www.novell.com/documentation

Troubleshooting 3: Message Flow and Directory Structure

GroupWise 2012

August 2014

Novell.

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see the Novell Documentation Web site (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide 7

Part I Message Flow Diagrams 9

1 Message Delivery in the Local Post Office 11

1.1 Online Mode . . . 11
1.2 Caching Mode . . . 13

2 Message Delivery to a Different Post Office 17

2.1 TCP/IP Link Open: Transfer between Post Offices Successful . . . 17
2.2 TCP/IP Link Closed: Transfer between Post Offices Delayed . . . 20

3 Message Delivery to a Different Domain 23

3.1 TCP/IP Link Open: Transfer between Domains Successful. . . . 23
3.2 TCP/IP Link Closed: Transfer between Domains Delayed. . . . 26

4 Message Delivery to and from the Internet 31

4.1 TCP/IP Link Open: Outbound Transfer to the Internet Successful . . . 31
4.2 TCP/IP Link Closed: Outbound Transfer to the Internet Delayed or Unsuccessful . . . 34
4.3 Mapped/UNC Link Open: Outbound Transfer to the Internet Successful . . . 37
4.4 Mapped/UNC Link Closed: Outbound Transfer to the Internet Delayed or Unsuccessful . . . 39
4.5 TCP/IP Link Open: Inbound Transfer from the Internet Successful . . . 41
4.6 Mapped/UNC Link Open: Inbound Transfer from the Internet Successful. . . . 43

5 Administrative Database Update 45

Part II Directory Structure Diagrams 47

6 Message Transfer/Storage Directories 49

6.1 Domain Directory . . . 49
6.1.1 *domain* directory . . . 50
6.1.2 wpcsin directory . . . 50
6.1.3 wpgate directory . . . 53
6.1.4 wpcsout directory. . . . 53
6.1.5 mtaname file . . . 54
6.1.6 wpdomain.db file . . . 54
6.1.7 wpdomain.dc file . . . 55
6.1.8 wphost.dc file . . . 55
6.1.9 gwdom.dc file. . . . 55
6.1.10 gwpo.dc file . . . 56
6.1.11 viewcopy.log file. . . . 56
6.1.12 uid.run file (Linux only). . . . 56
6.1.13 ncpChecked file (Linux only) . . . 56
6.2 Post Office Directory . . . 56

6.2.1 *post_office* directory . . . 58
6.2.2 wpcsin directory . . . 58
6.2.3 gwdms directory . . . 61
6.2.4 ofmsg directory . . . 62
6.2.5 ofuser directory . . . 63
6.2.6 offiles directory . . . 64
6.2.7 ofviews directory . . . 64
6.2.8 ofwork directory . . . 65
6.2.9 ofdirect directory . . . 65
6.2.10 oftemp directory . . . 65
6.2.11 wpcsout directory . . . 66
6.2.12 wphost.db file . . . 69
6.2.13 gwpo.dc file . . . 69
6.2.14 ngwguard.db file . . . 69
6.2.15 ngwguard.dc file . . . 69
6.2.16 ngwguard.fbk file . . . 70
6.2.17 ngwguard.rfl file . . . 70
6.2.18 ngwcheck.db . . . 70
6.2.19 *mmdd*chk.log . . . 70
6.2.20 viewcopy.log file . . . 70
6.2.21 uid.run file (Linux only) . . . 70
6.2.22 ncpChecked file (Linux only) . . . 71
6.3 MTA Local Queue Directory . . . 71
6.3.1 mslocal directory . . . 71
6.3.2 msglog directory . . . 72
6.3.3 gwinprog directory . . . 72
6.3.4 mshold directory . . . 73
6.3.5 domainms directory . . . 73
6.3.6 *postx* directories . . . 73
6.3.7 *gatewayx* directories . . . 73
6.3.8 *domainx* directories . . . 73
6.3.9 *0-7* directories . . . 74
6.3.10 mtaname files . . . 74
6.3.11 gwvsscan directory . . . 74
6.3.12 mtaconv directory . . . 74
6.4 Internet Agent Queue Directory . . . 74
6.4.1 *domain*\wpgate\gwia directory . . . 75
6.4.2 *gwia* directory . . . 79
6.5 Caching/Remote Mailbox Directory . . . 80
6.5.1 gw*xxxxxx* directory . . . 81
6.5.2 rofdata directory . . . 81
6.5.3 wpcsin directory . . . 83
6.5.4 wpcsout\ofs directory . . . 84
6.5.5 wpgwsend directory . . . 84
6.5.6 wpgwrecv directory . . . 84
6.5.7 remote*n*.log file . . . 84
6.5.8 status*n*.txt file . . . 85

7 Agent Installation Directories 87

7.1 GroupWise Agent Installation (MTA, POA, and DVA) . . . 87
7.1.1 Linux MTA, POA, and DVA Installation Directory . . . 87
7.1.2 Windows MTA, POA, and DVA Installation Directory . . . 94
7.1.3 DVA Working Directory . . . 97
7.2 Internet Agent Installation . . . 98
7.2.1 Linux Internet Agent Installation Directory . . . 98
7.2.2 Windows GWIA Installation Directory . . . 101
7.3 Monitor Agent Installation . . . 103
7.3.1 Linux Monitor Agent Installation Directory . . . 103

7.3.2 Windows Monitor Agent Installation Directory . . . 105

8 Web Application Installation Directories on Your Web Server 109

8.1 OES 11 Apache/Tomcat Installation Directories . . . 109
8.1.1 /etc/apache2 directory . . . 110
8.1.2 /etc/opt/novell/httpd directory . . . 110
8.1.3 /var/opt/novell/tomcat6 directory . . . 111
8.1.4 /usr/sbin directory . . . 112
8.2 OES 2 Linux Apache/Tomcat Installation Directories . . . 112
8.2.1 /etc/apache2 directory . . . 113
8.2.2 /etc/opt/novell/httpd directory . . . 113
8.2.3 /var/opt/novell/tomcat5 directory . . . 114
8.2.4 /usr/sbin directory . . . 115
8.3 SLES 11 Apache/Tomcat Installation Directories . . . 115
8.3.1 /etc/apache2 directory . . . 116
8.3.2 /usr/share/tomcat6 directory . . . 116
8.3.3 /usr/sbin . . . 117
8.4 SLES 10 Apache/Tomcat Installation Directories . . . 118
8.4.1 /etc/apache2 directory . . . 118
8.4.2 /srv/www/tomcat5/base directory . . . 119
8.4.3 /usr/sbin . . . 120
8.5 Windows IIS/Tomcat Installation Directories . . . 120
8.5.1 c:/inetpub/wwwroot directory . . . 121
8.5.2 c:/novell/tomcat6 directory . . . 121
8.6 GroupWise Web Application Working Directory . . . 122
8.6.1 /var/opt/novell/groupwise directory . . . 122

9 GroupWise Software Distribution Directory 125

9.1 Linux Software Distribution Directory . . . 125
9.1.1 /opt/novell/groupwise/software directory . . . 126
9.1.2 agents directory . . . 127
9.1.3 domain directory . . . 128
9.1.4 po directory . . . 129
9.1.5 admin directory . . . 129
9.1.6 internet directory . . . 130
9.1.7 client directory . . . 131
9.1.8 license directory . . . 131
9.1.9 docs directory . . . 131
9.1.10 gwinst directory . . . 132
9.2 Windows Software Distribution Directory . . . 132
9.2.1 c:\grpwise\software directory . . . 133
9.2.2 agents directory . . . 133
9.2.3 domain directory . . . 134
9.2.4 po directory . . . 135
9.2.5 admin directory . . . 135
9.2.6 internet directory . . . 136
9.2.7 client directory . . . 137
9.2.8 license directory . . . 138
9.2.9 docs directory . . . 138

10 GroupWise Client Installation Directory 139

10.1 Windows Client . . . 139
10.1.1 c:\Program Files\Novell\GroupWise . . . 140

Part III Documentation Updates 143

A   April 16, 2013 (GroupWise 2012 SP2) 145

B   September 20, 2012 (GroupWise 2012 SP1) 147

About This Guide

This Novell *GroupWise Troubleshooting 3 Guide* provides diagrams to help you understand the structure and functioning of your GroupWise system.

- Part I, "Message Flow Diagrams," on page 9
- Part II, "Directory Structure Diagrams," on page 47
- Part III, "Documentation Updates," on page 143

Other sources of troubleshooting assistance include:

- Novell Support and Knowledgebase (http://www.novell.com/support)

  To search the GroupWise documentation from the Novell Support Web site, click *Advanced Search*, select *Documentation* in the *Search In* drop-down list, select *GroupWise* in the *Products* drop-down list, type the search string, then click *Search*.
- GroupWise Support Forums (http://forums.novell.com/forumdisplay.php?&f=356)
- GroupWise Support Community (http://www.novell.com/support/products/groupwise)
- GroupWise Cool Solutions (http://www.novell.com/communities/coolsolutions/gwmag)

Audience

This guide is intended for network administrators who install and administer GroupWise.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Additional Documentation

For additional GroupWise documentation, see the following guides at the GroupWise 2012 Documentation Web site (http://www.novell.com/documentation/groupwise2012):

- *Installation Guide*
- *Server Migration Guide*
- *Administration Guide*
- *Multi-System Administration Guide*
- *Interoperability Guide*
- *Troubleshooting Guides*
- *GroupWise User Frequently Asked Questions (FAQ)*
- *GroupWise User Guides*
- *GroupWise User Quick Starts*

I Message Flow Diagrams

This part of *Troubleshooting 3: Message Flow and Directory Structure* helps you understand how messages travel between GroupWise users and how administrative updates to GroupWise databases occur.

- Chapter 1, "Message Delivery in the Local Post Office," on page 11
- Chapter 2, "Message Delivery to a Different Post Office," on page 17
- Chapter 3, "Message Delivery to a Different Domain," on page 23
- Chapter 4, "Message Delivery to and from the Internet," on page 31
- Chapter 5, "Administrative Database Update," on page 45

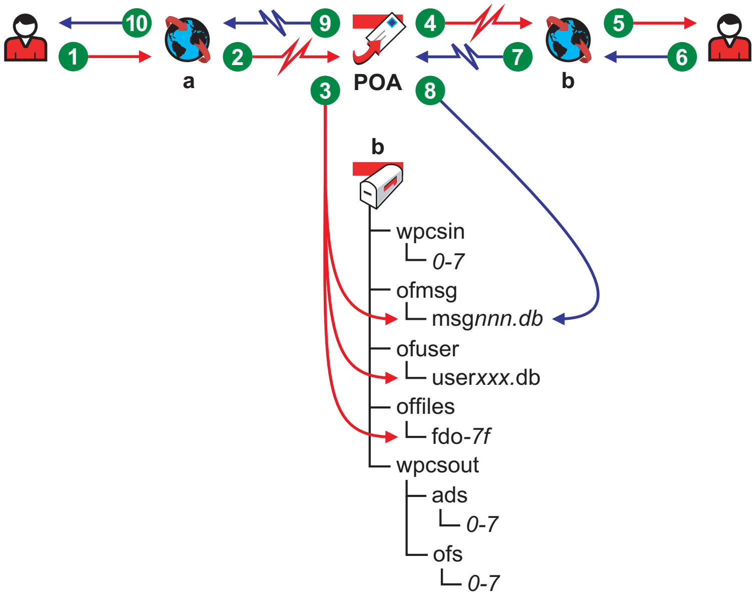
1 Message Delivery in the Local Post Office

The GroupWise 2012 client always uses client/server access to the post office but it can interact with the POA for the post office in different ways. For a review, see "Changing GroupWise Modes" in "Getting Organized" in the *GroupWise 2012 Windows Client User Guide*.

- Section 1.1, "Online Mode," on page 11
- Section 1.2, "Caching Mode," on page 13

1.1 Online Mode

This message flow diagram illustrates how a GroupWise message travels from one user to another in the local post office when the client and POA communicate by way of TCP/IP and the users are accessing their Online mailboxes.

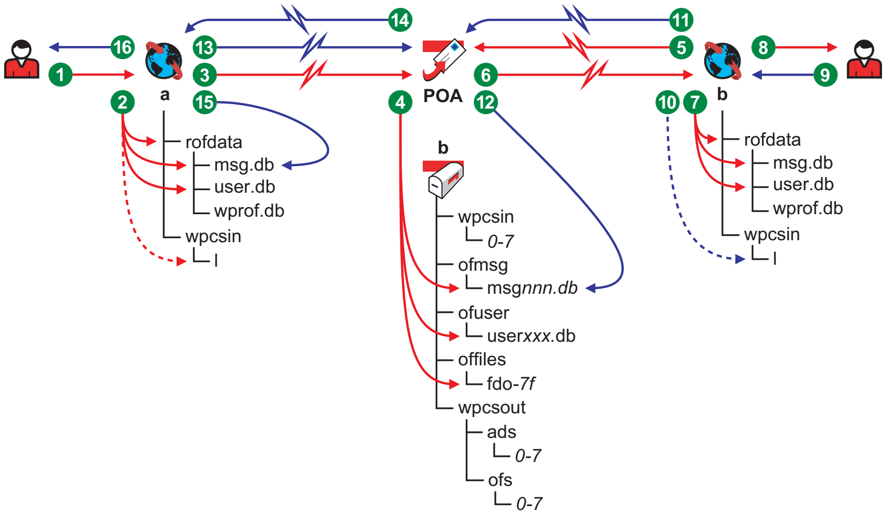


Stage	Icon	Description
❶ Sender	👤	The user sends a message to recipients in the same post office. The access mode setting for the post office is Client/Server Only.
❷ Sender's GroupWise Client	🌐	The GroupWise client communicates with the POA by way of TCP/IP.

Stage	Icon	Description
❸ POA for Local Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (msg*nnn*.db) assigned to the sender. ◆ Creates a pointer in the sender's user database (user*xxx*.db) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd*0-F6* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) The POA also performs the following actions for the recipient: ◆ Creates a pointer in each recipient's user database (user*xxx*.db) to the message in the message database (msg*nnn*.db) so the new message appears in the recipient's mailbox. ◆ Updates the message in the message database (msg*nnn*.db) with a Delivered status for each recipient.
❹ POA for Local Post Office	POA	The POA communicates to the GroupWise client by way of TCP/IP that a new message has arrived.
❺ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
❻ Recipient	👤	Each recipient opens the message in the GroupWise client.
❼ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client communicates the Opened status to the POA by way of TCP/IP.
❽ POA for Local Post Office	POA	The POA receives the Opened status from the GroupWise client and updates the message in the message database with the Opened status for each recipient who opens the message.
❾ POA for Local Post Office	POA	The POA communicate the Opened status to the sender's GroupWise client by way of TCP/IP.
❿ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays a status of Delivered for each recipient (and possibly Opened as well if the recipient has opened the message).

1.2 Caching Mode

This message flow diagram illustrates how a GroupWise message travels from one user to another in the local post office when the users are accessing their Caching mailboxes.



Stage	Icon	Description
1 Sender	👤	The user sends a message to recipients in the same post office. The user is in Caching mode.
2 Sender's GroupWise Client	🌐	The GroupWise client updates the sender's Caching mailbox (by performing the following actions: ◆ Adds the message to the message database (rofdata/msg.db) in the Caching mailbox. This database is local equivalent of the msg*nnn*.db database in the post office. ◆ Creates a pointer in the sender's user database (rofdata/user.db) in the Caching mailbox so the message appears in the sender's mailbox as a sent item. This database is the local equivalent of the user*xxx*.db database in the post office. ◆ Places attachments larger than 2 KB in the rofdata subdirectory in the Caching mailbox and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) There is no local equivalent to the offiles subdirectory in the post office, so attachments are placed directly in the rofdata subdirectory in the Caching mailbox. ◆ Places a copy of the message in the rofdata/wpcsin/1 priority subdirectory to await the next connection to the POA.
3 Sender's GroupWise Client	🌐	In Caching mode, sending a message always initiates an immediate connection with the POA in order to send the message. The GroupWise client communicates the message to the POA and deletes the copy in the rofdata/wpcsin/1 priority subdirectory when the POA has processed the message.

Stage	Icon	Description
❹ POA for Local Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender to update the sender's Online mailbox: ◆ Adds the message to the message database (`msgnnn.db`) assigned to the sender in the post office. ◆ Creates a pointer in the sender's user database (`userxxx.db`) in the post office. ◆ Places attachments larger than 2 KB in one of the `post_office/offiles/fd0-F6` subdirectories in the post office and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) The POA also performs the following actions for the recipients to update their Online mailboxes: ◆ Creates a pointer in each recipient's user database (`userxxx.db`) to the message in the message database (`msgnnn.db`) in the post office so that the new message appears in each recipient's mailbox. ◆ Updates the message in the message database (`msgnnn.db`) in the post office with a Delivered status for the recipients.
❺ POA for Local Post Office	POA	Because the recipients are also in Caching mode, they do not receive immediate notification that a new message has arrived in their Online mailboxes. Based on the *Send/Retrieve All Marked Accounts Every nn Minutes* option under *Accounts* > *Account Options* > *General*, each recipient's GroupWise client sends a request to the POA for items that have arrived in the recipient's Online mailbox since the last connection with the POA.
❻ POA for Local Post Office	POA	The POA responds by sending information to update each recipient's Caching mailbox and communicates to the GroupWise client that a new message has arrived.
❼ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client updates the recipient's Caching mailbox by performing the following actions: ◆ Adds the message to the message database (`rofdata/msg.db`) in the recipient's Caching mailbox. ◆ Creates a pointer in the recipient's user database (`rofdata/user.db`) to the message in the message database (`rofdata/msg.db`) so that the new message appears in the recipient's Caching mailbox.
❽ Recipient's GroupWise Client	🌐	The Notify component of each recipient's GroupWise client notifies the recipient that a new message has arrived.
❾ Recipient	👤	Each recipient opens the message in the GroupWise client.
❿ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client generates an Opened status and places it in the `rofdata/wpcsin/1` priority subdirectory to await the next connection with the POA.
⓫ Recipient's GroupWise Client	🌐	When each recipient sends a message or the time specified by the *Send/Receive All Marked Accounts Every nn Minutes* option has elapsed, each recipient's GroupWise client connects with the POA and communicates the Opened status to the POA, along with any other data that needs to be uploaded to the recipient's Online mailbox.

Stage	Icon	Description
⓬ POA for Local Post Office	POA	The POA receives the Opened status from the GroupWise client and updates the message in the sender's message database with the Opened status.
⓭ Recipient's GroupWise Client	🌐	Because the sender is in Caching mode, the sender does not immediately receive the Opened status. Based on the sender's actions and caching schedule, the sender's GroupWise client eventually sends a request to the POA for items that have arrived in the sender's Online mailbox since the last synchronization of the Caching mailbox.
⓮ POA for Local Post Office	POA	The POA responds by sending information to update the sender's Caching mailbox and communicates the Opened status to the sender's GroupWise client.
⓯ Recipient's GroupWise Client	🌐	The sender's GroupWise client updates the sender's Caching mailbox by performing the following action: ◆ Updates the message in the message database (`rofdata/msg.db`) with a Delivered and Opened status for each recipient.
⓰ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays a status of Delivered and Opened for each recipient.

2 Message Delivery to a Different Post Office

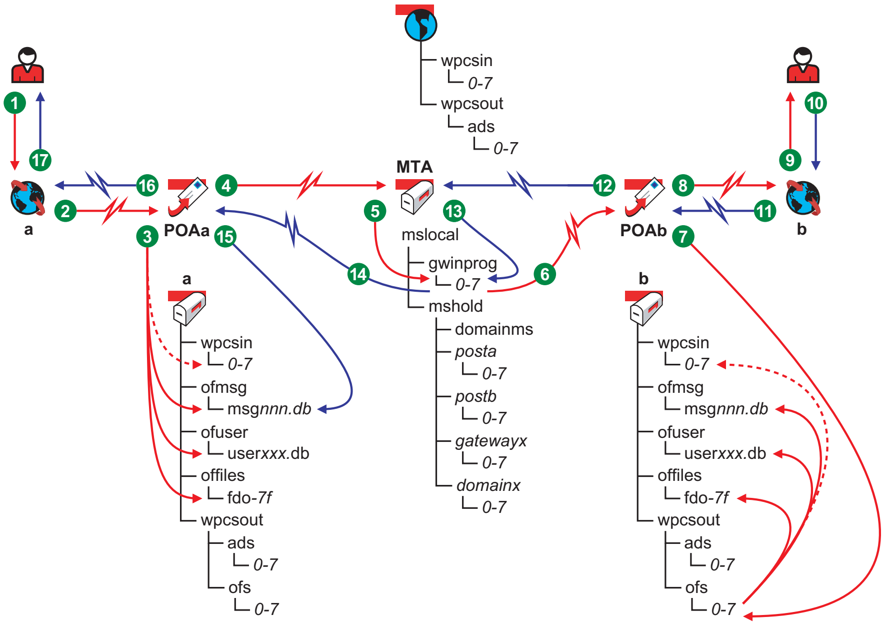
The MTA handles message transfer between post offices.

- Section 2.1, "TCP/IP Link Open: Transfer between Post Offices Successful," on page 17
- Section 2.2, "TCP/IP Link Closed: Transfer between Post Offices Delayed," on page 20

The message flow diagrams in *GroupWise 2012 Troubleshooting 3: Message Flow and Directory Structure* focus on TCP/IP links because they are the most common and convenient (unless you have a post office and a domain on the same server). For diagrams that include mapped/UNC links, see *GroupWise 6.5 Troubleshooting 3: Message Flow and Directory Structure* on the GroupWise 6.5 Documentation Web site (http://www.novell.com/documentation/gw65). For an explanation of link types and link protocols, see "Understanding Link Configuration" in "Domains" in the *GroupWise 2012 Administration Guide*.

2.1 TCP/IP Link Open: Transfer between Post Offices Successful

This message flow diagram illustrates how a GroupWise message travels from one user to another between post offices in the same domain when the TCP/IP link between the post office and the domain is open.

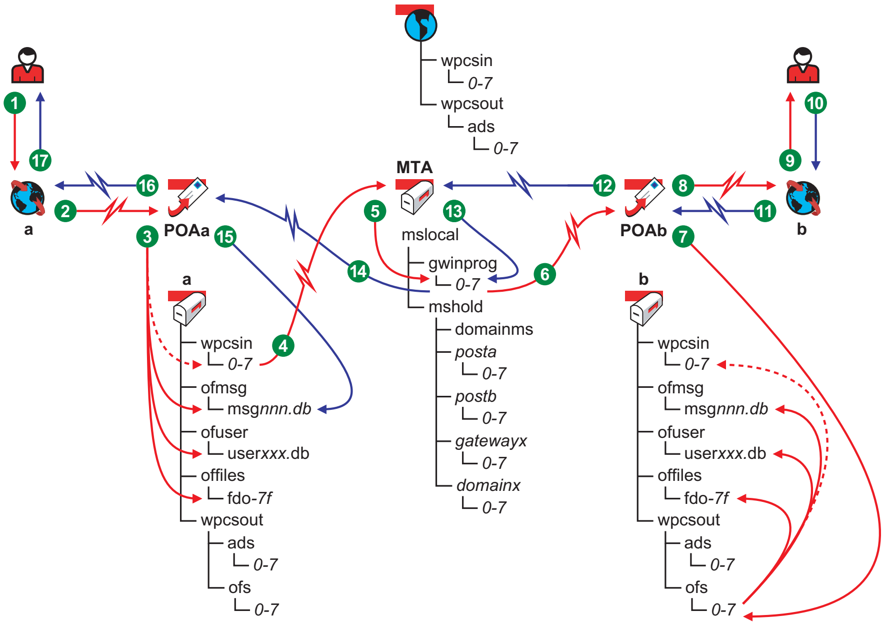


Stage	Icon	Description
❶ Sender	👤	The user sends a message to recipients in a different post office in the same domain. In this diagram, the access mode setting in the local post office is Client/Server Only.
❷ Sender's GroupWise Client	🌐	The GroupWise client communicates the message to the POA by way of TCP/IP.
❸ POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (msg*nnn*.db) assigned to the sender. ◆ Creates a pointer in the sender's user database (user*xxx*.db) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd*0-F6* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority 0-7 subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.
❹ POA for Sender's Post Office	POA	The POA then communicates the message to the MTA by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful. To see what happens if the TCP/IP link to the MTA is closed, see Section 2.2, "TCP/IP Link Closed: Transfer between Post Offices Delayed," on page 20.
❺ MTA for Local Domain	MTA	The MTA receives the message and places it into the MTA "in progress" (gwinprog) queue.
❻ MTA for Local Domain	MTA	The MTA then communicates the message to the POA in the recipient's post office by way of TCP/IP. When the transmission is successful, the MTA deletes the message from the MTA "in progress" queue. If the TCP/IP link to the recipient's post office is closed, the message is placed in the closed post office's holding queue in the MTA's mslocal directory for later transfer. The resulting message flow is parallel to what occurs when a domain is closed. See Section 3.2, "TCP/IP Link Closed: Transfer between Domains Delayed," on page 26 for a similar message flow that illustrates how messages to closed locations are handled.

Stage	Icon	Description
❼ POA for Recipient's Post Office	POA	When the POA receives the new message from the MTA, it places it into the MTA output queue in the post office (`wpcsout/ofs/0-7`) on behalf of the MTA. Then, the POA for the recipient's post office performs the following actions: ◆ Adds the message to the message database (`msgnnn.db`) corresponding to the one assigned to the sender. ◆ Creates a pointer in the recipient's user database (`userxxx.db`) so the new message appears in the recipient's mailbox and updates the notification information in the user database so the recipient can be notified of the message. ◆ Places attachments larger than 2 KB in one of the `post_office/offiles/fd0-F6` subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a Delivered status message in the appropriate priority `0-7` subdirectory of the MTA input queue in the recipient's post office. It also communicates the Delivered status message directly to the MTA by way of TCP/IP. When that transmission is successful, the copy in the MTA input queue is deleted.
❽ POA for Local Post Office	POA	The POA communicates to the GroupWise client by way of TCP/IP that a new message has arrived.
❾ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
❿ Recipient	👤	Each recipient opens the message in the GroupWise client.
⓫ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client communicates the Opened status message to the POA by way of TCP/IP.
⓬ POA for Recipient's Post Office	POA	The POA for the recipient's post office communicates the status message to the MTA by way of TCP/IP.
⓭ MTA for Local Domain	MTA	The MTA places the status message into the MTA "in progress" (`gwinprog`) queue.
⓮ MTA for Local Domain	MTA	The MTA communicates the status message to the POA for the sender's post office by way of TCP/IP.
⓯ POA in Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (`msgnnn.db`) with the Delivered status information (and possibly Opened as well if the recipient has opened the message).
⓰ POA for Local Post Office	POA	The POA communicates the status to the sender's GroupWise client by way of TCP/IP.
⓱ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays a status of Delivered for each recipient (and possibly Opened as well if the recipient has opened the message).

2.2 TCP/IP Link Closed: Transfer between Post Offices Delayed

This message flow diagram illustrates how a GroupWise message travels from one user to another between post offices in the same domain when the TCP/IP link between the post office and the domain is closed.



Stage	Icon	Description
❶ Sender		The user sends a message to recipients in a different post office in the same domain. In this diagram, the access mode setting in the local post office is Client/Server Only.
❷ Sender's GroupWise Client		The GroupWise client communicates the message to the POA by way of TCP/IP.
❸ POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (msg*nnn*.db) assigned to the sender. ◆ Creates a pointer in the sender's user database (user*xxx*.db) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd*0*-*F6* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority 0-7 subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.

Stage	Icon	Description
❹ POA for Sender's Post Office	POA	The POA then attempts to communicate the message to the MTA by way of TCP/IP, but the MTA does not respond. The POA leaves the copy of the message in the MTA input queue and periodically attempts to contact the MTA. When the MTA responds again, the POA communicates the message and deletes the copy in the MTA input queue after the TCP/IP transmission to the MTA is successful.
❺ MTA for Local Domain	MTA	The MTA receives the message and places it into the MTA "in progress" (gwinprog) queue.
❻ MTA for Local Domain	MTA	The MTA then communicates the message to the POA in the recipient's post office by way of TCP/IP. When the transmission is successful, the MTA deletes the message from the MTA "in progress" (gwinprog) queue. If the TCP/IP link to the recipient's post office is closed, the message is placed in the closed post office's holding queue in the MTA's mslocal directory for later transfer. The resulting message flow is parallel to what occurs when a domain is closed. For a similar message flow that illustrates how messages to closed locations are handled, see Section 3.2, "TCP/IP Link Closed: Transfer between Domains Delayed," on page 26.
❼ POA for Recipient's Post Office	POA	When it receives the new message, the POA for the recipient's post office performs the following actions: ◆ Adds the message to the message database (msg*nnn*.db) corresponding to the one assigned to the sender. ◆ Creates a pointer in the recipient's user database (user*xxx*.db) so the new message appears in the recipient's mailbox and updates the notification information in the user database so the recipient can be notified of the message. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd*0-F6* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a Delivered status message in the appropriate priority 0-7 subdirectory of the MTA input queue in the recipient's post office. It also communicates the Delivered status message directly to the MTA by way of TCP/IP and when that transmission is successful, the copy in the MTA input queue is deleted.
❽ POA for Local Post Office	POA	The POA communicates to the GroupWise client by way of TCP/IP that a new message has arrived.
❾ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
❿ Recipient	👤	Each recipient opens the message in the GroupWise client.
⓫ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client communicates the Opened status message to the POA by way of TCP/IP.
⓬ POA for Recipient's Post Office	POA	The POA for the recipient's post office communicates the status message to the MTA by way of TCP/IP.

Stage	Icon	Description
⓭ MTA for Local Domain	MTA	The MTA places the status message into the MTA “in progress” (`gwinprog`) queue.
⓮ MTA for Local Domain	MTA	The MTA communicates the status message to the POA for the sender’s post office by way of TCP/IP.
⓯ POA in Sender’s Post Office	POA	The POA for the sender’s post office updates the sender’s message database (`msg`*`nnn`*`.db`) with the Delivered status information (and possibly Opened as well if the recipient has opened the message).
⓰ POA for Local Post Office	POA	The POA communicates the Opened status to the sender’s GroupWise client by way of TCP/IP.
⓱ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays a status of Delivered for each recipient (and possibly Opened as well if the recipient has opened the message).

3 Message Delivery to a Different Domain

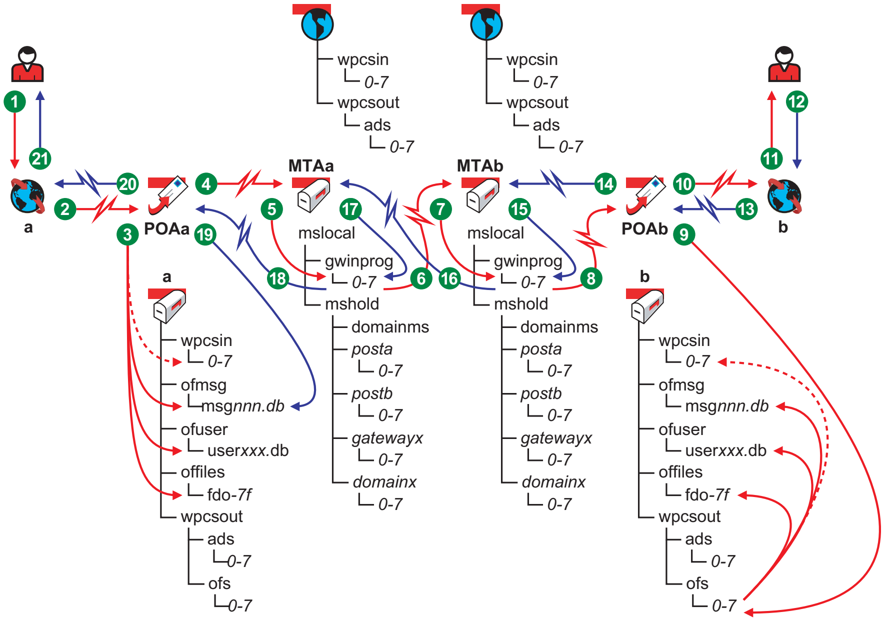
The MTA handles message transfer between domains.

- Section 3.1, “TCP/IP Link Open: Transfer between Domains Successful,” on page 23
- Section 3.2, “TCP/IP Link Closed: Transfer between Domains Delayed,” on page 26

The message flow diagrams in *GroupWise 2012 Troubleshooting 3: Message Flow and Directory Structure* focus on TCP/IP links because they are the most common and convenient (unless you have two domains on the same server). For diagrams that include mapped/UNC links, see *GroupWise 6.5 Troubleshooting 3: Message Flow and Directory Structure* on the GroupWise 6.5 Documentation Web site (http://www.novell.com/documentation/gw65). For an explanation of link types and link protocols, see “Understanding Link Configuration” in “Domains” in the *GroupWise 2012 Administration Guide*.

3.1 TCP/IP Link Open: Transfer between Domains Successful

This message flow diagram illustrates how a GroupWise message travels from one user to another when the domains are connected by a TCP/IP link and the link is open.



Stage	Icon	Description
1 Sender	👤	The user sends a message to recipients in a post office in a different domain. In this diagram, the access mode setting for the local post office is Client/Server Only.

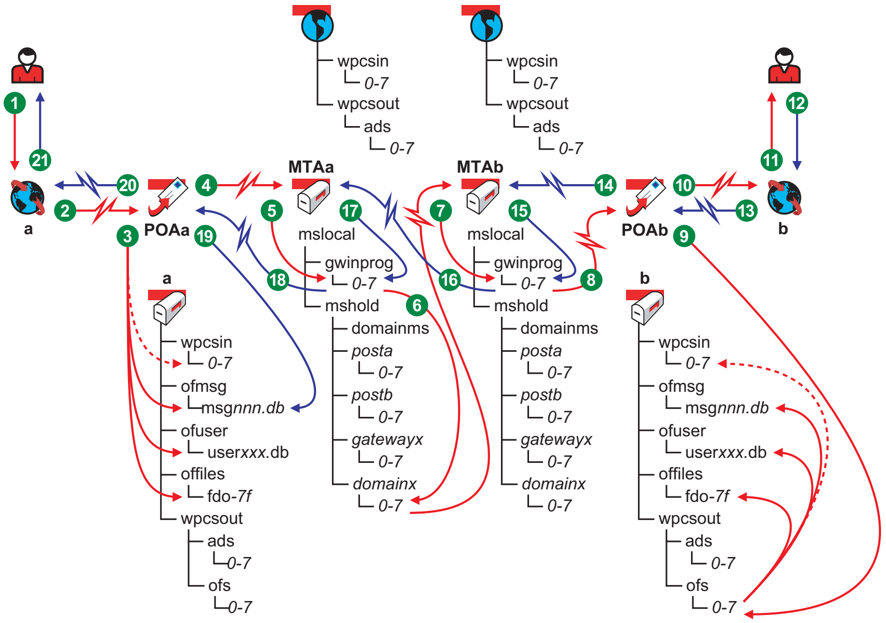
Stage	Icon	Description
❷ Sender's GroupWise Client	🌐	The GroupWise client communicates the message to the POA by way of TCP/IP.
❸ POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (msg*nnn*.db) assigned to the sender. ◆ Creates a pointer in the sender's user database (user*xxx*.db) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd*0-F6* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority 0-7 subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.
❹ POA for Sender's Post Office	POA	The POA then communicates the message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful. To see what would happen if the TCP/IP link to the MTA is closed, see Section 2.2, "TCP/IP Link Closed: Transfer between Post Offices Delayed," on page 20.
❺ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the message and places it into the MTA "in progress" (gwinprog) queue.
❻ MTA for Sender's Domain	MTA	The MTA for the sender's domain then communicates the message to the MTA for the recipient's domain by way of TCP/IP. If the TCP/IP link to the recipient's domain is closed, the message is placed in the closed domain's holding queue in the MTA's mslocal directory for later transfer. See Section 3.2, "TCP/IP Link Closed: Transfer between Domains Delayed," on page 26.
❼ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain receives the message and places it into the MTA "in progress" (gwinprog) queue.
❽ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain then communicates the message to the POA in the recipient's post office by way of TCP/IP.

Stage	Icon	Description
❾ POA for Recipient's Post Office	POA	When it receives the new message, the POA for the recipient's post office performs the following actions: ◆ Adds the message to the message database (msg*nnn*.db) corresponding to the one assigned to the sender. ◆ Creates a pointer in the recipient's user database (user*xxx*.db) so the new message appears in the recipient's mailbox and updates the notification information in the user database so the recipient can be notified of the message. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd*0*-*F6* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a Delivered status message in the appropriate priority 0-7 subdirectory of the MTA input queue in the recipient's post office. It also communicates the Delivered status message directly to the MTA by way of TCP/IP and when that transmission is successful, the copy in the MTA input queue is deleted.
❿ POA for Recipient's Post Office	POA	The POA for the recipient's post office communicates to the GroupWise client by way of TCP/IP that a new message has arrived.
⓫ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
⓬ Recipient	👤	Each recipient opens the message in the GroupWise client.
⓭ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client communicates the Opened status message to the POA by way of TCP/IP.
⓮ POA for Recipient's Post Office	POA	The POA for the recipient's post office communicates the status message to the MTA for the recipient's domain by way of TCP/IP.
⓯ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain places the status message into the MTA "in progress" (gwinprog) queue.
⓰ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain communicates the status message to the MTA for the sender's domain by way of TCP/IP.
⓱ MTA for Sender's Domain	MTA	The MTA for the sender's domain places the status message into the MTA "in progress" (gwinprog) queue.
⓲ MTA for Sender's Domain	MTA	The MTA for the sender's domain communicates the status message to the POA for the sender's post office by way of TCP/IP.
⓳ POA for Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (msg*nnn*.db) with the Delivered status information (and possibly Opened as well if the recipient has opened the message).
⓴ POA for Sender's Post Office	POA	The POA for the sender's post office communicates the status to the sender's GroupWise client by way of TCP/IP.

Stage	Icon	Description
21 Sender		When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays a status of Delivered for each recipient (and possibly Opened as well if the recipient has opened the message).

3.2 TCP/IP Link Closed: Transfer between Domains Delayed

This message flow diagram illustrates how a GroupWise message travels from one user to another when the domains are connected by a TCP/IP link and the link is closed.



Stage	Icon	Description
1 Sender		The user sends a message to recipients in a post office in a different domain. In this diagram, the access mode setting for the local post office is Client/Server Only.
2 Sender's GroupWise Client		The GroupWise client communicates the message to the POA by way of TCP/IP.

Stage	Icon	Description
❸ POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (msgnnn.db) assigned to the sender. ◆ Creates a pointer in the sender's user database (userxxx.db) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the post_office/offiles/fd0-F6 subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority 0-7 subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.
❹ POA for Sender's Post Office	POA	The POA then communicates the message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful. To see what would happen if the TCP/IP link to the MTA is closed, see Section 2.2, "TCP/IP Link Closed: Transfer between Post Offices Delayed," on page 20.
❺ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the message and places it into the MTA "in progress" (gwinprog) queue.
❻ MTA for Sender's Domain	MTA	The MTA for the sender's domain then attempts to communicate the message to the MTA for the recipient's domain by way of TCP/IP, but the recipient MTA does not respond. Therefore, the MTA stores the message in its holding queue for the recipient's domain in the mshold directory. When the MTA in the recipient's domain responds again, the MTA for the sender's domain transfers the delayed message from the domain holding queue to the MTA in the recipient's domain by way of TCP/IP.
❼ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain receives the message and places it into the MTA "in progress" (gwinprog) queue.
❽ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain then communicates the message to the POA in the recipient's post office by way of TCP/IP.

Stage	Icon	Description
❾ POA for Recipient's Post Office	POA	When it receives the new message, the POA for the recipient's post office performs the following actions: ◆ Adds the message to the message database (`msg`*`nnn`*`.db`) corresponding to the one assigned to the sender. ◆ Creates a pointer in the recipient's user database (`user`*`xxx`*`.db`) so the new message appears in the recipient's mailbox and updates the notification information in the user database so the recipient can be notified of the message. ◆ Places attachments larger than 2 KB in one of the *`post_office`*`/offiles/fd`*`0-F6`*subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a Delivered status message in the appropriate priority `0-7` subdirectory of the MTA input queue in the recipient's post office. It also communicates the Delivered status message directly to the MTA by way of TCP/IP and when that transmission is successful, the copy the MTA input queue is deleted.
❿ POA for Recipient's Post Office	POA	The POA for the recipient's post office communicates to the GroupWise client by way of TCP/IP that a new message has arrived.
⓫ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
⓬ Recipient	👤	Each recipient opens the message in the GroupWise client.
⓭ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client communicates the Opened status message to the POA by way of TCP/IP.
⓮ POA for Recipient's Post Office	POA	The POA for the recipient's post office communicates the status message to the MTA for the recipient's domain by way of TCP/IP.
⓯ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain places the status message into the "in progress" (`gwinprog`) queue.
⓰ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain communicates the status message to the MTA for the sender's domain by way of TCP/IP.
⓱ MTA for Sender's Domain	MTA	The MTA for the sender's domain places the status message into the MTA "in progress" (`gwinprog`) queue.
⓲ MTA for Sender's Domain	MTA	The MTA for the sender's domain communicates the status message to the POA for the sender's post office by way of TCP/IP.
⓳ POA for Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (`msg`*`nnn`*`.db`) with the Delivered status information (and possibly Opened as well if the recipient has opened the message).
⓴ POA for Sender's Post Office	POA	The POA for the sender's post office communicates the status to the sender's GroupWise client by way of TCP/IP.

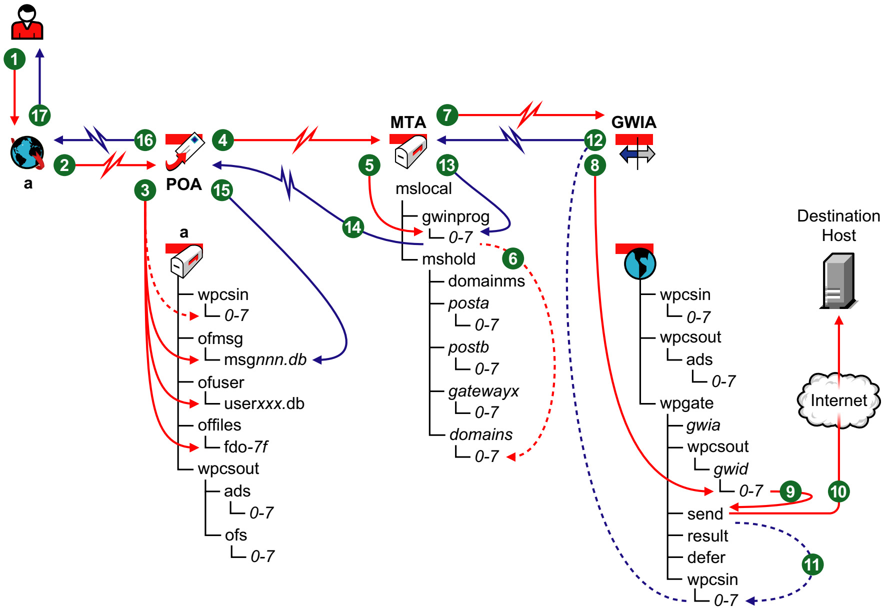
Stage	Icon	Description
㉑ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays a status of Delivered for each recipient (and possibly Opened as well if the recipient has opened the message).

4 Message Delivery to and from the Internet

- Section 4.1, "TCP/IP Link Open: Outbound Transfer to the Internet Successful," on page 31
- Section 4.2, "TCP/IP Link Closed: Outbound Transfer to the Internet Delayed or Unsuccessful," on page 34
- Section 4.3, "Mapped/UNC Link Open: Outbound Transfer to the Internet Successful," on page 37
- Section 4.4, "Mapped/UNC Link Closed: Outbound Transfer to the Internet Delayed or Unsuccessful," on page 39
- Section 4.5, "TCP/IP Link Open: Inbound Transfer from the Internet Successful," on page 41
- Section 4.6, "Mapped/UNC Link Open: Inbound Transfer from the Internet Successful," on page 43

4.1 TCP/IP Link Open: Outbound Transfer to the Internet Successful

This message flow diagram shows how outbound messages travel through the GroupWise directory structure to the Internet when there is a TCP/IP link between the MTA and the GWIA and when the GWIA can communicate successfully with the Internet host to which the message is addressed.

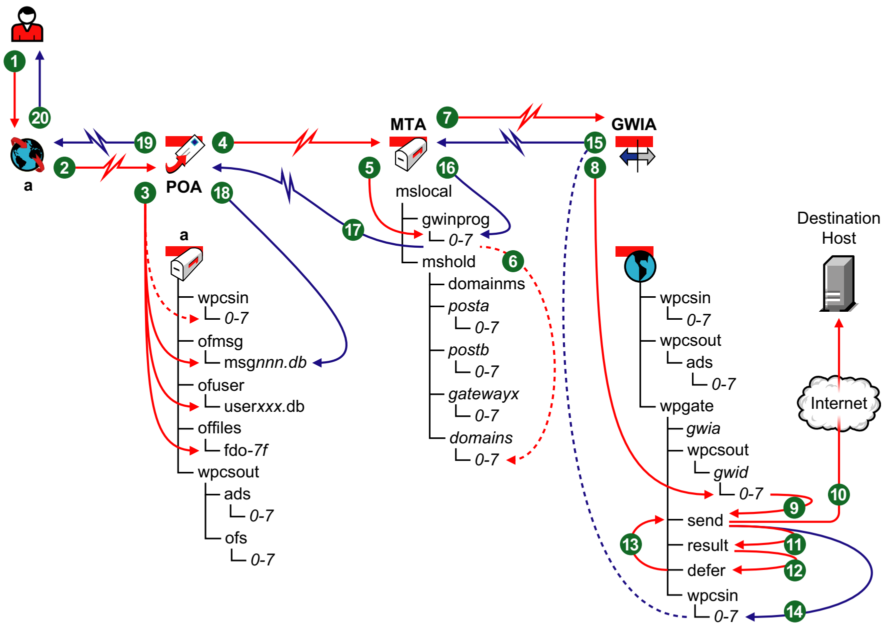


Stage	Icon	Description
❶ Sender	👤	The user sends a message to recipients across the Internet by providing their Internet addresses. In this diagram, the access mode setting for the local post office is Client/Server Only.
❷ Sender's GroupWise Client	🌐	The GroupWise client communicates the message to the POA by way of TCP/IP.
❸ POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (`msg`*`nnn`*`.db`) assigned to the sender. ◆ Creates a pointer in the sender's user database (`user`*`xxx`*`.db`) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *`post_office`*`/offiles/fd`*`0-F6`* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority `0-7` subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.
❹ POA for Sender's Post Office	POA	The POA then communicates the message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful.
❺ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the message and places it into the MTA "in progress" (`gwinprog`) queue.
❻ MTA for Sender's Domain	MTA	The MTA determines that the message must be sent out across the Internet. Because there is a TCP/IP link between the MTA and the GWIA, the MTA creates a copy of the message in the appropriate priority `0-7` subdirectory of the GWIA hold queue (`mslocal/mshold/`*`gatewayx`*`/0-7`), in case the TCP/IP link to the GWIA is currently closed.
❼ MTA for Sender's Domain	MTA	The MTA then communicates the message to the GWIA for the sender's domain by way of TCP/IP, and deletes the copy in the GWIA holding queue because the TCP/IP transfer to the GWIA was successful.
❽ GWIA for Sender's Domain	GWIA	The GWIA receives the message and places it into the MTA output queue (`wpcsout/`*`gwid`*`/0-7`) on behalf of the MTA. The MTA output queue is the GWIA input queue.
❼ GWIA for Sender's Domain	GWIA	The GWIA scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the GWIA object in ConsoleOne. The GWIA picks up the file in binary-encrypted format from the `wpcsout/`*`gwid`*`/0-7` directory and converts it. The GWIA encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the GWIA saves it with S as the first character of the file name and places the message file in the *`domain`*`/wpgate/`*`gwia`*`/send` directory for processing.

Stage	Icon	Description
❿ GWIA for Sender's Domain	GWIA	While the GWIA is processing the message file in the send directory, it changes the first character of the file name to P. When processing is completed, the GWIA sends the message to the destination host across the Internet.
⓫ GWIA for Sender's Domain	GWIA	If the GWIA receives a 250 OK SMTP reply code from the destination Internet host, it places a Transferred status message into the input queue of the MTA for the sender's domain in case the TCP/IP link to the GWIA is currently closed.
⓬ GWIA for Sender's Domain	GWIA	The GWIA then communicates the Transferred status message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful.
⓭ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the Transferred status message and places it into the MTA "in progress" (`gwinprog`) queue for processing.
⓮ MTA for Sender's Domain	MTA	The MTA for the sender's domain communicates the Transferred status message to the POA for the sender's post office by way of TCP/IP.
⓯ POA for Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (`msg`*`nnn`*`.db`) with the Transferred status information.
⓰ POA for Sender's Post Office	POA	The POA for the sender's post office communicates the Transferred status to the sender's GroupWise client by way of TCP/IP.
⓱ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays the Transferred status because the GWIA was able to sent it successfully.

4.2 TCP/IP Link Closed: Outbound Transfer to the Internet Delayed or Unsuccessful

This message flow diagram shows how outbound messages travel through the GroupWise directory structure to the Internet when there is a TCP/IP link between the MTA and the GWIA and when the GWIA cannot communicate successfully with the Internet host to which the message is addressed.



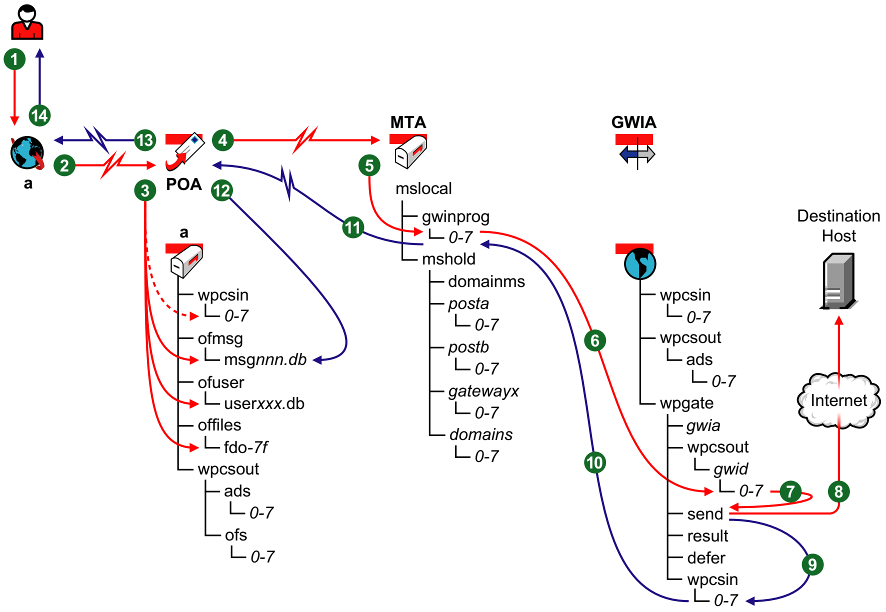
Stage	Icon	Description
1 Sender		The user sends a message to recipients across the Internet by providing their Internet addresses. In this diagram, the access mode setting for the local post office is Client/Server Only.
2 Sender's GroupWise Client		The GroupWise client communicates the message to the POA by way of TCP/IP.
3 POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (`msg`*`nnn`*`.db`) assigned to the sender. ◆ Creates a pointer in the sender's user database (`user`*`xxx`*`.db`) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *`post_office`*`/offiles/fd`*`0-F6`* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority `0-7` subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.

Stage	Icon	Description
❹ POA for Sender's Post Office	POA	The POA then communicates the message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful.
❺ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the message and places it into the MTA "in progress" (`gwinprog`) queue.
❻ MTA for Sender's Domain	MTA	The MTA determines that the message must be sent out across the Internet. Because there is a TCP/IP link between the MTA and the GWIA, the MTA creates a copy of the message in the appropriate priority `0-7` subdirectory of the GWIA hold queue (`mslocal/mshold/`*`gatewayx`*`/0-7`), in case the TCP/IP link to the GWIA is currently closed.
❼ MTA for Sender's Domain	MTA	The MTA then communicates the message to the GWIA for the sender's domain by way of TCP/IP, and deletes the copy in the GWIA holding queue because the TCP/IP transfer to the GWIA was successful.
❽ GWIA for Sender's Domain	GWIA	The GWIA receives the message and places it into the MTA output queue (`wpcsout/`*`gwid`*`/0-7`) on behalf of the MTA. The MTA output queue is the GWIA input queue.
❾ GWIA for Sender's Domain	GWIA	The GWIA scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the GWIA object in ConsoleOne. The GWIA picks up the file in binary-encrypted format from the `wpcsout/`*`gwid`*`/0-7` directory and converts it. The GWIA encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the GWIA saves it with S as the first character of the file name and places the message file in the *`domain`*`/wpgate/`*`gwia`*`/send` directory for processing.
❿ GWIA for Sender's Domain	GWIA	While the GWIA is processing the message file in the send directory, it changes the first character of the file name to P. When processing is completed, the GWIA sends the message to the destination host across the Internet.
⓫ GWIA for Sender's Domain	GWIA	If the GWIA does not receive a 250 OK SMTP reply code from the destination Internet host, the GWIA renames the `P*.*` message file back to `S*.*` and creates a file named `R*.*` that records the SMTP reply codes (error messages) in the `wpgate/`*`gwia`*`/result` directory. After the GWIA completes the communication with the destination host, it moves the `S*.*` message file from the send directory to the result directory along with the corresponding `R*.*` file.
⓬ GWIA for Sender's Domain	GWIA	The GWIA analyzes the files in the result directory, comparing the SMTP reply codes in the `R*.*` file. If the `R*.*` file has a temporary transmission error (meaning it has a 400-level SMTP reply code such as 450 Host Down), the GWIA moves the `S*.*` message file to the `defer` directory. Continue with Stage ⓭ If the `R*.*` file has a fatal error (meaning it has a 500-level SMTP reply code such as 550 Host Unknown), the GWIA deletes the `S*.*` file because it is undeliverable. Skip to Stage ⓮
⓭ GWIA for Sender's Domain	GWIA	Based on the *Intervals to the Retry a Deferred Message* setting on the SMTP/MIME Settings property page of the GWIA object in ConsoleOne, the GWIA requeues the `S*.*` message file back into the send directory for reprocessing.

Stage	Icon	Description
⓮ GWIA for Sender's Domain	GWIA	After an `S*.*` message receives 400-level SMTP reply codes until the *Maximum Number of Hours to Retry a Deferred Message* setting is reached, or if a message receives 500-level SMTP reply codes, the GWIA deletes all related schedule files from the defer directory because the message is undeliverable. The GWIA then creates an Undeliverable status message in the MTA input queue (`wpgate/`*`gwia`*`/wpcsin/`*`0-7`*) in case the TCP/IP link to the MTA is currently closed.
⓯ GWIA for Sender's Domain	GWIA	The GWIA then communicates the Transferred status message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful.
⓰ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the Transferred status message and places it into the MTA "in progress" (`gwinprog`) queue for processing.
⓱ MTA for Sender's Domain	MTA	The MTA for the sender's domain communicates the Transferred status message to the POA for the sender's post office by way of TCP/IP.
⓲ POA for Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (`msg`*`nnn`*`.db`) with the Transferred status information.
⓳ POA for Sender's Post Office	POA	The POA for the sender's post office communicates the Transferred status to the sender's GroupWise client by way of TCP/IP.
⓴ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays the Transferred status because the GWIA was able to send it successfully.

4.3 Mapped/UNC Link Open: Outbound Transfer to the Internet Successful

This message flow diagram shows how outbound messages travel through the GroupWise directory structure to the Internet when there is a mapped/UNC link between the MTA and the GWIA and when the GWIA can communicate successfully with the Internet host to which the message is addressed.

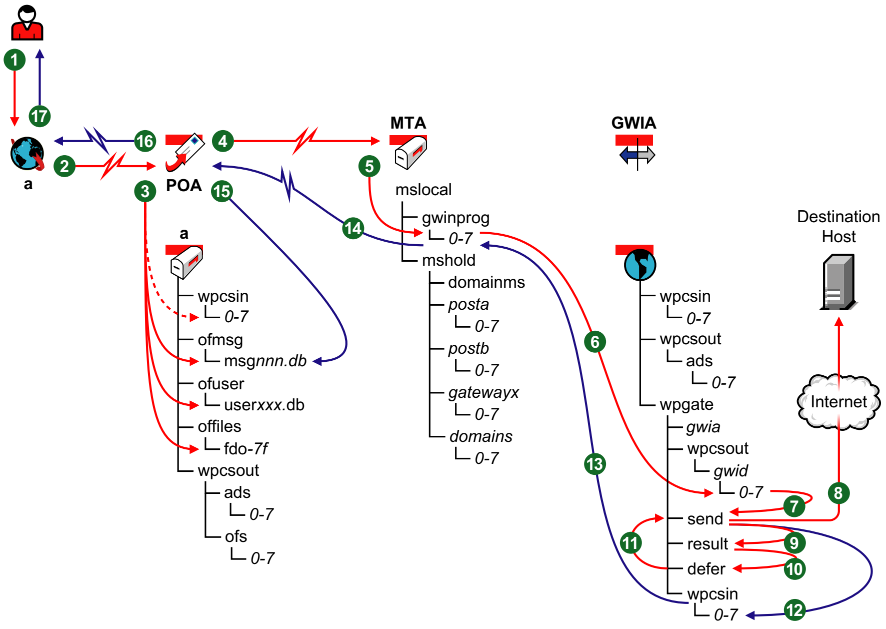


Stage	Icon	Description
1 Sender		The user sends a message to recipients across the Internet by providing their Internet addresses. In this diagram, the access mode setting for the local post office is Client/Server Only.
2 Sender's GroupWise Client		The GroupWise client communicates the message to the POA by way of TCP/IP.
3 POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (msg*nnn*.db) assigned to the sender. ◆ Creates a pointer in the sender's user database (user*xxx*.db) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd0-F6 subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority 0-7 subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.

Stage	Icon	Description
❹ POA for Sender's Post Office	POA	The POA then communicates the message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful.
❺ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the message and places it into the MTA "in progress" (gwinprog) queue.
❻ MTA for Sender's Domain	MTA	The MTA determines that the message must be sent out across the Internet. Because there is a mapped/UNC link between the MTA and the GWIA, the MTA places the message in its output queue in the GWIA's gateway directory (*domain*/wpgate/*gwia*/wpcsout/*gwid*/*0-7*).
❼ GWIA for Sender's Domain	GWIA	The GWIA scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the GWIA object in ConsoleOne. The GWIA picks up the file in binary-encrypted format from the wpcsout/*gwid*/*0-7* directory and converts it. The GWIA encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the GWIA saves it with S as the first character of the file name and places the message file in the *domain*/wpgate/*gwia*/send directory for processing.
❽ GWIA for Sender's Domain	GWIA	While the GWIA is processing the message file in the send directory, it changes the first character of the file name to P. When processing is completed, the GWIA sends the message to the destination host across the Internet.
❾ GWIA for Sender's Domain	GWIA	If the GWIA receives a 250 OK SMTP reply code from the destination Internet host, it places a Transferred status message into the input queue of the MTA for the sender's domain.
❿ MTA for Sender's Domain	MTA	Because of its mapped/UNC link with the GWIA, the MTA regularly scans its input queue in the GWIA's gateway directory based on the *Scan Cycle* setting on the Agent Settings page of the MTA object in ConsoleOne. It picks up the Transferred status messages and transfers them to its "in progress" (gwinprog) directory for processing.
⓫ MTA for Sender's Domain	MTA	The MTA for the sender's domain communicates the Transferred status messages to the POA for the sender's post office by way of TCP/IP.
⓬ POA for Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (msg*nnn*.db) with the Transferred status information.
⓭ POA for Sender's Post Office	POA	The POA for the sender's post office communicates the Transferred status to the sender's GroupWise client by way of TCP/IP.
⓮ Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays the Transferred status because the GWIA was able to sent it successfully.

4.4 Mapped/UNC Link Closed: Outbound Transfer to the Internet Delayed or Unsuccessful

This message flow diagram shows how outbound messages travel through the GroupWise directory structure to the Internet when there is a mapped/UNC link between the MTA and the GWIA and when the GWIA cannot communicate successfully with the Internet host to which the message is addressed.



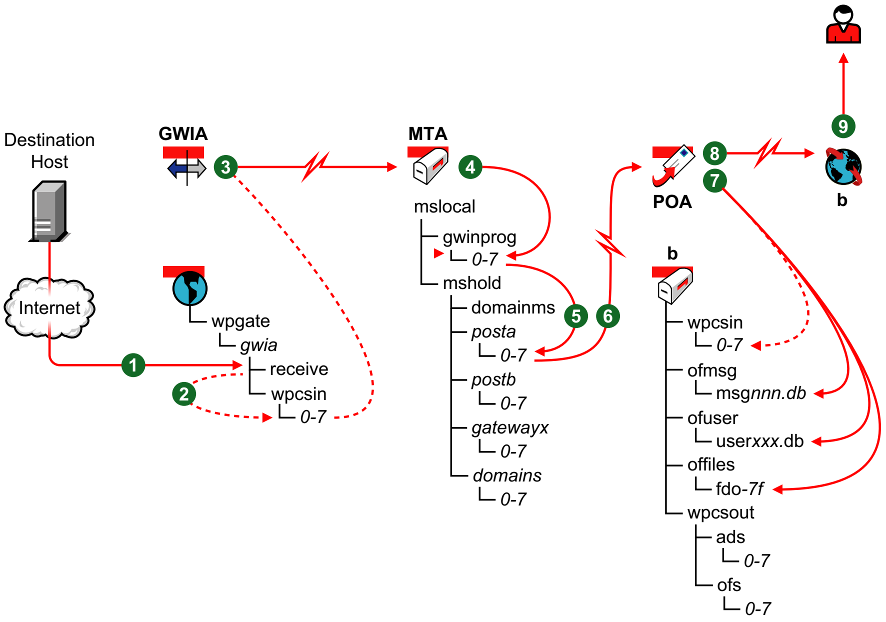
Stage	Icon	Description
1 Sender		The user sends a message to recipients across the Internet by providing their Internet addresses. In this diagram, the access mode setting for the local post office is Client/Server Only.
2 Sender's GroupWise Client		The GroupWise client communicates the message to the POA by way of TCP/IP.
3 POA for Sender's Post Office	POA	The POA receives the message from the GroupWise client and performs the following actions for the sender: ◆ Adds the message to the message database (msgnnn.db) assigned to the sender. ◆ Creates a pointer in the sender's user database (userxxx.db) so the message appears in the sender's mailbox as a sent item. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd0-F6 subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) ◆ Creates a copy of the message in the appropriate priority 0-7 subdirectory of the MTA input queue in the sender's post office, in case the TCP/IP link to the MTA is currently closed.

Stage	Icon	Description
❹ POA for Sender's Post Office	POA	The POA then communicates the message to the MTA for the sender's domain by way of TCP/IP, and deletes the copy in the MTA input queue because the TCP/IP transfer to the MTA was successful.
❺ MTA for Sender's Domain	MTA	The MTA for the sender's domain receives the message and places it into the MTA "in progress" (`gwinprog`) queue.
❻ MTA for Sender's Domain	MTA	The MTA determines that the message must be sent out across the Internet. Because there is a mapped/UNC link between the MTA and the GWIA, the MTA places the message in its output queue in the GWIA's gateway directory (*`domain`*`/wpgate/`*`gwia`*`/wpcsout/`*`gwid`*`/`*`0-7`*).
❼ GWIA for Sender's Domain	GWIA	The GWIA scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the GWIA object in ConsoleOne. The GWIA picks up the file in binary-encrypted format from the `wpcsout/`*`gwid`*`/`*`0-7`* directory and converts it. The GWIA encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the GWIA saves it with S as the first character of the file name and places the message file in the *`domain`*`/wpgate/`*`gwia`*`/send` directory for processing.
❽ GWIA for Sender's Domain	GWIA	While the GWIA is processing the message file in the send directory, it changes the first character of the file name to P. When processing is completed, the GWIA sends the message to the destination host across the Internet.
❾ GWIA for Sender's Domain	GWIA	If the GWIA does not receive a 250 OK SMTP reply code from the destination Internet host, the GWIA renames the `P*.*` message file back to `S*.*` and creates a file named `R*.*` that records the SMTP reply codes (error messages) in the `wpgate/`*`gwia`*`/result` directory. After the GWIA completes the communication with the destination host, it moves the `S*.*` message file from the send directory to the result directory along with the corresponding `R*.*` file.
❿ GWIA for Sender's Domain	GWIA	The GWIA analyzes the files in the result directory, comparing the SMTP reply codes in the `R*.*` file. If the `R*.*` file has a temporary transmission error (meaning it has a 400-level SMTP reply code such as 450 Host Down), the GWIA moves the `S*.*` message file to the `defer` directory. Continue with Stage ⓫ If the `R*.*` file has a fatal error (meaning it has a 500-level SMTP reply code such as 550 Host Unknown), the GWIA deletes the `S*.*` file because it is undeliverable. Skip to Stage ⓬
⓫ GWIA for Sender's Domain	GWIA	Based on the *Intervals to the Retry a Deferred Message* setting on the SMTP/MIME Settings property page of the GWIA object in ConsoleOne, the GWIA requeues the `S*.*` message file back into the send directory for reprocessing.
⓬ GWIA for Sender's Domain	GWIA	After an `S*.*` message receives 400-level SMTP reply codes until the *Maximum Number of Hours to Retry a Deferred Message* setting is reached, or if a message receives 500-level SMTP reply codes, the GWIA deletes all related schedule files from the defer directory because the message is undeliverable. The GWIA then creates an Undeliverable status message for the MTA to pick up and return to the sender.

Stage	Icon	Description
13 MTA for Sender's Domain	MTA	Because of its mapped/UNC link with the GWIA, the MTA scans its input queue in the GWIA's gateway directory based on the *Scan Cycle* setting on the Agent Settings page of the MTA object in ConsoleOne. The MTA picks up the Undeliverable status messages and transfers them to its "in progress" (`gwinprog`) directory for processing.
14 MTA for Sender's Domain	MTA	The MTA for the sender's domain communicates the Transferred status messages to the POA for the sender's post office by way of TCP/IP.
15 POA for Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (`msg`*`nnn`*`.db`) with the Transferred status information.
16 POA for Sender's Post Office	POA	The POA for the sender's post office communicates the Transferred status to the sender's GroupWise client by way of TCP/IP.
17 Sender	👤	When the sender checks the sent items in his or her mailbox in the GroupWise client, the message displays the Transferred status because the GWIA was able to send it successfully.

4.5 TCP/IP Link Open: Inbound Transfer from the Internet Successful

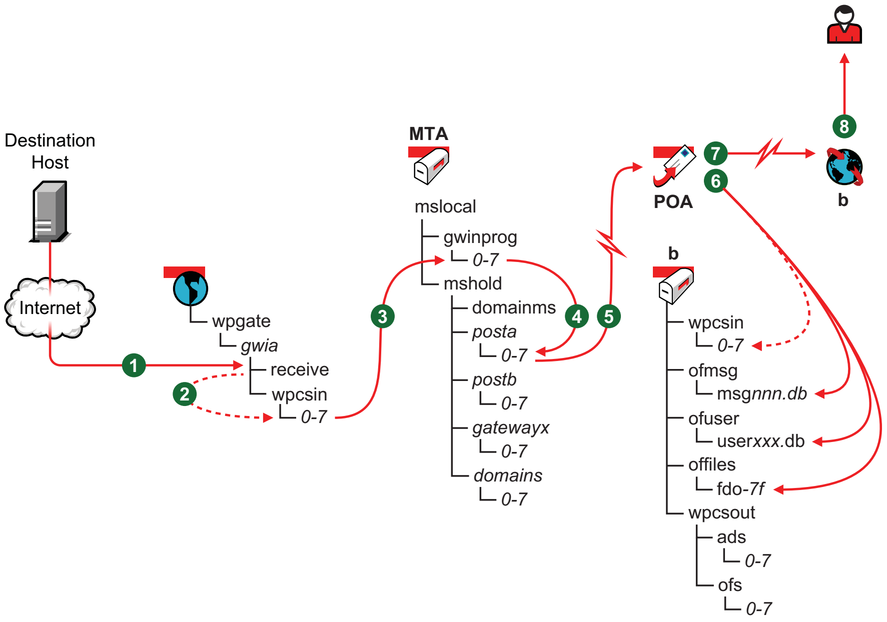
This message flow diagram illustrates how inbound message flow from the Internet through the GroupWise directory structure to the GroupWise recipient. The link between the GWIA and the MTA for the recipient's domain is a TCP/IP link.



Stage	Actor	Action
❶ GWIA for Recipient's Domain	GWIA	An Internet user sends a message to a GroupWise user. The GWIA receives the message from the external Internet host and places the message file in the `wpgate/gwia/receive` directory.
❷ GWIA for Recipient's Domain	GWIA	The GWIA polls the receive directory according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the GWIA object in ConsoleOne. It picks up the message file, converts it to GroupWise format, and places a copy in the `wpgate/gwia/wpcsin/0-7` directory, where *0-7* is one of the priority subdirectories from `0-7`. The GWIA puts messages only in the 4 directory, used for normal priority messages.
❸ GWIA for Recipient's Domain	GWIA	The GWIA then communicates the message to the MTA for the recipient's domain by way of TCP/IP. When the transmission is successful, it deletes the copy in the in the `wpgate/gwia/wpcsin/0-7` directory.
❹ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain receives the message and places it into the MTA "in progress" (`gwinprog`) queue.
❺ MTA for Recipient's Domain	MTA	The MTA determines which post office in the domain the recipient is located in, then moves the message to that post office's hold queue (`mslocal/mshold/postx/0-7`).
❻ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain then communicates the message to the POA in the recipient's post office by way of TCP/IP.
❼ POA for Recipient's Post Office	POA	When it receives the new message, the POA for the recipient's post office performs the following actions: ◆ Adds the message to the message database (`msgnnn.db` file) corresponding to the one assigned to the sender. ◆ Creates a pointer in the recipient's user database (`userxxx.db` file), so the message appears in the recipient's Mailbox and updates the notification information in the user database so the recipient can be notified of the message. ◆ Places attachments larger than 2 KB in one of the `post_office/offiles/fd0-F6` subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and recipient lists larger than 2 KB are also handled as attachments.)
❽ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
❽ Recipient	👤	Each recipient opens the message in the GroupWise client.

4.6 Mapped/UNC Link Open: Inbound Transfer from the Internet Successful

This message flow diagram illustrates how inbound message flow from the Internet through the GroupWise directory structure to the GroupWise recipient. The link between the GWIA and the MTA for the recipient's domain is a mapped/UNC link.



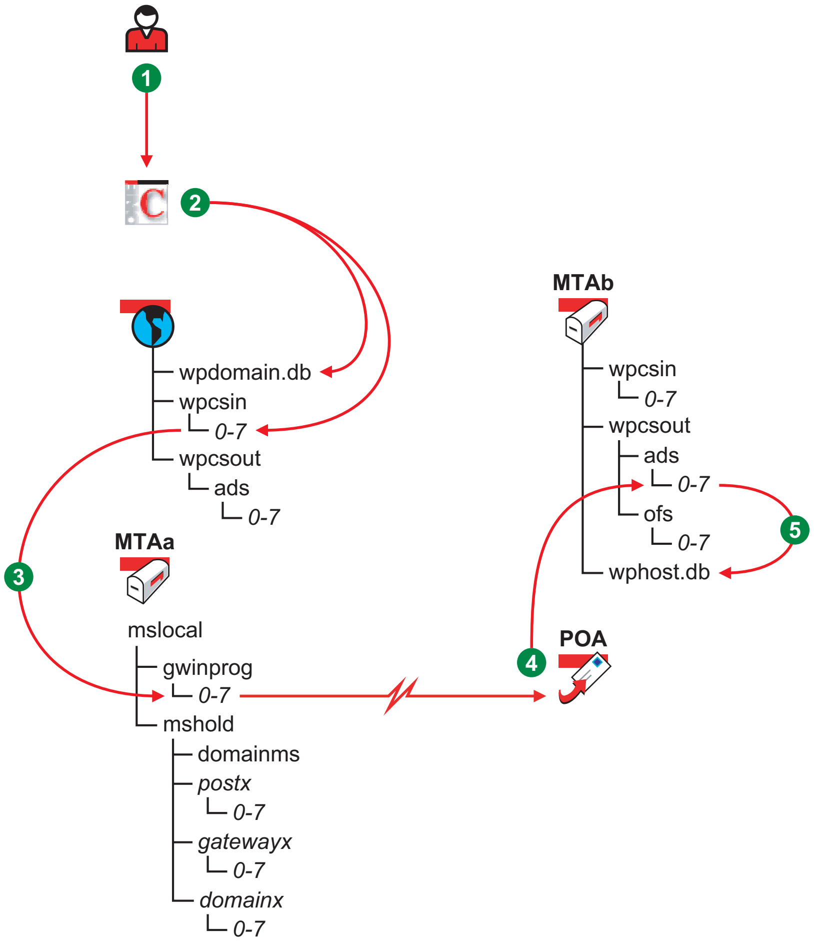
Stage	Actor	Action
❶ GWIA for Recipient's Domain	GWIA	An Internet user sends a message to a GroupWise user. The GWIA receives the message from the external Internet host and places the message file in the `wpgate/gwia/receive` directory.
❷ GWIA for Recipient's Domain	GWIA	The GWIA polls the receive directory according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the GWIA object in ConsoleOne. It picks up the message file, converts it to GroupWise format, and places it in the `wpgate/gwia/wpcsin/0-7` directory, where *0-7* is one of the priority subdirectories from `0-7`. The GWIA puts messages only in the 4 directory, used for normal priority messages.
❸ MTA for Recipient's Domain	MTA	The MTA polls the `domain/wpgate/gwia/wpcsin/fd0-7F` directory based on the *Scan Cycle* setting on the Agent Settings page of the MTA object in ConsoleOne. It picks up the message file and moves it to its "in progress" (`gwinprog`) queue.
❹ MTA for Recipient's Domain	MTA	The MTA determines which post office in the domain the recipient is located in, then moves the message to that post office's hold queue (`mslocal/mshold/postx/0-7`).
❺ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain then communicates the message to the POA in the recipient's post office by way of TCP/IP.

Stage	Actor	Action
❻ POA for Recipient's Post Office	POA	When it receives the new message, the POA for the recipient's post office performs the following actions: ◆ Adds the message to the message database (msg*nnn*.db file) corresponding to the one assigned to the sender. ◆ Creates a pointer in the recipient's user database (user*xxx*.db file), so the message appears in the recipient's Mailbox and updates the notification information in the user database so the recipient can be notified of the message. ◆ Places attachments larger than 2 KB in one of the *post_office*/offiles/fd*0*-*F6* subdirectories and creates pointers from the message to its attachments. (For database efficiency, messages and recipient lists larger than 2 KB are also handled as attachments.)
❼ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
❽ Recipient	👤	Each recipient opens the message in the GroupWise client.

5 Administrative Database Update

ConsoleOne and the agents handle database updates throughout the GroupWise system.

This message flow diagram illustrates how an administrative message, such as a database update request, passes from ConsoleOne to the agents so that databases are updated throughout the GroupWise system.



Stage	Actor	Action
1 GroupWise Administrator	👤	The administrator uses the GroupWise Administrator snap-in in ConsoleOne to add, modify, or delete a GroupWise object in a single-domain, single-post office GroupWise system. An object could be a GroupWise user, resource, distribution list, post office, secondary domain, and so on.

Stage	Actor	Action
❷ ConsoleOne	C	ConsoleOne performs the following actions: ◆ Updates the domain database (wpdomain.db) to reflect the addition, modification, or deletion performed in ConsoleOne. ◆ Creates an administrative message in the priority 2 subdirectory of the domain's MTA input queue (wpcsin) to replicate the update.
❸ MTA for Domain	MTA	The MTA for the domain transfers the administrative message to the MTA "in progress" (gwinprog) queue. From there, the MTA communicates the administrative message to the POA in the post office by way of TCP/IP. The administrative message notifies the POA that a GroupWise object has been added, modified, or deleted. Historical Note: In earlier versions of GroupWise, this function of the POA was handled by a separate agent, the Administration Agent (ADA). The ADA no longer exists in GroupWise.
❹ POA for Post Office	POA	The POA creates a copy of the administrative message in the priority 2 subdirectory of the administrative input queue (wpcsout/ads) in the post office. After the update is made successfully, the copy is deleted.
❺ POA for Post Office	POA	The POA updates the post office database (wphost.db) to reflect the addition, modification, or deletion performed in ConsoleOne and deletes the administrative message from its administrative input queue.

II Directory Structure Diagrams

This part of *Troubleshooting 3: Message Flow and Directory Structure* helps you understand the structure of GroupWise message transfer/storage directories (such as domains and post offices) and software installation directories.

- Chapter 6, "Message Transfer/Storage Directories," on page 49
- Chapter 7, "Agent Installation Directories," on page 87
- Chapter 8, "Web Application Installation Directories on Your Web Server," on page 109
- Chapter 9, "GroupWise Software Distribution Directory," on page 125
- Chapter 10, "GroupWise Client Installation Directory," on page 139

6 Message Transfer/Storage Directories

Message transfer and storage directories are the locations through which messages pass as they travel from user to user through your GroupWise system.

- Section 6.1, "Domain Directory," on page 49
- Section 6.2, "Post Office Directory," on page 56
- Section 6.3, "MTA Local Queue Directory," on page 71
- Section 6.4, "Internet Agent Queue Directory," on page 74
- Section 6.5, "Caching/Remote Mailbox Directory," on page 80

6.1 Domain Directory

domain	Domain directory
mslocal	MTA local working directory
wpcsin	MTA input queue directory
0	Live interactive requests
1	Other interactive requests
2	High priority messages
3	High priority status responses
4	Normal priority messages
5	Normal priority status responses
6	Low priority messages
7	Low priority status responses
wpgate	GroupWise gateway directory
wpcsout	MTA output queue directory
ads	MTA admin thread input queue directory
0	Restart requests
1	Directory synchronization requests
2	Database updates
3	Reserved; not currently used
4	Reserved; not currently used
5	Reserved; not currently used
6	Reserved; not currently used
7	Reserved; not currently used

css	MTA input queue directory for administrative messages
0	MTA restart requests
1	Statistics requests
2	Other non-priority administrative requests
3	Reserved; not currently used
4	Reserved; not currently used
5	Reserved; not currently used
6	Reserved; not currently used
7	Reserved; not currently used
problem	Directory for undeliverable messages
mtaname	Domain name identifier
wpdomain.db	Domain database
wpdomain.dc	Data dictionary for 4.*x* domain databases
wphost.dc	Data dictionary for 4.*x* post office databases
gwdom.dc	Data dictionary for 2012, 8, 7, 6.*x*, and 5.*x* domain databases
gwpo.dc	Data dictionary for 2012, 8, 7, 6.*x*, and 5.*x* post office databases
viewcopy.log	Log file recording view file updates for post offices
uid.run	File recording the authorized user to run the agents (Linux only)
ncpChecked	File confirming that cross-protocol locks are enabled (Linux only)

6.1.1 *domain* directory

Within the GroupWise system, a domain is hierarchically the highest level object. It organizes post offices into a logical grouping for addressing and routing purposes. Each user in the domain has an address that consists of the user's GroupWise user ID, the user's post office name, and the domain name (*user.post_office.domain*). The explicit name is not displayed in the Address Book, but is stored in the domain database (wpdomain.db).

6.1.2 wpcsin directory

The wpcsin subdirectory in the domain is the MTA input queue in each domain. It contains eight priority subdirectories to handle different types of message traffic.

- Incoming user messages are queued by priority for routing to recipients' post offices in the local domain.
- Incoming status messages are queued by priority for routing to senders' post offices in the local domain.
- Outgoing administrative messages are queued for replication to other domains.
- In a routing domain, messages pass through this directory on their way to the next domain.

When a new message arrives, the MTA routes it to the appropriate destination.

For TCP/IP links, the MTA is notified immediately when a message arrives for processing. For mapped and UNC links, the MTA scans its input queue for messages to process. You can control the rate at which the MTA scans its input queues. See "Adjusting MTA Polling of Input Queues in the Domain, Post Offices, and Gateways" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). The Message Transfer Agent (MTA) was originally named the Connection Server (CS). Hence, the directory name `wpcsin` for the MTA input queue. Some naming conventions were originally preserved for backward compatibility.

0 directory

The priority `0` subdirectory of the MTA input queue (`wpcsin`) in the domain is for service requests that demand an immediate response from the MTA. For example:

- ConsoleOne places restart requests and queue reconfiguration requests here for the MTA and gateways.
- MTAs for other domains route Busy Search requests through here when users in other domains check schedules of users in the local domain.

You can increase throughput for the priority `0` subdirectory. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

1 directory

The priority `1` subdirectory of the MTA input queue (`wpcsin`) in the domain is for service requests of the next highest priority. For example:

- ConsoleOne places directory synchronization requests here for the MTA admin thread.
- ConsoleOne places statistics requests here for the MTA to relay to the message logging module for processing.
- MTAs for other domains route remote GroupWise client requests through here when remote GroupWise users do not connect to the post office where their master mailboxes are located.

You can increase throughput for the priority `1` subdirectory. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

2 directory

The priority `2` subdirectory of the MTA input queue (`wpcsin`) in the domain is for high priority messages. For example:

- MTAs for other domains place incoming high priority user messages here. The local MTA then routes the messages to recipients' post offices.
- MTAs for other domains place incoming administrative messages here to replicate database updates in the local domain.
- The MTA admin thread places outgoing administrative messages here to replicate database updates to other domains.

You can increase throughput for the priority `2` and `3` subdirectories. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

3 directory

The priority `3` subdirectory of the MTA input queue (`wpcsin`) in the domain is for high priority status messages routed back to senders in local post offices.

For example, MTAs for other domains place status responses to high priority user messages here. The local MTA then routes the status messages to senders' post offices, so senders' mailboxes can be updated with current message status.

You can increase throughput for the priority `2` and `3` subdirectories. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

4 directory

The priority `4` subdirectory of the MTA input queue (`wpcsin`) in the domain is for normal priority user messages routed to recipients in local post offices.

For example, MTAs for other domains place normal priority user messages here. The local MTA then routes the messages to recipients' post offices. Most messages in your GroupWise system pass through the priority 4 subdirectory.

You can increase throughput for the priority `4` subdirectory. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

5 directory

The priority `5` subdirectory of the MTA input queue (`wpcsin`) in the domain is for normal priority status messages routed back to senders in local post offices.

For example, MTAs for other domains place status responses to normal priority user messages here. The local MTA then routes the status messages to senders' post offices, so senders' mailboxes can be updated with current message status.

6 directory

The priority `6` subdirectory of the MTA input queue (`wpcsin`) in the domain is for low priority user messages routed to recipients in local post offices.

For example, MTAs for other domains place low priority user messages here. The local MTA then routes the messages to recipients' post offices.

7 directory

The priority `7` subdirectory of the MTA input queue (`wpcsin`) in the domain is for low priority status messages routed back to senders in local post offices.

For example, MTAs for other domains place status responses to low priority user messages here. The local MTA then routes the status messages to senders' post offices, so senders' mailboxes can be updated with current message status.

6.1.3 wpgate directory

The wpgate subdirectory in the domain contains a subdirectory for use by the GWIA, which was originally a gateway.

All other GroupWise gateways are legacy products and are not support in GroupWise 2012.

6.1.4 wpcsout directory

The wpcsout subdirectory in the domain is the MTA output queue in each domain. It contains subdirectories that function as input queues for the processes to which the MTA delivers messages.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). The Message Transfer Agent (MTA) was originally named the Connection Server (CS). Hence, the directory name wpcsout for the MTA output queue. Some naming conventions were originally preserved for backward compatibility.

ads directory

The ads subdirectory of the MTA output queue (wpcsout) in the domain is the input queue for the MTA admin thread in each domain. It contains priority subdirectories where incoming administrative messages are queued for processing. When a new administrative message arrives, the MTA admin thread performs the requested action.

Historical Note: The MTA admin thread was previously part of a separate agent, the Administration Agent (ADA), which was originally named the Administration Server (ADS). Hence, the directory name ads. Some naming conventions were originally preserved for backward compatibility.

0 directory

The priority 0 subdirectory of the MTA admin thread input queue (wpcsout\ads) in the domain is for service requests that demand an immediate response from the MTA admin thread.

For example, when you create or delete a post office in ConsoleOne, a restart request is placed here. The domain MTA admin thread processes the request and then restarts.

1 directory

The priority 1 subdirectory of the MTA admin thread input queue (wpcsout\ads) in the domain is for service requests of the next highest priority.

2 directory

The priority 2 subdirectory of the MTA admin thread input queue (wpcsout\ads) in the domain is for high priority administrative messages. For example:

- The MTA places administrative messages from other domains here. The administrative messages might instruct the MTA admin thread to add, modify, or delete users, post offices, or other objects in the domain. The MTA admin thread then processes the messages and makes the specified updates.
- When you use the Synchronize utility in ConsoleOne, a synchronization request is placed here. The MTA admin thread then resends the specified administrative messages to produce the required database updates.

css directory

The css subdirectory of the MTA output queue (wpcsout) in the domain is processed by a specialized MTA thread that responds to requests regarding its own configuration. It contains the eight standard priority subdirectories.

Historical Note: In an earlier version of GroupWise, the Message Transfer Agent (MTA) was called the Connection Server (CS) and this specialized subprocess was called the Connection Server Server (css). Some naming conventions were originally preserved for backward compatibility.

0 directory

The priority 0 subdirectory of the CSS input queue (wpcsout\css) in the domain is for service requests that demand an immediate response from the MTA.

For example, when you restart the MTA at the MTA agent console or in ConsoleOne, a restart request is placed here. The MTA processes the request and restarts.

1 directory

The priority 1 subdirectory of the CSS input queue (wpcsout\css) in the domain is for service requests of the next highest priority.

For example, each time the statistics are updated on the MTA agent console, a statistics request is placed here. The MTA then gathers the statistics and displays them on the MTA agent console.

2 directory

The priority 2 subdirectory of the css input queue (wpcsout\css) in the domain is for non-priority requests.

problem directory

The problem subdirectory of the MTA output queue (wpcsout) in the domain is where the MTA places message files that cannot be delivered because they are damaged in some way. Message files in the problem directory must be handled by the GroupWise administrator. See "Message Is Dropped in the problem Directory in the Domain" in *GroupWise 2012 Troubleshooting 2: Solutions to Common Problems*.

6.1.5 mtaname file

The mtaname file in the domain provides the domain name associated with the domain directory structure. This can help you locate the domain information for the directory structure in ConsoleOne. It can also help you check links between MTAs.

6.1.6 wpdomain.db file

The wpdomain.db file in the domain is the domain database. It contains all administrative information for the domain.

In the primary domain, the wpdomain.db file contains all administrative information for your entire GroupWise system (all its domains, post offices, users, and so on). Because the wpdomain.db file in the primary domain is so crucial, you should back it up regularly and keep it secure. (You can recreate your entire GroupWise system from the primary domain wpdomain.db file; however, if the primary domain wpdomain.db file becomes unusable, you can no longer make administrative updates to your GroupWise system.)

In a secondary domain, the wpdomain.db file contains administrative information about that secondary domain only.

In GroupWise 2012, 8, 7, 6.*x*, and 5.*x* domains, the data dictionary for the wpdomain.db file is the gwdom.dc file. In GroupWise 4.*x* domains, the data dictionary is the wpdomain.dc file. As a result, wpdomain.db files have different structures (schemas) depending on whether they were created for 2012, 8, 7, 6.*x*, and 5.*x* or 4.*x* domains.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Hence, the wp in wpdomain.db. Some naming conventions were originally preserved for backward compatibility.

6.1.7 wpdomain.dc file

The wpdomain.dc file in the domain is the data dictionary for rebuilding GroupWise 4.*x* domain databases (wpdomain.db files) in secondary domains.

If the wpdomain.dc file is missing from the primary domain, you cannot rebuild GroupWise 4.*x* secondary domains. The original wpdomain.dc file is located in the domain subdirectory of the software distribution directory or in the GroupWise software image.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Hence, the wp in wpdomain.dc. Some naming conventions were originally preserved for backward compatibility.

6.1.8 wphost.dc file

The wphost.dc file in the domain is the data dictionary for rebuilding GroupWise 4.*x* post office databases (wphost.db files).

If the wphost.dc file is missing from a domain, you cannot rebuild GroupWise 4.*x* post offices in that domain. The original wphost.dc file is located in the domain directory of the software distribution directory or in the GroupWise software image.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Post offices were originally called hosts. Hence, the name wphost.dc. Some naming conventions were originally preserved for backward compatibility.

6.1.9 gwdom.dc file

The gwdom.dc file in the domain is the data dictionary for creating and rebuilding GroupWise 2012, 8, 7, 6.*x*, and 5.*x* domain databases (wpdomain.db files) in secondary domains.

If the gwdom.dc file is missing from the primary domain, you cannot create or rebuild GroupWise 2012, 8, 7, 6.*x*, and5.*x* secondary domains. The original gwdom.dc file is located in the domain directory of the software distribution directory or in the GroupWise software image.

6.1.10 gwpo.dc file

The gwpo.dc file in the domain is the data dictionary for creating and rebuilding GroupWise 2012 8, 7, 6.*x*, and 5.*x* post office databases (wphost.db files).

If the gwpo.dc file is missing from a domain, you cannot create or rebuild GroupWise 2012, 8, 7, 6.*x*, and 5.*x* post offices in that domain. The original gwpo.dc file is located in the domain directory of the software distribution directory or in the GroupWise software image.

6.1.11 viewcopy.log file

The viewcopy.log file in the domain is created by the GroupWise Installation program if you update the Windows client software and the Installation program is unable to copy the view files to any post offices in the domain. You can manually update the view files later, as described in "Refreshing the Client View Files in the Post Office" in "Post Offices" in the *GroupWise 2012 Administration Guide*.

6.1.12 uid.run file (Linux only)

The uid.run file in the domain records the non-root user that is authorized to run the MTA for the domain. See "Running the Linux GroupWise Agents as a Non-root User" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

6.1.13 ncpChecked file (Linux only)

The ncpChecked file in the domain shows that cross-protocol locks are enabled. See "Configuring the OES Linux Server for NCP Access from Windows" in "System" in the *GroupWise 2012 Administration Guide*.

6.2 Post Office Directory

post_office	Post office directory
wpcsin	MTA input queue directory
0	Live interactive requests
1	Other interactive requests
2	High priority messages
3	High priority status responses
4	Normal priority messages
5	Normal priority status responses
6	Low priority messages
7	Low priority status responses
problem	Directory for undeliverable messages
gwdms	Document Management Services directory
dmsh.db	Shared Document Management Services database

📁 `lib0001-`*`FF`*	Library directories
📁 `dm`*`xxnn01-FF`*`.db`	Document databases
📁 `index`	QuickFinder index for library
📁 `archive`	Archive directory for library
📁 `docs`	Large document directory for library
📁 `fd`*`00-FF`*	Subdirectories for documents
📁 `ofmsg`	Message database directory
📄 `msg`*`nnn`*`.db`	As many as 255 message databases
📄 `ngwdfr.db`	Deferred message database
📁 `guardbak`	Backup guardian database
📁 `ofuser`	User database directory
📄 `user`*`xxx`*`.db`	User databases (one per user)
📄 `pu`*`xxxxx`*`.db`	Databases for shared folders
📁 `index`	QuickFinder index for messages
📁 `offiles`	Attachment store directory
📁 `fd`*`0-f6`*	Subdirectories for attachments
📁 `ofviews`	GroupWise client view files
📁 `ofwork`	GroupWise working directory
📁 `ofdirect`	Remote direct connection directory
📁 `oftemp`	GroupWise temporary files
📁 `gwdca`	Document Converter Agent (DCA) temporary files
📁 `in`	DCA input queue
📁 `out`	DCA output queue
📁 `problem`	DCA quarantine
📁 `log`	DCA log files (Windows only)
📁 `wpcsout`	MTA output queue directory
📁 `ofs`	POA input queue directory
📁 `0`	Live interactive requests
📁 `1`	Other interactive requests
📁 `2`	High priority messages
📁 `3`	High priority status responses
📁 `4`	Normal priority messages
📁 `5`	Normal priority status responses
📁 `6`	Low priority messages
📁 `7`	Low priority status responses
📁 `defer`	Directory to temporarily store deferred messages
📄 *`mmdd`*`poa.`*`nnn`*	POA log files (WIndows only)
📄 `wprof50.db`	Downloadable system Address Book

ads	POA admin thread input queue directory
0	Restart requests
1	Directory synchronization requests
2	Database updates
3	Reserved; not currently used
4	Reserved; not currently used
5	Reserved; not currently used
6	Reserved; not currently used
7	Reserved; not currently used
chk	GWCheck working directory
0-3	GWCheck priority subdirectories
defer	GWCheck subdirectory for deferred database maintenance requests
problem	Directory for undeliverable messages
wphost.db	Post office database
gwpo.dc	Data dictionary for GroupWise 2012, 8, 7, 6.*x*, and 5.*x* post office databases
ngwguard.db	Guardian database
ngwguard.dc	Data dictionary for databases
ngwguard.fbk	Guardian database backup
ngwguard.rfl	Guardian database roll forward log
ngwcheck.db	GWCheck control database
*mmdd*chk.log	GWCheck log files
viewcopy.log	Log file recording view file updates for post offices
uid.run	File recording the authorized user to run the agents (Linux only)
ncpChecked	File confirming that cross-protocol locks are enabled (Linux only)

6.2.1 *post_office* directory

Conceptually, a post office contains mailboxes for a set of network users. The users on the post office send and receive messages through their mailboxes.

Physically, a post office is a directory structure on a network file server. The directory structure contains subdirectories and databases that store messages and the information used to distribute the messages.

6.2.2 wpcsin directory

The `wpcsin` subdirectory in the post office is the MTA input queue in each post office. It contains eight priority subdirectories to handle different types of message traffic.

- Outgoing user messages are queued by priority for routing to recipients in other post offices.
- Outgoing status messages are queued by priority for routing back to senders’ post offices.

- Outgoing Busy Search requests are queued for routing to other post offices so users' schedules can be checked.
- Remote GroupWise client requests are queued for routing to remote GroupWise users' master mailboxes.

When a new message arrives, the MTA routes it to the appropriate destination.

For mapped and UNC links, the MTA scans its input queue for messages to process. You can control the rate at which the MTA scans its input queues. See "Adjusting MTA Polling of Input Queues in the Domain, Post Offices, and Gateways" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

For TCP/IP links, the POA passes messages to the MTA via TCP/IP. A copy is kept in the MTA input queue until the POA has successfully transferred the message.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). The Message Transfer Agent (MTA) was originally named the Connection Server (CS). Hence, the directory name `wpcsin` for the MTA input queue. Some naming conventions were originally preserved for backward compatibility.

0 directory

The priority `0` subdirectory of the MTA input queue (`wpcsin`) in the post office is for service requests that demand an immediate response from the MTA.

For example, the GroupWise client places Busy Search requests here. The MTA then routes the requests to the appropriate post offices, so users' schedules can be checked.

For mapped and UNC links, you can increase throughput for the priority `0` subdirectory. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

For TCP/IP links, the `0` subdirectory is used only if the POA is unable to immediately transfer the request to the MTA by way of the TCP/IP link.

1 directory

The priority `1` subdirectory of the MTA input queue (`wpcsin`) in the post office is for service requests of the next highest priority. For example, GroupWise Remote with a direct connection places requests here for routing to remote GroupWise users' master mailboxes.

For mapped and UNC links, you can increase throughput for the priority `1` subdirectory. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

For TCP/IP links, the `1` subdirectory is used only if the POA is unable to immediately transfer the service request to the MTA by way of the TCP/IP link.

2 directory

The priority `2` subdirectory of the MTA input queue (`wpcsin`) in the post office is for high priority user messages routed to recipients in other post offices, domains, or systems.

For example, the GroupWise client places high priority user messages here. The MTA then routes the messages to the appropriate destinations.

For mapped and UNC links, you can increase throughput for the priority 2 and 3 subdirectories. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

For TCP/IP links, the 2 subdirectory is used only if the POA is unable to immediately transfer the high priority user messages to the MTA by way of the TCP/IP link.

3 directory

The priority 3 subdirectory of the MTA input queue (`wpcsin`) in the post office is for high priority status messages routed back to senders in other post offices, domains, or systems.

For example, the GroupWise client and local POA place status responses to high priority user messages here. The MTA then routes the status messages to the appropriate post offices, so senders' mailboxes can be updated with current message status.

For mapped and UNC links, you can increase throughput for the priority 2 and 3 subdirectories. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

For TCP/IP links, the 3 subdirectory is used only if the POA is unable to immediately transfer the high priority status responses to the MTA by way of the TCP/IP link.

4 directory

The priority 4 subdirectory of the MTA input queue (`wpcsin`) in the post office is for normal priority user messages routed to recipients in other post offices, domains, or systems.

For example, the GroupWise client places normal priority user messages here. The MTA then routes the messages to the appropriate destinations. Most messages in your GroupWise system pass through the priority 4 subdirectory.

For mapped and UNC links, you can increase throughput for the priority 4 subdirectory. See "Adjusting the Number of MTA Scanner Threads for the Domain and Post Offices" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

For TCP/IP links, the 4 subdirectory is used only if the POA is unable to immediately transfer the normal priority user messages to the MTA by way of the TCP/IP link.

5 directory

The priority 5 subdirectory of the MTA input queue (`wpcsin`) in the post office is for normal priority status messages routed back to senders in other post offices, domains, or systems.

For example, the GroupWise client and local POA place status responses to normal priority user messages here. The MTA then routes the status messages to the appropriate post offices, so senders' mailboxes can be updated with current message status.

For TCP/IP links, the 5 subdirectory is used only if the POA is unable to immediately transfer the normal priority status responses to the MTA by way of the TCP/IP link.

6 directory

The priority `6` subdirectory of the MTA input queue (`wpcsin`) in the post office is for low priority user messages routed to recipients in other post offices, domains, or systems.

For example, the GroupWise client places low priority user messages here. The MTA then routes the messages to the appropriate destinations.

For TCP/IP links, the `6` subdirectory is used only if the POA is unable to immediately transfer the low priority user messages to the MTA by way of the TCP/IP link.

7 directory

The priority `7` subdirectory of the MTA input queue (`wpcsin`) in the post office is for low priority status messages routed back to senders in other post offices, domains, or systems.

For example, the GroupWise client and local POA place status responses to low priority user messages here. The MTA then routes the status messages to the appropriate post offices, so senders' mailboxes can be updated with current message status.

For TCP/IP links, the `7` subdirectory is used only if the POA is unable to immediately transfer the low priority status responses to the MTA by way of the TCP/IP link.

problem directory

The `problem` subdirectory of the MTA input queue (`wpcsin`) in the post office is a holding area for damaged message files. Problem files are marked with an extension indicating which GroupWise agent placed each file in the problem directory.

6.2.3 gwdms directory

The `gwdms` subdirectory in the post office is the Document Management Services (DMS) directory in each post office. It contains the document libraries associated with the post office.

dmsh.db file

The `dmsh.db` file in the document management subdirectory (`gwdms`) in the post office is a database shared by all libraries in the post office. It contains a list of all available libraries and lookup tables for each library.

lib00*01-FF* directories

The `lib00`*`01-FF`* subdirectories in the `gwdms` subdirectory in the post office contain the libraries for the post office, with one library per directory. You can create a maximum of 256 libraries in a post office.

dm*xxnn01-FF*.db files

The `dm`*`xxnn01-FF`*`.db` files in the library subdirectories (`lib00`*`01-ff`*) in the post office are databases for library and document information.

The *nn* in the file names represents the partition number, which is generated by a hashing algorithm to guarantee uniqueness.

The *01-ff* in the file names represents the library number, matching the number on the library directory in which the database is found.

dmsd*nn01-ff*.db file The dmsd*nn01-ff*.db file in each library holds system data for the library, such as library configuration information.

dmdd*nn01-ff*.db file The dmdd*nn01-ff*.db file in each library holds document data for the library. Document data is the document property information for documents in the library.

dmdl*nn01-ff*.db file The dmdl*nn01-ff*.db file in each library holds document logging data for the library. Document logging data records all activities performed on documents in the library.

index directory

The index subdirectories in the library subdirectories (lib00*01-ff*) in the post office contain the QuickFinder index for the documents contained in the library.

archive directory

The archive subdirectories in the library subdirectories (lib00*01-FF*) in the post office contain an array of subdirectories for holding archived documents. The subdirectories are numbered sequentially. When the first archive subdirectory reaches its maximum allowable size, archived documents are stored in the next sequential directory, and so on.

docs directory

The docs subdirectories in the library subdirectories (lib00*01-FF*) in the post office contain an array of subdirectories for storing documents.

FD*0-FF* directories The FD*0-FF* subdirectories in the docs subdirectory in the post office store documents that are equal to or greater than 2 KB in size. The *0-FF* variable represents hexadecimal number 0 through FF, so the subdirectories are named FD0 through FDFF. The document databases (dm*xxnn01-FF*.db files) contain pointers to documents stored in the subdirectories of the docs directory.

6.2.4 ofmsg directory

The ofmsg subdirectory in the post office contains as many as 255 databases where messages are stored. It serves as centralized storage for all users in the post office. A message must be stored only once to be delivered to any number of users in the same post office.

Historical Note: An earlier version of GroupWise, designed by WordPerfect Corporation (WPCorp), was named WP Office. Hence, the of in ofmsg. Some naming conventions were originally preserved for backward compatibility.

msg*nnn*.db file

The msg*nnn*.db files in the ofmsg subdirectory in the post office are the message databases where users' messages smaller than 2 KB are stored. To increase database efficiency, messages, attachments, and recipient lists equal to or greater than 2 KB are stored outside the msg*nn*.db files in an array of subdirectories in the offiles directory. After the 2 KB limit is reached, only pointers are stored in the message databases.

The *nnn* variable in the database names is a three-digit number from 0 to 254. A hashing algorithm takes each user's GroupWise file ID (FID) to derive which database the user's outgoing mail is assigned to. The contents of the messages databases are encrypted so the text of message can only be read through GroupWise.

Multiple users are assigned to the same message database. You can use GWCheck to determine which database a specific user has been assigned to. See "GroupWise Check" in "Stand-Alone Database Maintenance Programs" in the *GroupWise 2012 Administration Guide*.

The maximum size for a message database is 4 GB.

ngwdfr.db file

The ngwdfr.db file in the ofmsg subdirectory in the post office holds deferred messages that users have specified for delivery at a later time. When users delay delivery on messages, the messages are transferred to the receiving post office and held in the ngwdfr.db file until the delay expires.

Historical Note: Earlier versions of GroupWise handled deferred messages through the ofpend directory in the post office.

guardbak directory

The guardbak subdirectory in the ofmsg subdirectory in the post office holds a backup copy of the ngwguard.fbk file.

6.2.5 ofuser directory

The ofuser subdirectory in the post office contains a separate database (mailbox) for each GroupWise user.

Historical Note: An earlier version of GroupWise, designed by WordPerfect Corporation (WPCorp), was named WP Office. Hence, the of in ofuser. Some naming conventions were originally preserved for backward compatibility.

user*xxx*.db file

The user*xxx*.db files in the ofuser subdirectory in the post office are user databases where the contents of users' mailboxes are stored, as displayed in the GroupWise client In addition, each user database contains:

- Some personal GroupWise client program settings
- Personal appointments
- Personal groups
- Personal notes
- Rules

Personal client settings that remain the same regardless of what workstation a user logs in to are stored in the user database. Personal client settings that are customized for a particular workstation are stored in the Windows registry.

The *xxx* variable in the database names is each user's GroupWise file ID (FID).

The maximum size for a user database is 4 GB.

pu*xxxxx*.db file

The `pu`*`xxxxx`*`.db` files in the `ofuser` subdirectory in the post office are databases for replicated items such as shared folders. These databases prevent conflicts between user names of shared items from users in other post offices and user names in the local post office.

index directory

The `index` subdirectory in the `ofuser` subdirectory in the post office contains the QuickFinder index for users' messages stored in the post office.

6.2.6 offiles directory

The `offiles` subdirectory in the post office contains subdirectories for messages, attachments, and recipient lists that are equal to or greater than 2 KB in size. These larger messages, attachments, and recipient lists are stored outside the actual message databases in the `ofmsg` directory to increase database efficiency.

Historical Note: An earlier version of GroupWise, designed by WordPerfect Corporation (WPCorp), was named WP Office. Hence, the `of` in `offiles`. Some naming conventions were originally preserved for backward compatibility.

fd*0-f6* directories

The `fd`*`0-f6`* subdirectories in the `offiles` subdirectory in the post office store messages, attachments, and recipient lists that are equal to or greater than 2 KB in size. The *nn* variable represents hexadecimal number 0 through f6, so the subdirectories are named `fd0` through `fdf6`. The message databases (`msg`*`nnn`*`.db` files) contain pointers to messages, attachments, and recipient lists stored in the subdirectories of offiles.

6.2.7 ofviews directory

The `ofviews` subdirectory in the post office contains subdirectories for GroupWise client platforms. Within the platform-specific subdirectories (for example, `win`) are view (`*.vew`) files that create the various views displayed in the GroupWise client.

The `gwviewxx.ini` and `ofviewxx.ini` files configure the standard views on the menus where users select views. The `gwviewxx.ini` file configures GroupWise 2012, 8, 7, 6.*x*, and 5.5 standard views. The `ofviewxx.ini` file configures standard views from earlier versions of GroupWise.

Historical Note: An earlier version of GroupWise, designed by WordPerfect Corporation (WPCorp), was named WP Office. Hence, the `of` in `ofviews`. Some naming conventions were originally preserved for backward compatibility.

6.2.8 ofwork directory

The ofwork subdirectory in the post office is a working directory for requests from the GroupWise client in Remote mode.

Historical Note: An earlier version of GroupWise, designed by WordPerfect Corporation (WPCorp), was named WP Office. Hence, the of in ofwork. Some naming conventions were originally preserved for backward compatibility.

6.2.9 ofdirect directory

The ofdirect subdirectory in the working directory (ofwork) in the post office is used by the GroupWise client in Remote mode for direct connections when the network is available.

Historical Note: An earlier version of GroupWise, designed by WordPerfect Corporation (WPCorp), was named WP Office. Hence, the of in ofdirect. Some naming conventions were originally preserved for backward compatibility.

6.2.10 oftemp directory

The oftemp subdirectory in the post office holds various temporary files such as the MIME files created during access by IMAP email clients.

gwdca directory

The gwdca subdirectory in the post office holds subdirectories used by the Document Converter Agent (DCA) as it converts documents to HTML for indexing by the QuickFinder thread. For more information, see "Configuring the Document Converter Agent (DCA)" in "Post Office Agent" in the *GroupWise 2012 Administration Guide*.

in directory

The POA decrypts attachments and delivers them into the in directory for processing by the DCA.

out directory

The DCA converts documents placed in the in directory into HTML, then places them in the out directory for indexing by the POA. After the POA indexes each HTML file, it deletes the HTML version.

problem directory

If the DCA cannot convert a file, and if *Quarantine Files That Fail during Conversion* is selected on the POA object in ConsoleOne, the DCA places the document in the problem subdirectory. Documents in the problem directory are not encrypted.

log directory (Windows only)

When it starts, the DCA writes its current configuration settings into its log file in the log directory. If a document fails conversion, error information is written to the log file.

On Linux, DCA log files are stored in /var/log/novell/groupwise/gwdca.

6.2.11 wpcsout directory

The wpcsout subdirectory in the post office is the MTA output queue in each post office. It contains subdirectories which function as input queues for the other agents to which the MTA delivers messages.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). The Message Transfer Agent (MTA) was originally named the Connection Server (CS). Hence, the directory name wpcsout for the MTA output queue. Some naming conventions were originally preserved for backward compatibility.

ofs directory

The ofs subdirectory of the MTA output queue (wpcsout) in the post office is the POA input queue in each post office. It contains eight priority subdirectories to handle different types of message traffic.

- Incoming user messages are queued by priority for delivery to recipients' mailboxes in the local post office.
- Incoming status messages are queued by priority for delivery to senders' mailboxes in the local post office.
- Incoming Busy Search requests are queued for the POA to check users' schedules in the local post office.

The POA scans these priority subdirectories regularly. When a new message arrives, the POA processes the messages and performs the required actions.

0 directory

The priority 0 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for service requests that demand an immediate response from the POA.

For example, the MTA places Busy Search requests here so the POA can check recipients' schedules and quickly return the schedule information to the sender.

1 directory

The priority 1 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for service requests of the next highest priority.

For example, the MTA places requests from remote GroupWise users for items in their master mailboxes here. The POA then processes the messages and returns the requested items.

2 directory

The priority 2 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for high priority user messages being delivered to recipients in the local post office.

For example, the MTA places high priority user messages here. The POA then updates the message databases and recipients' mailboxes.

3 directory

The priority 3 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for high priority status messages coming back to senders in the local post office.

For example, the MTA places status responses to high priority user messages here. The POA then updates the message databases and senders' mailboxes with current message status.

4 directory

The priority 4 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for normal priority user messages being delivered to recipients in the local post office.

For example, the MTA places normal priority user messages here. The POA then updates the message databases and recipients' mailboxes. Most messages in your GroupWise system pass through the priority 4 subdirectory.

5 directory

The priority 5 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for normal priority status messages coming back to senders in the local post office.

For example, the MTA places status responses to normal priority user messages here. The POA then updates the message databases and senders' mailboxes with current message status.

6 directory

The priority 6 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for low priority user messages being delivered to recipients in the local post office.

For example, the MTA places low priority messages here. The POA then updates the message databases and recipients' mailboxes.

7 directory

The priority 7 subdirectory of the POA input queue (wpcsout\ofs) in the post office is for low priority status messages coming back to senders in the local post office.

For example, the MTA places status responses to low priority user messages here. The POA then updates the message databases and senders' mailboxes with current message status.

defer directory

The defer subdirectory of the POA input queue (wpcsout\ofs) in the post office is used to temporarily store deferred messages when the ngwdfr.db database is locked. This might occur if backup software has locked the ngwdfr.db database. After the ngwdfr.db database is available again, deferred messages are written to the ngwdfr.db database as usual.

*mmdd*poa.*nnn* files (Windows only)

The *mmdd*poa.*nnn* files are POA log files. The POA creates log files to inform you of its processing and any problems it encounters. By default, these log files are created in the wpcsout\ofs directory on Windows. On Linux, they are created in the /var/log/novell/groupwise/*post_office.poa* directory You can change the location if needed. See "Using POA Log Files" in "Post Office Agent" in the *GroupWise 2012 Administration Guide* guide.

The first two digits of the file name represent the month, the next two digits represent the day of the month, and the next three characters indicate what program created the log. The three-digit extension is a sequence number for multiple log files created on the same day. For example, 0518poa.002 is the second POA log file created on May 18.

wprof50.db file

The wprof50.db file in the wpcsout\ofs directory is the downloadable system Address Book for Remote client users. By default, it is automatically re-created once a day to keep it up to date. See "Performing Nightly User Upkeep" in "Post Office Agent" in the *GroupWise 2012 Administration Guide* guide.

ads directory

The ads subdirectory of the MTA output queue (wpcsout) in the post office is the input queue for the POA admin thread in each post office. It contains priority subdirectories where administrative messages are queued for processing.

Historical Note: The POA admin thread was previously part of a separate agent, the Administration Agent (ADA), which was originally named the Administration Server (ADS). Hence, the directory name *ads*. Some naming conventions were originally preserved for backward compatibility.

0 directory

The priority *0* subdirectory of the POA admin thread input queue (wpcsout\ads) in the post office is for service requests that demand an immediate response from the POA admin thread.

1 directory

The priority 1 subdirectory of the POA admin thread input queue (wpcsout\ads) in the post office is for service requests of the next highest priority.

For example, a directory synchronization request that could not be performed when the POA admin thread received it in its domain input queue would be placed here in the post office for later processing.

2 directory

The priority 2 subdirectory of the POA admin thread input queue (wpcsout\ads) in the post office is for high priority administrative messages.

For example, a database update request that could not be performed when the POA admin thread received it in its domain input queue would be placed here in the post office for later processing.

chk directory

The chk subdirectory of the MTA output queue (wpcsout) in the post office is the working directory where the multithreaded GWCheck process keeps temporary files during database maintenance and where it tracks the activities of its various threads. The defer subdirectory is used when the ngwcheck.db database is locked, for example, by a backup program.

problem directory

The problem subdirectory of the MTA output queue (wpcsout) in the post office is a holding area for damaged message files. Problem files are marked with an extension indicating which GroupWise agent placed each file in the problem directory.

You should check this directory periodically for problem files, resolve the problem, then place the files back into the appropriate queue for continued processing. For assistance, see "Message Is Dropped in the problem Directory in the Post Office" in "Strategies for Message Delivery Problems" in the *GroupWise 2012 Troubleshooting 2: Solutions to Common Problems*.

6.2.12 wphost.db file

The wphost.db file in the post office is the post office database. It contains all administrative information for the post office. It also contains the Address Book for the post office.

In GroupWise 2012, 8, 7, 6.*x*, and 5.*x* post offices, the data dictionary for the wphost.db file is the gwpo.dc file. In GroupWise 4.*x* post offices, the data dictionary is the wphost.dc file. As a result, wphost.db files have different structures (schemas) depending on whether they were created for GroupWise 2012, 8, 7, 6.*x*, 5.*x* or 4.*x* post offices.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Post offices were originally called hosts. Hence, the name wphost.db. Some naming conventions were originally preserved for backward compatibility.

6.2.13 gwpo.dc file

The gwpo.dc file in the post office is the data dictionary for creating and rebuilding GroupWise 2012, 8, 7, and 6.*x* post office databases (wphost.db files).

If the gwpo.dc file is missing from a post office and its domain, you cannot create or rebuild GroupWise 2012, 8, 7, and 6.*x* post offices in that domain. The original gwpo.dc file is located in the domain directory of the software distribution directory or in the *GroupWise 2012* downloaded software image.

6.2.14 ngwguard.db file

The ngwguard.db file in the post office is the guardian database. See "Information Stored in the Post Office" in "Post Office Agent" in the *GroupWise 2012 Administration Guide*.

6.2.15 ngwguard.dc file

The ngwguard.dc file in the post office is the data dictionary for building the following databases in the post office:

- ngwguard.db (guardian database)

- dm*xxnn*01-*ff*.db (document management databases)
- msg*nnn*.db (message databases)
- user*xxx*.db (user databases)
- pu*xxxxx*.db (databases for replicated items like shared folders)

6.2.16 ngwguard.fbk file

The ngwguard.fbk file in the post office is a "fall back" copy of the ngwguard.db file. If the ngwguard.db file becomes damaged, the ngwguard.fbk file, along with the ngsguard.rfl file, can be used to rebuild a valid, current ngwguard.db file. The ngwguard.fbk file is so important that an additional copy of it is kept in the ofmsg\guardbak subdirectory in case the copy in the post office directory is inadvertently deleted. See "Guardian Databases" in "Databases" in the *GroupWise 2012 Administration Guide*.

6.2.17 ngwguard.rfl file

The ngwguard.rfl file in the post office is a roll-forward transaction log of every database transaction that has taken place since the last copy of the ngwguard.fbk file was created. See "Guardian Databases" in "Databases" in the *GroupWise 2012 Administration Guide*.

6.2.18 ngwcheck.db

The ngwcheck.db file in the post office is the database that controls GWCheck's multithreaded processing. It contains job and task records that are used to synchronize and summarize GWCheck requests as they progress.

6.2.19 *mmdd*chk.log

The *mmdd*chk.log files in the post office are the log files created by the POA during database maintenance, as described in "Scheduling Database Maintenance" in "Post Office Agent" in the *GroupWise 2012 Administration Guide*.

6.2.20 viewcopy.log file

The viewcopy.log file in the post office is created by the GroupWise Installation program if you update the Windows client software and the Installation program is unable to copy the view files to the post office. You can manually update the view files later, as described in "Refreshing the Client View Files in the Post Office" in "Post Offices" in the *GroupWise 2012 Administration Guide*.

6.2.21 uid.run file (Linux only)

The uid.run file in the post office records the non-root user that is authorized to run the POA for the post office. See "Running the Linux GroupWise Agents as a Non-root User" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

6.2.22 ncpChecked file (Linux only)

The ncpChecked file in the post office shows that cross-protocol locks are enabled. See "Configuring the OES Linux Server for NCP Access from Windows" in "System" in the *GroupWise 2012 Administration Guide*.

6.3 MTA Local Queue Directory

```
mslocal                 MTA local working directory
   mmddxxx.nn           MTA log files on Windows

   msglog               Message logging directory
      mmddmsg.nn        Message logging files

   gwinprog             MTA "in progress" queue directory
      0-7               Priority subdirectories

   mshold               MTA holding directory
      domainms          Processing directory for MTA
         0-7            Priority subdirectories
         mtaname        Location identifier
      postx             Holding directories for post offices
         0-7            Priority subdirectories
         mtaname        Location identifier
      gatewayx          Holding directories for gateways
         0-7            Priority subdirectories
         mtaname        Location identifier
      domainx           Holding directories for other domains
         0-7            Priority subdirectories
         mtaname        Location identifier

   gwvsscan             Working directory for third-party virus scanning programs

   mtaconv              Work area for 5.x to 4.x conversion
```

6.3.1 mslocal directory

The mslocal directory is the MTA local working directory. The --work switch of the MTA specifies the location of the mslocal directory. It must be located on the hard disk of the server where the MTA runs so it is always accessible. Adequate disk space must be available to hold messages going to destinations that are temporarily closed.

Initially, the mslocal directory is created as a subdirectory of the domain directory.

If the Windows MTA runs on a different server from where the domain directory structure is located, you can move the mslocal directory. To move the mslocal directory to the server where the Window MTA is running, stop the MTA, the copy the mslocal directory, along with all of its subdirectories, to the new location. Then restart the MTA and specify the new location using the --work switch.

Historical Note: In earlier versions of GroupWise, the Message Transfer Agent (MTA) was called the Message Server (MS). Hence, the ms in mslocal. Some naming conventions were originally preserved for backward compatibility.

mmddxxx.nnn files

The *mmddxxx*.*nnn* file are MTA log files. The MTA creates log files to inform you of its processing and any problems it encounters. By default, these log files are created in the mslocal directory. You can change the location if needed. See "Using MTA Log Files" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

The first two digits of the file name represent the month; the next two digits represent the day of the month; the next three characters indicate what program created the log. The three-digit extension is a sequence number for multiple log files created on the same day. For example, 0518mta.002 is the second MTA log file created on May 18.

On Linux, the MTA log files are stored in the /var/log/novell/groupwise/*domain*.mta directory

Historical Note: In earlier versions of GroupWise, the Message Transfer Agent (MTA) was called the Message Server (MS). Hence, the ms indicator representing the MTA. Some naming conventions were originally preserved for backward compatibility.

6.3.2 msglog directory

The msglog subdirectory contains message logging files. It is created when you turn on message logging. The MTA receiver threads log messages as they arrive so the MTA worker threads can process messages without having to scan the MTA input queues to look for work.

The resources used for message logging are configurable. See "Optimizing the Routing Queue" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

More detailed message logging by the MTA is also available, but is turned off by default. See "Enabling MTA Message Logging" in "Configuring the MTA" in the *GroupWise 2012 Administration Guide*.

*mmdd*msg.*nnn* files

The *mmdd*msg.*nnn* files in the message logging subdirectory (msglog) in the MTA local directory are used by the MTA to track messages in its "in progress" queue.

The first two digits of the file name represent the month; the next two digits represent the day of the month. The three-digit extension is a sequence number for multiple files created on the same day. For example, 0518msg.002 is the second message logging file created on May 18.

6.3.3 gwinprog directory

The gwinprog subdirectory is the MTA "in progress" queue. It contains eight priority subdirectories parallel to those found in wpcsin. All messages for recipients in the domain pass through gwinprog, no matter whether they arrived by way of TCP/IP or by way of message files deposited into the MTA input queue by a POA or another MTA.

The resources used to process the "in progress" queue are configurable. See "Optimizing the Routing Queue" in "Optimizing the MTA" in the *GroupWise 2012 Administration Guide*.

6.3.4 mshold directory

The mshold subdirectory is a holding queue for messages addressed to domains, post offices, or gateways that are currently closed.

A location might be closed because its server is down or because the MTA is unable to communicate with it for any other reason. When a closed location is again open, the MTA moves messages from the holding queue back into the normal message flow.

Historical Note: In earlier versions of GroupWise, the Message Transfer Agent (MTA) was called the Message Server (MS). Hence, the ms in mshold. Some naming conventions were originally preserved for backward compatibility.

6.3.5 domainms directory

The domainms subdirectory in the holding directory (mshold) is used for internal processing by the MTA. It does not contain any files a GroupWise administrator needs to access.

Historical Note: In earlier versions of GroupWise, the Message Transfer Agent (MTA) was called the Message Server (MS). Hence, the ms in domainms. Some naming conventions were originally preserved for backward compatibility.

6.3.6 *postx* directories

The *postx* subdirectories in the holding directory (mshold) represent post offices in the domain. If a post office is closed, the MTA routes messages for that post office into its holding queue in mshold. When the post office is open, the MTA moves the messages from the holding queue back into the regular message flow. For more information, see "Message Delivery to a Different Post Office" on page 17.

The name of the holding queue for each post office consists of the first three characters of the post office name, followed by four hashed characters to ensure uniqueness.

6.3.7 *gatewayx* directories

The *gatewayx* subdirectories in the holding directory (mshold) represent gateways in the domain. If a gateway is closed, the MTA routes messages for that gateway into its holding queue in mshold. When the gateway is open, the MTA moves the messages from the holding queue back into the regular message flow through the gateway.

The name of the holding queue for each gateway consists of the first three characters of the gateway name, followed by four hashed characters to ensure uniqueness.

6.3.8 *domainx* directories

The *domainx* subdirectories in the holding directory (mshold) represent domains to which the current domain has a direct link. If a domain is closed, the MTA routes messages for that domain into its holding queue in mshold. When the domain is open, the MTA moves the messages from the holding queue back into the regular message flow. For more information, see "Message Delivery to a Different Domain" on page 23.

The name of the holding queue for each domain consists of the first three characters of the domain name, followed by four hashed characters to ensure uniqueness.

6.3.9 *0-7* directories

The priority 0-7 subdirectories in each holding queue in the mshold subdirectory correspond to the priority 0-7 subdirectories located in each domain, post office, or gateway. See the following directory structures for more information about its priority 0-7 subdirectories:

- Section 6.1, "Domain Directory," on page 49
- Section 6.2, "Post Office Directory," on page 56

6.3.10 mtaname files

The mtaname files in the closed location holding queues provide the name associated with the domain, post office, or gateway holding queue. They can help you check links between MTAs in ConsoleOne without going to the MTA agent console to determine the location name. To associate a location name with its holding queue directory from the MTA agent console, click *Configuration Status* > select the location > click *Details*.

6.3.11 gwvsscan directory

The gwvsscan subdirectory is the working directory where third-party virus scanning programs that snap in to the MTA can perform their processing.

6.3.12 mtaconv directory

The mtaconv subdirectory is the working directory where the MTA converts GroupWise 2012, 8, 7, 6.*x*, and 5.*x* messages to 4.*x* format for transfer to a GroupWise 4.*x* system. After the conversion is finished, this directory should be empty.

6.4 Internet Agent Queue Directory

The following directories and files are found under the \domain\wpgate\ structure for the GWIA after the software has been installed and the GWIA has processed messages.

domain\wpgate\gwia	GroupWise Internet Agent home directory
000.prc	GWIA message processing directory
cmd	Not currently used
gwwork	Holding directory for temporary files used during processing
mmddlog.nnn	Log files
acct	Accounting file
set	Settings file for screen colors, log levels, and so on
stat	Statistics file for GWIA operation
proc	Process lock file indicating that the GWIA is running
pulse.tmp	Temporary file to verify GWIA operation
wpcsin	MTA input queue directory
0-7	Message priority subdirectories

wpcsout	MTA output queue
gwixxxx	System-defined directory
0-7	Message priority subdirectories
problem	Hold directory for damaged outbound messages
gwhold	Message hold directory
qfiles	Delayed delivery hold directory
gwprob	Holding directory for damaged inbound messages
gwchars	Directory for character conversion tables
save	Directory for old configuration files from reinstalls or upgrades
gwia.cfg	GWIA configuration file for switches
route.cfg	Route configuration file to customize routing
gwauth.cfg	Host authentication configuration file
mimetype.cfg	MIME encoding configuration file for various file types
exepath.cfg	Configuration file pointing ConsoleOne to the *gwia*.cfg file
frgnames.cfg	Foreign domain name configuration file
xspam.cfg	Anti-spam configuration file
gwac.db	Access control database
gwac.dc	Database dictionary file used to create the access control database
preamble.txt	Message for recipients who lack a MIME-compliant mail reader
preamble.all	Preamble message in various languages
blocked.txt	List of blocked Internet sites
status*xx*.xml	File for customizing status messages

gwia	SMTP service (daemon) home directory
send	Outbound hold directory for converting messages into Internet format
receive	Incoming hold directory for converting messages into GroupWise format
result	Send and result files to confirm transmission
defer	Holding directory for re-queued and deferred messages
work	Schedule files for SMTP service operations on deferred messages
dsnhold	Delivery Status Notification (DSN) hold directory

6.4.1 *domain*\wpgate\gwia directory

The *domain*\wpgate\gwia directory is the GWIA home directory where GWIA configuration files and queue directories are located. The name is established when you install the GWIA. The default is wpgate\gwia in the domain directory. You can change the location using the --home switch in the GWIA configuration file (*gwia*.cfg).

000.prc directory

The GWIA uses the 000.prc directory to process messages.

gwwork directory

The gwwork directory stores temporary files created by the GWIA as it converts and builds messages for transfer across the Internet.

*mmdd*log.*nnn* file

The mmddlog.*nnn* files hold error and status messages about the functioning of the GWIA. The GWIA creates a log file each day with a unique name, where *mm* is the month, *dd* is the day, and *nnn* is a sequential number indicating the sequence of log files in a single day. For more information log files, see "Using GWIA Log Files" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

acct file

The acct file contains information about the messages the GWIA sends each day. It is emailed to the accounts each day at midnight. For more information about the accounting files, see "Tracking Internet Traffic with Accounting Data" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

set file

The set file stores GWIA console settings such as color, log settings, and so on. For more information, see "Using the GWIA Server Console".

stat file

The stat file stores statistics about the GWIA's functioning. For information about the statistics provided by the GWIA, see "Statistics" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

proc file

The proc file is the lock file for the GWIA process. The proc file is opened and locked when the GWIA starts. This prevents multiple GWIAs from being started for the same domain.

pulse.tmp file

The pulse.tmp file is re-created by the GWIA every time it completes a cycle (after an idle loop). If you are not at the GWIA console but need to know if the GWIA is running, you can delete the pulse.tmp file. If the GWIA is running, it re-creates the file.

wpcsin directory

For a mapped/UNC link, the GWIA places inbound messages in one of the wpcsin priority subdirectories (0-7). Most messages go in the 4 directory, although some administrative and status messages might go in other directories. The MTA retrieves the messages and delivers them to the proper destinations.

For a TCP/IP link, the GWIA and the MTA communicate by way of TCP/IP rather than by transferring message files. For a comparison, see Chapter 4, "Message Delivery to and from the Internet," on page 31.

wpcsout directory

For a mapped/UNC link, the `wpcsout` directory is the MTA output queue as well as being the GWIA input queue.

For a TCP/IP link, the GWIA and the MTA communicate by way of TCP/IP rather than by transferring message files. For a comparison, see Chapter 4, "Message Delivery to and from the Internet," on page 31.

gwixxxx directory

The *`gwixxxx`* directory is a system-defined directory, where *gwi* represents the first three letters of the GWIA object name as defined during installation and displayed in ConsoleOne, and *xxxx* is a randomly-generated string. Here, the MTA places outbound messages in the appropriate `0-7` priority subdirectory for the GWIA to retrieve and process.

problem directory

The `problem` directory holds messages that the MTA cannot process.

You should check this directory periodically for problem files, resolve the problem, then place the files back into the appropriate queue for continued processing. For assistance, see "Message Is Dropped in the problem Directory in the Domain" in "Strategies for Message Delivery Problems" in the *GroupWise 2012 Troubleshooting 2: Solutions to Common Problems*.

gwhold directory

The `gwhold` directory holds messages that are scheduled for delayed delivery.

qfiles directory

The `qfiles` directory holds messages that cannot be sent during the current Send/Receive cycle. The messages are queued to this directory until the next cycle.

The delayed delivery messages waiting in the `qfiles` directory remain in encrypted format until the GWIA transfers them to the send directory for processing by the SMTP service.

gwprob directory

The GWIA uses the `gwprob` directory for messages it cannot process. These are usually messages that have been damaged during transmission or that contain incorrectly formed MIME data.

These messages cannot be recovered. You can delete them to conserve disk space.

gwchars directory

This directory contains conversion tables that the GWIA uses to convert message attachments between character sets.

save directory

If you reinstall or upgrade the GWIA, your old configuration files are copied to the `save` directory as a backup. If you reinstall or upgrade repeatedly, the files are overwritten each time.

gwia.cfg file

The *gwia*.cfg file is the GWIA configuration file that contains switches. Some switches are set during installation. You can set others as needed. For more information, see "Using GWIA Startup Switches" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

Linux:	The Linux GWIA uses the *gwia*.cfg file created in /opt/novell/groupwise/agents/share during installation. The *gwia*.cfg file under the domain is just a boilerplate file with no switches set during installation.
Windows:	Only the Windows GWIA actually uses the *gwia*.cfg file under the domain.

route.cfg file

The route.cfg file enables you to customize routing for specific hosts. For more information, see "Using a Route Configuration File" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

gwauth.cfg file

The gwauth.cfg file enables the GWIA to log in to SMTP hosts that require authentication. For more information, see "SMTP Host Authentication" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

mimetype.cfg file

The mimetype.cfg file enables you to customize MIME content-type mappings for various attachment types. For more information, see "Customizing MIME Content-Type Mappings" in "Internet Agent" in the *GroupWise 2012 Administration Guide*

exepath.cfg file

The exepath.cfg file is used by ConsoleOne to locate the *gwia*.cfg file. This enables ConsoleOne to write any configuration setting changes to the *gwia*.cfg file or update Novell eDirectory with any changes from the file. The file must contain the path to the *gwia*.cfg file in the /opt/novell/groupwise/agents/share directory on Linux, or the *domain*\wpgate*gwia* directory on Windows.

frgnames.cfg file

The frgnames.cfg file lets you list more Internet domain names than can fit in the *Foreign ID* field on the Identification page of the GWIA object in ConsoleOne. For more information, see "Configuring How the GWIA Handles Email Addresses" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

xspam.cfg file

The xspam.cfg file lists "X" header fields that your anti-spam service writes to the MIME header, along with the values that flag the message as spam. The GWIA examines the MIME header for any field listed in the xspam.cfg file. When a match occurs, the message is marked for handling by the GroupWise client Junk Mail Handling feature. For more information, see "Customized Spam Identification" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

gwac.db file

The `gwac.db` file is the access control database that stores information about the classes of service you have created. For more information, see "Maintaining the Access Control Database" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

gwac.dc file

The `gwac.dc` file is the data dictionary file from which the `gwac.db` is created.

preamble.txt file

The `preamble.txt` file is an ASCII text file that is automatically included with any MIME multipart message and is displayed when the message recipient lacks a MIME-compliant mail reader. For more information, see "Customizing MIME Preamble Text" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

preamble.all file

The `preamble.all` file contains the preamble text in multiple languages. For more information, see "Customizing MIME Preamble Text" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

blocked.txt file

The `blocked.txt` file contains a list of Internet sites that you have added to the Prevent Messages From list for your default class of service in ConsoleOne. For more information, see "Controlling User Access to the Internet" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

status*xx*.xml file

The `statusxx.xml` file enables you to customize the messages that users receive regarding message delivery status. For more information, see "Customizing Delivery Status Notifications" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

6.4.2 *gwia* directory

The *`gwia`* directory is the SMTP service (daemon) home directory where messages are converted between GroupWise format and Internet format. On Linux, the default location is `wpgate/gwia`, the same as the GWIA home directory. On Windows, the default location is the GWIA installation directory. You can change the location using the --dhome switch in the GWIA configuration file (*gwia*.cfg).

send directory

The GWIA SMTP service places outbound messages in the `send` directory after they have been converted out of GroupWise format into SMTP format. The SMTP service polls the send directory and sends any messages to the destination SMTP host.

receive directory

The GWIA SMTP service places inbound messages in the `receive` directory, converts them into GroupWise format, and then passes them to the MTA by placing them in the wpcsin directory.

result directory

When the GWIA SMTP service processes the message, it builds a file, `r*.*`, in the `result` directory that contains several lines of comments and SMTP reply codes, which might indicate possible errors or confirm correct transmission. After the GWIA SMTP service has completed the transmission with the destination host, it moves another file, `s*.*` from the `send` directory to the `result` directory. The file names for both files are identical, except for the first letter, which is either “s” or “r”. The `s*.*` file is the converted message file. The SMTP service looks at the “s” and “r” files in the `result` directory and compares the conversation. If the `r*.*` file contains the correct (250 OK) SMTP reply codes, the SMTP service deletes the file and sends a transferred status message to the user’s Sent Items folder in the GroupWise client.

defer directory

The `defer` directory holds messages that are deferred and re-queued according to the Retry Schedule. If the GWIA SMTP service receives a temporary error, such as Host Down, it places the message in the `defer` directory for a specified time, then transfers the file to the `send` directory for another attempt at sending to the Internet. For more information, see “Configuring Basic SMTP/MIME Settings” in “Internet Agent” in the *GroupWise 2012 Administration Guide*.

dsnhold directory

The `dsnhold` directory stores header information for inbound messages that request delivery status notifications. For more information, see “Using Extended SMTP (ESMTP) Options” in “Internet Agent” in the *GroupWise 2012 Administration Guide*.

6.5 Caching/Remote Mailbox Directory

A Remote mailbox has the same structure as a Caching mailbox. The same directory structure can be accessed using either Caching mode or Remote mode.

`c:\User\user_name\AppData\Roaming\Novell\GroupWise\gwxxxxxx`	GroupWise Caching mailbox on Windows 7
`c:\User\user_name\AppData\Local\Novell\GroupWise\gwxxxxxx`	GroupWise Caching mailbox on Windows Vista
`c:\Documents and Settings\user_name\Local Settings\Application Data\Novell\GroupWise\gwxxxxxx`	GroupWise Caching mailbox on Windows XP
`rofdata`	Caching mailbox database directory
`msg.db`	Cached message database
`user.db`	Cached user database
`wprof.db`	Cached Address Book
`wprof.dc`	Data dictionary for cached Address Book

ngwguard.db	Guardian database
ngwguard.dc	Data dictionary for guardian database
ngwguard.rfl	Guardian database roll forward log
ngwguard.fbk	Guardian database “fall back” file
pu*xxxxx*.db	Database for shared folders
ngwcheck.db	GroupWise Check database
gwcheck*n*.log	Log file created by the Repair Mailbox feature
gwdms	Document Management Services directory
dmsh.db	Shared DMS database
dm*xxnn01-FF*.db	Document databases
docs	Subdirectory for documents in the Caching mailbox
index	QuickFinder index for documents in the Caching mailbox
index	QuickFinder index for messages in the Caching mailbox
wpcsin	Input queue for the Caching mailbox
0-7	Priority subdirectories
wpcsout\ofs	Output queue for the Caching mailbox
0-7	Priority subdirectories
wpgwsend	Output queue to the Online mailbox
wpgwrecv	Input queue from the Online mailbox
remote*n*.log	Connection log
status*n*.txt	Status Window log

6.5.1 gw*xxxxxx* directory

Your GroupWise Caching/Remote mailbox is a directory structure that functions similarly to a post office. Like a post office, it contains databases and input/output queues. The default location varies by platform.

For the Windows client, the same directory structure is used for a Caching mailbox as for a Remote mailbox. However, a Caching mailbox is a complete copy of your Online mailbox, but you can restrict what gets downloaded into your Remote mailbox.

6.5.2 rofdata directory

The rofdata directory contains the databases accessed by the GroupWise client when running in Caching mode. The databases in rofdata are similar to the databases found in post offices. For comparison, see Section 6.2, “Post Office Directory,” on page 56.

Historical Note: An earlier version of the GroupWise client Remote mode, designed by WordPerfect Corporation (WPCorp), was named WP Office Remote. Hence, the rof in rofdata. Some naming conventions were originally preserved for backward compatibility.

msg.db file

The msg.db file is the cached equivalent of the msg*nnn*.db files in the ofmsg directory in your post office. The msg.db file contains copies of messages from your Online mailbox.

user.db file

The user.db file is the cached equivalent of the userxxx.db files in the ofuser directory in your post office.

wprof.db file

The wprof.db file contains the cached version of the GroupWise Address Book.

Historical Note: An earlier version of the GroupWise client Remote mode, designed by WordPerfect Corporation (WPCorp), was named WP Office Remote. Hence, the wprof in wprof.db. Some naming conventions have been preserved for backward compatibility.

wprof.dc file

The wprof.dc file is the data dictionary for the cached Address Book (wprof.db).

Historical Note: An earlier version of the GroupWise client Remote mode, designed by WordPerfect Corporation (WPCorp), was named WP Office Remote. Hence, the wprof in wprof.dc. Some naming conventions have been preserved for backward compatibility.

ngwguard.db file

The ngwguard.db file is the guardian database for your Caching mailbox. It is parallel in function to the ngwguard.db file in the post office.

ngwguard.dc file

The ngwguard.dc file is the data dictionary for building the databases in the GroupWise Caching mailbox. It is parallel in function to the ngwguard.dc file in the post office.

ngwguard.rfl file

The ngwguard.rfl file is a roll-forward transaction log of every database transaction that has taken place since the last copy of the ngwguard.fbk file was created. It is parallel in function to the ngwguard.rfl file in the post office.

ngwguard.fbk

The ngwguard.fbk file “fall back” copy of the ngwguard.db file. It is parallel in function to the ngwguard.fbk file in the post office.

pu*xxxxx*.db files

The pu*xxxxx*.db files are databases for replicated items such as shared folders. These databases prevent conflicts between user names of shared items from users in other post offices and user names in your own post office. They are parallel to the pu*xxxxx*.db files in the post office.

ngwcheck.db file

The ngwcheck.db file tracks GroupWise Check threads and the databases being checked. In the GroupWise client, GroupWise Check is run using *Tools* > *Repair Mailbox*.

gwcheck*n*.log

The gwcheck*n*.log file records any errors that occurred during mailbox repair. For assistance with GroupWise Check errors, see "GroupWise Check Error Codes" in "Administration Error Messages" in the *GroupWise 2012 Troubleshooting 1: Error Messages*.

gwdms directory

The gwdms directory is the Document Management Services directory. It contains information about the libraries in your GroupWise system. It has the same structure as the gwdms subdirectory in the post office.

dmsh.db file

The dmsh.db file is a database shared by all libraries that contains a list of all available libraries and lookup tables for each library.

dm*xxnn01-FF*.db files

The dm*xxnn01-FF*.db files are databases for library and document information. They are parallel to the dm*xxnn01-FF*.db files in the post office.

docs directory

The docs directory holds cached copies of the documents in your Online mailbox.

index directory

The index directory under the gwdms directory contains the QuickFinder index for the documents in your Caching mailbox.

index directory

The index directory under the rofdata directory contains the QuickFinder index for the messages in your Caching mailbox.

6.5.3 wpcsin directory

The wpcsin subdirectory is the input queue for the connection that transfers messages to your GroupWise system for delivery. Messages from the GroupWise client in Caching mode are processed through the priority 1 subdirectory of wpcsin.

When you send a message in Caching mode, the GroupWise client connects to your GroupWise system. It polls the wpcsin\1 directory and compresses any outgoing messages, requests, or both into a file. If the compressed file totals over 50 KB, additional compressed files are created. The GroupWise client then moves the compressed files into the wpgwsend directory.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). The Message Transfer Agent (MTA) was originally named the Connection Server (CS). Hence, the directory name wpcsin for the input queue, although the MTA is not involved in processing messages in your Caching mailbox. Some naming conventions were originally preserved for backward compatibility.

0-7 directories

The priority 0-7 subdirectories in the connection input queue (wpcsin) parallel those found in the wpcsin directory in your post office.

6.5.4 wpcsout\ofs directory

The wpcsout\ofs directory is the output queue for the connection that transfers messages from your Online mailbox. Messages from your GroupWise system are processed through the priority 1 subdirectory of wpcsout\ofs.

The GroupWise client scans the wpcsout\ofs\1 subdirectory and updates the user.db and msg.db files with the information received from your Online mailbox.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). The Message Transfer Agent (MTA) was originally named the Connection Server (CS). Hence, the directory names wpcsin and ofs for the input queue, though the MTA is not involved in processing messages in your Remote mailbox. Some naming conventions were originally preserved for backward compatibility.

0-7 directories

The priority 0-7 subdirectories in the connection output queue (wpcsout\ofs) parallel those found in the ofs directory in your post office.

6.5.5 wpgwsend directory

The wpgwsend directory holds compressed files that contain outgoing messages, requests, or both. When a connection to your GroupWise system is established, the GroupWise client uploads the files to your Online mailbox.

Historical Note: WP Office Remote, the predecessor of the GroupWise client Remote mode, was originally designed by WordPerfect Corporation (WPCorp). Hence, the name wpgwsend. Some naming conventions were originally preserved for backward compatibility.

6.5.6 wpgwrecv directory

The wpgwrecv directory holds compressed files that contain messages or other information that have been received from your Online mailbox. The GroupWise client decompresses the files and places the message files into the wpcsout\ofs\1 directory.

Historical Note: WP Office Remote, the predecessor of the GroupWise client Remote mode, was originally designed by WordPerfect Corporation (WPCorp). Hence, the name wpgwrecv. Some naming conventions were originally preserved for backward compatibility.

6.5.7 remote*n*.log file

The remote*n*.log files are saved versions of the connection logs you can view in the GroupWise client by clicking *Accounts* > *Connection Log*. These log files can be useful for troubleshooting problems with your connection to your Online mailbox.

6.5.8 status*n*.txt file

The `statusn.txt` files are saved versions of the content that displays in the GroupWise client when you click *Accounts* > *Show Status Window*. These log files can be useful for troubleshooting problems with your connection to your Online mailbox.

7 Agent Installation Directories

- Section 7.1, “GroupWise Agent Installation (MTA, POA, and DVA),” on page 87
- Section 7.2, “Internet Agent Installation,” on page 98
- Section 7.3, “Monitor Agent Installation,” on page 103

7.1 GroupWise Agent Installation (MTA, POA, and DVA)

The Message Transfer Agent (MTA), Post Office Agent (POA), and Document Viewer Agent (DVA) are always installed together. The agent installation directory differs depending on the platform where the agents are installed.

- Section 7.1.1, “Linux MTA, POA, and DVA Installation Directory,” on page 87
- Section 7.1.2, “Windows MTA, POA, and DVA Installation Directory,” on page 94
- Section 7.1.3, “DVA Working Directory,” on page 97

7.1.1 Linux MTA, POA, and DVA Installation Directory

/opt/novell/groupwise/agents	Linux agent installation directory
bin	Subdirectory for GroupWise agent executables
gwmta	Message Transfer Agent (MTA) executable
gwpoa	Post Office Agent (POA) executable
gwdva	Document Viewer Agent (DVA) executable
gwdva.dir	DVA working directory
strtupxx.mta	Boilerplate MTA startup file
strtupxx.poa	Boilerplate POA startup file
strtup.dva	Boilerplate DVA startup file
gwdca	Document Converter Agent (DCA) executable
gwha	GroupWise High Availability service executable
grpwise	GroupWise agent startup script
gwcsrgen	GroupWise Generate CSR utility
gwtmstmp	GroupWise Time Stamp utility
lib	Subdirectory for GroupWise agent library files
gwmtaxxx.fil	MTA language information file
gwpoaxxx.fil	POA language information file
gwen1xxx.fil	Agent engine language information file

📁 share	Subdirectory for agent shared files
🗋 *domain*.mta	MTA startup file for a specific domain
🗋 *post_office*.poa	POA startup file for a specific post office
🗋 gwdva.dva	DVA startup file
📁 agtcon	Subdirectory for agent console files
📁 help	Subdirectory agent console help files
📁 webcon	Subdirectory for agent Web console files
📁 help	Subdirectory for agent Web console help files
📁 gwcsrgen	Subdirectory for the GWCSRGEN utility files
📁 help	Subdirectory for GWCSRGEN utility help files
📁 /etc/init.d	Standard Linux location for application startup scripts
🗋 grpwise	Startup script for the MTA, POA, and DVA
📁 rc3.d	Standard Linux location for runlevel-3 symbolic links
🗋 S*nn*grpwise	Symbolic link to the startup script for the MTA, POA, and DVA
📁 rc5.d	Standard Linux location for runlevel-5 symbolic links
🗋 S*nn*grpwise	Symbolic link to the startup script for the POA and MTA
📁 /usr/sbin	Standard Linux location for application startup script links
🗋 rcgrpwise	Link to the startup script for the MTA, POA, and DVA
📁 /etc/opt	Standard Linux location for application configuration files
📁 novell/groupwise	Subdirectory for GroupWise configuration files
🗋 gwha.conf	GroupWise High Availability service configuration file
📁 agents	Subdirectory for GroupWise agent information
🗋 uid.conf	Agent configuration file for running as a non-root user
📁 /etc/xinetd.d	Standard Linux location for xinetd configuration files
🗋 gwha	Configuration file for the GroupWise High Availability service

📁 /var/log/	Standard Linux location for application log files
📁 novell/groupwise	Subdirectory for GroupWise agent log files
📁 *domain*.mta	Domain-specific subdirectory for MTA log files
🗋 *mmdd*mta.*nnn*	MTA log files
📁 *post_office*.poa	Post office-specific subdirectory for POA log files
🗋 *mmdd*poa.*nnn*	POA log files
📁 gwdva	Subdirectory for DVA log files
🗋 *mmdd*dva.*nnn*	DVA log files
📁 gwdca	Subdirectory for DCA log files
🗋 *mmdd*dca.*nnn*	DCA log files

agents directory

On a Linux server, the agents are always installed in subdirectories of /opt/novell/groupwise/agents.

bin directory

The bin directory holds GroupWise executable files.

gwmta file

The gwmta file is the MTA executable. You run this executable file to start the MTA. See “Starting the Linux Agents with a User Interface” in “Installing GroupWise Agents” in the *GroupWise 2012 Installation Guide*.

gwpoa file

The gwpoa file is the Post Office Agent executable. You run this executable file to start the Post Office Agent. See “Starting the Linux Agents with a User Interface” in “Installing GroupWise Agents” in the *GroupWise 2012 Installation Guide*.

gwdva file

The gwdva file is the Document Viewer Agent executable. You run this executable file to start the Document Viewer Agent. See “Starting the Linux Agents with a User Interface” in “Installing GroupWise Agents” in the *GroupWise 2012 Installation Guide*.

strtupxx.mta file

The strtup*xx*.mta file is the boilerplate file from which a domain-specific *domain*.mta file is created in the share directory. The *xx* in the startup file name represents a two-letter language code.

strtup*xx*.poa file

The strtup*xx*.poa file is the boilerplate file from which a post office-specific *post_office*.poa file is created in the share directory. The *xx* in the startup file name represents a two-letter language code.

strtup.dva file

The strtup.dva file is the boilerplate file from which a customizable gwdva.dva file is created in the share directory.

gwdca file

The gwdca file is the Document Converter Agent executable. The POA runs this executable file to start the Document Converter Agent. See "Enabling the Document Viewer Agent (DVA) for Indexing" in "Post Office Agent" in the *GroupWise 2012 Administration GuideGroupWise 2012 Installation Guide*.

gwha file

The gwha file is the GroupWise High Availability service executable. If the MTA, POA, or DVA goes down for any reason, the High Availability service automatically restarts it. See "Enabling the GroupWise High Availability Service for the Linux GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

grpwise file

The grpwise script is created automatically during installation. You can use the script to start, restart, stop, and display status information about the GroupWise agents. For more information about starting the agents, see "Installing and Starting the Linux GroupWise Agents" in "Installing a Basic GroupWise System" in the *GroupWise 2012 Installation Guide*.

gwcsrgen file

The gwcsrgen file is the GroupWise Generate CSR utility. If you enable SSL for the agents, they need access to a server certificate and private key. You can use the GroupWise Generate CSR utility (GWCSRGEN) to generate a Certificate Signing Request (CSR) file and a Private Key file. For more information, see "Server Certificates and SSL Encryption" in "Security Administration" in the *GroupWise 2012 Administration Guide*.

gwtmstmp file

The gwtmstmp file is the GroupWise Time Stamp utility. If you deselect *Allow Purge of Items Not Backed Up* in ConsoleOne, user databases (userxxx.db) must be time-stamped every time a backup is performed so that items can be purged only after being backed up. You can use the GroupWise Time Stamp (GWTMSTMP) utility to ensure that GroupWise user databases include the dates when they were last backed up, restored, and retained. For more information, see "GroupWise Time Stamp Utility" in "Databases" in the *GroupWise 2012 Administration Guide*.

lib directory

The lib directory holds GroupWise library files.

*.fil files

These files contain all language-specific information for the MTA and the POA. The first five characters of the file name are the agent name. The last two characters *xx* are a language code.

share directory

The share directory holds agent startup files and files that are used by the agent server consoles and Web consoles.

domain.mta file

The MTA startup file contains switches for the MTA. Switch settings placed in the MTA startup file override comparable options set for the MTA in ConsoleOne.

During installation, a customized version of the `strtupxx.mta` file, named *`domain`*`.mta`, is created in the `share` directory. This customized version has the --home switch automatically set to the domain directory the MTA will service. See "Using MTA Startup Switches" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

post_office.poa file

The POA startup file contains switches for the POA. Switch settings placed in the POA startup file override comparable options set for the POA in ConsoleOne.

During installation, a customized version of the `strtupxx.poa` file, named *`post_office`*`.poa`, is created in the `share` directory. This customized version has the --home switch automatically set to the post office directory the POA will service. See "Using POA Startup Switches" in "Post Office Agent" in the *GroupWise 2012 Administration Guide*.

gwdva.dva file

The DVA startup file contains switches for the DVA. No switches are set during installation. DVA configuration must be performed after installation, as described in "Configuring the DVA" in "Document Viewer Agent" in the *GroupWise 2012 Administration Guide*.

agtcon directory

The `agtcon` directory holds subdirectories and files used by the agent server consoles, such as help files. See "Using the POA Server Console" in "Post Office Agent" and "Using the MTA Server Console" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

webcon directory

The `webcon` directory holds subdirectories and files used by the agent Web consoles, such as help files. See "Using the POA Web Console" in "Post Office Agent" and "Using the MTA Web Console" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

gwcsrgen directory

The `gwcsrgen` directory holds subdirectories and files used by the GroupWise Generate CSR (GWCSRGEN) utility, such as help files. See "Generating a Certificate Signing Request" in "Security Administration" in the *GroupWise 2012 Administration Guide*.

/etc/init.d directory

The `/etc/init.d` directory is the standard location for Linux startup scripts.

grpwise file

The `grpwise` script is created automatically during installation. You can use the script to start, restart, stop, and display status information about the GroupWise agents. For more information about starting the agents, see "Installing and Starting the Linux GroupWise Agents" in "Installing a Basic GroupWise System" in the *GroupWise 2012 Installation Guide*.

rc3.d directory

The `rc3.d` directory holds symbolic links to scripts that you want your Linux server to run when it is booted to runlevel 3 (multi-user; boots to a text mode login prompt without the X Window System). The symbolic link to the `grpwise` script is `S`*`nn`*`grpwise`. It is created if you choose during installation to have the agents start automatically when the server boots. See "Starting the Linux Agents on System Startup" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

rc5.d directory

The `rc5.d` directory holds symbolic links to scripts that you want your Linux server to run when it is booted to runlevel 5 (multi-user; boots to the X Window System login dialog box). The symbolic link to the `grpwise` script is `S`*`nn`*`grpwise`. It is created if you choose during installation to have the agents start automatically when the server boots. See "Starting the Linux Agents on System Startup" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

/usr/sbin directory

The `/usr/sbin` directory is the standard location for application scripts that can be run from any directory on the Linux server. The `/usr/sbin` directory is always included in the PATH environment variable. Files in this directory are links to the corresponding script files in `/etc/init.d`.

/etc/opt/novell/groupwise directory

The `/etc/opt` directory is the standard Linux location for application configuration files. Files that configure how the GroupWise agents interact with Linux are stored in the `novell/groupwise` subdirectory.

gwha.conf file

The `gwha.conf` file is on of the configuration files for the GroupWise High Availability service (gwha). It is created automatically during installation and provides information necessary for the High Availability service to restart the GroupWise agents if they go down unexpectedly. See "Enabling the GroupWise High Availability Service for the Linux GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

uid.conf file

The `uid.conf` file configures the GroupWise agents to run as a non-root user. See "Running the Linux GroupWise Agents as a Non-root User" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

/etc/xinetd.d directory

The `/etc/xinetd.d` directory is the standard Linux location for configuration files for services controlled by the Extended Internet Services daemon.

gwha file

The `gwha` file is one of the configuration files for the GroupWise High Availability service (gwha). It is created automatically during installation and provides information necessary for the High Availability service to restart the GroupWise agents if they go down unexpectedly. See "Enabling the GroupWise High Availability Service for the Linux GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

/var/log/novell/groupwise directory

The /var/log directory is the standard location for log files on Linux. All GroupWise agent log files are created in the novell/groupwise subdirectory.

domain.mta directory

The *domain*.mta directory is a domain-specific location for MTA log files.

Within the *domain*.mta directory, the MTA creates log files (*mmddxxx.nnn*) to inform you of its processing and any problems it encounters. For more information about log files, see "Using MTA Log Files" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

The first two digits of the file name represent the month; the next two digits represent the day of the month; the next three characters indicate what program created the log. The three-digit extension is a sequence number for multiple log files created on the same day. For example, 0518mta.002 is the second MTA log file created on May 18.

post_office.poa directory

The *post_office*.poa directory is a post office-specific location for POA log files.

Within the *post_office*.poa directory, the POA creates log files (*mmdd*poa.*nnn*) to inform you of its processing and any problems it encounters. For more information about log files, see "Using POA Log Files" in "Post Office Agent" in the *GroupWise 2012 Administration Guide* guide.

The first two digits of the file name represent the month, the next two digits represent the day of the month, and the next three characters indicate what program created the log. The three-digit extension is a sequence number for multiple log files created on the same day. For example, 0518poa.002 is the second POA log file created on May 18.

gwdva directory

The gwdva directory holds DVA log files.

Within the gwdva directory, the DVA creates log files (*mmdd*dva.*nnn*) to inform you of its processing and any problems it encounters. For more information about log files, see "Using DVA Log Files" in "Document Viewer Agent" in the *GroupWise 2012 Administration Guide* guide.

The first two digits of the file name represent the month, the next two digits represent the day of the month, and the next three characters indicate what program created the log. The three-digit extension is a sequence number for multiple log files created on the same day. For example, 0518dva.002 is the second DVA log file created on May 18.

gwdca directory

The gwdca directory holds DCA log files.

Within the gwdca directory, the DCA creates log files (*mmdd*dca.*nnn*) to inform you of its processing and any problems it encounters. For more information about the DCA, see "Configuring the Document Converter Agent (DCA)" in "Post Office Agent" in the *GroupWise 2012 Administration Guide* guide.

The first two digits of the file name represent the month, the next two digits represent the day of the month, and the next three characters indicate what program created the log. The three-digit extension is a sequence number for multiple log files created on the same day. For example, 0518dva.002 is the second DVA log file created on May 18.

7.1.2 Windows MTA, POA, and DVA Installation Directory

c:\Program Files (x86)\Novell\GroupWise Server\Agents	Windows agent installation directory (64-bit machine)
c:\Program Files\Novell\GroupWise Server\Agents	Windows agent installation directory (32-bit machine)
gwenv1a.dll	GroupWise Agent Engine
gwmta.exe	Message Transfer Agent (MTA) program
gwpoa.exe	Post Office Agent (POA) program
gwdva.exe	Document Viewer Agent (DVA) program
gwdca.exe	Document Converter Agent (DCA) program
gwdva.dir	DVA working directory
strtup*xx*.mta	Boilerplate MTA startup file
strtup*xx*.poa	Boilerplate POA startup file
strtup.dva	Boilerplate DVA startup file
domain.mta	Customized MTA startup file for a specific domain
post_office.poa	Customized POA startup file for a specific post office
gwdva.dva	Customized DVA startup file
gwmta*nxx*.chm	MTA online help file
gwpoa*nxx*.chm	POA online help file
gwpoa*nxx*.dll	POA language information files
gwmta*nxx*.dll	MTA language information files
gwsnmp.dll	Customized DLL program for SNMP
gwwww1.dll	Customized DLL program for MIME
dbcopy.exe	GroupWise Database Copy utility
gwtmstmp.exe	GroupWise Timestamp utility
help	Subdirectory for GroupWise agent Web console help files

c:\Program Files\Novell\GroupWise\Agents

On a Windows server, the GroupWise agents can be installed in any directory you choose. The default is `c:\Program Files\Novell\GroupWise\Agents`. The agent icons are set up to include the full path to whatever directory you choose.

gwenv1a.dll file

The `gwenv1a.dll` file is the GroupWise Agent Engine, a program that is shared by the MTA, POA, and DVA. It provides the following services to the agents:

- Database management
- File operations
- Message handling
- Thread management

- Semaphores (file/record locking)
- Date/time services

The first agent started on a server automatically starts the GroupWise Agent Engine DLL.

gwmta.exe file

The `gwmta.exe` file is the MTA program. You run this executable file to start the MTA. See "Starting the Windows GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

gwpoa.exe file

The `gwpoa.exe` file is the Post Office Agent program. You run this executable file to start the Post Office Agent. See "Starting the Windows GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

gwdva.exe file

The `gwdva.exe` file is the Document Viewer Agent executable. You run this executable file to start the Document Viewer Agent. See "Starting the Windows GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

gwdca.exe file

The `gwdca.exe` file is the Document Converter Agent executable. The POA runs this executable file to start the Document Converter Agent. See "Enabling the Document Viewer Agent (DVA) for Indexing" in "Post Office Agent" in the *GroupWise 2012 Administration GuideGroupWise 2012 Installation Guide*.

gwsnmp.dll file

The `gwsnmp.dll` file provides interaction with the Windows SNMP Service, so that you can monitor the GroupWise agents using an SNMP monitoring program.

gwwww1.dll file

The `gwwww1.dll` file provides parsing of MIME messages received from the Internet.

strtup*xx*.poa file and *post_office*.poa file

The POA startup file contains switches for the POA. Switch settings placed in the POA startup file override comparable options set for the POA in ConsoleOne. The *xx* in the startup file name represents a two-letter language code.

During installation, a customized version of the POA startup file, named *`post_office`*`.poa`, is created. This customized version has the --home switch automatically set to the post office directory the POA will service. See "Using POA Startup Switches" in "Post Office Agent" in the *GroupWise 2012 Administration Guide*.

strtup*xx*.mta file and *domain*.mta file

The MTA startup file contains switches for the MTA. Switch settings placed in the MTA startup file override comparable options set for the MTA in ConsoleOne. The *xx* in the startup file name represents a two-letter language code.

During installation, a customized version of the MTA startup file, named `domain.MTA`, is created. This customized version has the --home switch automatically set to the domain directory the MTA will service. See "Using MTA Startup Switches" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

strtup.dva file and gwdva.dva file

The `strtup.dva` file is the boilerplate file from which a customizable `gwdva.dva` file is created. No switches are set during installation. DVA configuration must be performed after installation, as described in "Configuring the DVA" in "Document Viewer Agent" in the *GroupWise 2012 Administration Guide*.

agentnxx.chm files

The `.chm` files contain the online documentation for the agents. Online Help is available by clicking Help in the agent consoles on the server where the agents are running. In addition, dialog boxes have a Help button for context-sensitive Help.

The first five characters of the file name are the agent name. The digit *n* is a version number. The last two characters *xx* are a language code.

agentnxx.dll files

The `agentnxx.dll` files contain all language-specific information for the agents. The digit *n* is a version number. The last two characters *xx* are a language code.

dbcopy.exe file

The `dbcopy.exe` file is the GroupWise Database Copy utility, which copies files from a live GroupWise post office or domain to a static location for backup. During the copy process, DBCopy prevents the files from being modified, using the same locking mechanism used by other GroupWise programs that access databases. This ensures that the backed-up versions are consistent with the originals even when large databases take a substantial amount of time to copy. For more information, see "GroupWise Database Copy Utility" in "Databases" in the *GroupWise 2012 Administration Guide*.

gwtmstmp.exe file

The `gwtmstmp.exe` file is the GroupWise Time Stamp utility. If you deselect *Allow Purge of Items Not Backed Up* in ConsoleOne, user databases (`userxxx.db`) must be time-stamped every time a backup is performed so that items can be purged only after being backed up. You can use the GroupWise Time Stamp (GWTMSTMP) utility to ensure that GroupWise user databases include the dates when they were last backed up, restored, and retained. For more information, see "GroupWise Time Stamp Utility" in "Databases" in the *GroupWise 2012 Administration Guide*.

help directory

The `help` directory contains language-specific subdirectories for the help files available from the POA and MTA Web consoles. See "Using the POA Web Console" in "Post Office Agent" and "Using the MTA Web Console" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

7.1.3 DVA Working Directory

The Document Viewer Agent (DVA) creates its working directory as a subdirectory of its installation directory. The default location varies by platform:

Linux: `/opt/novell/groupwise/agents/bin/gwdva.dir`

Windows: `c:\Program Files\Novell\GroupWise Server\Agents\gwdva.dir`

You can specify the location for the DVA working directory using the --home switch. See "Setting the DVA Home Directory" in "Document Viewer Agent" in the *GroupWise 2012 Administration Guide*.

📁 `gwdva.dir`	Document Viewer Agent working directory
🗋 `gwdav`*`nnn`*`.ste`	State files for Document Viewer Agent worker threads
📁 `quarantine`	Subdirectory for quarantined documents that failed HTML conversion
📁 `log`	Subdirectory for Document Viewer Agent log files (Windows only)
📁 `temp`	Subdirectory for the temporary files using during HTML conversion
📁 `template`	Subdirectory for the template file used for HTML conversion

gwdva.dir directory

The `gwdva.dir` directory is the working directory for the Document Viewer Agent.

gwdva*nnn*.ste files

The `gwdva`*`nnn`*`.ste` files hold state information about each DVA worker thread. To control the number of worker threads, see "Controlling Thread Usage" in "Document Viewer Agent" in the *GroupWise 2012 Administration Guide*.

quarantine directory

The `quarantine` directory contains document files that cannot be converted to HTML format, so that they can be examined manually if necessary. See "Enabling the DVA Document Quarantine" in "Document Viewer Agent" in the *GroupWise 2012 Administration Guide*.

log directory (Windows only)

The `log` directory stores log files produced by the Document Viewer Agent. See "Using DVA Log Files" in "Document Viewer Agent" in the *GroupWise 2012 Administration Guide*.

NOTE: On Linux, DVA log files are stored in the typical location for log files on Linux (`/var/log/novell/groupwise/gwdva`), rather than in the `gwdva.dir` directory.

temp directory

The `temp` directory is used for the temporary files created during HTML conversion.

template directory

The `template` directory holds the HTML template file that the DVA uses when converting document files into HTML format.

7.2 Internet Agent Installation

- Section 7.2.1, "Linux Internet Agent Installation Directory," on page 98
- Section 7.2.2, "Windows GWIA Installation Directory," on page 101

7.2.1 Linux Internet Agent Installation Directory

/opt/novell/groupwise/agents	Linux agent installation directory
bin	Subdirectory for GroupWise agent executables
gwia	GWIA executable
gwha	GroupWise High Availability service executable
grpwise	GroupWise agent startup script
lib	Subdirectory for GroupWise agent library files
gwiaxxx.fil	GWIA language information file
share	Subdirectory for agent shared files
gwia.cfg	GWIA configuration file
agtcon	Subdirectory for agent console files
help	Subdirectory agent console help files
webcon	Subdirectory for agent Web console files
help	Subdirectory for agent Web console help files
/etc/init.d	Standard Linux location for application startup scripts
grpwise	Startup script for the GWIA
rc3.d	Standard Linux location for run-level-3 symbolic links
*Snn*grpwise	Symbolic link to the startup script for the GWIA
rc5.d	Standard Linux location for run-level-5 symbolic links
*Snn*grpwise	Symbolic link to the startup script for the GWIA
/usr/sbin	Standard Linux location for application startup script links
rcgrpwise	Link to the startup script for the GWIA

/etc/opt	Standard Linux location for application configuration files
novell/groupwise	Subdirectory for GroupWise configuration files
gwha.conf	GroupWise High Availability service configuration file
agents	Subdirectory for GroupWise agents
uid.conf	Agent configuration file for running as a non-root user

/var/log/	Standard Linux location for application log files
novell/groupwise	Subdirectory for GroupWise agent log files
gwia.*domain*	Domain-specific subdirectory for GWIA log files
mmddgwia.nnn	GWIA log files

See also Section 6.4, "Internet Agent Queue Directory," on page 74.

agents directory

On a Linux server, the GWIA is always installed in subdirectories of /opt/novell/groupwise/agents.

bin directory

The bin directory holds GroupWise executable files.

gwia file

The gwia file is the GWIA executable. You run this executable file to start the GWIA. See "Starting the GWIA" in "Installing the GroupWise Internet Agent" in the *GroupWise 2012 Installation Guide*.

gwha file

The gwha file is the GroupWise High Availability service executable. If the GWIA goes down for any reason, the High Availability service automatically restarts it. See "Enabling the GroupWise High Availability Service for the Linux GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

grpwise file

The grpwise script is created automatically during installation. You can use the script to start, restart, stop, and display status information about the GWIA. For more information about starting the agents, see "Installing and Starting the Linux GroupWise Agents" in "Installing a Basic GroupWise System" in the *GroupWise 2012 Installation Guide*.

lib directory

The lib directory holds GroupWise shared library files.

gwia*xx*.fil files

These files contain all language-specific information for the GWIA. The last two characters *xx* are a language code.

share directory

The share directory holds agent startup files and files that are used by the agent consoles and Web consoles.

gwia.cfg

The GWIA configuration file contains switches for the GWIA. Switch settings placed in the GWIA configuration file override comparable configuration options set for the GWIA in ConsoleOne. The startup file is named the same as the GWIA object in ConsoleOne. The default is gwia.

During installation, the GWIA startup file is created in the share directory with the --home switch automatically set to the domain directory where the GWIA queue directory is located. See "Using GWIA Startup Switches" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

agtcon directory

The agtcon directory holds subdirectories and files used by the agent consoles, such as help files.

webcon directory

The webcon directory holds subdirectories and files used by the agent Web consoles, such as help files.

/etc/init.d directory

The /etc/init.d directory is the standard location for Linux startup scripts.

grpwise file

The grpwise script is created automatically during installation. You can use the script to start, restart, stop, and display status information about the GWIA. For more information about starting the GWIA, see "Starting the GWIA" in "Installing the GroupWise Internet Agent" in the *GroupWise 2012 Installation Guide*.

rc3.d directory

The rc3.d directory holds symbolic links to scripts that you want your Linux server to run when it is booted to runlevel 3 (multi-user; boots to a text mode login prompt without the X Window System). The symbolic link to the grpwise script is S*nn*grpwise. It is created if you choose during installation to have the GWIA start automatically when the server boots.

rc5.d directory

The rc5.d directory holds symbolic links to scripts that you want your Linux server to run when it is booted to runlevel 5 (multi-user; boots to the X Window System login dialog box). The symbolic link to the grpwise script is S*nn*grpwise. It is created if you choose during installation to have the GWIA start automatically when the server boots.

/usr/sbin directory

The `/usr/sbin` directory is the standard location for application scripts that can be run from any directory on the Linux server. The `/usr/sbin` directory is always included in the PATH environment variable. Files in this directory are links to the corresponding script files in `/etc/init.d`.

/etc/opt/novell/groupwise directory

The `/etc/opt` directory is the standard location of application configuration files on Linux. Files that configure how the GWIA interacts with Linux are stored in the `novell/groupwise` subdirectory.

gwha.conf file

The `gwha.conf` file is the GroupWise High Availability service configuration file. It is created automatically during installation and provides the information necessary for the High Availability service to restart the GWIA if it goes down unexpectedly. See "Enabling the GroupWise High Availability Service for the Linux GroupWise Agents" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

uid.conf file

The `uid.conf` file configures the GWIA to run as a non-root user. See "Running the Linux GroupWise Agents as a Non-root User" in "Installing GroupWise Agents" in the *GroupWise 2012 Installation Guide*.

/var/log/novell/groupwise directory

The `/var/log` directory is the standard location for log files on Linux. All GroupWise agent log files are created in the `novell/groupwise` subdirectory.

gwia.*domain* directory

The `gwia.`*`domain`* directory is a domain-specific location for GWIA log files.

Within the *`domain`*`.gwia` directory, the GWIA creates log files (*mmdd*`gwia.`*`nnn`*) to inform you of its processing and any problems it encounters. For more information about log files, see "Using GWIA Log Files" in "Internet Agent" in the *GroupWise 2012 Administration Guide*.

The first two digits of the file name represent the month; the next two digits represent the day of the month. The three-digit extension is a sequence number for multiple log files created on the same day. For example, `0518gwia.002` is the second GWIA log file created on May 18.

7.2.2 Windows GWIA Installation Directory

`c:\Program Files (x886)\Novell\GroupWise Server\GWIA`	Windows Internet Agent installation directory (64-bit machine)
`c:\Program Files\Novell\GroupWise Server\GWIA`	Windows Internet Agent installation directory (32-bit machine)
`gwenv1a.dll`	GroupWise Agent Engine
`gwia.exe`	GWIA program

gwia.cfg	GWIA configuration file
gwia*nxx*.chm	GWIA online help file
gwia*nxx*.dll	GWIA language information file
gwwww1.dll	Customized DLL program for MIME

See also Section 6.4, “Internet Agent Queue Directory,” on page 74.

c:\Program Files\Novell\GroupWise Server\GWIA

On a Windows server, the GWIA can be installed in any directory you choose. The default location is c:\Program Files\Novell\GroupWise Server\GWIA. The GWIA icon is set up to include the full path to whatever directory you choose.

gwenv1a.dll file

The gwenv1a.dll file is the GroupWise Agent Engine, a program that is shared by all GroupWise agents. It provides the following services to the agents:

- Database management
- File operations
- Message handling
- Thread management
- Semaphores (file/record locking)
- Date/time services

The first agent started on a server automatically starts the GroupWise Agent Engine DLL.

gwia.exe file

The gwia.exe file is the GWIA program. You run this executable file to start the GWIA. See “Starting the GWIA” in “Installing the GroupWise Internet Agent” in the *GroupWise 2012 Installation Guide*.

gwia.cfg file

On Windows, the *gwia*.cfg file in the installation directory is simply a pointer to the *gwia*.cfg file under the domain directory where the actual GWIA configuration file is located.

gwia*nxx*.chm file

The .chm files contain the online documentation for the GWIA. Online help is available by clicking *Help* in the GWIA console on the server where the agent is running. In addition, dialog boxes have a Help button for context-sensitive Help.

The digit *n* is a version number. The last two characters *xx* are a language code.

gwia*nxx*.dll files

The gwia*nxx*.dll files contain all language-specific information for the GWIA. The digit *n* is a version number. The last two characters *xx* are a language code.

gwwww1.dll file

The gwwww1.dll file provides parsing of MIME messages received from the Internet.

7.3 Monitor Agent Installation

- Section 7.3.1, "Linux Monitor Agent Installation Directory," on page 103
- Section 7.3.2, "Windows Monitor Agent Installation Directory," on page 105

7.3.1 Linux Monitor Agent Installation Directory

/opt/novell/groupwise/agents	Linux agent installation directory
bin	Subdirectory for GroupWise agent executables
gwmon	Monitor Agent executable
lib	Subdirectory for GroupWise agent library files
gwmonxxx.fil	Monitor Agent language information files
share	Subdirectory for agent shared files
monitor.xml	Monitor Agent configuration file
/etc/init.d	Standard Linux location for application startup scripts
grpwise-ma	Startup script for the Monitor Agent
rc3.d	Standard Linux location for run-level-3 symbolic links
S*nn*grpwise-ma	Symbolic link to the startup script for the Monitor Agent
rc5.d	Standard Linux location for run-level-5 symbolic links
S*nn*grpwise-ma	Symbolic link to the startup script for the Monitor Agent
/usr/sbin	Standard Linux location for application startup script links
rcgrpwise-ma	Link to the startup script for the Monitor Agent
/var/log/	Standard Linux location for application log files
novell/groupwise	Subdirectory for GroupWise agent log files
gwmon	Subdirectory for Monitor Agent log files
*mmdd*mon.*nnn*	Monitor Agent event log files
*mmdd*hist.*nnn*	Monitor Agent history log files
acct	Subdirectory for Monitor Agent accounting files

Monitor Application files are integrated into your Web server to support the Monitor Web console. See Section 8, "Web Application Installation Directories on Your Web Server," on page 109.

agents directory

On a Linux server, the Monitor Agent is always installed in subdirectories of `/opt/novell/groupwise/agents`.

bin directory

The `bin` directory holds GroupWise executable files.

gwmon file

The `gwmon` file is the Monitor Agent executable. You run this executable file to start the Monitor Agent. See “Starting the Linux Monitor Agent as a Daemon” in “Installing GroupWise Monitor” in the *GroupWise 2012 Installation Guide*.

lib directory

The `lib` directory holds GroupWise shared library files.

gwmon*xxx*.fil files

These files contain all language-specific information for the Monitor Agent. The last two characters *xx* are a language code.

share directory

The `share` directory holds agent startup files and files that are used by the agent consoles and Web consoles.

monitor.xml file

The `monitor.xml` file is a specialized configuration file for use in the XIS environment.

/etc/init.d directory

The `/etc/init.d` directory is the standard location for Linux startup scripts.

grpwise-ma file

The `grpwise-ma` script is created automatically during installation. You can use the script to start, restart, stop, and display status information about the Monitor Agent. For more information about starting the Monitor Agent, see “Starting the Linux Monitor Agent as a Daemon” in “Installing GroupWise Monitor” in the *GroupWise 2012 Installation Guide*.

rc3.d directory

The `rc3.d` directory holds symbolic links to scripts that you want your Linux server to run when it is booted to runlevel 3 (multi-user; boots to a text mode login prompt without the X Window System). The symbolic link to the `grpwise-ma` script is S*nn*`grpwise-ma`. It is created if you choose during installation to have the Monitor Agent start automatically when the server boots.

rc5.d directory

The `rc5.d` directory holds symbolic links to scripts that you want your Linux server to run when it is booted to runlevel 5 (multi-user; boots to the X Window System login dialog box). The symbolic link to the `grpwise-ma` script is `S`*nn*`grpwise-ma`. It is created if you choose during installation to have the Monitor Agent start automatically when the server boots.

/usr/sbin directory

The `/usr/sbin` directory is the standard location for application scripts that can be run from any directory on the Linux server. The `/usr/sbin` directory is always included in the PATH environment variable. Files in this directory are links to the corresponding script files in `/etc/init.d`.

/var/log/novell/groupwise directory

The `/var/log` directory is the standard location for log files on Linux. All GroupWise agent log files are created in the `novell/groupwise` subdirectory.

gwmon directory

Within the `gwmon` directory, the Monitor Agent creates two different types of log files. The *mmdd*`mon.`*nnn* `log` files inform you of its processing and any problems it encounters. The *mmdd*`hist.`*nnn* `log` files record dumps of all MIB values gathered during each poll cycle. For more information about log files, see "Configuring Monitor Agent Log Settings" in "Monitor" in the *GroupWise 2012 Administration Guide*.

The first two digits of the file name represent the month; the next two digits represent the day of the month. The three-digit extension is a sequence number for multiple log files created on the same day. For example, `0518mon.002` is the second Monitor Agent log file created on May 18.

acct directory

Within the `gwmon` directory, the Monitor Agent creates an `acct` subdirectory for storing accounting files received from the GWIA and GroupWise gateways. For more information, see "Receiving and Forwarding the Accounting Files" in "Monitor" in the *GroupWise 2012 Administration Guide*.

7.3.2 Windows Monitor Agent Installation Directory

c:\Program Files (x86)\Novell\GroupWise Server\Monitor	Windows Monitor Agent installation directory (64-bit machine)
c:\Program Files\Novell\GroupWise Server\Monitor	Windows Monitor Agent installation directory (32-bit machine)
gwenv1a.dll	GroupWise Agent Engine
gwmon.exe	Monitor Agent program
monitor.xml	Monitor Agent configuration file
gwmon*xx*.dll	Monitor Agent language information file
gwsnmp.dll	Customized DLL program for SNMP
*mmdd*mon.*nnn*	Monitor Agent event log files
*mmdd*hist.*nnn*	Monitor Agent history log files

📁 monwork	Monitor Agent working directory
📁 acct	Monitor Agent directory for accounting files

Monitor Application files are integrated into your Web server to support the Monitor Web console. See Section 8, "Web Application Installation Directories on Your Web Server," on page 109.

c:\Program Files\Novell\GroupWise Server\Monitor directory

On a Windows server, the Monitor Agent can be installed in any directory you choose. The default location is `c:\Program Files\Novell\GroupWise Server\Monitor`. The Monitor Agent desktop icon is set up to include the full path to whatever directory you choose.

gwenv1a.dll file

The `gwenv1a.dll` file is the GroupWise Agent Engine, a program that is shared by all GroupWise agents. It provides the following services to the agents:

- Database management
- File operations
- Message handling
- Thread management
- Semaphores (file/record locking)
- Date/time services

The first agent started on a server automatically starts the GroupWise Agent Engine DLL.

gwmon.exe file

The `gwmon.exe` file is the Monitor Agent program. You run this executable file to start the Monitor Agent. See "Windows: Setting Up GroupWise Monitor" in "Installing GroupWise Monitor" in the *GroupWise 2012 Installation Guide*.

monitor.xml

The `monitor.xml` file stores the Monitor Agent configuration information that you establish in the Monitor Agent console, as described in "Configuring the Monitor Agent" in "Monitor" in the *GroupWise 2012 Administration Guide*.

gwmon*xx*.dll files

The `gwmonxx.dll` files contain all language-specific information for the Monitor Agent. The last two characters *xx* are a language code.

gwsnmp.dll file

The `gwsnmp.dll` file provides interaction with the Windows SNMP Service, so that the Monitor Agent can throw SNMP traps. See "Configuring SNMP Trap Notification for Agent Problems" in "Monitor" in the *GroupWise 2012 Administration Guide*.

*mmdd*mon.*nnn* files and *mmdd*hist.*nnn* files

The Monitor Agent creates two different types of log files. The *mmdd*mon.*nnn* log files inform you of its processing and any problems it encounters. The *mmdd*hist.*nnn* log files record dumps of all MIB values gathered during each poll cycle. For more information about log files, see "Configuring Monitor Agent Log Settings" in "Monitor" in the *GroupWise 2012 Administration Guide*.

The first two digits of the file name represent the month; the next two digits represent the day of the month. The three-digit extension is a sequence number for multiple log files created on the same day. For example, 0518mon.002 is the second Monitor Agent log file created on May 18.

monwork directory

The monwork directory is used for temporary files used in calculating system performance, as described in "Measuring Agent Performance" in "Monitor" in the *GroupWise 2012 Administration Guide*

acct directory

The acct directory is used for storing accounting files received from the GWIA and GroupWise gateways, as described in "Receiving and Forwarding the Accounting Files" in "Monitor" in the *GroupWise 2012 Administration Guide*.

8 Web Application Installation Directories on Your Web Server

WebAccess, the Calendar Publishing Host, and Monitor rely on the presence of a Web server in order to fulfil their functions. A number of Web servers are supported. See "GroupWise System Requirements" in "Installing a Basic GroupWise System" in the *GroupWise 2012 Installation Guide*.

On Linux, the GroupWise Web applications are installed into the Apache Web server and the Tomcat Java Servlet Container. On Windows, the GroupWise Web applications are installed into the Internet Information Server (IIS) and the Tomcat Java Servlet Container. On both platforms, configuration files, log files, and working files, are stored in a separate directory outside of your Web server and Tomcat installations.

- Section 8.1, "OES 11 Apache/Tomcat Installation Directories," on page 109
- Section 8.2, "OES 2 Linux Apache/Tomcat Installation Directories," on page 112
- Section 8.3, "SLES 11 Apache/Tomcat Installation Directories," on page 115
- Section 8.4, "SLES 10 Apache/Tomcat Installation Directories," on page 118
- Section 8.5, "Windows IIS/Tomcat Installation Directories," on page 120
- Section 8.6, "GroupWise Web Application Working Directory," on page 122

8.1 OES 11 Apache/Tomcat Installation Directories

In addition to the files installed into the Apache and Tomcat installation directories, GroupWise Web application files are also installed into the GroupWise Web Application Working Directory.

📁 `/etc/apache2`	Main Apache installation directory
📄 `httpd.conf`	Main Apache Web server configuration file

📁 `/etc/opt/novell/httpd`	Novell Apache installation directory
📁 `conf.d`	Novell Apache application configuration directory
📄 `gw.conf`	WebAccess Application configuration file
📄 `gwcal.conf`	Calendar Publishing Host Application configuration file
📄 `gwmon.conf`	Monitor Application configuration file

📁 `/var/opt/novell/tomcat6`	Novell Tomcat installation directory
📁 `webapps`	Web applications directory

```
    gw                          WebAccess document root directory
       index.html               WebAccess Login page
       webaccess
          yyyymmddhhmm
             help               Directory for WebAccess and Web console help files

    gwcal                       Calendar Publishing Host document root directory
       index.html               Calendar Publishing Host Web page

    gwmon                       GroupWise Monitor document root directory
       index.html               GroupWise Monitor Web console page
       com/novell
          gwmonitor
             help               Directory for Monitor Web console help files

    gw.war                      Web application archive file for WebAccess
    gwcal.war                   Web application archive file for the Calendar Publishing Host
    gwmon.war                   Web application archive file for Monitor


/usr/sbin                       Standard directory for application scripts
   rcapache2                    Script to start and stop the Apache Web server
   rcnovell-tomcat6             Script to start and stop the Novell version of Tomcat
```

8.1.1 /etc/apache2 directory

The /etc/apache2 directory is the standard Apache Web server installation directory on OES 11.

httpd.conf file

The httpd.conf file is the main Apache configuration file.

8.1.2 /etc/opt/novell/httpd directory

The /etc/opt/novell/httpd directory is the Novell Apache Web server installation directory on OES 11.

conf.d directory

The conf.d directory holds configuration information for each Web application that has been installed for use with Apache.

gw.conf file

The gw.conf file includes the information that enables Apache to access the WebAccess Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/webaccess/gw.conf.

gwcal.conf file

The gwcal.conf file includes the information that enables Apache to access the Calendar Publishing Host Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/calhost/gwcal.conf.

gwmon.conf file

The gwmon.conf file includes the information that enables Apache to access the Monitor Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/monitor/gwmon.conf.

8.1.3 /var/opt/novell/tomcat6 directory

The /var/opt/novell/tomcat6 directory is the standard Tomcat installation directory on OES 11.

webapps directory

The webapps directory holds a subdirectory for each GroupWise Web application that is installed on the server.

gw directory

The gw directory holds all WebAccess Application files that are installed into Tomcat.

index.html file

The index.html file in the gw directory is the WebAccess Login page. It is displayed when you go to the following URL:

```
http://web_server_address/gw
```

For more information, see "Accessing Your GroupWise Mailbox in a Web-Based Environment" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

gwcal directory

The gwcal directory holds all Calendar Publishing Host Application files that are installed into Tomcat.

index.html file

The index.html file in the gwcal directory is the default HTML page for the GroupWise Calendar Publishing Host. It is displayed when you go to the following URL:

http://*web_server_address*/gwcal

For more information, see "Calendar Publishing Host" in the *GroupWise 2012 Administration Guide*.

gwmon directory

The gwmon directory holds all Monitor Application files that are installed into Tomcat.

gwmon/index.html file

The index.html file in the gwmon directory is the default HTML page for the GroupWise Monitor Web console. It is displayed when you go to the following URL:

```
http://web_server_address/gwmon
```

For more information, see "Monitor Web Console" in "Monitor" in the *GroupWise 2012 Administration Guide*.

*.war files

The *.war files are Web application archive files, compressed files from which the Web application files are extracted during installation to create the corresponding application subdirectory.

8.1.4 /usr/sbin directory

The /usr/sbin directory is the standard location for application scripts that can be run from any directory on the Linux server. The /usr/sbin directory is always on included in the PATH environment variable. Files in this directory are links to the corresponding script files in /etc/init.d.

8.2 OES 2 Linux Apache/Tomcat Installation Directories

In addition to the files installed into the Apache and Tomcat installation directories, GroupWise Web application files are also installed into the GroupWise Web Application Working Directory.

📁 /etc/apache2	Main Apache installation directory
🗋 httpd.conf	Main Apache Web server configuration file

📁 /etc/opt/novell/httpd	Novell Apache installation directory
📁 conf.d	Novell Apache application configuration directory
🗋 gw.conf	WebAccess Application configuration file
🗋 gwcal.conf	Calendar Publishing Host Application configuration file
🗋 gwmon.conf	Monitor Application configuration file

📁 /var/opt/novell/tomcat5	Novell Tomcat installation directory
📁 webapps	Web applications directory
📁 gw	WebAccess document root directory
🗋 index.html	WebAccess Login page
📁 webaccess	
📁 *yyyymmddhhmm*	
📁 help	Directory for WebAccess and Web console help files
📁 gwcal	Calendar Publishing Host document root directory
🗋 index.html	Calendar Publishing Host Web page

gwmon	GroupWise Monitor document root directory
index.html	GroupWise Monitor Web console page
com/novell	
gwmonitor	
help	Directory for Monitor Web console help files
gw.war	Web application archive file for WebAccess
gwcal.war	Web application archive file for the Calendar Publishing Host
gwmon.war	Web application archive file for Monitor

/usr/sbin	Standard directory for application scripts
rcapache2	Script to start and stop the Apache Web server
rcnovell-tomcat5	Script to start and stop the Novell version of Tomcat

8.2.1 /etc/apache2 directory

The `/etc/apache2` directory is the standard Apache Web server installation directory on OES 2 Linux.

httpd.conf file

The `httpd.conf` file is the main Apache configuration file.

8.2.2 /etc/opt/novell/httpd directory

The `/etc/opt/novell/httpd` directory is the Apache Web server installation directory on OES 2 Linux.

conf.d directory

The `conf.d` directory holds configuration information for each Web application that has been installed for use with Apache.

gw.conf file

The `gw.conf` file includes the information that enables Apache to access the WebAccess Application files that have been installed into Tomcat. This file is a link to `/etc/opt/novell/groupwise/webaccess/gw.conf`.

gwcal.conf file

The `gwcal.conf` file includes the information that enables Apache to access the Calendar Publishing Host Application files that have been installed into Tomcat. This file is a link to `/etc/opt/novell/groupwise/calhost/gwcal.conf`.

gwmon.conf file

The `gwmon.conf` file includes the information that enables Apache to access the Monitor Application files that have been installed into Tomcat. This file is a link to `/etc/opt/novell/groupwise/monitor/gwmon.conf`.

8.2.3 /var/opt/novell/tomcat5 directory

The `/var/opt/novell/tomcat5` directory is the standard Tomcat installation directory on OES 2 Linux.

webapps directory

The `webapps` directory holds a subdirectory for each GroupWise Web application that is installed on the server.

gw directory

The `gw` directory holds all WebAccess Application files that are installed into Tomcat.

index.html file

The `index.html` file in the `gw` directory is the WebAccess Login page. It is displayed when you go to the following URL:

`http://web_server_address/gw`

For more information, see "Accessing Your GroupWise Mailbox in a Web-Based Environment" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

gwcal directory

The `gwcal` directory holds all Calendar Publishing Host Application files that are installed into Tomcat.

index.html file

The `index.html` file in the `gwcal` directory is the default Web page for the GroupWise Calendar Publishing Host. It is displayed when you go to the following URL:

`http://`*`web_server_address`*`/gwcal`

For more information, see "Calendar Publishing Host" in the *GroupWise 2012 Administration Guide*.

gwmon directory

The `gwmon` directory holds all Monitor Application files that are installed into Tomcat.

gwmon/index.html file

The `index.html` file in the `gwmon` directory is the default Web page for the GroupWise Monitor Web console. It is displayed when you go to the following URL:

`http://`*`web_server_address`*`/gwmon`

For more information, see "Monitor Web Console" in "Monitor" in the *GroupWise 2012 Administration Guide*.

*.war files

The `*.war` files are Web application archive files, compressed files from which the Web application files are extracted during installation to create the corresponding application subdirectory.

8.2.4 /usr/sbin directory

The `/usr/sbin` directory is the standard location for application scripts that can be run from any directory on the Linux server. The `/usr/sbin` directory is always on included in the PATH environment variable. Files in this directory are links to the corresponding script files in `/etc/init.d`.

8.3 SLES 11 Apache/Tomcat Installation Directories

In addition to the files installed into the Apache and Tomcat installation directories, GroupWise Web application files are also installed into the GroupWise Web Application Working Directory.

`/etc/apache2`	Apache installation directory
`httpd.conf`	Main Apache Web server configuration file
`conf.d`	Apache application configuration directory
`gw.conf`	WebAccess Application configuration file
`gwcal.conf`	Calendar Publishing Host Application configuration file
`gwmon.conf`	Monitor Application configuration file

`/usr/share/tomcat6`	Tomcat installation directory
`webapps`	Web applications directory
`gw`	WebAccess document root directory
`index.html`	WebAccess Login page
`webaccess`	
yyyymmddhhmm	
`help`	Directory for WebAccess and Web console help files
`gwcal`	Calendar Publishing Host document root directory
`index.html`	Calendar Publishing Host page
`gwmon`	GroupWise Monitor document root directory
`index.html`	GroupWise Monitor Web console page
`com/novell`	
`gwmonitor`	
`help`	Directory for Monitor Web console help files
`gw.war`	Web application archive file for WebAccess
`gwcal.war`	Web application archive file for the Calendar Publishing Host
`gwmon.war`	Web application archive file for Monitor

/usr/sbin	Standard directory for application scripts
rcapache2	Script to start and stop the Apache Web server
rctomcat6	Script to start and stop Tomcat

8.3.1 /etc/apache2 directory

The /etc/apache2 directory is the standard Apache Web server installation directory on SLES 11.

httpd.conf file

The httpd.conf file is the main Apache configuration file.

conf.d directory

The conf.d directory holds configuration information for each Web application that has been installed for use with Apache.

gw.conf file

The gw.conf file provides the information that enables Apache to access the WebAccess Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/webaccess/gw.conf.

gwcal.conf file

The gwcal.conf file provides the information that enables Apache to access the GroupWise Calendar Publishing Host Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/calhost/gwcal.conf.

gwmon.conf file

The gwmon.conf file provides the information that enables Apache to access the Monitor Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/monitor/gwmon.conf.

8.3.2 /usr/share/tomcat6 directory

The /usr/share/tomcat6 directory is the standard Tomcat installation directory on SLES 11.

webapps directory

The webapps directory holds a subdirectory for each GroupWise Web application that is installed on the server.

gw directory

The gw directory holds all WebAccess Application files that are installed into Tomcat.

index.html file

The index.html file in the gw directory is the WebAccess Login page. It is displayed when you go to the following URL:

```
http://web_server_address/gw
```

For more information, see "Accessing Your GroupWise Mailbox in a Web-Based Environment" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

gwcal directory

The gwcal directory holds all Calendar Publishing Host Application files that are installed into Tomcat.

index.html file

The index.html file in the gwcal directory is the default page for the GroupWise Calendar Publishing Host. It is displayed when you go to the following URL:

http://*web_server_address*/gwcal

For more information, see "Calendar Publishing Host" in the *GroupWise 2012 Administration Guide*.

gwmon directory

The gwmon directory holds all Monitor Application files that are installed into Tomcat.

index.html file

The index.html file in the gwmon directory is the default page for the GroupWise Monitor Web console. It is displayed when you go to the following URL:

http://*web_server_address*/gwmon

For more information, see "Monitor Web Console" in "Monitor" in the *GroupWise 2012 Administration Guide*.

*.war files

The *.war files are Web application archive files, compressed files from which the Web application files are extracted during installation to create the corresponding application subdirectory.

8.3.3 /usr/sbin

The /usr/sbin directory is the standard location for application scripts that can be run from any directory on the Linux server. The /usr/sbin directory is always included in the PATH environment variable. Files in this directory are links to the corresponding script files in /etc/init.c.

8.4 SLES 10 Apache/Tomcat Installation Directories

In addition to the files installed into the Apache and Tomcat installation directories, GroupWise Web application files are also installed into the GroupWise Web Application Working Directory.

`/etc/apache2`	Apache installation directory
`httpd.conf`	Main Apache Web server configuration file
`conf.d`	Apache application configuration directory
`gw.conf`	WebAccess Application configuration file
`gwcal.conf`	Calendar Publishing Host Application configuration file
`gwmon.conf`	Monitor Application configuration file

`/srv/www/tomcat5/base`	Tomcat installation directory
`webapps`	Web applications directory
`gw`	WebAccess document root directory
`index.html`	WebAccess Login page
`webaccess`	
yyyymmddhhmm	
`help`	Directory for WebAccess and Web console help files
`gwcal`	Calendar Publishing Host document root directory
`index.html`	Calendar Publishing Host page
`gwmon`	GroupWise Monitor document root directory
`index.html`	GroupWise Monitor Web console page
`com/novell`	
`gwmonitor`	
`help`	Directory for Monitor Web console help files
`gw.war`	Web application archive file for WebAccess
`gwcal.war`	Web application archive file for the Calendar Publishing Host
`gwmon.war`	Web application archive file for Monitor

`/usr/sbin`	Standard directory for application scripts
`rcapache2`	Script to start and stop the Apache Web server
`rctomcat5`	Script to start and stop Tomcat

8.4.1 /etc/apache2 directory

The `/etc/apache2` directory is the standard Apache Web server installation directory on SLES 11.

httpd.conf file

The `httpd.conf` file is the main Apache configuration file.

conf.d directory

The conf.d directory holds configuration information for each Web application that has been installed for use with Apache.

gw.conf file

The gw.conf file provides the information that enables Apache to access the WebAccess Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/webaccess/gw.conf.

gwcal.conf file

The gwcal.conf file provides the information that enables Apache to access the GroupWise Calendar Publishing Host Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/calhost/gwcal.conf.

gwmon.conf file

The gwmon.conf file provides the information that enables Apache to access the Monitor Application files that have been installed into Tomcat. This file is a link to /etc/opt/novell/groupwise/monitor/gwmon.conf.

8.4.2 /srv/www/tomcat5/base directory

The /srv/www/tomcat5/base directory is the standard Tomcat installation directory on SLES 10.

webapps directory

The webapps directory holds a subdirectory for each GroupWise Web application that is installed on the server.

gw directory

The gw directory holds all WebAccess Application files that are installed into Tomcat.

index.html file

The index.html file in the gw directory is the WebAccess Login page. It is displayed when you go to the following URL:

```
http://web_server_address/gw
```

For more information, see "Accessing Your GroupWise Mailbox in a Web-Based Environment" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

gwcal directory

The gwcal directory holds all Calendar Publishing Host Application files that are installed into Tomcat.

index.html file

The `index.html` file in the `gwcal` directory is the default page for the GroupWise Calendar Publishing Host. It is displayed when you go to the following URL:

http://*web_server_address*/gwcal

For more information, see “Calendar Publishing Host” in the *GroupWise 2012 Administration Guide*.

gwmon directory

The gwmon directory holds all Monitor Application files that are installed into Tomcat.

index.html file

The `index.html` file in the `gwmon` directory is the default page for the GroupWise Monitor Web console. It is displayed when you go to the following URL:

http://*web_server_address*/gwmon

For more information, see “Monitor Web Console” in “Monitor” in the *GroupWise 2012 Administration Guide*.

*.war files

The `*.war` files are Web application archive files, compressed files from which the Web application files are extracted during installation to create the corresponding application subdirectory.

8.4.3 /usr/sbin

The `/usr/sbin` directory is the standard location for application scripts that can be run from any directory on the Linux server. The `/usr/sbin` directory is always included in the PATH environment variable. Files in this directory are links to the corresponding script files in `/etc/init.c`.

8.5 Windows IIS/Tomcat Installation Directories

In addition to the files installed into the Apache and Tomcat installation directories, GroupWise Web application files are also installed into the GroupWise Web Application Working Directory.

📁 `c:/inetpub/wwwroot`	IIS directory for publishing content to the Web
📁 `c:/novell/tomcat6`	Tomcat installation directory
📁 `webapps`	Web applications directory
📁 `gw`	WebAccess document root directory
🗋 `index.html`	WebAccess Login page
📁 `webaccess`	
📁 *yyyymmddhhmm*	
📁 `help`	Directory for WebAccess and Web console help files
📁 `gwcal`	Calendar Publishing Host document root directory
🗋 `index.html`	Calendar Publishing Host Web services page

gwmon	GroupWise Monitor document root directory
index.html	GroupWise Monitor Web services page
com/novell	
gwmonitor	
help	Directory for Monitor Web console help files
gw.war	Web application archive file for WebAccess
gwcal.war	Web application archive file for the Calendar Publishing Host
gwmon.war	Web application archive file for Monitor

8.5.1 c:/inetpub/wwwroot directory

The `/inetpub/wwwroot` directory is the standard Internet Information Services (IIS) installation directory.

8.5.2 c:/novell/tomcat6 directory

The `/novell/tomcat6` directory is where the GroupWise Installation program installs Tomcat for use with IIS.

webapps directory

The `webapps` directory holds a subdirectory for each GroupWise Web application that is installed on the server.

gw directory

The `gw` directory holds all WebAccess Application files that are installed into Tomcat.

index.html file

The `index.html` file in the `gw` directory is the WebAccess Login page. It is displayed when you go to the following URL:

```
http://web_server_address/gw
```

For more information, see "Accessing Your GroupWise Mailbox in a Web-Based Environment" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

gwcal directory

The `gwcal` directory holds all Calendar Publishing Host Application files that are installed into Tomcat.

index.html file

The `index.html` file in the `gwcal` directory is the default page for the GroupWise Calendar Publishing Host. It is displayed when you go to the following URL:

```
http://web_server_address/gwcal
```

For more information, see "Calendar Publishing Host" in the *GroupWise 2012 Administration Guide*.

gwmon directory

The gwmon directory holds all Monitor Application files that are installed into Tomcat.

index.html file

The `index.html` file in the `gwmon` directory is the default page for the GroupWise Monitor Web console. It is displayed when you go to the following URL:

```
http://web_server_address/gwmon
```

For more information, see “Monitor Web Console” in “Monitor” in the *GroupWise 2012 Administration Guide*.

*.war files

The `*.war` files are Web application archive files, compressed files from which the Web application files are extracted during installation to create the corresponding application subdirectory.

8.6 GroupWise Web Application Working Directory

📁 /var/opt/novell/groupwise	Directory for GroupWise Web application files on Linux
📁 c:\novell\groupwise	Directory for GroupWise Web application files on Windows
📁 webaccess	Subdirectory for WebAccess
🗋 webacc.cfg	WebAccess Application configuration file
🗋 customization.cfg	WebAccess interface customization configuration file
🗋 ldap.cfg	Access control configuration file
🗋 gwac.xml	
📁 logs	Directory for WebAccess Application log files
📁 users	Directory for WebAccess users’ session files
📁 calhost	Subdirectory for the Calendar Publishing Host
🗋 calhost.cfg	Calendar Publishing Host configuration file
📁 logs	Directory for Calendar Publishing Host log files
📁 monitor	Subdirectory for GroupWise Monitor
🗋 gwmonitor.cfg	Monitor Application configuration file
📁 logs	Directory for Monitor Application log files

8.6.1 /var/opt/novell/groupwise directory

The `/var/opt/novell/groupwise` directory holds a subdirectory for each GroupWise Web application that is installed on the Linux server.

The `c:\Novell\GroupWise` directory holds a subdirectory for each GroupWise Web application that is installed on the Windows server.

webaccess directory

The webaccess directory is created when you install WebAccess. It holds files used by the WebAccess Application.

webacc.cfg file

The webacc.cfg file holds WebAccess configuration information. For more information, see "Customizing the WebAccess Application" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

customization.cfg file

The customization.cfg file holds configuration information for customizing the logos that appear in the WebAccess interface. For more information, see "Customizing the WebAccess User Interface with Your Company Logo" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

ldap.cfg file

The ldap.cfg file holds LDAP configuration information. For more information, see "Enabling an LDAP Address Book" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

gwac.xml file

The gwac.xml file holds WebAccess access control configuration information. For more information, see "Controlling WebAccess Usage" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

logs directory

The logs subdirectory holds WebAccess Application log files. For more information, see "Configuring WebAccess Application Log Settings" in "WebAccess" in the *GroupWise 2012 Administration Guide*.

users directory

The users subdirectory holds session files for WebAccess users.

If WebAccess times out after a period of user inactivity, the user's session information is saved. When the user logs back in, the session information is retrieved so that the user can continue working without loss of data. Also, users' message text is saved during each session, so that if the WebAccess Application is restarted or goes down, users do not lose the message text they were composing at the time.

calhost directory

The calhost directory is created when you install the Calendar Publishing Host. It holds files that are used by the Calendar Publishing Host Application.

calhost.cfg file

The calhost.cfg file holds Calendar Publishing Host configuration information. For more information, see "Configuring the Calendar Publishing Host" in "Calendar Publishing Host" in the *GroupWise 2012 Administration Guide*.

logs directory

The `logs` subdirectory holds Calendar Publishing Host Application log files. For more information, see "Using Calendar Publishing Host Log Files" in "Calendar Publishing Host" in the *GroupWise 2012 Administration Guide*.

monitor directory

The `monitor` directory is created when you install Monitor. It holds files that are used by the Monitor Application.

gwmonitor.cfg file

The `gwmonitor.cfg` file holds Monitor Application configuration information For more information, see "Configuring the Monitor Application" in "Monitor" in the *GroupWise 2012 Administration Guide*.

logs directory

The `logs` subdirectory holds Monitor Application log files. For more information, see "Configuring Monitor Application Log Settings" in "Monitor" in the *GroupWise 2012 Administration Guide*.

9 GroupWise Software Distribution Directory

- Section 9.1, “Linux Software Distribution Directory,” on page 125
- Section 9.2, “Windows Software Distribution Directory,” on page 132

9.1 Linux Software Distribution Directory

/opt/novell/groupwise/software	Master copy of GroupWise software
install	GroupWise Installation script
agents	GroupWise agent software
linux	Subdirectory for Linux software
*.rpm	GroupWise agent and High Availability packages
startups	Subdirectory for startup files
language	Language-specific files
strtup*xx*.mta	MTA startup file
strtup*xx*.poa	POA startup file
strtup.dva	DVA startup file
grpwise	GroupWise agent startup script (MTA, POA, DVA, and GWIA)
snmpmibs	Subdirectory for GroupWise agent MIB files
domain	Domain data dictionary files
wpdomain.dc	Data dictionary for GW 4.*x* domain databases
gwdom.dc	Data dictionary for GW 2012, 8, 7, 6.*x*, and 5.*x* domain databases
wphost.dc	Data dictionary for GW 4.*x* post office databases
gwpo.dc	Data dictionary for GW 2012, 8, 7, 6.*x*, and 5.*x* post office databases
po	Post office data dictionary files
ngwguard.dc	Data dictionary for message store databases
wphost.dc	Data dictionary for GW 4.*x* post office databases
gwpo.dc	Data dictionary for GW 2012, 8, 7, 6.*x*, and 5.*x* post office databases
admin	GroupWise administrator software
*.rpm	GroupWise snap-ins and utilities packages

monitor	GroupWise Monitor software
linux	Subdirectory for Linux software
*.rpm	Monitor Agent and Application packages
startup	Subdirectory for startup files
grpwise-ma	Monitor startup script
utility	GroupWise utility software
setupip	TCP/IP Setup utility
internet	Internet connectivity software
gwia	GWIA software
linux	Subdirectory for Linux software
*.rpm	GWIA and GroupWise High Availability packages
root	Subdirectory for boilerplate supporting files
startup	Subdirectory for startup files
grpwise	GWIA startup script
webacces	GroupWise WebAccess software
linux	Subdirectory for Linux software
*.rpm	GroupWise WebAccess Application package
calhost	GroupWise Calendar Publishing Host software
linux	Subdirectory for Linux software
*.rpm	GroupWise Calendar Publishing Host Application package
client	GroupWise Windows client software
license	GroupWise Software License Agreement in multiple languages
docs	GroupWise pointer Readme
gwinst	Directories and files used by the Installation program
clusterimport.conf	Cluster configuration file

9.1.1 /opt/novell/groupwise/software directory

The GroupWise software distribution directory resides initially in the *GroupWise 2012* software image. GroupWise Administration is installed directly from the original image. In addition, during installation, you can create a software distribution directory on your network from which you subsequently install the GroupWise agents and GroupWise client software.

The default software distribution directory is `/opt/novell/groupwise/software`.

install file

The `install` file is the script you use to install all components of GroupWise on Linux. It starts the Linux GroupWise Installation program. See "Linux: Setting Up a Basic GroupWise System" in the *GroupWise 2012 Installation Guide*.

9.1.2 agents directory

The agents subdirectory contains all files associated with GroupWise agents:

- Post Office Agent (POA)
- Message Transfer Agent (MTA)

*.rpm files

- The novell-groupwise-agents.*version*.*build*.i586.rpm file is the GroupWise agent package that is installed by the Installation program. It installs the MTA and POA. You can install the agent package manually if necessary. To see what files are installed by the package, see Section 7.1.1, “Linux MTA, POA, and DVA Installation Directory,” on page 87.
- The novell-groupwise-dva.*version*.*build*.i586.rpm file is the GroupWise Document Viewer Agent (DVA) package that is installed by the Installation program. The DVA is installed along with the MTA and POA. You can install the DVA package manually if necessary. To see what files are installed by the package, see Section 7.1.1, “Linux MTA, POA, and DVA Installation Directory,” on page 87.
- The novell-groupwise-gwha.*version*.*build*.i586.rpm file is the GroupWise High Availability service package that is installed by the Installation program. It is automatically installed along with the MTA, POA, and GWIA. You can install the High Availability package manually if necessary. To see what files are installed by the package, see Section 7.1.1, “Linux MTA, POA, and DVA Installation Directory,” on page 87.

startups directory

The startups subdirectory contains the default startup files for the GroupWise agents. During installation, a customized startup file is created for each agent that includes the location of the domain or post office serviced by that agent. The customized MTA and POA startup files are named after the domain or post office for which they are created. See Section 7.1.1, “Linux MTA, POA, and DVA Installation Directory,” on page 87.

grpwise file

The grpwise file is the GroupWise agent startup script. It is installed in /etc/init.d and in /opt/novell/groupwise/agents/bin. If you choose to have the agents start automatically when the server reboots, the Installation program places symbolic links to it named S*nn*grpwise in the rc3.d and rc5.d subdirectories of /etc/init.d. For more information, see “Installing and Starting the Linux GroupWise Agents” in “Installing a Basic GroupWise System” in the *GroupWise 2012 Installation Guide*.

snmpmibs directory

The snmpmibs directory contains Management Information Base (MIB) files for the GroupWise agents. SNMP-enabled agents can be monitored and managed using an SNMP management program. See the following sections in the *GroupWise 2012 Administration Guide*:

- “Using an SNMP Management Console” in “Monitoring the POA”
- “Using an SNMP Management Console” in “Monitoring the MTA”
- “Using an SNMP Management Console” in “Monitoring the GWIA”

9.1.3 domain directory

The domain subdirectory contains the files from which domains are created.

wpdomain.dc file

The wpdomain.dc file is the distribution copy of the data dictionary for rebuilding GroupWise 4.*x* domain databases (wpdomain.db files) in secondary domains.

If the wpdomain.dc file is missing from the primary domain, you cannot rebuild GroupWise 4.*x* secondary domains. The original wpdomain.dc file is located in the domain directory of the GroupWise software image.

Historical Note: WordPerfect Office (WP Office), the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Hence, the wp in wpdomain.dc. Some naming conventions were originally preserved for backward compatibility.

gwdom.dc file

The gwdom.dc file is the distribution copy of the data dictionary for creating and rebuilding GroupWise 2012, 8, 7, 6.*x*, and 5.*x* domain databases (wpdomain.db files) in secondary domains.

If the gwdom.dc file is missing from the primary domain, you cannot create or rebuild GroupWise 2012, 8, 7, 6.*x*,and 5.*x* secondary domains. The original gwdom.dc file is located in the domain subdirectory of the GroupWise software image.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Hence, the wp in wpdomain.db. Some naming conventions were originally preserved for backward compatibility.

wphost.dc file

The wphost.dc file is the distribution copy of the data dictionary for rebuilding GroupWise 4.*x* post office databases (wphost.db files).

If the wphost.dc file is missing from a domain, you cannot rebuild GroupWise 4.*x* post offices in that domain. The original wphost.dc file is located in the domain subdirectory of the GroupWise software image. There is also a copy in the po subdirectory.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Post offices were originally called hosts. Hence, the name wphost.dc. Some naming conventions were originally preserved for backward compatibility.

gwpo.dc file

The gwpo.dc file is the distribution copy of the data dictionary for creating and rebuilding GroupWise 2012, 8, 7, 6.*x*, and 5.*x* post office databases (wphost.db files).

If the gwpo.dc file is missing from a domain, you cannot create or rebuild GroupWise 2012, 8, 7, 6.*x*, and 5.*x* post offices in that domain. The original gwpo.dc file is located in the domain directory of the GroupWise software image. There is also a copy in the po directory.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Post offices were originally called hosts. Hence, the name wphost.db. Some naming conventions were originally preserved for backward compatibility.

9.1.4 po directory

The po subdirectory contains the files from which post offices are created.

ngwguard.dc file

The ngwguard.dc file is the distribution copy of the data dictionary for building the following databases in the post office:

- ngwguard.db (guardian database)
- dm*xxnn*01-*FF* (document management databases)
- msg*nnn*.db (message databases)
- user*xxx*.db (user databases)
- pu*xxxxx*.db (databases for replicated items like shared folders)

If the ngwguard.dc file is missing from a post office, new databases cannot be created in the post office, so the post office cannot grow. The original ngwguard.dc file is located in the po directory of the GroupWise software image.

In Remote mode, the GroupWise client also uses the ngwguard.dc file as the data dictionary for its local databases.

9.1.5 admin directory

The admin subdirectory contains subdirectories for administrative tools that can be used with GroupWise.

*.rpm files

- The novell-groupwise-admin.*version*.build.i586.rpm file is the GroupWise administrator package that is installed by the Installation program. It contains the GroupWise Administrator snap-ins to ConsoleOne. You can install the admin package manually if necessary.
- The novell-groupwise-dbcopy.*version*.*build*.i586.rpm file is the GroupWise Database Copy utility package. It is not installed by the Installation program but must be installed manually, as described in "Using DBCopy on Linux" in "Databases" in the *GroupWise 2012 Administration Guide*.
- The novell-groupwise-gwcheck.*version*.*build*.i586.rpm file is the GroupWise Check utility package. It is not installed by the Installation program but must be installed manually, as described in "Using GWCheck on Linux" in "Databases" in the *GroupWise 2012 Administration Guide*.

monitor directory

The monitor subdirectory contains the GroupWise Monitor software. See "Monitor" in the *GroupWise 2012 Administration Guide*.

*.rpm files

- The novell-groupwise-gwmon.*version*.*build*.i586.rpm file is the GroupWise Monitor Agent package that is installed by the Installation program. You can install the package manually if necessary. To see what files are installed by the package, see Section 7.3, "Monitor Agent Installation," on page 103 and Section 8, "Web Application Installation Directories on Your Web Server," on page 109.
- The novell-groupwise-monitor.*version*.*build*.i586.rpm file is the GroupWise Monitor Application package that is installed by the Installation program. You can install the package manually if necessary. To see what files are installed by the package, see Section 7.3, "Monitor Agent Installation," on page 103 and Section 8, "Web Application Installation Directories on Your Web Server," on page 109.

grpwise-ma file

The grpwise-ma file is the Monitor Agent startup script. During installation, it is placed in /etc/init.d. For more information, see "Starting the Linux Monitor Agent as a Daemon" in "Installing GroupWise Monitor" in the *GroupWise 2012 Installation Guide*.

utility directory

The utility subdirectory contains a subdirectory for the TCP/IP Setup utility for use in installing the Windows client software from a GroupWise installation on Linux. See "Using GroupWise AutoUpdate and SetupIP to Distribute the GroupWise Windows Client" in "Client" in the *GroupWise 2012 Administration Guide*.

9.1.6 internet directory

The internet subdirectory contains subdirectories for GroupWise components that provide and support Internet connectivity.

gwia directory

The gwia subdirectory contains the GWIA software, used to connect GroupWise systems across the Internet and to allow GroupWise users to exchange email with users of various Internet email programs. See "Internet Agent" in the *GroupWise 2012 Administration Guide*.

*.rpm files

- The novell-groupwise-gwia.*version*.*build*.i586.rpm file is the GWIA package that is installed by the Installation program. You can install the package manually if necessary. To see what files are installed by the package, see Section 7.2, "Internet Agent Installation," on page 98.
- The novell-groupwise-gwha.*version*.*build*.i586.rpm file is the GroupWise High Availability service package that is installed by the Installation program. It is automatically installed along with the GWIA. You can install the High Availability package manually if necessary. To see what files are installed by the package, see Section 7.2.1, "Linux Internet Agent Installation Directory," on page 98.

root directory

The root directory contains boilerplate versions of the configuration files that are installed to the GWIA root directory under the domain directory. For more information, see Section 6.4, "Internet Agent Queue Directory," on page 74.

grpwise file

The grpwise file is the GWIA startup script. During installation, it is placed in /etc/init.d. If you choose to have the GWIA start automatically when the server reboots, the Installation program places symbolic links to it named S*nn*grpwise in the rc3.d and rc5.d subdirectories of /etc/init.d. For more information, see “Linux: Starting the GWIA” in “Installing the GroupWise Internet Agent” in the *GroupWise 2012 Installation Guide*.

webacces directory

The webacces subdirectory contains the GroupWise WebAccess software, which allows users to access their GroupWise mailboxes from a Web browser. See “WebAccess” in the *GroupWise 2012 Administration Guide*.

*.rpm files

- The novell-groupwise-webaccess.*version*.*build*.i586.rpm file is the WebAccess Application package that is installed by the Installation program. You can install the package manually if necessary. To see what files are installed by the package, seeChapter 8, “Web Application Installation Directories on Your Web Server,” on page 109.

calhost directory

The calhost subdirectory contains the GroupWise Calendar Publishing Host software, which allows users to access published GroupWise calendars from a Web browser. See “Calendar Publishing Host” in the *GroupWise 2012 Administration Guide*.

*.rpm files

- The novell-groupwise-calhost.*version*.*build*.i586.rpm file is the Calendar Publishing Host Application package that is installed by the Installation program. You can install the package manually if necessary. To see what files are installed by the package, see Chapter 8, “Web Application Installation Directories on Your Web Server,” on page 109.

9.1.7 client directory

The client subdirectory contains all files associated with GroupWise Windows client. See “Client” in the *GroupWise 2012 Administration Guide*.

9.1.8 license directory

The license subdirectory contains the GroupWise Software License Agreement in multiple languages. Use the more command to view the License Agreement.

9.1.9 docs directory

The docs subdirectory contains the GroupWise pointer Readme that links to the GroupWise 2012 Documentation Web site (http://www.novell.com/documentation/groupwise2012).

9.1.10 gwinst directory

The `gwinst` subdirectory contains supporting files and subdirectories used by the Installation program.

clusterimport.conf file

The `clusterimport.conf` file is created by the GroupWise Installation program when you select *Configure GroupWise for Clustering*. It stores your responses from when you installed a GroupWise agent on the preferred cluster node, so that on subsequent nodes, the *Import Clustering Data* option is available. For information about clustering GroupWise on Linux, see "Novell Cluster Services on Linux" in the *GroupWise 2012 Interoperability Guide*.

9.2 Windows Software Distribution Directory

`c:\grpwise\software`	Master copy of GroupWise software
`setup.exe`	GroupWise Installation program
`agents`	GroupWise agent software
`win32`	Windows agent software
`startups`	Agent startup files
`snmpmibs`	GroupWise MIB files
`help`	Agent Web console help files
`domain`	Domain data dictionary files
`wpdomain.dc`	Data dictionary for GroupWise 4.*x* domain databases
`gwdom.dc`	Data dictionary for GroupWise 2012, 8, 7, 6.*x*, and 5.*x* domain databases
`wphost.dc`	Data dictionary for GroupWise 4.*x* post office databases
`gwpo.dc`	Data dictionary for GroupWise 2012, 8, 7, 6.*x*, and 5.*x* post office databases
`po`	Post office data dictionary files
`ngwguard.dc`	Data dictionary for message store databases
`wphost.dc`	Data dictionary for GroupWise 4.*x* post office databases
`gwpo.dc`	Data dictionary for GroupWise 2012, 8, 7, 6.*x*, and 5.*x* post office databases
`admin`	GroupWise administrator software
`c1admin`	GroupWise snap-ins to ConsoleOne
`monitor`	GroupWise Monitor software
`utility`	GroupWise administrative utilities
`tools`	GroupWise tools and utilities
`setupip`	GroupWise TCP/IP Setup utility
`internet`	Internet connectivity software
`gwia`	GroupWise Internet Agent software
`webaccess`	GroupWise WebAccess software
`calhost`	GroupWise Calendar Publishing Host software

client	GroupWise client software
win32	GroupWise client for Windows
setup.exe	GroupWise client installation program
setup.cfg	Configuration file for the AutoUpdate feature
gwcheck	GroupWise Check utility
ofviews	GroupWise client view files
ppforms	GroupWise client day planner forms
uwl	GroupWise client user word lists
jaws	JAWS screen reader software for use with GroupWise
license	GroupWise Software License Agreement in multiple languages
docs	GroupWise pointer Readme

9.2.1 c:\grpwise\software directory

The GroupWise software distribution directory resides initially on the *GroupWise 2012* software image. GroupWise Administration is installed directly from the original media. In addition, during installation, you create a software distribution directory on your network from which you subsequently install the GroupWise agents and GroupWise client software.

The default software distribution directory is `\grpwise\software`.

setup.exe file

The `setup.exe` file is the GroupWise Installation program. You can run it in a software distribution directory to install software from that location.

9.2.2 agents directory

The `agents` subdirectory contains all files associated with GroupWise agents:

- Message Transfer Agent (MTA)
- Post Office Agent (POA)
- Document Viewer Agent (DVA)

win32 directory

The `win32` subdirectory contains the GroupWise agent files installed on Windows servers. See "Windows MTA, POA, and DVA Installation Directory" on page 94.

startups directory

The `startups` subdirectory contains the default startup files for the GroupWise agents. During installation, a customized startup file is created for each agent that includes the location of the domain or post office serviced by that agent. The customized startup files are named after the domain or post office for which they are created. See "Windows MTA, POA, and DVA Installation Directory" on page 94.

snmpmibs directory

The snmpmibs directory contains Management Information Base (MIB) files for the GroupWise agents. SNMP-enabled agents can be monitored and managed using an SNMP management program. See the following sections in the *GroupWise 2012 Administration Guide*:

- "Using an SNMP Management Console" in "Monitoring the POA"
- "Using an SNMP Management Console" in "Monitoring the MTA"
- "Using an SNMP Management Console" in "Monitoring the GWIA"

help directory

The help directory holds the help files that you can view from the agent Web consoles. See "Using the POA Web Console" in "Post Office Agent" and "Using the MTA Web Console" in "Message Transfer Agent" in the *GroupWise 2012 Administration Guide*.

9.2.3 domain directory

The domain subdirectory contains the files from which domains are created.

wpdomain.dc file

The wpdomain.dc file is the distribution copy of the data dictionary for rebuilding GroupWise 4.*x* domain databases (wpdomain.db files) in secondary domains.

If the wpdomain.dc file is missing from the primary domain, you cannot rebuild GroupWise 4.*x* secondary domains. The original wpdomain.dc file is located in the domain directory of the GroupWise software image.

Historical Note: WordPerfect Office (WP Office), the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Hence, the wp in wpdomain.dc. Some naming conventions were originally preserved for backward compatibility.

gwdom.dc file

The gwdom.dc file is the distribution copy of the data dictionary for creating and rebuilding GroupWise 2012, 8, 7, 6.*x*, and GroupWise 5.*x* domain databases (wpdomain.db files) in secondary domains.

If the gwdom.dc file is missing from the primary domain, you cannot create or rebuild GroupWise 2012, 8, 7, 6.*x*, and 5.*x* secondary domains. The original gwdom.dc file is located in the domain subdirectory of the GroupWise software image.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Hence, the wp in wpdomain.db. Some naming conventions were originally preserved for backward compatibility.

wphost.dc file

The wphost.dc file is the distribution copy of the data dictionary for rebuilding GroupWise 4.*x* post office databases (wphost.db files).

If the wphost.dc file is missing from a domain, you cannot rebuild GroupWise 4.*x* post offices in that domain. The original wphost.dc file is located in the domain subdirectory of the GroupWise software image. There is also a copy in the po subdirectory.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Post offices were originally called hosts. Hence, the name wphost.dc. Some naming conventions were originally preserved for backward compatibility.

gwpo.dc file

The gwpo.dc file is the distribution copy of the data dictionary for creating and rebuilding GroupWise 2012 8, 7, 6.*x*, and GroupWise 5.*x* post office databases (wphost.db files).

If the gwpo.dc file is missing from a domain, you cannot create or rebuild GroupWise 2012 8, 7, 6.*x*, and 5.*x* post offices in that domain. The original gwpo.dc file is located in the domain directory of the GroupWise software image. There is also a copy in the po directory.

Historical Note: WP Office, the predecessor of GroupWise, was originally designed by WordPerfect Corporation (WPCorp). Post offices were originally called hosts. Hence, the name wphost.db. Some naming conventions were originally preserved for backward compatibility.

9.2.4 po directory

The po subdirectory contains the files from which post offices are created.

ngwguard.dc file

The ngwguard.dc file is the distribution copy of the data dictionary for building the following databases in the post office:

- ngwguard.db (guardian database)
- dm*xxnn*01-*FF* (document management databases)
- msg*nnn*.db (message databases)
- user*xxx*.db (user databases)
- pu*xxxxx*.db (databases for replicated items like shared folders)

If the ngwguard.dc file is missing from a post office, new databases cannot be created in the post office, so the post office cannot grow. The original ngwguard.dc file is located in the po directory of the GroupWise software image.

In Remote mode, the GroupWise client also uses the ngwguard.dc file as the data dictionary for its local databases.

9.2.5 admin directory

The admin subdirectory contains subdirectories for administrative tools that can be used with GroupWise.

c1admin directory

The c1admin subdirectory contains the GroupWise snap-ins to ConsoleOne. See "ConsoleOne Administration Tool" in "System" in the *GroupWise 2012 Administration Guide*.

monitor directory

The monitor subdirectory contains the GroupWise 2012, 8, 7, 6.*x* Monitor program, an SNMP monitoring program for use with the GroupWise agents. See "Monitor" in the *GroupWise 2012 Administration Guide*.

utility directory

The utility subdirectory contains helpful GroupWise utilities.

tools directory

The tools subdirectory contains the following tools and utilities:

- GroupWise Check utility (gwcheck.exe). See "GroupWise Check" in "Stand-Alone Database Maintenance Programs" in the *GroupWise 2012 Administration Guide*.
- GroupWise Generate CSR utility (gwcsrgen.exe). See "Server Certificates and SSL Encryption" in "Security Administration" in the *GroupWise 2012 Administration Guide*.
- GroupWise Database Copy utility (dbcopy.exe). See "GroupWise Database Copy Utility" in "Stand-Alone Database Maintenance Programs" in the *GroupWise 2012 Administration Guide*.
- GroupWise Tuner utility (gwtuner.exe). See "Using ZENworks Configuration Management to Distribute the GroupWise Windows Client" in "Novell ZENworks" in the *GroupWise 2012 Interoperability Guide*.

setupip directory

The setupip subdirectory contains supporting programs for the TCP/IP Setup utility. See "Using GroupWise AutoUpdate and SetupIP to Distribute the GroupWise Windows Client" in "Client" in the *GroupWise 2012 Administration Guide*.

9.2.6 internet directory

The internet subdirectory contains subdirectories for GroupWise components that provide Internet connectivity.

gwia directory

The gwia subdirectory contains the GWIA software, used to connect GroupWise systems across the Internet and to allow GroupWise users to exchange email with users of various Internet email programs. For information about GWIA files and directories, see Section 6.4, "Internet Agent Queue Directory," on page 74 and Section 7.2, "Internet Agent Installation," on page 98.

For information about the GWIA, see "Internet Agent" in the *GroupWise 2012 Administration Guide*.

webaccess directory

The webaccess subdirectory contains the GroupWise WebAccess software, which allows users to access their GroupWise mailboxes from a Web browser. For information about WebAccess files and directories, see Section 8, "Web Application Installation Directories on Your Web Server," on page 109

For information about WebAccess, see "WebAccess" in the *GroupWise 2012 Administration Guide*.

calhost directory

The calhost subdirectory contains the GroupWise Calendar Publishing Host software, which allows users to access published GroupWise calendars from a Web browser. For information about Calendar Publishing Host files and directories, see Section 8, "Web Application Installation Directories on Your Web Server," on page 109

For information about the Calendar Publishing Host, see "Calendar Publishing Host" in the *GroupWise 2012 Administration Guide*.

9.2.7 client directory

The client subdirectory contains all files associated with GroupWise Windows client. See "Client" in the *GroupWise 2012 Administration Guide*.

win32 directory

The win32 subdirectory contains all GroupWise client files installed for use with Windows. See Section 10.1, "Windows Client," on page 139.

setup.exe file

The setup.exe file is the program GroupWise client users run to install and set up the GroupWise client on their Windows workstations. See "Client" in the *GroupWise 2012 Administration Guide*.

setup.cfg file

The setup.cfg file enables you to automate installation of the GroupWise Windows client so that your users do not need to respond to the Setup program's prompts. For more information, see "Understanding the Setup Configuration File" in "Client" in the *GroupWise 2012 Administration Guide*.

jaws directory

The jaws subdirectory contains the JAWS screen reader software for use with GroupWise. See "Accessibility for People with Disabilities" in the *GroupWise 2012 Windows Client User Guide*.

gwcheck directory

The gwcheck subdirectory contains the GroupWise Check utility that can be made available in the GroupWise client by clicking *Tools* > *Repair Mailbox*. For more information, see "Repairing Your Mailbox" in "Maintaining GroupWise" in the *GroupWise 2012 Windows Client User Guide*.

ofviews directory

The `ofviews` subdirectory contains platform-specific subdirectories of view files for use by the GroupWise client. In addition, the `gwviewxx.ini` and `ofviewxx.ini` files configure custom views on the menus where users select views. The `gwviewxx.ini` file configures GroupWise 2012, 8, 7, 6.*x*, and GroupWise 5.5 views. The `ofviewxx.ini` file configures views from earlier versions of GroupWise.

ppforms directory

The `ppforms` subdirectory contains day planner forms for printing GroupWise calendars and tasks.

uwl directory

The `uwl` directory stores the `wt61xx.uwl` files that are used when users add words to the word list during spell checking.

9.2.8 license directory

The `license` subdirectory contains the GroupWise Software License Agreement in multiple languages. Run `license.exe`, then select a language.

9.2.9 docs directory

The `docs` subdirectory contains language-specific subdirectories for the GroupWise pointer Readme in HTML format. This Readme links you to the GroupWise 2012 Documentation Web site (http://www.novell.com/documentation/groupwise2012).

10 GroupWise Client Installation Directory

- Section 10.1, “Windows Client,” on page 139

10.1 Windows Client

c:\Program Files (x86)\Novell\GroupWise	GroupWise 2012 Windows client installation directory (64-bit machine)
c:\Program Files\Novell\GroupWise	GroupWise 2012 Windows client installation directory (32-bit machine)
grpwise.exe	GroupWise client program
gwdca.exe	Document Converter Agent (DCA) program
notify.exe	GroupWise Notify program
addrbook.exe	GroupWise Address Book program
htrsetup.exe	Hit the Road setup program
gwimpexe.exe	GroupWise E-Mail Importer Utility
gwmailto.exe	Web browser support program
gwreload.exe	Mailbox mode switching program
gwsync.exe	Mailbox synchronization program
ngwguard.dc	Data dictionary for databases
wprof.dc	Data dictionary for Remote Address Book
*.dll	DLL programs to support GroupWise Windows client
*.flt	Graphics filters for Oracle viewers
gwappint.inf	Document application integration file
gwhelp_*xx*.chm	GroupWise Windows client Help file
ofviews	GroupWise view files
win	GroupWise view files for Windows
*.vew	View files
*.ini	View initialization files
ppforms	Day planner forms
*.bfp	Form description files
*.prs	Print resource files
gwcheck	GroupWise Check utility

10.1.1 c:\Program Files\Novell\GroupWise

The GroupWise Windows client installation directory contains all files necessary to run the GroupWise client.

grpwise.exe file

The `grpwise.exe` file is the GroupWise 2012 client program for use with Windows.

gwdca.exe file

The `gwdca.exe` file is the Document Converter Agent (DCA), which converts attached documents to HTML for indexing. Most indexing is performed by the DCA associated with the POA, as described in "Configuring the Document Converter Agent (DCA)" in "Post Office Agent" in the *GroupWise 2012 Administration Guide*. However, when the GroupWise client is not in Online mode, it can perform indexing using the local DCA.

notify.exe file

The `notify.exe` file is the program that alerts you whenever you have incoming items, when your outgoing items are opened, or when you have an upcoming appointment. Notify can alert you in four ways: a sound, a dialog box, a small icon, or by launching an application.

addrbook.exe file

The `addrbook.exe` file is the program that accesses the GroupWise Address Book. The Address Book contains information for all Novell eDirectory users, GroupWise external users, GroupWise external entities, GroupWise email distribution lists, and so on.

htrsetup.exe file

The `htrsetup.exe` file is the setup program for Hit the Road, which creates your Remote mailbox. See "Remote Mode" in "Client" in the *GroupWise 2012 Administration Guide*.

gwimpexe.exe file

The `gwimpexe.exe` file is the GroupWise E-Mail Importer Utility, which enables you to import supported POP3/IMAP4 accounts into GroupWise. See "POP3 and IMAP4 Accounts" in the *GroupWise 2012 Windows Client User Guide*.

gwmailto.exe file

The `gwmailto.exe` file is the program that integrates the GroupWise client with your Web browser. If you select *Internet Browser Mail Integration* during installation, the GroupWise client becomes the default email program on your workstation. See "Installing the GroupWise Client" in the *GroupWise 2012 Installation Guide*.

gwreload.exe file

The gwreload.exe file is the program that restarts the GroupWise client when you switch between Online, Caching, and Remote modes. See “Using Caching Mode” and “Using Remote Mode” in the *GroupWise 2012 Windows Client User Guide*.

gwsync.exe file

The gwsync.exe file is the program that synchronizes your Online and Caching mailboxes. See “Using Caching Mode” in the *GroupWise 2012 Windows Client User Guide*.

ngwguard.dc file

The ngwguard.dc file is the data dictionary for building databases in the post office. In Remote mode, the GroupWise client also uses the ngwguard.dc file as the data dictionary for its local databases.

wprof.dc file

The wprof.dc file is the data dictionary for the remote GroupWise Address Book (wprof.db).

Historical Note: An earlier version of the GroupWise client Remote mode, designed by WordPerfect Corporation (WPCorp), was named WP Office Remote. Hence, the wprof in wprof.dc. Some naming conventions were originally preserved for backward compatibility.

*.dll files

*.dll files are dynamically linked libraries of program code used by executable programs. Language-independent *.dll files are typically located in the same directory with the executable programs they support. Language-specific *.dll files are grouped into subdirectories by language.

*.flt files

*.flt files are graphics filters used by the Oracle viewers incorporated into the GroupWise client. The viewers enable you to view a wide variety of file types from within GroupWise.

gwappint.inf file

The gwappint.inf file controls how document-producing applications are integrated with the GroupWise Windows client. See “Setting Up Integrations Using the gwappint.inf File” in “Libraries and Documents” in the *GroupWise 2012 Administration Guide*.

gwhelp_*xx*.chm file

The gwhelp_xx chm file contain the online documentation for all components of GroupWise. The xx is a language code.

ofviews directory

The ofviews directory contains platform-specific subdirectories of view files for use by the GroupWise client.

Historical Note: An earlier version of GroupWise, designed by WordPerfect Corporation (WPCorp), was named WP Office. Hence, the of in ofviews. Some naming conventions were originally preserved for backward compatibility.

win directory

The win subdirectory contains view (*.vew) files for use by the GroupWise client. It also contains initialization (*.ini) files to control display of views.

ppforms directory

The ppforms directory contains day planner forms for printing GroupWise calendars and tasks.

*.bfp files

*.bfp files are form description files that contain binder, filler, and page attributes for forms.

*.prs files

*.prs files are print resource files that indicate which forms are available for which languages. For example, day planners used by English-speaking people in the United States have different forms from day planners used for various languages in Europe.

gwcheck directory

The gwcheck directory contains the GroupWise Check utility that can be made available in the GroupWise client by clicking *Tools* > *Repair Mailbox*. For more information, see “Repairing Your Mailbox” in “Maintaining GroupWise” in the *GroupWise 2012 Windows Client User Guide*.

III Documentation Updates

This section lists updates to *GroupWise 2012 Troubleshooting 3: Message Flow and Directory Structure* that have been made since the initial release of GroupWise 2012. The information helps you to keep current on documentation updates and, in some cases, software updates (such as a Support Pack release).

The information is grouped according to the date when *GroupWise 2012 Troubleshooting 3: Message Flow and Directory Structure* was republished. Within each dated section, the updates are listed by the names of the main table of contents sections.

GroupWise 2012 Troubleshooting 3: Message Flow and Directory Structure has been updated on the following dates:

- Appendix A, “April 16, 2013 (GroupWise 2012 SP2),” on page 145
- Appendix B, “September 20, 2012 (GroupWise 2012 SP1),” on page 147

A April 16, 2013 (GroupWise 2012 SP2)

Location	Change
Directory Structure Diagrams	
"msgnnn.db file" on page 63	The maximum size for a message database is 4 GB.
"userxxx.db file" on page 63	The maximum size for a user database is 4 GB.

B September 20, 2012 (GroupWise 2012 SP1)

Location	Change
Directory Structure Diagrams	
“customization.cfg file” on page 123	Added the configuration file for customizing the WebAccess user interface.