NAME

smtp − Client-side tcl implementation of the smtp protocol

SYNOPSIS

package require **Tcl**

package require **mime ?1.5.4?**

package require **smtp ?1.5?**

::smtp::sendmessage *token option...*

DESCRIPTION

The **smtp** library package provides the client side of the Simple Mail Transfer Protocol (SMTP) (1) (2).

::smtp::sendmessage *token option...*

This command sends the MIME part (see package **mime**) represented by *token* to an SMTP server. *options* is a list of options and their associated values. The recognized options are:

-servers

A list of SMTP servers. The default is **localhost**.

If multiple servers are specified they are tried in sequence. Note that the **-ports** are iterated over in tandem with the servers. If there are not enough ports for the number of servers the default port (see below) is used. If there are more ports than servers the superfluous ports are ignored.

-ports A list of SMTP ports. The default is **25**.

See option **-servers** above regardig the behaviour for then multiple servers and ports are specified.

-client The name to use as our hostname when connecting to the server. By default this is either localhost if one of the servers is localhost, or is set to the string returned by **info hostname**.

-queue Indicates that the SMTP server should be asked to queue the message for later processing. A boolean value.

-atleastone

Indicates that the SMTP server must find at least one recipient acceptable for the message to be sent. A boolean value.

-originator

A string containing an 822-style address specification. If present the header isn't examined for an originator address.

-recipients

A string containing one or more 822-style address specifications. If present the header isn't examined for recipient addresses). If the string contains more than one address they will be separated by commas.

-header

A list containing two elements, an smtp header and its associated value (the -header option may occur zero or more times).

-usetls This package supports the RFC 3207 TLS extension (3) by default provided the tls package is available. You can turn this off with this boolean option.

-tlsimport

This boolean flag is **false** by default. When this flag is set the package will import TLS on a sucessfully opened channel. This is needed for connections using native TLS negotiation instead of **STARTTLS**. The **tls** package is automatically required when needed.

-tlspolicy

This option lets you specify a command to be called if an error occurs during TLS setup. The command is called with the SMTP code and diagnostic message appended. The command should return 'secure' or 'insecure' where insecure will cause the package to continue on the unencrypted channel. Returning 'secure' will cause the socket to be closed and the next server in the **-servers** list to be tried.

-username

-password

If your SMTP server requires authentication (RFC 2554 (4)) before accepting mail you can use **-username** and **-password** to provide your authentication details to the server. Currently this package supports DIGEST-MD5, CRAM-MD5, LOGIN and PLAIN authentication methods. The most secure method will be tried first and each method tried in turn until we are either authorized or we run out of methods. Note that if the server permits a TLS connection, then the authorization will occur after we begin using the secure channel.

Please also read the section on **Authentication**, it details the necessary prequisites, i.e. packages needed to support these options and authentication.

If the **-originator** option is not present, the originator address is taken from **From** (or **Resent-From**); similarly, if the **-recipients** option is not present, recipient addresses are taken from **To**, **cc**, and **Bcc** (or **Resent-To**, and so on). Note that the header key/values supplied by the **-header** option (not those present in the MIME part) are consulted. Regardless, header key/values are added to the outgoing message as necessary to ensure that a valid 822-style message is sent.

The command returns a list indicating which recipients were unacceptable to the SMTP server. Each element of the list is another list, containing the address, an SMTP error code, and a textual diagnostic. Depending on the **-atleastone** option and the intended recipients, a non-empty list may still indicate that the message was accepted by the server.

AUTHENTICATION

Beware. SMTP authentication uses **SASL**. I.e. if the user has to authenticate a connection, i.e. use the options **-user** and **-password** (see above) it is necessary to have the **sasl** package available so that **smtp** can load it.

This is a soft dependency because not everybody requires authentication, and **sasl** depends on a lot of the cryptographic (secure) hashes, i.e. all of **md5**, **otp**, **md4**, **sha1**, and **ripemd160**.

EXAMPLE

```
proc send_simple_message {recipient email_server subject body} {
    package require smtp
    package require mime

    set token [mime::initialize -canonical text/plain \
        -string $body]
    mime::setheader $token Subject $subject
    smtp::sendmessage $token \
        -recipients $recipient -servers $email_server
    mime::finalize $token
}

send_simple_message someone@somewhere.com localhost \
    "This is the subject." "This is the message."
```

TLS SECURITY CONSIDERATIONS

This package uses the **TLS** package to handle the security for **https** urls and other socket connections.

Policy decisions like the set of protocols to support and what ciphers to use are not the responsibility of **TLS**, nor of this package itself however. Such decisions are the responsibility of whichever application is using the package, and are likely influenced by the set of servers the application will talk to as well.

For example, in light of the recent *POODLE attack* [http://googleonlinesecurity.blogspot.co.uk/2014/10/this-poodle-bites-exploiting-ssl-30.html] discovered by Google many servers will disable support for the SSLv3 protocol. To handle this change the applications using **TLS** must be patched, and not this package, nor **TLS** itself. Such a patch may be as simple as generally activating **tls1** support, as shown in the example below.

```
package require tls
tls::init -tls1 1 ;# forcibly activate support for the TLS1 protocol

... your own application code ...
```

REFERENCES

[1] Jonathan B. Postel, "SIMPLE MAIL TRANSFER PROTOCOL", RFC 821, August 1982. (*http://www.rfc-editor.org/rfc/rfc821.txt*)

[2] J. Klensin, "Simple Mail Transfer Protocol", RFC 2821, April 2001. (*http://www.rfc-editor.org/rfc/rfc2821.txt*)

[3] P. Hoffman, "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002. (*http://www.rfc-editor.org/rfc/rfc3207.txt*)

[4] J. Myers, "SMTP Service Extension for Authentication", RFC 2554, March 1999. (*http://www.rfc-editor.org/rfc/rfc2554.txt*)

BUGS, IDEAS, FEEDBACK

This document, and the package it describes, will undoubtedly contain bugs and other problems. Please report such in the category *smtp* of the *Tcllib Trackers* [http://core.tcl.tk/tcllib/reportlist]. Please also report any ideas for enhancements you may have for either package and/or documentation.

When proposing code changes, please provide *unified diffs*, i.e the output of **diff -u**.

Note further that *attachments* are strongly preferred over inlined patches. Attachments can be made by going to the **Edit** form of the ticket immediately after its creation, and then using the left-most button in the secondary navigation bar.

SEE ALSO

ftp, http, mime, pop3

KEYWORDS

email, internet, mail, mime, net, rfc 2554, rfc 2821, rfc 3207, rfc 821, rfc 822, smtp, tls

CATEGORY

Networking

COPYRIGHT