

Out of the Inbox: Mail processing with the new R5 mail agents

by Julie Kadashevich

[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.]

Ever wanted to process incoming mail *before* it reached a user's mail box? In Notes R5, you can do it!

Notes R5 includes new *pre-delivery* mail agents that you can set up to run before new mail arrives. This means that you can do things like:

- File newly delivered messages into a folder other than the Inbox when those messages meet certain conditions. For example, you may want to file bug reports into a Bugs folder. Because the agent runs when the message is delivered, the user does not see the message appear in the Inbox and then move to another folder.
- Remove large attachments from messages before delivering those messages to the user's mail box. The agent removes the attachment before the entire message writes to the user's mail box; thus, you can conserve disk space and eliminate the need to replicate extra data.
- Determine (based on something about the incoming message) that the message should not be delivered and delete it. For example, you may want to delete messages from a known "Spam" user. Again, you can save both your disk and replication resources.

This article will talk more about pre-delivery mail agents, and how you can use them to perform these operations. We will cover the agent's behavior, the restrictions imposed on this type of agent, and the situations where you should use the agents, versus when you should use the post-delivery mail agents. Then, we'll look at how to debug the agents, and go over some specific examples. You'll see how these agents can be useful for efficiency and storage capacity, as well as usability.

Note: Prior releases of Domino and Notes are not aware of pre-delivery mail agents and the new trigger type "Before new mail arrives." If an R5 server that has pre-delivery agents enabled fails over to a pre-R5 server, the router will deliver the mail without executing the pre-delivery agent. If you create a pre-delivery agent on a Notes R5 client and then attempt to edit or turn it off from a pre-R5 client, you will receive an error that the agent has an unknown trigger.

An introduction to pre-delivery mail agents

Mail agents in R5 now come in two types of flavors according to when they run: *before* new mail arrives or *after* new mail arrives. The new *pre-delivery* mail agents are identified by the new agent trigger "Before new mail arrives." The old-style R4.x mail agents are still supported in R5, and are now identified by the agent trigger "After new mail arrives." We will refer to these old-style agents as *post-delivery* mail agents. The following screen shows the new mail agent triggers:

Name:

☒ Shared Agent

When should this agent run?

Before New Mail Arrives

Before New Mail Arrives

After New Mail Has Arrived

If Documents Have Been Created or Modified

If Documents Have Been Pasted

On Schedule More Than Once A Day

demo (Agent) : Action

Run

The pre-delivery mail agents process mail before it arrives in the user's mail database; for example, to move incoming mail to a folder. The agents are run by the mail router, so they are guaranteed to run before a new mail message deposits into the user's mail file. In contrast, post-delivery agents run after a mail message arrives in the user's mail database. So, you can use post-delivery agents for operations that do not depend on the timing of mail delivery; for example, to respond to mail messages.

Although you can use both types of mail agents in the same mail database, you can enable only one pre-delivery mail agent at a time. You can have an unlimited number of disabled pre-delivery mail agents. In addition, you can have an unlimited number of enabled and disabled post-delivery agents. The following screen shows the agent view of a mail database containing both pre-delivery and post-delivery mail agents. Notice that the blue arrow below the checkbox identifies the currently active (enabled) pre-delivery agent:

New Agent				
	Name/Comment	Alias	Trigger	Owner
<input type="checkbox"/>	File bug reports into bugs folder		Before mail arrives	Shared
<input checked="" type="checkbox"/>	Forward mail to admin assistant		After mail arrives	Shared
<input checked="" type="checkbox"/>	Notify sender that attachment was remov		After mail arrives	Shared
<input checked="" type="checkbox"/>	Remove large attachments		Before mail arrives	Shared
<input type="checkbox"/>	Remove old mail		Scheduled	Shared

The order of execution

If you enable both pre-delivery and post-delivery mail agents in a database, the pre-delivery agent always execute first (before the mail message is deposited into the database). Then, the post-delivery agents execute (after the mail message has been deposited into the database). The post-delivery agents are triggered by the new message, even if the pre-delivery agent filed it into a folder other than the Inbox. If the pre-delivery mail agent deletes a mail message, the message is never delivered to the database. This means that the post-delivery agents are not invoked for this particular message.

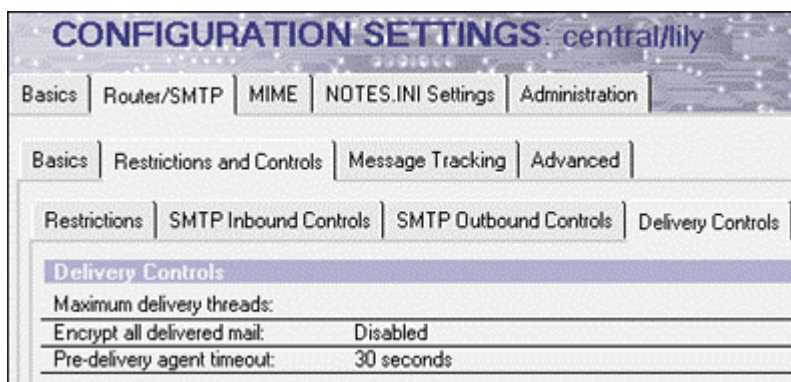
In addition to mail agents, R5 provides two other mail filtering options: new router controls and mail template rules. You can use the built-in router controls to allow mail only from designated domains, allow mail from designated organizations, deny mail from designated organizations, and to impose size restrictions on incoming mail. With the mail template rules, you can specify to watch for messages from a certain sender (such as your boss), or messages that contain a certain subject, and then select what to do when those messages arrive (copy or move them to a folder, delete them, or change the importance of the message).

So, if you use all of the mail filtering options in R5, the options execute in the following order:

1. The router controls verify that a message is allowed to route through the domain.
2. The pre-delivery mail agent processes the incoming message.
3. The action specified in the mail template rules occurs.
4. The post-delivery mail agents process the message.

Configuration settings

Because pre-delivery mail agents are run by the router, Agent Manager settings have no effect on these agents. Therefore, these agents have separate configuration settings. First, the pre-delivery mail agents have a separate maximum execution time, specified in the "Pre-delivery agent timeout" field in the "Router/SMTP" /Restrictions and Controls/Delivery Controls tab of a server's Configuration Settings document. (shown below) By default, this setting is 30 seconds. (LotusScript and Java agents that run by all other triggers use the execution time specified in the Server Tasks/Agent Manager tab of the Server document.)



If the pre-delivery agent exceeds the maximum timeout, it aborts in the middle of its execution. The terminate event fires, and thus gives you, the agent writer, a chance to do some minimum amount of work to clean up. If you write the agent in LotusScript, any files that the agent opened close automatically. If you write the agent in Java, the JVM (Java Virtual Machine) schedules all objects used in the agent for garbage collection. Although it may not happen immediately, the JVM closes any open files when the corresponding objects are collected.

In addition, you can control the number of concurrent pre-delivery mail agents that can execute by configuring the maximum delivery threads for the router, specified in the "Router/SMTP" /Restrictions and Controls/Delivery Controls tab of a server's Configuration Settings document. The number of delivery threads can vary from 3 to 25. By default, Domino determines the number of delivery threads based on the size and performance characteristics of the server.

Designing with pre-delivery mail agents

When designing your mail processing application, you can put both pre-delivery and post-delivery mail agents to work for you. If possible, try to separate your processing into whatever is "critical" to do before a message reaches the user's mail box, and whatever can occur after the message reaches the user's mail box. For example, you might want to have a pre-delivery agent strip off large attachments, and set a flag when it performs the operation. Then, the post-delivery agent can notify the mail sender that the message contained an attachment that was too large to deliver. To see a summary of what pre-delivery and post-delivery mail agents can do, see the sidebar ["Comparing pre-delivery and post-delivery mail agents."](#)

When you begin designing your pre-delivery mail agents, you'll need to keep the following things in mind:

- Pre-delivery agents have built-in restrictions to ensure their efficiency
- There are some key coding differences for pre-delivery and post-delivery mail agents
- Pre-delivery agents can run automatically when servers failover

- Folder operations work differently for pre-delivery agents
- Debugging pre-delivery mail agents is different than other agents

The next sections describe each of these areas in more detail.

Built-in restrictions for pre-delivery mail agents

Because the router executes the pre-delivery mail agents, it's vital for these agents to be as short and efficient as possible. Otherwise, they may slow down the router, which would be highly undesirable for server performance. So, to help ensure the efficiency of the pre-delivery agents, we placed the following restrictions on these types of agents:

- As mentioned earlier, you can enable only one pre-delivery agent per database. This allows Domino to retrieve an agent in an efficient manner for any number of users.
- You cannot set up pre-delivery mail agents to call other agents. This facilitates an efficient caching scheme.
- You cannot set up pre-delivery mail agents to modify attachments. The agents can only examine attachments and detach them. Currently, there are no methods that can modify attachments; you can only modify them by using an OLE operation. Therefore, this restriction means that the methods `Activate` and `DoVerb` on `NotesEmbeddedObject` are not allowed.
- As mentioned earlier, the maximum execution time for pre-delivery agents is separate from other agents, and we recommend you keep the setting at a much smaller number than for other agents. By default, the setting is 30 seconds.

Coding differences for pre-delivery and post-delivery mail agents

Remember that pre-delivery agents execute before the new mail message is written to the user's mail box. This means that as an agent writer, you will not find this new document in any collection obtained from the user's mail database. The new document is available only through the document context. In general, you will need to slightly modify the logic that you used in R4.x mail agents to work with the new mail trigger (that is, you will need to change how you obtain the document that is being delivered).

Another difference between post-delivery (R4.x) mail agents and the new pre-delivery agents is that with the new agent, you know that it will operate on only one document -- the new mail message that the router is delivering. In contrast, a post-delivery mail agent can operate on any number of documents, because the agent runs on all new documents that arrived in the user's mail database since the last time it ran.

Agent failover for pre-delivery mail agents

The router executes pre-delivery mail agents on the same server where it delivers mail. If mail delivery fails over to another server in a cluster, the agent execution fails over as well (provided the second server is also an R5 server). That is, the router on the second server automatically runs the pre-delivery mail agent on any mail it delivers.

In contrast, post-delivery agents are designed by default to run on the home mail server of the agent signer (the person who last modified the agent). Before the agents run, they perform a check to see if the current server is the home mail server of the agent signer. The home mail server is determined by taking the name of the agent signer, performing a lookup in the Domino Directory, and retrieving the mail server from the user's Person document. If the home mail server is not the same as the server on which the agent is attempting to run, the agent does not run. This means that the post-delivery agent does not run unless it resides on the user's mail server. Mail-in databases also should reside on the same server as the mail server listed in the user's Person document. You can change this default behavior by setting the `NOTES.INI AMgr_DisableMailLookup` variable to 1. This setting suppresses the check for the user's home mail server and allows the agent to run on any server. Be aware that depending on the logic of your agent, allowing the agent to run on any server may cause replication conflicts. (This Agent Manager setting has no effect for pre-delivery mail agents, because the router runs those agents.) For more information on dealing with mail agents, see "[Troubleshooting agents](#)."

Managing folders in pre-delivery mail agents

Pre-delivery mail agents operate on a message while it's in transit -- that is, before the router delivers the message to the user's mail box. This results in the following special behavior in folder manipulation operations:

- When the router delivers the message to the user's mail database, only one (the first) PutInFolder operation takes effect. If your agent uses LotusScript or Java, the following runtime error appears if more than one PutInFolder operation is detected: "Invalid sequence of operations in mail pre-delivery agent." If you are using a simple agent and have multiple move operations, the Agent Log generates a warning notifying you which folder operations have been ignored.

Because only one PutInFolder operation can take effect, any newly delivered documents can appear in a maximum of two folders (Inbox and any other folder), but in any number of views at the same time. Note that views and folders are two different things. Membership in a view is determined by the selection formula for the view (All Documents selects all documents). For example, you can create a view called "All Mail From Julie" that selects all documents that have "Julie" in the From field. You can define any number of such views, and one document can match any number of these criteria so that it can appear in any number of views. Folders, on the other hand, are populated explicitly. By default, the router inserts a document into the Inbox folder on delivery. The agent can decide not to do that, and can also decide to put the document into another folder. So, when you use a pre-delivery mail agent, the document can appear in zero, one, or two folders, but in any number of views.

- If the pre-delivery mail agent removes the message from the Inbox via the operation RemoveFromFolder("{\$InBox}"), the mail message is delivered, but only appears in the All Documents view.
- If you want a new mail document to appear in only a different folder and not in the Inbox, the agent needs to perform two folder operations: PutInFolder("NewFolder") and RemoveFromFolder("{\$InBox}").

The following table summarizes the sequences and results of folder methods.

Method 1	Followed by Method 2	Result
PutInFolder("A")	•	Document in Folder "A", Inbox, and all Documents
PutInFolder("A")	RemoveFromFolder("{\$InBox}")	Document in Folder "A" and all Documents
RemoveFromFolder("{\$InBox}")	•	Document in All Documents only
Remove	•	Document not delivered
PutInFolder("A")	PutInFolder("B")	Error
RemoveFromFolder("B")	•	Error
PutInFolder("A")	RemoveFromFolder("B")	Error

Debugging pre-delivery mail agents

Because the pre-delivery mail agent operates on the message being delivered, you cannot test or run the logic of this agent effectively as a manual agent. If you do, you will receive the error "No documents have been selected" since the context is not set properly.

However, you have the following options for debugging pre-delivery mail agents:

- You can change the agent trigger to another type, such as "Selected documents," and then verify that the agent logic works correctly when the agent context is the document selected in a view. When you are satisfied with the logic, switch to the "Before mail arrives" trigger.
- You can debug the logic of the agent by writing debug information to the Agent Log.
- You can use MessageBox or Print methods to print debug information to the server console and server log.

Since the router executes pre-delivery mail agents, the Agent Manager NOTES.INI settings for logging and debugging have no effect. To control the errors that log to the server console for pre-delivery agents, you can specify the "Logging level" setting for the router in the "Router/SMTP" /Advanced/Controls tab of a server's Configuration Settings document. You can set the Logging level to Minimal, Normal (the default), Informational, or Verbose. If you set this variable to Verbose, your agent errors log to the console. Note that you will also get a very verbose output of the router operations. By using this separate setting to control the output of pre-delivery mail agents, you can tune the performance of the router, while still having your other agents generate as much output as needed for your other operations.

You will notice that the output from the pre-delivery agent on the server console is prefixed by "Addin:", as shown in the following:

"Addin: Agent message box: Message generated by mail pre-delivery agent."

This happens because the messages are generated by the router task (which is executing Agent Manager routines via API calls) and the router task is a server add-in. The same prefix appears on the error messages generated by the agent to the server console. For more general information on debugging agents, see "[Troubleshooting agents](#)."

Examples of pre-delivery mail agents

Now let's put everything we learned about the pre-delivery mail agents into action. We will go over the following examples of some typical operations you can perform with the pre-delivery agents:

- Filing a message into a folder other than Inbox based on the subject of the message
- Detaching attachments if they are larger than our threshold
- Deleting a message and forwarding a copy to another person
- Splitting a task into critical and non-critical portions, and using both types of mail agents to perform the task

Please note that the sample agents include a lot of debugging code for illustration purposes. To write these agents for a production system, you should remove all debug code (including the writing to the NotesLog, print, and MessageBox statements) after you complete your testing to make the agents as small and as efficient as possible.

Example One: Filing a message

This LotusScript agent files a message into a folder other than Inbox if the subject of the message is "Vacation request."

Sub Initialize

```
Dim session As New NotesSession
Dim note As NotesDocument
Dim dbug As NotesLog
Dim db As NotesDatabase
Dim it As NotesItem
```

```
Set session = New NotesSession
```

```
Set sourcedb = session.CurrentDatabase
```

```
REM Log steps in our processing for debug purposes
```

```
Set dbug = New NotesLog("Router log")
```

```
dbug.LogActions = True
```

```
dbug.OpenAgentLog
```

```
dbug.LogAction("begin")
```

```
Set db = session.CurrentDatabase
```

```
REM Make sure we have the note set correctly
```

```
If db Is Nothing Then dbug.LogAction("db is not set") Else dbug.LogAction("db is set")
```

```
Set note = session.DocumentContext
```

```
If note Is Nothing Then dbug.LogAction("note is not set") Else dbug.LogAction("note is set")
```

```
REM Note the Subject of all messages
```

```
dbug.LogAction("Subject ->" + note.Subject(0))
```

```
REM Is this message has the special subject, store it in the special folder
```

```
If note.Subject(0) = "vacation request" Then
```

```
    Call note.PutInFolder( "Vacation" )
```

```
    REM PutInFolder leaves a message in the Inbox view as well.
```

```
    REM Since we want to have it only the Vacation Folder we need to remove it from Inbox
```

```
    Call note.RemoveFromFolder("$InBox")
```

```
    dbug.LogAction("File into Vacation Folder")
```

```
End If
```

```
dbug.LogAction("done")
```

```
dbug.Close
```

```
End Sub
```

Example Two: Detaching attachments

This LotusScript agent detaches attachments that are bigger than a certain size (MaxSize).

```
Sub Initialize
```

```
    Dim session As New NotesSession
```

```
    Dim db As NotesDatabase
```

```
    Dim doc As NotesDocument
```

```
    Dim dbug As NotesLog
```

```
    Dim ritem As Variant
```

```
    Dim fileCount As Integer
```

```
    Dim it As NotesItem
```

```
REM Specify the size limit for attachments
```

```
Const MaxSize = 5000
```

```
fileCount = 0
```

```
Set dbug = New NotesLog("Router log")
```

```
    dbug.LogActions = True
```

```
    dbug.OpenAgentLog
```

```
    dbug.LogAction("begin")
```

```
REM get the incoming mail message
```

```

Set doc = session.documentcontext

REM Log the subject name of the message for debug purposes
Set it = doc.GetFirstItem("Subject")
dbug.LogAction("doc subject from context" + "-> " + it.Text)

Set ritem = doc.GetFirstItem( "Body" )
If ( ritem.Type = RICHTEXT ) Then
    Forall o In ritem.EmbeddedObjects

        REM Note how many files we have processed
        fileCount = fileCount + 1
        dbug.LogAction("file count:"+Cstr(fileCount))

        If ( o.Type = EMBED_ATTACHMENT ) And ( o.FileSize > MaxSize ) Then
            Call o.ExtractFile( "c:\tmp\newfile" & Cstr( fileCount ) )
            Call o.Remove
            REM Note that we removed an attachment
            dbug.LogAction("attachment removed")
            REM Created a field noting that we removed an attachment
            doc.stripped = "yes"
            Call doc.Save( True, True )
        End If
    End Forall
End If
REM Finish up agent log processing
dbug.LogAction("Mail preprocessing agent is done")
dbug.Close
End Sub

```

Example Three: Deleting messages

This LotusScript agent deletes messages that come from "Joe Spam" after forwarding them to the postmaster.

```

Sub Initialize
    Dim session As New NotesSession
    Dim note As NotesDocument
    Dim dbug As NotesLog
    Dim db As NotesDatabase
    Dim it As NotesItem

    Set session = New NotesSession
    Set sourcedb = session.CurrentDatabase

    Set dbug = New NotesLog("Router log")
    dbug.LogActions = True
    dbug.OpenAgentLog
    dbug.LogAction("begin")

    Set db = session.CurrentDatabase
    Set note = session.DocumentContext

    dbug.LogAction("Subject -> " + note.From(0))
    If note.From(0) = "CN=Joe Spam/O=SpamFactory" Then

```

```

        Call note.Send(False, "Administrator/Lily")
        dbug.LogAction("Send memo")
        Call note.Remove(True)
    End If
    dbug.Close
End Sub

```

Example Four: Using both pre-delivery and post-delivery agents

This example shows how to perform only critical mail processing in the pre-delivery agent, and all other processing in the post-delivery agent. Our overall goal is to strip any large attachments from incoming mail and to notify the mail senders that their attachments weren't delivered due to their size. The pre-delivery agent strips off large attachments, since this reduces the overhead and disk space. The post-delivery agent notifies the mail sender, since this part of the task is not time critical.

For the first part of the task, we will use the pre-delivery agent created in Example Two. Notice in that agent that when we remove an attachment, we set the status of a field "stripped" to "yes." In the post-delivery agent, we will check that field and if it was set, we will generate a reply to the sender.

```

Sub Initialize
    Dim session As New NotesSession
    Dim db As NotesDatabase
    Set session = New NotesSession
    Set db = session.CurrentDatabase
    Set docs = db.UnprocessedDocuments
    Count = docs.Count

    REM if we have new mail start processing
    If docs.Count > 0 Then
        For n = 1 To docs.Count
            Set memo = docs.GetNthDocument(n)
            If Not( memo.SentByAgent ) Then
                REM if attachemetns were stripped, send a reply
                If (memo.stripped(0) = "yes") Then
                    Set reply = memo.CreateReplyMessage( False )
                    reply.Subject = "Re: " & memo.Subject( 0 )
                    reply.Body = "The message you mailed contained attachments that were too large. They were
removed before mail delivery."
                    Call reply.Send( False )
                End If
            End If
            Call session.UpdateProcessedDoc(memo)
        Next
    End If
End Sub

```

Conclusion

This article introduced you to the new pre-delivery agent, and how you can use this type of agent to process mail *before* it reaches a user's mail box. We covered some tips for how you can begin designing with the pre-delivery agent, as well as some typical examples. I hope that this article gives you enough ammunition to use the new type of agents with confidence and makes it easier and quicker for you to get your job done.

ABOUT THE AUTHOR

Julie Kadashevich came to Iris in March of 1997 after working at FTP Software on Java and C++ mobile agent technology. She has been focusing on the Agent Manager as well as Java. Previously, she worked in the area of applied Artificial Intelligence at Wang Labs and received five patents in the field. She has Bachelor's and Master's degrees from Boston University, both in computer science. Outside of work, her new hobby is kayaking. She recently bought her own kayak and is working on taming it.

Copyright 1999 Iris Associates, Inc. all rights reserved.