

WIZCOM LTD.
***VISION* Message Engine**
Whitepaper v1.0.4

- all contents are subject to change -

Contents

1.	Introduction.....	3
2.	Soap Interface Details.....	4
2.1.	Set-up	4
2.2.	Interface Specification	4
2.3.	Plain Text	4
2.4.	Wap Push	5
2.5.	Binary.....	5
2.6.	Responses.....	6
3.	Delivery Reports.....	6
3.1.	Response to the server	6
4.	Appendix.....	8
4.1.	WSDL	8
4.2.	Example SOAP responses	10
4.2.1	Response to a valid request	10
4.2.2	Response to an invalid request	10

1. Introduction

The Wizcom *VISION* Message Engine enables customers to send messages to a mobile device.

2. Soap Interface Details

Customers send requests to send sms messages via a SOAP interface to the Wizcom servers.

2.1. Set-up

Wizcom will provide the customer with the following information:

Data	Description	Example
Username	Username of the customer account	user1732
Password	Password for the customer account	password
Send Limit	The maximum messages sent to a msisdn in a day.	500
URL	URL of the Wizcom <i>VISION</i> server	http://sms.apps.wizcommerce.com

2.2. Interface Specification

There are three interfaces available, provided by a SOAP interface (WSDLs in Appendix)

1. Plain Text – this interface enables the customer to send a plain text message
2. Wap Push – this interface enables the customer to send a wap push message
3. Binary – this interface enables the customer to send a binary message

2.3. Plain Text

This interface provides sends a plain text message.

The SOAP interface provides one method `sendPlainText`, which accepts the following parameters.

Name	Description	Type	
Username	Username	String	Required
password	Password	String	Required
notification URL	URL to send delivery reports to	String	Optional
message	Message text	String	Required
msisdn	MSISDNs of the message recipients (e.g. 07771824154)	String[]	Required
Originator	Who is sending the message	String	Optional
Delivery Date	The Date/Time to send the message	DateTime	Optional
Transaction Id	A user specific id for the message.	String	Optional

2.4. Wap Push

This interface provides sends a wap push message.

The SOAP interface provides one method `sendWapPush`, which accepts the following parameters.

Name	Description	Type	
username	Username	String	Required
password	Password	String	Required
notification URL	URL to send delivery reports to	String	Optional
URL	The wap push URL	String	Required
Title	Text for the link	String	Required
msisdn	MSISDN of the message recipient (e.g. 07771824154)	String	Required
Originator	Who is sending the message	String	Optional
Delivery Date	The Date/Time to send the message	DateTime	Optional
Transaction Id	A user specific id for the message.	String	Optional

2.5. Binary

This interface provides sends a binary message.

The SOAP interface provides one method `sendBinary`, which accepts the following parameters.

Name	Description	Type	
username	Username	String	Required
password	Password	String	Required
notification URL	URL to send delivery reports to	String	Optional
UDH	The UDH that corresponds to each part	String[]	Required Must be the same length as "parts" and must not contain null or empty Strings
parts	The binary message part that corresponds to each UDH	String[]	Required Must be the same length as "UDH" and must not contain null or empty Strings
msisdn	MSISDN of the	String	Required

	message recipient (e.g. 07771824154)		
Originator	Who is sending the message	String	Optional
Delivery Date	The Date/Time to send the message	DateTime	Optional
Transaction Id	A user specific id for the message.	String	Optional

2.6. Responses

All soap interfaces methods will returns an array of `messageSenderResult` composite objects. Each object will correspond to one `msisdn` specified in the interface. The `messageSenderResult` has the members:

Name	Description
Id	The unique id assigned to the request and <code>msisdn</code> . (-1 if an error has occurred)
Msisdn	The <code>msisdn</code> that the message is sent to
Errors	A list of errors that have occurred (empty if no errors)

All methods except *Plain Text 2.3* will have exactly one `messageSenderResult`. The plain text method allows bulk messaging by specifying multiple `msisdns`.

3. Delivery Reports

If the URL is specified in the soap request, then when confirmation of either successful deliver or failed delivery of the message is received, an HTTP GET request will be made to the supplied URL. If a URL was not supplied then the get request will be made to the default URL associated with the account. The request will contain two parameters:

Parameter	Description
id	The ID of the request, returned when the request was made.
status	Either 600, 601 or 603 600 = SUCCESS 601 = FAILURE 603 = DISPATCHED to network

For example:

<https://server.com/drpt.cgi?requestId=1234&status=600>

3.1. Response to the server

The customer's server should respond with a HTTP **200** response with a non-empty body.

4. Appendix

4.1. WSDL

```
<wsdl:definitions
targetNamespace="http://sms.apps.wizcommerce.com/webservices/services/SendMessageService">
  <wsdl:types>
    <schema
targetNamespace="http://sms.apps.wizcommerce.com/webservices/services/SendMessageService">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="ArrayOf_xsd_string">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
wsdl:arrayType="xsd:string[]" />
          </restriction>
        </complexContent>
      </complexType>
      <complexType name="ArrayOf_tns1_messageSenderResult">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType"
wsdl:arrayType="tns1:messageSenderResult[]" />
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://results.services.account.me.wizcom.com">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType abstract="true" name="AbstractMessageSenderResult">
        <sequence>
          <element name="errors" nillable="true"
type="impl:ArrayOf_xsd_string" />
          <element name="phoneNumber" nillable="true" type="xsd:string" />
          <element name="id" type="xsd:long" />
        </sequence>
      </complexType>
    </schema>
    <schema targetNamespace="urn:BeanService">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="messageSenderResult">
        <complexContent>
          <extension base="tns2:AbstractMessageSenderResult">
            <sequence />
          </extension>
        </complexContent>
      </complexType>
    </schema>
  </wsdl:types>
  <wsdl:message name="sendBinaryRequest">
    <wsdl:part name="username" type="xsd:string" />
    <wsdl:part name="password" type="xsd:string" />
    <wsdl:part name="notificationUrl" type="xsd:string" />
    <wsdl:part name="_udh" type="impl:ArrayOf_xsd_string" />
    <wsdl:part name="_parts" type="impl:ArrayOf_xsd_string" />
    <wsdl:part name="_msisdn" type="xsd:string" />
    <wsdl:part name="originator" type="xsd:string" />
    <wsdl:part name="deliveryDate" type="xsd:dateTime" />
    <wsdl:part name="transactionId" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="sendWapPushResponse">
    <wsdl:part name="sendWapPushResponse" type="impl:ArrayOf_tns1_messageSenderResult" />
  </wsdl:message>
  <wsdl:message name="sendBinaryResponse">
    <wsdl:part name="sendBinaryResponse" type="impl:ArrayOf_tns1_messageSenderResult" />
  </wsdl:message>
  <wsdl:message name="sendPlainTextRequest">
    <wsdl:part name="_username" type="xsd:string" />
    <wsdl:part name="password" type="xsd:string" />
    <wsdl:part name="notificationUrl" type="xsd:string" />
    <wsdl:part name="_message" type="xsd:string" />
    <wsdl:part name="_msisdn" type="impl:ArrayOf_xsd_string" />
  </wsdl:message>
</wsdl:definitions>
```

```

        <wsdl:part name="_originator" type="xsd:string"/>
        <wsdl:part name="_deliveryDate" type="xsd:dateTime"/>
        <wsdl:part name="_transactionId" type="xsd:string"/>
    </wsdl:message>
    <wsdl:message name="sendPlainTextResponse">
        <wsdl:part name="sendPlainTextResponse" type="impl:ArrayOf tns1 messageSenderResult"/>
    </wsdl:message>
    <wsdl:message name="sendWapPushRequest">
        <wsdl:part name="_username" type="xsd:string"/>
        <wsdl:part name="_password" type="xsd:string"/>
        <wsdl:part name="notificationUrl" type="xsd:string"/>
        <wsdl:part name="url" type="xsd:string"/>
        <wsdl:part name="title" type="xsd:string"/>
        <wsdl:part name="_msisdn" type="xsd:string"/>
        <wsdl:part name="_originator" type="xsd:string"/>
        <wsdl:part name="_deliveryDate" type="xsd:dateTime"/>
        <wsdl:part name="transactionId" type="xsd:string"/>
    </wsdl:message>
    <wsdl:portType name="MessageSenderSoapService">
        <wsdl:operation name="sendPlainText" parameterOrder="_username _password
_notificationUrl _message _msisdn _originator _deliveryDate _transactionId">
            <wsdl:input message="impl:sendPlainTextRequest" name="sendPlainTextRequest"/>
            <wsdl:output message="impl:sendPlainTextResponse" name="sendPlainTextResponse"/>
        </wsdl:operation>
        <wsdl:operation name="sendBinary" parameterOrder="_username _password _notificationUrl
_udh _parts _msisdn _originator _deliveryDate _transactionId">
            <wsdl:input message="impl:sendBinaryRequest" name="sendBinaryRequest"/>
            <wsdl:output message="impl:sendBinaryResponse" name="sendBinaryResponse"/>
        </wsdl:operation>
        <wsdl:operation name="sendWapPush" parameterOrder="username password notificationUrl
_url _title _msisdn _originator _deliveryDate _transactionId">
            <wsdl:input message="impl:sendWapPushRequest" name="sendWapPushRequest"/>
            <wsdl:output message="impl:sendWapPushResponse" name="sendWapPushResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="SendMessageServiceSoapBinding" type="impl:MessageSenderSoapService">
        <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="sendPlainText">
            <wsdlsoap:operation soapAction=""/>
            <wsdl:input name="sendPlainTextRequest">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://soap.services.account.me.wizcom.com" use="encoded"/>
            </wsdl:input>
            <wsdl:output name="sendPlainTextResponse">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://sms.apps.wizcommerce.com/webservices/services/SendMessageService" use="encoded"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="sendBinary">
            <wsdlsoap:operation soapAction=""/>
            <wsdl:input name="sendBinaryRequest">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://soap.services.account.me.wizcom.com" use="encoded"/>
            </wsdl:input>
            <wsdl:output name="sendBinaryResponse">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://sms.apps.wizcommerce.com/webservices/services/SendMessageService" use="encoded"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="sendWapPush">
            <wsdlsoap:operation soapAction=""/>
            <wsdl:input name="sendWapPushRequest">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://soap.services.account.me.wizcom.com" use="encoded"/>
            </wsdl:input>
            <wsdl:output name="sendWapPushResponse">
                <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://sms.apps.wizcommerce.com/webservices/services/SendMessageService" use="encoded"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="MessageSenderSoapServiceService">
        <wsdl:port binding="impl:SendMessageServiceSoapBinding" name="SendMessageService">
            <wsdlsoap:address
location="http://sms.apps.wizcommerce.com/webservices/services/SendMessageService"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>

```

4.2. Example SOAP responses

4.2.1 Response to a valid request

Where **method** is the SOAP method called, **07XXXXXXXX** is the msisdn that the message is sent to, and **1234** is an unique identifier for the request. In the case of *Plain Text 2.3* there will be multiple `messageSenderResult []`.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <methodResponse>
      <methodResponse xsi:type="soapenc:Array" soapenc:arrayType="ns1:messageSenderResult[1]"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="urn:BeanService">
        <item href="#id0"/>
          </methodResponse>
        </methodResponse>
      <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns2:messageSenderResult" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="urn:BeanService">
        <msisdn xsi:type="xsd:string">07XXXXXXXX</msisdn>
        <id xsi:type="xsd:long">1234</id>
      </multiRef>
    </soapenv:Body>
  </soapenv:Envelope>
```

4.2.2 Response to an invalid request

Where **method** is the SOAP method called, **07XXXXXXXX** is the msisdn that the message is sent to, and **1234** is an unique identifier for the request. In the case of *Plain Text 2.3* there will be multiple `messageSenderResult []`.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <methodResponse>
      <methodResponse xsi:type="soapenc:Array" soapenc:arrayType="ns1:messageSenderResult[1]"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="urn:BeanService">
        <item href="#id0"/>
          </methodResponse>
        </methodResponse>
      <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns2:messageSenderResult" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="urn:BeanService">
        <msisdn xsi:type="xsd:string" xsi:nil="1"/>
        <errors xsi:type="soapenc:Array" soapenc:arrayType="xsd:string[2]">
          <item>110 : Invalid MSISDN specified.</item>
          <item>270 : Not all required values have been supplied.</item>
        </errors>
        <id xsi:type="xsd:long">-1</id>
      </multiRef>
    </soapenv:Body>
  </soapenv:Envelope>
```