BofA Global Foreign Currency Secured SOAP (Rate) Message Exchange High Level Design

Bank of America
Foreign Item Systems

October 2006

Version 1.2

Table of Contents

ARTIFACT REVISION HISTORY

Rev. #	**Date** mm/dd/yyyy	**Modification**	**Page #**	**Previous Page #**	**Action Taken**	**Addendum / New Page**	**Author**
0.1	08/04/2005	Baseline document published					FIS
0.2	09/15/2005	Introduce new field to B05, B07, S05,S07 D01 Added <repeatFlag> where needed					FIS
0.3	09/15/2005	Add security token element					FIS
0.4	10/05/2005	Added support for retail transactions					FIS
0.5	10/24/2005	SOAP message technical changes. Added Appendix 2 and 3					FIS
0.6	10/31/05	Clarification on Transaction delete for inventory transactions					FIS
1.1	06/10/06	Version 1.1 Baseline					FIS
1.2	10/10/06	Add Trans status listing					FIS

Overview

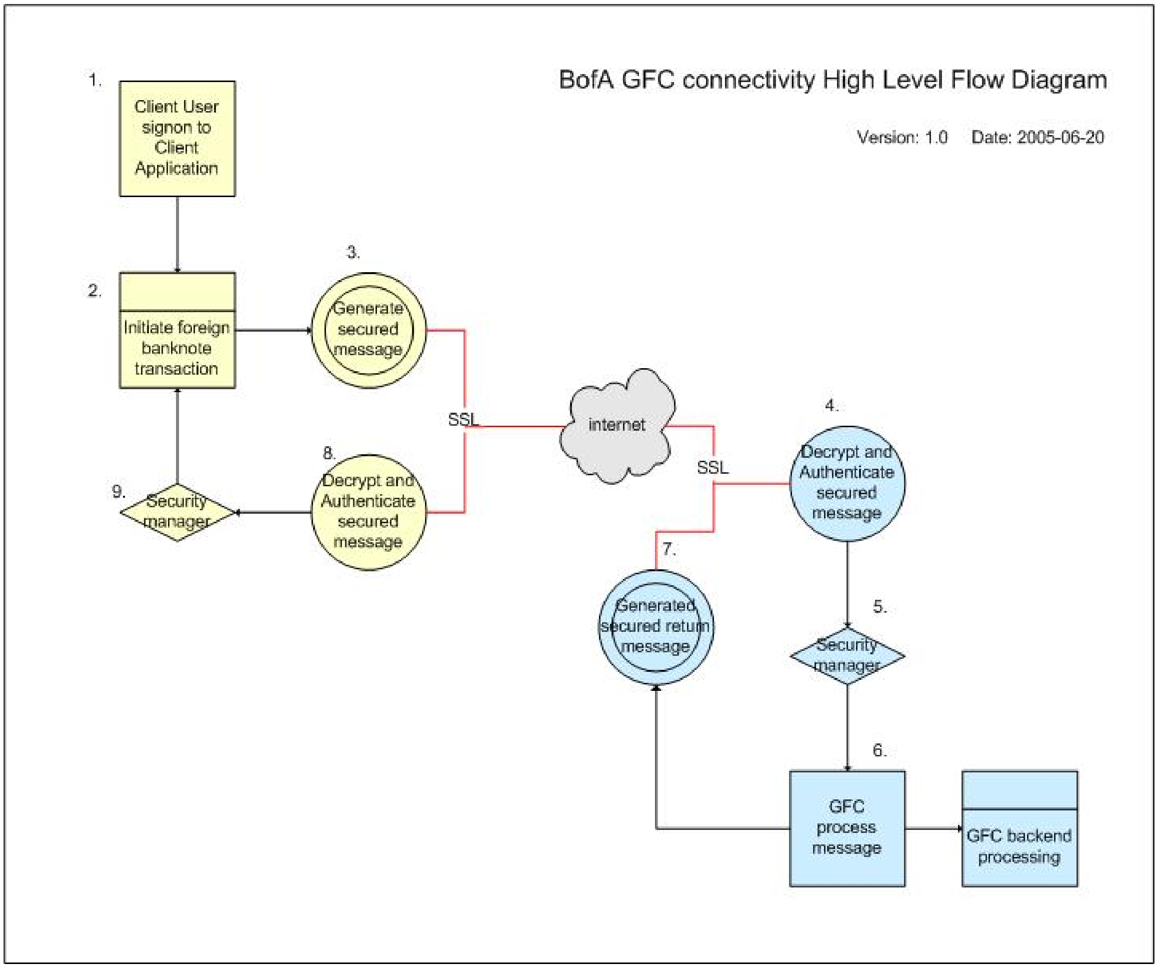
This document is a subset of the "**BofA Global Foreign Currency Secured SOAP Message Exchange High Level Design".** This document contains the details of "Quote" function.

Following is the high level flow of messages between BofA Global Foreign Currency (GFC) and Client Application:

1. Get Transaction Quote

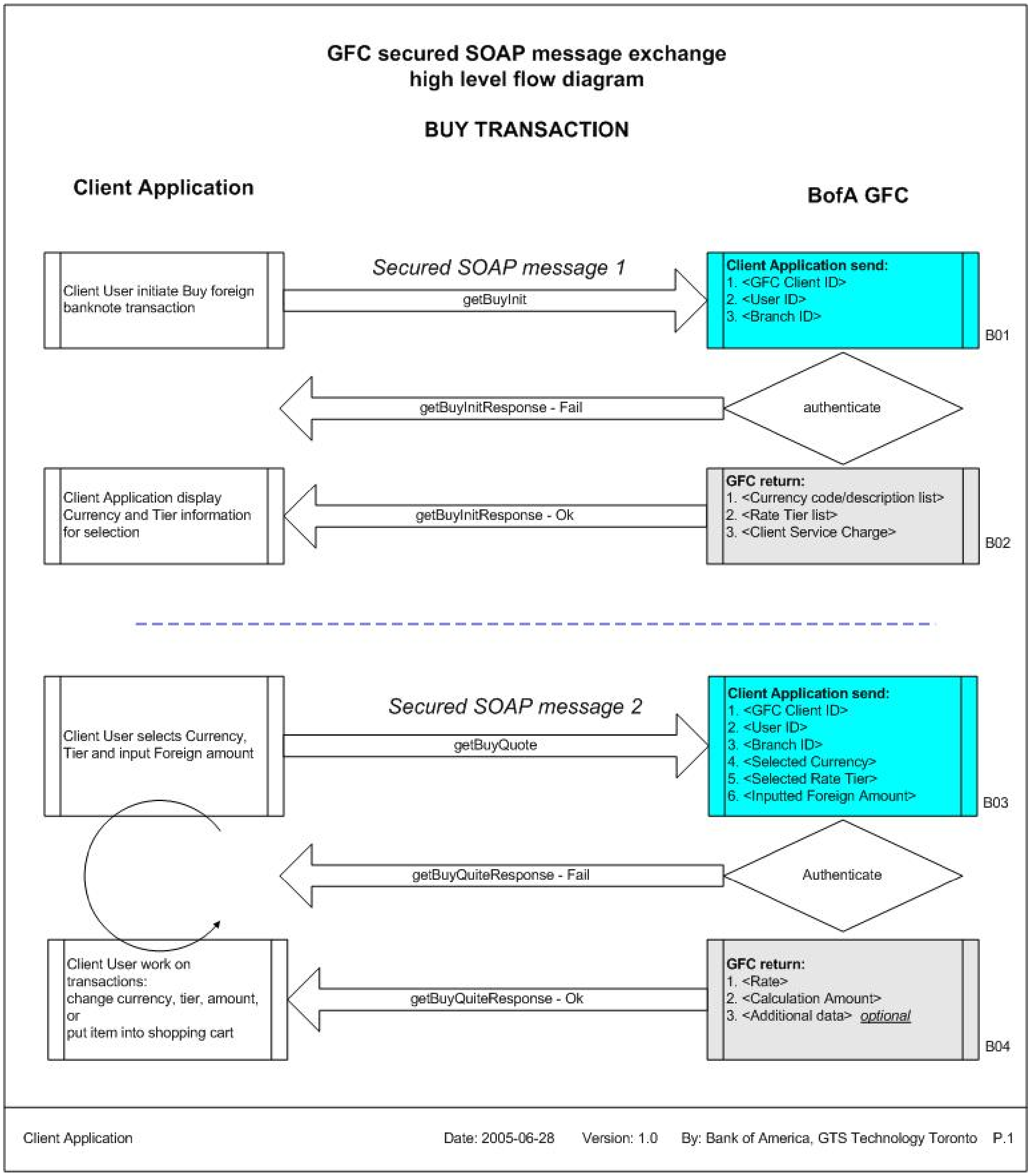
- Client user works on the foreign banknote transaction information (currency code, rate tire, transaction amount) and requests price quote.
- Client Application sends a message to GFC to request for "transaction quote information".
- The request message is sent in SOAP format and transported to GFC using https (128-bit SSL encryption with client digital certificate authentication) through the internet.
- GFC authenticates the message using the attached security token.
- GFC return "transaction quote information" to Client Application.
- The return message is sent in SOAP format and transported to Client Application using https (128-bit SSL encryption) through the internet.

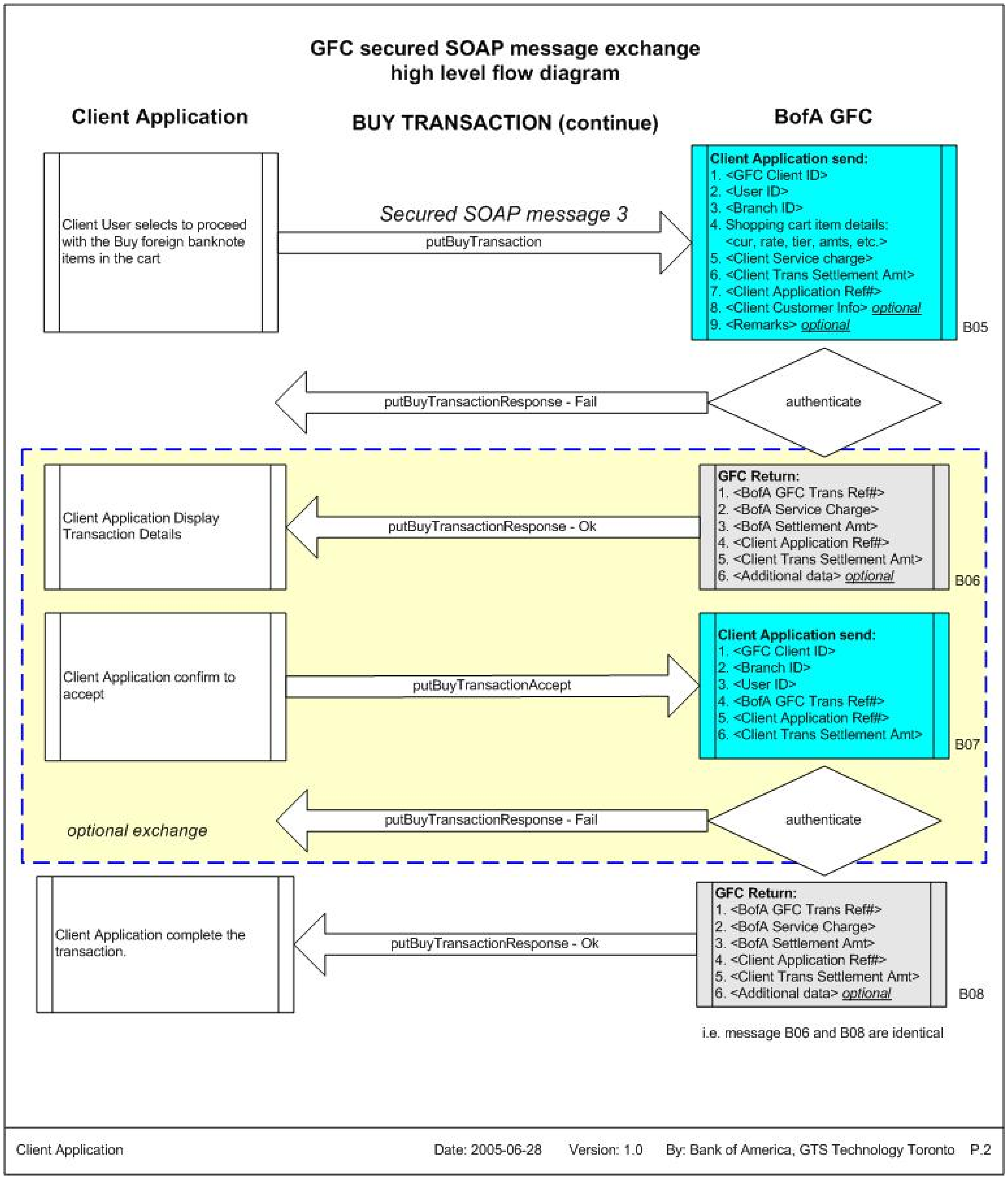
2. High Level Flow Diagram


BofA GFC connectivity High Level Flow Diagram
Version: 1.0 Date: 2005-06-20
1.
Client User signon to Client Application
2.
Initiate foreign banknote transaction
3.
Generate secured message
SSL
internet
SSL
4.
Decrypt and Authenticate secured message
5.
Security manager
6.
GFC process message
GFC backend processing
7.
Generated secured return message
8.
Decrypt and Authenticate secured message
9.
Security manager

Message Exchange Details

Buy Transaction


GFC secured SOAP message exchange
high level flow diagram
BUY TRANSACTION
Client Application
BofA GFC
Client User initiate Buy foreign banknote transaction
Secured SOAP message 1
getBuyInit
Client Application send:
1. <GFC Client ID>
2. <User ID>
3. <Branch ID>
B01
getBuyInitResponse - Fail
authenticate
Client Application display Currency and Tier information for selection
getBuyInitResponse - Ok
GFC return:
1. <Currency code/description list>
2. <Rate Tier list>
3. <Client Service Charge>
B02
Client User selects Currency, Tier and input Foreign amount
Secured SOAP message 2
getBuyQuote
Client Application send:
1. <GFC Client ID>
2. <User ID>
3. <Branch ID>
4. <Selected Currency>
5. <Selected Rate Tier>
6. <Inputted Foreign Amount>
B03
getBuyQuiteResponse - Fail
Authenticate
Client User work on transactions:
change currency, tier, amount,
or
put item into shopping cart
getBuyQuiteResponse - Ok
GFC return:
1. <Rate>
2. <Calculation Amount>
3. <Additional data> optional
B04
Client Application
Date: 2005-06-28
Version: 1.0
By: Bank of America, GTS Technology Toronto
P.1



GFC secured SOAP message exchange
high level flow diagram

Client Application

BUY TRANSACTION (continue)

BofA GFC

Client User selects to proceed with the Buy foreign banknote items in the cart

Secured SOAP message 3

putBuyTransaction

Client Application send:
1. <GFC Client ID>
2. <User ID>
3. <Branch ID>
4. Shopping cart item details:
   <cur, rate, tier, amts, etc.>
5. <Client Service charge>
6. <Client Trans Settlement Amt>
7. <Client Application Ref#>
8. <Client Customer Info> *optional*
9. <Remarks> *optional*

B05

putBuyTransactionResponse - Fail

authenticate

Client Application Display Transaction Details

putBuyTransactionResponse - Ok

GFC Return:
1. <BofA GFC Trans Ref#>
2. <BofA Service Charge>
3. <BofA Settlement Amt>
4. <Client Application Ref#>
5. <Client Trans Settlement Amt>
6. <Additional data> *optional*

B06

Client Application confirm to accept

putBuyTransactionAccept

Client Application send:
1. <GFC Client ID>
2. <Branch ID>
3. <User ID>
4. <BofA GFC Trans Ref#>
5. <Client Application Ref#>
6. <Client Trans Settlement Amt>

B07

optional exchange

putBuyTransactionResponse - Fail

authenticate

Client Application complete the transaction.

putBuyTransactionResponse - Ok

GFC Return:
1. <BofA GFC Trans Ref#>
2. <BofA Service Charge>
3. <BofA Settlement Amt>
4. <Client Application Ref#>
5. <Client Trans Settlement Amt>
6. <Additional data> *optional*

B08

i.e. message B06 and B08 are identical

Client Application Date: 2005-06-28 Version: 1.0 By: Bank of America, GTS Technology Toronto P.2

B03 – getBuyQuote

Second message exchange – designed to provide rate / settlement amount calculation.

- **<**msgName> -- Name Constant represents the name of the message
- <clientID> - unique GFC client ID for Client must be set up in GFC. If client ID presented not valid *getBuyQuoteResponse Fail* message returned.
- <branchID> – Client branch ID booking the transaction – will be validated against the branch file maintained in GFC. If branch ID presented not valid *getBuyQuoteResponse Fail* message returned.
- <tranType> - Transaction type INVENTORY / INVENTORY-MAIN / NON-INVENTORY describes the transaction type
- <userID> – For *NON-INVENTORY* transactions user identification not validated by GFC or if tranType not presented. For *INVENTORY* transactions userID must be a user attached to an acknowledged cash box in the branch presented in the <branch id field>. For *INVENTORY-MAIN* transactions cashbox 001 must exist in the branch.
- <currCode> – Currency code validated by GFC against currency list provided in *getBuyInitResponse*. If currency code presented not valid *getBuyQuoteResponse Fail* message returned.
- <spreadType> – validated by GFC against spread list provided in *getBuyInitResponse*. If spread type presented not valid *getBuyQuoteResponse Fail* message returned
- <foreignAmount> – must be numeric and > 0 (zero). If foreign amount presented not valid *getBuyQuoteResponse Fail* message returned.

XML for message B03

```
<getBuyQuote>
      <msgName>getBuyQuote</msgName>
      <clientID></clientID>
      <branchID></branchID>
      <tranType></tranType>
      <userID></userID>
      <currCode></currCode>
      <spreadType></spreadType>
      <foreignAmount></foreignAmount>
      <requestSecToken></requestSecToken>
      <reservedGFC></reservedGFC>
</getBuyQuote>
```

B04 – getBuyQuoteResponse

GFC return message. Contains the following data elements:

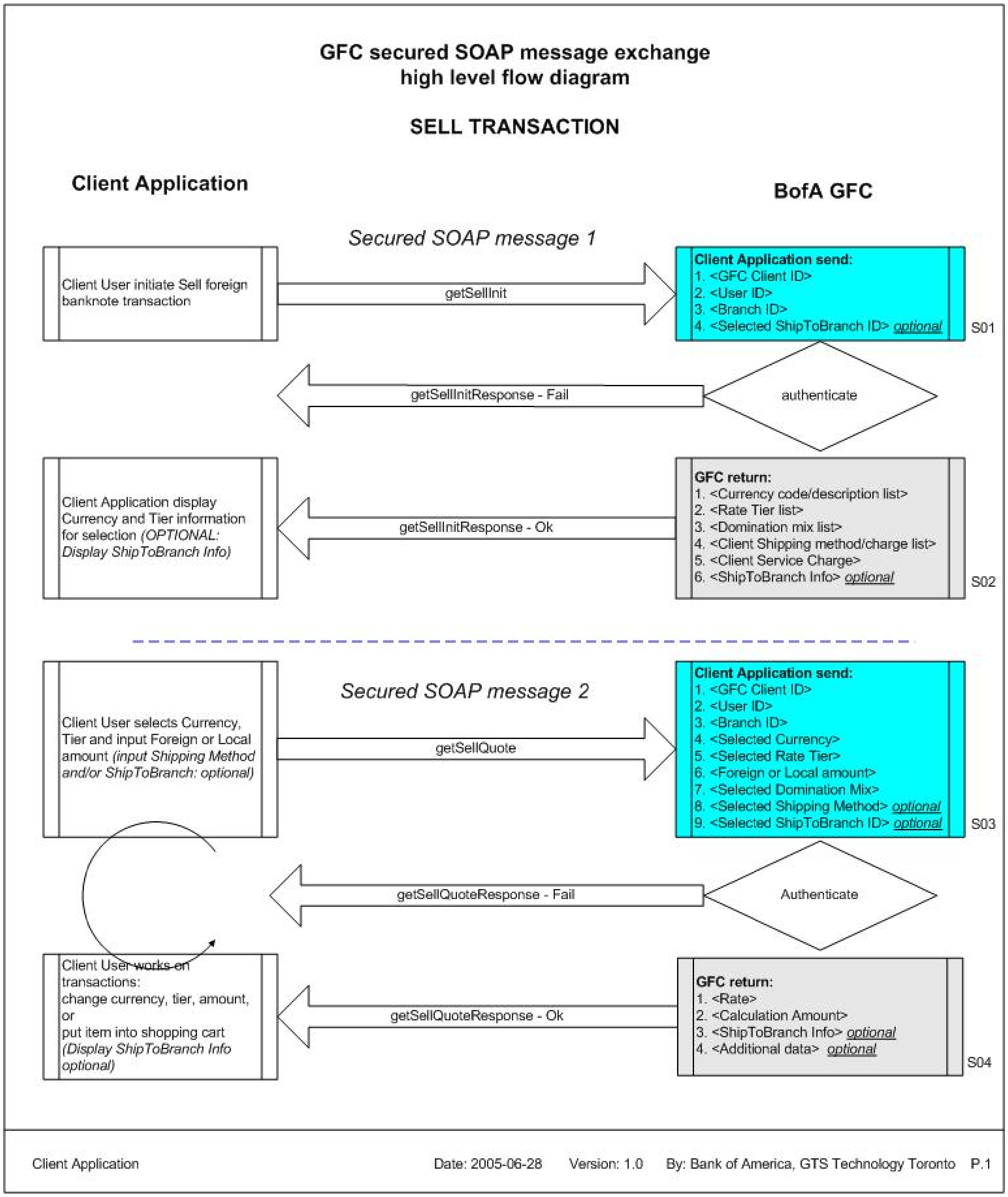
- **<**msgName> -- Name Constant represents the name of the message
- <msgResult> -- Ok or Fail
- <rate> – exchange rate used for buying the foreign currency. It is the buy rate as selected tier in direct terms.
- <localAmount> – Foreign amount requested x rate (half adjusted).
- <errMsgID> -- Error Message ID
- <errMsgText> -- Error Message Text

* the B03-B04 sequence will be repeated for each cart element.

XML for message B04

```
<getBuyQuoteResponse>
   <getBuyQuoteReturn>
      <msgName>getBuyQuoteResponse</msgName>
      <msgResult></msgResult>
      <ok>
            <rate></rate>
            <localAmount></localAmount>
      </ok>
      <fail>
            <errMsgID></errMsgID>
            <errMsgText></errMsgText>
      </fail>
      <responseSecToken></responseSecToken>
      <reservedGFC></reservedGFC>
   </getBuyQuoteReturn>
</getBuyQuoteResponse>
```

Sell Transaction



GFC secured SOAP message exchange
high level flow diagram

SELL TRANSACTION

Client Application

BofA GFC

Secured SOAP message 1

Client User initiate Sell foreign banknote transaction

Client Application send:
1. <GFC Client ID>
2. <User ID>
3. <Branch ID>
4. <Selected ShipToBranch ID> *optional*

Client Application display Currency and Tier information for selection *(OPTIONAL: Display ShipToBranch Info)*

GFC return:
1. <Currency code/description list>
2. <Rate Tier list>
3. <Domination mix list>
4. <Client Shipping method/charge list>
5. <Client Service Charge>
6. <ShipToBranch Info> *optional*

Secured SOAP message 2

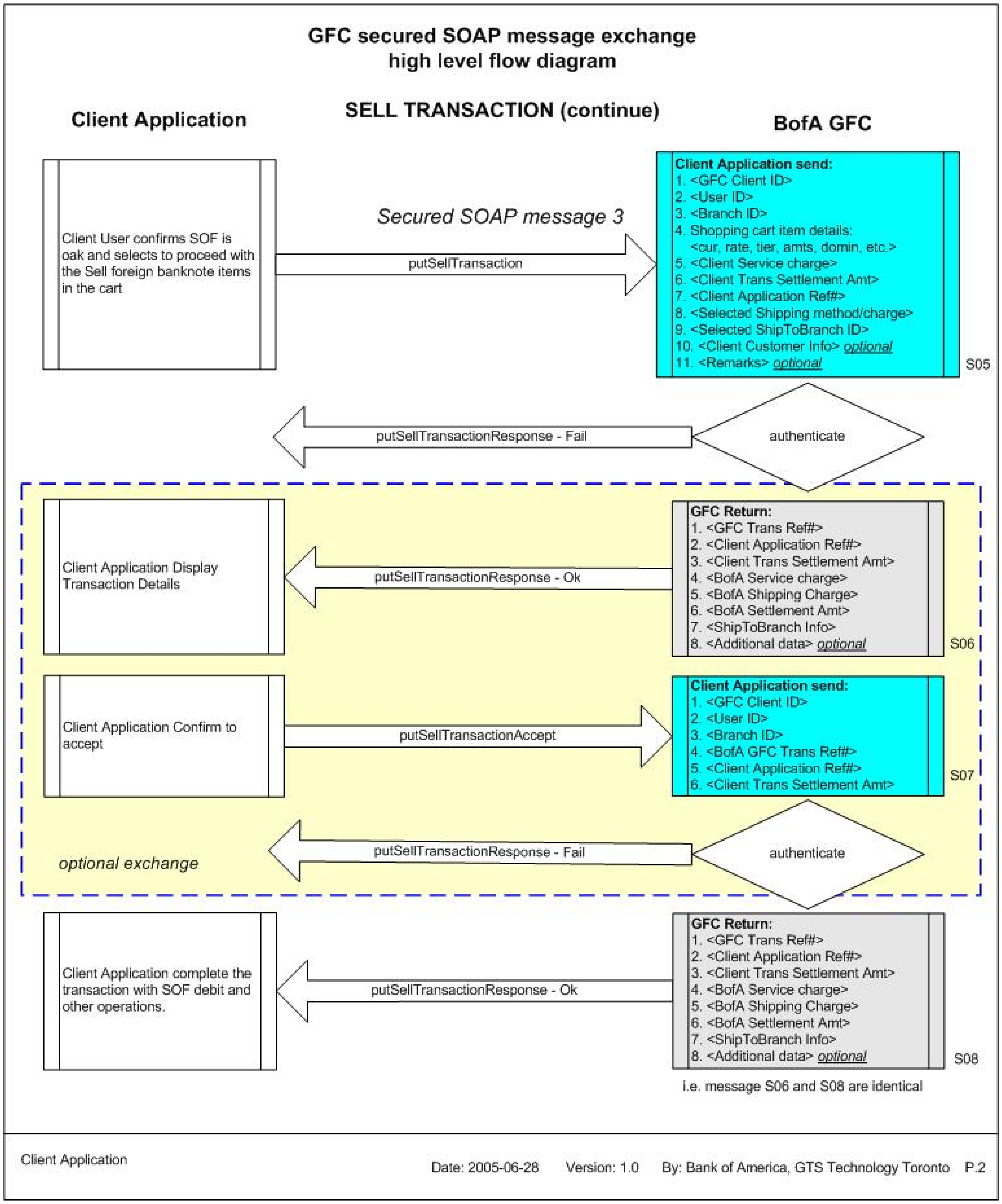
Client User selects Currency, Tier and input Foreign or Local amount *(input Shipping Method and/or ShipToBranch: optional)*

Client Application send:
1. <GFC Client ID>
2. <User ID>
3. <Branch ID>
4. <Selected Currency>
5. <Selected Rate Tier>
6. <Foreign or Local amount>
7. <Selected Domination Mix>
8. <Selected Shipping Method> *optional*
9. <Selected ShipToBranch ID> *optional*

Client User works on transactions:
change currency, tier, amount,
or
put item into shopping cart
(Display ShipToBranch Info optional)

GFC return:
1. <Rate>
2. <Calculation Amount>
3. <ShipToBranch Info> *optional*
4. <Additional data> *optional*

Client Application Date: 2005-06-28 Version: 1.0 By: Bank of America, GTS Technology Toronto P.1



GFC secured SOAP message exchange
high level flow diagram

SELL TRANSACTION (continue)

Client Application **BofA GFC**

Client User confirms SOF is oak and selects to proceed with the Sell foreign banknote items in the cart

Secured SOAP message 3

putSellTransaction

Client Application send:
1. <GFC Client ID>
2. <User ID>
3. <Branch ID>
4. Shopping cart item details:
   <cur, rate, tier, amts, domin, etc.>
5. <Client Service charge>
6. <Client Trans Settlement Amt>
7. <Client Application Ref#>
8. <Selected Shipping method/charge>
9. <Selected ShipToBranch ID>
10. <Client Customer Info> optional
11. <Remarks> optional

S05

putSellTransactionResponse - Fail

authenticate

Client Application Display Transaction Details

putSellTransactionResponse - Ok

GFC Return:
1. <GFC Trans Ref#>
2. <Client Application Ref#>
3. <Client Trans Settlement Amt>
4. <BofA Service charge>
5. <BofA Shipping Charge>
6. <BofA Settlement Amt>
7. <ShipToBranch Info>
8. <Additional data> optional

S06

Client Application Confirm to accept

putSellTransactionAccept

Client Application send:
1. <GFC Client ID>
2. <User ID>
3. <Branch ID>
4. <BofA GFC Trans Ref#>
5. <Client Application Ref#>
6. <Client Trans Settlement Amt>

S07

putSellTransactionResponse - Fail

authenticate

optional exchange

Client Application complete the transaction with SOF debit and other operations.

putSellTransactionResponse - Ok

GFC Return:
1. <GFC Trans Ref#>
2. <Client Application Ref#>
3. <Client Trans Settlement Amt>
4. <BofA Service charge>
5. <BofA Shipping Charge>
6. <BofA Settlement Amt>
7. <ShipToBranch Info>
8. <Additional data> optional

S08

i.e. message S06 and S08 are identical

Client Application Date: 2005-06-28 Version: 1.0 By: Bank of America, GTS Technology Toronto P.2

S03 – getSellQuote

This is the second Sell message exchange – designed to provide rate / foreign or settlement amount calculation.

- **<**msgName> -- Name Constant represents the name of the message
- <clientID> **-** unique GFC client ID for Client must be set up in GFC. If client ID presented not valid *getSellQuoteResponse Fail* message returned.
- <branchID> – Client branch ID booking the transaction – will be validated against the branch file maintained in GFC. If branch ID presented not valid *getSellQuoteResponse Fail* message returned
- <tranType> **-** Transaction type INVENTORY / INVENTORY-MAIN / NON-INVENTORY describes the transaction type
- <userID> – For *NON-INVENTORY* transactions user identification not validated by GFC or if tranType not presented. For *INVENTORY* transactions userID must be a user attached to an acknowledged cash box in the branch presented in the <branch id field>. For *INVENTORY-MAIN* transactions cashbox 001 must exist in the branch.
- <currency> – validated by GFC against currency list provided in *getSellInitResponse*. If currency code presented not valid *getSellQuoteResponse Fail* message returned.
- <spreadType> – validated by GFC against spread type list provided in *getSellInitResponse*. If rate tier presented not valid *getSellQuoteResponse Fail* message returned
- <foreignAmount> – if provided must be numeric. If provided must be the only amount provided for the message. If foreign amount presented not valid *getSellQuoteResponse Fail* message returned. For inventory transactions must be less then or equal to inventory on hand in the user's cashbox or cashbox 001 for INVENTORY-MAIN
- <localAmount> **-** if provided must be numeric and greater or equal to 1 (one). If provided must be the only amount provided for the message. If local amount presented not valid *getSellQuoteResponse Fail* message returned
- <denominationMix> **--** Validated by GFC against denomination mix list provided in *getSellInitResponse*. If selected denomination mix presented not valid *getSellQuoteResponse Fail* message returned.
- <shipMethod> (optional) If presented validated by GFC against shipping method list provided in *getSellInitResponse*. If selected shipping method presented not valid *getSellQuoteResponse Fail* message returned.
- <shipBranchID> (optional)* – Client branch ID where the order will be shipped. If received it will be validated against the branch file maintained in GFC. If ship to branch ID presented not valid *getSellQuoteResponse Fail* message returned (see *). Ship to branch information ignored for inventory transactions

XML for message S03

```
<getSellQuote>
      <msgName>getSellQuote</msgName>
      <clientID></clientID>
      <branchID></branchID>
      <tranType></tranType>
      <userID></userID>
      <currCode></currCode>
      <spreadType></spreadType>
      <foreignAmount></foreignAmount>
      <localAmount></localAmount>
      <denominationMix></denominationMix>
      <shipMethod></shipMethod>
      <shipBranchID></shipBranchID>
      <requestSecToken></requestSecToken>
      <reservedGFC></reservedGFC>
</getSellQuote>
```

S04 – getSellQuoteResponse

GFC return message if getSellQuote authenticated. Contains the following data elements:

- <msgName> -- Name Constant represents the name of the message
- <msgResult> -- Ok or Fail.
- <rate> used – exchange rate used for the sell of foreign currency. It is the sell rate for the selected tier in direct terms.
- <foreignAmount> ** (can be different from foreign amount requested as it will be rounded to be a multiple of the smallest denomination available for the currency in Bank of America GFC vaults). If not presented it is calculated using the local amount presented divided by the rate and rounded down to be a multiple of the smallest denomination for the currency available in the Bank of America GFC vault.
- <localAmount> = Foreign amount x rate (half adjusted), may be different from the local amount presented (see**)
- <shippingInfo> (name, address) – this data element is optional and will be returned only if Ship to branch ID received (see *)
- <shipCharge> this data element is optional and will be returned only if Shipping Method received
- <errMsgID> -- Error Message ID
- <errMsgText> -- Error Message Text

* the S03-S04 sequence will be repeated for each cart element.

XML for message S04

```
<getSellQuoteResponse>
   <getSellQuoteReturn>
      <msgName>getSellQuoteResponse</msgName>
      <msgResult></msgResult>
      <ok>
            <rate></rate>
            <foreignAmount></foreignAmount>
            <localAmount></localAmount>
                  <shippingInfo>
                  <shipBranchName></shipBranchName>
                  <shipBranchAddress>
                        <shipAdrLine1></shipAdrLine1>
                        <shipAdrLine2></shipAdrLine2>
                        <shipAdrLine3></shipAdrLine3>
                  </shipBranchAddress>
             </shippingInfo>
             <shipCharge></shipCharge>
      </ok>
      <fail>
            <errMsgID></errMsgID>
            <errMsgText></errMsgText>
      </fail>
      <responseSecToken></responseSecToken>
      <reservedGFC></reservedGFC>
   </getSellQuoteReturn>
</getSellQuoteResponse>
```

XML data description

This table represents all the variables identified so far for the data exchange between Client Application and GFC.
An attempt was made to assign error messages and validation rules for the variables presented. When a message is received the content of the message will be validated until the first error encountered. If an error is found the error message returned in the “Fail” message will contain only one error.

Variable Name	Description	Len *	Error message/ reason
branchID	Branch ID booking the transaction	10	**GFC0020**- Unable to determine Branch ID. Branch ID not setup on GFC for the Client ID. **GFC0300 –** Unable to determine the branch type **GFC0310 -** Branch "Inventory Main" needs to have acknowledged cash box 001
branchType		20	Returns the branch type INVENTORY/NON-INVENTORY as set up in GFC
currency X 120 maximum			
currCode	Currency Code	3	**GFC040** – Invalid Currency code. Currency code presented not a valid GFC currency.
currDesc	Description	35	
currGroup	Group (01 to 99) for optional sorting control	2	
customer			
custAcct	Customer account	35	
custAddress			
custAdrLine1	Address Line	35	
custAdrLine2	Address Line	35	
custAdrLine3	Address Line	35	
custEmail	E-mail address	70	
custFirstName	First name	35	
custLastName	Last name	35	
custNo	CustomerNumber	16	
custPhone			
custPhCnty	Country Code	5	
custPhArea	Area Code	5	
custPhNo	Phone No	11	
custPhExt	Extension	9	

clientID	Client ID assigned by Bank of America GFC	6 0	**GFC0010**- Unable to determine Client ID. Client not set up on GFC server or non WEB services client.
denominationMix	Denomination request	1	**GFC0180**- Unable to determine denomination Mix requested. Denomination Mix requested different from available denominations in denomination table setup on GFC.
errMsgText	Error Text	80	
errMsgID	Error ID	7	
foreignAmount	Foreign Amount	12 2	**GFC0101** – Foreign amount in error, amount not numeric or zero. **GFC0280 –** Foreign amount over inventory on hand for curr **GFC0320 –**Foreign Amount ordered must be a multiple of the smallest
GFCRefNumber	GFC reference number	9 0	**GFC220 –** GFC reference not found.
GFCServiceCharge	BofA GFC service charge	7 2	
GFCSettlementAmt	Amount to be paid by BofA GFC for buy transaction or Amount to be paid to BofA GFC for sell orders	12 2	
GFCShippingCharge	BofA GFC shipping charge	7 2	
GFCTranStatus	GFC transaction status	10	
GFCTranStsDate	GFC transaction status date in yyyy-mm-dd format	10	

GFCTranStsTime	GFC transaction status time in hh:mm:ss format	8	
GFCTranType	GFC transaction type: *BUYSTGBN=Buy in transit *SELORDBN=Sell Order *BUYRETBN= Buy into Inventory *SELRETBN= Sell from Inventory	10	
localAmount	Local (settlement) equivalent Amount	12 2	**GFC0100** – Local amount in error USD amount not numeric or zero. **GFC0110** – Local amount calculation error. Local amount not equal Foreign amount * Rate (half adjusted +/- tolerance (0.01). **GFC0210 –** Local amount requested to small to fill an order.
msgName	Name of the SOAP message (constant)		**GFC0005 –** Invalid msgName.
msgResult	Ok or Fail If Ok, the <ok> section will be filled and the <fail> section will be empty. If Fail, the <ok> section will be empty and the <fail> section will be filled.	4	

origRefNumber	Originator (Client Application) Reference number	16	**GFC0150** - Originator Reference number not presented. **GFC0160** - Originator Reference number already used. **GFC0230 –** Originator and GFC reference number presented do not match. **GFC0250** – Originator Reference number already accepted. **GFC0260** – Originator Reference not found for acceptance.
rate	Exchange Rate	12 8	**GFC0080** – Rate in error. Rate presented not numeric or zero.
remarks	Remarks	256	
repeatFlag	Repeat flag – designed to indicate if the initiating application is presenting the message for the first time or not	1	**GFC0240** – repeat flag value not Y/N
requestSecToken	Request Msg Security Token (see appendix for details)		
responseSecToken	Response Msg Security Token (see appendix for details)		

reservedGFC	Field reserved for GFC internal usage		
serviceCharge	Service charge	7 2	**GFC0060** – Service charge amount in error. Service charge presented not numeric.
settlementAmount	Settlement amount for the transaction	12 2	**GFC0120** – Settlement amount in error. Settlement amount not numeric or zero or Negative. **GFC0130** – Settlement amount presented different from settlement amount calculated Settlement amount not sum of LocalAmount +/- Service Charge + ShipCharge. **GFC0140 -** Settlement amount presented different from settlement amount on file for same Originator reference Number.
shipBranchAddress			
shipAdrLine1	Address Line	35	
shipAdrLine2	Address Line	35	
shipAdrLine3	Address Line	35	
shipBranchID	Ship to Branch ID	10	**GFC0170**- Unable to determine ship to Branch ID. Branch ID not setup on GFC.
shipBranchName	Branch Name	35	
shipCharge	Shipping Charge	7 2	**GFC0200**- Shipping Charge amount in error. Shipping charge amount presented not numeric.
shipMethod information X 20 maximum			
shipMethod	Shipping method	10	**GFC0190**- Shipping Method requested not available. Shipping Method requested different from available shipping methods.

spread X 20 maximum			
spreadName	Description	35	
spreadType	Spread type code	10	**GFC0050** – Invalid Spread Type. Spread Type presented not a valid GFC Spread Type.
tranBranchID	Transaction booked form branch ID	10	
tranCboxID	Transaction Booked form Cashbox	3	
tranType	Values: INVENTORY INVENTORY-MAIN NON-INVENTORY	20	**GFC0270** Unable to determine the transaction type
tranUserID	Transaction Booked by user ID	35	
tranSettled	Transaction Settled Flag	1	Y – transaction was settled N – transaction not settled yet
userCashbox	User has a cashbox flag Y / N	1	
userID	UserID booking the transaction	35	**GFC0030**- User ID not presented. User ID field not presented / all blanks or string presented less then User ID Minimum Length as per GFC client setup. **GFC0290** User id not authorized for inventory transactions

Notation:

- number = represent number of digits for an alphanumeric field (Ex. 35)
- number space number = represents numeric field; first number total length, second number positions after decimal (Ex. 12 2)

Appendix 1

requestSecToken (request message security token)

The purpose of requestSecToken is to provide message “authentication” and “non-repudiation” security. A security token will be generated and attached to each request message; the receiving side of the message will use the token to authenticate the message. The request message security token will contain the following XML data:

<timestamp><timestamp>	GMT Time stamp (e.g. 2005-09-06T16:55:34.127GMT)
<msgName></msgName>	The message name constant listed in the message body
<clientID></clientID>	Client ID listed in the message body
<branchID></branchID>	Branch ID listed in the message body
<userID></userID>	User ID listed in the message body
<randomStr></randomStr>	A 10 bytes random string to randomize the token data

1. Generate a security token with the above XML data
2. Encrypt the token using Advanced Encryption Standard (AES) encryption algorithm with 128bits symmetric secret[1] key (i.e. the secret key to be replaced every year on agreed schedule).
3. The encrypted data is then converted to Base-64 encoding format.

   Example of generated security token before and after encryption in Base-64 encoding format:

```
<requestSecToken>
   <token>
         <timestamp>2005-09-06T16:55:34.127GMT</timestamp>
         <msgName>getBuyInit</msgName>
         <clientID>12345</clientID>
         <branchID>main</branchID>
         <userID>user4567</userID>
         <randomStr>AyeH7i61Kd</randomStr>
   </token>
</requestSecToken>
```

```
<requestSecToken>
   cc1V//9/+mT2GerGLrCb2q7cur79S2RSx/g8V2CytHuO/KeGbGms88OKadFZB144z
   GSgDFg/g8V2CytHuO/KeGbGmsBM5uKvdN1yDxSiYBWXZ3e8rBByYa3N9PW1BaHFVO
   dTMOlc4m9yDxSiYBWXZ3e8MLpVyzhgSvR7gaoeJWU6Zs/g8V2CytHuO/KeGbGms5u
   KvdN1yDxSiYBWXZ3e8rBByYa3N9PW1BaHFVOdTMOlc4m9MLpVyzhGb/t6utrpIdga
   Vrpp/g8V2CytHuO/KeGbGmsPi1s1MN7oFn/CUbCH6V5FsQFDNOkpd8dULAtArgNw4
   I4I3YfHO8v/A==
</requestSecToken>
```

[1] Symmetric private key only known by Client and BofA designated body.

4. Upon receiving the message, the security token will be decrypted using the symmetric secret key. If security token not find, message will be rejected and Security Token error message (SEC9010) will be returned. If token find but decryption fails, message will be rejected and Security Token error message (SEC9020) will be returned.
5. If decryption is oak, the time stamp will be checked. If the time stamp is older than 10 minutes from the current GMT, the message will be rejected and Security Token error message (SEC9030) will be returned.
6. The security token msgName, clientID, branchID and userID contents will be checked against the same data in the message body. If any content doesn't match, the message will be rejected and Security Token error message (SEC9040) will be returned.

responseSecToken (response message security token)

A security token will be generated and attached to each response message (i.e. security token related error response will not contain responseSecToken). The response message security token contains the following XML data:

<timestamp><timestamp>	GMT Time stamp (e.g. 2005-09-06T16:55:34.127GMT)
<msgName></msgName>	The message name constant listed in the message body
<randomStr></randomStr>	A 10 bytes random string to randomize the token data

Example:

```
<responseSecToken>
   <token>
         <timestamp>2005-09-06T16:55:33.015GMT</timestamp>
         <msgName>getBuyInit</msgName>
         <randomStr>ACu98yRjuE</randomStr>
   </token>
</responseSecToken>


<requestSecToken>
   dTMOlc4m9yDxSiYBWXZBM5uKvdN1yDxSiYBWX3e8MLpVyzhgSvR7gaoeJWU6Zs/g8
   V2CytHuO/KeGbGms5uVrpp/g8V2CytHuO/KeGbGmsPi1s1MN7oFn/CUbCH6V5FsQF
   DNOkpd8dULAtArgNw4I4BM5uKvdN1yDxSiYBWX==
</requestSecToken>
```

Appendix 2

Attention: All messages listed here are for example and demo purposes only.

SOAP Message Example (getBuyQuote)

```
<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header />
<soapenv:Body>
  <getBuyQuote xmlns="http://api.ws.ba.com">
        <msgName>getBuyQuote</msgName>
        <clientID>84597</clientID>
        <branchID>1023</branchID>
        <tranType>NON-INVENTORY</tranType>
        <userID>gfcsupport2</userID>
        <currCode>AUD</currCode>
        <spreadType>STANDARD</spreadType>
        <foreignAmount>200.0</foreignAmount>
        <requestSecToken>v9m3sQSGDNPar7YVx8UNm+iiMJOmvulwrV/sFWN
        Cojzwyl+aLrt4MRwto78U2pq7n5A3avcvQB30WJ6Xb0lhI67AwHJfNchI
        BGzbxGt7ZqOhKau2ch+N0EydODo2CKPnNIYw6NXGvFC5xp3wikthUY0
        KTUso1j1ce72923HFNyKtkYUix5TPn1yT9eeqvIqrAy6wLxB3EAXtuJow
        AC3UU6FUvmH3G3Em/Kq85+AszgCzctwSmjx/I0Tuc1IuPzzojDloqaX0
        ZZELJ7qIrN1dXA==</requestSecToken>
  </getBuyQuote>
</soapenv:Body>
</soapenv:Envelope>
```

SOAP Message Example (getBuyQuoteResponse)

```
<?xml version="1.0" encoding="utf-8" ?>
 <soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Header />
 <soapenv:Body>
   <getBuyQuoteResponse xmlns="http://api.ws.ba.com">
        <getBuyQuoteReturn>
            <fail xsi:nil="true" xmlns="http://bean.ws.ba.com" />
            <msgName
                xmlns="http://bean.ws.ba.com">getBuyQuoteResponse</msgN
                ame>
            <msgResult xmlns="http://bean.ws.ba.com">Ok</msgResult>
            <ok xmlns="http://bean.ws.ba.com">
                <localAmount>146.2</localAmount>
                <rate>0.731</rate>
            </ok>
            <responseToken
                xmlns="http://bean.ws.ba.com">ybq6wip+l+yTlrEjBqPfLKJieH
                xr/CX7CRD27hpfs+LIzf6CSx9YPk7vW2nEm2iTaeWNl3Yb4yEEg
                VLa7EFyKX41QTmbaSe0qIBDBGHv/hb+ghkaJsGfsCwdK4rtD8Q
                QMq5/O3jR18PFpYHMEBqD9NtGoShth1wOuUnC69Wl+hajpshq
                Esxo7PQ2aYd0Taw4</responseToken>
        </getBuyQuoteReturn>
   </getBuyQuoteResponse>
 </soapenv:Body>
 </soapenv:Envelope>
```

SOAP Error[2] Message Example (getBuyInit)

```
<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header />
<soapenv:Body>
   <getBuyInit xmlns="http://api.ws.ba.com">
        <msgName>getBuyInit</msgName>
        <clientID>84597</clientID>
        <branchID>1023ABC</branchID>
        <tranType />
        <userID>gfcsupport2</userID>
        <requestSecToken>CtBWaI2f196SBinTlzht4sej9CF6V1wxwEUYuWgoek
        18Lsk/8Qkx5DmqkKdROTIh27KEQ5o2iN8boIbxE8Vz+0f/+/QOjz3Sg0
        KiG6uw3Cj0uzLeN4BYfPB2Ut5NwrM5Rl0Rlf3xwDUgDjoEsFsMX8qiXCU
        PVTNOQuISI1vqFtSC7RPc1T9jEnVnevGFSL/nh649+A1VtPK7IUcD2/qt
        gGzXjPbJrrlkzNRCjtenkfVx/NTQXTSShb7PKNxQAl+8xO716aWU012Q
        OuuKqBEJYg==</requestSecToken>
  </getBuyInit>
</soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header />
<soapenv:Body>
  <getBuyInitResponse xmlns="http://api.ws.ba.com">
     <getBuyInitReturn>
        <ok xsi:nil="true" xmlns="http://bean.ws.ba.com" />
        <fail xmlns="http://bean.ws.ba.com">
             <errMsgID>GFC0020</errMsgID>
             <errMsgText>Unable To Determine Branch ID.</errMsgText>
        </fail>
        <msgName
        xmlns="http://bean.ws.ba.com">getBuyInitResponse</msgName>
        <msgResult xmlns="http://bean.ws.ba.com">Fail</msgResult>
        <responseSecToken xsi:nil="true" xmlns="http://bean.ws.ba.com" />
    </getBuyInitReturn>
  </getBuyInitResponse>
</soapenv:Body>
</soapenv:Envelope>
```

[2] GFC generated error message.

SOAP Error[3] Message Example (putTransaction)

```
<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header />
<soapenv:Body>
  <putBuyTransaction xmlns="http://api.ws.ba.com">
        <msgName>putBuyTransaction</msgName>
        <clientID>84597</clientID>
        <branchID>1023</branchID>
        <userID>gfcsupport2</userID>
        <tranType>NON-INVENTORY</tranType>
        <item xsi:type="soapenc:Array" soapenc:arrayType="xsd:anyType[1]">
          <item xsi:type="ns-1838558088:CartItem" xmlns:ns-
                1838558088="http://bean.ws.ba.com" xmlns="">
                <ns-1838558088:currCode>AUD</ns-1838558088:currCode>
                <ns-1838558088:denomination />
                <ns-1838558088:foreignAmount>5.0</ns-
                        1838558088:foreignAmount>
                <ns-1838558088:localAmount>3.66</ns-1838558088:localAmount>
                <ns-1838558088:rate>0.731</ns-1838558088:rate>
                <ns-1838558088:spreadType>STANDARD</ns-
                        1838558088:spreadType>
          </item>
        </item>
        <serviceCharge>10.0</serviceCharge>
        <settlementAmount>-6.34</settlementAmount>
        <origRefNumber>20051024051344</origRefNumber>
        <remarks />
        <customer>
                <custAcct xmlns="http://bean.ws.ba.com" />
                <custAddress xmlns="http://bean.ws.ba.com">
                  <custAdrLine1 />
                  <custAdrLine2 />
                  <custAdrLine3 />
                </custAddress>
                <custEmail xmlns="http://bean.ws.ba.com" />
                <custFirstName xmlns="http://bean.ws.ba.com" />
                <custLastName xmlns="http://bean.ws.ba.com" />
                <custNo xmlns="http://bean.ws.ba.com" />
                <custPhone xmlns="http://bean.ws.ba.com">
                  <custPhArea />
                  <custPhCnty />
                  <custPhExt />
                  <custPhNo />
                </custPhone>
```

[3] SOAP engine generated error message.

```
         </customer>
         <repeatFlag>N</repeatFlag>
         <requestSecToken>FhqpWECs3JvxYzNpl0u8a3VmiE0ZqQ0LmsOm1B7lY
         EiwfFf6CxBB4h3wMop8lVFoMRP5hydlkVqHqgS4PsqVOKDqQnzGJXfyr
         15/fK6tsQtJBWlOjTe4vDCP/NEVTgszuZyguNWt17qP7+kjomjIcuzkmH
         N7XkQvA758dAKhvoNoqfNAnPSK2DOLPeYdXHBqimqIPLsKUVaU4nyA
         +wzWOBp2dAbPBH/7AQVjBv7Nkq/0HbVVtsTQWy2EnguMIRTkVr6UQ
         1WXjLZqMW9oNocItc6fleRYR8tr9o1h4S+w1Ic=</requestSecToken>
   </putBuyTransaction>
</soapenv:Body>
</soapenv:Envelope>


<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header />
<soapenv:Body>
   <soapenv:Fault>
         <faultcode xmlns="">Server.generalException</faultcode>
         <faultstring xmlns="">com.ba.gfcapi.util.GFCProcessingException:
         GwsputtrnSendRecord: setDs_setamt() parameter not valid numeric
         format. Parameter is: -6.34</faultstring>
         <detail xmlns="" />
    </soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>
```

SOAP Message Service End Point

The following table listed the BofA GFC API service end points:

- where <instance> is the name of the “context-root” for the API service.
  - For DEMO and TESTING: <instance> = **demo-api**
  - For PRODUCTION: <instance> = **online-api**

e.g.: https://api.globalforeigncurrency.com/demo-api/services/QuoteGenerator

- *Bank of America GFC https connection is using 128-bit SSL encryption with digital certificate authentication. Only authorized parties with valid digital certificate are allowed to connect to api.globalforeigncurrency.com domain and instance*.

https://api.globalforeigncurrency.com/<instance>/services/QuoteGenerator

This service end point supports the getBuyQuote and getSellQuote API services.

Appendix 3 List of Error Messages

Message ID	Message Text (max 80 chars)
GFC0005	Invalid msgName &1.
GFC0010	Unable to determine Client ID.
GFC0020	Unable to determine Branch ID.
GFC0030	User ID not presented.
GFC0040	Invalid Currency Code.
GFC0050	Invalid Spread Type.
GFC0060	Service Charge Amount in error.
GFC0070	Foreign Amount in error.
GFC0080	Rate in error.
GFC0100	Local Amount in error.
GFC0110	Local Amount Calculation error.
GFC0120	Settlement Amt in error.
GFC0130	Settlement Amt presented different from Settlement Amt calculated.
GFC0140	Sett Amt presented different from Sett Amt on file for same Originator Ref.
GFC0150	Originator Reference Number not found.
GFC0160	Originator Reference Number already used.
GFC0170	Unable to determine Ship-To Branch ID.
GFC0180	Unable to determine Denomination Mix requested.
GFC0190	Shipping Method requested not available.
GFC0200	Shipping Charge Amount in error.
GFC0210	Local Amount requested too small to fill an order.
GFC0220	GFC Reference Number not found.
GFC0230	Originator and GFC Reference Number presented do not match.
GFC0240	Repeat Flag value not Y/N.
GFC0250	Originator Reference Number already accepted.
GFC0260	Originator Reference Number not found for acceptance.
GFC0270	Unable to determine Transaction Type.
GFC0280	Foreign Amt over inventory on hand for Curr (&1).
GFC0290	User ID not authorized for Inventory Transaction.
GFC0300	Unable to Determine The Branch Type.
GFC0310	Branch Type “Inventory-Main” needs to have acknowledged Cashbox 001.
GFC0320	Foreign Amt ordered must be a multiple of the smallest denomination.
GFC0330	Delete not allowed. &1 - Transaction settled - Transaction Status not eligible - Branch requesting Delete different - Cash box balancing - Different cash box - Will cause negative balance
GFC0340	Original and Repeat Transaction data integrity validation failed.
GFC0350	Transaction not allowed when GFC Cashbox Balance function is running.
GFC0360	Transaction not allowed when GFC Cashbox not acknowledged.

GFC0370	Transaction Confirmation Period has expired.
GFC9000	GFC message: &1
SEC9010	Security Token Not Found.
SEC9020	Security Token Authentication (cryptographic error) Failed.
SEC9030	Security Token Authentication (expired timestamp error) Failed.
SEC9040	Security Token Authentication (&1 validation error) Failed.

i.e. where &1 is substitution variable constructed in real-time.

* * * End of Document * * *