

Sterling Integrator[®]

EBICS Banking Server Module Services

Version 5.1

Sterling Commerce
An IBM Company

Contents

- EBICS Order Authorization Service.....3**
 - EBICS Order Authorization Service.....3
- EBICS ES Packaging Service.....7**
 - EBICS ES Packaging Service.....7
- EBICS Order Processing Service.....11**
 - EBICS Order Processing Service.....11
- EBICS Order Streaming Service.....17**
 - EBICS Order Streaming Service.....17
- EBICS Compression Service.....21**
 - EBICS Compression Service.....21
- EBICS Encryption Service.....24**
 - EBICS Encryption Service.....24
- EBICS Encoding Service.....27**
 - EBICS Encoding Service.....27
- EBICS Signing Service.....30**
 - EBICS Signing Service.....30
- EBICS Subscription Manager Service.....34**
 - EBICS Subscription Manager Service.....34
- EBICS Server Service.....39**
 - EBICS Server Service.....39
- EBICS Export Certificate Service.....45**
 - EBICS Export Certificate Service.....45
- EBICS Import Certificate Service.....47**
 - EBICS Import Certificate Service.....47
- EBICS Reporting Service.....49**
 - EBICS Reporting Service.....49

EBICS Order Authorization Service

EBICS Order Authorization Service

The following table provides an overview of the EBICS Order Authorization Service:

System name	EBICS Order Authorization Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service handles incoming order requests for the bank-technical upload order type. If an order has obtained the number of signatures required, this service will forward 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.
Business usage	This is a system service used internally by EBICS. This service provides EBICS protocol handler support in Sterling Integrator.
Usage examples	You want to use this service together with EBICS Server Service to receive EBICS' bank technical order data with order type FUL from a subscriber. The order data will be processed, verified, and forwarded to the subscriber's upload mailbox.
Preconfigured?	This service is preconfigured as part of the system installation.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	<ul style="list-style-type: none">• HTTP Server Adapter for EBICS• EBICS Server Service• EBICS Compression Service

	<ul style="list-style-type: none"> • EBICS Encoding Service • EBICS Encryption Service • EBICS Signing Service • EBICS Order Streaming Service
Application requirements	None
Initiates business processes?	This service invokes EbicsOrderProcessing business process.
Invocation	This service is invoked inside the business process that is bootstrapped by the EBICS Server Service.
Business process context considerations	This service is provided out of the box inside the EBICSOrderAuthorizationProcessing business process. No configuration is required, and the parameters are taken from the process data during run time.
Returned status values	<p>Fatal - Nonrecoverable error</p> <p>Transient – Recoverable error</p> <p>Logic - Recoverable error</p> <p>Success</p> <p>Warning</p>
Restrictions	None
Persistence level	System Default
Debug	<p>To test this adapter, post a valid EBICS request message, using the HTTP Client, to the preconfigured EBICS HTTP Server Adapter.</p> <p>Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS.</p>

Configuring the EBICS Order Authorization Service

To configure the EBICS Order Authorization service, you must specify the settings for the following field in the Graphical Process Modeler (GPM):

Field	Description
Transaction ID	Required. The transaction ID relevant to an order. Default value is null.
HostId	Required. The host ID defined in the EBICS Banking Server Module and used in the transaction.

Business Process Examples

A business process, which is a flow of tasks, can either be edited in the GPM or in the BPML (Business Process Modeling Language) code. The following business process example in BPML format illustrates using the EBICS Order Authorization service to handle a request message:

```
<operation>
  <participant name=
"EBICSOrderAuthorizationService"/>
  <output message="handleRequest">
    <assign to="." from="*" />
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>
```

The following business process example in BPML format illustrates using the EBICS Order Authorization service to handle a response message:

```
<operation>
  <participant name=
"EBICSOrderAuthorizationService"/>
  <output message="handleResponse">
    <assign to="." from="*" />
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>
```

The following business process example in BPML format illustrates the way in which EBICS Order Authorization Service can be used:

```
<process name="Order Authorization">
  <sequence name="main">
    <operation name="set user token">
      <participant name="SetUserToken"/>
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <!-- handle incoming EBICS Order -->
    <operation>
      <participant name=
"EBICSOrderAuthorizationService"/>
      <output message="handleRequest">
        <assign to="." from="*" />
      </output>
      <input message="Xin">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

```

    <!--internal processing by invoking a
subprocess -->
    <!--The subprocess name will be dynamically
determined based on Order Type
in the ProcessData -->
    <!--The invoke mode will be dynamically
determined based on order Type
in the ProcessData -->
    <operation name="Invoke Sub Process Service">
        <participant name="InvokeSubProcessService"/>
        <output message="Xout">
            <assign to="INVOKE_MODE" from=
"/ProcessData/OrderAuthInfo/InvokeBPMMode/text()"/>
            <assign to="WFD_NAME" from=
"/ProcessData/OrderAuthInfo/InvokeBPName/text()"/>
            <assign to="." from="*" />
        </output>
        <input message="Xin">
            <assign to="." from="*" />
        </input>
    </operation>
    <!-- handle EBICS response -->
    <operation>
        <participant name=
"EBICSOrderAuthorizationService"/>
        <output message="handleResponse">
            <assign to="." from="*" />
        </output>
        <input message="Xin">
            <assign to="." from="*" />
        </input>
    </operation>
</sequence>
</process>

```

EBICS ES Packaging Service

EBICS ES Packaging Service

The following table provides an overview of the EBICS Electronic Signature (ES) Packaging service:

System name	EBICS ES Packaging Service
Graphical Process Modeler (GPM) categories	All Services
Description	Use this service either to pack or unpack EBICS ES.
Business usage	Use this service to either pack or unpack key information that is used when signing and verifying the ES.
Usage examples	A business process that is a part of EBICS requires access to either unpack key information or pack key information with its response.
Preconfigured?	This service must be installed and deployed before it is invoked.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	None
Invocation	This service is invoked from a business process.
Business process context considerations	Use this service to either pack or unpack an ES.

Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System Default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Output from a Business Process to EBICS ES Packaging Service

The following parameters must be passed to the ES Packaging Service when it is being invoked, with the value of the output message type set to `Unpackage`.

Parameter	Description
HostId	Required. The host ID defined in the EBICS Banking Server Module and used in the transaction.
UserId	Required. The user ID used in the transaction.
PartnerId	Required. The partner ID used in the transaction.
OrderType	Required. The order type used in the transaction, for example, INI, HIA.
ProtocolVersion	Required. The supported EBICS protocol version, for example, H003.
SymmetricKeyId	Required. The document ID containing the symmetric key.
ESDocId	Required. The document ID of an ES. This document contains the packaged SignatureData element.
TransactionId	Required. The ID of the EBICS transaction.
WfId	Required. The workflow ID associated with the original workflow initiated in the EBICS request.

The following parameters must be passed to the ES Packaging service when it is being invoked, with the value of the output message type set to `Package`.

Parameter	Description
SignatureValue	Required. The calculated signature value that has been base64 encoded.
TransactionId	Required. The ID of the EBICS transaction.
WfId	Required. The workflow ID associated with the original workflow initiated in the EBICS request.

Output from EBICS ES Packaging Service to a Business Process

The following parameter must be passed to the business process when it is being invoked, with the value of the output message type set to Unpackage.

Parameter	Description
UserSignatureData	Required. The XML element containing information about an ES.

The following parameters must be passed to the business process when it is being invoked, with the value of the output message type set to Package.

Parameter	Description
ESDocId	Required. The document ID of ES. This document contains the SignatureData element.
SymmetricKeyId	Required. The document ID containing the symmetric key.

Business Process Examples

The following example business process illustrates the options required when attempting to unpack an ES.

```
<process name="getESUnpacked">
  <sequence>
    <operation name="ESInfo">
      <participant name="EBICSESPackagingService"/>
      <output message="outMsg">
        <assign to="." from="*" />
        <assign to="action" from="'Unpackage'"/ >
        <assign to="ESPackedDocId">neith::1100909</assign>
      </output>
      <input message="inMsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

The following example business process illustrates the use of the UserSignatureData element that is returned by the service:

```
<UserSignatureData>
  <OrderSignatureData>
    <SignatureVersion>A005</SignatureVersion>
    <SignatureValue>ABCD</SignatureValue>
    <CertId>neith::0090909</CertID>
  </OrderSignatureData>
  <OrderSignatureData>
    <SignatureVersion>A005</SignatureVersion>
    <SignatureValue>ABCD</SignatureValue>
    <CertID>neith::1180909</CertID>
  </OrderSignatureData>
</UserSignatureData>
```

The following example business process illustrates the options required when packing an ES:

```
<process name="getESUnpacked">
  <sequence>
    <operation name="ESInfo">
      <participant name="EBICSESPackagingService"/>
      <output message="outMsg">
        <assign to="." from="*" />
        <assign to="action">Package</assign>
        <assign to="SignatureValue">dfdsf4454==</assign>
      </output>
      <input message="inMsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

EBICS Order Processing Service

EBICS Order Processing Service

The following table provides an overview of the EBICS Order Processing Service:

System name	EBICS Order Processing Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service performs EBICS transaction and subscriber retrieval, and packing and unpacking of encrypted symmetric keys.
Business usage	This service is used as a central point to retrieve the information used by other services as part of the Order Data Processing activities.
Usage examples	Use this in a business process, which is a part of EBICS Order Data processing that requires information to perform tasks such as, signing, compression, encryption, and Base64 encoding.
Preconfigured?	This service must be installed and deployed before it is invoked. No configuration parameters are required.
Requires third party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	None

Invocation	This service is invoked from a business process.
Business process context considerations	The service is executed in a business process when you require the transaction details associated with the Primary Document, Symmetric Key creation or retrieval, and transaction status updating.
Returned status values	Fatal - Non-recoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System Default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Output from EBICS Order Processing Service to Business Process

The following table contains the parameters passed from the EBICS Order Processing service to a business process when it is invoked, with the value of the output message type set to `getTrxDetails`.

Parameter	Description
HostId	Required. The host ID of the system.
PartnerId	Required. The partner ID of the system.
UserId	Required. The user ID of the system.
ProtocolVersion	Required. The supported EBICS protocol version, for example, H003.
CommSessionId	Required. The session ID that is used as part of the DMI logging. Ensures that all the processes are tracked under the same session.
Order Type	Required. Specifies the order type. Valid values are INI, HIA, PUB, HCA, HCS, HPB, HPD, HEV, SPR, FUL, and FDL.
FlowDirection	Required. Indicates the direction of the order data. Valid values are Upload and Download.
OrderDataType	Required. Indicates the type of order data. Valid values are System and Technical.

Parameter	Description
ResponseType	Required. Indicates the response type. Valid values are Sync and Async.
CompressReqd	Required. Indicates if compression is required for Download flows or decompression is required for Upload flows. Valid values are True and False.
EncryptReqd	Required. Indicates if encryption is required for Download flows or decryption is required for Upload flows. Valid values are True and False.
SignReqd	Required. Indicates if signing is required for Download flows or verification is required for Upload flows. Valid values are True and False.
SignVersion	Optional. Indicates the version used in signing and verifying signatures.
EncryptedDocId	Optional. The document ID of the encrypted key. This value is extracted from the DataEncryptionInfo element found in the Upload message.
ESDocID	Optional. The document ID of the Electronic Signature (ES). This document contains the packaged SignatureData element found in the Upload message.
HostEncrPrivKeyId	Required. Host Private Certificate ID used in decrypting the transaction key to retrieve the symmetric key.
HostEsPrivKeyId	Required. Host Private Certificate ID used in signing messages.
PtnerEncrPubKeyId	Optional. Partner's Public Certificate ID used in encrypting messages.
WfId	Required. The workflow ID that is used to launch the initial business process that processes the transaction. This workflow ID is used when inserting records into a table that requires archive and restore capabilities to ensure that related records are archived and restored as a group.

The following table contains the parameter passed from the EBICS Order Processing Service to a business process when it is invoked, with the value of the output message type set to `getEncryptKey`.

Parameter	Description
SymmetricKeyId	Required. ID of the document containing the symmetric key.

The following table contains the parameter passed from the EBICS Order Processing service to a business process when it is invoked, with the value of the output message type set to `setEncryptKey`.

Parameter	Description
EncryptedDocId	Required. The ID of the document containing the encrypted symmetric key.

Output from Business Process to EBICS Order Processing Service

The following table contains the parameter passed from a business process to the EBICS Order Processing service when it is invoked, with the value of the output message type set to `getTrxDetails`.

Parameter	Description
Primary Document	Required. The primary document containing the order data.

The following table contains the parameters passed from a business process to the EBICS Order Processing service when it is invoked, with the value of the output message type set to `getEncryptKey`.

Parameter	Description
EncryptedDocId	Required. The document ID containing the encrypted symmetric key.
HostEncrPrivKeyId	Required. The certificate ID of the host's private key used for encryption.
HostEncrPubKeyId	Required. The certificate ID of the host's public key. Used for validation of the hash value sent by the subscriber.
PtnerEncrPubKeyId	Required. The certificate ID of the subscriber's public key used for encryption.

The following table contains the parameters passed from a business process to the EBICS Order Processing service when it is invoked, with the value of the output message type set to `setEncryptKey`.

Parameter	Description
PtnerEncrPubKeyId	Required. The certificate ID of the subscriber's public certificate used to encrypt the symmetric key.
TransactionId	Required. The transaction ID that is retrieved based on the body name of the Primary Document. This transaction ID is the key for all transaction-related records.
WfId	Required. The workflow ID of the transaction. Used to manage the archival process.
SymmetricKeyId	Required. ID of the document containing the symmetric key.

The following table contains the parameters passed from a business process to the EBICS Order Processing service when it is invoked, with the value of the output message type set to `setStatus`.

Parameter	Description
Status	Required. Indicates the processing status. Valid values are Success and Error.
TransactionId	Required. The transaction ID that is retrieved based on the body name of the Primary Document. This transaction ID is the key for all transaction-related records.
Wfld	Required. The workflow ID of the transaction. Used to manage the archival process.
FlowDirection	Required. The direction of the order flow. Valid values are Upload and Download.
UncompressedSize	Optional. The size of the uncompressed document in bytes.
OrderDataType	Required. The type of order. Valid values are System and Technical.

Business Process Examples

The following example business process illustrates the options required when retrieving transaction and subscriber details. It assumes that the packaged order data is in the primary document:

```
<process name="getTrxDetails">
<sequence>
<operation name="TrxInfo">
<participant name="EBICSOrderProcessingService"/>
<output message="outMsg">
<assign to="." from="*" />
<assign to="action" from="'getTrxDetails'"/ >
</output>
<input message="inMsg">
<assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</process>
```

The following example business process illustrates the process to unpack a document containing the decrypted symmetric key:

```
<process name="getEncryptedKey">
<sequence>
<operation name="DecryptKey">
<participant name="EBICSOrderProcessingService"/>
<output message="outMsg">
```

```

<assign to="." from="*" />
<assign to="action" from="'getEncryptedKey'"/ >
<assign to="EncryptedDocId">neith::1100909</assign>
</output>
<input message="inMsg">
    <assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</process>

```

The following example illustrates the process to pack a document containing the encrypted symmetric key.

```

<process name="setEncryptedKey">
<sequence>
<operation name="EncryptKey">
<participant name="EBICSOrderProcessingService"/>
<output message="outMsg">
<assign to="." from="*" />
<assign to="action" from="'setEncryptedKey'"/ >
<assign to="SymmetricKeyId">neith::2211909</assign>
</output>
<input message="inMsg">
    <assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</process>

```

The following example illustrates how to set the processing status.

```

<process name="setStatus">
<sequence>
<operation name="setStatus">
<participant name="EBICSOrderProcessingService"/>
<output message="outMsg">
<assign to="." from="*" />
<assign to="Status" from="'SUCCESS'"/ >
</output>
<input message="inMsg">
    <assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</process>

```


EBICS Order Streaming Service

EBICS Order Streaming Service

The following table provides an overview of the EBICS Order Streaming Service:

System name	EBICS Order Streaming Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service performs EBICS packing and unpacking of order type data using the pipeline functionality in Sterling Integrator.
Business usage	This service is used to pack and unpack an EBICS message in a streamed manner prior to delivering the message to the endpoint.
Usage examples	Use this service in a business process that is a part of the EBICS Order Data Processing and that requires packing and unpacking of the payload to be done through streaming.
Preconfigured?	This service must be installed and deployed before it is invoked. No configuration parameters are required.
Requires third party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	None
Invocation	This service is invoked from a business process.

Business process context considerations	You must be familiar with the internal service (workflow and business process parameters) that invoked this service. Workflow parameters are the values that were passed into the internal service. Business process parameters are the values specified in the BPML code.
Returned status values	Fatal - Non-recoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System Default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Output from Business Process to EBICS Order Streaming Service

The following table contains the parameters passed from a business process to the EBICS Order Streaming service:

Parameter	Description
Mode	Optional. Indicates if the service is being invoked by the EBICS Banking Server Module or the EBICS client. The client mode of operation is reserved for future use. Valid values are Server and Client. Default value is Server. If you do not provide a value for this parameter, the default value, Server, is used.
FlowDirection	Required. Indicates the direction of the order data. Valid values are Upload and Download. Default value is Upload.
CommSessionId	Required. The comm session ID with which the service is associated.
TransactionId	Required. The ID of the EBICS transaction with which the service is associated.
WfId	Required. The workflow ID with which the service is associated.
SignReqd	Required. Indicates if signing or verification is required for the message. Valid values are True and False. Default value is False.

Parameter	Description
EncryptReqd	Required. Indicates if encryption or decryption is required for the message. Valid values are True and False. Default value is False.
CompressReqd	Required. Indicates if compression or decompression is required for the message. Valid values are True and False. Default value is False.
PipelineTimeout	Optional. The pipeline timeout value. Default value is 300 seconds.
UserSignatureData	Required if signature verification has to be done for an Upload message. The XML element containing information about an Electronic Signature.
SymmetricKeyld	Required if an upload message has to be decrypted. The document ID containing the secret key byte array.
Certld	Optional. The certificate ID.
CertAlias	Optional. Required if the value of certld is null.
SignatureVersion	Optional. The signature version for signing the message. Valid values are A005 and A006.

Output from EBICS Order Streaming Service to Business Process

The following table contains the parameters passed from the EBICS Order Streaming Service to a business process:

Parameter	Description
SymmetricKeyld	Required if a message has to be encrypted. The document ID containing the secret key byte array.
SignatureValue	Optional. Signature value in Base64 encoded format of the hash of the message.

Business Process Example

The following example business process illustrates using the EBICS Order Streaming Service:

```
<process name="unPackage">
  <sequence>

    <operation name="TrxInfo">
      <participant name="EBICSOrderStreamingService"/>
      <output message="outMsg">
        <assign to="." from="*" />
      </output message>
    </operation name="TrxInfo">
  </sequence>
</process name="unPackage">
```

```
</output>  
<input message="inMsg">  
  <assign to="." from="*"></assign>  
</input>  
</operation>  
  
</sequence>  
</process>
```

EBICS Compression Service

EBICS Compression Service

The following table provides an overview of the EBICS Compression Service:

System Name	EBICS Compression Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service performs EBICS compression and decompression using zlib in pipeline mode.
Business usage	The EBICS Compression Service performs a part of the packing and unpacking of order data.
Usage examples	A business process is called by the EBICS Order Processing Service to either compress or decompress payload data.
Preconfigured?	This service must be installed and deployed before it is invoked. No configuration parameters are required.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	Use this service in a business process to either compress or decompress order data.

Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Output from Business Process to EBICS Compression Service

The following table contains the parameters passed from a business process to the EBICS Compression service:

Parameter	Description
action	Required. The action to be taken. Valid values are COMPRESS and DECOMPRESS.
CommSessionId	Required. The comm session ID to which the service belongs to, for example, 012345.
pipelineTimeout	Optional. The pipeline timeout value. Default value is 300 seconds.
TransactionId	Required. The ID of the EBICS transaction, for example, TransactionId012345.
WfId	Required. The workflow ID of the transaction, for example, WfId012345.
mode	Optional. Indicates if the service is being invoked by the EBICS Banking Server Module or the EBICS client. Valid values are Server and Client. The default value is Server. If you do not provide a value for this parameter, the default value, Server, is used.

Business Process Examples

The following example business process illustrates payload compression:

```
<process name="ebicscompression">
<sequence>
<operation>
<participant name="EBICSCompressionService"/>
```

```
<output message="compressRequest">
<assign to="." from="*" />
<assign to="action" from="'COMPRESS'"/>
</output>
<input message="compressResponse">
<assign to="." from="*" />
</input>
</operation>
</process>
```

The following example business process illustrates payload decompression:

```
<process name="ebicsdecompression">
  <operation>
    <participant name="EBICSCompressionService"/>
    <output message="decompressRequest">
      <assign to="." from="*" />
      <assign to="action" from="'DECOMPRESS'"/>
    </output>
    <input message="decompressResponse">
      <assign to="." from="*" />
    </input>
  </operation>
</process>
```

EBICS Encryption Service

EBICS Encryption Service

The following table provides an overview of the EBICS Encryption Service:

System Name	EBICS Encryption Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service performs EBICS encryption and decryption of the order data using the AES-128 algorithm in pipeline mode.
Business usage	The EBICS Encryption Service performs a part of the packing and unpacking of order data.
Usage examples	A business process is called by the EBICS Order Processing Service to either encrypt or decrypt payload data.
Preconfigured?	This service must be installed and deployed before it is invoked. No configuration parameters are required.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	Use this service in a business process to either encrypt or decrypt order data.

Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Output from Business Process to EBICS Encryption Service

The following table contains the parameters passed from a business process to the EBICS Encryption Service:

Parameter	Description
action	Required. The action to be performed. Valid values are ENCRYPT and DECRYPT.
CommSessionId	Required. The comm session ID to which the service belongs to, for example, 012345.
pipelineTimeout	Optional. The pipeline timeout value. Default value is 300 seconds.
SymmetricKeyId	Optional if the value of the action parameter is Encrypt and required if the value of the action parameter is Decrypt. The ID of the document containing the secret key byte array, for example, sedna:node1:1200d56b272:400.
TransactionId	Required. The ID of the EBICS transaction, for example, TransactionId012345.
WfId	Required. The workflow ID of the transaction, for example, WfId012345.
mode	Optional. Indicates if the service is being invoked by the EBICS Banking Server Module or the EBICS client. Valid values are Server and Client. The default value is Server. If you do not provide a value for this parameter, the default value, Server, is used.

Output from EBICS Encryption Service to Business Process

The following table contains the parameters passed from the EBICS Encryption Service to a business process.

Parameter	Description
SymmetricKeyId	The ID of the document containing the secret key byte array if it is generated by this service, for example, sedna:node1:1200d56b272:400.

Business Process Examples

The following example business process illustrates payload encryption:

```
<process name="ebicsencryption">
  <sequence>
    <operation>
      <participant name="EBICSEncryptionService"/>
      <output message="encryptRequest">
        <assign to="." from="*" />
        <assign to="action" from="'ENCRYPT'"/>
      </output>
      <input message="encryptResponse">
        <assign to="." from="*" />
      </input>
    </operation>
  </process>
```

The following example business process illustrates payload decrypting:

```
<process name="ebicsdecryption">
  <operation>
    <participant name="EBICSEncryptionService"/>
    <output message="decryptRequest">
      <assign to="." from="*" />
      <assign to="action" from="'DECRYPT'"/>
    </output>
    <input message="decryptResponse">
      <assign to="." from="*" />
    </input>
  </operation>
</process>
```

EBICS Encoding Service

EBICS Encoding Service

The following table provides an overview of the EBICS Encoding Service:

System Name	EBICS Encoding Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service performs EBICS encoding and decoding using base64 method in pipeline mode.
Business usage	The EBICS Encryption Service performs a part of the packing and unpacking of order data.
Usage examples	A business process is called by the EBICS Order Processing Service to either encode or decode payload data.
Preconfigured?	This service must be installed and deployed before it is invoked. No configuration parameters are required.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	Use this service in a business process to either encode or decode order data.

Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Output from Business Process to EBICS Encoding Service

The following table contains the parameters passed from a business process to the EBICS Encoding service:

Parameter	Description
action	Required. The action to be performed. Valid values are ENCODE and DECODE.
CommSessionId	Required. The comm session ID to which the service belongs to, for example, 012345.
pipelineTimeout	Optional. The pipeline timeout value. Default value is 300 seconds.
TransactionId	Required. The ID of the EBICS transaction, for example, TransactionId012345.
WfId	Required. The workflow ID of the transaction, for example, WfId012345.
mode	Optional. Indicates if the service is being invoked by the EBICS Banking Server Module or the EBICS client. Valid values are Server and Client. The default value is Server. If you do not provide a value for this parameter, the default value, Server, is used.

Business Process Examples

The following example business process illustrates payload encoding.

```
<process name="ebicsencoding">
<sequence>
<operation>
<participant name="EBICSEncodingService"/>
```

```
<output message="encodeRequest">
<assign to="." from="*" />
<assign to="action" from="'ENCODE'"/>
</output>
<input message="encodeResponse">
<assign to="." from="*" />
</input>
</operation>
</process>
```

The following example business process illustrates payload decoding.

```
<process name="ebicsdecoding">
  <operation>
    <participant name="EBICSEncodingService"/>
    <output message="decodeRequest">
      <assign to="." from="*" />
      <assign to="action" from="'DECODE'"/>
    </output>
    <input message="decodeResponse">
      <assign to="." from="*" />
    </input>
  </operation>
</process>
```

EBICS Signing Service

EBICS Signing Service

The following table provides an overview of the EBICS Signing Service:

System Name	EBICS Signing Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service performs the EBICS signing and verification of order data on the SHA-256 digest computed in pipeline mode.
Business usage	The EBICS Signing Service performs a part of the packing and unpacking of order data.
Usage examples	A business process is called by the EBICS Order Processing Service to either sign or verify payload data.
Preconfigured?	This service must be installed and deployed before it is invoked. No configuration parameters are required.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	This service is invoked from a business process.

Business process context considerations	Use this service in a business process to either sign or verify order data.
Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Output from Business Process to EBICS Signing Service

The following table contains the parameters passed from a business process to the EBICS Signing Service:

Parameter	Description
action	Required. The action to be performed. Valid values are SIGN and VERIFY.
certAlias	Required if the value of certId is null. Sterling Integrator certificate alias used by the EBICS client if certId is not provided.
certId	Required. Certificate ID.
CommSessionId	Required. The comm session ID to which the service belongs to, for example, 012345.
pipelineTimeout	Optional. The pipeline timeout value. Default value is 300 seconds.
signatureValue	Required only if the value of the action parameter is Verify. Signature value in Base64 encoded format that must be provided for verification of order data. It is not required for signing.
signatureVersion	Required. EBICS signature version, for example, A005 and A006.
TransactionId	Required. The ID of the EBICS transaction, for example, TransactionId012345.

Parameter	Description
WfId	Required. The workflow ID of the transaction, for example, WfId012345.
trxSignId	Required if the value of mode is Server. ID of each EBICS signature.
mode	Optional. Indicates if the service is being invoked by the EBICS Banking Server Module or the EBICS client. Valid values are Server and Client. The default value is Server. If you do not provide a value for this parameter, the default value, Server, is used.

Output from EBICS Signing Service to Business Process

The following table contains the parameter passed from the EBICS Signing service to a business process.

Parameter	Description
signatureValue	Required only if the value of the action parameter is Sign. Signature value in Base64 encoded format that must be provided for signing of order data. It is not required for verification.

Business Process Examples

The following example business process illustrates payload signing:

```
<process name="ebicssigning">
  <sequence>
    <operation>
      <participant name="EBICSSigningService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action" from="'SIGN'"/>
        <assign to="certId" from="'certid'"/>
      </output>
      <input message="signResponse">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

The following example business process illustrates payload verification:

```
<process name="ebicssigning">
  <operation>
    <participant name="EBICSSigningService"/>
    <output message="verifyRequest">
```



```
        <assign to="." from="*" />
        <assign to="action" from="'VERIFY'"/>
<assign to="certId" from="'certid'"/>
  <assign to="signatureValue"
from="'joeRiPzfV78gwZdENC3tH7nZqwWFPkMqT3fBmyUfNlFCAGUlGwLnBChRhRFR
17jkg6IuRBhTDvhnmmIfl034XFbra6hzdyQicgdE3UzrDMAM5701RkrKfwueA0jdb8
BlBsm8ol5Ibe2PRDC10X5TMWsyfcR1LoxKvGsEGI/gXI='"/>
    </output>
    <input message="verifyResponse">
      <assign to="." from="*" />
    </input>
  </operation>
</process>
```

EBICS Subscription Manager Service

EBICS Subscription Manager Service

The following table provides an overview of the EBICS Subscription Manager Service:

System name	EBICS Subscription Manager Service
Graphical Process Modeler (GPM) categories	All Services
Description	Use the EBICS Subscription Manager service to perform the following tasks: <ul style="list-style-type: none">• Validate the keys on the Initialization letters for INI and HIA• Update the status of the user to Ready before exchanging the EBICS messages• Import or export EBICS profiles created through Subscription Manager
Business usage	Use this service to validate the keys on the INI and HIA initialization letters. On successful validation, the status of the user is updated to Ready. Use this service to import or export subscription manager data to or from the database.
Usage examples	Validate the keys that you received in the initialization letter.
Preconfigured?	This service is preconfigured as part of the system installation.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None

Application requirements	<ul style="list-style-type: none"> • The subscription manager master data must be available with the bank. • The EBICS Server must be configured. • The validation of initialization letters can be carried out only after the INI and HIA initialization letters have been sent to the bank.
Initiates business processes?	None
Invocation	This service is invoked from a business process.
Business process context considerations	Use this service to either pack or unpack an ES.
Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System Default
Debug	To test this service, run the EBICS Subscription Manager Service business process and verify if it completes successfully. The status report (for validation and initialization letters) and instance data (for import and export) display the test results.

Configuring the EBICS Subscription Manager Service

To configure the EBICS Subscription Manager Service, you must specify the settings for the following fields in the Graphical Process Modeler (GPM) or in the Business Processing Modeling Language (BPML):

Field	Description
Partner ID (partnerID)	Required for INI and HIA initialization letters. Partner ID of the customer.
User ID (userID)	Required for INI and HIA initialization letters. User ID of the subscriber.
Public Electronic Signature Key (INI) (esKey)	Required for validation of the hash value of the client certificates. To set the status of the user to Ready, the bank must receive the INI and HIA initialization letters. Initialization letter for INI contains the user's public signature key.

Field	Description
	The hash value is in hexadecimal, for example, 3C B0 19 66 C9 9C 6E 2C A5 BA 6A 2B 56 01 92 35 2A B4 91 53 E9 0B BA 34 C1 5E B5 9F 4A 64 F7.
Public Encryption Key (HIA) (encrKey)	<p>Required for validation of the hash value of the client certificates. To set the status of the user to Ready, the bank must receive the INI and HIA initialization letters.</p> <p>Initialization letter for HIA contains the user's public encryption key.</p> <p>The hash value is in hexadecimal, for example, 3C B0 19 66 C9 9C 6E 2C A5 BA 6A 2B 56 01 92 35 2A B4 91 53 E9 0B BA 34 C1 5E B5 9F 4A 64 F7.</p>
Public Identification and Authentication Key (HIA) (authKey)	<p>Required for validation of the hash value of the client certificates. To set the status of the user to Ready, the bank must receive the INI and HIA initialization letters.</p> <p>Initialization letter for HIA contains the user's public identification and authentication key.</p> <p>The hash value is in hexadecimal, for example, 3C B0 19 66 C9 9C 6E 2C A5 BA 6A 2B 56 01 92 35 2A B4 91 53 E9 0B BA 34 C1 5E B5 9F 4A 64 F7.</p>
Electronic Signature Key Hash Algorithm (esKeyHashAlgo)	Optional. Hash algorithm for the user's public electronic signature Key. Valid values are SHA256 and SHA1. SHA256 is the default value.
Encryption Key Hash Algorithm (encrKeyHashAlgo)	Optional. Hash algorithm for the user's public encryption key. Valid values are SHA256 and SHA1. SHA256 is the default value.
Identification and Authentication Key Hash Algorithm (authKeyHashAlgo)	Optional. Hash algorithm for the user's identification and authentication key. Valid values are SHA256 and SHA1. SHA256 is the default value.

Business Process Examples

The following example business process illustrates using the EBICS Subscription Manager Service to validate a user's initialization letters, both INI and HIA, using the default SHA256 hash algorithm:

```
<operation>
  <participant name=" EBICSSubscrMgrService"/>
  <output message="validateSubscrKey">
    <assign to="partnerID">partner002</assign>
    <assign to="userID">user003</assign>
      <assign to="esKey">D4 7A 24 27 5C 5F D8 0D 50 1B CF 28 C5
38 FE 1F 51 DD 24 8B 3E 5C 72 D5 CD 47 9D 82 79 0C EF 52</assign>
    <assign to="encrKey">B8 3C B0 19 66 C9 9C 6E 2C A5 BA 6A 2B 56 01
92 35 2A B4 91 53 E9 0B BA 34 C1 5E B5 9F 4A 64 F7</assign>
```

```

<assign to="authKey">9D 2D C0 AF 55 6E D4 D9 04 00 BB 23 AF C8
1B AB 91 A3 7A 2E 97 A9 31 6D D0 01 79 5F C6 D0 CD 54</assign>
  <assign to="." from="*" />
</output>
<input message="testing">
  <assign to="." from="*" />
</input>
</operation>

```

The following example illustrates importing Subscription Manager data:

```

<operation>
  <participant name=" EBICSSubscrMgrService" />
  <output message="importSubscrMgrInfo">
    <assign to="." from="*" />
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>

```

The following example illustrates exporting host related data. A user can call either the FileSystem Adapter or Mailbox Service to place a file in a mailbox.

```

<operation>
  <participant name=" EBICSSubscrMgrService" />
  <output message="exportSubscrMgrInfo">
    <assign to="." from="*" />
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>
<!-- to extract to filesystem location -->
<operation name="To extract to a filesystem location">
  <participant name="extractFSA" />
  <output message="extractFile">
    <assign to="Action">FS_EXTRACT</assign>
    <assign to="extractionFolder">/my/location</assign>
    <assign to="assignFilename">>true</assign>
    <assign to="assignedFilename">exported.xml</assign>
    <assign to="." from="PrimaryDocument" />
  </output>
  <input message="NOP" />
</operation>
<!-- or to add into a mailbox -->
<operation>
  <participant name="MailboxAdd" />
  <output message="AddExportedFileInfoMailbox">
    <assign to="." from="*" />
    <assign to="MailboxPath">myMailbox</assign>
    <assign to="MessageName">exported.xml</assign>
    <assign to="ExtractableCount">1</assign>
  </output>
</operation>

```

```
</output>  
<input message="inmsg">  
  <assign to="MessageAddResults" from="*" />  
</input>  
</operation>
```

EBICS Server Service

EBICS Server Service

The following table provides an overview of the EBICS Server Service:

System name	EBICS Server Service
Graphical Process Modeler (GPM) categories	All Services
Description	This 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 the subscriber.
Business usage	Use this service to handle EBICS request, response, and error messages according to the EBICS protocols. The business value of this service is to provide support for EBICS protocols in Sterling Integrator.
Usage examples	You want to use this service together with the EBICS Order Authorization Service to receive EBICS' bank technical order data with order type FUL from a subscriber. You can use this service to parse and verify the request message, and forward the order data to EBICS Order Authorization Service for further asynchronous processing.
Preconfigured?	This service is preconfigured as part of the system installation.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	<ul style="list-style-type: none">• HTTP Server Adapter for EBICS• EBICS Order Authorization Service• EBICS Compression Service

	<ul style="list-style-type: none"> • EBICS Encoding Service • EBICS Encryption Service • EBICS Signing Service • EBICS ES Packaging Service
Application requirements	You must configure the EBICS HTTP Server Adapter for this setup.
Initiates business processes?	For system order type and technical download order type, this service invokes the EbicsOrderProcessing business process. For technical upload order type, this service invokes the EbicsAuthOrderProcessing business process.
Invocation	This service is invoked inside a business process, which is invoked by the EBICS HTTP Server Adapter.
Business process context considerations	The configuration parameters are picked by the service during run time.
Returned status values	<p>Fatal - Nonrecoverable error</p> <p>Transient – Recoverable error</p> <p>Logic - Recoverable error</p> <p>Success</p> <p>Warning</p>
Restrictions	None
Persistence level	System Default
Debug	<p>To test this adapter, post a valid EBICS request message, using the HTTP Client, to the preconfigured EBICS HTTP Server Adapter.</p> <p>Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS.</p>

Configuring the EBICS Server Service

To configure the EBICS Server Service, you must specify the settings for the following fields in the Graphical Process Modeler (GPM) or in the Business Processing Modeling Language (BPML):

Field	Description
Maximum Idle Time (MaxIdleTime)	Optional. The maximum period (in minutes) allocated to complete the transaction before it is cancelled. Valid value is any positive integer. The default value is 60 minutes.
Maximum Open Transaction Per Subscriber (MaxOpenTransactionPerSubscriber)	Optional. The bank system can limit the number of open transactions with positive recovery counter per subscriber. Valid value is any positive integer. The default value is 0,

Field	Description
	which indicates that unlimited number of transactions can be open.
Replay Testing's Tolerance Period (ReplayTolerance)	Optional. The tolerance period (in minutes, from the current time) of EBICS Server before the request (with timestamp older than this period) is rejected with a suspected replay error code. Valid value is any positive integer. The default value is 360 minutes.
<Scheduler settings>	<p>Required. The business process runs a task to cancel idle transactions when they exceed the Maximum Idle Time period.</p> <p>If the transactions have been idle for more than the configured maximum idle time, the scheduler aborts the transactions. If the FDL transactions have been idle for more than the configured maximum idle time, the intermediate messages held during the idle transactions are restored to their original names.</p> <p>Scheduler will perform the housekeeping task to cancel idle transactions and delete expired Nonce records.</p> <p>Following are the valid values:</p> <ul style="list-style-type: none"> • Do not use schedule <p>If you select this field, the EBICS Server service does not start a business process and does not run on a schedule. It is recommended that you do not use this parameter.</p> • Run based on timer <p>Valid values are the hour and minutes at which the service must be run. If you select a time interval, the valid values are the hours and minutes pertaining to the interval. Add or delete selections as necessary. Specify schedule exclusions or date exclusions, if any. Indicate whether you want the service to run at startup. It is recommended that you use this parameter every two hours.</p> • Run daily <p>Valid values are the hour and minutes at which the service must be run daily. If you select a time interval, the valid values are the hour and minutes pertaining to the interval. Add or delete selections as necessary. Specify date exclusions, if any. Indicate whether you want the service to run at startup.</p> • Run based on day(s) of the week <p>Valid values are the days of a week, the hour, and the minutes that specify when the service must be run. If you select a time interval, the valid values are the hours and minutes pertaining to the intervals. Add or delete selections as necessary. Specify date exclusions, if any.</p>

Field	Description
	<ul style="list-style-type: none"> • Run based on day(s) of the month <p>Valid values are the days of a month, hour, and minutes that specify when the service must be run. If you select a time interval, the valid values are the hours and minutes pertaining to the intervals. Add or delete selections as necessary. Specify date exclusions, if any.</p>

Business Process Examples

The following example business process illustrates using the EBICS Server Service to handle a request message:

```
<operation>
  <participant name="EBICSServerService"/>
  <output message="handleRequest">
    <assign to="." from="*" />
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>
```

The following example business process illustrates using the EBICS Server Service to handle a response message:

```
<operation>
  <participant name="EBICSServerService"/>
  <output message="handleResponse">
    <assign to="." from="*" />
  </output>
  <input message="testing">
    <assign to="." from="*" />
  </input>
</operation>
```

The following example illustrates using the EBICS Server service:

```
<process name="handleEBICSRequest">
  <rule name="SubProcessRequired">
    <condition>EBICSRequestInfo/SubProcessRequired = 'Y'</condition>
  </rule>
  <sequence name="main">
    <operation name="set user token">
      <participant name="SetUserToken"/>
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

```

</operation>
<!-- handle EBICS request -->
<operation>
  <participant name="EBICSServerService"/>
  <output message="handleRequest">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
<choice name="SubProcess">
  <select>
    <case ref="SubProcessRequired" activity="Invoke Sub
Process Service"/>
  </select>
  <!-- internal processing by invoking a subprocess -->
  <!-- The subprocess name will be dynamically determined based
on Order Type in the ProcessData -->
  <!-- The invoke mode will be dynamically determined based
on order Type in the ProcessData -->
  <operation name="Invoke Sub Process Service">
    <participant name="InvokeSubProcessService"/>
    <output message="Xout">
      <assign to="INVOKE_MODE"
from="EBICSRequestInfo/InvokeBPMode/text()" />
      <assign to="WFD_NAME"
from="EBICSRequestInfo/InvokeBPName/text()" />
      <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
      <assign to="." from="*" />
    </output>
    <input message="Xin">
      <assign to="." from="*" />
    </input>
  </operation>
</choice>
<!-- handle EBICS response -->
<operation>
  <participant name="EBICSServerService"/>
  <output message="handleResponse">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
<operation name="HttpRespond">
  <participant name="HttpRespond"/>
  <output message="HttpRespondInputMessage">
    <assign to="doc-has-headers">false</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>

```

```

        </input>
    </operation>
    <onFault>
        <!-- On Fault, we will clear PrimDoc and
construct response message -->
        <sequence>
            <operation name="ReleasePrimDoc">
                <participant name="ReleaseService"/>
                <output message="outmsg">
                    <assign to="TARGET">/ProcessData/PrimaryDocument</assign>
                    <assign to="." from="*" />
                </output>
                <input message="inmsg" />
            </operation>
            <operation>
                <participant name="EBICSServerService"/>
                <output message="handleError">
                    <assign to="." from="*" />
                    <assign to="transactionID" from="EBICSRequestInfo/TransactionID/text()" />
                </output>
                <input message="Xin">
                    <assign to="." from="*" />
                </input>
            </operation>
            <operation name="HttpRespond">
                <participant name="HttpRespond"/>
                <output message="HttpRespondInputMessage">
                    <assign to="doc-has-headers">false</assign>
                    <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
                    <assign to="." from="*"></assign>
                </input>
            </operation>
        </sequence>
    </onFault>
</sequence>
</process>

```

EBICS Export Certificate Service

EBICS Export Certificate Service

The following table provides an overview of the EBICS Export Cert Service:

System Name	EBICS Export Cert Service
Graphical Process Modeler (GPM) categories	All Services
Description	Use this service for exporting certificates from the Sterling Integrator repository into the primary document in the Base64 encoding format. The certificates can be extracted and added to an external repository.
Business usage	Use this service to export the certificates present in Sterling Integrator to an external system. Use this service when you want to synchronize the certificates present in Sterling Integrator with an external database or system.
Usage examples	You want to export Sterling Integrator Trusted or CA certificates to your system for performing various operations, such as, encryption, verification, and SSL transport.
Preconfigured?	Yes
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	Import Cert Service
Application requirements	None
Initiates business processes?	No

Invocation	None
Business process context considerations	None
Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS .

Business Process Example

The following example business process illustrates exporting Sterling Integrator certificates to an external repository:

```
<process name="ebics_test_ExportCertService">
  <sequence>
    <operation name="Export Cert Service">
      <participant name="EBICSExportCert"/>
      <output message="EBICSExportCertTypeInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

EBICS Import Certificate Service

EBICS Import Certificate Service

The following table provides an overview of the EBICS Import Cert Service:

System Name	EBICS Import Cert Service
Graphical Process Modeler (GPM) categories	All Services
Description	Use this service for performing create and delete operations in the Sterling Integrator repository.
Business usage	Use this service to add certificates from an external repository to Sterling Integrator. You can also delete the expired or invalid certificates.
Usage examples	You want to import the certificates from an external certificate repository into Sterling Integrator.
Preconfigured?	Yes
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	Export Cert Service
Application requirements	None
Initiates business processes?	No
Invocation	None
Business process context considerations	None

Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	None
Persistence level	System default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS.

Business Process Example

The following example business process illustrates importing certificates from an external repository into Sterling Integrator:

```
<process name="ebics_testImportService">
  <sequence>
    <operation name="Import Cert Service">
      <participant name="TestEBICSCertImport"/>
      <output message="TestEBICSCertImportTypeInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```


EBICS Reporting Service

EBICS Reporting Service

The following table provides an overview of the EBICS Reporting service:

System name	EBICS Reporting Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service composes Payment Status Report (PSR).
Business usage	Use this service to generate a PSR report with every 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 subscriber's download mailbox.
Usage examples	Use this service for a business process that needs to generate a PSR report.
Preconfigured?	This service is pre-configured as part of the system installation.
Requires third-party files?	No
Platform availability	All the supported Sterling Integrator platforms.
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	This service is invoked from a business process.

Business process context considerations	You must be familiar with the internal service (workflow and business process parameters) that invoked this service. Workflow parameters are the values that were passed into the internal service. Business process parameters are the values specified in the BPML code.
Returned status values	Fatal - Nonrecoverable error Transient – Recoverable error Logic - Recoverable error Success Warning
Restrictions	Set the user token in the BPML code before invoking this service. Setting the user token in the BPML code ensures that the workflow executor has the right to insert the report into a mailbox.
Persistence level	System Default
Debug	Debug information for this service is located in Sterling Integrator. Navigate to Operations > System > Logs > EBICS and Operations > System > Logs > EBICS Server .

Output from Business Process to EBICS Reporting Service

The following table contains the parameters passed from the business process to the EBICS Reporting service:

Parameter	Description
trx_id	Required. Specify the EBICS transaction ID.
lang	Optional. The language to generate the value of AddtlStsRsnInf. Valid values are en (English) and de (Deutsch). The default value is en (English). If you do not provide a value for this parameter, the service will use the English version. If the value of the parameter is invalid or not a supported language code, English version will be used.

Output from EBICS Reporting Service to Business Process

The following table contains the parameter passed from the EBICS Reporting service to the business process:

Parameter	Description
EBICSReportInfo/StatusCode	Required. The status code of the result of the service. Valid values are OK and Error.

Business Process Examples

The following example business process illustrates using the EBICS Reporting service to generate a PSR request with node value of AddtlStsRsnInf in English. The user token must be set in the BPML before invoking the service. Setting the user token in the BPML ensures that the workflow executor has the right to insert the report into a mailbox. In the EBICSreportingService operation, the output message value is generatePSRRequest.

```
<process name="EBICS_ReportingService">
  <sequence>

    <operation name="set user token">
      <participant name="SetUserToken"/>
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>

    <operation name="EBICSReportingService">
      <participant name="EBICSReportingService"/>
      <output message="generatePSRReport">
        <assign to="." from="*" />
        <assign to="trx_id">xxxx</assign>
      </output>
      <input message="generatePSRReportResult">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

The following example business process illustrates using the EBICS Reporting service to generate a PSR request with node value of AddtlStsRsnInf in German. The user token must be set in the BPML before invoking the service. Setting the user token in the BPML ensures that the workflow executor has the right to insert the report into a mailbox. In the EBICSreportingService operation, the output message value is generatePSRRequest.

```
<process name="EBICS_ReportingService">
  <sequence>

    <operation name="set user token">
      <participant name="SetUserToken"/>
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>

  </sequence>
</process>
```

```

</operation>

<operation name="EBICSReportingService">
  <participant name="EBICSReportingService"/>
  <output message="generatePSRReport">
    <assign to="." from="*" />
    <assign to="trx_id">xxxx</assign>
    <assign to="lang " >de</assign>
  </output>
  <input message="generatePSRReportResult">
    <assign to="." from="*"></assign>
  </input>
</operation>

</sequence>
</process>

```

PSR Reports - Examples

The following example is of a PSR report when 'GrpSts' is set to 'RJCT':

```

<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.02">
  <pain.002.001.02>
    <GrpHdr>
      <MsgId>463353943D942C7E840110DA9D0494E5</MsgId>
      <CreDtTm>2009-07-07T03:29:37Z</CreDtTm>
      <InitgPty>
        <Id>
          <OrgId>
            <BIC>HOSTIDAA</BIC>
          </OrgId>
        </Id>
      </InitgPty>
    </GrpHdr>
    <OrgnlGrpInfAndSts>
      <OrgnlMsgId>0001</OrgnlMsgId>
      <OrgnlMsgNmId>pain.xxx.cfonb160.ict</OrgnlMsgNmId>
      <OrgnlCreDtTm>2009-03-01T00:00:00Z</OrgnlCreDtTm>
      <GrpSts>RJCT</GrpSts>
      <StsRsnInf>
        <AddtlStsRsnInf>Transmission successful</AddtlStsRsnInf>
      </StsRsnInf>
      <StsRsnInf>
        <AddtlStsRsnInf>Decryption Error</AddtlStsRsnInf>
      </StsRsnInf>
    </OrgnlGrpInfAndSts>
  </pain.002.001.02>
</Document>

```

The following example is of a PSR report when 'GrpSts' is set to 'RCVD':

```

<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.02">
  <pain.002.001.02>

```

```
<GrpHdr>
  <MsgId>33F60D9949576D6AE926CF3BD6BFF0BB</MsgId>
  <CreDtTm>2009-07-07T03:29:39Z</CreDtTm>
  <InitgPty>
    <Id>
      <OrgId>
        <BIC>HOSTIDAA</BIC>
      </OrgId>
    </Id>
  </InitgPty>
</GrpHdr>
<OrgnlGrpInfAndSts>
  <OrgnlMsgId>0008</OrgnlMsgId>
  <OrgnlMsgNmId>pain.xxx.cfonb160.ict</OrgnlMsgNmId>
  <OrgnlCreDtTm>2009-03-01T00:00:00Z</OrgnlCreDtTm>
  <GrpSts>RCVD</GrpSts>
</OrgnlGrpInfAndSts>
</pain.002.001.02>
</Document>
```

Copyright

Licensed Materials - Property of Sterling Commerce

© Copyright Sterling Commerce, an IBM Company 2000, 2010 All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by contract with Sterling Commerce

Additional copyright information is located on the Sterling Integrator 5.1 Documentation Library:

<http://www.sterlingcommerce.com/Documentation/SI51/CopyrightPage.htm>