

Web Services Security and Communication

Description

Content

Content	2
1 Web Services file transfer	2
1.1 Communication via TCP/IP network.....	3
1.1.1 Canonicalization.....	3
1.2 Backup systems.....	3
1.3 Security.....	3
1.3.1 Certificates.....	3
1.3.2 Automatic download.....	3
Step by step instructions:.....	4
1.3.3 Manual download ³	5
1.3.4 Roles.....	5
1.3.5 Signature.....	6
Replay:.....	6
1.4 Response message and duplicate check.....	6
1.5 WSDL.....	6
1.6 Messages to/from the bank - Upload/Download a file.....	6
1.7 SOAP envelope ApplicationRequest and payload.....	7
1.8 Standards used.....	7
2 Transfer to the bank - Upload File	7
2.1 Transfers from the bank - Download file.....	8
2.2 File operations.....	8
3 Testing	8
3.1 Testing in production environment using general demo certificate.....	8
3.2 Testing in production using customer's own production certificate.....	9
4 Schedules	9
4.1 Files to the bank.....	9
4.2 Files from the bank.....	9
5 File transfer parameters and rules	10
5.1 Encoding Rules.....	10
5.2 Validation Rules.....	10
5.3 Compressing rules for XML payload.....	10
5.4 Time zone.....	11
5.5 Authentication, Authorization, Integrity and Non-Repudiation.....	11
5.5.1. Authentication.....	11
5.5.2. Authorization.....	11
5.5.3. Integrity control.....	11
5.5.4. Non-Repudiation.....	11
5.6 Use of S/MIME or SoA (SOAP with Attachments) within Web Services.....	11
6 References	12
7 Message Structure	12
7.1 Service Content (Payload).....	12
7.2 ApplicationRequest.....	12
7.3 ApplicationRequest elements.....	13
7.4 ApplicationResponse.....	21
7.4.1 Error codes.....	37
7.5 SOAP envelope.....	38
7.6 Examples of SOAPs and ApplicationRequest/ApplicationResponse.....	39
8 Customer support	46
8.1 Customer support – Finland.....	46
• E-support for Corporate Customers, tel 0200 67210 (in Finnish).....	46
• E-support for Corporate Customers, tel 0200 67220 (in Swedish).....	46
• E-support for Corporate Customers, tel 0200 67230 (in English).....	46
8.2 Corporate eGateway Support.....	46
• +46 771 77 69 75.....	46
9 Additional information	46

1 Web Services file transfer

This document describes the Web Services (hereinafter WS) -file transfer provided by Nordea (hereinafter Nordea or the bank). WS file transfer is used together with WS data communication protocol providing PKI authentication and security according to standards defined by WS-I organization (www.ws-i.org). This document describes the standard when applicable from Nordea perspective.

Nordea's Web Services data communication protocol is described in more detail in separate instructions. These instructions are mainly intended for companies producing bank connection software to ensure that all the properties and safety features of the Web Services can be complied with accurately.

The instructions are divided into the following classes (the date in the filename extension indicates the latest version and can vary):

1. WebServices_Messages_20081022_105.pdf

- Nordea, OP-Pohjola Group, Danske Bank: Security and Message Specification for Financial Messages Using Web Services, 2008.
- This description is drafted by Finnish banks and it can be used to produce WS client software compatible with the services of all banks applying this description.

2. Web Services Description Language, WSDL

- Technical description of WS client software. This is a configuration file in a XML format created for the automatic processing of a client application. The file name is in the format BankCorporateFileService-yyyymmdd.wsdl, in which 'yyyymmdd' indicates version update. The Finnish banks have one common WSDL file.

3. ApplicationRequest-yyyymmdd.xsd and ApplicationResponse-yyyymmdd.xsd, in which 'yyyymmdd' indicates version update; for example 20080114.

- The banks have common schema files of these XML structures.

4. Nordea's Web Services Service Description

- The description defines Nordea's requirements for the use of the WS protocol in more detail.

5. Nordea Web Services Security and Communication Description

- The description specifies in detail the message structure and its security features and the field contents in line with Nordea's requirements.

6. CertificateService_20100219.WSDL

- Technical description of WS client software for downloading WS certificates. The banks do not have a common procedure for certificate download.

Documents 1, 2 and 3 can be downloaded from the website of the Federation of Finnish Financial Services at www.fkl.fi.

Documents 4, 5 and 6 are available on www.nordea.fi website: *Corporate customers >> Payments >> Web Services >> Instructions and sample files >> Testing >> Web Services*

Nordea Web Services provides local services from Finland, Estonia, Latvia and Lithuania. These Cash Management Services are local services and file types are local file types. Therefore terms and conditions of using the local services must be taken into account. WS only harmonizes the communication and security. Web Services connection supports also file types which are used in Corporate eGateway service.

Corporate eGateway supports a centralized payment and collection factory.

It is Nordea's file-based, mass payment service with one point of entry for bulk payments and collections in the Nordic and Baltic countries, Germany, Russia, UK, Canada and the US. The service provides a uniform file interface that covers all relevant types of domestic and cross-border payments, including direct debits in the Nordic area.

1.1 Communication via TCP/IP network

The Nordea WS is reachable through Internet TCP/IP network using HTTPS protocol. Communication parameters are described in *section 5*.

WS specifies a reliable messaging mechanism and supplies a choreography model that WS will follow. Communication is always activated by the customer, also when requesting a file.

Some requirements from the vendor point of view are:

- No conversational session state (services are stateless)
- Use of PKI for authentication and signing
- Use of Web Services standards
- Certificate handling software

1.1.1 Canonicalization

Canonicalization is a strategy for standardizing XML structures so that they compare identically across platforms. It is important for a signed document because the digest may change and signature validation will fail. Therefore, XML is always canonicalized before being hashed or signed, and both sides of the communication must agree on the canonicalization method used. The standard used for Nordea WS algorithm: <http://www.w3.org/2001/10/xml-exc-c14n#>.

1.2 Backup systems

In case of a failure in the WS connection, as a backup system we recommend using *Nordea Corporate Netbank*. The backup system will be activated by the Corporate's own request. The backup system must always be tested in advance and it requires a separate agreement, security and identification methods.

1.3 Security

WS defines the overall message exchange of the business documents, with elements to support authentication, authorization and integrity; details of the bindings for the transfer protocols (e.g. HTTPS); and the specification for a reliable exchange of messages between partners.

The SSL (Secure Socket Layer) protocol will be used for securing the transport between client application and Nordea. The SSL version must be at least 3.0.

With the WS messages the actual payment instruction or the Payload/Business Message is complemented with elements according to WS standards (SOAP). It forms an overall envelope or container within which all business documents are connected. The payload can be any file format including a XML and it is transported through the service wrapped in a digitally signed XML structure, called ApplicationRequest or ApplicationResponse.

WS specifies the header called the SOAP header, an instance of which must always precede a business document instance in SOAP body.

With the WS connection the security requirements in data transfer are fulfilled by secured connection (HTTPS) and digital signatures, which authenticates the parties, the transferred data and the actual transfer.

1.3.1 Certificates

The bank supplies customer's certificate as a certificate file which must be used in communication. Both parties supply each other the public part of their signature keys within message content, in its signature element. Nordea offers company based and user based¹ certificates.

Contact your local Nordea branch office to make an agreement of the service and receive details to get your certificate by using one of the methods below.

1.3.2 Automatic download

Nordea offers an automated way of downloading customer certificates to be used with WS channel. The service is based on Certificate Signing Request (CSR) provided and sent to Nordea by banking software using Web Services channel. There exists a separate WSDL to be used for this service. It can be retrieved together with other information and examples, www.nordea.fi: *Corporate customers >> Payments >> Web Services >> Instructions and sample files >> Testing >> Web Services*

¹ Only a company based certificate is available when the Corporate eGateway service is used.

The certificate service has a separate URL: <https://filetransfer.nordea.com/services/CertificateService>

The PKCS#10 formatted CSR is in CertApplicationRequest in base64 coded format. In addition, in CertApplicationRequest there is a HMAC check which is generated using the CSR and a customer specific 10-digit activation code. This activation code is received by the customer via SMS, to the mobile number which was registered in the bank together with the agreement.

The response to the CertApplicationRequest is CertApplicationResponse, which contains signed certificate in PKCS#7 format, if the needed information was correct in the request and the customer could have been identified based on this information. Three information fields are important in the CSR request:

1. CN = Customer's name from the agreement
2. SerialNumber = Customer's User ID from the agreement
3. C = country code, e.g. EE, FI, LT, LV

The received certificate will be connected to the corresponding private key at the customer, and must be stored securely and protected with a strong password given by the user.

Please take into consideration different risk scenarios while working with certificates which are in form of data file. The customer is always responsible of storing the certificate secure, without unauthorized access to it.

Step by step instructions:

1. The administrator makes an agreement for the company using Web Services and receives 10-digit activation code to his/her mobile phone via SMS. It is valid for 7 days, but a new one can be ordered again from the Customer support (see *section 8*)
2. Banking software generates certificate signing request using the following information, which must be available when requesting a certificate from Nordea.
 - Company name (company based certificate) or user name (user based² certificate) as in agreement
 - UserID as in agreement country code, e.g. EE, FI, LT, LV
 - 10 digit activation code received by administrator registered into the agreement
3. A banking software creates a key pair for the certificate and generates the PKCS#10 request (a private key is newer sent to the bank).
 - use: key length 2048bit, SHA-256 algorithm, DER –encoded
 - Subject info: CN=name, serialNumber=userID, C=country (as above)
4. Create HMAC seal
 - use DER coded PKCS#10 above as input
 - Activation code as the key (10-digits)
5. Send PKCS#10 with HMAC seal to Nordea
 - using schema: CertApplicationRequest
 - put PKCS#10 in base64 format to Content element
 - put calculated HMAC to HMAC element
 - put code "service" to Service-element
 - put "GetCertificate" to Command element
 - place CertApplicationRequest in base64-encoded format in body element of the SOAP
 - The SOAP message need not to be signed

² Only a company based certificate is available when the Corporate eGateway service is used.

If everything is OK:

6. Receive the certificate from Nordea

- Schema: CertApplicationResponse
- The certificate is in pkcs#7 format
- Connect the private key with the certificate and store it safely

If request contained invalid subject info name:

7. Recreate CSR with correct name received from Nordea. Correct name is in Fault text tag in the response message.

8. Recreate HMAC seal (using same activation code as in 1st round)

9. Resend seal and PKCS#10 to Nordea

- if valid request, see step 6

If 2nd request was still invalid or User ID or country was false in the 1st request:

- error message from Nordea

1.3.3 Manual download³

The certificate can be downloaded to the customer's system by using the Nordea Security Client (NSC) which is provided by Nordea. Only the Windows environment is supported. A downloaded certificate can be used with any operating system (Windows, Linux etc.).

Identification data for downloading, such as company name or user name and user ID, are found in the Web Services agreement. The one-time activation code is received as a SMS message to the administrator's mobile phone number indicated in the agreement.

You can find more information on www.nordea.fi website: *Corporate customers >> Payments >> Web Services >> PKI certificate*

In order to use the certificate in the bank connection software, you must give the software access to the certificate file. More detailed instructions are available in each bank connection program's own instructions.

You can save the certificate to an USB memory stick, in which case no copy of the certificate will be saved on the hard drive. The USB stick can also serve as a backup copy if it is stored in a secure place. If the certificate is temporarily saved on the hard drive, it should be deleted after use for security reasons. Take a backup copy and delete the copies from the hard drive.

1.3.4 Roles

In communication there can be different players. The end user or the customer is typically the one who created the payment file or activates the request of a file. The customer has an agreement for used services with the bank. However, it may be another person or maybe a standalone technical environment which does the actual communication later than the creation of the file occurred. Also a third party operator can be involved in transmission. The requester that activates the physical connection is called a sender.

When Nordea Web Services is activated, the service first authenticates the service requester/sender by checking the certificate. Later the end user will be authorized by checking the signature of the ApplicationRequest.

If the sender and the end user is the same person, also the certificate can be just one for both signatures.

³ Nordea prefers automatic certificate downloading instead of the manual downloading method.

1.3.5 Signature

XML Signature is a W3C recommendation that defines XML syntax for digital signatures. The service requester uses the certificate to sign the data and so a base64-encoded signature is created as an output from the certificate client software.

Authorization check will always be performed by the bank for the sender who must be authorized to deliver the SOAP messages. The certificate used to sign the SOAP message can be the same which is used to sign the ApplicationRequest. However, the customer may choose a third party to execute the physical transportation of the files and this third party needs a certificate of their own, issued by Nordea.

Also a payment factory setup in an enterprise may need this kind of setup if a centralized unit manages the bank communication for other units and when units will have their own service agreements and they sign their own requests before transportation to centralized communication unit.

Standard used for WS communication at Nordea for SOAP envelope is:

•*SignatureMethod Algorithm* = <http://www.w3.org/2001/04/xmldsig-more#rsa-sha256>

•*DigestMethod Algorithm* = <http://www.w3.org/2001/04/xmlenc#sha256>

Authorization check will also be performed by the bank for the service requester/customer by checking the signature on protected data, included in the service request (ApplicationRequest).

The specification allows one or several enveloped type XML digital signatures in one ApplicationRequest. Currently only one signature is allowed at Nordea Web Services. One signature represents a corporate or one user in the corporate, who has the authority for services registered for him. Each certificate refers to an individual user ID in the bank's agreement system. user ID is used also for support purposes and it should be available before contacting the Customer support.

Replay:

It is assumed that messages include digitally signed elements like TimeStamp, Sequence number etc. to allow message recipients to detect replays of the message when the messages are exchanged via an open network. WS channel is not checking replays, some services may check duplicates.

1.4 Response message and duplicate check

A response message acknowledges that the recipient has received the message and it should be received immediately in the same open HTTPS channel. If timeout occurs, the customer must contact the Customer support (see *section 8*). The Cash Management payment message can be resent only if the Customer support allows it after cancelling the previous initiation, if possible. In case the SOAP message is resent exactly as the original message, Nordea service can prohibit double processing.

A signed response message (ApplicationResponse), containing data or (error-) message, will be returned in the same connection as the service request. SOAP message is signed also.

1.5 WSDL

A WSDL is a XML description file that is used for Web Services to build up the functionality with the Server. The WSDL can be downloaded from Federation of Finnish Financial Services (FFFS, www.fkl.fi). A common WSDL is used by all banks in Finland, but the communication address in WSDL must be replaced by the TCP/IP address of each bank.

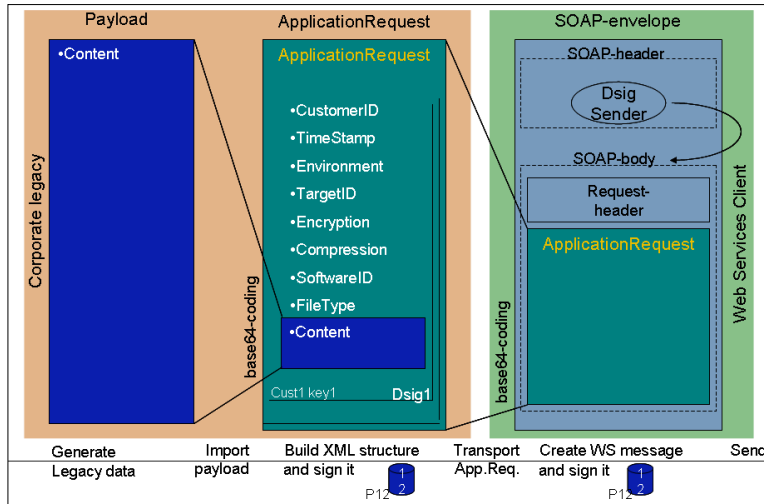
1.6 Messages to/from the bank - Upload/Download a file

The file transfer service validates the customer's message upon reception. An incorrect message, or a message without a corresponding Cash Management service agreement, is rejected. After reception the bank sends to the customer a Response, which informs the customer that the bank has received the message and forwards it to further processing, or that the message has been rejected due to, for example technical errors or errors in the content. If the customer doesn't receive a Response message, the customer must confirm the status of transfer with Customer support (see *section 8*). Uploaded files are for example payments to be processed in a respective Nordea unit.

For download files the customer initiates a query to retrieve information (a file of specific file type) from the bank. Information about downloadable files can be retrieved by sending a query of Download files and receiving a list of files as a response.

1.7 SOAP envelope ApplicationRequest and payload

WS do not use encryption of the payload as it is transferred using HTTPS protocol. Digital signing of payload prevents unauthorized access to payload during transfer. SOAP envelope message structure is described in *section 7.4*. An example of message structure is illustrated in the following picture.



1.8 Standards used

- WS-I Basic Security Profile Version 1.0
- Web Services Security X.509 Certificate Token Profile, OASIS Standard
- X.509 will be used for digitally signing digests of uploaded files and web service requests
- SOAP 1.1
- HTTPS 1.1

2 Transfer to the bank - Upload File

Each request reply is independent from each other. There is no specific order required for the requests. Web Services will provide the transport of the customers file to Nordea. The response from Nordea will be a transport acknowledgement. Some backend system will process the files in batch mode. This means that the only verification of a file transfer, successful or not, will be a transfer acknowledgement. The client will not receive any other notification.

- The customer sends the SOAP message to the bank as a WS message with digital signature, by which the bank recognises and authenticates the customer.
- The bank sends an acknowledgement or response immediately after reception back to the sender.

Customer	TCP/IP network	Nordea Web Services
Signed SOAP Request with Payload in Signed ApplicationRequest	→	Authentication of the sender, Authentication and authorization check for the customer, validation of the Payload
Authentication of the sender, check for the status in response	←	Signed SOAP Response with status either in SOAP fault or in Signed ApplicationResponse

2.1 Transfers from the bank - Download file

Each request reply is independent from each other. There is no specific order required for the requests.

- The customer requests a file by sending a digitally signed query.
- The bank responds to the customer with a digitally signed message, by which the customer authenticates the bank and retrieves the requested information.

2.2 File operations

The Nordea Web Services provides these individual operations

GetUserInfo	The service will provide the client with information of authorized user file types and service ID's
DownloadFileList	The service will provide the client with a list of files that are available for download from Nordea
DownloadFile	<p>The service will provide the client with requested files. Downloadable files can be checked by DownloadFileList –service. The query may be:</p> <ul style="list-style-type: none"> • download single file • download multiple files • download all files of type • download all files <p>File(s) are flagged as downloaded. They can be downloaded again (with help of Customer support) but will no longer show up as “new files”.</p>
UploadFile	<p>The Service will provide the transport of the customers file to Nordea. The response from Nordea will be a transport acknowledgement with details regarding the status of the transport.</p> <p>Backend system will process the files in batch mode. This means that the only verification of a file transfer, successful or not, will be a transfer acknowledgement.</p> <p>The client will not usually receive any other notification and the result must be retrieved with a new call later.</p> <p>Dependent on file type, a “confirmation file” is sent allowing the system to confirm the file(s), or a separate ConfirmFile use case is sent. (Confirm not in use yet)</p>

3 Testing

Nordea offers vendors and developers a possibility to test the WS connection towards production environment to ensure smooth usage later for their customers.

Nordea recommends that, before testing, you read this *Web Services Security and Communication Description*, *Web Services Service Description*, and testing instructions on www.nordea.fi website: *Corporate customers >> Payments >> Web Services >> Instructions and sample files >> Testing >> Web Services >> Testing*

Nordea is not liable for damage caused by the incorrect functioning of bank connection software and a bank connection software must be tested before implementation.

3.1 Testing in production environment using general demo certificate

Testing towards production environment can be carried out using the general demo certificate. The demo certificate and instructions for downloading are available on www.nordea.fi website: *Corporate customers >> Web Services >> Instructions and sample files >> Testing >> Web Services >> Testing*

With the connection test, the specific test certificate (demo certificate) must be used, and the word **PRODUCTION** must be entered in the Environment field of the ApplicationRequest. The connection test is performed in Nordea’s Production environment using the demo certificate and Nordea’s other general test IDs.

Demo certificate parameters:

- User ID (CustomerId): 11111111
- PIN: WSNDEA1234
- File type set (TargetId): 1111111A1

The service codes and payment accounts used for testing each Cash Management service are specified in the relevant service description.

Note that, when downloading files or account feedback with test IDs, you **will always receive the default test files** regardless of the files you have uploaded.

Customer support assists in matters related to file testing (see *section 8*).

3.2 Testing in production using customer's own production certificate

The purpose for testing with customer's production certificate, is to verify in new implementations, that customer material meets the requirements and is validated without errors. All accounts, references and other content of the file should be real production information. Also Cash Management service agreements must be in place.

The Environment-field in ApplicationRequest must contain a word **TEST**. The account numbers and service Id's in the file are customer's own. Validations are performed and respective feedback is generated. However, the material sent to the bank, will not be processed and no account debit/credit will be executed.

Note: If there is not a word **TEST** in the Environment field in the ApplicationRequest, the sent file is forwarded to **normal payment processing**. In special cases, if agreed beforehand, the service agreement can have a status which prohibits forwarding of the file to execution. However, it is recommended to use word **TEST** in the Environment field in all testing actions.

Customer support assists in matters related to file testing; see *section 8*.

4 Schedules

The WS file transfer service is open 24 hours a day, 7 days a week.

The processing of uploaded files follows services' cut-off times. The schedule of each Cash Management-service is available on Nordea's website: www.nordea.com, www.nordea.fi.

4.1 Files to the bank

The bank forwards messages for further processing on banking days. Depending on their content, files are distributed for processing into different Cash Management services. Each service has its own service descriptions which are available on Nordea's websites: www.nordea.com, www.nordea.fi. The bank will confirm the reception of the Cash Management-message file by responding with SOAP response message, informing the sender of the accepted or rejected reception of the file in the bank's back-end systems. If the customer does not receive any response in specified time, the Customer support should be contacted before resending the file. Otherwise there is a risk of double execution of payments or that the file never reached bank's back-end systems.

The payment or other file sent to the bank for execution may be rejected later for different reasons. An immediate file transfer response with error code 00, stating OK, does not mean that the execution will be guaranteed on back-end systems.

4.2 Files from the bank

Messages are stored according to their completion and storage schedules in Nordea's file transfer systems. The schedules are available on Nordea's websites: www.nordea.com, www.nordea.fi.

During this period the customer can retrieve the file(s) from Nordea by sending respective request for Downloading files. The default is downloading only files created after the last retrieval, but files can be retrieved again by setting appropriate parameter to the request (NEW, DOWNLOADED, ALL).

5 File transfer parameters and rules

The WS connection is established using open Internet network (TCP/IP) following secure HTTPS protocol. The parameters for production and test connections are:

Company name	Nordea Bank AB
URL for file upload/download (Production environment)	https://filetransfer.nordea.com/services/CorporateFileService/sha2
URL for file upload/download (Test environment, available only when Corporate eGateway service is used)	http://filetransfer.test.nordea.com/services/CorporateFileService/sha2
Outbound traffic IP (sending IP)	No sending, only Push-Pull
Encryption tool used	SSL
SSL Encryption Algorithm	SHA256 key length: 3072bit
Certificate download (Production environment)	https://filetransfer.nordea.com/services/CertificateService/sha2
Certificate download (Test environment, available only when Corporate eGateway service is used)	http://filetransfer.test.nordea.com/services/CertificateService/sha2
Certificate standard	base-64 encoded X.509
Certificate expiration period	2 years
Signature Algorithm	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Public key certified by	Nordea Bank AB

5.1 Encoding Rules

For service content (the Payload), WS permits UTF-8 encoding schemes, or what character set is stated in used service's service description. Nordea uses ISO-8859-1 in internal processes.

5.2 Validation Rules

The SOAP envelope is validated in the bank when received. The following is the minimum level of parts that are required:

- SOAP Header
- SOAP Body, including Request Header and ApplicationRequest/ApplicationResponse, with optional Service Content (Payload)
- Elements in header and body parts must follow the specifications described in WebServices_Messages.pdf –document provided by Federation of Finnish Financial Services (FFFS).
- ApplicationRequest.xml and ApplicationResponse.xml structures must be validated towards respective schemas.

5.3 Compressing rules for XML payload

If a XML format file sent to the bank, e.g. XML payments includes tens of thousands of transactions, the file must be compressed. Do this before signing and uploading the file. In this way the size of the file can be decreased significantly for the signature process and the transmission. The size of the largest acceptable single file is 50 megabytes (about 50,000 uncompressed transactions).

The supported compression algorithms are GZIP and PKZIP (only the deflate algorithm is supported). Nordea recommends using the GZIP compression algorithm.

The Compression element of the compressed file's ApplicationRequest message must include the value 'true' and the value of the CompressionMethod element must be the name of the compression algorithm, e.g. GZIP.

Status message for compressed file content is not available during the same Web Services connection.

Only a technical receipt is available in the ApplicationResponse. A customer has to download the status for the payload contents separately later on.

Note: A compression is available only when a XML file format is used. Compression can also be used in XML files downloadable from the bank.

5.4 Time zone

ISODateTime -time should be used for all date and time references specified on requests and responses. If no time zone is specified, UTC time zone assumed. That is, however, not the case for date and time specified in the payloads, because those are created for local purposes in different units, where local time is used as time reference.

5.5 Authentication, Authorization, Integrity and Non-Repudiation

The sending partner is required to digitally sign the messages sent to its partner. The receiving partner authenticates the message by following the standard PKI mechanisms to verify the digital signatures.

5.5.1. Authentication

Authentication is used for making sure that the creator/sender of a message is who the creator/sender claims to be. This is accomplished by digitally signing the message (ApplicationRequest/ApplicationResponse and SOAP-body).

In Nordea Web Services communication business messages (ApplicationRequest/ApplicationResponse) are always digitally signed (Enveloped type), following the specification in XML digital signature standard:

<http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/Overview.html>

The specification allows more than one Enveloped-type signature, but currently Nordea supports only one digital signature per ApplicationRequest/ApplicationResponse message.

The sender of the SOAP message signs the SOAP message (body, detached type of signature) using above mentioned standards and Web Services Security standards (WSS). Nordea authenticates the Sender by checking the validity of the signature, but Sender as such has no authorization to use any underlying services. In some cases the private key is used to sign the SOAP message, can be the same as with signing the ApplicationRequest.

5.5.2. Authorization

Authorization is the act of making sure that the signer of a message is permitted or authorized to use the agreed service at the receiving partner. The agreement of using services is used for control that the partner (as identified in the ApplicationRequest) is authorized to use the service subject to message.

Authority to use services is validated only by checking the signature in ApplicationRequest.

5.5.3. Integrity control

When a message is digitally signed, any change in the message will invalidate the signature.

5.5.4. Non-Repudiation

A message's recipient insist the sender to attach a signature in order to make later repudiation difficult, since the recipient can show the signed message to a third party (e.g., a court), thus an user cannot repudiate a signed message without repudiating their signature key. This prevents an initiating partner later denying that they originated contents of a Business Document.

5.6 Use of S/MIME or SoA (SOAP with Attachments) within Web Services

The WS Specification does not support any type of MIME entities or attachments to the messages. The Payload is always in ApplicationRequest/applicationResponse, in an element, called Content, in base64-coded format.

6 References

WS-I Basic Security Profile Version 1.0:

<http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>

Web Services Security X.509 Certificate Token Profile, OASIS Standard:

<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf>

XML Signature:

<http://www.w3.org/TR/xmlsig-core/>

SOAP:

<http://www.w3.org/TR/soap/>

7 Message Structure

The Business message consists of the following components:

- Service Content, Payload (optional)
- ApplicationRequest, signed, or
- ApplicationResponse, signed
- SOAP envelope, signed (body)

The above mentioned components are described below in detail.

7.1 Service Content (Payload)

Service Content is the actual data content aimed for service agreed with the bank. The format of the data is not visible to the Web Services channel because it is base64-coded. This means that any file format (Binary, ASCII, XML etc.) can be transported through the channel.

The content is irrelevant for used transport channel and is presented the same way for example in Corporate Netbank File Transfer. This enables also any backup channel to be used if appropriate identification and authority to use this channel is in place.

The signer of ApplicationRequest must have the authority to deliver this file type to the bank.

This component is optional in the message structure, because ApplicationRequest is used also for requesting data from the bank (DownloadFile). In this case there is no data to be sent to the bank and thus Content element is empty.

7.2 ApplicationRequest

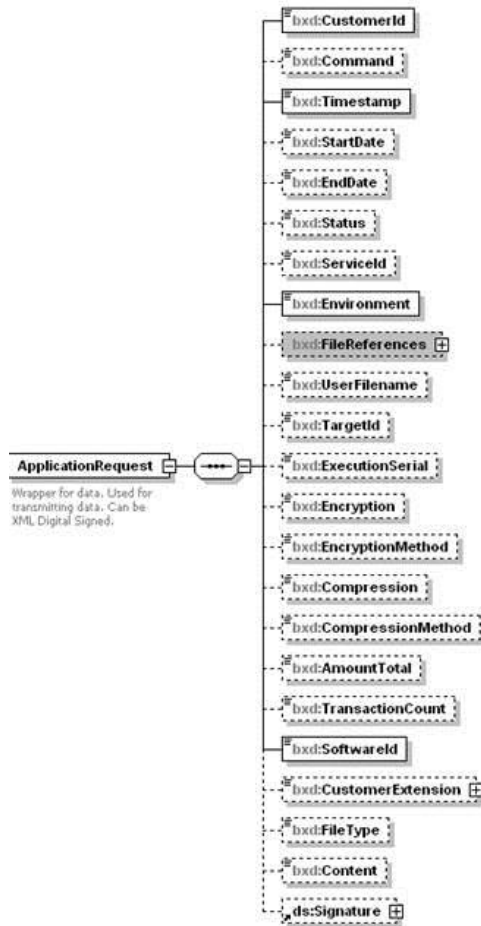
ApplicationRequest is a wrapper for transported data (Payload) to enable a XML digital signature to any file type, whether it is a XML, ASCII or binary format. The Payload is inserted in one element, called Content. It is base64-coded to make it loose from ApplicationRequest schema.

ApplicationRequest has some supporting elements for information purposes, like CustomerId, TimeStamp, SoftwareId etc.


7.3 ApplicationRequest elements

The following describes each element in ApplicationRequest and how it is used when delivered to Nordea bank.

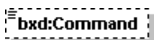
ApplicationRequest_20080918.xsd



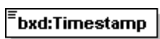
element ApplicationRequest/CustomerId

diagram	
Nordea rule: Mandatory	Signature authenticates the identity of the customer. CustomerId must correspond to Signature. This element is always mandatory in all operations. user ID from the Web Services Agreement Example: 1205585055
type	restriction of xs:string
properties	isRef 0 content simple nillable false
facets	minLength 1 maxLength 16


element ApplicationRequest/Command

diagram	
Nordea rule: Mandatory	This element specifies the requested operation. The values are case sensitive. Element must be included in the request and its content must match the operation specified by other means. One of the following codes must be used: <ul style="list-style-type: none"> • UploadFile • DownloadFileList • DownloadFile • GetUserInfo Example: UploadFile
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple nillable false
facets	minLength 1 maxLength 32


element ApplicationRequest/Timestamp

diagram	
Nordea rule: Mandatory	Time and date when the ApplicationRequest was created. Data Type: ISODateTime, UTC or local time Example: 2008-06-03T14:45:35.424+03:00
type	xs:dateTime
properties	isRef 0 content simple


element ApplicationRequest/StartDate

diagram											
Nordea rule: Optional	<p>When requesting data from the bank, e.g. with the DownloadFileList operation, this element can be used to specify filtering criteria. This element contains a date which specifies the starting point of the time filter, inclusive. If this element is not present, but EndDate is given, it means the filtering criteria does not have a starting point.</p> <p>Data Type: ISODate Example: 2008-06-03 (yyyy-mm-dd)</p>										
type	xs:date										
properties	<table> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>0</td></tr> <tr><td>maxOcc</td><td>1</td></tr> <tr><td>content</td><td>simple</td></tr> <tr><td>nillable</td><td>false</td></tr> </table>	isRef	0	minOcc	0	maxOcc	1	content	simple	nillable	false
isRef	0										
minOcc	0										
maxOcc	1										
content	simple										
nillable	false										

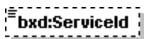
element ApplicationRequest/EndDate

diagram									
Nordea rule: Optional	<p>When requesting data from the bank, e.g. with the DownloadFileList operation, this element can be used to specify filtering criteria. This element contains a date which specifies the ending point of the time filter, inclusive. If this element is not present, but StartDate is given, it means the filtering criteria does not have an ending point.</p> <p>Data Type: ISODate Example: 2008-06-03</p>								
type	xs:date								
properties	<table> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>0</td></tr> <tr><td>maxOcc</td><td>1</td></tr> <tr><td>content</td><td>simple</td></tr> </table>	isRef	0	minOcc	0	maxOcc	1	content	simple
isRef	0								
minOcc	0								
maxOcc	1								
content	simple								


element ApplicationRequest/Status

diagram									
Nordea rule: Optional DownloadFile	<p>When requesting data from the bank with the DownloadFile or DownloadFileList operation, this element must be used as filtering criteria. This status is used to filter the requested data. One of the following codes are possible to use: NEW, DOWNLOADED, ALL.</p> <p>Example: DOWNLOADED</p>								
type	restriction of xs:string								
properties	<table> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>0</td></tr> <tr><td>maxOcc</td><td>1</td></tr> <tr><td>content</td><td>simple</td></tr> </table>	isRef	0	minOcc	0	maxOcc	1	content	simple
isRef	0								
minOcc	0								
maxOcc	1								
content	simple								
facets	<table> <tr><td>minLength</td><td>1</td></tr> <tr><td>maxLength</td><td>10</td></tr> </table>	minLength	1	maxLength	10				
minLength	1								
maxLength	10								


element ApplicationRequest/ServiceId

diagram	
Nordea rule: Optional	Additional identification information of the Customer, for example an Account Number or similar. The content can be copied from response of "GetUserInfo" request or from Agreement of Payment services. Example: NDEAFIHXXX-FI1-EUR-1950180000010
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 256

element ApplicationRequest/Environment

diagram	
Nordea rule: Mandatory	This field specifies how the request will be executed, in production or testing in production or test environment. For normal production the code is: PRODUCTION. For testing the Upload use case in production environment with production certificate: If production certificate is used and code is TEST, the sent file will NOT be executed. The response received is based on sent file.
type	bxd:EnvironmentCode
properties	isRef 0 content simple
facets	pattern PRODUCTION / TEST

element ApplicationRequest/FileReferences

	
Nordea rule: Mandatory DownloadFile. Ignored in other operations	Unique identification of the file which is the target of the operation This element is used and is mandatory in operation DownloadFile to specify one specific file as the target of the operation The customer must have obtained the FileReference values beforehand, e.g. using the DownloadFileList or UploadFile operations. This value is generated in the bank. <FileReferences> <FileReference> 123456789</FileReference> <FileReference> ABCDEFGHI</FileReference> </FileReferences>
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 16

element ApplicationRequest/UserFilename

diagram	
Nordea rule: Optional	A name given to the file by the customer in the operation. It is stored in the bank logs for customer support, but is not used by any other bank systems. Please note that the real identification of a file is the FileReference. Example: SEPA_CPS.XML
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 80


element ApplicationRequest/TargetId

diagram	
Nordea rule: Mandatory Optional in GetUserInfo	The logical folder name where the file(s) of the customer are stored in the bank. A user can have access to several folders. TargetId's are included in the customer service agreement. Example: 0012345678
type	restriction of xs:string
properties	isRef 0 content simple
facets	minLength 1 maxLength 80


element ApplicationRequest/ExecutionSerial

diagram	
Nordea rule: Optional	An identifier given the customer to identify the particular request. The bank does not enforce the uniqueness of this identifier -the value is used only by the customer. It is returned in response. This element is optional. Using ISO timestamp is recommended. Example: 001:
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 32


element ApplicationRequest/Encryption

diagram	
Nordea rule: Optional but must contain "false" if present	If this element is present and the content is "false" (case-sensitive) it means that the Content is NOT encrypted or the requested data should NOT be encrypted by the bank. The value "true" is not implemented yet. If the value is true, request will be rejected for now. Example: false
type	xs:boolean
properties	isRef 0 minOcc 0 maxOcc 1 content false


element ApplicationRequest/EncryptionMethod

diagram	
Nordea rule: Not used	

element ApplicationRequest/Compression

diagram	
Nordea rule: Optional but must contain a value if present. To be used only with UploadFile – command.	Compression indicator for the content. If this element is present and the content is "false" (case-sensitive) it means that the uploaded Content is not compressed. The value "true" indicates that the Uploaded Content is compressed. In this case also CompressionMethod element must have a value. Optional, but must contain a value if present. Example: false or 0
type	xs:boolean
properties	isRef 0 minOcc 0 maxOcc 1 content simple

element ApplicationRequest/CompressionMethod

diagram	
Nordea rule: Optional but must contain a value if present. To be used only with UploadFile – command.	Compression algorithm used is RFC1952 GZIP. Nordea supports also PKZIP but recommends using GZIP. Compression is performed on the original raw data, i.e. before base64 encoding and before placing payload in the Content element. Application Request Compression-element must contain value true. If compression is used CompressionMethod-element must state used compression algorithm GZIP or PKZIP. Example: GZIP

element ApplicationRequest/AmountTotal

diagram	
Nordea rule: Not used	

element ApplicationRequest/TransactionCount

diagram	
Nordea rule: Not used	

element ApplicationRequest/SoftwareId

diagram	
Nordea rule: Mandatory	This element contains the name and version of the client side software which generated the ApplicationRequest. It is used for customer support purposes Example: ABC Soft version 1.0
type	restriction of xs:string
properties	isRef 0 content simple
facets	minLength 1 maxLength 80

element ApplicationRequest/CustomerExtension

diagram	
Nordea rule: Not used	.

element ApplicationRequest/FileType

diagram	
Nordea rule: Mandatory	Specifies the type of file in the UploadFile request. Can also be used as a filter in the operations DownloadFile and DownloadFileList. Available file types are specified in the WS service description. Example: NDCORPAYS
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 40

element ApplicationRequest/Content

diagram	
Nordea rule: Mandatory in operation UploadFile. Ignored in other operations	The actual file in the UploadFile operation. The content is in Base64 format. Example: TE0wMjAwMTY2MDMwMDEwMDY4MjcgICAglCAw
type	xs:base64Binary
properties	isRef 0 minOcc 0 maxOcc 1 nillable true

element ApplicationRequest/Signature

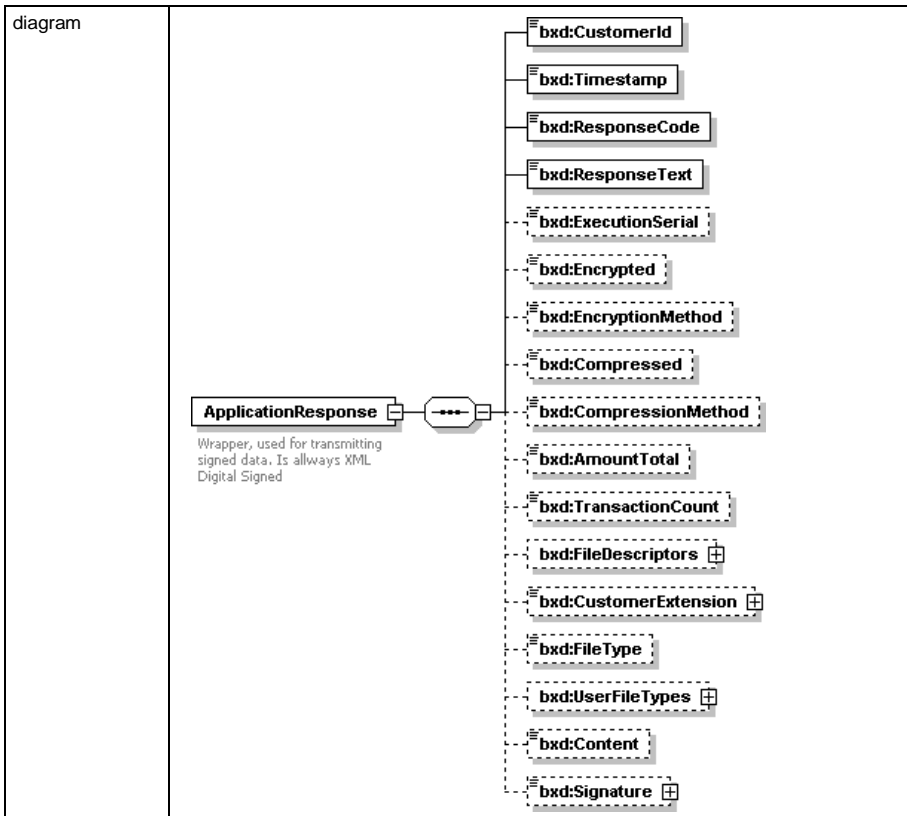
diagram													
Nordea rule Mandatory	This element is created by the signature operation by the customer. Its content is specified by the XML Digital Signature standard. This element is mandatory when sending any request to the bank as it is used for integrity verification and authentication. This element is defined as optional in the schema because the recipient can remove the signature element after verification of the signature, before schema validation Example: <pre><Signature xmlns="http://www.w3.org/2000/09/xmldsig#"> <SignedInfo> <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/> <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/> <Reference URI=""> <Transforms> <Transform Algorithm="http://www.w3.org/2001/10/xml-exc17#xslt" /> </Transforms> <DigestMethod Algorithm="http://www.w3.org/2001/05/xmldsig01#sha256"/> <DigestValue>PKNgWl1Tqr1D2YddYKA4a95XcNs</DigestValue> </Reference> </SignedInfo> <SignatureValue>OoAzRt70BLo2baxrQOoBXmuObrUMa</SignatureValue> </Signature></pre>												
type	xs:anyType												
properties	isRef 0 minOcc 0 maxOcc 1 content complex mixed true												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation						
Name	Type	Use	Default	Fixed	annotation								

7.4 ApplicationResponse

ApplicationResponse is a wrapper for transported data (Payload) to enable XML digital signature to any file type, whether it is XML, ASCII or binary format. The Payload is inserted in one element, called Content. It is base64-coded to make it loose from ApplicationResponse schema. ApplicationResponse has some supporting elements for information purposes, like returned Customer ID, TimeStamp, ResponseCode etc.

The following describes each element in ApplicationResponse, and how it is used when sent from Nordea bank.


element ApplicationResponse




element ApplicationResponse/CustomerId

diagram	bxd:CustomerId
Nordea rule: Mandatory	Example: 1234567890
type	restriction of xs:string
properties	isRef 0 content simple nillable false
facets	minLength 1 maxLength 16


element ApplicationResponse/Timestamp

diagram	 bxid:Timestamp
Nordea rule: Mandatory	Example: 2008-06-03T14:45:35.424+03:00
type	xs:dateTime
properties	isRef 0 content simple


element ApplicationResponse/ResponseCode

diagram	 bxid:ResponseCode
Nordea rule: Mandatory	Example: 00
type	restriction of xs:string
properties	isRef 0 content simple nillable false
facets	minLength 1 maxLength 16

element ApplicationResponse/ResponseText

diagram	 bxid:ResponseText
Nordea rule: Mandatory	Example: OK
type	restriction of xs:string
properties	isRef 0 content simple nillable false
facets	minLength 1 maxLength 80

element ApplicationResponse/ExecutionSerial

diagram	 bxid:ExecutionSerial
Nordea rule: Optional	Example: WS-000125
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 32

element ApplicationResponse/Encrypted

diagram	
Nordea rule: Optional	Example: false
type	xs:boolean
properties	isRef 0 minOcc 0 maxOcc 1 content simple

element ApplicationResponse/EncryptionMethod

diagram	
Nordea rule: Not used	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 35

element ApplicationResponse/Compressed

diagram	
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 35

element ApplicationResponse/CompressionMethod

diagram	
Nordea rule: Not used	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 35

element ApplicationResponse/AmountTotal

diagram	
Nordea rule: Optional	
type	xs:double
properties	isRef 0 minOcc 0 maxOcc 1 content simple

element ApplicationResponse/TransactionCount

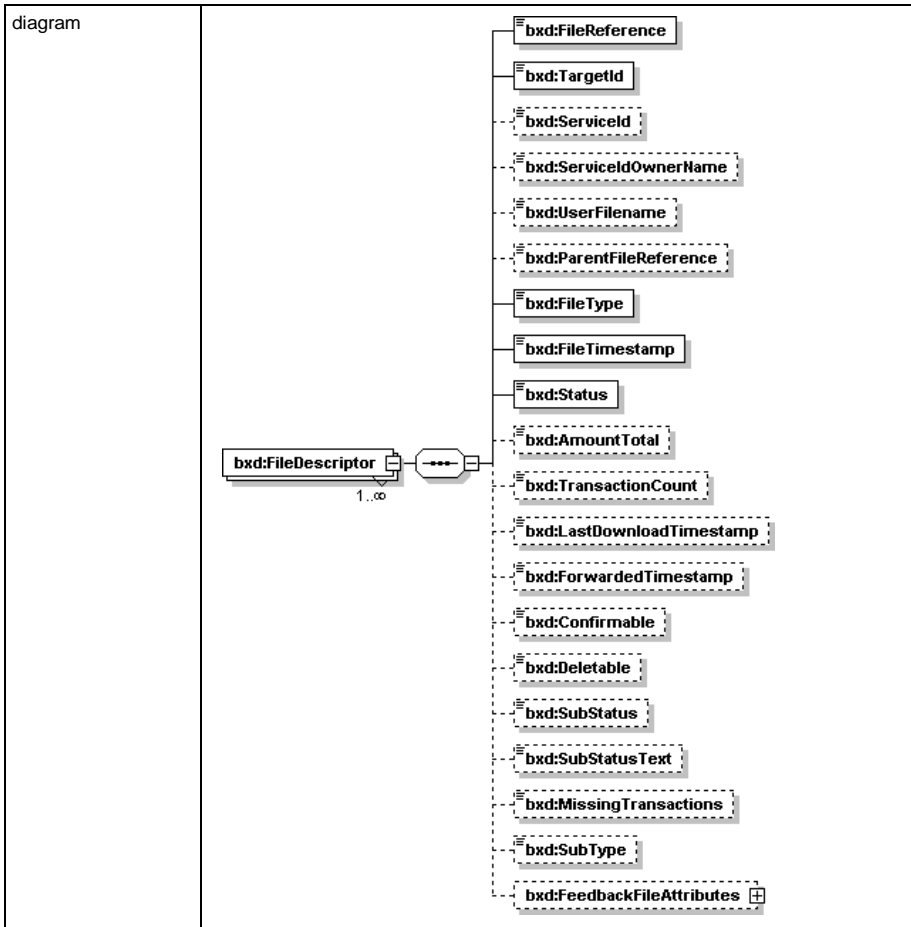
diagram	
Nordea rule: Optional	
type	xs:long
properties	isRef 0 minOcc 0 maxOcc 1 content simple

element ApplicationResponse/FileDescriptors

diagram	
---------	--

1..∞


element ApplicationResponse/FileDescriptors/FileDescriptor




element ApplicationResponse/FileDescriptors/FileDescriptor/FileReference

diagram	
Nordea rule: Mandatory	Example: 00001500922006030316603001006827
type	restriction of xs:string
properties	isRef 0 content simple
facets	minLength 1 maxLength 32


element ApplicationResponse/FileDescriptors/FileDescriptor/TargetId

diagram	 bxid:TargetId
Nordea rule: Mandatory	Example: 0012345678:
type	restriction of xs:string
properties	isRef 0 content simple
facets	minLength 1 maxLength 80


element ApplicationResponse/FileDescriptors/FileDescriptor/ServiceId

diagram	 bxid:ServiceId
Nordea rule: Mandatory	Example: 0012345678:
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 256


element ApplicationResponse/FileDescriptors/FileDescriptor/ServiceIdOwnerName

diagram	 bxid:ServiceIdOwnerName
Nordea rule: Optional	Example: WEB SERVICE TEST CUSTOMER V101
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 256


element ApplicationResponse/FileDescriptors/FileDescriptor/UserFilename

diagram	 bxid:UserFilename
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 80


element ApplicationResponse/FileDescriptors/FileDescriptor/ParentFileReference

diagram	 bxid:ParentFileReference
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 16


element ApplicationResponse/FileDescriptors/FileDescriptor/FileType

diagram	 bxid:FileType
Nordea rule: Mandatory	
type	restriction of xs:string
properties	isRef 0 content simple
facets	minLength 1 maxLength 40


element ApplicationResponse/FileDescriptors/FileDescriptor/FileTimestamp

diagram	 bxid:FileTimestamp
Nordea rule: Mandatory	Example:: 2006-03-03T00:00:00.0Z
type	xs:dateTime
properties	isRef 0 content simple

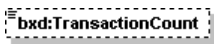
element ApplicationResponse/FileDescriptors/FileDescriptor/Status

diagram	 bxid:Status
Nordea rule: Mandatory	Example:10
type	restriction of xs:string
properties	isRef 0 content simple
facets	minLength 1 maxLength 10


element ApplicationResponse/FileDescriptors/FileDescriptor/AmountTotal

diagram	 bxid:AmountTotal
Nordea rule: Optional	
type	xs:double
properties	isRef 0 minOcc 0 maxOcc 1 content simple


element ApplicationResponse/FileDescriptors/FileDescriptor/TransactionCount

diagram	
Nordea rule: Optional	
type	xs:long
properties	isRef 0 minOcc 0 maxOcc 1 content simple


element ApplicationResponse/FileDescriptors/FileDescriptor/LastDownloadTimestamp

diagram	
Nordea rule: Optional	
type	xs:dateTime
properties	isRef 0 minOcc 0 maxOcc 1 content simple


element ApplicationResponse/FileDescriptors/FileDescriptor/ForwardedTimestamp

diagram	
Nordea rule: Optional	
type	xs:dateTime
properties	isRef 0 minOcc 0 maxOcc 1 content simple

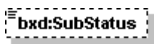
element ApplicationResponse/FileDescriptors/FileDescriptor/Confirmable

diagram	
Nordea rule: Optional	
type	xs:boolean
properties	isRef 0 minOcc 0 maxOcc 1 content simple


element ApplicationResponse/FileDescriptors/FileDescriptor/Deletable

diagram	
Nordea rule: Optional	
type	xs:boolean
properties	isRef 0 minOcc 0 maxOcc 1 content simple


element ApplicationResponse/FileDescriptors/FileDescriptor/SubStatus

diagram	
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 35

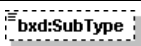
element ApplicationResponse/FileDescriptors/FileDescriptor/SubStatusText

diagram	
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 70

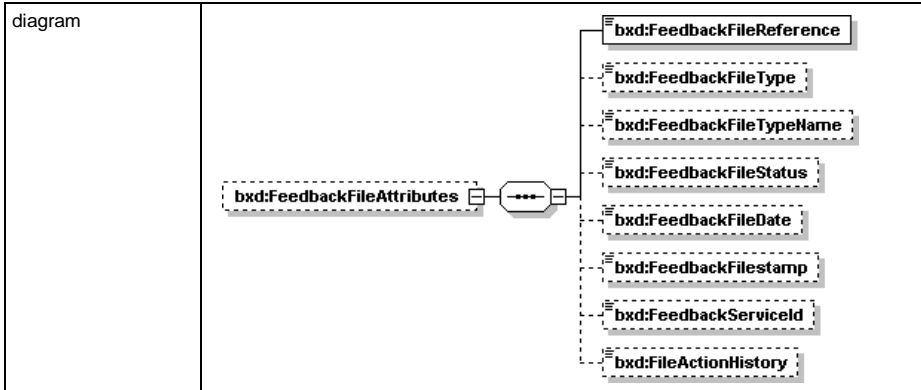
element ApplicationResponse/FileDescriptors/FileDescriptor/MissingTransactions

diagram	
Nordea rule: Optional	
type	xs:boolean
properties	isRef 0 minOcc 0 maxOcc 1 content simple

element ApplicationResponse/FileDescriptors/FileDescriptor/SubType

diagram	
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 35

element ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes



element ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FeedbackFileReference

diagram	
Nordea rule:	Mandatory
type	restriction of xs:string
properties	isRef 0 content simple
facets	minLength 1 maxLength 16


element ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FeedbackFileType

diagram	
Nordea rule:	Optional
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 35


element ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FeedbackFileTypeName

diagram	
Nordea rule:	Optional
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 80


element `ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FeedbackFileStatus`

diagram	 <code>bxid:FeedbackFileStatus</code>
Nordea rule: Optional	
type	restriction of <code>xs:string</code>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 16


element `ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FeedbackFileDate`

diagram	 <code>bxid:FeedbackFileDate</code>
Nordea rule: Optional	
type	<code>xs:date</code>
properties	isRef 0 minOcc 0 maxOcc 1 content simple

element `ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FeedbackFilestamp`

diagram	 <code>bxid:FeedbackFilestamp</code>
Nordea rule: Optional	
type	<code>xs:dateTime</code>
properties	isRef 0 minOcc 0 maxOcc 1 content simple

element `ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FeedbackServiceId`

diagram	 <code>bxid:FeedbackServiceId</code>
Nordea rule: Optional	
type	restriction of <code>xs:string</code>
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 35

element ApplicationResponse/FileDescriptors/FileDescriptor/FeedbackFileAttributes/FileActionHistory

diagram	
Nordea rule:	
Nordea rule:	Optional
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 1 maxLength 16

element ApplicationResponse/CustomerExtension

diagram													
Nordea rule:													
Nordea rule:	Optional												
type	xs:anyType												
properties	isRef min(0 0 maxOcc 1 content complex mixed true												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	annotation						
Name	Type	Use	Default	Fixed	annotation								

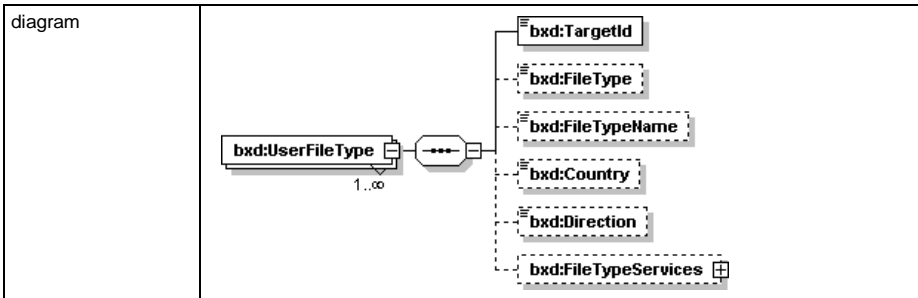
element ApplicationResponse/FileType

diagram	
Nordea rule:	
Nordea rule:	Mandatory
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1

element ApplicationResponse/UserFileTypes

diagram	
---------	--

element ApplicationResponse/UserFileTypes/UserFileType



element ApplicationResponse/UserFileTypes/UserFileType/TargetId

diagram	
Nordea rule:	Mandatory
type	restriction of xs:string
properties	isRef 0 content simple
facets	maxLength 80

element ApplicationResponse/UserFileTypes/UserFileType/FileType

diagram	
Nordea rule:	Optional
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	maxLength 35

element ApplicationResponse/UserFileTypes/UserFileType/FileTypeName

diagram	
Nordea rule:	Optional
type	restriction of xs:string
properties	isRef 0 minOcc 0 maxOcc 1 content simple
facets	minLength 0 maxLength 80

element ApplicationResponse/UserFileTypes/UserFileType/Country

diagram	
Nordea rule:	Optional
properties	isRef 0 minOcc 0 maxOcc 1

element ApplicationResponse/UserFileTypes/UserFileType/Direction

diagram	
Nordea rule:	Optional
properties	isRef 0 minOcc 0 maxOcc 1

element ApplicationResponse/UserFileTypes/UserFileType/FileTypeServices

diagram	
---------	--

element ApplicationResponse/UserFileTypes/UserFileType/FileTypeServices/FileTypeService


diagram	
---------	--

element ApplicationResponse/UserFileTypes/UserFileType/FileTypeServices/FileTypeService/ServiceId


diagram	
Nordea rule:	Mandatory
type	restriction of xs:string
properties	isRef 0 content simple
facets	maxLength 60

element


ApplicationResponse/UserFileTypes/UserFileType/FileTypeServices/FileTypeService/ServiceIdOwnerName

diagram	 bxid:ServiceIdOwnerName
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 max'occ 1 content simple
facets	maxLength 256


element ApplicationResponse/UserFileTypes/UserFileType/FileTypeServices/FileTypeService/ServiceIdType

diagram	 bxid:ServiceIdType
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 max'occ 1 content simple
facets	maxLength 80

element ApplicationResponse/UserFileTypes/UserFileType/FileTypeServices/FileTypeService/ServiceIdText

diagram	 bxid:ServiceIdText
Nordea rule: Optional	
type	restriction of xs:string
properties	isRef 0 minOcc 0 max'occ 1 content simple
facets	maxLength 80

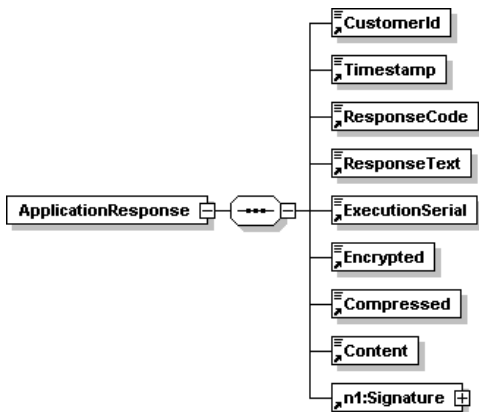
element ApplicationResponse/Content

diagram	 bxid:Content
Nordea rule: Mandatory	The actual content, payload in the DownloadFile operation. The file is in Base64 for
type	xs:base64Binary
properties	isRef 0 minOcc 0 max'occ 1 content simple nillable false

element ApplicationResponse/Signature

diagram											
Nordea rule:	Mandatory										
type	xsd:anyType										
properties	<table> <tr><td>isRef</td><td>0</td></tr> <tr><td>minOcc</td><td>0</td></tr> <tr><td>maxOcc</td><td>1</td></tr> <tr><td>content</td><td>complex</td></tr> <tr><td>mixed</td><td>true</td></tr> </table>	isRef	0	minOcc	0	maxOcc	1	content	complex	mixed	true
isRef	0										
minOcc	0										
maxOcc	1										
content	complex										
mixed	true										

Sample: DownloadFile:



7.4.1 Error codes

The response code is given by the bank to indicate the result of the requested operation. The codes are:

Code	Name	Remarks
00	OK.	
01	Pending.	not used
02	SOAP signature error.	signature verification failed
03	SOAP signature error.	certificate not valid for this id
04	SOAP signature error.	certificate not valid
05	Operation unknown.	
06	Operation is restricted.	
07	SenderID not found.	
08	SenderID locked.	
09	Contract locked.	
10	SenderID outdated	
11	Contract outdated	
12	Schema validation failed.	
13	CustomerID not found.	
14	CustomerID locked.	
15	CustomerID outdated.	
16	Product contract outdated.	
17	Product contract locked.	
18	Content digital signature not valid	
19	Content certificate not valid.	
20	Content type not valid.	
21	Deflate error.	
22	Decrypt error.	
23	Content processing error.	
24	Content not found.	
25	Content not allowed.	
26	Technical error.	
27	Cannot be deleted.	
28	[not used]	not used
29	Invalid parameters.	
30	Authentication failed.	
31	Duplicate message rejected.	SOAP.Body.RequestHeader.SenderId + SOAP.Body.RequestHeader.RequestId
32	Duplicate application request rejected. (In certificate download: GENERAL ERROR)	
33	Error in customer information	See name in Error text
34	Contract not found / error in MA	
35	Authorization failed	
36	Technical error, contact bank hel	

7.5 SOAP envelope

SOAP envelope is a standard message format used to send and receive requests/responses in Web Services communication, and follows WS-I recommendations. It is always digitally signed with certificate for authority to communicate to bank with Web Services. SOAP messages are generated by using any WS-I compliant tool like Java and Microsoft DotNet.

It has two parts: **SOAP:Header** and **SOAP:Body**. Please, see the example below. The SOAP:Header part contains all the information regarding security and signatures. The SOAP:Body part can be divided further in two: RequestHeader and ApplicationRequest.

The ApplicationRequest has been explained detailed above. The field contains the ApplicationRequest XML-structure in base64 coded format, including the payload in any format.

The RequestHeader contains information about the sender, like: SenderId, RequestId, Timestamp, Language, UserAgent and ReceiverId. Those fields are described below.

When SOAP message has been received by Nordea, first the validity of the Sender certificate is checked. If OK then the ApplicationRequest will be extracted and processed further. If the validity of the sender certificate is not OK, then a SOAP fault -message is sent and ApplicationRequest will not be processed.

The following describes each element in SOAP:Body/RequestHeader/ResponseHeader, and how it is used when uploaded/downloaded to/from the bank.

RequestHeader:

SenderId: The unique identification of the sender of this request message. User ID for the used certificate. The message sender can be a 3rd party service bureau. This identification is issued and managed by the bank. The SenderId identity is authenticated by the digital signature in the SOAP:Header.

RequestId: The unique identification for this request. This unique ID is copied to the ResponseHeader. This value must be unique for three months.

Timestamp: Time and date when the request was sent. ISODateTime, if no time zone specified, UTC time zone is assumed.

Language: Language attribute is used to request language version for certain information in human readable format. One of the following codes must be used: EN, SV or FI (not used currently).

UserAgent: The name and version of the software which was used to send this request.

ReceiverId: Identification of the receiver of this request message

ResponseHeader:

SenderId: The unique identification of the sender of the original request message for this response

RequestId: The unique identification copied from the original request for this response.

Timestamp: Time and date when the response was sent, ISODateTime

ResponseCode: The code is used to indicate the file delivery condition. The codes are indicated in this *Web Services Security and Communication Description*.

ResponseText: The textual explanation of the condition.

ReceiverId: Identification of the receiver of the original request message for this response (the sender of this response)

7.6 Examples of SOAPs and ApplicationRequest/ApplicationResponse

SOAP request example (part of the XML):

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
S:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd" wsu:Id="XWSSGID-1641978992783830948719">
        <wsu:Created>2022-01-12T09:16:32.733Z</wsu:Created>
        <wsu:Expires>2022-01-12T09:21:32.733Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-
token-profile-1.0#X509v3" wsu:Id="XWSSGID-1641978992675-
1475305108">MIIC+zCAeOgAwIBAgIDASxVMA0GCSqGSIb3DQEBBQUAMGQxCzAJBgNVBAYTAiFNM
R4wHAYDVQQKEgVOb3JkZWVgQmFuayBBOiAocHVibCkxHzAdBgNVBAMTFk5vcmlRiYSDb3Jwb3JhdGU
gZmljYXRIRMRmEQYDVQQFEwo1NzgWODYwMjM4MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQCClmzyaVAEr2cT5gGPxuiMxZSJZRIDHUwyUMlagn/YbKcQ/MhumUEDDAKMAgGBiqFecBAzATBgNV
HSMEDDAKGAhAC3XW288LpzAOBgNVHQ8BAf8EBAMCBAAwDQYJKoZIhvcNAQEFBQADggEBADVuzhr
4KJwDXHph5fm5BOqufA10fnUP5rFYfpDz3gRbyicRcBFj2hkIG+8wMUKiTfASheRYL1hydk4EIJ3gaeieD7Yn9O
xMILy+svh3YXGWnw9z9msRRyYJdVNLwss2sUgxlV66iPJR0qVIT55fLed9YXbfdb/tPE-g10Qw62kXyZkDN0xe
I8IUuihFLX20H/SPARRHCAootUoNxuzFuEH/5zL3FMBWSSxdkfGrmqzF8/C5a31GXWggn/JK7KI1BYKx/weV
WRui0F15nfJQ=</wsse:BinarySecurityToken>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="XWSSGID-
1641978992673254148581">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <InclusiveNamespaces xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="wsse S
SOAP-ENV"/>
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
          <ds:Reference URI="#XWSSGID-1641978992781-412569123">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <InclusiveNamespaces xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="S
SOAP-ENV ns2"/>
              </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256"/>
            <ds:DigestValue>2hDXWPwOTrMIXCksTkj4ISZwrligdYtQbafUr4=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#XWSSGID-1641978992783830948719">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <InclusiveNamespaces xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="wsu
wsse S SOAP-ENV"/>
              </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256"/>
            <ds:DigestValue>pqwOmR08W2V5KEgsh3+fFmu94EHukEk=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>MrQNHTalxYFm5FukMlrhb+XaOL+zk1615+PsUHju9Nco/YQ&#13;
SexTD/5LN6q7OMdpJ2dhA3xtEtU5LOz4geQ6Gh1DycI6WuLxce3cNDpGL7gPUhw&#13;
TBBnyl+OhA/yIJeLfmA=</ds:SignatureValue>
      <ds:KeyInfo>
        <wsse:SecurityTokenReference xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1641978992756-860740920">
          <wsse:Reference URI="#XWSSGID-1641978992675-1475305108" ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
        </wsse:SecurityTokenReference>
      </ds:KeyInfo>
    </ds:Signature>
  </wsse:Security>
</SOAP-ENV:Header>

```



```
<S:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="XWSSGID-1641978992781-412569123">
  <ns2:downloadFileListin xmlns="http://model.bxd.fi" xmlns:ns2="http://bxd.fi/CorporateFileService">
    <RequestHeader>
      <SenderId>1205585055</SenderId>
      <RequestId>1</RequestId>
      <Timestamp>2022-01-12T11:13:50.907+02:00</Timestamp>
      <Language>FI</Language>
      <UserAgent>NEA</UserAgent>
      <ReceiverId>123</ReceiverId>
    <ApplicationRequest>PD94bWwgdMvyc2lvcj0iMS4wliB1bmNvZGluZz0iVVRGLTgiPz48QXBwbGljYXRpb25S
ZXF1ZXN0IHhtbG5zPSJodHRwOi8vYnNhkLmZpL3htbGRhdGEvIj4NCiAgICA8Q3VzdG9tZXJJZD4xMTEzMTE
xMTwvQ3VzdG9tZXJJZD4NCiAgICA8Q29tbWwvZD5E3dubG9hZEEZpbGVMaXN0PC9Db21tYW5kPg0KICAg
IDxUaW1lc3RhbnhA+
jZpUEpSMHFWSVQ1NWZMZWQ5WVhZiYjMTM7DQpkYi90UEUrzZzEwUXc2MmtYeVo5MzROS0xVWVHZA
eGdBaWlwWXSd0FWTHNsYmcwZThQT0tCeFQ1MjdmNHRXRHc1cGVpS3A3bHRzRUNmJiMxMzsNCkYjY
XZjMGEtM94ZUk4bFV1aWhGTFgyMEgvU1BhUIJIQ0Fvb3RVb054dXpGSXVFSWwvNXpMM0ZnQldTc3hka
2ZHcnFtekY4L0M1YTmmlzEzOw0KMUdYV0dnbi9KSzdLSTFCWUt4L3dlVldSdWkwRkk1bmZlSIE9PC9YNT
A5Q2VydGlmaWNhdGU+PFg1MDJc3N1ZXJlZD4xMTEzMTEwvQ3VzdG9tZXJJZD4xMTEzMTEwvQ3VzdG9t
NTMxMzYzNDMwMzYyZDMwMzEzMjMwLENOPU5vcmlR1YSBDb3Jwb3JhdGUgQ0EgMDEsTz1Ob3JkZWVh
QmFuayBBQiAocHVibCksQz1TRTwwWDUwOUlzc3Vlck5hbWU+PFg1MDITZlJpYWxOdW1iZXI+NmY4ODU
8L1g1MDITZlJpYWxOdW1iZXI+PC9YNTA5S3NzdWV5U2VyaWFsPjwvWDUwOURhdGE+PC9LZXIjbmZv
PjwvU2lnbmF0dXJlPjwvQXBwbGljYXRpb25SZXF1ZXN0Pg==</ApplicationRequest>
  </ns2:downloadFileListin>
</S:Body>
</S:Envelope>
```

SOAP response example (part of the XML, without signature):

```
<soapenv:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cor="http://bxd.fi/CorporateFileService" xmlns:mod="http://model.bxd.fi"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" soapenv:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Timestamp-a0f4d549-cca6-4a1c-bbf9-1672976fda29">
        <wsu:Created>2022-01-03T10:18:19Z</wsu:Created>
        <wsu:Expires>2022-01-03T10:23:19Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" wsu:Id="SecurityToken-7c9e3edc-6cad-41c1-8585-1672976fbafd">MIIDizCCAnOgAwIBAgIDASP+
AwEAAaM1MDMwCQYDVROTBAlwADARBgNVHQ4ECgQIQrtcU9YxVbUwEwYDVR0jBAwwCoAIRMeT2
dl7VsEwDQYJKoZIhvcNAQELBQADggEBBAKD2ilB86c/lmtXkAqjAzZMipRVSN1/ZZOtmBuf56ziUQQMLO
RipMjlyP9grDhJIIQqhsL1jo+BSdceW0o8U5bhrNeC6roLnGT864Ozj4tS354sgE5UUkCjYZUiu70PGP9
gJKlp6kLFTIKNhJ0oI8ealtbU5A6c7KXgou44SfaN12l4m/wBZY1147PQa8X5B1npHruVbhbbhQccVurmF3q
jpLHFkslnBI39tJ4taYwBqRe81pR3S6p2Hm9lWFK5nU3Ef/LleeFsWVCpfr2X0BOB0gFbLLOPFjC+FfQvLZ
2A4l4-gjL321mVz7sp4zVZ86oKXolrTzexg+E=</wsse:BinarySecurityToken>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
          <Reference URI="#Timestamp-a0f4d549-cca6-4a1c-bbf9-1672976fda29">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmenc#sha256" />
            <DigestValue>U9cNMMytdSuDpQ1SU9WU++I0UTWrijmTELDTcsfkitzU=</DigestValue>
          </Reference>
          <Reference URI="#Body-e4b05d21-45c7-4dfb-ba48-1672976ff6ca">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmenc#sha256" />
            <DigestValue>Oif1RYFdcq4rnsfmzf00khrF800Ye7A6Pv2HMBFxs=</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>pmQ1E5zlpwRB5JLliUoa7mHyolZuoG5o5Vf9dqHmfemBu1qCobe0saMC4SWZLaShvV
F9rlg+i4mM+t2L07X9jhoprpyH1O3ZoVDQ5DHRnnBlJJxy2X9N3m/q2zUjVp2mEKp8VT1pTDbGtfMPqPV
z5eq15SPVHsX3wLYpB4Qu+BYqjH4ZBQ8M6ybF3HObqH80vz8n3tWK53vY6OobBu9itFbywvSGL163ZU
xH4BtjvOk0S2gB0bSecudSKLpRxcOYSxTjcZJbno4bsohwRnaHKZvr4TJhKH1eMQx/TdEprSrxWdvoUlll
wV71LJ0MOW=</SignatureValue>
      </KeyInfo>
      <wsse:SecurityTokenReference xmlns="">
        <wsse:Reference URI="#SecurityToken-7c9e3edc-6cad-41c1-8585-1672976fbafd"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
      </wsse:SecurityTokenReference>
    </KeyInfo>
  </Signature>
</wsse:Security>
</soapenv:Header>
<soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Body-e4b05d21-45c7-4dfb-ba48-1672976ff6ca">
  <cor:getUserInfoout>
    <mod:ResponseHeader>
      <mod:SenderId>1205585055</mod:SenderId>
      <mod:RequestId>1</mod:RequestId>
      <mod:Timestamp>2022-01-03T11:18:18+01:00</mod:Timestamp>
      <mod:ResponseCode>00</mod:ResponseCode>
      <mod:ResponseText>OK.</mod:ResponseText>
      <mod:ReceiverId>123</mod:ReceiverId>
    </mod:ResponseHeader>
  </cor:getUserInfoout>
</soapenv:Body>
</Envelope>
```


ApplicationRequest example (part of the XML):

```
<ApplicationRequest xmlns="http://bxd.fi/xmldata/">
  <CustomerId>11111111</CustomerId>
  <Command>DownloadFileList</Command>
  <Timestamp>2022-01-12T11:13:50.907+02:00</Timestamp>
  <Status>ALL</Status>
  <Environment>PRODUCTION</Environment>
  <SoftwareId>NordeaTest</SoftwareId>
  <FileType>VKEUR</FileType>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xm1-c14n-20010315#WithComments"/>
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xm1enc#sha256"/>
        <DigestValue>k9HODYPZaWAoM3GCLTbodlCPzZxsH+OLKb7kAEc4yv4=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>MvLEc8SsHIZ45dsQ8cBWWJo2QqshIAAGidEPgJ26mNNv2GJLAY1i9URg2Kawf9/flk7a/45T4uMwUtVcVw5useEo=</SignatureValue>
    <KeyInfo>
      <X509Data>
        <X509Certificate>MIIC9DCCAdygAwIBAgICP3UwDQYJKoZIhvcNAQELBQAwaTELMAkGA1UEBhMCUOUxHjAcBgNVBAoT
        FU5vcmlRiYSBCYW5rIEFCIChdWJsKTEkMCIGA1UEAxMmTm9yZGVhIFRlc3QgQ29ycG9yYXRIENB
        5mecJ0qsNc0GI2BtWoZ1FBlwR8huZ5u6yFwOV2UNWGk0msFS11DyLxRuLMpXcdhaaBznB5RvpLIC
        UcDAqpYXrk+nYtHmJJFSCsl6WA==</X509Certificate>
        <X509IssuerSerial>
          <X509IssuerName>2.5.4.5=#130b3531363430362d30313230,CN=Nordea Corporate CA
          01,O=Nordea Bank AB (publ),C=SE</X509IssuerName>
          <X509SerialNumber>76885</X509SerialNumber>
        </X509IssuerSerial>
      </X509Data>
    </KeyInfo>
  </Signature>
</ApplicationRequest>
```

ApplicationResponse example (part of the XML):

```
<c2b:ApplicationResponse xmlns:c2b="http://bx.d.fi/xmldata/">
  <c2b:CustomerId>1753419215</c2b:CustomerId>
  <c2b:Timestamp>2022-01-03T11:18:01:00</c2b:Timestamp>
  <c2b:ResponseCode>00</c2b:ResponseCode>
  <c2b:ResponseText>OK.</c2b:ResponseText>
  <c2b:Encrypted>>false</c2b:Encrypted>
  <c2b:Compressed>>false</c2b:Compressed>
  <c2b:UserFileTypes>
    <c2b:UserFileType>
      <c2b:TargetId>0000602153</c2b:TargetId>
      <c2b:FileType>INFO</c2b:FileType>
      <c2b:FileName>Info</c2b:FileName>
      <c2b:Country>FI</c2b:Country>
      <c2b:Direction>Download</c2b:Direction>
      <c2b:FileTypesServices>
        <c2b:FileTypeService>
          <c2b:ServiceId/>
          <c2b:ServiceIdOwnerName/>
          <c2b:ServiceIdType>N</c2b:ServiceIdType>
          <c2b:ServiceIdText/>
        </c2b:FileTypeService>
      </c2b:FileTypesServices>
    </c2b:UserFileType>
  </c2b:UserFileTypes>
  <c2b:UserFileType>
    <c2b:TargetId>0000602153</c2b:TargetId>
    <c2b:FileType>VKEUR</c2b:FileType>
    <c2b:FileName>Rates of exchange</c2b:FileName>
    <c2b:Country>FI</c2b:Country>
    <c2b:Direction>Download</c2b:Direction>
    <c2b:FileTypesServices>
      <c2b:FileTypeService>
        <c2b:ServiceId/>
        <c2b:ServiceIdOwnerName/>
        <c2b:ServiceIdType>N</c2b:ServiceIdType>
        <c2b:ServiceIdText/>
      </c2b:FileTypeService>
    </c2b:FileTypesServices>
  </c2b:UserFileType>
</c2b:UserFileTypes>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
</SignedInfo>
<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xm1-c14n-20010315"/>
<SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
<Reference URI="">
  <Transforms>
    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    <Transform Algorithm="http://www.w3.org/TR/2001/REC-xm1-c14n-20010315"/>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2001/04/xm1enc#sha256"/>
  <DigestValue>Vs/PeGRlIXx/qSWirbhchY3REYse6HyvjaF6kl=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>qG54/FBZeoOP0Kk470/b0Gz51m/SHgQoi0IwVtkn3s57v3PV3EsxkRRjMdal1InZn4K
Hon9Ae6mgdHS5wJbj/OV64jINevao5+gMjLaG2qFzklVWpGgZiEweGyMWQ7X4Bilw0BtUnWkK+zlsrj+3d
Jkp95OcRvXnUXX1dDYp83eup5IF+zMx3rqghg7xUMh1honXRtlRm+R24wNUPCYDKsoeh4AUfZmpQO9
LHgKLL0GZXXK20G+LGWhOAo+gcNe/gfMmVkGeMCQtimx/T6mejnVv0BqB0XlCsEclVv/2zIFeFJHC9eiL
Dc3DxC5QtW2loGWKK8qHFLuuvlRQ==</SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509Certificate>MIIDizCCAnOgAwIBAgIDASP+MA0GCSqGSIb3DQEBCwUAMGsxGzA1BjBGNVBAZTAI
NFMR4wHAYDVQQKEExvob3JkZWEGQmFuayBBQIACOCkxJjBGNVBAZTA1BjBGNVBAZTA1BjBGNVBAZTA1
hdGUgU2VydmljZmVzZDCCASlwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAPJE8EzsdUL4vg/01
0yMjExMjcxMjZNTBaMG4xCzA1BjBGNVBAZTA1AFMRgwFgYDVQKDA9Ob3JkZWEGQmFuayBBYyNAXG
zAZBGNVBAZTA1BjBGNVBAZTA1BjBGNVBAZTA1BjBGNVBAZTA1BjBGNVBAZTA1BjBGNVBAZTA1BjBGNV
2VydmljZmVzZDCCASlwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAPJE8EzsdUL4vg/01
xmhDWDnYhP0jsWrpWkXD8RPM2XmVhslNHriZNEXC15AUzep27fVf3Y7ujbu+vJCU7ECawEAAm1
MDMwCQYDVRTBA1wADARBgNVHQ4ECCgQIQRtcU9YxVbUwEwYDVROjBAwwCoAIRMeT2dl7VsEwDQ
YJKoZIhvcNAQELBQADggEBAKD2tlB86c/lmkG2ntXkAqiAzZMipRVSN1/ZZkOtmBuf56ziUQQMLORipMjI
yP9grDhJlIQqhsL1jo+GSDBSdceW0o8Uj5bhrgGT864Ozj4tS354sgE5UUKcUYZUiw70PGP9gJKlp6kLF
TIKnhJ0i8ealtU5A6c7KXgou44SfaN12l4m/wBZY1147PQa8X5B1npHruVbhbbhQccVurnF3qjpLHFksln
```

```
BI39tJ4taYwBqRe81pR3S6p2Hm9IWFK5nU3Ef/LleeFsWVCpfR2x0BOB0gFbL0PFjC+FfQvLZ2A4I4+gjL
321mVz7sp4zVZ86oKXolrTzexg+E=</X509Certificate>
  <X509IssuerSerial>
    <X509IssuerName>serialNumber=516406-0120, CN=Nordea Corporate Server CA 01,
O=Nordea Bank AB (publ), C=SE</X509IssuerName>
    <X509SerialNumber>74750</X509SerialNumber>
  </X509IssuerSerial>
</X509Data>
</KeyInfo>
</Signature>
</c2b:ApplicationResponse>
```

8 Customer support

8.1 Customer support – Finland

- **E-support for Corporate Customers, tel 0200 67210 (in Finnish)**
 - Open on banking days 8.00–17.00
 - on short banking days* 8.00–14.00
- **E-support for Corporate Customers, tel 0200 67220 (in Swedish)**
 - Open on banking days 9.00–16.30
 - on short banking days* 9.00–14.00
- **E-support for Corporate Customers, tel 0200 67230 (in English)**
 - Open on banking days 9.00–17.00
 - on short banking days* 9.00–14.00

Calls are charged at the local network charge/mobile call charge; no separate service charges.

* New Year's Eve and Maundy Thursday

8.2 Corporate eGateway Support

- **+46 771 77 69 75**
 - E-mail: egatewaysupport@nordea.com

9 Additional information

More detailed information on the service is available at www.nordea.fi website: *Corporate customers >> Payments >> Web Services*

Test messages, instructions and testing tools for software houses, vendors are available at www.nordea.fi website: *Corporate customers >> Payments >> Web Services >> Instructions and sample files >> Testing >> Web Services >> Testing*