REMITA INTEGRATION DOCUMENTATION

Component Name	Remita Integration Document
General Description	Outlines the Remita Integration Implementation guide
Target Audience	Corporates who receive and/or make payments

TABLE OF CONTENT

1.0 INTRODUCTION....3
2.0 RECEIVING PAYMENTS....4
2.1 MERCHANT SETUP....5
2.2 IMPLEMENTATION....5
2.2.1 Payment Initiation:....5
2.2.2 Sample Transaction Request HTML form....6
2.2.3 Generating Request hash....7
2.2.4 Split Payments....7
2.2.5 Response Processing:....8
2.2.6 Using custom Payment Response:....9
2.3 TRANSACTION STATUS:....10
2.3.1 Status Check Parameter:....10
2.3.2 Sample JSON Response....10
2.3.3 Sample JSONP Response....11
2.3.4 Sample XML Response....11
3.0 MAKING PAYMENTS....12
3.1 OVERVIEW....12
3.2 CONFIGURING YOUR INTEGRATOR CLIENT....13
3.3 FILE DATASOURCE....14
APPENDIX A....16

1.0 INTRODUCTION

Remita is a modular e-Payments, e-Invoicing, e-Collections, e-Payroll, e-Schedules solution on a single multibank platform. Remita can remit funds to accounts in any Financial Institution and can also act as a gateway to “Collect” funds from various payment channels (e.g. Bank Branch, Credit/Debit Cards, Internet Banking, Mobile Wallets, etc) to pre-configured designated.

This document details the integration of a corporate’s platform to Remita. The purpose of this integration is to enable the corporate or merchant leverage on Remita’s diverse and seamless payment options to complete customers’ transactions in a handshake implementation known as Remita integration.

The following segments will outline, in detail, the processes and activities involved in integrating the merchant onto the Remita Platform for Collections and Payments.

2.0 RECEIVING PAYMENTS

Corporates can receive payments from their customers/payers into designated accounts (known as 'Collecting Accounts') on the Remita platform via various payment channels. This is called e-Collections. The Remita platform provides one of the most diverse payment/collection channels in the financial industry for corporates to leverage on for their collections. This is achieved with a unique reference number called the Remita Retrieval Reference (RRR). Payers can make payments to corporates' collecting accounts via any of the various channels listed below at their convenience:

- **Payment via Bank branch** – Customers can simply walk into the nearest branch to their location and use their RRR to pay cash for the services they are subscribed to from the merchants. Neither the merchant nor their customers need have an account in a bank for an RRR to be processed from its branches. The funds will be transferred to the biller/merchant within minutes.

- **Payment via Debit/Credit Card** – With the RRR, customers who want to pay with their cards (MasterCard, VISA, Verve etc) can do so by clicking either a link on the merchant's website or via the Remita website (www.remita.net). Remita e-Collection integration offers this class of customers the luxury of payment right from the comfort of their homes, offices or anywhere they are in the world so long as they have internet access.

- **Payment via Bank Internet Site** – Customers could as well log in to their online banking applications to pay with their RRR. They simply log in, click on the Remita Payments menu on their online banking application profile and make their payments via Remita e-Collection integration right from wherever they are.

- **Payments via Mobile Money/Wallet** – Customers have the option of paying from their mobile wallet accounts, provided there is an integration between Remita and their mobile wallet service provider.

- **Payment via POS/ATM**: Customers can also pay their merchants with RRRs through Automated Teller Machines (ATM) and Point of Sale (POS) channels.

In addition to multiple payment channels outlined above, Remita e-Collections offers additional premium features to Merchants:

- **Electronic Invoicing** - Merchants can generate and issue digital invoices containing Remita Retrieval References (RRRs) to notify their customers/payers of outstanding bill payments. They can also remind their customers of pending payments by re-sending these invoices to them periodically.

- **Electronic Receipts Issuance** - Remita sends a digital receipt carrying the merchant's corporate logo to the payers once payments have been made via the RRRs. Remita thus provides the immensely convenient benefit of issuance of invoices for bill payments **AND** receipts for payments made, all without lifting a finger after the RRRs have been sent!

2.1 MERCHANT SETUP

Before a corporate/merchant can use Remita e-Collection integration, it must be registered as a corporate on Remita. This is the creation of a profile on Remita for the merchant to logon as a user and administer/process the payments it receives and makes. The merchant also has access to associated reports to track/monitor all inflows and outflows to its collection and paying accounts.

How A Merchant is Setup For Remita Collections integration

a) Visit the Remita website at www.remita.net
b) Click the 'Sign up to Use Remita' button to display the Remita Registration Form
c) Complete the Registration Form and click 'Submit'. An RRR is generated and a notification containing your user credentials is sent to the email address you supplied. You are now registered as a corporate on Remita.
d) SystemSpecs Limited will issue a serviceTypeId and api_key to get you provisioned as a web merchant on the system.

The integration implementation requirements are outlined below.

2.2 IMPLEMENTATION

This section will guide you through the technical steps required for integration:

2.2.1 Payment Initiation:

You are required to create an HTML form which contains payment parameters. The Remita payment page is loaded on submitting the form from which a payment method can be chosen to complete the payment. When the payment type option is set in the request, the user is taken directly to the payment option of choice. If payment type is not set, the user is taken to the invoice page to select which payment option to use.

NOTE: if the paymenttype is set to RRRGEN, a Remita Retrieval Reference that is used to complete the payment transaction is generated and passed back to the merchant.

The HTML form Parameters are outlined in the table below:

Parameter Names	Description	Type
merchantId	**Required** This is a unique identifier of merchants.	String
serviceTypeId	**Required**	String
orderId	**Required**	String
hash	**Required** merchantId+serviceTypeId+orderId+amount+amt+responseurl+api_key	String

payerName	**Optional** This is the name of the customer to be displayed on the payment page.	
payerEmail	**Optional**	
payerPhone	**Optional**	
amt	**Required** This is the total monetary value of the transaction.	
paymenttype	**Optional** It can be either of the following ATM,POS,VERVE,VISA,MASTERCARD, UNION_PAY,REMITA,BANK_BRANCH,BANK_INTERNET,WALLET, RRRGEN	String
responseurl	**Required** The URL to which Remita should send transaction status report to on completion of transaction.	

2.2.2 Sample Transaction Request HTML form

```
<!DOCTYPE HTML>
 <html>
 <body>
 <form action="https://{server}/remita/ecomm/init.reg" name="SubmitRemitaForm"
method="POST">
 <input name="merchantId" value="1509328648353" type="hidden">
 <input name="serviceTypeId" value="1509328543428" type="hidden">
 <input name="orderId" value="8792" type="hidden">
 <input name="hash" value="ABCED12D3E1476DEFA12" type="hidden">
 <input name="payerName" value="Customer Name" type="hidden">
 <input name="payerEmail" value="customer_email@email.com" type="hidden">
 <input name="payerPhone" value="080211111111" type="hidden">
 <input name="amt" value="7000" type="hidden">
 <input name="responseurl" value="http://www.yourwebsite.com/response.php"
type="hidden">
 <input type ="submit" name="submit_btn" value="Pay Via Remita"></td>
 </form>
 </body>
 </html>
```

2.2.3 Generating Request hash

For security reasons you are required to hash your payment details with your API Key. Upon registration on Remita you will be given an API Key which should be kept secret. A valid payment request hash is generated by concatenating the following payment details and hashed using SHA512 algorithm and the assigned API Key:
merchantId+serviceTypeId+orderId+amt+responseurl+api_key

2.2.4 Split Payments

The corporate or merchant can also leverage on Remita's diverse and seamless payment options to complete distributed transactions to multiple receiving accounts.

The payment request will be sent via the JSON POST as shown below. Remita will facilitate payment of the specified amounts through the indicated service types into the corresponding collecting accounts as applicable.

SAMPLE JSON POST FOR SPLIT PAYMENTS

```
{
"merchantId":"2547933",
"serviceTypeId":"35457353800",
"totalAmount":"7000",
"hash":"9980c10768896f34d1bb202a45e53879122c01c1e01639d3e5ab90bad63ae37d2774",
"payerName":"Oshadami Mike",
"payerEmail":"oshadami@example.com",
"payerPhone":"08012345678",
"orderId":"POS3455J2",
"responseurl":"www.yourresponseurl.com",
"lineItems":[
{"id":"itemid1","beneficiaryName":"Oshadami
Mike","beneficiaryAccount":"1234567890","bankCode":"020","beneficiaryAmount":"2000"},
{"id":"itemid2","beneficiaryName":"Ogunseye
Mujib","beneficiaryAccount":"0987654321","bankCode":"022","beneficiaryAmount":"5000"}
]
}
```

Post required URL- https://{server}/remita/ecomm/v2/init.reg
Upon completion of the process, Remita sends JSON Response to the response url containing the following:

- **status** of the Transaction
- **RRR** (Remita Retrieval Reference) – If successful.

RRR PAYMENT
To make payment for the RRR generated, a HTTP POST FORM should be sent to Remita.

2.2.5 Sample Transaction Request HTML form

```
<!DOCTYPE HTML>
 <html>
 <body>
 <form action="https://{server}/remita/ecomm/finalize.reg" name="SubmitRemitaForm"
method="POST">
 <input name="merchantId" value="1509328648353" type="hidden">
 <input name="hash" value="ABCED12D3E1476DEFA12" type="hidden">
 <input name="rrr" value="Y11095959" type="hidden">
  <input name="responseurl" value="http://www.yourwebsite.com/response.php" type="hidden">
 <input type ="submit" name="submit_btn" value="Pay Via Remita"></td>
 </form>
 </body>
 </html>
```

2.2.6 Generating Request hash

For security reasons you are required to hash your payment details with your API Key. Upon registration on Remita you will be given an API Key which should be kept secret. A valid payment request hash is generated by concatenating the following payment details and hashed using SHA512 algorithm and the assigned API Key:

merchantId+rrr+api_key

2.2.5 Response Processing:

There are two types of responses that the user can get using the system:

1. **Instant Response**: This is gotten when the user is paying using REMITA_PAY, or any of the other card options. Responses are immediately returned to the merchant response url passed in with the request.
2. **Batched Response**: For transacting consummated via Bank Branch, POS or bank internet, you would not get a response until the transaction is completed. Here, feedback is posted to the Payment Notification Url configured at the collection rule setup in the format below:

```
[
{
  "id": 336360963,
  "rrr": "A39359490111111112",
```

```
  "channnel": "BRANCH",
  "amount": "6500.00",
  "responseCode": "01",
  "transactiondate": "19/01/2015",
  "debitdate": "19/01/2015",
  "bank": "033",
  "branch": "033152763",
  "serviceTypeId": "4430731",
  "dateSent": "20/01/2015",
  "dateRequested": "19/01/2015",
  "orderRef": "L4086852511",
  "payerName": "mujib ishola",
  "payerEmail": "ishola@systemspecs.com.ng",
  "payerPhoneNumber": "08141376868",
  "uniqueIdentifier": 10001
}
]
```


Upon completion of payment process, the Remita default payment response page is displayed to the customer. The Remita payment response page contains the following:

- Original detail of the payment transaction that was posted from the merchants website
- **status** of the payment
- **RRR** (Remita Retrieval Reference)
- **orderId** (generated by the merchant) and

2.2.6 Using custom Payment Response:

Merchants who choose to use a different payment response page should set the value of the **responseurl** to the page the customer should be redirected to upon completion or termination of payment. The response sent to the **responseurl** by Remita will include the **status**, **orderId** and **RRR** in the query.

The custom payment response page must display the following information to the customer:

- Remita Retrieval Reference (RRR)
- Order Id
- Status (Successful transaction notification should be BLUE and other transactions should be RED)
- Support contact details of the merchant (Address/Email Address, Phone Number)

2.3 TRANSACTION STATUS:

Although a status code is sent to the responseurl, a re-query for the status of the transaction is required to confirm the transaction status on Remita on completion of the transaction. For security reasons, a server side HTTP GET request should be made to either of the URLs below:

- The payment status URL via RRR:
  http://{server}/remita/ecomm/merchantId/RRR/hash/RESPONSE_TYPE/status.reg

- To Check transaction status via OrderID, status URL:
  http://{server}/remita/ecomm/merchantId/OrderID/hash/RESPONSE_TYPE/orderstatus.reg

See Appendix A for list of all response codes.

2.3.1 Status Check Parameter:

Parameter Names	Description	Type
merchantId	**Required** This is a unique identifier of merchants.	String
RRR	**Required**	String
hash	**Required** SHA512 value of **RRR+api_key+merchantId**	String
RESPONSE_TYPE	**Optional** The format in which the response will be formated.valid options are xml, jsonp or json. Default is json.	String

2.3.2 Sample JSON Response

{"statusmessage":"Transaction Approved","merchantId":"1509328648353","status":"01","RRR": O11615747,"transactiontime":2014-08-04 01:39:48 PM,"orderId":"1021232"}


```
{
  "statusmessage": "Transaction Approved",
  "merchantId": "1509328648353",
  "status": "01",
  "RRR": "O11615747",
  "transactiontime": "2014-08-04 01:39:48 PM",
  "orderId": "1021232"
}
```

2.3.3 Sample JSONP Response

callback('{"statusmessage":"Transaction
Approved","merchantId":"1509328648353","status":"01","RRR":
O11615747,"transactiontime":2014-08-04 01:39:48 PM,"orderId":"1021232"}')

```
callback('{
  "statusmessage": "Transaction Approved",
  "merchantId": "1509328648353",
  "status": "01",
  "RRR": "O11615747",
  "transactiontime": "2014-08-04 01:39:48 PM",
  "orderId": "1021232"
}')
```

2.3.4 Sample XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
        <statusMessage>Invalid Request</ statusMessage >
        <status>022</status>
        <RRR>J11616141</RRR>
        <orderId>1021232</orderId>
        <transactiontime>2014-08-04 01:39:48 PM </transactiontime>
        <merchantid>1509328648353</merchantid>
</response>
```

3.0 MAKING PAYMENTS

3.1 OVERVIEW

Corporates can also make payments via Remita Integration. This is facilitated by client known as the Remita Payments Integrator.

The Remita Payments Integrator is an all-inclusive tool that offers an array of payment transmission options to Integrating partners. Three methods (known as Datasource options) are available for transmitting payment instructions to REMITA. These include the Database, File and FTP data source options.

Integrating partners do not need to expend any extra development effort towards using the Remita Payments Integrator. The parameters for how payment instructions will be transmitted to the Remita payment gateway for onward processing are completely available within the integrator client on installation. The data source options are summarily described below.

i. **Database Datasource:** In this case, the Remita Payments Integrator reads payment instructions off a database to send to the payment gateway for processing. Configuration settings for connection/authentication to several types of industry-supported databases are available in this mode. Queries against the database for payment information retrieval can also be specified for the integrator to run periodically. Feedback on payments processed is instantaneous.

ii. **File Data Source:** The Remita Payments Integrator is set up to retrieve payment instruction files locally (This means that the file location is local to the Remita Payments Integrator client installation) to transmit to the payment gateway at specified intervals. Several configuration options are available to specify the attributes of the payment file for the Remita Payments Integrator client to recognise. Feedback in this case is not returned until all transactions in the session have been processed.

iii. **FTP Data Source:** The Remita Payments Integrator retrieves payment instruction files from a remote location in this case for transmission to the payment gateway for processing. This can be done via all available types of file transfer protocols (FTP, SFTP and FTPS).

Integrating partners can also send feedback to REMITA on transactions processed via the Remita Payments Integrator.

3.2 CONFIGURING YOUR INTEGRATOR CLIENT

This section shows how to configure the Remita payments integrator client depending on the datasource you have chosen to make your payments with. As mentioned above, the File, Database and FTP datasource types are available for use.

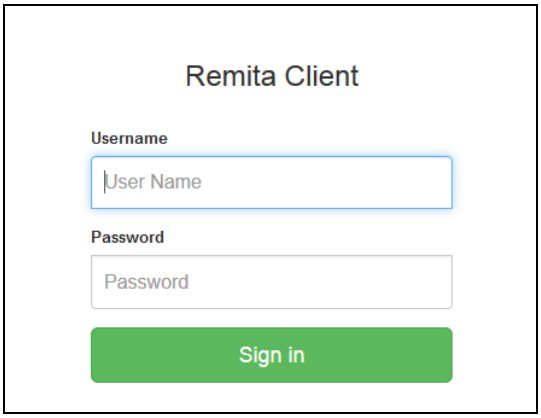

Figure 1 (Integrator Login Screen)

After logging onto the integrator client interface (See Fig1 above), the first thing to do is define a datasource by supplying a description and the funding account details for your payments.

You can do this as shown at Fig2 below by selecting the following menu options: Administration >> Manage Datasources.

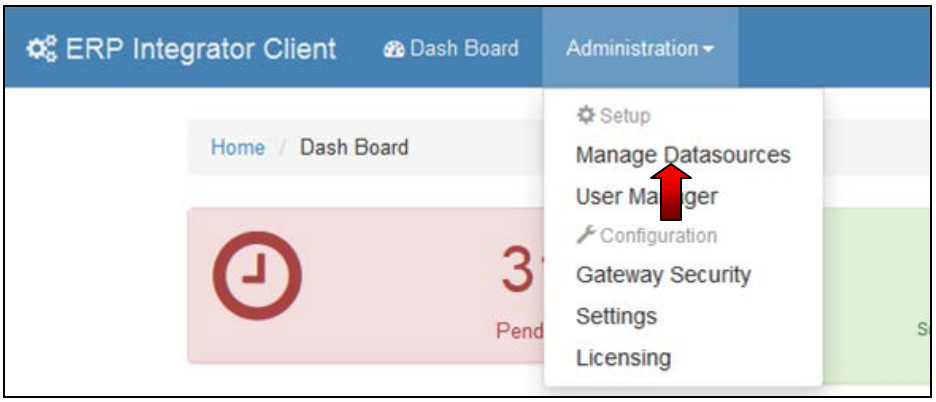

Figure 2 (Manage Datasource)

The datasource setup screen (shown at Fig3 below) is displayed when the 'Manage Datasources' menu is selected. Click on the 'Add' button (indicated by the red arrow) to display a new datasource type selection page (See Fig4).

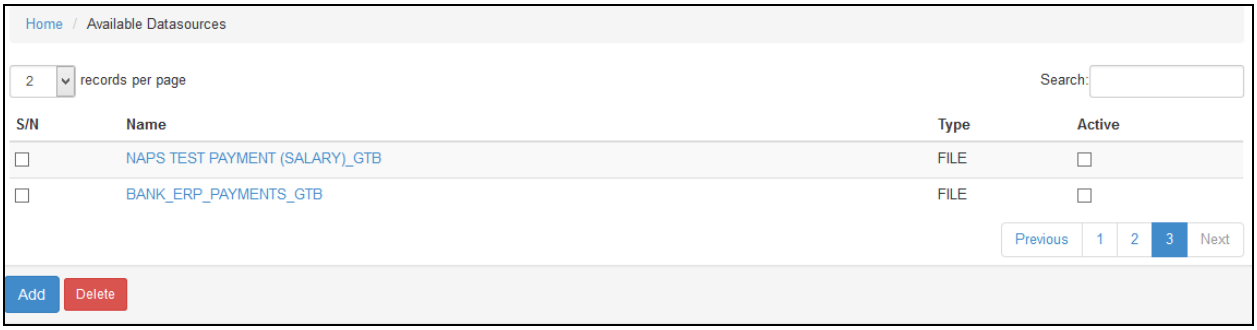

Figure 3 (Datasource Setup)

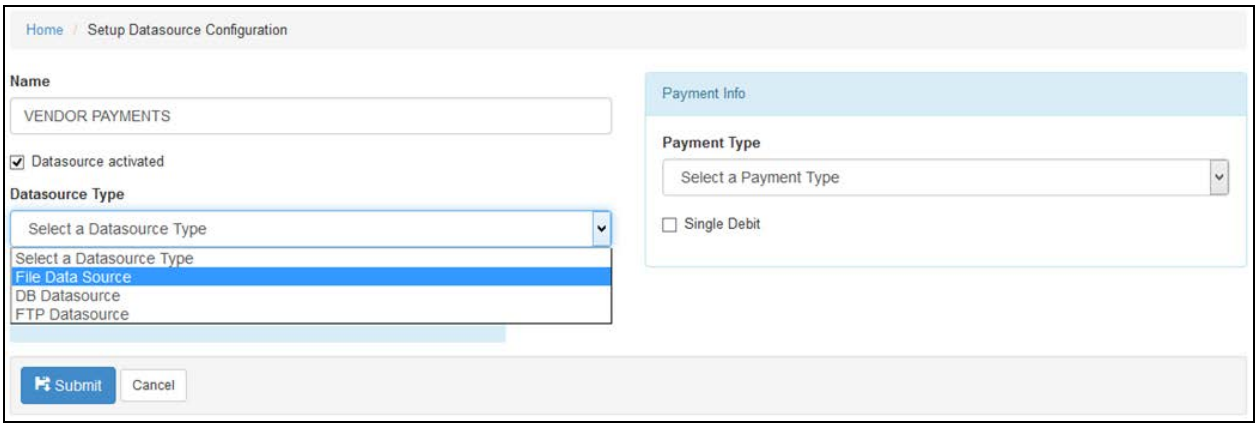

Figure 4 (Supply Datasource details)

Supply the datasource name and select payment type. Then choose your preferred datasource option to expose its corresponding payment settings. These are discussed in detail at the following sections.

3.3 FILE DATASOURCE

You are basically specifying your payment instruction file format and defining the pick-up location to the integrator here. There are four settings to complete, namely Send, Feedback, Status Configuration and Notifications.

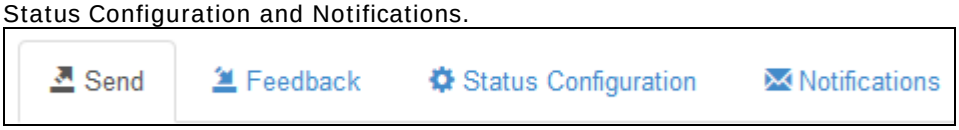

Figure 5 (File Datasource settings)

(a) SEND: This tab manages payment instruction file sending settings. It is comprised of template settings, file settings, payer info and beneficiary info.

i. Template settings: Interface is shown at You can indicate the type of file the integrator will pick-up from your local location (Fixed length, delimited or xml?), whether the file has got headers (and if yes, how many header rows?) and tell the client how the file instructions are funded (a default account already configured at a payment gateway or internal to REMITA?).

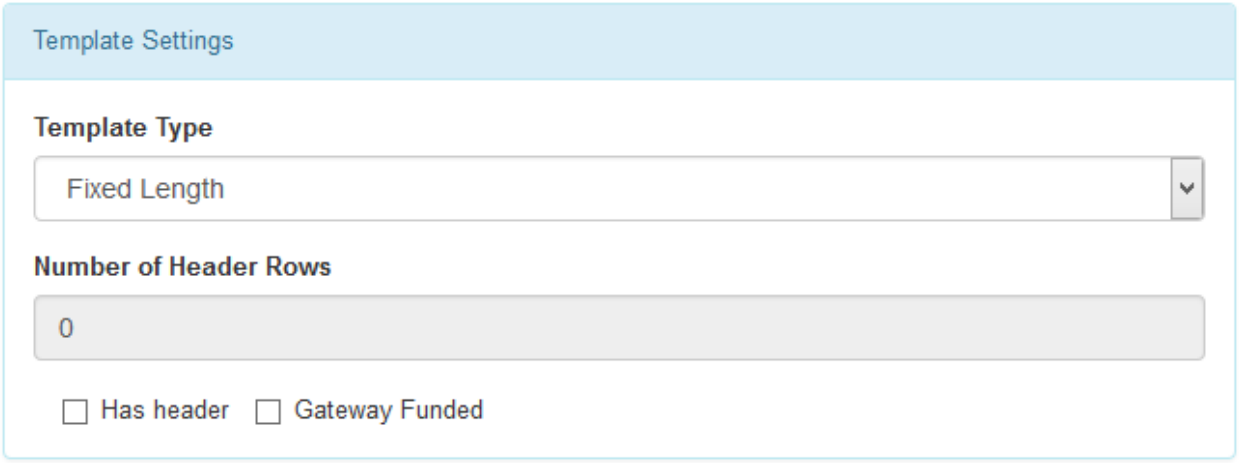

Figure 6 (Template Settings)

Note: See useful links at appendix

Merchant Name, Signature & Date.....

APPENDIX A

Status	Description
00	*SUCCESS*
01	*APPROVED*
02	*TRANSACTION_FAILED*
012	*ABORTED*
999	*UNKNOWN*
020	*AUTHENTICATION_ERROR*
030	*NO_FUND*
040	*REQUEST_OK*
031	*NO_ACCOUNT*
021	*PENDING*
022	*INVALID_REQUEST*
023	*INVALID_SERVICE_MERCHANT*
029	*INVALID_BANKCODE*

Useful Links

OASIS Username Token Profile Specification: http://tinyurl.com/65n78j
OASIS X.509 Certificate Token Profile Specification: http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf

A more detailed description of key generation can be found here:
http://java.sun.com/javase/6/docs/technotes/tools/solaris/keytool.html