Vodafone Messaging

Direct Access Protocols

Introduction

Vodafone has a number of methods for message input. All methods have backup systems that may be local, interstate or both. All inputs have maximum input rate thresholds that can be varied upon request.

A maximum of 160 characters is supported for paging and some of the input protocols, regardless of destination, while SMSerTech offers up to 1000 characters (which may be split into multiple messages of 160 characters dependant on the destination device). The destination device may be one of or a combination of mobile phones, pagers or email.

Message input methods are either by means of an Australian operator, web pages or customer machines communicating directly with Vodafone machines.

SMSerTech is a web based products (detailed in the SMPP section of this document). For assistance with using the interface, please contact Vodafone Messaging Care.

A variety of access points are available, including leased lines, TCP/IP (via the internet), web pages or dial up.

Care can set up all accounts and supply all passwords as necessary. They can be contacted at messagingcare@vodafone.com.au or 1300-133-585.

PET

PET (Paging Entry Protocol) is also known as TAP, IXO and sometimes CAT. It is a text based protocol that includes checksum and error feedback from the PET receiver. It was originally designed for RS232 via dial up. Access is available via dial up, leased line and Internet (TCP/IP).

The table below indicates access points in each state. The 1300 numbers are preferred as VHA can re direct these numbers at will to another state in case of system failure. PET passwords are national and hence a customer's PET system can dial into any point and use a single allocated password. If the customer has more than one PET system, it is advisable for each system to have a unique password as this helps with trouble shooting.

Internet access is available via e2m.smser.com.au (203.18.188.132) - port 3030. This is a floating IP on two machines. If you telnet to e2m.smser.com.au (203.18.188.132) - port 3030 and press Enter, you should receive the challenge "ID=".

	Adelaide	Brisbane	Melbourne	Perth	Sydney
Quote the 1300 number where possible. High speed 8N1	1300132846 (08)83732541	1300134838 (07)38362569	1300135494 (03)96202912	1300136191 (08)93219844	1300139335 (02)94183288
2400 baud 8N1	N/A	(07)38318192	(03)96295436	(08)93221414	N/A
300 baud Bell 8N1	N/A	(07)38317124	(03)96298498	N/A	(02)94200200
300 Baud CCITT 8N1	(08)82722066	(07)38322113	N/A	N/A	(02)94200133

SMTP

To email a message, the destination address is;

<number>.<password>@e2m.smser.com.au

Do not query for mail as you will get nothing, it is a receiver only. All mail is accepted (for spam control reasons), even if the pager number or password is incorrect. You may need to contact Messaging Care to get a valid email password or get your existing PET password attached to email input.

Email can be sent by hand via popular applications such as Outlook. The fields sent to the destination device can be changed by Care, but generally are the subject followed by the body of the text until the 370 character limit is reached. The message is truncated at this point. No attachments are permitted and will be discarded.

The received message is generally in the following format;

From: "Sender Name",Subj: test in subject, Msg: The quick brown fox jumped over the lazy dogs back

The SMTP receiver is available on e2m.smser.com.au (203.18.188.132) - port 25. This is a floating IP on two machines. If you telnet to e2m.smser.com.au (203.18.188.132) - port 25 and press enter, you should receive the response "220 smspapp3.three.com.au e2m_smtpd

SMPP

HTML entry is either by hand or via machine input. The hand entry is available at the URL http://www.smsertech.com and is known as SMSerTech.

SMSerTech can send messages to international numbers of 12 digits such as +61411222222

SMPP is achieved by connecting to the back end of this web interface by means of a string such as:
https://www.smsertech.com/apisend?username=fred.bloggs@dodgy.com.au&password=happy1&to=0411111111&message=test to Rob v2&format=text .

The full protocol is detailed in appendix 2. This is a single string and has been formatted to fit on the page.

Appendix 1

Introduction

The PET protocol provides an industry standard protocol that can be implemented by paging users to connect to the Vodafone paging network. This document describes the implementation of this protocol within the Vodafone network. It is provided to self-entry paging users to assist in the implementation of this protocol with the aim of providing convenient and trouble free connection.

The PET protocol has been implemented by virtually every paging terminal manufacturer implementing alphanumeric paging. PET is a subset protocol of the Telocator Alphanumeric Input Protocol (TAP).

PET protocol basics

PET protocol is a serial access system whereby characters move between the Vodafone network and the user's paging terminal device. Characters are formed by 10 bits; being 1 start bit, 8 data bits and 1 stop bit. An alphabet of 128 ASCII characters is defined. Transmissions from the user's paging terminal device may send 7 data bits and 1 party bit rather than 8 data bits, however, the Vodafone network will ignore the value of the parity bit (i.e. it may be mark, space, odd or even parity).

Paging data is transferred to the Vodafone network in blocks with one paging message being transferred per block. Each block (message) is acknowledged on receipt by the network. A log-on and log-off sequence are specified which defines the start and end of a PET session.

The PET protocol

1. Dial number

User's PET device initiates a session by dialling a telephone number to access the VHA paging network. PET access is provided by all VHA paging operation centres in Australia. Dial number is not needed for IP connectivity or leased lines.

2. Answer call

VHA network answers the ring signal and generates a modem answer tone. A separate pool of modems (with their own telephone number) is provided for each of the supported modem protocols including BELL and CCITT at speeds from 300 baud to 33k6 baud. Some centres may not support all protocols and speeds.

3. <CR> send at 2 second intervals

The user's PET devices send <CR> (0D hexadecimal). The <CR> is repeated at two second intervals until the Vodafone network responds with "ID=" at the correct baud rate. Note that the "< and >" characters and the quotation marks are used for notation within this document and are not transmitted.

The Vodafone network will wait (with an idle line) up to 8 seconds for the <CR> character. Other characters may be transmitted and will be ignored but the <CR> must be received within 30 seconds (or 8 seconds with an idle line). If not, the VHA network will disconnect its modem and hang up the line.

4. "ID="

VHA responds with "ID= " within 1 second of the <CR>. Note that at some centres "ID=<CR>" or "ID=<CR><LF>" may be transmitted.

5. Transmit log-on sequence

The user's PET device sends: "<ESC>PG1pppppp<CR>"

The <ESC> character signifies that the calling device wishes to communicate in automatic dump mode. The PET protocol provides for a manual entry mode although this method is not supported by the VHA network.

The "PG1" sequence defines the type of service requested by the PET device. PG1 service is the only service type supported by Vodafone and provides for the transmission of alphanumeric paging message in blocks containing two fields. The two fields are the pager ID number and a message (where applicable).

"ppppp"is a six character alphanumeric password. The password is agreed upon by the customer and Vodafone. The log-on sequence is terminated by a single <CR> character.

The maximum possible log-on sequence length (including all terminating characters) is 16 characters. When 16 characters have been received (regardless of whether a <CR> terminator is present) the sequence is assumed to be complete and will be processed. Vodafone will wait up to 4 seconds for each character. A pause of more than 4 seconds will be taken to indicate the end of the log-on sequence.

6. The VHA network will respond to the log-on sequence in one of three ways.

(i) Log-on accepted

"Message sequence<CR><ACK><CR>

" This response indicates that the log-on sequence has been accepted and that the calling device is cleared to proceed with the session. In this response, and in all VHA Network PET responses, there can be an optional printable ASCII message sequence string. This string is completely optional, and if supplied, it is <CR> terminated. The message string length shall not be greater than 40 characters. The message sequence strings are subject to change, but they are intended to be easily understandable by end users.

Usually the log-on sequence will say something like "Welcome to VHA paging" or similar. Note however, that the insertion of message sequences (Including the terminating <CR>) is entirely at the option of the paging network and may or may not be present in various implementations. Refer also to paragraph 9. (iii). Also note that the <CR> terminator may (at the option of VHA paging) be replaced by <CR><LF> anywhere that it appears. In other words, any of the following sequences are possible:

<ACK><CR>

Or <ACK><CR><LF>

Or Welcome to VHA paging<CR><LF><ACK><CR><LF

(ii) Request log-on again

"Message sequence<CR><NAK><CR>ID=<CR>"

The log-on sequence was not valid for some reason and should be sent again. Possible reasons are that the sequence was incorrectly received (due to noise) or that the password was unacceptable or a service was requested that is not supported. The log-on sequence should be re-sent. Vodafone will allow 3 attempts to provide a valid log-on sequence. As for the above, the message sequence is entirely optional and the same possibilities apply in relation to the <CR> terminators.

(iii) Forced disconnect

"Message sequence<CR><ESC><EOT><CR>"

This sequence is sent after 3 attempts have been made to send a log-on sequence. After being transmitted, VHA will disconnect the modem and hang up.

7. Message go-ahead

"message sequence<CR><ESC>[p<CR>"

Immediately after sending the log-on acceptance sequence, VHA will transmit the message go-ahead sequence above. The message sequence (which, with its <CR> or optional <CR><LF> terminator may or may not be present) will prompt for the first data block to be transmitted thus:

"send first block"

Note: the '['(square open bracket) should be transmitted literally. Also note the LOWER CASE 'p'.

8. Send data blocks

The user's PET device is now clear to send the first block. While the PET protocol provides for individual transactions to span multiple blocks, the PG1 service provided does not. Blocks may be up to 256 characters in length (including ALL characters) and have the following form:

<STX>p...p<CR>m...m<CR><ETX>ccc<CR>

The <STX>, <CR> and <ETX> characters should be sent as shown. The value "p...p" and "m...m" are the pager ID number and message fields respectively.

Pager numbers are specified in decimal and may include any of the ASCII digits '0' through '9'. There is no fixed field width so numbers do not need to be padded with zeros.

The message may be any length (including zero) up to the maximum block length limit described above. Note however that many pagers will only accept messages of a shorter length and care should be taken to ensure that such limitations are not exceeded. The message may include any printable ASCII character (no control characters). Note also that VHA's maximum length of a message is 160 characters - any characters after this will not be sent to the pager. Furthermore, for numeric pagers, only a limited alphabet of characters is allowed for messages which are generally restricted to much shorter lengths. VHA may or may not check that all messages provided exceed any or all of the various pager defined limitations. The user should therefore take care to consider such factors.

The three characters "ccc" are the block check-sum. The check-sum is calculated by performing the simple arithmetic sum of the 7 bit values (i.e. ignore the party bit) of all characters preceding it in that block. (Note that the <STX> and <ETX> are included in the sum).
The check-sum is then the least significant 12 bits sent as 3 characters (4 bits per character) offset by 30 (Hexadecimal.ASCII '0'). I.e. a check-sum value A3F (hexadecimal) is sent as 3A (Hex). 33(Hex). 3F (Hex). = ":3?"

9. In response to a block, VHA will send one of 3 possible responses:

(i) Block accepted "message sequence<CR><ACK><CR>"

The block was acceptable in all respects including pager number, message content (such as was checked) and check-sum. The "<ACK><CR>" will indicate this, prompting for the next block.

The message "block accepted, send next block" is an example of a message sequence sent before this <ACK>

(ii) Block check-sum bad

"message sequence<CR><NAK><CR>"

The block check-sum was not correct, or there was some structural problem <ETX>, <STX> missing etc. The block should be retransmitted. It is the responsibility of the user's equipment to limit the number of retries that are performed on a particular block. Typically 3 tries would be an acceptable count. VHA may limit the total number of blocks received in a given session or the total session time before forcing disconnection and hanging up. This limit may take into account all blocks (good or bad) and total messages sent. This should be taken into account in determining retry limits.

(iii) Abandon block

"message sequence<CR><RS><CR>"

This sequence may occur when the check-sum is OK but the block violates some system rule. Most commonly this will mean an invalid pager ID or unacceptable message content. The block should not be sent again, but rather discarded or logged to an error file (possibly with the message sequence) for later inspection.

10. End of session

When all blocks have been transmitted, and after the reception of the <ACK> or <RS> for the last block, the user's PET terminal should send a disconnect sequence "<EOT><CR>" meaning that there are no more transactions remaining.

11. VHA disconnects

VHA will send its own disconnect sequence:

"Message sequence<CR><ESC><EOT><CR>" and then disconnect the modem and hang up.
Note: The PET protocol provides a facility to enable slower paging networks to catch errors that may have occurred earlier in the session and report these to the calling device. VHA does not utilise this facility as all errors are checked and reported on a block-by-block basis.

A complete PET session example

The following is an example of a complete PET session. The characters displayed in bold type indicate the actions of the VHA network. The other (normal) characters indicate the actions of the user's PET device.

Call VHA network

VHA modem answers call

<CR>

lD=<CR><LF>

<ESC>PG1freddy<CR>

Welcome to VHA paging<CR><LF>

<ACK><CR><LF>

Send first message<CR><LF>

<ESC>[p<CR><LF>

<STX>122<CR>

ABC<CR>

<ETX>17;<CR>

Bad block check-sum, resend block<CR><LF>

<NAK><CR><LF>

<STX>123<CR>

ABC<CR>

<ETX>17;<CR>

Message accepted, send next block<CR><LF>

<ACK><CR><LF>

<EOT><CR>

Thank you for calling. Goodbye<CR><LF>

<EOT><CR><LF>

VHA hangs<up

Caller hangs-up

Appendix 2

SMSerTech API Guide

1. Introduction

The SMSer API is a simple Application Programming Interface which provides access to the basic functionality of sending and receiving Short Messaging Service content.

The SMSer API enforces a number of restrictions:

- Retrieval of replies is dependent on the user configuration in the SMSer application. Replies can only be received when the configuration is not set to Reply to Mobile or Reply to Email.
- A single message may result in the transmission of multiple SMSs with each being charged at the standard rate. Each standard SMS is 160 characters long.
- Access is only permitted via the HTTPS protocol.
- The rate at which messages can be sent may be limited.
- The rate at which status or incoming queries are performed may be limited.
- All messages should only contain characters from the UTF-8 character set. Anything which is non UTF-8 will be silently discarded.

Note that in the various examples that follow values between curly brackets are user supplied values (without the curly brackets, of course).

2. Sending Messages

Sending a message can be performed using either the POST or GET protocols.

2.1 POST Protocol

https://www.smsertech.com/apisend

The available parameters are:

Parameter	Description
username	This is the username for the SMSer account
password	This is the password for the SMSer account
to OR to[]	This is the recipient's mobile phone number, pager number, email address, or an address book entry name. A phone number may optionally commence with a country code, otherwise it should commence with a zero. To send to multiple recipients separate the values with a comma, or specify multiple *to[]* parameters. square-bracketed form.
message	This is the SMS message content.
format	This is one of *text*, *html*, *json*, *xml* and determines the format of responses to this request.
flash	If this is set to 1 then messages will be 'flashed' on the recipient's device.

2.2 GET Protocol

https://www.smsertech.com/apisend?username={USER}&password={PASS}&to={PHONE}&message={MESSAGE}&format={FORMAT}

https://www.smsertech.com/apisend?username={USER}&password={PASS}&to={PHONE1},{PHONE2},{PHONE3}&message={MESSAGE}&format={FORMAT}

https://www.smsertech.com/apisend?username={USER}&password={PASS}&to[]={PHONE1}&to[]={PHONE2}&to[]={PHONE3}&message={MESSAGE}&format={FORMAT}

The parameters are identical to the parameters for the POST protocol.

2.3 Response

The format of the response is dependent on the value of the *format* parameter if supplied. By default the format is plain *text*.

Plain Text Response Example:

```
OK,uid=s9999999
```

XML Response Example:

```
<root>
        <result>
          <status>OK</status>
          <uid>s9999999</uid>
        </result>
</root>
```

JSON Response Example:

```
[{"status":"OK","uid":"s9999999"}]
```

Errors are usually displayed in the same manner as good results. The HTTPS header status code is always set to success (200) and is not used for API error purposes.

The following parameters may appear:

Parameter	Description
status	Will either be *OK* or *ERROR*. An *ERROR* indicates failure and that the message was not accepted for delivery. *OK* means the message was accepted for delivery but does mean that delivery will occur (a recipients hardware might be disabled, for example). For the text format the status parameter is not labelled but is always the first value in the delimited output.
code	This three-digit parameter only appears when the status parameter is set to *ERROR*. The first digit of the code defines the class of the response. 2xx Success — The message was successfully accepted and is pending final delivery. 4xx Client Error — The request contains bad syntax or cannot be fulfilled. 5xx Server Error — The server failed to fulfill an apparently valid request. A subsequent retry might succeed.
reason	This parameter only appears when the status parameter is set to *ERROR*. It gives a detailed explanation of the error. Use this information to decide what corrective action to take.
uid	This parameter only occurs when status is *OK*. Is is the unique identifier for the message and can be used to track the message delivery status.

3. Message Status

Querying message status can be performed using either the POST or GET protocols.

3.1 POST Protocol

https://www.smsertech.com/apistatus

The available parameters are:

Parameter	Description
username	This is the username for the SMSer account
password	This is the password for the SMSer account
uid OR uid[]	This is the unique identifier for the message for which status information is being requested. It is returned by the response to a Send request. To query for multiple messages separate each one with a comma, or use multiple *uid[]* parameters.
format	This is one of *text*, *html*, *json*, *xml* and determines the format of responses to this request.

3.2 GET Protocol

https://www.smsertech.com/apistatus?username={USER}&password={PASS}&uid={UID}&format={FORMAT}

https://www.smsertech.com/apistatus?username={USER}&password={PASS}&uid[]={UID1}&uid[]={UID2}&format={FORMAT}

https://www.smsertech.com/apistatus?username={USER}&password={PASS}&uid={UID1},{UID2}&format={FORMAT}

The parameters are identical to the parameters for the POST protocol.

3.3 Response

In addition to the error parameters returned by the Send request the following parameters are also available.

Parameter	Description
statuscode	This parameter shows the message delivery status. *E* Enroute *D* Delivered

	R Rejected *T* Timeout - Rejected
date	This parameter shows the date and time when the message was delivered (if the statuscode has value *D*).

4. Receiving Messages

4.1 POST Protocol

https://www.smsertech.com/apiget

The available parameters are:

Parameter	Description
username	This is the username for the SMSer account
password	This is the password for the SMSer account
format	This is one of *text*, *html*, *json*, *xml* and determines the format of responses to this request.

4.2 GET Protocol

https://www.smsertech.com/apiget?username={USER}&password={PASS}&format={FORMAT}

The parameters are identical to the parameters for the POST protocol.

4.3 Response

In addition to the error parameters returned by the Send request the following parameters are also available.

Parameter	Description
date	This parameter shows the date and time when the message was received.
from	This parameter shows the sender of the incoming message.
message	This parameter shows the incoming message text.