Sterling B2B Integrator

EBICS Client Overview

Version 5.2.4

Sterling B2B Integrator

EBICS Client Overview

Version 5.2.4

Note

Before using this information and the product it supports, read the information in "Notices" on page 31.

Copyright

This edition applies to Version 5 Release 2 Modification 4 of Sterling B2B Integrator and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Overview of EBICS

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

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

The following entities are involved in EBICS Client transactions:

Organization
The organization or company that uses EBICS Client to transact with the bank.

Bank Financial institutions with which the organization transacts. The EBICS Banking Server is installed in the bank.

Partner
The department or unit in the organization that interacts with the bank.

User or Subscriber
Personnel in the department, who perform the EBICS transactions.

An organization has to fulfill a range of prerequisites for it to be able to implement bank-technical EBICS transactions with a particular bank. The basic prerequisite to implement EBICS transactions is the completion of a contract between the partner and the bank. The EBICS protocol defines bank transactions (order types) for communication. The following details are agreed upon in this contract:

- Type of business transactions.
- Information about the user's bank accounts.
- Information about the partners users working with the bank's system.
- Authorizations and permissions of the users.

The partner receives the access data of the bank (bank parameters) after the contract is signed. The bank configures the partner and user master data in the bank system in accordance with the contractual agreements. Other prerequisites include subscriber initialization, download of the bank’s public certificates by the user, verification of the user's public certificates by the bank, and verification of the bank’s certificates by the trading partner.

IBM® Sterling B2B Integrator offers a complete EBICS solution by providing a secure, flexible, and efficient platform to banks and organizations for performing the transactions. The implementation of this solution is divided into two major components: EBICS Banking Server and EBICS Client. EBICS Banking Server represents a bank and EBICS Client represents an organization. Both the server and the client are deployed over Sterling B2B Integrator.

Chapter 2. Overview of EBICS Client

EBICS Client of Sterling B2B Integrator is a client server application. It provides an end-to-end EBICS solution for an organization to transact with banks. Using EBICS Client, a partner or partner user can configure and manage multiple banks, partners, and users. Multiple users can interact with multiple banks (EBICS Banking Servers) over HTTP or HTTPS and exchange EBICS-compliant transaction messages.

Partners can perform the following tasks in the EBICS Client dashboard interface:

- Configure users
- Configure banks
- Configure security settings for users
- Verify security settings of a bank
- Create and manage file formats
- Create and manage user permissions
- Create and manage offers
- Configure orders
- Submit orders
- View order-related events and reports
- Search for orders pending at the VEU management store in the server
- View and monitor pending tasks

Note: EBICS Client supports French and German implementation of EBICS version 2.4.2.

Chapter 3. EBICS Client Architecture and Key Features

The EBICS Client architecture and key features section provides an overview of the EBICS Client architecture and describes the key features of EBICS Client.

EBICS Client Architecture

EBICS Client is deployed over Sterling B2B Integrator and reuses some of the following core functionalities of Sterling B2B Integrator:

- Creating and managing trading partner
- Managing digital certificates
- Creating and managing mailboxes
- Creating users
- Running services and adapters
- Scheduling business processes

The following diagram illustrates the EBICS Client architecture.

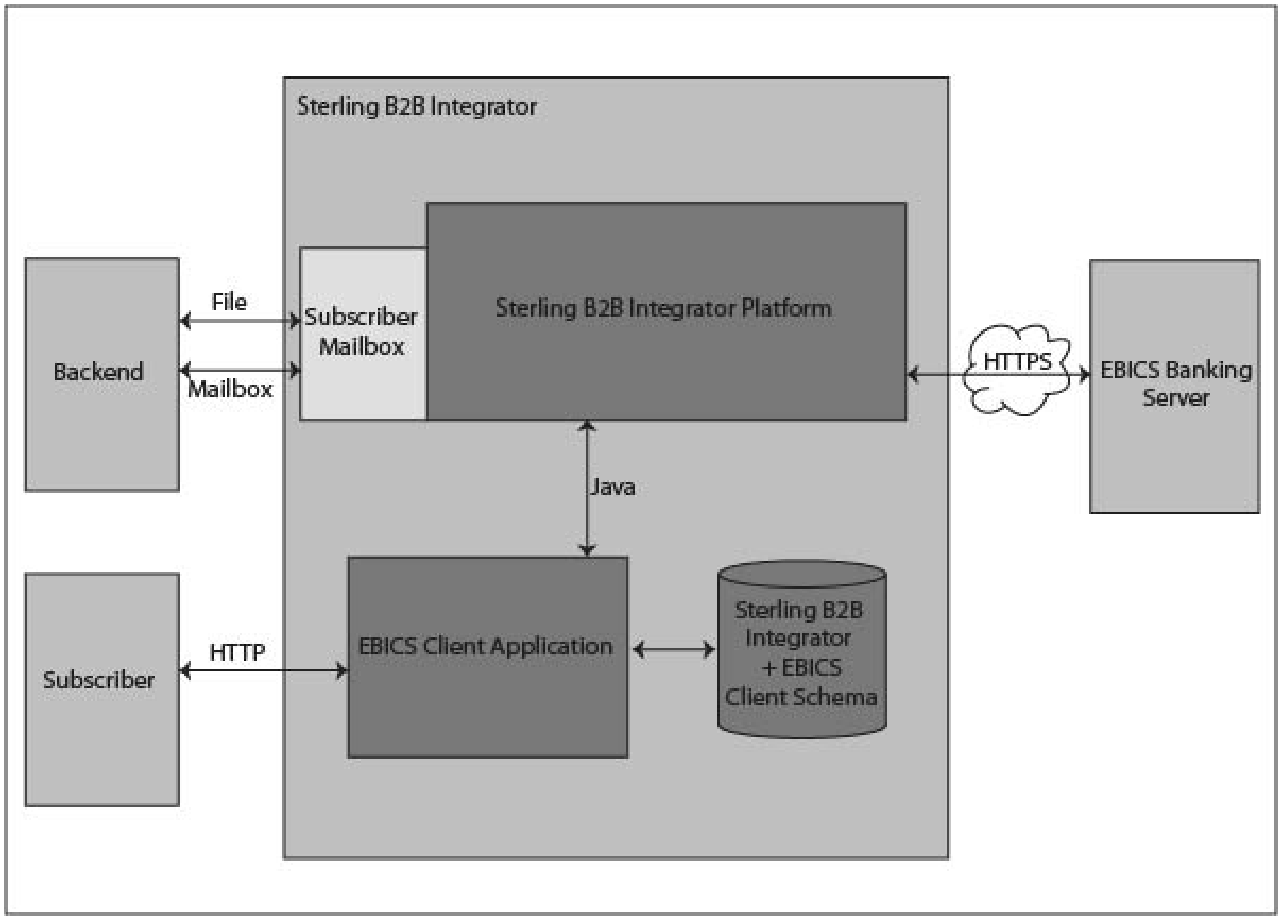

Figure 1. EBICS Client architecture

The following sections describe the components of EBICS Client architecture.

Subscriber

A subscriber uses EBICS Client to perform bank transactions. The subscriber or user communicates with the EBICS Client application using the HTTP protocol to send and receive messages.

EBICS Client Application

EBICS Client application provides a dashboard interface to enable you to configure and manage the following EBICS entities:

- User
- Bank
- Offers
- File format
- User permissions
- Order submission
- Pending tasks
- Bank key validation
- Keys

EBICS Client application interfaces with Sterling B2B Integrator and EBICS schema database to retrieve and store subscription and order related data.

Sterling B2B Integrator and EBICS Client Schema

Sterling B2B Integrator and EBICS Client schema tables are stored in a common database to enable the EBICS Client application to access the following data:

- Native Sterling B2B Integrator data in Sterling B2B Integrator schema tables.
- EBICS Client data in EBICS schema tables.

Sterling B2B Integrator Platform

The Sterling B2B Integrator platform on which the EBICS Client is deployed.

Subscriber Mailbox

The Subscriber Mailbox provides a safe access mechanism to send and receive messages between EBICS Client and EBICS Banking Server. The following mailboxes are configured for each user:

EBClientOrderMetadata

The EBClientOrderMetadata mailbox is a common mailbox associated to all users. OrderMetadata associated with a payload is posted in the EBClientOrderMetadata mailbox for processing the payload submitted by a technical or non-technical user.

Download (Inbox)

The Download mailbox is used for posting downloaded response in case of HEV order type and unpacked data in case of download orders.

Upload (Outbox)

The Upload mailbox is used for posting the payloads.

When an upload (FUL) order type is submitted, the payload is routed to the Upload mailbox and the related OrderMetadata is routed to the

EBClientOrderMetadata mailbox. The arrival of an OrderMetadata in the EBClientOrderMetadata mailbox, either for an upload (FUL) or download (FDL) order type, triggers an EBICS Client request workflow.

Back-end

A Sterling B2B Integrator adapter that can trigger the EBClientOrderPreProcess business process is used to submit orders automatically from the back-end.

EBICS Banking Server

The EBICS Banking Server is installed in a bank. EBICS Client communicates with the EBICS Banking Server application using the HTTP or HTTPS protocol to send and receive information about users, trading partners, digital certificates, order data, file formats, and order types. For more information about EBICS Banking Server, see the *Sterling B2B Integrator EBICS Banking Server* documentation.

EBICS Client Components

EBICS Client consists of two major components: EBICS Client Graphical User Interface and EBICS Client Runtime. Both the components use mailboxes (Upload, Download, and EBClientOrderMetadata) assigned to the individual subscribers. When a subscriber posts an EBICS request, the EBICS Client Business Process picks the posted request from the Upload and EBClientOrderMetadata mailboxes and based on the order type request, delivers the request to the EBICS Banking Server. The response received from the server is processed according to the order type and is posted to the subscriber Download mailbox.

The following diagram illustrates the components of EBICS Client.

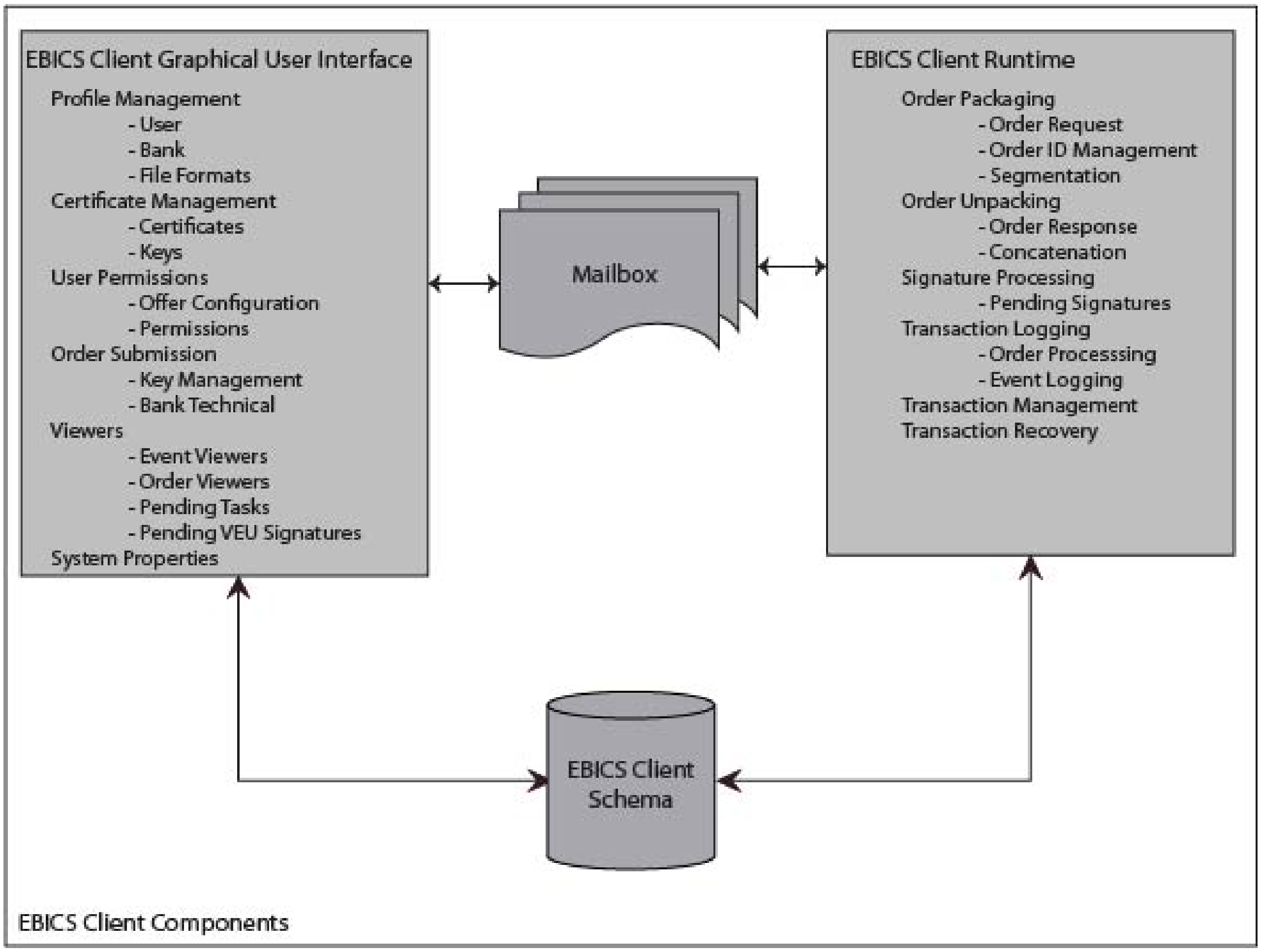

Figure 2. EBICS Client components

The following sections describe the components of EBICS Client.

EBICS Client Graphical User Interface

EBICS Client Graphical User Interface consists of the following components:

Profile Management
This component enables you to configure and manage bank profiles, configure existing Sterling B2B Integrator users as EBICS Client users, and to configure file formats.

Certificate Management
This component is responsible for the verification of electronic signatures (ES), identification and authentication, and encryption certificates or keys of banks and users.

User Permissions
This component enables you to configure and manage offers and user permissions.

Order Submission
This component enables you to process key management orders and bank-technical upload and download orders.

Viewers

This component provides a summary view of selected orders or order-related events and allows users to sign or submit pending orders.

System Properties

This component provides a summary view of the system properties values. An EBICS Client admin or EBICS Client super admin can update the values if required.

EBICS Client Runtime

EBICS Client Runtime consists of the following components:

Order Packaging

This component invokes appropriate packaging handlers and ensures that the order is packaged according to the specifications. It is also responsible for order data segmentation.

Order Unpacking

This component is responsible for unpacking the payload received from the EBICS Banking Server. Unpacking includes providing appropriate order response to users and concatenation of order data segments in case of downloads.

Signature Processing

This component verifies whether the required signatures for an order are available or not. If not, the Pending tasks page on EBICS Client dashboard interface is updated with the pending order details. When an EBICS Client user with pending orders (signing or submitting) logs in to the EBICS Client dashboard interface, the Pending tasks page opens prompting the user to sign or submit the order.

Transaction Logging

This component is responsible for logging transaction data in to appropriate tables in the database.

Transaction Management

This component initiates a session to communicate with the EBICS Banking Server, manages transactions between the client and the server, and processes the responses received from the server.

Transaction Recovery

This component is responsible for recovering failed transactions. The number of recovery attempts depends on the value specified in the bank profile configuration. The EBICS Banking Server maintains the recovery attempts count. The server cancels the recovery after the specified number attempts.

Mailbox

The mailbox component interfaces between the EBICS Client Graphical User Interface and EBICS Client Runtime components. EBICS Client uses the following mailboxes to store and process order request and responses:

- EBClientOrderMetadata. This is a common mailbox associated to all users.
- Download (Inbox)
- Upload (Outbox)

Separate upload and download mailboxes are created and assigned to each EBICS Client user.

EBICS Client Schema

EBICS Client-related data are stored in the EBICS Client Schema tables. The EBICS Client Graphical User Interface and EBICS Client Runtime components access the tables to retrieve information about users, banks, offers, user permissions, and orders.

EBICS Client Key Features

This section describes some of the key features of EBICS Client.

Managing profiles for users

You can configure the following permission types to enable a user to access the EBICS Client dashboard interface:

EBICS Client Admin
An EBICS Client admin can configure an existing Sterling B2B Integrator user as an EBICS Client user or EBICS Client operator. The admin can also configure the following entities in EBICS Client dashboard interface:

- Bank profile
- User profile
- Offer
- File format
- Keys
- User permission
- View events
- Search for orders

However, an EBICS Client admin user cannot submit orders.

EBICS Client Operator
An EBICS Client operator can view information about user and bank profiles, view events, and search for orders. However, the EBICS Client operator cannot perform any create, edit, or delete operations in the EBICS Client dashboard interface.

EBICS Client User
An EBICS Client user can sign and submit orders, search self-submitted orders and view events for self-submitted orders.

Technical User

EBICS Client also supports a technical user. A technical user is an EBICS Client user configured to submit orders on behalf of a non-technical (human user) EBICS Client user using a back-end system. The technical user is associated with a non-technical user. The **SystemID** field in the EBICS request is populated with the technical subscriber user ID. Electronic signature (ES), authorization and encryption certificates are linked to the system ID and are verified accordingly. If a payload is received over a file system adapter, or any other technical adapter, such as, JSM or FTP, then EBICS Client application uses the ID of the technical user specified in the XML file and submits the order. If the payload is received over an EBICS Client user's mail box, then the user ID of the user is used to submit the order. Permissions for order submission are inherited from the user ID when an order is submitted over a user's mailbox. For example, if an upload order type with file format pain.xxx.cfonb160 is being submitted, EBICS Client verifies the permissions of the user to submit the order type file format combination. However, EBICS Client verifies the certificates for electronic signature, authentication, and encryption specified for the system ID. The electronic signature for a technical user is set to Transport signature of type T.

A compressed file (.zip) containing an XML file, ordermetadata.xml, and optionally the payload (for FUL and similar order types) are uploaded to the EBICS Client from the back-end. If the file name of the payload has non-ASCII characters, then use the jar utility that comes with the Java Development Kit (JDK) installed with Sterling B2B Integrator to create a compressed file. You have to execute the jar utility from the command prompt in Windows or the terminal in UNIX with the following parameters: `jar cFM <zip_fileName> ordermetadata.xml <payload_fileName with non-ASCII characters>`. If the file name of the payload has only ASCII characters, then either the jar utility or any application such as WinZip or WinRAR can be used to create a compressed file.

Note: Java Home must be set to the JDK.

EBICS Client collects the compressed file through an adapter such as the File System Adapter (FSA) configured on Sterling B2B Integrator. After the compressed file is received, the EBClientOrderPreProcess business process extracts the contents of the XML file and generates an EBICS request based on the values specified in the XML file. The XML file must conform to the following XSD. The technical user is specified in the System ID field.

```
<?xml version="1.0" encoding="UTF-8"?><xsd:schema xmlns:xsd=
       "http://www.w3.org/2001/XMLSchema">
  <xsd:element name="PartnerID" type="xsd:string"/>
  <xsd:element name="UserID" type="xsd:string"/>
  <xsd:element name="OrderType" type="xsd:string"/>
  <xsd:element name="SystemID" type="xsd:string"/>
  <xsd:element name="Parameter">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Name" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Value" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Type" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ParameterList">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Parameter" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="UserAuthNewPubKeyID" type="xsd:string"/>
  <xsd:element name="UserAuthNewPriKeyAlias" type="xsd:string"/>
  <xsd:element name="PAYLOADMSGID" type="xsd:string"/>
  <xsd:element name="Product" type="xsd:string"/>
  <xsd:element name="Value" type="xsd:string"/>
  <xsd:element name="UserEncrNewPubKeyAlias" type="xsd:string"/>
  <xsd:element name="UserNewSignatureVersion" type="xsd:string"/>
  <xsd:element name="UserSignNewPubKeyID" type="xsd:string"/>
  <xsd:element name="Type" type="xsd:string"/>
  <xsd:element name="UserEncrNewPubKeyID" type="xsd:string"/>
  <xsd:element name="UserNewAuthVersion" type="xsd:string"/>
  <xsd:element name="PreValidation" type="xsd:string"/>
  <xsd:element name="UserAuthNewPriKeyID" type="xsd:string"/>
  <xsd:element name="UserSignNewPriKeyAlias" type="xsd:string"/>
  <xsd:element name="UserAuthNewPubKeyAlias" type="xsd:string"/>
  <xsd:element name="HostID" type="xsd:string"/>
  <xsd:element name="autoSubmit" type="xsd:string"/>
  <xsd:element name="Name" type="xsd:string"/>
  <xsd:element name="UserSignNewPriKeyID" type="xsd:string"/>
  <xsd:element name="UserEncrNewPriKeyID" type="xsd:string"/>
  <xsd:element name="UserNewEncVersion" type="xsd:string"/>
  <xsd:element name="FileFormat" type="xsd:string"/>
  <xsd:element name="DownloadDateRangeEnd" type="xsd:string"/>
  <xsd:element name="SecurityMedium" type="xsd:string"/>
  <xsd:element name="UserSignNewPubKeyAlias" type="xsd:string"/>
  <xsd:element name="UserEncrNewPriKeyAlias" type="xsd:string"/>
  <xsd:element name="orderIdPrefix" type="xsd:string"/>
  <xsd:element name="DownloadDateRangeStart" type="xsd:string"/>
  <xsd:element name="orderMetaData">
    <xsd:complexType>
      <xsd:all>
        <xsd:element ref="HostID"  minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="PartnerID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="PAYLOADMSGID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="PreValidation" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="Product" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="orderIdPrefix" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="SecurityMedium" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="OrderType" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="SystemID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="FileFormat" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="autoSubmit" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="DownloadDateRangeStart" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="DownloadDateRangeEnd" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserSignNewPubKeyAlias" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserSignNewPubKeyID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserSignNewPriKeyAlias" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserSignNewPriKeyID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserAuthNewPubKeyAlias" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserAuthNewPubKeyID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserAuthNewPriKeyAlias" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserAuthNewPriKeyID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserEncrNewPubKeyAlias" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserEncrNewPubKeyID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserEncrNewPriKeyAlias" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserEncrNewPriKeyID" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserNewSignatureVersion" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserNewAuthVersion" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="UserNewEncVersion" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="ParameterList" minOccurs="0" maxOccurs="1"/>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Managing certificates and keys for users

EBICS Client supports both Keys and X.509 certificate types for user's identification and authentication, encryption, and electronic signatures. EBICS Client supports the following versions:

- Electronic signature - A005 and A006
- Identification and authentication - X002
- Encryption - E002

Certificates

X.509 is a standard used to define digital certificates. EBICS Client supports use of X.509 to verify digital signatures. EBICS Client users can use one of the following certificate types:

- Self-signed certificates with hash algorithm SHA256
- CA-signed certificates

When X.509 certificate type is used for authentication, encryption, and ES of an EBICS Client user, an EBICS Client admin specifies appropriate public and private keys while configuring the user profile. The EBICS Client user then shares the public keys for ES with the bank through the INI (Initialization) order type and public keys for identification and authentication and encryption through the HIA order type.

Note: Self-signed certificates cannot be used for electronic signatures and consequently for user initialization (INI order type). An EBICS Client user using self-signed certificates for identification and authentication and encryption, has to use CA certificates for electronic signatures.

EBICS Client supports hardware keystore for electronic signature certificate. The hardware keystore support is available only for 3SKey hardware key type.

Keys

When Keys are used for authentication, encryption, and ES of an EBICS Client user, an EBICS Client admin generates or uploads private keys while configuring the user profile. The EBICS Client user then shares the public keys for ES with the bank through the INI order type and public keys for identification and authentication and encryption through the HIA order type.

Note: Use a third-party tool to generate the keys.

Submitting orders

An order type defines the nature of an EBICS transaction. EBICS Client supports the following order types:

Key management order type

This order type is used for downloading technical information such as bank key, user initialization, key management, cancelation of orders, VEU, and so on. Key management order type is also called as System order type.

Bank-technical order type

This order type is used for various upload and download transactions that happen between a subscriber and a bank. The contract signed between a bank and a partner specifies the order types the users can submit. The bank configures the order types using the Sterling B2B Integrator EBICS Banking Server. Bank-technical orders are classified as Upload orders (FUL) and Download orders (FDL). You can upload an order payload, that is submit order to a bank using an upload order. A download order allows you to download a report or statement from the bank.

French Order Types

The following table lists the supported upload key management order types for French implementation:

Table 1. Upload Key management order types for French implementation

Upload Key management 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 and 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: • Identification and authentication public certificate • Encryption public certificate 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: • Bank-technical public certificate • Identification and authentication public certificate • Encryption public certificate The order data is signed, compressed, encrypted, and base64-encoded.
SPR	Used to suspend a user's access authorization. Only the electronic signature of the EBICS Client user is sent. The order data is a blank character. The signature is compressed, encrypted, and base64 encoded as in regular upload.

The following table lists the supported download key management order types for French implementation:

Table 2. Download Key management order types for French implementation

Download Key management 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 and the order data are not signed.

Table 2. Download Key management order types for French implementation (continued)

Download Key management order type	Description
HPD	Used to download bank parameters from the EBICS Banking Server. The order data is signed, compressed, encrypted, and base64-encoded.
HEV	Used to download information about supported EBICS versions.
HKD	Used to download information about a partner and associated subscribers. The order data is compressed and base64-encoded.
HTD	Used to download information about a subscriber and associated partner. The order data is compressed and base64-encoded.

Submitting distributed signature

EBICS Client supports the German order types defined in the Annexure 2 to the EBICS Specification document (*EBICS_Annex2_OrderTypes-File_Formats_01_11_2010.pdf*). The File Format attribute identifies the type of file that is uploaded or downloaded. The file format attribute is required as part of the order details. In the German implementation of EBICS, the order types define the nature of the financial transaction, for example, an order type is used for credit transfer, another order type is used for direct debit, and so on. The order type identifiers for German implementation follow the legacy file transfer protocol (BCS FTAM). An EBICS Client user can upload or download order types for Germany similar to using the generic FUL (upload) and FDL (download) order types.

Distributed Electronic Signature (VEU) is a feature that allows orders to be authorized by multiple subscribers. According to EBICS implementation, a bank has to mandatorily provide support for VEU.

The following table lists the supported upload VEU order types for German implementation:

Table 3. Download VEU order types for German implementation

Download VEU order types	Description
HVU	Used download VEU overview. The order data is compressed and base64-encoded.
HVD	Used to retrieve VEU state. The order data is compressed and base64-encoded.
HVZ	Used to download VEU overview with additional information. The order data is compressed and base64-encoded.
HVT	Used to retrieve VEU transaction details. The order data is compressed and base64-encoded.

The following table lists the supported download VEU order types for German implementation.

Table 4. Upload VEU order types for German implementation

Upload VEU order types	Description
HVE	Used to add VEU signature. The order data is compressed and base64-encoded.
HVS	Used to add VEU cancelation. The order data is compressed and base64-encoded.

Pending Signatures

Depending on the configuration settings defined in an offer, multiple signatories might have to sign the order to process the order data. If an order is submitted for processing without all the required signatures, EBICS Client does not process the order. Pending signature notifications are sent to the mailboxes of the concerned signatories requesting them to attend to the order. The Pending signature page opens for a user with orders pending to be submitted or signed when the user logs in to the EBICS Client dashboard interface. The following statuses are displayed against an order:

Pending, Sign
The order is pending and needs to be signed.

Pending, Submit
All the required signatures are obtained, and the order needs to be submitted.

Hardware key token for personal signature

EBICS Client supports Hardware Signature Module (HSM) for Electronic Signatures (ES). If hardware security key for ES is configured for a user, the Electronic Signature window opens after the user clicks on **Sign**. The user has to provide appropriate hardware security key information to sign the order. Currently, EBICS Client supports Hardware Signature Module using 3SKey only. You must enable Java version 1.6_24 or greater in the browser for the Hardware Signature Module applet to open.

Recovering Transactions

Order recovery is an important feature of EBICS Client. For an upload transaction, order processing error might occur at the bank's end. For download transactions, response processing errors might occur at the subscriber or customer's end. Apart from the processing errors, transport errors might also occur.

Recovery mechanism requires the transaction ID of the EBICS transaction in question, and is based on the definition of transaction recovery points:

- For upload transactions, the recovery point is the last transaction step of the transaction whose EBICS request is successfully sent to the bank system and whose EBICS response is successfully transmitted. The recovery point is determined by the state of the transaction in the bank system.
- For download transactions, there might be several recovery points. The recovery points are the transaction steps of the transaction whose EBICS request is successfully received by the bank and whose EBICS response is successfully transmitted.

When transport or processing errors occur, a recovery point is used to continue the transaction from the transaction step that follows the recovery point in a sequence.

When transmitting transaction data, any of the following errors might occur:

- Transport error
- Processing error
- Time-out error

EBICS Client stores the status of each segment that is successfully sent or received from the bank. If any of the errors occur when 'n' segments are successfully sent or received, the client starts the recovery from (n+1)th segment.

EBICS Banking Server maintains a recovery counter and also the maximum number of recovery attempts allowed. The counter is incremented after each attempt. If the maximum number of attempts is reached and recovery is unsuccessful, then the server cancels the recovery process and fails the entire transaction.

Searching transactions and viewing reports

Users can search for orders and view the summary of the orders on EBICS Client dashboard interface based on one of the following parameters or a combination of the following parameters:

- Search location: Live tables, Archive tables

  Note: Recent orders are stored in live tables and archived orders are stored in restore tables.
- Order start and end date
- Order start and end time
- Bank ID (Host ID)
- Partner name
- Order ID
- Status: All, Success, Failed, In Progress, Pending at Server, Pending at Client
- Order type
- File format
- Permission type: Submitter, Signer
- User ID: Only an EBICS Client admin and EBICS Client operator can use this parameter

If an EBICS Client admin invokes the search, orders submitted by all the users in the system are displayed. If an EBICS Client user invokes the search, self submitted orders are displayed in the search result. The search result is displayed in a tabular format. It can be sorted in ascending or descending order. The search result can also be refreshed periodically by specifying refreshing time. You can click the order ID link to view the order information. The Order summary details page is divided into two sections: **Order data** and **Order details.**

The Order data section provides the following information about the selected order:

- Order ID
- Order Type
- File format
- Number of signatures (Signatures required to submit the order)
- Start date and time
- Last activity date and time

- Completion date and time
- Partner name
- User ID
- Bank ID (HostID)
- Status of the order
- Document (The order payload) - The order document link is displayed only for the EBICS Client user. Click on the link to view the payload (for upload and download technical orders) or the order request XML (for other order types).

The Order details section has three tabs:

Order event
Provides information about events pertaining to an order, such as, data compressed, data encoded, pending tasks created for submitter, and so on.

Activities
Provides information about the activities pertaining to an order, such as, Pending at client for signature, Submit action by submitter, and so on. The activities can be in one of the following states:

- In progress
- Completed
- Failed

Activities are not generated for INI, HIA, and HPB order types.

Pending Signatures
Lists the users whose signatures are pending for the selected order.

Tracking EBICS transactions

EBICS Client generates events related to orders. All events are predefined and metadata is populated in the system. A user cannot define the events. The events are logged against an order ID and there are no stand-alone events in EBICS Client.

Users can search for events and view the event details related to orders on EBICS Client dashboard interface based on one of the following or a combination of the following parameters:

- Search location: Live tables, Archive tables

  Note: Recent orders are stored in live tables and archived orders are stored in archive tables.
- Event start and end date
- Event start and end time
- Event type: All, Info, Warning, Error, Critical

If an EBICS Client admin invokes the search, events related to all the orders in the system are displayed. If an EBICS Client user invokes the search, events related to self submitted orders are displayed. The search result is displayed in a tabular format and can be sorted in an ascending or descending order. The search result can also be refreshed periodically by specifying a time interval to display the updated list of orders. You can click the required event ID link to view complete information about the event. The Event summary details page is divided into two sections: **Event details** and **Order details**.

The Event details section provides the following information about the selected event:

- Event code
- Event name
- Description of the event
- Event type
- User ID
- Timestamp

The Order details section provides the following information about the order associated with the event:

- Order ID
- Order type
- Document (Clicking the document icon displays the payload document)
- Partner name
- User ID
- Bank ID (Host ID)

Event Types

Events are classified as follows:

Info Provides information about events. For example, Encryption successful.

Warning
A warning message. For example, order received with warnings.

Error An event indicating an error condition. For example, delivery to bank failed.

Critical
An event indicating a critical condition. For example, the EBICS Client database is not functioning.

Restore Tables

Archiving transaction data, such as, order, event, and pending signature-related data protects critical data. Archiving conserves database disk space, and file system disk space (when using documents on disk), in turn improving the efficiency of EBICS Client.

An EBICS Client admin can configure a lifespan for transaction data on Sterling B2B Integrator. After the transaction data exceeds its lifespan, the Backup business process service moves the data into a physical media. The data can be restored later to the restore tables. The Backup business process service can be run either by schedule or manually. Archiving transaction data is a resource-intensive activity. Therefore, it is recommended to perform the activity during off-peak hours.

The Restore business process service restores archived transactional data from physical media to a restored data location, where it can be searched and viewed.

Chapter 4. Configuration Requirements

The Configuration requirements chapter defines the elements that have to be configured on EBICS Client to transact with a bank. For detailed configuration information, see the *EBICS Client User Guide.*

Prerequisite Configuration on Sterling B2B Integrator

Before using the EBICS Client, configure the following entities on Sterling B2B Integrator.

1. Check in the public keys shared by the bank to the Certificate Authority (CA) store or create a self-signed certificate with SHA256 hash algorithm.
2. Create an identity record for the partner, indicating the partner as the base identity.
3. Create a user account.
4. Configure an adapter that enables you to send and receive files and invokes the EBClientOrderPreProcess business process. For example, configure a File System Adapter to invoke the EBClientOrderPreProcess business process. EBClientOrderPreProcess business process is used to validate the payload received from a technical user and to extract the folder containing the payload or metadata and post the payload data to appropriate mailboxes.
5. Configure the following mail boxes and associate them with each user:
   - EBClientOrderMetadata (preconfigured)
   - Download (Inbox)
   - Upload (Outbox)
6. Verify that the EBClientMailboxArrivedMessage business process is associated with the EBClientOrderMetadata Mailbox. The automatic routing rule triggers the business process to perform automated functions, such as notifying an interested party about an incoming message in the mailbox.
7. Ensure that the MailboxEvaluateAllAutomaticRulesSubMin Schedule is enabled. The schedule periodically evaluates the routing rule to ensure proper functioning of the routing rule.

User Configuration and Initialization

Configuring an existing Sterling B2B Integrator user as an EBICS Client user is the first step towards starting transaction with the bank. It includes specifying values for attributes, such as, user type, certificate type, and so on. Three types of users can be configured on EBICS Client; EbicsClient Admin, EbicsClient Operator, EbicsClient User. Another user type, technical user can also be configured on the client. EBICS Client supports X.509 and RSA Keys standards to define digital certificates. For more information about user types, see the *User Types* topic. User configuration attributes vary based on the user type and certificate type. It is not required to configure Certificate and key related information for an EBICS Client admin or EBICS Client operator.

After an existing user is configured as an EBICS user, the user transmits the public certificates to the financial institution 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 the user is configured and assigned to a partner, the status of the user is New. If the user sends only the INI request to the 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 in the bank system and Ready in EBICS Client. The user generates the INI and HIA letters with the hash value of the keys using the EBICS Client dashboard interface, manually signs them and mails the letters to the bank. When the bank receives the initialization letters of INI and HIA, it verifies the hash values in the letters against the bank database. After successful validation, the status of the user is set to Ready, indicating that the user can now transact with the bank.

Bank Configuration

A bank is the primary entity in an EBICS transaction. It hosts the server with which a partner and users associated with the partner can perform EBICS transactions. The details of the bank configuration include:

- Bank ID (Host ID)
- Bank Name
- Bank URL
- Is RSA preferred
- Bank contact information
- E-mail address
- Public keys of authentication and encryption certificates
- Key versions of authentication and encryption certificates

An integral part of the bank configuration is the bank ID or host ID. After a contract is signed, the bank shares the bank ID or host ID together with the URL of the bank with the partner. An EBICS Client admin creates a bank profile using the information shared by the bank with the partner. If the bank URL uses a secure HTTP protocol, then an HTTPS certificate is required. The HTTPS certificate for the bank is created on Sterling B2B Integrator and configured using EBICS Client.

The bank status is set to New until the public identification and authentication and encryption bank keys are validated. Bank key validation includes the following steps:

1. An EBICS Client user submits an HPB order type to download the public bank keys.
2. After successful validation of the user's authentication and identification keys, the bank sends an HPB response. The HPB response contains the public bank keys. The keys are stored in the database of EBICS Client and hash value of the keys is generated.
3. The bank provides hash values of the public part of the keys to the user through a channel independent of EBICS. For example, a portal, mail, or the website of the bank.
4. The user copies the hash values from the portal and validates the bank keys using the EBICS Client dashboard interface.

5. The hash values shared by the bank are compared with the internally generated hash values. If the hash values match, the status of the bank is set to Activated. If the hash values of the bank and user do not match, the user is prompted to revalidate the bank keys.

After successful validation, the status of the bank changes to Activated, indicating that the partner and its associated users can now transact with the bank.

File Format Configuration

The format or type of file that is uploaded or downloaded is identified by the File Format attribute. An order type can have zero or more file formats. A file format contains the following attributes:

Country Code
Code of the country in which the file format is supported.

Supported order types
FUL (Upload), FDL (Download), and other order types.

File formats for FUL and FDL are based on the SWIFTNet request type. For more information about SWIFTNet, see http://www.swift.com/. The first part of the file format name must be one of the elements listed in the following table. The elements indicate the type of transaction.

Table 5. File format name element

Element	Description
pain.	Payment Initiation
camt.	Cash Management
tsrv.	Trade Services
tsmt.	Trade Services Management

Offer Configuration

An offer is a super set of order types and file formats. An offer is associated with a bank ID to specify the possible order types and file formats that can be used when transacting with the bank. Offers provide the advantage of grouping many bank transactions and handling them together. An offer contains the following attributes:

- Name: offer name
- Bank ID: The bank ID with which the offer is associated
- Order type: supported order type
- File format: supported file format

Note: Only an EBICS Client admin can configure file formats and offers.

User Permission Configuration

User permissions define the offers, order types, and file formats, an EBICS Client user can process on EBICS Client. One of the following permission types can be assigned to a user:

Signer A signer can only sign an order, but cannot submit it.

Submitter

A submitter can submit an order after the designated signer or signers sign the order.

The following table provides information about the authorization levels that can be specified for an EBICS Client user.

Table 6. Authorization levels

Authorization level	Permission type	Description
E	Signer	Single signature. It is the strongest authorization level.
A	Signer	Primary signature
B	Signer	Secondary signature
T	Submitter	Transport signature. Transport signatures are not used for authorization of bank-technical orders, but for authorized submission to the bank system.

If the electronic signature value is set to 1, then a single signature of E or A authorization level is required to process an order. If the ES value is set to 2, then a combination of E or A and B is required to process the order. ES value is set to 0 in case of key management orders.

Note: EBICS specification does not permit a combination of two secondary ES authorization levels (that is authorization level B) for processing an order.

Chapter 5. Order Submission

Order submission entails the transmission of orders to the bank system from the EBICS Client. Based on whether an order is uploaded to the bank system or downloaded from the bank system, orders are classified as Upload orders and Download orders. Each order has to pass through different transaction phases during order processing.

Upload Order

A user sends an upload (FUL) request to the bank. FUL is a bank-technical upload order type. The upload transaction consists of the following stages:

- Order initialization
- Order processing

Order initialization

A user initiates an upload transaction with the bank by submitting an upload (FUL) order request. The EBICS Client Runtime component verifies the authorization level of the user and the number of required signatures to process the order. If the required criteria are met, then the order is processed and sent to the bank system. Else, the order is stored in the database and the Pending tasks page on EBICS Client dashboard interface is updated with the pending order details. When an EBICS Client user with pending orders (signing or submitting) logs in to the EBICS Client dashboard interface, the Pending tasks page opens prompting the user to sign or submit the order.

Order processing

Steps involved while processing an order submission which includes upload (FUL) bank-technical order type and key management order type are as follows:

1. Order Packaging Module for encryption, authentication signatures, authorization signatures, encoding, and compression is invoked.
2. Order ID is generated.
3. If the order data exceeds the specified 1-MB size, the order data is segmented. A transaction log is maintained to record the segmentation.
4. XML module to construct the EBICS request is invoked.
5. Activity logging for updating the order state is generated.
6. Appropriate mailbox storage is invoked for storing the order data.
7. Events are collected and logged in to the database during each activity.
8. The order data is submitted to the bank system.
9. The post processing response received from the bank is stored in the user's download mailbox.

Download Order

A user submits the download order type (FDL) to the bank. FDL order type is bank technical download order type. The download transaction consists of the following stages:

- Order initialization

- Order processing
- Acknowledgement

Order initialization

A user initiates a download transaction with the bank by submitting a download (FDL) order request. The EBICS Client Runtime component verifies the authorization level of the user. If the required criteria are met, then the order is processed and sent to the bank system.

Order processing

Steps involved while processing an order response received from the bank system are as follows:

1. The response received from the bank is stored in the download mail box for the user.
2. Order Response Processor component is invoked to process the response.
3. The Response Processor performs the following tasks:
   a. Invokes the Order Unpacking Module for decompression, decoding, and decryption.
   b. Invokes Order Concatenation if the response is segmented.
   c. Invokes Activity Logging for updating the order state.
   d. Collects and logs events in to the EBICS Client database during each activity.

Acknowledgement

After receiving the last segment of the order data from the bank, the client initiates the last phase, the acknowledgement request, to indicate that the data transfer is successful. If the bank receives a positive acknowledgement (receipt code=0) from the client, 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 client, the bank retains the downloaded messages in the user's download mailbox.

Order Packaging

The order data is packaged according to the specified signing, compression, encryption, and encoding settings. For example, if the order type is FUL, the FULPackingHandler is invoked.

Order packaging also involves generation of unambiguous order IDs. EBICS Client allocates a unique order ID based on the bank, user ID, and the order type. The client generates the order ID as per EBICS specifications.

- An order ID is a four character alphanumeric ID.
- The first character is an alphabet. An EBICS Client user can specify the first character of the order ID.
- The second, third, and fourth characters of the order ID are alphanumeric in an ascending order (A-Z or 0-9).

Order Data Segmentation

As per EBICS specifications for data transfer, the size of a compressed file, encrypted, and encoded order data must be less than or equal to 1-MB. If the size exceeds 1 MB after compression, encryption and encoding, then the order data is segmented such that each segment does not exceed the fixed 1-MB size. The segments are then transmitted sequentially in a consecutive order in individual EBICS messages.

Order Unpacking

Order unpacking involves decoding, decryption, decompression, and verification of the order data. It also involves logging orders for which the security operations have failed and the reasons for failure.

Segment Concatenation

The recipient system (server or client) executes the algorithmic computations in reverse order to recover the original order data. The data segments are sequentially appended, decoded, decrypted, and expanded to obtain the original order data.

Index

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 2013. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2013.

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