



Shepherd.SMTP Service

J&J Computer Consulting Shepherd Server Publications

Overview

The Shepherd.SMTP Service implements an RFC 821 compliant Simple Mail Transfer Protocol message transfer agent. Through Shepherd Directory Services (SDS), Shepherd.SMTP provides support for virtual hosts with unique IP addresses or virtual hosts on a single IP address. Shepherd.SMTP also includes SPAM control, relay control, forwarding, and customizable local delivery.

Installation

Unzip the Shepherd.SMTP distribution to the Shepherd installation directory. This will install a sample smtp configuration for the directory in setup\smtp.ldif, the Service initialization file in services\smtp.svc, and the dynamic link library for the Shepherd.SMTP Service in services\smtp.dll. Before getting started with Shepherd.SMTP, you need to create an additional attributes required by Shepherd.SMTP and create a ShepherdService object in the directory. To create the additional attributes, use the atsetup option in the shepherd.ini file to create the attributes found in setup\smtp.attrib. Refer to "Shepherd Installation and Configuration" for further information on configuring a ShepherdService object for Shepherd.SMTP.

Configuration

Shepherd.SMTP requires new object classes and attributes throughout the directory to function properly and interact with Shepherd.POP3 or other local delivery systems. This section explains new objects and changes to existing objects and how they should be used.

ShepherdService

The Shepherd.SMTP ShepherdService object provides a few more options than typical ShepherdService objects due to its requirements for additional configuration data.

The **smtpRelay** attribute tells the Shepherd.SMTP Service whether or not it should act as a relay host and forward mail to external hosts. Possible values for this attribute include "enabled" or "disabled".

If smtpRelay is enabled, an **smtpRelayControl** attribute can be used to identify an **smtpRelayControl** object in the directory. The contents of the smtpRelayControl attribute should be the distinguished name of an smtpRelayControl object in the directory. smtpRelayControl objects will be described later in this section and are used to prevent unauthorized mail relaying through Shepherd.SMTP.

The **smtpSpamControl** attribute points to an **smtpSpamControl** object in the directory. If present, SPAM control is enabled and the destination object is used to provide an appropriate filter for host names and IP addresses connecting to Shepherd.SMTP. smtpSpamControl objects will be described later in this section.

The **smtpHeaderControl** attribute points to any number of **smtpHeaderControl** objects in the directory. The contents of the headerControl attribute should be the distinguished name of an smtpHeaderControl object in the directory. smtpHeaderControl objects will be described later in this section.

To prevent overloading of the Shepherd.SMTP Service with relay connections, the **maxRelayConnections** attribute is provided. With this attribute, you can specify the maximum number of relay connections permitted. Shepherd.SMTP uses this information to validate whether or not it can process a relay request prior to receiving the message data from the sender. Used in conjunction with the maxConnections attribute provided for all ShepherdService objects, the maximum number of inbound and outbound connections can be controlled for the Shepherd.SMTP Service.

To limit the size of incoming messages, the **maxMessageSize** attribute can be added to the Shepherd.SMTP Service. maxMessageSize should be expressed in bytes.

The **localDeliveryAPI** attribute gives Shepherd.SMTP the default processing method for local deliveries. The appropriate format follows:

```
<dll>!<api>
```

<dll> is the name of the DLL for the API such as "services\pop3.dll", and <api> is the name of the function to call. To work with Shepherd.POP3, the entry should be:

```
services\pop3ld.dll!smtpToPop3Delivery
```

If you would like to implement another local delivery API, you must create a DLL with a function defined as follows:

```
int smtpDelivery( char* msgfilename, unsigned long* size, char* mailaddress, void* dirinfo );
```

Shepherd.SMTP will pass the name of the message file, the size of the message file, the e-mail address of the local recipient, and directory specific information not available to external delivery APIs. The API should attempt to remove the message file if delivered successfully, but any inability to remove the message file should not result in an error.

The return value should be any of the RFC 821 defined reply codes. Shepherd.SMTP will return this reply code to the client if it is still connected, or in some cases, the reply code will be used to send an undeliverable notification message to the sender.

The **dnsReverseLookups** attribute tells Shepherd.SMTP whether or not it should perform reverse lookups on the client IP address. Possible values include "enabled" and "disabled". If you are using relay or spam control, this option directly impacts the function of those systems. By disabling reverse lookups, you prevent relay and spam control from using any host name entries in their filter lists.

The queue processor provides several options for configuration. The **mailQueue** attribute tells Shepherd.SMTP where it should store temporary queue files. If not present, Shepherd.SMTP will attempt to create a mail queue in the queue sub-directory of the Shepherd directory.

The **mailQueueDelay** attribute specifies the amount of time Shepherd.SMTP waits between queue checks. The value should be specified in minutes, and if not present, Shepherd.SMTP will wait 15 minutes between queue checks.

The **mailQueueTimeout** attribute, specified in days, determines the number of days the queue will attempt to send a message before returning an undeliverable notification message to the sender. A default value of 3 days is used if this attribute is not present.

Similarly, the **mailQueueWarning** attribute, specified in hours, determines the number of hours the queue will wait before warning the sender of a message not being sent. A warning is only sent once after the specified time elapses. If not present, Shepherd.SMTP will assume a default of 4 hours.

smtpRelayControl

The smtpRelayControl object allows you to control the hosts that can relay mail through the Shepherd.SMTP Service. Aside from the standard attributes, smtpRelayControl includes an **smtpHostFilter** attribute that includes any number of accept and reject statements. The accept and reject statements allow both ip addresses and host names (unless dnsReverseLookups is disabled). IP addresses and host names allow wildcards in any portion of the address or host name.

smtpSpamControl

The smtpSpamControl object allows you to control the hosts that can connect to the Shepherd.SMTP Service and send mail. As with the smtpRelayControl object, smtpSpamControl uses an smtpHostFilter object that contains accept and reject statements for ip addresses and host names.

smtpHeaderControl

The smtpHeaderControl object allows you to control actions taken on messages with specified information in the headers of the message. The **smtpHeaderFilter** attribute allows the information for the header to be identified. Multiple values can be used to differentiate by multiple fields. The format of the attribute's contents is:

<headerfield>: <value>

If the associated value exists as a substring of the header value in the message, the action will be taken. If more than one value is given for smtpHeaderFilter, each value must match the message for the action to be taken.

If there is a match, the **smtpHeaderControlAction** attribute specifies the action to take. The following actions are available:

reject	Rejects the message and sends an error to the client application.
send <filename> to (sender rcpt <address>)	Sends the e-mail message found in the specified file to the sender the intended recipient, or to a 3 rd party.

Multiple actions can be specified, so a message can be rejected and a message sent to the sender in response to a particular header.

PostOffice

The PostOffice object class helps Shepherd.SMTP determine which mail should be delivered locally and which mail should be delivered externally. The PostOffice object identifies domains under which mail received should be processed locally. An example of a PostOffice entry looks like:

```
dn: cn=yourdomain.com, c=US
cn: yourdomain.com
objectclass: top
objectclass: PostOffice
description: YourDomain.com Post Office
```

When mail comes into Shepherd.SMTP for user@yourdomain.com, Shepherd.SMTP looks up the PostOffice entry in the directory for yourdomain.com. If found, Shepherd considers the delivery to be either local or local relay and attempts to find "user" in the same path of the directory as it found "yourdomain.com".

ShepherdAccount

Once Shepherd.SMTP has found the appropriate user account for a local delivery, there are several attributes that control how it continues.

The **mailboxSize** attribute specifies the number of bytes of message data that can be stored in the user's mailbox at a time. If the mailbox meets or exceeds this size, Shepherd.SMTP will return a temporary, non-fatal error to the sender.

The **forward** attribute allows mail destined for this user to be sent to another address. Instead of processing the forward request immediately, the message is accepted and queued for delivery to the ultimate recipient. The mail address can be expressed as user@domain or user@domain.

The **mailHost** attribute lets you define separate mail hosts for users in the directory. For sites that require multiple mail servers, this attribute allows all mail to be addressed to the base domain and distributed to the appropriate mail server. This also allows any Shepherd.SMTP Service to know where to send mail for any user in the directory even if they are not local. In cases where other SMTP servers are used alongside Shepherd, either the forward or mailHost attributes can be used to send the mail to the appropriate server.

The **localDeliveryAPI** provides another level of customization of mail delivery. The ShepherdService specifies the default localDeliveryAPI for all users, but that API can be overridden for any user in the directory by specifying the localDeliveryAPI in the ShepherdAccount object.

Log Files

Shepherd.SMTP attempts to log several different sets of data and can use several log files. The `logFile` attribute of the `ShepherdService` object can point to one or more `ServiceLog` objects, but those objects must be identified appropriately with the **logType** attribute. Options for the `logType` attribute include:

- audit
- error
- spam
- relay
- header
- debug

Shepherd.SMTP will use audit, spam, relay, header, and error, if available, to log events.

The audit log identifies data from standard SMTP transactions including client ip address, server ip address, hello command domain, mail from address, all recipient addresses, the result of the mail transaction, total connection time, and bytes transferred.

The spam, relay, and header logs identify appropriate data for actions taken as the result of `smtpSpamControl`, `smtpRelayControl`, and `smtpHeaderControl` options.

The error log has no specific format and is primarily used to log failed mail exchange lookups and local processing problems.