a serial number).

In EBICS T, these signatories are attached to an EBICS user class T.
In EBICS TS, these signatories are attached to an EBICS user class E.

Reminder:
- each (transport or signatory) user (UserId) must hold 3 certificates (signature/ encryption/ authentication)
- for a signatory user (UserId), only the signature certificate must necessarily be issued by a CA.
- in the subscription initialization phase (INI and HIA orders), 2 cases arise to validate the user (UserId) :
  . If the certificate is an auto generated certificate: signature control of the mail sent to the Bank containing the hash of the certificate and signed by an authorized person in the company
  . If the certificate is issued by a CA: matching between the data mentioned in the contract and those issued from the certification chain

1.2.5. Integrity and signature of the file transfers

The ES Quantity parameter[3] is set up on the server based on the contract and on the FileFormat.
Following the procedures defined between the Bank and its client, contractual client must be set:

- **EBICS T**

In transport mode, EBICS T profile, client confirms its orders by another communication channel. In this case, the customer workstation is set up with "user class signature" = T (transport), ES quantity = 0.

Separate Signature (without DS/VEU mode) : EBICS profile T				
Personal signatures	**0**	**1**	**2 et +**	
Awaited ES = 0	**OK**	**REJ**	**REJ**	**OrderAttributes = DZHNN, all signatures are taken as transport signatures (regardless of class A, B, E ou T assigned to the user).**

REJ = Reject

[3] This parameter gives the number of personal signature (see EBICS specifications 2.4.2)

- **EBICS TS**

With EBICS TS (signature attached to the execution order), the user signs electronically the order file and sends jointly the order file and the execution orders via EBICS.

In this case, the customer workstation is set up with "user class signature" = E (electronic signature). The signature of this user has power to authorize the orders (as written in the contract between the user and the bank) through his signature made by a personal certificate issued to him by a CA and recognized by the Bank.

Depending on the number of Expected Signature (related to the FileFormat and described in the contract concluded between the client and the Bank), the signatory has the signature power alone or combined with a second signature. The indicator ES quantity = 1 or 2 sets the minimum number of signatories expected.

For ES quantity = 1, orders received with 2 signatures are accepted. This allows, for example, handling the case of a number of signatures depending on the amount of the order transmitted. In the French implementation, the final validation is not necessarily processed directly by the EBICS Protocol. It is therefore necessary for the EBICS server to forward all these orders to the Bank "business" application where the verification of the number of signatures expected is controlled.

Note:

- The contract describes for each FileFormat, the number of required signatures:
  - simple signature (ES quantity = 1),
  - one or two (ES quantity = 1) for example according to the amount,
  - two mandatory (ES quantity = 2),

- A rejection of the order, by the lack or excess of signature, may therefore occur downstream of the EBICS server after acceptance at the EBICS protocol level.
- To prevent rejects, the client workstation must control the number of expected signatures according to the contract prior to sending the file.
- In all cases, more than 2 signatures are not accepted.
- The EBICS protocol allows the use of 2 other signature classes A and B, but this use is not adopted, so far, in France.
- Any reception without signature or with more than 2 signatures is rejected by the EBICS server.

Joint Signature (without DS/VEU mode) : EBICS profile TS					
Personal Signature	**0**	**1**	**2**	**>2**	
Expected ES = 1	**REJ**	**OK**	**OK**	**REJ**	**One or two “E” class signatures are expected**
Expected ES = 1	**REJ**	**REJ**	**REJ**	**REJ**	**Any other combination with A, B ou T causes a reject**
Expected ES = 2	**REJ**	**REJ**	**OK**	**REJ**	**Double-signature required : 2 “E” class signatures are expected**
Expected ES = 2	**REJ**	**REJ**	**REJ**	**REJ**	**Double-signature required : All other combination with A, B ou T causes a reject**

- **Example with 1 personal signature expected**

In this case, one signature is required for national Direct debits FileFormat ddd.
Contractually, the definition for the file FileFormat ddd is ES quantity = 1.
However, the file is not rejected by EBICS protocol if it contains 2 signatures, but will be rejected by the back end.

Personal Signature	0	1	2	>2	Comment
Expected ES = 1	REJ	OK	OK	REJ	One « E » class signature is expected
Expected ES = 1	REJ	REJ	REJ	REJ	Any other combination with A, B ou T causes a reject

Reject if transport type signature, but not if double-signature.

- **Example with 1 or 2 personal signatures expected**

In this case, 1 signature is required for small amount credit transfers and 2 signatures for big amounts.
Contractually, the association for the file FileFormat sct is ES quantity = 1.
All files received with 1 or 2 signatures are accepted by the EBICS server. Big amounts credit transfers are rejected by the business application if they have not 2 signatures.

Personal Signature	0	1	2	>2	Comment
Expected ES = 1	REJ	OK	OK	REJ	Indicator "optional double-signature " : one « E » or two «E+E » class signatures are expected
Expected ES = 1	REJ	REJ	REJ	REJ	Indicator "optional double-signature " : Any other combination with A, B or T causes a reject

- **Example with 2 personal signatures 'double-signature mandatory'**

In this case, two signatures are required for the treasury credit transfers (Fileformat: ict).
Contractually, the association for the file FileFormat ict is ES quantity = 2.

Personal Signature	0	1	2	>2	Comment
Expected ES = 2	REJ	REJ	OK	REJ	'Double-signature mandatory': two « E » class signatures are expected.
Expected ES = 2	REJ	REJ	REJ	REJ	'Double-signature mandatory': Any other combination with A, B or T causes a reject

1.2.6. Specific characters

The authorized characters in the protocolar exchanges are defined by the XSD schema.
The specific characters used in France or Germany (e.g. é, è, à, ç, œ, ü, ß, Ä) are excluded from the protocol itself by the schema XSD.
In the files, the characters used are those defined in the standards.

1.2.7. ASCII / EBCDIC Coding :

The exchange of files in ASCII coded characters is recommended. If the bank proposes to send EBCDIC files, the indication "EBCDIC" must appear in the tags' FULOrderParams "on the following model:

```
<FULOrderParams>
        <Parameter>
                <Name>EBCDIC</Name>
                <Value>TRUE</Value>
        </Parameter>
        <FileFormat CountryCode="FR">pain.xxx.cfonb160.dct</FileFormat>
</FULOrderParams>
```

1.2.8. Parser :

Fixed format files can contain multiple logical iterations (order files,..) provided that they are with an identical FileFormat. In this case, they can be placed one after the other and merged in a single physical file.

The variable format (XML) files follow the same mechanism only in exchange for the bank to customer: a physical file can contain multiple logical messages merged without any tag. Accordingly, before any processing, it is necessary to disassemble these messages to integrate them in a parser.

1.2.9. Line feed delimiter management (CR, LF, CRLF, or no) :

For customer to bank exchanges, the bank server software must support files containing any line feed delimiter including no delimiter.

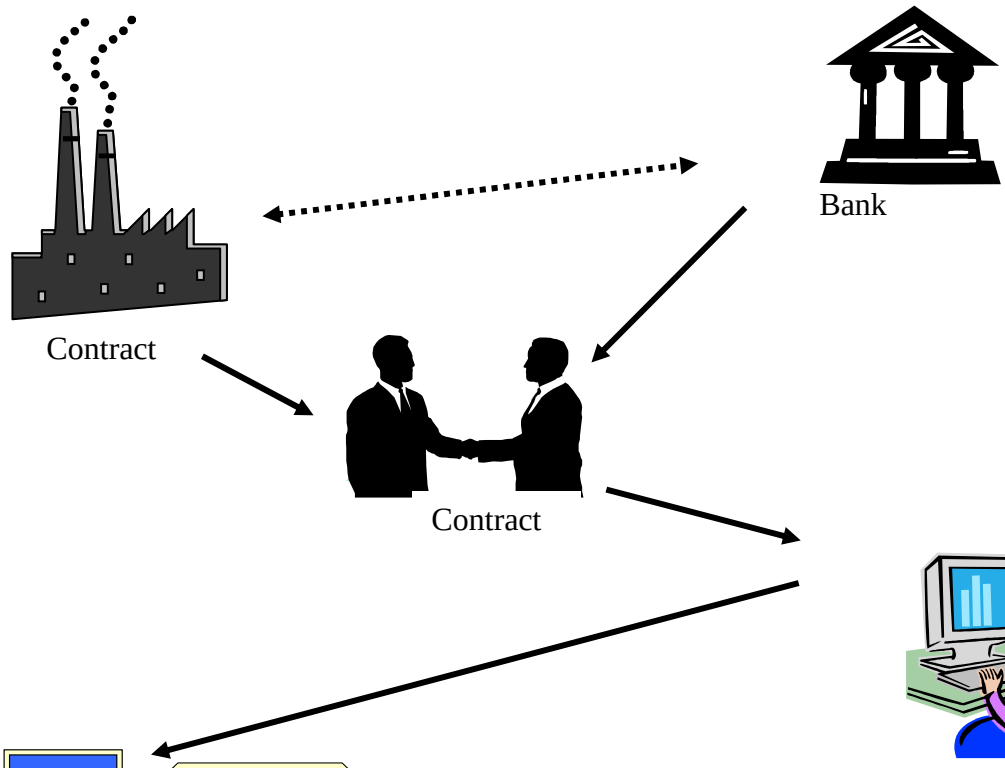
For bank to customer exchanges, the workstation software must also process files containing any line feed delimiter including no delimiter.

2. IMPLEMENTATION

2.1 Set-up

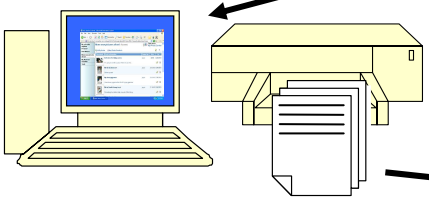
2.1.1 General schemata of the security settings

2.1.1.1 EBICS T example : execution order on a separate channel and usage of self-signed certificates



Capture of the customer contract. Production of the Customers Identifiers : HostId, UserId, PartnerId.

Capture and recording of the Customer keys on the bank server /host



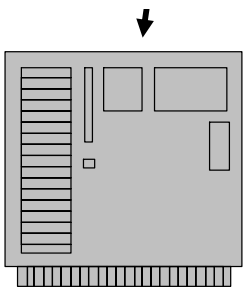
Recording of the Customer Identifiers and bank URL
Production of customer keys, printing and sending another channel (mail,…)

EBICS Initialisation
Sending of the 3 certificates : Authentification, Encryption and Signature with by INI/HIA commands

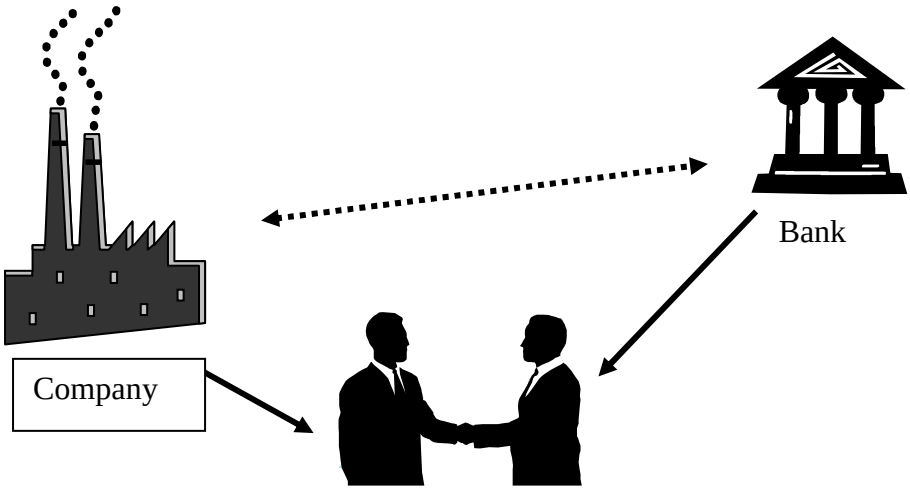
Downloading of the bank Certificates with by HPB command

Recording of the Bank certificates in the database of the customer workstation.

Tests,
First customer remittance of orders



2.1.1.2 EBICS TS example : joint execution order and usage CA certificates (Authentification, Encryption and Signature)

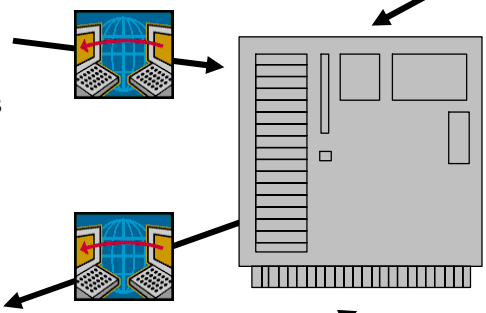


Capture of the customer contract.
Production of the Customers Identifiers: HostId, UserId, PartnerId.

Recording of the Customer Identifiers and bank URL
Loading of customer certificates

Capture of the Customer certificate parameters on the bank server /host (AC, DN, …)

EBICS Initialisation.
Sending of the 3 certificates: Authentification, Encryption and Signature with by INI/HIA commands



EBICS TS can also be implemented with self generated certificates for authentification and encryption sent by another channel (mail,..) and a signature certificate issued by a CA.

Certificate Vérification in relation with the CA DN, ValiditY, Usage, …
Matching with the contract

Downloading of the bank Certificates with by HPB command

Recording of the Bank certificates in the database of the customer workstation.

Tests,

2.1.2 Security set-up

2.1.2.1 Certificates

File transfers with the French banks servers (specified by the CFONB) are based on the use of X509 certificates to guarantee the integrity in phase 1 and the execution order in phase 2.
The details of these certificates (templates) are described in Appendix A3. The separation of uses (Authentication, Encryption and Signature : sealing in EBICS T, electronic signature in EBICS TS) is mandatory, so 3 keys will be used (one for each use) by the customer workstation (and the bank server).

Both physical persons do not have common certificates for authentication, encryption and digital signature (each element of this triplet must be different).
It is recalled that certificates for authentication, encryption and digital signature must be different from each other for single userID.

Nota bene 1: in two different contracts, physical person can have two different userID using the same triplet.

Nota bene 2: to the reconciliation, it is underlined that the customer workstation must be able to print all initialization letters (for each certificate of each userID).

2.1.2.2 Characteristics of eligible Certificates

When certificates are issued by a CA recognized the bank, the algorithm used by the certification authority to sign these certificates is:
- either RSA SHA-1 algorithm, and this for a temporary and non-renewable maximum period of 3 years (and replaced by the RSA SHA2 256 target),
- either directly with an algorithm RSA SHA2 256 (all certificates issued by a CA 3-year target).

In all cases, the EBICS messages are signed with a SHA2 algorithm. The SHA1 algorithm is allowed only for the CA's signature.

Certificates may be placed on different media:

- **On the customer workstation:**
  - authentication certificate:
    - on hardware or software media
    - self-signed or generated by a CA that is recognized by the Bank.
  - Encryption certificate:
    - on hardware or software media
    - self-signed or generated by a CA that is recognized by the Bank.

    Self-signed certificates must be renewed after a period of 5 years. Except change due to a security alert, they can be renewed at the end of this period either again by self-signed certificates or CA certificates.
    CA certificates will be renewed following the indications of the CA's certification policy and SEND to the server with EBICS orders: "PUB" and "HCS" (described in Chapter 10 specification EBICS 2.4.2).)

  - Signature certificate:
    - EBICS T: for the sealing of the messages, self-signed or generated by a CA that is recognized by the Bank, and on hardware or software media.
    - EBICS TS: mandatorily generated by a CA recognized by the Bank and on hardware for personal signature.

    The media type will be deducted from the value of the ID (OID) CA certificate policy.
    The signing certificate will be either exclusively used for signature, or be a multi-purpose certificate only used for signing. In all cases, it will be different from authentication and encryption certificates.

- **On the bank server:**
  Two certificates self-signed or generated by a CA with a RSA SHA algorithm 256 and 2048-bit key length. Self-signed certificates will be valid for 5 years at most

2.1.2.3 Initialization of customer certificates and sending of public keys to the bank server

The bank server needs three public keys from the customer workstation, one per usage (authentication, encryption and sealing).
Each public key is stored in a certificate on hardware or software media dependinbg on their eligibility characteristics (see § 2.1.2.2). Each of the 3 certificates for the three types of usages is transmitted to the server in an initialization file using the EBICS protocol, according to the XSD schema of EBICS V2.4.

The reconciliation process will differ depending on the case. Two cases are possible:
- using a self-signed certificate,
- using a certificate issued by a CA.

➢ **Using a self-signed certificate**

When using a self-signed certificate, this validation is not possible by the certification chain. The authentication must be ensured by a second mechanism different from the initialization file generated by the customer workstation.

This is done by sending to the bank, in parallel with the certificate via EBICS, a confirmation by another channel (printing and sending a pdf by mail, fax, download). The way to send this confirmation is out of the scope of this guide but must be stated in the customer bank contract.

In France, sending 3 documents, one per certificate, is mandatory.
This authentication can also be made by manual signature on the printed version of the initialization file (see Annex A4 - Print the certificate).
This document includes the certificate, in DER format, together with information identifying the user (user ID, partner ID, and possibly UserName) and the seal (hash) of the certificate in a printable format necessary for the reconciliation.
In the bank, the validation is done by matching these data.

Optionally, if proposed by the bank, this certificate confirmation file can be transmitted by another secured electronic channel (different from EBICS), as it is designed to be integrated automatically on the bank server

As the safety of a simple mail (not secured) is not guaranteed, sending this file by mail (not secured) is not recommended.

➢ **Using a certificate issued by a CA**

When using a certificate issued by a Certification Authority (CA), the control of the certification chain of the certificate is possible through a fully automated reconciliation following the internal rules of each bank.

In EBICS T, the certificate is processed as a self-generated certificate.
In EBICS TS, if the signature certificate has been issued by a CA non reconised by the bank, then the certificate is rejected in the initialization phase

2.1.2.4 Retrieving the public key certificates of the bank by the customer

The recommended targeted template of the certificates will be 2048 bits (for the RSA key length) and RSA2048-SHA2 (for the signature algorithm). The EBICS launch in France is done with these characteristics of the certificate.

Currently some CA's are not yet compatible with this target. It therefore is autorised to use temporarily for server certificates either self-signed certificates or private CA certificates (in RSA 2048 bits and RSA2048-SHA2 for the signature algorithm) and later to migrate towards certificates issued by CA (bank and non-bank) providing these characteristics.

The customer workstation will have to download the public keys from the bank server, using the HPB command (download public bank key) and then check them (see EBICS V2.4.2 specifications Chapter 4.4.2).
It is not necessary to automate the "HPB" command in every exchange. However, we strongly advise to send it upon receipt of an anomaly tending to the renewal of the bank certificate.

In "HPB" command, the bank must send the X509 certificates with the public keys.

As for the public keys of the customer workstation, this check may be made by sending (in parallel of the EBICS exchange) a confirmation document by another channel (this document may be the Print of the certificate - see Appendix A4).

The customer workstation should be able to provide a simple procedure to match the public keys transmitted using EBICS and the ones received by another channel.

2.1.2.5 Certificate revocation

Certificates issued by a CA may be revoked. Search for CRL (Certificate Revocation List) or control the status of the certificate in the OCSP (Online Certificate Status Protocol) must be done for each CA accepted by the Bank, so only if the certification chain is registered, and under the conditions of the CRL publication by the CA.
If the certificate is revoked, regardless of the SPR command user, using the server updates the user status to 8-Suspended by user (SPR) that no longer allows any file transfer. All file transfers initiated by the user will have to code error 091004 EBICS_INVALID_USER_STATE

2.1.2.6 Errors messages related to Certificates

Any error about certificate management must be announced by a message.

Disclaimer:
The list of error messages related to Certificates is available on the CFONB website :
www.cfonb.org
in the Documentation section: ETEBAC Migration to SWIFTNet and EBICS

2.1.3 Electronic signature and encryption

2.1.3.1 Main principles

The EBICS V2.4.1 version supports the electronic signature coded A004, A005 or A006.
But the A004 version of the signature will not be used in France because it is not compatible with the use of certificates, so only versions A005 and A006 will be used in France.
Using signature A006 is not recommended for problems of current availability of hardware .
The A005 version of the signature is recommended in France.

The AES encryption algorithm is used.

2.1.3.2 Remarks on signature computing

- System characters such as CR, LF and Ctrl-Z have not to be included in the hash computing in the versions A005 and A006.
- With A006, there is a double computing of the hash[4] : signature is not computed on the message itself [Si(Hash(M))]; but on the hash of the message [Si(Hash(Hash(M))].
- The signature is calculated on the SignedInfo tag after C14N cononicalization then the cononicalization is hashed.
- Hashing is calculated on the certificate previously encoded by DER - without characters (CR, LF Ctrl-Z)
- The 2 following paddings can be used:
  . ANSIX923 : The ANSIX923 padding string consists of a sequence of bytes filled with zeros before the length.
  . ISO10126 : The ISO10126 padding string consists of random data before the length.
  The detection of the padding method of is done by finding the type of padding: zeroes or random data.
- PKCS1 V1.5 should be used to encrypt the encryption key
- The cononicalization must add the default values. In an XML document, only the namespaces used in the XML document should be indicated. Other namespaces must not be shown. The link for the canonization is:
  http://www.ebics.org/index.php?id=38
- The XML / DSIG distinguishes between XML documents with or without comments. In EBICS: The Algorithm for cononicalization is defined via http://www.w3.org/2006/12/xml-c14n11
  This is the algorithm, where the XML-comments are erased.
  The identifier for the algorithm Which doen not erase the comments would be
  http://www.w3.org/2006/12/xml-c14n11 # WithComments

An example of hash computation is described in annex 6.

[4] For historical reasons

2.1.4 Initialization of IDs

Each bank must inform its customers about the following information necessary to set-up the customer workstation.

- The HostID (bank identification): it is recommended that banks use a BIC with 11 characters (possibly BIC 8 supplemented with XXX).

- The UserID, (user or department name): The syntax is free, in accordance with the specified format [a-zA-Z0-9 =] (1.35)

- The PartnerID (contract / subscription Number): The syntax is free, in accordance with the specified format [a-zA-Z0-9 =] (1.35))

The last two identifiers will be provided by the bank when signing the contract.
It is recommended not to fill in the SystemID item.
There is always at least one user UserID per subscription. The certificates are attached to the user.

Note:
- Partner ID = Subscription for a Company
- User ID = User of this subscription.

In EBICS T profile, in most cases, a customer (Company) will have only one user.
However, some companies may be more complex and may choice to have multiple users per subscription, for example, a user ID for the Accounting department, and another user ID for another department (eg Sourcing).

A given FileFormats/RequestType will have the same number of expected signatures for the different UserID of this PartnerID.

2.1.5 Management of several users in a single subscription

This part describes the recommendation for the management of multiple users (UserId) at a same Subscriber (PartnerId).
The MessageID must be unique. This identification is performed by the server by joining the transaction number (OrderId) with subscriber (PartnerId) and the type of order (OrderType).
It is therefore essential to manage transaction numbers (OrderId) regardless of the user (UserId).
When initializing the users (userID) of a subscriber (PartnerID) it is recommended for good management of the transaction numbers to assign slices of transaction numbers (OrderId) by users (UserId). For example, the slice A000 to AZZZ for the first user, B000 to BZZZ for the second user, …(with a maximum of 26_users by subscription) etc.

Reminder: OrderIDtype = {1} [A - Z] [A-Z0-9] {3}

This precaution could avoid the risk of duplicate OrderId and its rejection by the EBICS Bank server.

2.1.6 Service Providers

Several service providers generate files (e.g. payroll...) for multiple customers and send them to the banks of their customers.

In EBICS T profile, the customer signs a contract with the bank, but he may delegate the file transfer, in a transparent or not manner, to his provider.
Thus, a service provider who has several customers receives a PartnerID / UserID from each of its customers. It will have to initialize with each bank for each customer.
EBICS T profile/ Service provider shall bear the management of authentication, encryption and signing certificates.

EBICS TS profile: the signature certificate is under the responsibility of the customer and therefore under his exclusive permanent control. The service provider must provide a secure environment to the customer which has to sign himself each file before their delivery to the Bank.

2.1.7 Tests

A test of the file transfer is necessary before any transfer of real files. Therefore, each server must be able to receive test and production files.

Contractually, the customer will have to be able to test with a workstation software able to do so. In addition, it should be possible to be in test mode with a bank and in the operational mode with another one.

The following recommendation specifies the conditions of these tests.

The customer workstation must have a set-up which allow switching from test mode to production mode. Its use can only be made at the initiative of the customer in agreement with its bank.

In order to avoid potential confusion between operational and testing flows, it is not desirable that this distinction results from a manual intervention at the bank server level.

In France, it is recommended to use an additional set-up to FUL or FDL orders to distinguish the test files from the production files.
The presence of the set-up called "TEST" and its value “True” means that it’s a test file. The absence of the set-up means a production file.

The « TEST » indication must be included in the tag «OrderParam » in the following way:

```
<FULOrderParams>
  <Parameter>
    <Name>TEST</Name>
    <Value>TRUE</Value>
  </Parameter>
  <FileFormat
CountryCode="FR">pain.xxx.cfonb160.dct</FileFormat>
</FULOrderParams>
```

2.2 File transfers

Any EBICS transaction is made at least of two messages; one for initialization and another for the data transfer (upload or download mode).

Two OrderTypes will be used for file transfers: FUL and FDL. Those commands are used:

- FUL (Upload) for sending a remittance file by the customer to the bank
- FDL (Download) for the retrieval by the customer of a file generated by the bank (customer reporting,..)

2.2.1 Settings related to file transfers

In EBICS T profile, only the value “T (transport)" is recommended in the bank server repository for the class of signature. In this case the check of the signature of transport is required by the server (see below).
The customer workstation must specify in the message using the value “D” in the first field of “OrderAttribute”, that the file transfer is using Transport as signature class.

In EBICS TS profile: this field must contain the value “O”

Reminder: For other commands, the value of the first field of OrderAttribute will conform to EBICS 2 .4.2 specifications (see table on page 275 of the detailed specifications).

2.2.2 Processing of the order remittances

Using the FUL command, each signature class is in relation with a FIleFormat.

2.2.2.1 List of pre-validation controls

In Upload, in the first message (the initialization one), various controls so-called pre-validation are planned:
• Certificates control (3 public keys) based on the information sent by the customer workstation during the initialization stage,
• Amounts control (limits),
• Accounts checking

Among these, the certificates control is highly recommended for bank servers in France.

Checking the amounts and accounts is optional for EBICS bank servers in France and each bank can decide to implement them or not. If the bank server does not manage these controls, it replies by a negative response to the customer workstation.

2.2.2.2 Security checks of received remittance files

The first message, the initialization one, contains the hash of the remittance file calculated by the customer workstation as well as the file with the digital signature (s). At this stage, the remittance file is not yet uploaded, so the hash cannot be calculated on the bank server.

Thus this synchronous control does not provide all guarantee for non-repudiation of the remittance file since it is performed based on the hash sent by the customer workstation.

The EBICS protocol guarantees the integrity of each EBICS command by the electronic identification / authentication signature of the user issuing the message.

The signature is used at two levels:
• All remittance files are signed with an electronic signature. The result of this signature is stored in a field in the Initialization message transmitted with the calculated hash of the file on the customer workstation. The class of the signature is "execution order = personal signature" or "transport" depending on the user profile set for the user in the bank server.

• Each EBICS message is authenticated with the authentication key of the user sending the message. The signature mechanism uses a XMLDsig signature. The complete list of fields of the message included in the calculation of the signature can be found in the specifications. It may be recalled that the signed data are generally sensitive, such as the hash and the file containing the signatures.

The signature control of the remittance files is based on the matching between the hash sent by the customer workstation and that calculated by the bank. This one is done on the file received by packets of 1MB, the calculation could begin, in synchronous mode, before the end of the reception of the last packet, or in asynchronous mode after completing the communication with the customer workstation. Each server can choose its mode of operation: asynchronous or synchronous. The choice is left open for implementation.

In synchronous mode, it is possible to decompress, decrypt and check the signature of the remittance file before sending the response frame to the customer workstation, i.e. before the end of the EBICS transaction. Indeed in the last frame sent by the EBICS customer workstation, the last segment contains the data identified by the tag <SegmentNumber lastSegment= “true”>. The server has the opportunity to verify the signature and return, if any, an error (EBICS_SIGNATURE_VERIFICATION_FAILED) in the response frame to the receipt of the last segment.

In asynchronous mode, when the EBICS transaction is completed, the received remittance file has not yet been decompressed, decrypted and its signature integrity has not been checked. However, any errors detected on the remittance file, must be made available at all times for EBICS customer workstation by PSR or PTK, as described in the following paragraph.

2.2.2.3 Communication to the customer about the receipt of order remittance file

When using the asynchronous mode, the customer workstation must be informed by the bank server of the correct or incorrect reception of the file and especially controls and rejects described in the preceding paragraph and which are primarily on the business file signature verification.

The rejection/negative acknowledgments must mandatorily be transmitted to the customer workstation. Sending positive acknowledgments is optional and depends on the bank’s service offering. The customer workstation must be able to retrieve both types of acknowledgments.

In Germany, this functionality is currently performed using Kunden Protokoll (OrderType PTK) described in Chapter 10 of EBICS V2.4.2 Specifications.

For the implementation in France the target is the use of the PSR (Payment Status Report) retrieved using the FDL (Download) command or any other channel. However, the PTK can be used temporarily knowing that it has to be implemented on identifiers with a maximum length of 8 characters instead of the 35 characters of PSR identifiers.

Using the Payment Status Report:

- The Payment Status Report (ISO 20022 XML format - pain.002.001.xx available at ISO www.iso20022.org) must be generated at file level. Therefore, items of the level 3.0 and beyond will not be filled in, only GroupHeader level items or OriginalGroupInformationAndStatus will be filled in (see Appendix A5 - Format Payment Status Report).
- Otherwise, sending a Payment Status Report may be done by another channel (mail, Internet, web portal) by using a style sheet to make it readable.
- In Appendix A5 - Format of the Payment Status Report, are described the data needed to match a PSR with the corresponding remittance order.
- Generation of a Payment Status Report is mandatory for any failed transaction. It’s not necessary to create a PSR for each remittance file in failure, but in the case of multiple remittance files, the server may, at the end of the transmission create a technical concatenation of several XML files (presence of several headers in the same XML file) in a

single PSR. In this case, the PSR can not be deparsed directly by the customer workstation. It should be previously cut in unitary XML files before parsing each one.

- PSR only covers the FUL command (UPLOAD) and should not be generated for other commands (INI, HIA ,..).

2.2.3 File retrieving by the customer workstation: the Download command (FDL)

The Download order (FDL) without setting a date is used to retrieve all the files available (not yet retrieved): the bank server provides all the files in storage. Those of the same type are concatenated before being zipped and encrypted for the transfer, thus forming a single file.

After sending, the files are automatically archived on the bank server. Then, they can be retrieved again using the Download order (FDL) with a set-up including either a date or a range of dates depending on the services provided by the bank.
The range of possible dates will depend on the archiving duration of the bank server. Each bank may set a different archiving duration for each file type; this information should be included in the subscription contract.

Making files available beyond their online period (archiving duration) is not in the protocol field.

Note:

- Some banks may offer specific reporting services (various reporting frequencies,...). The method described for naming the files can cover the needs for these services. Banks may use proprietary extensions in the file name used in the FileFormat. This type of setting specific to an institution should be indicated in the annexes of the subscription contract.
- The FDL command has only one FILE FORMAT parameter. So only files with the same type could be concatenated. EBICS customer workstation will have to handle this case.
- The FDL command without date range allows the customer to get all available reports not yet retrieved.

3. USE OF THE CUSTOMER WORKSTATION

Before the implementation of the connection between a customer workstation and a bank server a contract must be signed between the customer and the bank. The contract must include a document indicating the specific data for this subscription.
So there will be a contract and a set-up for each bank.

It is the responsibility of software vendors to make available to customers the features needed to secure the access to the customer workstation and the sensitive data.
The customer must have the tools to secure the Internet connection (firewall, anti-virus, ..) and keep them up-to-date.
When the customer workstation stores certificates in a database, it will have to provide a level of security (data encryption) adequate and equivalent to that of the Windows storage or Java Key Store.
Each customer workstation and each bank server keep a record of all transactions (Log with : time stamp, amount ...). It must be possible to retrieve the end-to-end log of each transaction and to transmit it to the bank (mail,..).
It is recommended that the user interface and the prints are in French.
In order to provide any evidence in the case of a conflict, the client must keep the signed files such as delivered to the Bank, as well as all other necessary data.

4. ANNEXES

A1 : Order Type

EBICS Specifications 2.4.1 apply in France and Germany but the controls (OrderType) coupled to a specific file format will not be used in France.
For this feature, in France we use the OrderType FUL associated with specific FileFormat.

The protocol commands are classified into two categories:
• The “system orders” related to the management of the EBICS standard / protocol itself
• The “bank-technical orders” related to a format

The list of commands supported by the French servers is a subset of commands and EBICS are considered as system orders, except orders "FUL" and "LDF", which are the only commands and upload download file allowed in France.

The list below is the complete list of OrderType mandatory in France. The other Ordertypes are optional.
The implementation of the HTD Ordertype (Download subscriber’s customer and subscriber Data) is recommended.

A server receiving a command that it can’t process must send the return code:
EBICSS_UNSUPPORTED_ORDER_TYPE

Identification	Name	Format
HCA	Send amendment of the subscriber key for identification and authentication and encryption	
HCS	Transmission of the subscriber key for ES, identification and authentication and encryption	
HIA	Transmission of the subscriber key for identification and authentication and encryption within the framework of subscriber initialisation	
HPB	Transfer the public bank key (download)	
HPD	Download bank parameters	
INI	Send password initialisation	Customer’s public key for the ES
HEV	Download supported EBICSS versions	
PUB	Send public key for signature verification	Customer’s public key for the ES (see Appendix Chapter 15)
SPR	Suspension of access authorisation	Transmission of an ES file with a signature for a dummy file that only contains a space
FUL	File Upload	Upload files whose type is in parameter
FDL	File Download	Download files whose type is in parameter
PTK	Kunden Protokoll	

A2 : FileFormat / Request Type - Files names

Disclaimer:
This annex named : **Annexe 2 EBICS SWIFNet - Nommage Fichiers**
is available on the CFONB website : www.cfonb.org
in the Documentation section: **Migration ETEBAC vers EBICS et SWIFTNet**

A3 : Certificates

A3.1 Errors messages related to Certificates

Disclaimer:
This annex named : **Annexe 3 : Certificates error codes**
is available on the CFONB website : www.cfonb.org
in the Documentation section: **Migration ETEBAC vers EBICS et SWIFTNet**

A3.2 Structure of the certificates for EBICS customer workstations

The certificate for EBICS customers can be self-signed or imported on the customer workstation if it is issued by a private Certificate Authority CA.
Three usages are defined for EBICS and three certificates are required.

Self-Signed Certificate Usage :

(Self-Signed) Certificate for signature (EBICS T only)		
Field X509	**Value** **N/A=Not Applicable**	**Mandatory** **Y=Yes N=No**
version	=2 (for X509V3)	Y
serial Number	Random Number of maximum 20 Bytes if self-signed	Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer	=subject	Y
validity	Validity : 5 years[1]	Y
subject (object or DN)	The attribute is « commonname »	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length – rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=SubjectKeyIdentifier of the CA or of the current certificate	Y
SubjectKeyIdentifier		Y
KeyUsage	NonRepudiation	Y
ExtendedKeyUsage		N
CRLDistributionPoints	N/A	N

(Self-Signed) Certificate for Authentication (EBICS T or TS)		
Field X509	**Value**	**Mandatory Y=Yes N=No**
version	=2	Y
serialNumber	Random Number of maximum 35 octets if self-signed	Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer	=subject	Y
validity	Validity : 5 years[1]	Y
subject (object or DN)	The attribute is « commonname	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length – rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=SubjectKeyIdentifier of the CA or of the current certificate	Y
SubjectKeyIdentifier		Y
KeyUsage	DigitalSignature	Y
ExtendedKeyUsage	N/A	N
CRLDistributionPoints	N/A	N

(Self-Signed) Certificate for Encryption (EBICS T or TS)		
Field X509	**Value N/A=Not Applicable**	**Mandatory Y=Yes N=No**
version	=2	Y
serialNumber	Random Number of maximum 35 octets if self-signed	Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer	=subject	Y
validity	Validity : 5 years[1]	Y
subject (object or DN)	The attribute is « commonname	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length - rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=SubjectKeyIdentifier of the CA or of current certificate	Y
SubjectKeyIdentifier		Y
KeyUsage	keyEncipherment or keyAgreement	Y
ExtendedKeyUsage	N/A	N
CRLDistributionPoints	N/A	N

[1] This is valid only for self-signed certificates. The term of validity of CA certificates will depend on the Policy of the CA for this type of certificate

CA certificate use :

Each bank determines the certificates, compliant with the structure described below, that it agrees for the personal signature.

AC Signature Certificate (Mandatory on hardware device for TS profile)

Field X509	Value N/A=Not Applicable	Mandatory Y=Yes N=No
version	=2 (for X509V3)	Y
serialNumber	Single by AC registered with max length 20 Bytes	Y
Signature Algorithm	RSA-SHA2 (256) or SHA1 (160) intermediary phase for 3 years.	Y
issuer	=AC DN	Y
validity	3 years	Y
subject (objet ou DN)	User Id including the « CommonName »	Y
subjectPublicKeyInfo	RSA Key with 2048 bits Key Length-rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=AC SubjectKeyIdentifier	Y
SubjectKeyIdentifier		Y
KeyUsage	NonRepudiation bit or ContentCommitment =1	Y
ExtendedKeyUsage	id-kp-emailProtection	N
Subject Alternative Name	(may include mail address) Be careful to critical character	N but non critical if present
Issuer Alternative Name	Be careful to critical character	N but non critical if present
CRLDistributionPoints	May be completed with AuthorityInformation access if OCSP service.	Y
Freshest CRL	If DeltaCRL is used	Y but non critical with DeltaCRL
Authority Information Access	If OCSP service.	Y but non critical with OCSP
QCStatement	If the qualified certificate contents OID pointing out the certificate is qualified and the private key of the certificate is stored within a SSCD.	Y if Qualified Certificate

AC Authenticate Certificate (On hardware or software device)

Field X509	Value N/A=Not Applicable	Mandatory Y=Yes N=No
version	=2 (for X509V3)	Y
serialNumber	Single for AC Name max length 20 Bytes	Y
Signature Algorithm	RSA-SHA2 (256) or SHA1 (160) intermediary phase for 3 years	Y
issuer	=AC DN	Y
validity	3 years	Y
subject (objet ou DN)	User Id including the « CommonName »	Y
subjectPublicKeyInfo	RSA Key with 2048 bits Key Length-rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=AC SubjectKeyIdentifier	Y
SubjectKeyIdentifier		Y
KeyUsage	DigitalSignature =1	Y
ExtendedKeyUsage	id-kp-clientAuth	N
Subject Alternative Name	(may include mail address) Be careful to critical character	N but non critical if present
Issuer Alternative Name	Be careful to critical character	N but non critical if present
CRLDistributionPoints	May be completed with AuthorityInformation access if OCSP service.	Y
Freshest CRL	If DeltaCRL is used	Y but non critical with DeltaCRL
Authority Information Access	If OCSP service.	Y but non critical with OCSP

AC Authenticate Encipherment (On hardware or software device)

Field X509	Value N/A=Not Applicable	Mandatory Y=Yes N=No
version	=2 (for X509V3)	Y
serialNumber	Single for AC Name max length 20 Bytes	Y
Signature Algorithm	RSA-SHA2 (256) or SHA1 (160) intermediary phase for 3 years	Y
issuer	=AC DN	Y
validity	3 years	Y
subject (objet ou DN)	User Id including the « CommonName »	Y
subjectPublicKeyInfo	RSA Key with 2048 bits Key Length-rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=AC SubjectKeyIdentifier	Y
SubjectKeyIdentifier		Y
KeyUsage	KeyEncipherment =1	Y
ExtendedKeyUsage	id-kp-emailProtection	N
Subject Alternative Name	(may include mail address) Be careful to critical character	N but non critical if present
Issuer Alternative Name	Be careful to critical character	N but non critical if present
CRLDistributionPoints	Possibly may be completed with AuthorityInformation access if OCSP service.	Y
Freshest CRL	If DeltaCRL is used	Y but non critical with DeltaCRL
Authority Information Access	If OCSP service	Y but non critical with OCSP

A3.3 Structure of the certificates for EBICS bank servers

It is necessary to have a certificate for each usage: ie in the current version 2.4.1, 2 certificates per server bank (authentication and encryption).
Both server certificates are treated as SSL TLS certificates and must therefore have both the KeyUsage of DigitalSignature and of KeyEncipherment.
The signature certificate is not provided in EBICS 2.4.1 for ETEBAC 3 and ETEBAC 5 migration.

Server Certificate for Authentication		
field X509	**Value**	**Mandatory Y=Yes N=No**
version	=2 (for X509V3)	Y
serialNumber		Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer		Y
validity	Validity : 5 years[2]	Y
subject (object or DN)	The attribute is « commonname	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length - rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier		Y
KeyUsage	DigitalSignature;KeyEncipherment	Y
CertificatePolicies		Y
CRLDistributionPoints		Y
FreshestCRL		N
ExtendedKeyUsage		N

Serveur Certificate for Encipherment.		
field X509	**Value**	**Mandatory Y=Yes N=No**
version	=2 (for X509V3)	Y
serialNumber		Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer		Y
validity	Validity : 5 years[2]	Y
subject (objet ou DN)	The attribute is « commonname	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length - rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier		Y
KeyUsage	DigitalSignature;keyEncipherment	Y
CertificatePolicies		Y
CRLDistributionPoints		Y
FreshestCRL		N
ExtendedKeyUsage		N

[2] This is valid only for self-signed certificates. The term of validity of CA certificates will depend on the Policy of the CA for this type of certificate

A4. Print Certificate

In France sending of the 3 certificates (one per usage) is mandatory.
Each certificate is printed in PEM format.
The hash printed consists of the certificate built in the standard X509 with the algorithm SHA2 (256).
The hash is printed in hexadecimal and in uppercase.
The following letters are given as examples of presentation. For an example of calculated hash refer to Annex 6.

When the certificate is issued by a certification authority, to facilitate the reconciliation, it is recommended to print the following certificate data on the initialization letter or in the contract for an automatic reconciliation process:

- "certificate issued to: name-surname / ID" (subject - field NC)
- "certificate issued by: name the certification authority " (issuer - field CN)

Initialization Letter for the certificate with signature of transport

Date: TT.MM.JJJJ

Time : HH:MM:SS

Host Id : “BIC11 of the bank”

Bank : « name of the bank »

User-ID : Xxxxxxxx

Partner-ID : Yyyyyyyy

Version: Signature A005

Signature certificate

Type A005

If CA certificate:
Certificate issued to: name-surname or identifier
Certificate issued by: name the certification authority

-----BEGIN CERTIFICATE-----
MIICcjCCAdugAwIBAgIBDzANBgkqhkiG9w0BAQQFADA6MQswCQYDVQQGEwJGUjEY
MBYGA1UECxMPYmFucXVlcG9wdWxhaOdrLaNGZtYDVQQDEwggdHVyYm9zYTAeFw0w
ODA5MTAxMzI0MzZaFw0xODA5MDgxMzI0MzZaMDQxCzAJBgNVBAYTAmZyMRgwFgYD
VQQLEw9iYW5xdWVwb3B1bGFpcmUxCzAJBgNVBAMTAmpmMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCkscideEsfU0+UPqM13kPUQVBFYB4xOcHCYqzr6PPgl7Co
GwsjK5o4CKUm/7qWS0BdnqNOdrLaNGZ4kCXlXDg1SemWMlOgPtWl9T3XAiyyr88L
Ei+9sisIUA/JE/3leQWuk0gJXohtxKUwR/fbsWrQjqLspxNK09Urbqz8hwehPQID
AQABo4GNMIGKMA4GA1UdDwEB/wQEAwIE8DA4BgNVHR8EMTAvMC2gK6AphidodHRw
Oi8vODYuNjQuMTAuMTM4LytDU0NPQ0ErL2FzYV9jYS5jcmwwHwYDVR0jBBgwFoAU
zM7nNDE4VQkAUz33C9ztXhG9P3gwHQYDVR0OBBYEFNd6cAJ8L04eB7TiCzpcumIn
gFSsMA0GCSqGSIb3DQEBBAUAA4GBAEm2OLlVyMIzf7Bk7ZUNBCQacvUEdl2o58Pg
py+CMN+K1OdrLaNGZ77TIVKbydqwI2t7hIpuC81c8D3O9r3LiYSDrgxMFhxeUKLD
sIo1dusXjV8nHm5V2zu4hOdrLaNGZix3bEEEFH+cpzOp5y/ogwHWVpz6h3r36Lgo
Vl1S6JU6
-----END CERTIFICATE-----

Hash of the signature certificate (SHA-256) :

B8 3C B0 19 66 C9 9C 6E
2C A5 BA 6A 2B 56 01 92
35 2A B4 91 53 E9 0B BA
34 C1 5E B5 9F 4A 64 F7

Date : **Signature :**

Initialization Letter for the authentification certificate

Date : TT.MM.JJJJ

Time : HH:MM:SS

Host Id : “BIC11 of the bank”

Bank : « name of the bank »

User-ID : Xxxxxxxx

Partner-ID : Yyyyyyyy

Version : Authentication X002

Authentification Certificate

Type : X002

If CA certificate:
Certificate issued to: name-surname or identifier
Certificate issued by: name the certification authority

-----BEGIN CERTIFICATE-----
MIICcjCCAdugAwIBAgIBDzANBgkqhkiG9w0BAQQFADA6MQswCQYDVQQGEwJGUjEY
zM7nNDE4VQkAUz33C9ztXhG9P3gwHQYDVR0OBBYEFNd6cAJ8L04eB7TiCzpcumIn
MBYGA1UECxMPYmFucXVlcG9wdWxhaOdrLaNGZtYDVQQDEwggdHVyYm9zYTAeFw0w
ODA5MTAxMzI0MzZaFw0xODA5MDgxMzI0MzZaMDQxCzAJBgNVBAYTAmZyMRgwFgYD
VQQLEw9iYW5xdWVwb3B1bGFpcmUxCzAJBgNVBAMTAmpmMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCkscideEsfU0+UPqM13kPUQVBFYB4xOcHCYqzr6PPgl7Co
GwsjK5o4CKUm/7qWS0BdnqNOdrLaNGZ4kCXlXDg1SemWMlOgPtWl9T3XAiyyr88L
Ei+9sisIUA/JE/3leQWuk0gJXohtxKUwR/fbsWrQjqLspxNK09Urbqz8hwehPQID
AQABo4GNMIGKMA4GA1UdDwEB/wQEAwIE8DA4BgNVHR8EMTAvMC2gK6AphidodHRw
Oi8vODYuNjQuMTAuMTM4LytDU0NPQ0ErL2FzYV9jYS5jcmwwHwYDVR0jBBgwFoAU
gFSsMA0GCSqGSIb3DQEBBAUAA4GBAEm2OLlVyMIzf7Bk7ZUNBCQacvUEdl2o58Pg
py+CMN+K1OdrLaNGZ77TlVKbydqwI2t7hIpuC81c8D3O9r3LiYSDrgxMFhxeUKLD
sIo1dusXjV8nHm5V2zu4hOdrLaNGZix3bEEEFH+cpzOp5y/ogwHWVpz6h3r36Lgo
Vl1S6JU6
-----END CERTIFICATE-----

Hash of the authentification certificate (SHA-256) :

2C A5 BA 6A 2B 56 01 92
35 2A B4 91 53 E9 0B BA
B8 3C B0 19 66 C9 9C 6E
34 C1 5E B5 9F 4A 64 F7

Date : **Signature :**

Initialization Letter for the encryption certificate

Date : TT.MM.JJJJ

Time : HH:MM:SS

Host Id : “BIC11 of the bank”

Bank : « name of the bank »

User-ID : Xxxxxxxx

Partner-ID : Yyyyyyyy

Version : Encryption E002

Encryption Certificate
Type : E002

If CA certificate:
Certificate issued to: name-surname or identifier
Certificate issued by: name the certification authority

-----BEGIN CERTIFICATE-----
MIICcjCCAdugAwIBAgIBDzANBgkqhkiG9w0BAQQFADA6MQswCQYDVQQGEwJGUjEY
zM7nNDE4VQkAUz33C9ztXhG9P3gwHQYDVR0OBBYEFNd6cAJ8L04eB7TiCzpcumIn
MBYGA1UECxMPYmFucXVlcG9wdWxhaOdrLaNGZtYDVQQDEwggdHVyYm9zYTAeFw0w
ODA5MTAxMzI0MzZaFw0xODA5MDgxMzI0MzZaMDQxCzAJBgNVBAYTAmZyMRgwFgYD
VQQLEw9iYW5xdWVwb3B1bGFpcmUxCzAJBgNVBAMTAmpmMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCkscideEsfU0+UPqM13kPUQVBFYB4xOcHCYqzr6PPgl7Co
GwsjK5o4CKUm/7qWS0BdnqNOdrLaNGZ4kCXlXDg1SemWMlOgPtWl9T3XAiyyr88L
Ei+9sisIUA/JE/3leQWuk0gJXohtxKUwR/fbsWrQjqLspxNK09Urbqz8hwehPQID
AQABo4GNMIGKMA4GA1UdDwEB/wQEAwIE8DA4BgNVHR8EMTAvMC2gK6AphidodHRw
Oi8vODYuNjQuMTAuMTM4LytDU0NPQ0ErL2FzYV9jYS5jcmwwHwYDVR0jBBgwFoAU
gFSsMA0GCSqGSIb3DQEBBAUAA4GBAEm2OLlVyMIzf7Bk7ZUNBCQacvUEdl2o58Pg
py+CMN+K1OdrLaNGZ77TIVKbydqwI2t7hIpuC81c8D3O9r3LiYSDrgxMFhxeUKLD
sIo1dusXjV8nHm5V2zu4hOdrLaNGZix3bEEEFH+cpzOp5y/ogwHWVpz6h3r36Lgo
Vl1S6JU6
-----END CERTIFICATE-----

Hash of the Encryption Certificate (SHA-256) :

2C A5 BA 6A 2B 56 01 92
35 2A B4 91 53 E9 0B BA
B8 3C B0 19 66 C9 9C 6E
34 C1 5E B5 9F 4A 64 F7

Date :

Signature :

A5. Payment Status Report Format

Disclaimer:
This annex named: Annex 5 : « EBICS Payment Status Report »
is available on the CFONB website : www.cfonb.org
in the Documentation section: Migration ETEBAC vers EBICS et SWIFTNet

A6. Example of Hash calculation

The hash can be checked using the openssl following command:
openssl x509 -sha256 -fingerprint -in cert.pem

Results into this output:
SHA256 Fingerprint =
A6:16:4F:86:65:AF:84:D5:84:AB:70:51:19:37:2F:4D:61:36:AE:69:C2:6A:F6:AF:31:79:CD:01:37:3C:D4:81


cert.pem


ExempleSignatureTransport.doc

A7 Glossary

Name or Acronym	Acronym Signification	Definition
CA	Certificate Authority/ Certification Authority	In cryptography, a certificate authority or certification authority (CA) is an entity that issues digital certificates for using by other parties.
API	Application Programming Interface	An application programming interface (API) is a set of routines, data structures, object classes and/or protocols provided by libraries and/or operating system services in order to support the building of applications.
Authentication		Authentication is the act of establishing or confirming someone as authentic. This confirms that the identity registered is a trusted one.
Bank		The term "bank" used in this document should be understood as a Payment Service Provider (PSP) under 2007/064/CE Payment Services Directive dated November 17th 2007. The French transposition is the « Ordonnance 2009-866 » dated July 15 th 2009.
CFONB	Comité Français d'Organisation et de Normalisation Bancaires.	French Standard Organization dedicated to resolve technical problems on the banking area context. In particular, it normalizes exchanges between banks and between banks-customers.
Certificate / Digital certificate		In cryptography, a public key certificate (also known as a digital certificate) is an electronic document which uses a digital signature to bind together a public key with a registered identity. If it isn't self-signed, the certificate can be used to verify that a public key belongs to an individual.
Certification chain (trusted chain)	Chaine de certification	Verification of the certification up to the root Aurthority
Self signed certificate		Certificate made by server or workstation, without Certificate Authority
Encryption		In cryptography, encryption is the process to transform information using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to a key. The result of the process is encrypted information (in cryptography, referred to a cipher text).
DER	Distinguished Encoding Rules	ASCII text format allowing transmitting features of an electronic certificate.
EBICS	Electronic Banking Internet Communication Standard	EBICS is a new Standard Communication Protocol between Banks and Customers over TCP/IP and Internet. It provides you with a multi-bankable application.
ETEBAC	Echanges Télématiques Entre Banques et Customers	Current File Transfer Protocol used in France to exchange between Banks and customers. EBICS will replace this protocol over TCP/IP.
Data file / Remittance file		File containing order's remittance data
FileFormat		Typically for France it is the nature of the transaction in order FUL and FDL

Name or Acronym	Acronym Signification	Definition
Hash Seal Seal of the certificate		Single numeric value representing data for integrity control operation.
IP	Internet Protocol	The Internet Protocol (IP) is a protocol used for data transmission across a packet-switched internet network using the Internet Protocol Suite, also referred to TCP/IP.
Order Type :		Type of remittance or upload transaction. In Germany, it includes the nature of the transaction.
Execution order		See order's signature
PEM	Privacy Enhanced Mail	PEM is a DER Format encoded in base64 with ASCII headers. (See also DER). It may contain private keys, public keys and X509 certificates.
Request Type		Type of the request included in the EBICS transaction.
Seal calculation		Mathematical function allowing data integrity's control.
Transport signature		Signature allowing to guarantee the origin and integrity of a message's data. It is not a personal signature. It does not have mandate value.
Personal signature/ Execution order/ Confirmation		The signature of the order is also called "execution order" or "personal electronic signature". It authenticates personally the sender of a message
Separate signature		The instructions and signature of the order are not transmitted in same flow but in an asynchronous way. They can be transmitted through the same or different channel at to the same moment or shifted time.
SWIFT	Society for Worldwide Interbank Financial Transaction	
TLS	Transport Layer Security	
VEU :	Verteilte Elektronische Unterschrift	Separate Signature Process in EBICS.
X25	X25	Normalized packet switching network used in France by ETEBAC.
ZKA	Zentraler KreditAusschuss	German Standard Organization to normalize banking area context in Germany. New name since 2011 : DK = Die Deutsche Kreditwirtschaft