Sterling B2B Integrator

# EBICS Banking Server Concepts

*Version 5.2.5*

Sterling B2B Integrator

# EBICS Banking Server Concepts

*Version 5.2.5*

**Note**
Before using this information and the product it supports, read the information in "Notices" on page 33.

# Contents

# Chapter 1. Overview of EBICS Banking Server

Electronic Banking Internet Communication Standard (EBICS) is an Internet-based communication and security standard that is primarily used for remote data transfer between your organization and a bank for corporate payment transactions.

EBICS allows data file exchange independent of message standards and formats. EBICS uses established digital signature and encryption procedures. Its features are based on international standards for internet communication and improved security, for example, XML, HTTPS, TLS, and SSL. EBICS also has multibank capability wherein the corporate clients in the countries that have adopted EBICS can transact with any bank in those countries using the same software.

A range of prerequisites must be fulfilled by a user (associated with a partner) to be able to implement bank-technical EBICS transactions with a particular bank. The basic prerequisite to implement EBICS transactions is the signing of a contract between the partner and the bank. The following details are agreed upon in this contract:

- The nature of business transactions (bank-technical order types) the partner will conduct with the bank
- Information about the user's bank accounts
- The partner's users working with the banks system
- The authorizations and permissions the user's possess

The partner receives the banks access data (bank parameters) after the contract is signed. The bank sets up the partner and user master data in the bank system in accordance with the contractual agreements.

Other prerequisites are successful subscriber initialization, downloading of the bank's public certificates by the user, and successful verification of the user's public certificates by the bank.

The Sterling B2B Integrator EBICS Banking Server is a complete EBICS solution involving a bank, a partner, and user management, certificate management, secure file transactions, error recovery, and reporting. Use Sterling B2B Integrator to send and receive EBICS transactions.

The Sterling B2B Integrator EBICS Banking Server supports EBICS Specification V2.5 for both French and German implementations.

Sterling File Gateway operates on the Sterling B2B Integrator platform, enabling secure file transfer between internal and external partners using either the same or different communication protocols, file naming conventions, and file formats. Sterling File Gateway supports the movement of large and high-volume file transfers, with visibility of file movement in a process-oriented and highly-scalable framework that alleviates file transfer challenges, such as protocol and file brokering, automation, and data security.

## File System Space Requirements for FDL Requests

Because the FDL order type uses the file system to store the payload, it is important to plan file system storage accordingly. A large FDL payload requires

about 6 times as much file space as the payload size itself. For example, a 5 GB payload requires over 30 GB of file space in Sterling B2B Integrator to process the request.

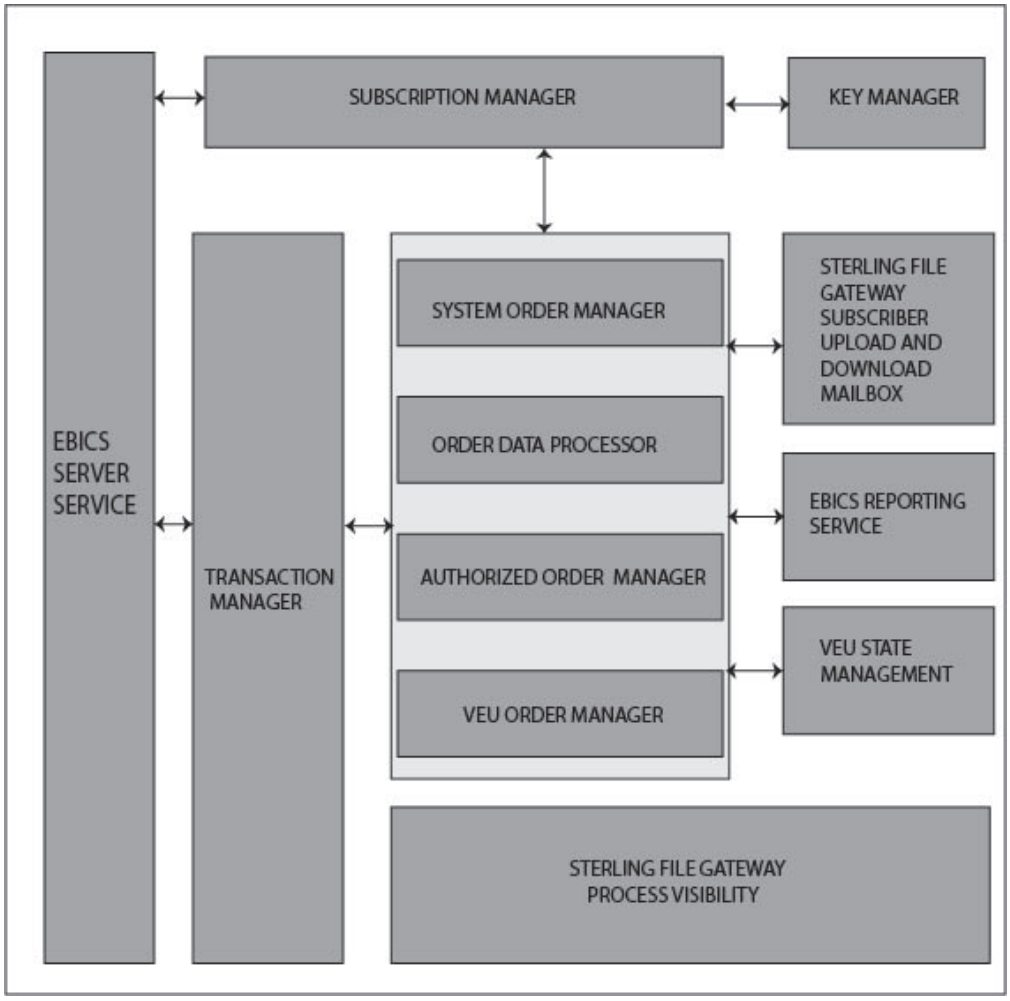
When using EBICS Banking Server in a cluster environment, you must configure the shared file system as document storage between nodes, even if the default document storage type is set to "Database". See the appropriate *Installation* documentation for instructions.

# Chapter 2. EBICS Banking Server Architecture

EBICS Banking Server enables you to transact with partners and users using EBICS.

Its features include creating and managing profiles (bank, partner, and user), associating partners and users with order types and file formats, assigning user permissions, creating and managing certificates, processing of order data, storing and retrieving profile information, certificates, and messages, managing message flows and transaction flows, transferring files using secure protocols, and so on.

The following diagram illustrates the EBICS Banking Server architecture:




*Subscription Manager* includes the following features:

- Profile Management - For creating and managing bank, partner, and user profiles
- Order Type Configuration - For configuring order types and file formats
- Offer Configuration - For grouping a set of order types and file formats to a list of customers

- User Permission Configuration - For assigning order types and file formats to users
- Import of Subscription Manager Information - For importing configuration details related to bank, partner, user, offer, user permissions, order types, and file formats into the EBICS Banking Server from an external repository
- Export of Subscription Manager Information - For exporting configuration details related to bank, partner, user, offer, user permissions, order types, and file formats into an external repository from the EBICS Banking Server

*Subscriber's upload and download mailboxes* are configured in Subscription Manager during the user subscription setup.

*Key Management* interfaces mainly with Subscription Manager to create, update, delete, and query certificates.

Key Management includes the following features:

- Self-Signed certificates - For generating and managing self-signed certificates using 2048-key length
- CA certificates - For managing CA certificates
- Key storage - For providing the key stores for the certificates and managing the renewal and expiration of certificates
- Import and Export certificates - For importing and exporting certificates
- Subscriber key validation - For validating user certificate hash values
- Certificate hash value - For supporting the creation of certificate hash value using SHA256

*EBICS Server Service* interfaces with Subscription Manager to retrieve the profile information of banks, partners, users, and order types necessary for verification and authentication of messages and transactions. It works in close collaboration with Transaction Manager to manage all the EBICS transactions.

EBICS Server Service includes the following features:

- Request and Response - For handling incoming EBICS requests (through HTTP and HTTPS) according to EBICS protocol specifications, and generating an appropriate response back to the requestor
- Message Flow - For managing the message flow for the initialization and file transfer phases of the EBICS transactions
- Authentication and Authorization - For performing message authentication and user authorization checks

*Transaction Manager* interfaces closely with the EBICS Server Service to manage the upload and download flow of system order types and bank-technical order types.

Transaction Manager includes the following features:

- Asynchronous Transaction - For managing the asynchronous transaction flow for upload bank-technical order type (FUL). It manages the authorized order processing flow in collaboration with the Order Data Processor to unpack the order data and deliver the unpacked order data to the destination upload mailbox as defined in the user profile settings.
- Synchronous Transaction - For managing the synchronous transaction flow for upload and download system order and bank-technical order types. It manages the system order processing, report processing (FDL, PSR) and download bank-technical order (FDL) processing flows.

- Segmentation and Recovery - For managing no-replay, segmentation, and error recovery

*System Order Manager* is responsible for updating and querying key management information and user referential information.

System Order Manager works closely with Transaction Manager and Subscription Manager to update and query the user's key certificates and referential information, and to download bank parameters and bank certificates.

*Authorized Order Manager* is responsible for initiating the Order Data Processor to unpack the order data received from the FUL order type request, routing the unpacked order data to the backend subscriber's upload mailbox, and renaming it according to a defined naming convention.

*VEU Order Manager* is responsible for handling VEU orders (order types HVD, HVE, HVS, HVT, HVU, or HVZ).

*Order Data Processor* is responsible for packing and unpacking order data. It interfaces with Subscription Manager and Transaction Manager to retrieve the relevant information required for packing and unpacking the order data. Its features include:

- Packing - For packing order data such as signing, compression, encryption, and base64 encoding depending on the requirement of the order type
- Unpacking - For unpacking order data such as verification, decompression, decryption, and base64 decoding depending on the requirement of the order type

*Reporting Service* is responsible for generating the Payment Status Report (PSR) associated with the unpacking of order data during an asynchronous upload of bank-technical order transaction flow.

*VEU State Management* is responsible for maintaining information regarding VEU orders which are not completely authorized (e.g have pending signatures).

*Sterling File Gateway* uses templates to describe how each EBICS transaction is interpreted to determine how and where it should be delivered and provides visibility into the details of the transfers for auditing and troubleshooting.

Sterling File Gateway includes the following features:

- File or File Name Transformations - For mapping input to output file names, system-wide, group, and partner-specific policies, common file processing tasks such as compression and decompression, PGP encryption and decryption, and signing
- File Transfer Visibility - Events are recorded for monitoring and reporting; detailed tracking for input-output file structure processing and dynamic route determination; ability to view and filter data flows for all users
- Broad Communications Protocol Support - FTP, FTP/S, SSH/SFTP, SSH/SCP, and Sterling Connect:Direct are supported upon installation, and additional protocols (such as AS2, AS3, or Odette FTP) can be configured using the extensibility feature

- Partner Interface (myFileGateway) - Web browser-based interface that enables partners to upload and download files, subscribe to notifications about events, manage passwords, search and view file transfer activity, and generate reports about file transfer activity
- Flexible Mailbox Structures - Ability to specify mailbox structures that leverage pattern-matching policies and specify attributes that must be true for all partners or a subset of partners
- Dynamic Routing - Consumer derived at run time, either through mailbox structure, file name, business process-derived consumer name, or map-derived consumer name

# Chapter 3. Managing Subscription Manager Information

The Subscription Manager menu in Sterling B2B Integrator enables you to:

- Create and manage bank, partner, and user profiles in the system database
- Create and manage offers
- Assign order types and file formats to an offer
- Assign permissions to users

A bank can have only one profile with a unique bank ID. A bank profile contains the following information:

- Unique ID of the bank

  **Note:** Each bank ID should have a unique port number.
- Name of the bank
- Address of the bank
- Public and private encryption, authentication and identification certificates
- HTTP URL of the bank
- EBICS protocol version

A bank can have multiple URLs. The corresponding bank URL is given to a user to send requests to the bank. The Uniform Resource Indicator (URI) is configured in the HTTP Server adapter to listen at the port and receive EBICS requests, if any.

The following versions of bank protocol and process types are supported:

- EBICS protocol version - H004, H003. H000
- Signature versions - A005, A006
- Authentication version - X002
- Encryption version - E002

Each partner can have one or more account information and partner IDs. You must specify the account number, either in national (German) or international (IBAN) format. You can associate a partner ID with an offer. The partner profile contains the following information:

- Unique ID of the partner
- Organization code of the partner
- Name of the partner
- Address of the partner
- Account ID and account holder's name
- Currency in which transaction is performed
- Account number
- Bank code

A user can be under one or more partners. A bank can create a user with or without associating a user with a partner. To enable exchange of EBICS messages between a partner and a user, you must associate a user ID with a partner ID.

A user transmits the public certificates to the bank through two independent communication paths:

- INI - Sends the public bank-technical key
- HIA - Sends the public identification and authentication key and the public encryption key

When a user is first assigned to a partner, the status of the user is New. If the user sends only the INI request to the corresponding bank, the status is changed to Partly Initialized (INI). If the user sends only the HIA request to the bank, the status is changed to Partly Initialized (HIA). After the user sends both the INI and HIA requests to the bank, the status is changed to Initialized. The user mails the initialization letters of the INI and HIA keys to the bank. When the bank receives the initialization letters pertaining to INI and HIA, it verifies the hash values in the certificates against its database. After successful verification, the status of the user is set to Ready, indicating that the user can now transact with the bank. The user then downloads the bank's public certificates by using the HPB system order type.

You can use the HKD and HTD order types to retrieve subscriber information stored by the bank after the user status is set to *Ready*.

Use the EBICS Subscription Manager Service to validate the keys on the INI and HIA initialization letters. On successful validation, the status of the user is updated, for example, Ready, indicating that the user has sent the HIA and INI initialization letters to the bank. You can also use this service to import or export subscription manager data to or from the bank system database.

The user profile contains the following information:

- Unique ID of the user
- Name of the user
- Address of the user
- Partner ID to which the user is associated
- Mailbox settings to enable uploading, downloading, and archiving of messages

EBICS order types specify the various transactions that can take place between the EBICS server and an EBICS client. An order type can have zero or more file formats. You can associate file formats with the bank-technical upload and download order types. You can use upload order types to upload order data from an EBICS client to an EBICS server and download order types to download order data from an EBICS server to an EBICS client. An order type contains the following attributes:

- The order type
- EBICS protocol version
- Transfer type - Upload or Download
- Order data type - System or Technical

A file format contains the following attributes:

- The file format
- Country code of the file format

A bank can create one or more offers. An offer provides an easy method of grouping a set of order types and file formats to a list of partners. Each partner is allocated a list of order types to enable transactions between the bank and the

partner. An offer provides an easy way for the bank to set up a contract with the partner. An offer contains the following information:

- Bank ID
- Name of the offer
- The order types and file formats using which the partner can exchange messages
- Level of authorization for the order type
- Number of signatures required to authorize the order

A partner can be associated with one or more users. A bank assigns the following permissions to a user:

- The order types and file formats using which the user can exchange messages
- Level of authorization for the order type
- The maximum amount (for a specific partner account) a user can transact. You can associate multiple partner accounts with different maximum amount.
- The currency in which the maximum amount for the user is specified. The currency depends on the partner account associated with the maximum amount.

# Chapter 4. Managing EBICS Transactions

Transaction Manager in the EBICS Server is responsible for maintaining the transaction states. It determines the segment that is required to generate the XML response message.

Transaction Manager handles the upload and download transaction flows and supports segmentation and recovery of order data.

## Upload From a Subscriber (FUL)

The FUL order type is used to upload data to a bank.

The upload transaction comprises the following phases:

- Initialization
- Data Transfer

The user sends the upload (FUL) request to the bank. FUL is a bank-technical upload order type.

**Important:** For large FUL payloads, the Maximum Idle Time (MaxIdleTime) setting in the EBICS Server Service should be increased. If the MaxIdleTime setting is too low, the transaction could be cancelled before it completes. An appropriate setting for large FUL payloads is 300 minutes.

The EBICS Order Authorization service handles incoming order requests for the bank-technical upload order type. If an order has obtained the number of signatures required, this service forwards the order to the subscriber upload mailbox. Otherwise, this service retains the order data in the database until all the required number of signatures is obtained.

The handleEBICSRequest business process receives a user's request. If the user's request contains the last segment of the order data, it invokes the EBICSOrderAuthorizationProcessing business process asynchronously to unpack the order data and generate the following files:

**Note:** Unpacking order data includes decoding, decrypting, and decompressing the order data.

- .DAT - Contains the unpacked order data in a user's upload mailbox
- .SIG - Contains the signature of the order data in a user's upload mailbox
- .PRM - Contains the order parameters in the user's upload mailbox
- .PSR - Contains a status report of asynchronous processing in the user's download mailbox

### Processing Initialization

A user initiates a transaction by submitting the requests containing information about the incoming order. Based on this information, the EBICS Server verifies the order type, performs the message replay test, verifies message authentication, and checks user authorization before accepting the request.

After successful verification of the order data, the bank generates a transaction ID and includes the ID in its response to the user.

### Processing Data Transfer

When more than one segment is required to transfer order data, the bank performs message authentication, verifies the transaction, verifies the segment number and size. After the EBICS Server receives the last segment of the order data, the complete order data is forwarded to the EBICSOrderAuthorizationProcessing business process asynchronously and the transaction ends.

The EBICSOrderAuthorizationProcessing business process unpacks the order data and forwards it to the user upload mailbox. The EBICSOrderAuthorizationProcessing business process generates post processing report (PSR) and routes it to the user's download mailbox. This business process also generates the .SIG and .PRM files to be forwarded to the user's upload mailbox. An .err file is generated when EBICSOrderAuthorizationProcessing business process encounters an error, for example, an invalid electronic signature. Use the .err file to inspect an invalid order data file, if necessary.

## Download From EBICS Server (FDL)

The FDL order type is used to download data from a bank.

The download transaction comprises the following phases:

- Initialization
- Data Transfer
- Acknowledgement

A user submits the FDL order type to the bank. The user requests the download of the .PSR report to get the status of the FUL request. The user can also request to download valid file formats other than .PSR by using the FDL order type.

**Important:** For large FDL payloads, the Maximum Idle Time (MaxIdleTime) setting in the EBICS Server Service should be increased. If this setting is too low, the transaction could be cancelled before it completes. An appropriate setting for large FDL payloads is 300 minutes.

### Processing Initialization

The bank verifies the message from the user. After the bank verifies the user’s request, the bank collects the order data from the user’s download mailbox based on the file format information in the request.

If more than one message matches the file format, the bank joins the contents of each message into a single order data and invokes the order data processor synchronously to pack the order data.

If the encoded form of the order data exceeds 1 MB, the order data is separated into segments. The first segment of the order data and the transaction ID is included in the response to the user.

### Processing Data Transfer

The user sends the request for the next data segment. The bank authenticates the message, verifies the transaction, and the segment number and size.

In each transfer phase, the bank transfers all the segments until the last segment of the order data is included in its response to the user.

### Processing Data Acknowledgement

After receiving the last segment of the order data from the bank, the user initiates the last phase, the acknowledgement request, to indicate that the data transfer has been successful.

If the bank receives a positive acknowledgement (receipt code=0) from the user, the bank moves the downloaded messages from the user download mailbox to the user archive mailbox. If the bank receives a negative acknowledgement from the user, the bank retains the downloaded messages in the user's download mailbox.

If a user wants to download valid file formats other than the .PSR reports from the user's archive mailbox, the user must specify a date range in the EBICS request. The user must ensure that the date range matches the drop date of the .DAT file when moved from the user's download mailbox to the user's archive mailbox.

## Segmentation and Recovery

The order data request (upload or download) cannot exceed 1 MB in compressed, encrypted, base64 encoded form. If the order data request exceeds 1 MB, the encoded form must be separated into segments. EBICS Banking Server is responsible for combining all these segments in order to reinstate the order data to its original form.

If an error occurs during the delivery of the order data segments, recovery can be performed. The user can download or upload the appropriate segment according to the recovery point sent in response by the server.

Recovery allows the transmission of an order to continue despite the occurrence of an error, without necessitating the retransmission of all order data segments that have been transmitted successfully.

A recovery point can be used to continue transactions from the transaction step that follows this recovery point in the transaction step sequence. Recovery points must be set during the recovery process:

- For upload transactions, the recovery point is the last transaction step wherein the bank has successfully received the request message and transmitted a response to the user. The recovery point is determined by the state of the transaction in the bank system.
- For download transactions, several recovery points may exist. All the previous transaction steps of the transaction wherein the bank has successfully received the request message and transmitted a response to the user.

## VEU Processing

EBICS Banking Server supports the Distributed Electronic Signature (VEU), which allows multiple partners (or subscribers) to authorize an order.

VEU is a German abbreviation meaning Distributed Electronic Signature. With VEU, multiple partners(or subscribers) can authorize an order. Different partners from different customers or the same customer can sign a particular order. Partners can request their orders that have pending signatures and sign or cancel them. The VEU management system in EBICS Banking Server saves the orders for which signatures are pending from different partners until one of the following occurs:

- The necessary number of authorized signatures have been received.
- The order is canceled.

VEU uses the following order types:

- HVU
- HVD
- HVZ
- HVE
- HVS
- HVT (Optional)

Authorized signatories of a customer can use different signature processes which may support different hash processes resulting in different hash values. In the VEU process, the hash value of the order data is provided when the order types HVD and HVZ are executed. This hash value is derived from the signature version used by the subscriber executing HVZ and HVD. The hash value is provided with the signature version used as an attribute.

Here is a summary of the typical VEU process:

1. An EBICS Customer (PartnerA) initiates an order by transmitting the order data in an EBICS transaction with the order attribute `OZHNN` and signing with signature class `E`, or `T`.
2. When received by the EBICS Banking Server, the VEU management system analyzes the order type and signatures that have already been submitted, including their class. If further signatures are necessary for processing of the order, it is stored intermediately for the VEU process together with its hash value.
3. Another EBICS customer (Partner B) who has a pending signature and needs to sign a stored order will inquire using order type HVU or HVZ to find out which orders they are authorized to sign. The response includes information about the:
   - order type
   - order number
   - number of signatures required and number already provided (including whether their own signature is still required or has already been provided)
   - original order party
   - size of the uncompressed order data
   - (Order type HVZ only) hash value of the order data

   If order type HVZ was used, skip the next step.
4. Partner B uses order type HVD to check the order and get the hash value of that order.
5. Optional. If order type HVT is supported by the bank, Partner B can download additional order details using order type HVT. Depending on the request parameters, they receive either information on the individual order transactions

(account data, amount information, processing date, utilization data and other descriptions) or the complete order data.

6. When all required information is received, Partner B can sign the order using order type HVE. The VEU management system in EBICS Banking Server validates and adds the signature to the order.
7. Partner B could choose to cancel the order using order type HVS.
8. When all signatures are completed, EBICS Banking Server will completely process the order.

# Chapter 5. Managing Keys

You can insert, update, and retrieve certificates present in the Sterling B2B Integrator repository.

You can insert a base64-encoded certificate (public or private) and import and export certificates into the Sterling B2B Integrator repository.

You can also perform the following tasks in Sterling B2B Integrator:

- Create a self-signed certificate with the key length 2048 for EBICS
- Manage CA certificates
- Store certificates, and manage the renewal and expiration of certificates
- Accept a public certificate of a user
- Validate the following subscriber keys using SHA256 as the hash algorithm:
  - Identification and Authentication Key Hash Value (in Hex format)
  - Encryption Key Hash Value (in Hex format)
  - Electronic Signature Key Hash Value (in Hex format)

Use the EBICS Export Certificate service to export the certificates present in Sterling B2B Integrator to an external system. Use this service when you want to synchronize the certificates present in Sterling B2B Integrator with an external database or system.

Use the EBICS Import Certificate service to add certificates from an external repository to Sterling B2B Integrator. You can also delete the expired or invalid certificates.

## Functions of the Key Manager

The Key Management and Storage performs the following functions:

- Duplicate Key Validation - The certificate used for authentication or encryption cannot be the same as the ES certificate. Use a unique set of keys for authentication or encryption and signing.
- X.509 Key Usage Extension – EBICS Banking Server supports the use of X.509 as the key usage extension.
- OCSP and CRL certificate verification

The Key Manager manages the certificates in the Sterling B2B Integrator repository. It inserts, updates, and retrieves certificates in the Sterling B2B Integrator repository and runs functions such as, calculating the hash value of the certificate, on the certificates.

The Key Manager validates the client certificates checked into the server before they can be used. You must obtain the CA-signed certificates from a Certificate Authority. In a CA-signed certificate, the issuer signs the certificate. To verify the authenticity of the user certificate, the EBICS Banking Server performs chained signature verification up to the root CA certificate.

The EBICS administrator must check in the CA-signed certificates and Intermediate CA-signed certificates in the Sterling B2B Integrator CA certificate store before commencing the EBICS transactions.

The client must provide three types of certificates:

- Authentication certificate
- Encryption certificate
- Electronic Signature (ES) certificate

The public key of the authentication certificate is used to verify digital signatures. Authentication certificates can be either CA-signed or self-signed. The value of the key usage field for an authentication certificate is Digital Signature. A digital signature is used for entity authentication and data origin authentication with integrity.

The public key of the encryption certificate is used to encrypt order data. Encryption certificates can be either CA-signed or self-signed. The value of the key usage field for an encryption certificate is Key Encipherment. In EBICS, a symmetric key is used to stream encrypted or decrypted order data. The symmetric key is encrypted with the public key value of encryption certificate for transportation. Key Encipherment is used when a certificate with a protocol that encrypts keys exists.

The public key of the Electronic Signature (ES) certificate is used to verify the signature of order data. The public key value of an Electronic Signature certificate should not be the same as an authentication or encryption certificate. The value of the key usage field for an electronic signature certificate is Non-Repudiation. Non-repudiation protects against the signing entity falsely denying an action, excluding certificate or CRL signing. Electronic Signatures are of two types:

- Transport Signature – can be CA-signed or self-signed
- Personal Signature – must be CA-signed

# Chapter 6. Generating and Retrieving EBICS Reports

Use the EBICS Reporting service to generate a payment status report (PSR) with every upload order (FUL) request. The .PSR report is in an XML format and follows the pain.002.001.02 schema. After the .PSR report is generated successfully, it is placed in the EBICS user's download mailbox.

A .PSR report is generated after asynchronous order processing of each FUL. A user can send an FDL request with the FileFormat pain.002.001.02.ack to retrieve the .PSR report. If no date range is specified in the EBICS request, the bank concatenates the PSR reports in the user's download mailbox, and packages the order data in the EBICS response.

When the bank receives a positive acknowledgement from the user based on the parameter value provided under the FDLOrderParams element in the FDL request, the .PSR reports in the user's download mailbox are moved to the user's archive mailbox. If no positive acknowledgement is received after a specified time-out period, the EBICS Server Service scheduler changes the Extractable Count back to 1 for the .PSR reports in the user's download mailbox, enabling the user to download the .PSR reports again.

If the user wants to download the .PSR reports from the user's archive mailbox, the user must specify a date range in the EBICS request. The user must ensure that the date range matches the drop date of the .PSR reports when moved from the user's download mailbox to the user's archive mailbox.

# Chapter 7. Managing the EBICS Server

The EBICS Server is implemented as a service in Sterling B2B Integrator. The EBICS Server service is responsible for handling incoming EBICS requests (through HTTP and HTTPS) according to the EBICS protocol specifications, and generating and sending the appropriate response back to users.

The EBICS Server processes the generation and verification of electronic signature (ES), and identification and authentication of EBICS messages. It also interfaces with Subscription Manager to retrieve the profile information of banks, partners, users, and order types necessary for verification and authentication of messages and transactions. The processing flows (asynchronous and synchronous) of requests, such as, FUL and FDL, are also managed by the service. You can configure the service to update the EBICS repository and send event notifications to an external application during a synchronous transaction. Managing the message flow for the initialization and transfer phases of EBICS transactions is also one of the key responsibilities of the service. The lifecycle of the EBICS transactions in the bank system and the status of open transactions are managed by the EBICS Server, which also acts as an intermediate storage for transmitted order data segments and Electronic Signatures (ES).

When downloading bank-technical order data, the EBICS Server collects all the available order data in the user's mailbox, and concatenates them into a single document and sends the document to the order data processor to pack the document, that is, sign, compress, encrypt, and encode the document.

For information about configuring EBICS Server Service, see *EBICS Server Service*.

# Chapter 8. Managing System Order

System Order Manager works closely with Transaction Manager and Subscription Manager to update and query a user's key certificates and referential information, and to download bank parameters and bank certificates. It generates and retrieves XML order data based on the profile information.

System Order Manager also handles the implementation of upload and download system orders. The following table lists the supported upload system order types for EBICS transactions:

| Upload System Order Type | Description |
|---|---|
| INI | Used in subscriber initialization. Sends the bank-technical public certificate of a customer to the EBICS Banking Server. The order data is compressed and base64-encoded. |
| HIA | Used to transmit user public certificates for identification, authentication and encryption within the framework of subscriber initialization. The order data is compressed and base64-encoded. |
| PUB | Used to update customer's certificates. Sends the bank-technical public certificate of the customer for updating the EBICS Banking Server. The order data is signed, compressed, encrypted, and base64-encoded. |
| HCA | Used to update customer's certificate. Sends the following certificates for updating the EBICS Banking Server:<br>• Identification and authentication public certificate<br>• Encryption public certificate<br>The order data is signed, compressed, encrypted, and base64-encoded. |
| HCS | Used to update customer's certificate. Sends the following certificates for updating the EBICS Banking Server:<br>• Bank-technical public certificate<br>• Identification and authentication public certificate<br>• Encryption public certificate<br>The order data is signed, compressed, encrypted, and base64-encoded. |
| SPR | Used to suspend a user's access authorization. The order data is signed, compressed, encrypted, and base64-encoded. |

The following table lists the supported download system order types for EBICS transactions:

| Download System Order Type | Description |
|---|---|
| HPB | Used to download bank public certificates from the EBICS Banking Server. The order data is compressed, encrypted, and base64-encoded. The response message is signed with an XML digital signature using the authentication certificate. The order data is not signed. |
| HPD | Used to download bank parameters from the EBICS Banking Server. The order data is compressed, encrypted, and base64-encoded. The response message is signed with an XML digital signature using the authentication certificate. The order data is not signed. |
| HEV | Used to download information on supported EBICS versions. The response message is in clear text. There is no order data in an HEV response. |
| HKD | Used to download customer and subscriber data. Can be used when the user is in ready status. Retrieves information stored by the bank pertaining to the subscriber's company and associated subscribers (including the bank's own information). The order data is compressed, encrypted, and base64-encoded. The response message is signed with an XML digital signature using the authentication certificate. The order data is not signed. |
| HTD | Used to download customer and subscriber data. Can be used when the user is in a ready status. Retrieves information stored by the bank pertaining to the subscriber's company or the bank's own information. The order data is compressed, encrypted, and base64-encoded. The response message is signed with an XML digital signature using the authentication certificate. The order data is not signed. |

The System Order manager retrieves information stored by the bank pertaining to the subscriber's company. The subscribers can retrieve information stored by the bank pertaining to the subscriber's company and all the associated subscribers using HKD and HTD order types after the user status is set to 'Ready' indicating that the user can transact with the bank. The bank's response contains a list of the accounts of the customer.

The account information is included in the HKD response if at least one of the following conditions is met:

- In the contractual agreement with the bank, it is specified that the statements of the account will be shared with the customer
- At least one of the customer's subscribers is authorized to sign for the account

The subscribers can retrieve information stored by the bank pertaining to the subscriber's company or the bank's own information using the HTD order type. However, information pertaining to the company's associated subscribers is not shared in this order type. You must use the HKD order type to retrieve information pertaining to the company and the associated subscribers (including the bank's own information). The HKD and HTD response lists the associated accounts of the partner to which the subscriber has permission to access.

The response message of the HKD Download System Order includes the following parameters:

- HostID
- PartnerInfo - Includes details of the partner, such as the address, the account information for which the subscriber has permission to access, and the order types the partner is authorized to use.
- UserInfo - Includes details of the subscriber, such as the User ID, status of the subscriber, and the user permission information. The user permission information includes the authorization level of the list of order types, associated accounts, and the amount threshold limit.

# Chapter 9. Processing Order Data

To ensure secure transfer of order data, the order data must be packed. Packing the order data includes signing, compression, encryption, and base64 encoding depending on the requirements of the order type. The receiver must unpack the order data to view the attributes. Unpacking the order data includes verification, decompression, decryption, and base64 decoding depending on the requirements of the order type.

The Order Data Processor is responsible for packing and unpacking the order data. It interfaces with the Subscription Manager and Transaction Manager to retrieve the relevant information required for packing and unpacking the order data. For example, the profile information may include the transaction ID, the direction of the flow (upload or download), response type (synchronous or asynchronous), type of processes required, object ID of the encrypted key, and object ID of the Electronic Signature (ES). EBICS Order Processing service performs EBICS transactions and user retrieval, and packing and unpacking of encrypted symmetric keys. Based on the profile information that is retrieved, the EBICS Order Processing service determines if packing or unpacking the order data is required, and invokes the appropriate packing or unpacking service.

Authorized Order Manager is responsible for initiating the Order Data Processor to unpack the order data received from the FUL order type request, routing the unpacked order data to the backend subscriber's upload mailbox, and renaming it according to a defined naming convention.

Apart from the EBICS Order Processing service, the following services are available in Sterling B2B Integrator to process order data:

- The EBICS Order Authorization service handles incoming order requests for the bank-technical upload order type (FUL). If an order has fulfilled the number of signatures required, this service forwards the order to the subscriber upload mailbox. Otherwise, this service forwards the order to the pending order mailbox.
- The EBICS Order Streaming service packs and unpacks order data using the pipeline functionality in Sterling B2B Integrator.
- The EBICS ES Packaging service either packs or unpacks key information that is used when signing and verifying the ES.
- The EBICS Compression service performs compression and decompression of order data using zlib in pipeline mode.
- The EBICS Encryption service performs encryption and decryption of order data using the AES-128 algorithm in pipeline mode. E002 encryption algorithm is supported.
- The EBICS Encoding service performs encoding and decoding of order data using the base64 method in pipeline mode.
- The EBICS Signing service performs the signing and verification of order data on the SHA-256 digest computed in pipeline mode. A005 and A006 signing algorithm is supported.

Order data must be unpacked for upload transactions, and packed for download transactions.

The packing process involves the following sequence. However, based on the order type, one or more of the following processes may not be required:

1. Signing
2. Compressing
3. Encrypting
4. Base64 encoding

The following example illustrates encryption of an order type. A business process invokes the Encryption service. If the order data has been signed, the business process passes the symmetric key to the Encryption service. If the order data has not been signed, the Encryption service generates and returns the symmetric key to the business process. If the symmetric key was created, the business process invokes the EBICS Order Processing service with the output message type set to setEncryptedKey.

The unpacking process involves the following sequence. However, based on the order type, one or more of the following processes may not be required:

1. Base64 decoding
2. Decrypting
3. Decompressing
4. Verifying the signature

The following example illustrates decryption of an order type. A business process invokes the EBICS Order Processing service with the output message type set to getEncryptedKey. The base64-encoded secret key is retrieved and set in the process data for use by the Encryption service.

## Electronic Signatures

The Electronic Signature (ES) ensures the authentication of the order data. The signatures ensure the integrity and non-repudiation of order data sent by the client to the banking server.

EBICS specifies two signature classes of ES:

- Personal signature
  - Single signature of type E
  - First signature of type A
  - Second signature of type B
- Transport signature of type T

Sterling B2B Integrator supports the following signature types:

- Transport signature of type T
- Personal signature or Bank-technical ES of type E – Single signature

Transport signatures can be either self-signed or CA-signed certificates. Personal signatures must be CA-signed and recognized by the bank. Use the transport signature to submit the order and the personal signature to authorize the order.

In personal signatures, you must specify the number of signatures for each order type or file format in the contract to process the order data. The maximum number of personal signature allowed is 2. Personal signature of type E can contain the following signatures:

- Single
- Optional Dual
- Mandatory Dual

## Prevalidation

When using bank-technical upload order types, the subscriber can send information in the first transaction step to the bank. The bank can prevalidate the order data. Prevalidation of order data includes the following:

- Data digest verification
- Account authorization
- Amount limit verification

After prevalidation of the order data is successful, the bank system receives the FUL file. The bank can use prevalidation to process the order data if the following prerequisites are met:

- The bank supports the prevalidation functionality
- Prevalidation node exists in the incoming request

**Prevalidation of Data digest**

The bank can verify data digest if the following prerequisites are met:

- The bank supports the prevalidation functionality.
- Prevalidation or DataDigest node exists in the incoming request.
- The order type is set to any upload order type except the SPR request.

**Prevalidation of account authorization and amount limit**

The bank can verify account authorization and amount limit if the following prerequisites are met:

- The bank supports the prevalidation functionality.
- Prevalidation or AccountAuthorization node exists in the incoming request.
- The OrderAttribute attribute is not set to DZHNN in the incoming request.
- The Order Type is set to Technical upload order type (FUL) in the incoming request.
- The signature class of the signatory is at least B in the contract permission.

Prevalidation verifies the signatory-designated account information and the amount limit if the minimum and maximum number of personal signatures required to authorize the order is defined. The account listed under AccountAuthorization must be a valid partner account. All the signatories must be configured with user permission to all the accounts listed in Prevalidation. The amount of a specified currency value must not exceed the maximum amount set in the User Permission configuration for any signatory.

# Chapter 10. Integrating with Sterling File Gateway

Sterling File Gateway enables secure file transfer between internal and external partners using the same or different communication protocols, file naming conventions, and file formats. Sterling File Gateway supports EBICS for movement of large and high-volume file transfers, with end-to-end visibility of file movement in a process-oriented and highly-scalable framework that alleviates file transfer challenges, such as protocol and file brokering, automation, and data security.

Files move between the EBICS server and Sterling File Gateway through shared mailboxes and partners. The Subscription Manager creates mailboxes in the structure of User/Partner/Inbox during partner creation.

Sterling File Gateway uses Provisioning Facts as part of the Routing Channel Template definition. Routing channel templates used in EBICS scenarios must include the configuration of provisioning facts. Routing channels using the templates must include the specification of values for provisioning facts.

For inbound scenarios, the EBICS Order Data Processor (ODP) takes an EBICS order file upload (FUL) from an EBICS client to an EBICS Server, unpacks the payload and deposits into a User/Partner/Inbox mailbox structure. Sterling File Gateway is configured to route from that mailbox for downstream processing and ultimate delivery to a consumer.

In the outbound scenario, Sterling File Gateway is configured to deposit a message in a consumer mailbox, which is routed and stored in User/Partner/Outbox. On an EBICS order file download (FDL) from an EBICS client to an EBICS Server, the EBICS Order Data Processor (ODP) packages the message and makes it available to the client.

Sterling File Gateway enables operators to search for transactions and view details of routes and deliveries.

Certain procedures are necessary to initiate integration with Sterling File Gateway. For more information about integrating with Sterling File Gateway, see *Sterling File Gateway Integration with EBICS* at http://www.ibm.com/support/knowledgecenter/SS4TGX_2.2.0/com.ibm.help.sfg_ebics.doc/SFGEB_IntegrationwEBICS.html.

# Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive*

*Armonk, NY 10504-1785*

*U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan Ltd.*

*19-21, Nihonbashi-Hakozakicho, Chuo-ku*

*Tokyo 103-8510, Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation*

*J46A/G4*

*555 Bailey Avenue*

*San Jose, CA 95141-1003*

*U.S.A.*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2014. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2014.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise®, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce®, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Printed in USA