



IBM Software Group Enterprise Networking Solutions  
z/OS® V1R11 Communications Server

## *New SMTP client for sending Internet mail*

*z/OS Communications Server Development, Raleigh, North Carolina*



© Copyright International Business Machines Corporation 2009. All rights reserved.

This presentation will discuss a new SMTP client for sending Internet mail.

## Existing SMTPD

- The SMTP server (SMTPD) mail gateway provides important mail services for business applications on z/OS
- Heavily used for sending mail from MVS batch and TSO to internet destinations
- SMTPD reads JES spool data sets created from batch jobs and TSO users locally and from NJE network
- Acts as SMTP MTA, does not use the system resolver and can resolve individual recipient addresses to deliver mail to their destinations
- Acts as listening MTA server, accepting mail and sending to the next hop or delivers to local or NJE users



The SMTP server (SMTPD) mail gateway provides important mail services for business applications on z/OS. SMTPD is heavily used for sending mail from MVS batch and TSO to internet destinations. SMTPD reads from the JES spool data set created from batch jobs and TSO users locally and from NJE network. SMTPD acts like an SMTP MTA since these batch spool files contain mail commands, headers, and data that conform closely to SMTP RFC 821 and RFC 822. SMTPD does not use the system resolver; it can resolve individual recipient addresses to deliver mail to their destinations. SMTPD acts as a listening MTA server, accepting mail from other TCP/IP mail clients and can either send them to the next hop over TCP/IP or deliver them to local or NJE users.

**SMTPD is showing its age**

- Many requirements need addressing from customers
- Uses older RFC 821 and RFC 822, when newer RFCs are available
- AT-TLS for TLS/SSL is not supported
- Performance and DASD problems:
  - Single threaded
  - Fully capable MTA role
    - Resolves recipient with an attempt to deliver each note (with required retry attempts with a minimum of one day) causes heavy I/O use
  - Stores each mail read from spool to DASD



- Sendmail has not been very popular among z/OS installations
  - Perceived as being too complex for the purpose of just mailing from z/OS

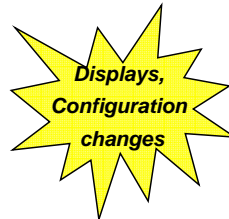
SMTPD is starting to show its age and there are many requirements that need addressing from customers. SMTPD uses the Pascal socket API. AT-TLS for TLS/SSL is not supported in the Pascal API. SMTPD uses older RFC 821 and RFC 822 and needs to be updated for RFC 2821, RFC 2822, RFC 1870 (message size) and RFC 3207 for security.

SMTPD has performance and DASD utilization problems. SMTPD is a single-threaded Pascal application. In its fully capable MTA role, it resolves each recipient to determine the target SMTP server, thus causing heavy resolver usage. SMTPD does not use the system resolver. SMTPD retry processing (with required retry attempts with a minimum of one day) causes heavy I/O use. SMTPD stores each mail read from a JES spool data set to DASD which causes heavy DASD usage.

Sendmail is also supported on z/OS, but does not have support for sending mail that has been written to the JES spool. Most customers have expressed concern with the complexity of sendmail and have also expressed a continued need to be able to send bulk mail from their existing applications that write the mail to the JES spool.

## CSSMTP - New SMTP client for sending Internet mail

- Allows existing users of SMTPD that use forwarder feature to migrate easily
- Uses newer mail standards and additional message size and security RFCs
- Improves performance and storage management issues with SMTPD when forwarding mail
- Improved usability features



- Allows multiple JES spool processing threads and concurrent IP connection threads
- Supports both IPv4 and IPv6 addresses



In z/OS V1R11, CSSMTP was introduced and provides several advantages over SMTPD.

CSSMTP was written to allow existing users of SMTPD that use forwarder feature to migrate easily. This means that you can use existing batch jobs that use the old RFC 821 and RFC 822 without making any changes. You must change the external writer name if CSSMTP will co-exist with SMTPD.

Use CSSMTP if you need the latest versions of the SMTP RFCs, including RFCs for message size and security. CSSMTP uses the newer mail standards RFC 2821 and RFC 2822 for interacting with server MTAs and supports additional RFCs for message size (RFC 1870) and security (RFC 3207).

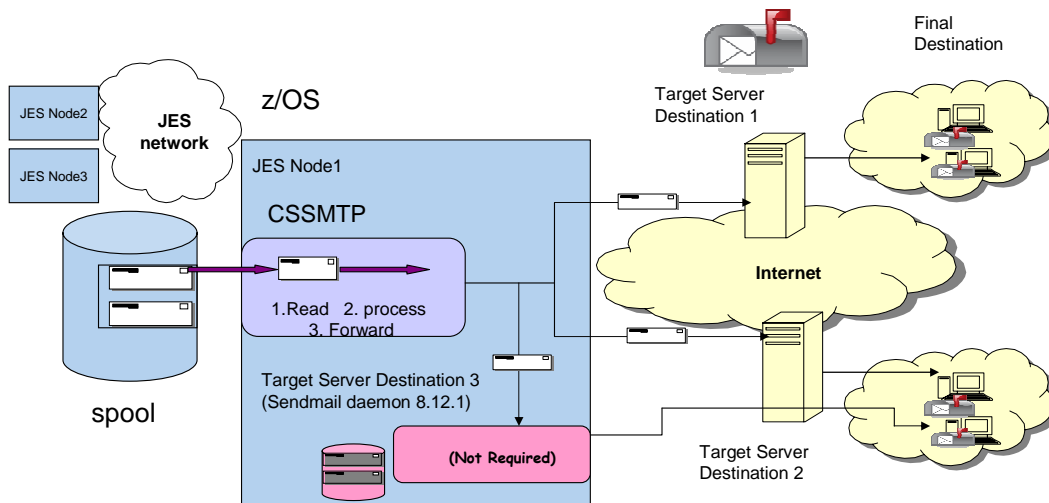
CSSMTP improves performance and storage management issues with SMTPD when forwarding mail. CSSMTP eliminates the heavy disk I/O bound nature that was a characteristics of SMTPD. CSSMTP does not require permanent disk storage for storing "active" mail messages.

CSSMTP offers improved usability features, for example, console commands for displays, changing configuration and logging.

CSSMTP uses newer platform technologies and is a multi-tasking LE application. It uses a multi-threaded implementation to provide multiple JES processing threads and concurrent IP connection threads.

Finally, CSSMTP supports both IPv4 and IPv6 addresses.

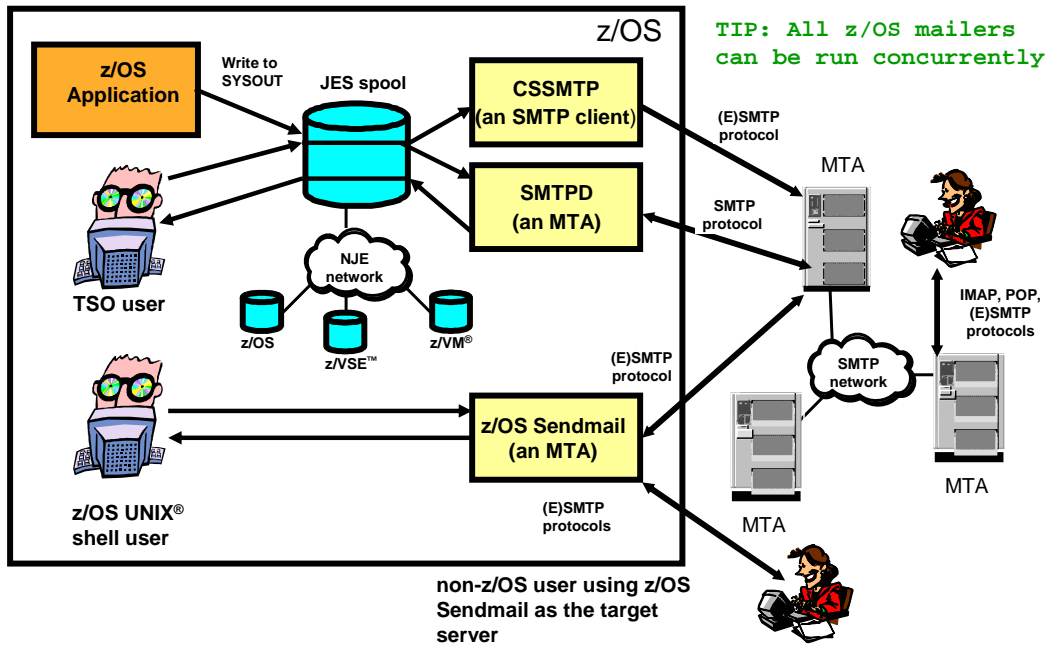
## CSSMTP: Read and forward mail messages from JES spool data set



Here is an overview of CSSMTP forwarding mail messages from a JES spool data set. This diagram shows that CSSMTP can read mail from a JES2 or JES3 spool data set. CSSMTP is a mail-forwarding SMTP client application only and is not a fully capable MTA. The primary use for CSSMTP is for MVS batch jobs that want to send mail to the Internet from z/OS, for example you are using bulk mail where replies to the originator are typically not required. CSSMTP forwards the mail to the target servers without resolving each recipient. CSSMTP improves the performance, scalability, and availability of the client function over SMTPD. These target servers will likely be non-local and they will either forward or deliver to the final recipient. If you need to store and forward the mail messages, you can consider having a local sendmail server or zLinux sendmail server on the same CEC defined above as target server destination three.

CSSMTP can coexist with SMTPD, and multiple instances of CSSMTP can run on a single host. You can only start one instance of CSSMTP per external writer name. The external writer name is used in the selection criteria for JES2 or JES3 subsystem spool files.

### CSSMTP, SMTPD and Sendmail



This picture shows the available z/OS mailers and will give you direction on picking the best z/OS mailer for you. The previous slides showed the advantages of using CSSMTP.

## Configuring CSSMTP configuration file

- CONFIG DD in started procedure is name of PDS(E), an MVS sequential data set or z/OS UNIX file
- Sample configuration file is located in SEZAINST(CSSMTPCF)
- CSSMTP will fail initialization if no configuration file is found
- TargetServer statement is the only required configuration statement



CSSMTP will first check to see if the CONFIG DD statement is defined in the started procedure. If the CONFIG DD statement is defined then CSSMTP uses the specified MVS PDS(E), MVS sequential file or z/OS UNIX file for the configuration file. If the CONFIG DD statement is not defined in the started procedure then CSSMTP uses the default MVS sequential file: *jobname.CSSMTP.CONFIG* for the configuration file name. A sample started procedure for CSSMTP is located in SEZAINST(CSSMTP). A sample configuration file for CSSMTP is located in SEZAINST(CSSMTPCF). CSSMTP will fail initialization if no configuration file is found. TargetServer statement is the only required configuration statement. All other configuration statements and parameter values have default values defined.

## Configuring TargetServer statement

- Use to configure target servers and their connection attributes
- Target servers are used for sending mail messages
- Use either parameters 1) or 2) to define target servers
  - 1) TargetIP, TargetName (or both)
  - 2) TargetMx
- Multiple TargetServer statements can be used for TargetIP, TargetName or both
- TargetIP defines a single configured IP address, TargetName or TargetMx resolves to one or more IP addresses each representing a target server



Use the TargetServer statement to configure your target servers and attributes. A target server is used for sending mail messages. You must configure at least one TargetServer statement. You can use either the parameters: TargetIP, TargetName (or both), or a single TargetMx parameter. A TargetMx parameter cannot be configured with a TargetIP parameter or a TargetName parameter. Multiple TargetServer statements can be used when using TargetIP, TargetNames or both. TargetIP defines a single configured IP address (which can be IPv4 or IPv6). TargetName or TargetMx is used to define the host name or MX name which is resolved into one or more target IP addresses. Only the first four configured or resolved target IP addresses are used to forward the mail.

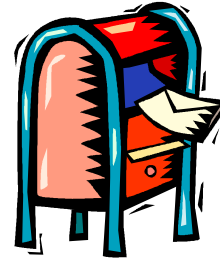
The name defined on TargetName is the host name or fully qualified domain name used for a resolver A or AAAA query.

The name defined on TargetMx is the name or a fully-qualified domain name used for a resolver MX query.



## Configuring mail error handling

- RetryLimit statement
  - Retry count and interval time when attempting to resend mail messages that are not immediately deliverable
- Undeliverable statement
  - Method to use for handling undeliverable mail messages and whether to create an undeliverable mail notification
- Report statement
  - Use to set the action required for error reporting on JES spool files
- MailAdministrator Statement
  - Defines an e-mail address to receive error reports



This slide lists the configuration statements for handling mail errors.

You can configure the RetryLimit statement with retry count and interval time to wait when attempting to resend mail messages that are not immediately deliverable. When mail is being processed under this time interval, it is considered to be in 'long retry'. This retry limit is limited to two hours for a count and interval time combination. For example, four retries with 30 minutes per retry result in the maximum time of two hours. You can set count or interval time to zero if you do not want to attempt to resend mail messages that are not immediately deliverable. When retry limits are exhausted, the ReturnToMailFrom parameter on the Undeliverable statement is used to determine whether to return this mail message to the originator.

You can configure the Undeliverable statement to indicate how you want to handle undeliverable mail messages and whether to create an undeliverable mail notification to be returned to the mail originator. A dead letter is an undeliverable mail notification that cannot be returned to the mail originator.

You can configure the Report statement to indicate the action required for reporting problems for JES spool files. If set to ADMIN, an error report is created and sent to the configured mail administrators, at least one MailAdministrator statement must be configured.

The MailAdministrator statement defines an e-mail address to which CSSMTP will deliver error reports when the Report statement value is set to ADMIN. The format for the e-mail address is *user@host.domain*. At least one MailAdministrator is required if you configure Report statement to ADMIN.

## Starting and stopping CSSMTP

- Start CSSMTP as a z/OS started procedure
- Sample in SEZAINST(CSSMTP) - you should start and stop from operator console  
start *jobname* or stop *jobname*
- Start options:
  - p or -P *tcpipJobName*
    - Use in common INET environment to choose a specific TCPIP stack
  - f or -F
    - Use to perform a cold start and flush any checkpoint records from previous execution
- Must run under a SAF user ID with an associated OMVS segment
- Must reside in an APF-authorized library



You can only start CSSMTP as a z/OS started procedure. A sample started procedure is shipped in SEZAINST(CSSMTP). You can start CSSMTP with the -f or -p start options in the started procedure. Stop the CSSMTP from the MVS console by issuing this command: stop *jobname*. The -p or -P with the *tcpipJobName* can be used in a common INET environment when you want to choose a specific TCPIP stack for this execution of CSSMTP. You can also use the \_BPXK\_SETIBMOPT\_TRANSPORT environment variable to choose a specific TCPIP stack, but the -p overrides this environment variable. Another start option is the -f or -F which can be used when you want to do a cold start and flush any checkpoint records from a previous execution of CSSMTP. For a warm start, do not use -f or -F and configure CHKPOINT DD statement in the started procedure. This checkpointing feature will reduce the likelihood of duplicate mail messages being sent on restart of CSSMTP. CSSMTP must run under a SAF user Id with an associated OMVS segment, and it must reside in an APF-authorized library.

If the external writer name is not defined in the configuration file, the external writer name is the job name.

Use

Start *jobname.extwrt*

To start an alternate CSSMTP procedure where the writer name is "extwrt" when the external writer name is not defined in the configuration file.

The checkpoint file can be shared between multiple CSSMTP procedures. The checkpoint file is extended for each new CSSMTP procedure.

## Display CSSMTP targets

- How to follow mail progress using the display targets command:

```

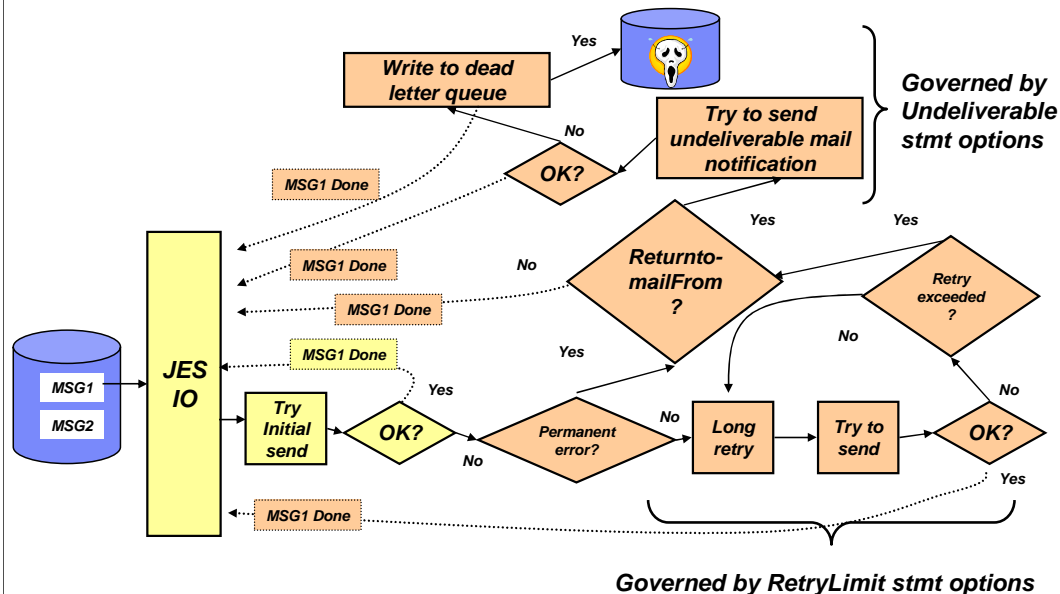
F CSSMTP,D,TARGET
EZD1831I CSSMTP TARGETS:
GLOBAL INFORMATION:
MAIL SENT      : 1562                TOTAL RETRY   : 15
DEADLETTER    : 0                   CURRENT RETRY : 0
UNDELIVER     : 0
TARGET SERVER 2000:197:2:107::1
STATE         : ACTIVE
ESMTP        : YES                  MESSAGE SIZE  : 0
STARTTLS     : NO                   MAIL ATTEMPTS: 493
MAIL SENT    : 438                  CONNECT FAIL  : 0
TARGET SERVER 197.11.108.1
STATE        : ACTIVE
ESMTP       : NO                    MESSAGE SIZE  : 524288
STARTTLS    : NO                    MAIL ATTEMPTS: 350
MAIL SENT   : 350                   CONNECT FAIL  : 0

```

Using the modify D,TARGET command to display target server information. The global information section contains numbers for this instance since the start of the CSSMTP application. This section shows how many mail messages went through various processing steps. MAIL SENT indicates the number of mail messages sent with all recipients sent successfully. TOTAL RETRY indicates the number of mail messages that went through long retry processing. CURRENT RETRY indicates the number of mail messages currently on the long retry queue. DEADLETTER indicates the number of mail messages that went to dead letter processing. Dead letter processing occurs when an undeliverable mail message cannot be returned to the originator of the mail message. UNDELIVER indicates the number of mail messages that went to undeliverable processing (sending to some recipients might have been successful). These fields are not meant to be added up but to show that a processing step has occurred. This is because these fields are based on a mail message boundary and not the number and state of each recipient associated with the mail message.

Associated with the label target server is the specific IP address and additional information such as the IP address state, whether it is an ESMTP (the old SMTP), and the configured message size. If the target server is an ESMTP, the size information is retrieved from EHLO response and the zero indicates it is unlimited. MAIL ATTEMPTS is the number of mail messages put on the queue for this IP address. MAIL SENT is the number of mail messages that the target server has accepted (at least one recipient was accepted). Note that these fields sometimes do not add up due to mail message attributes, number of recipients, and whether the target server accepts them all. CONNECT FAIL is the number of times that the SMTP connection cannot be established to send mail messages to the remote SMTP server.

## The logic of trying to send a mail message



JES will process the mail messages from JES spool to be sent to a target server by CSSMTP. CSSMTP uses selection criteria to process and send the mail message to a target server. JES is notified when processing is done for the mail messages.

For the mail message with retry-able (not permanent) errors, CSSMTP will retry the mail message with configured RetryLimit statement values.

Mail messages with either permanent errors or the retries have been exceeded become undeliverable mail messages. If the Undeliverable Statement / ReturnToMailFrom is set to Yes, then CSSMTP creates the undeliverable mail notification to be returned to the mail originator. If ReturnToMailFrom is set to No, the undeliverable mail messages are logged and marked done. If the original spool file contains no errors other than undeliverable errors, the spool file is deleted.

An undeliverable mail notification that cannot be returned to the original sender, becomes a dead letter. If Undeliverable statement / DeadLetterAction is set to store, then CSSMTP will store each dead letter as a separate file within the dead letter z/OS UNIX file directory.

## Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

z/OS z/VM z/VSE

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.