Notes from Support - Under the Microscope
The Add-in and Delivery Report Task of the SMTP MTA

by Bret Sweeden

[Editor's note: This article resides in “Iris Today”, the technical webzine located on the http://www.notes.net Web site produced by Iris Associates, the developers of Domino and Notes.

*The Under the Microscope column is brought to you by **Lotus Support**, and features in-depth technical articles that explore how a particular Domino or Notes feature works. This month's article is the final article in a three-part series on the SMTP MTA, and explores the Add-In Controller and Delivery Report Task (DRT).]*

We've finally arrived at the last article in the three-part series on the SMTP Message Transfer Agent (MTA). If you've followed the series since the beginning, you know how it's progressed; however, if you haven't been with us since the first article, here's a quick summary of what we've covered so far:

- **Transport services of the SMTP MTA** (first article in the series): This article explained how the MTA sends and receives messages.
- **Conversion services of the SMTP MTA** (second article in the series). This article explained how the MTA converts messages from Notes-to-Internet format, and vice versa.

Most of what the MTA does was described in the first two articles; however, to complete our exploration of the SMTP MTA, we need to discuss the MTA’s final two components:

- The Add-In Controller: This task starts and stops the MTA’s other tasks, and performs nightly housekeeping.
- The Delivery Report Task (DRT): This task cleans up the work queue databases and the SMTP.BOX, sends delivery failure notifications when necessary, and tracks the MTA’s statistics.

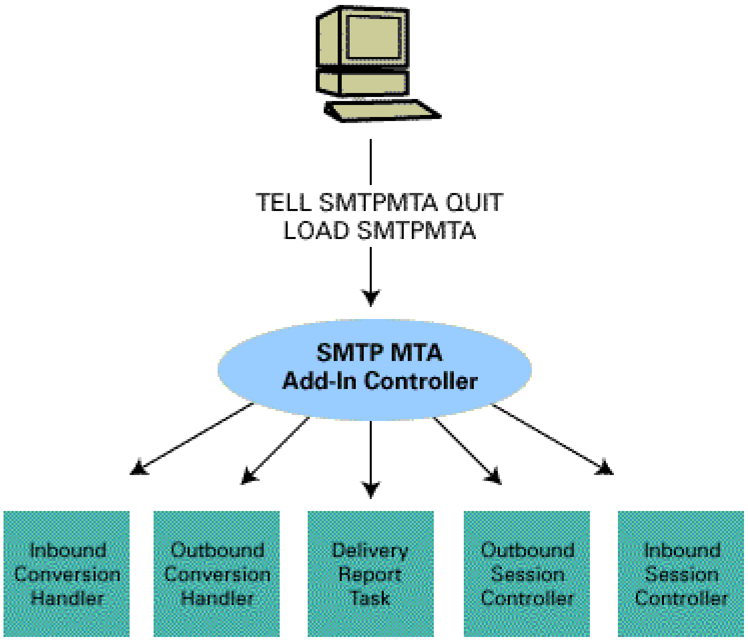
This article explores these final components, and explains how they work with the rest of the MTA; however, unlike the first two articles, there are no hands-on exercises. Instead, I've included some "final thoughts" that suggest ways to continue putting the SMTP MTA *Under the Microscope*.

The Add-In Controller: Telling everyone what to do

The Add-In Controller is the MTA task that starts and stops the Inbound/Outbound Conversion Handler, the Inbound/Outbound Session Controllers, and the Delivery Report Task. The only MTA threads that aren't directly managed by the Add-In Controller are the Inbound/Outbound Session Handlers. These processes are controlled by their respective session controllers.

When you load the SMTP MTA, the Add-In Controller starts. The controller appears on the server console as **SMTPMTA**. Once this controller is initialized, it reads the MTA’s configuration settings from the Public Address Book and the NOTES.INI file. After these settings are read into memory, the controller starts the remaining MTA tasks. Whenever you send a TELL command to the MTA, such as TELL SMTPMTA QUIT, the Add-In Controller receives the command and passes it to the appropriate MTA process for execution (for details about Tell commands, see the sidebar, "**A Guide to Tell Commands**").

Note: The Add-In Controller reads the MTA configuration information during initialization and checks for configuration changes every two minutes during normal operation. If a change is detected, the Add-In Controller updates the configuration information stored in memory. This information becomes "live" when the other MTA tasks reload their configuration.


TELL SMTPMTA QUIT
LOAD SMTPMTA
SMTP MTA
Add-In Controller
Inbound Conversion Handler
Outbound Conversion Handler
Delivery Report Task
Outbound Session Controller
Inbound Session Controller

Aside from receiving and forwarding MTA TELL commands, the Add-In Controller also performs daily housekeeping. This activity takes place every night at 1:00 AM (by default). When housekeeping takes place, the Add-In Controller shuts down all of the MTA processes and compacts the Inbound/Outbound Work Queue databases and the SMTP.BOX. Once database compacting is complete, the Add-In Controller restarts the various MTA tasks (Inbound/Outbound Conversion Handler, Inbound/Outbound Session Controllers, and the Delivery Report Task).

Note: Deleting documents from a database leaves blocks of unused space within the database. New documents fill in most of the unused space. Sometimes, however, unused space remains. Large amounts of unused space can degrade database performance. Routine database compacting helps reclaim the unused space not previously claimed by new documents.

The DRT task: Cleaning up after everyone

The SMTP add-in controller starts the Delivery Report Task (DRT) when the MTA loads. DRT appears on the server console as **SMTPMTA drt**. This task performs two specific functions, it:

- Manages messages in the Inbound/Outbound Work Queues and the SMTP.BOX
- Updates the MTA's statistics

Each message coming in or going out through the MTA passes through a series of "states." These "states" have different meanings depending on which database a message is in. There are a total of four "states"; however, a message doesn't pass through each state during its lifetime.

- **Pending**: A message with this "state" in the Inbound Work Queue or the SMTP.BOX is pending conversion. A message with this "state" in the Outbound Work Queue is pending transmission.
- **Success**: A message in this "state" in the Inbound Work Queue has been successfully converted. A message with this "state" in the Outbound Work Queue has been successfully transmitted to its next hop.
- **Failure**: There are two types of failure: temporary and permanent. A temporary "state" of failure is usually found only in the Outbound Work Queue (appearing as a failed conversion or failed

transmission). When a message transmission fails, the failure "state" is temporary until the MTA reaches the maximum number of "retry" attempts as defined in the Public Address Book. The failure becomes permanent once the maximum retry transmission is reached. Permanent failures also occur when message conversion is not possible.

- **WQ_LOCKED_FOR_XMIT**: This "state" occurs when a message is in the process of transmission, queued for transmission, or queued for a transmission retry.

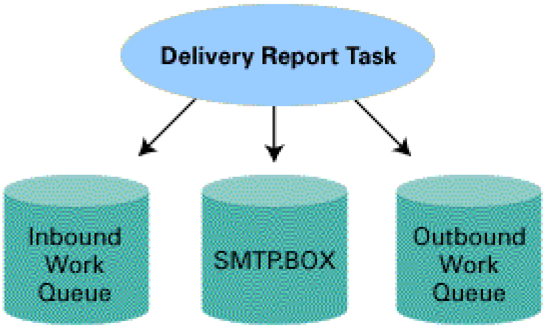
Managing messages in the Outbound Work Queue
The DRT task polls the Outbound Work Queue every two minutes (the default poll time set in the Server document) for messages with a status of success or failure. If a message is in a "state" of success, this indicates that the message was successfully transmitted. The DRT task looks at the message in the work queue, and locates its parent document in the SMTP.BOX database. The document count in the parent decrements by one, and the child document in the Outbound Work Queue deletes. If the document count equals zero, the parent document deletes from the SMTP.BOX. If the count has not reached zero, there are still child documents that have not reached a final "state" of success or failure. Therefore, the parent document remains in the SMTP.BOX so the DRT task can use it when processing the remaining child documents.

If a message has reached a final "state" other than success, it's probably a failure. When the DRT task encounters messages in this "state," it creates an undeliverable message notification based on the information stored in the failed message's parent document in SMTP.BOX. The notification is sent to the Notes sender, the document count in the parent document decrements by one, and the child document deletes from the Outbound Work Queue. When the document count in the parent reaches zero, the parent deletes from the SMTP.BOX database. Otherwise the parent document remains.

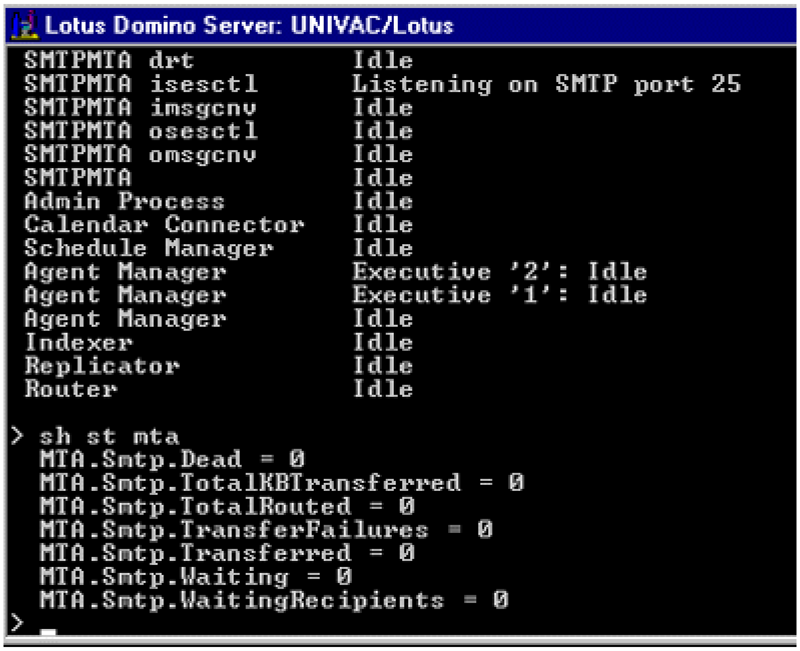
There is a strong relationship between parent documents in the SMTP.BOX and child documents in the Outbound Work Queue. However, messages relayed to other mail systems, or nondelivery reports, are documents that don't have parents. When these messages reach a "state" of success, the DRT task deletes them from the Outbound Work Queue and doesn't bother with the parent document. If, however, these documents reach a "state" of failure, the DRT task marks these messages dead and they remain in the Outbound Work Queue until you manually delete them.

Note: Dead messages are not removed by the Add-In Controller during daily housekeeping. Therefore, you should routinely check the Inbound/Outbound Work Queue for "dead" messages. If found, analyze these messages and try to determine the cause of the failure (for example, missing Person documents). Then, manually delete these messages from the work queue.

Managing messages in the Inbound Work Queue
Once the DRT task checks the Outbound Work Queue for messages in a "state" of success or failure, it checks the Inbound Work Queue for messages in the same "states." If a message is in a "state" of success, the DRT task deletes the message from the work queue. If a message is in a "state" of failure, the DRT task creates a nondelivery report (NDR) and puts it in the Outbound Work Queue. This report is delivered to the message originator the same way any other outbound message is delivered. If, for whatever reason, the DRT task cannot build a failure report, the message is marked "dead" and remains in the Inbound Work Queue database.



Updating statistics
Aside from managing messages in the various MTA databases, the DRT task also updates the statistics for the MTA. The DRT task performs this duty at the end of its polling period. The DRT task cleans up the Outbound Work Queue first, then the Inbound Work Queue, and finally updates the MTA's statistics. This statistical information is available from the server console at any time by typing SHOW STAT MTA.



```
 SMTPMTA drt          Idle
 SMTPMTA isesctl      Listening on SMTP port 25
 SMTPMTA imsgcnv      Idle
 SMTPMTA osesctl      Idle
 SMTPMTA omsgcnv      Idle
 SMTPMTA              Idle
 Admin Process        Idle
 Calendar Connector   Idle
 Schedule Manager     Idle
 Agent Manager        Executive '2': Idle
 Agent Manager        Executive '1': Idle
 Agent Manager        Idle
 Indexer              Idle
 Replicator           Idle
 Router               Idle

> sh st mta
  MTA.Smtp.Dead = 0
  MTA.Smtp.TotalKBTransferred = 0
  MTA.Smtp.TotalRouted = 0
  MTA.Smtp.TransferFailures = 0
  MTA.Smtp.Transferred = 0
  MTA.Smtp.Waiting = 0
  MTA.Smtp.WaitingRecipients = 0
> _
```

Final thoughts

While the MTA is becoming increasingly easier to configure and use, an intimate understanding of the MTA definitely helps with implementation, daily operation, and troubleshooting. Hopefully, this in-depth series has heightened your understanding of this sometimes complex feature. Unfortunately, articles are open for interpretation. What one reader finds illuminating, another reader finds confusing. The only "real" way to understand the SMTP MTA is to use it. When I say "use it," I don't mean install it and then walk away. I'm talking about actively using the feature and analyzing what's going on. Therefore, here are a few tips for "active" MTA use:

- The SMTP MTA has four different logging levels (minimal, normal, informational, and verbose). The higher the logging level, the more activity information appears on the server console and in the Notes log. When you're testing the MTA or using it for the first time, I suggest setting the logging level to "verbose" so you can see as much of the activity as possible. Don't forget, however, to lower this setting once you've gotten comfortable with the feature. Otherwise, you'll have an enormous Notes log database.

- The SMTP MTA includes nine different NOTES.INI settings, which allow you to "trace" a particular MTA activity. Add one of these settings to your NOTES.INI file and restart your MTA, or wait for the configuration to reload. In most cases, you'll receive more information about a particular MTA activity than you ever wanted. Most of the information appears on the server console, but even more of the detailed trace information appears in the Notes log database. Personally, I like to enable a particular setting, send some messages, and then review the trace results in the Notes log. The information is great for troubleshooting a problem, or learning more about what the MTA is doing. For details about each trace setting, refer to the **Domino Administrator's Guide**. Don't forget to remove these trace settings once you're done experimenting, otherwise you'll have a massive Notes log database.
- Read the Release Notes. Each new iteration of the MTA comes with a new set of Release Notes. These notes are historical. Therefore, everything that appeared in the last set of Release Notes also appears in the new set. These notes are a gold mine of the latest and greatest information about the MTA. Unfortunately, the Release Notes are often overlooked by most administrators.

Where do we go from here

The SMTP MTA is an intricate feature. This three-part series has taken significant steps towards revealing "how the MTA works;" however, there is still more to talk about. In the near future, we'll feature articles that discuss the SMTP MTA's international character sets, the anti-spam feature, and how the MTA is becoming a core component of Notes mail in Release 5.0. Until that time, keep working with the MTA. And as always, please tell us what you think about our articles.

A guide to Tell commands (sidebar)

Tell commands allow you to easily start and stop individual MTA tasks, and display important information about the MTA such as version and current configuration. Enter these commands at the server console.

Tell Command	**Description**
TELL SMTPMTA QUIT	Shuts down the SMTP MTA.
TELL SMTPMTA STOP DRTASK	Stops the Delivery Report Task.
TELL SMTPMTA STOP INBOUND TRANSPORT	Stops the Inbound Session Controller. Prior to stopping, the controller will stop all of its associated session handlers.
TELL SMTPMTA STOP OUTBOUND TRANSPORT	Stops the Outbound Session Controller. Prior to stopping, the controller will stop all of its associated session handlers.
TELL SMTPMTA STOP INBOUND CONVERSION	Stops the Inbound Message Handler.
TELL SMTPMTA STOP OUTBOUND CONVERSION	Stops the Outbound Message Handler.
TELL SMTPMTA START DRTASK	Starts the Delivery Report Task.
TELL SMTPMTA START INBOUND TRANSPORT	Starts the Inbound Session Controller. The controller then spawns session handlers as needed.
TELL SMTPMTA START OUTBOUND TRANSPORT	Starts the Outbound Session Controller. The controller then spawns session handlers as needed.
TELL SMTPMTA START INBOUND CONVERSION	Starts the Inbound Message Handler.
TELL SMTPMTA START OUTBOUND CONVERSION	Starts the Outbound Message Handler.
TELL SMTPMTA SHOW VERSION	Displays the version number of the SMTP MTA currently running.

TELL SMTPMTA SHOW CONFIG	Displays all of the SMTP MTA’s current configuration settings.
TELL SMTPMTA REFRESH	Forces all of the MTA tasks to reload their configuration.
TELL SMTPMTA STOP	Stops all MTA tasks except the Add-In Controller.
TELL SMTPMTA START	Starts all MTA tasks except the Add-In Controller. This command is NOT interchangeable with LOAD SMTPMTA, which starts the Add-In Controller.
TELL SMTPMTA STOP INBOUND	Stops all inbound tasks: Inbound Message Handler and Inbound Session Controller.
TELL SMTPMTA START INBOUND	Starts all inbound tasks: Inbound Message Handler and Inbound Session Controller.
TELL SMTPMTA STOP OUTBOUND	Stops all outbound tasks: Outbound Message Handler and Outbound Session Controller.
TELL SMTPMTA START OUTBOUND	Starts all outbound tasks: Outbound Message Handler and Outbound Session Controller.