Novell GroupWise®

7

August 15, 2005

www.novell.com

TROUBLESHOOTING 3: MESSAGE FLOW AND DIRECTORY STRUCTURE

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. Please refer to 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. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.novell.com/company/legal/patents/ and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

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

Novell Trademarks

Third-Party Materials

Contents

About This Guide

This Novell® *GroupWise® Troubleshooting 3 Guide* provides diagrams to help you understand the structure and functioning of your GroupWise system. The guide is divided into the following sections:

- Part I, "Message Flow Diagrams," on page 11
- Part I, "Directory Structure Diagrams," on page 61

Other sources of troubleshooting assistance include:

- Novell Support (http://support.novell.com)
- Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp)
- GroupWise 7 Support Forums (http://support.novell.com/forums/2gw.html)
- Novell GroupWise Support Community (http://support.novell.com/products/groupwise)
- GroupWise Cool Solutions (http://www.novell.com/coolsolutions/gwmag/index.html)

Audience

This guide is intended for *****.

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, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of the *GroupWise 7 Troubleshooting 3 Guide*, visit the GroupWise 7 Documentation Web site (http://www.novell.com/documentation/gw7).

Additional Documentation

For additional GroupWise documentation, see the following guides at the GroupWise 7 Documentation Web site (http://www.novell.com/documentation/gw7):

- *Installation Guide*
- *Administration Guide*
- *Multi-System Administration Guide*
- *Interoperability Guide*
- *Troubleshooting 1: Error Messages*
- *Troubleshooting 2: Solutions to Common Problems*
- *GroupWise Client User Guides*

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (™, ®, etc.) denotes a Novell trademark. An asterisk denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux*, should use forward slashes as required by your software.

When a startup switch can be written with a forward slash for some platforms or a double hyphen for other platforms, the startup switch is presented with a forward slash. Users of platforms that require a double hyphen, such as Linux, should use double hyphens as required by your software.

Message Flow Diagrams

I

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.

Message Delivery in the Local Post Office

1

The GroupWise® 7 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 "Using Different GroupWise Modes (Online, Caching, and Remote)" in "Getting Started" in the *GroupWise 7 Windows Client User Guide*.

- Section 1.1, "Online Mode," on page 13
- Section 1.2, "Caching Mode," on page 15
- Section 1.3, "Remote Mode," on page 18

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.

Figure 1-1 *Message Flow in Online Mode*

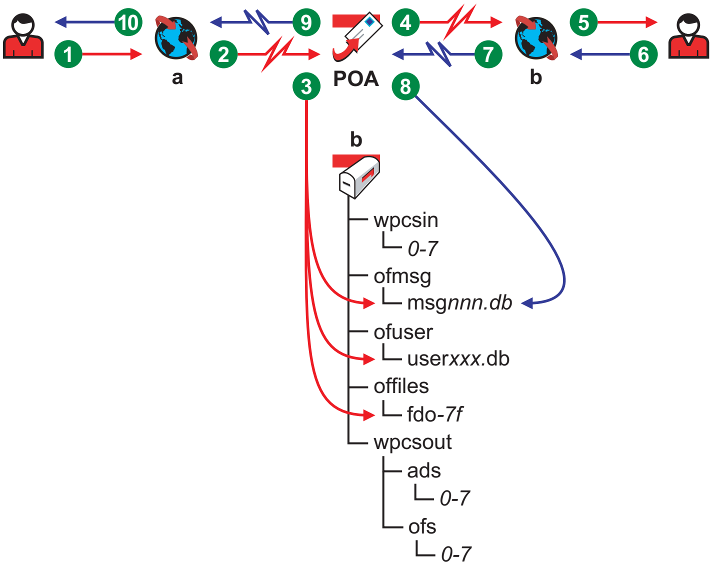

Table 1-1 *Message Flow in Online Mode Stages*

Stage	Icon	Description
1 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.

Stage	Icon	Description
❷ Sender's GroupWise Client	🌐	The GroupWise client communicates with the POA by way of TCP/IP.
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.

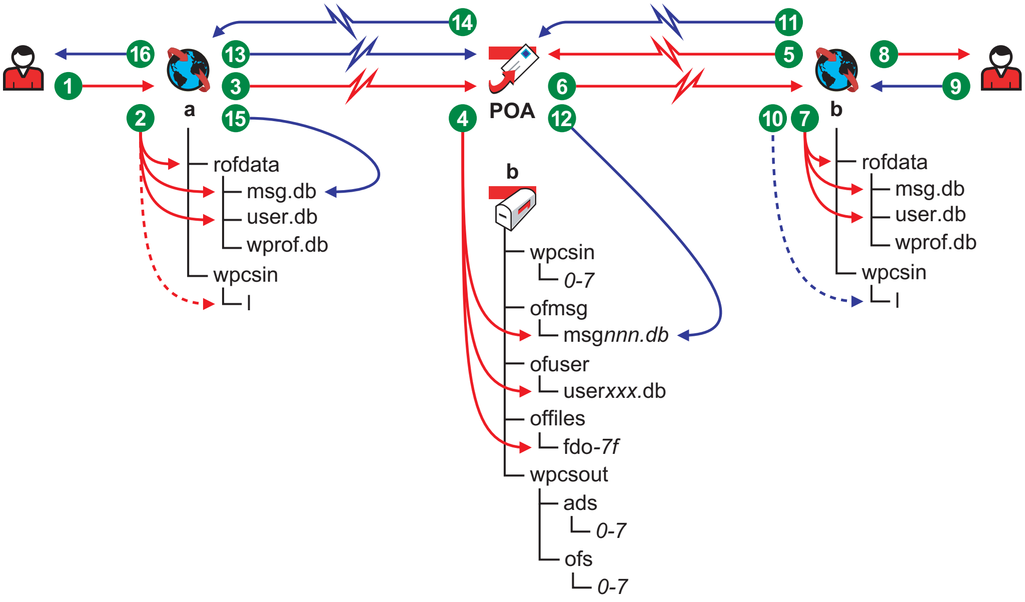

Figure 1-2 *Message Flow in Caching Mode*

Table 1-2 *Message Flow in Caching Mode Stages*

Stage	Icon	Description
1 Sender	👤	The user sends a message to recipients in the same post office. The user is in Caching mode.

Stage	Icon	Description
❷ Sender's GroupWise Client		The GroupWise client updates the sender's Caching mailbox (\novell\groupwise\gw*xxxxxxx*\rofdata) 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.
❸ 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.
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 (msg*nnn*.db) assigned to the sender in the post office. • Creates a pointer in the sender's user database (user*xxx*.db) in the post office. • Places attachments larger than 2 KB in one of the *post_office*\offiles\fd*0-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 (user*xxx*.db) to the message in the message database (msg*nnn*.db) in the post office so that the new message appears in each recipient's mailbox. • Updates the message in the message database (msg*nnn*.db) in the post office with a Delivered status for the recipients.

Stage	Icon	Description
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 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.
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.

Stage	Icon	Description
⑮ 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.
⑯ 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.

1.3 Remote Mode

Before you can use Remote mode, you must create a Remote mailbox on your computer and have a connection to your master GroupWise system. See Chapter 5, "Message Delivery through a Modem Connection," on page 51.

Message Delivery to a Different Post Office

2

The MTA handles message transfer between post offices.

- Section 2.1, "TCP/IP Link Open: Transfer between Post Offices Successful," on page 19
- Section 2.2, "TCP/IP Link Closed: Transfer between Post Offices Delayed," on page 22

The message flow diagrams in *GroupWise 7 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*. For an explanation of link types and link protocols, see "Understanding Link Configuration" in "Domains" in the *GroupWise 7 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.

Figure 2-1 *Message Flow When the TCP/IP Link is Open*

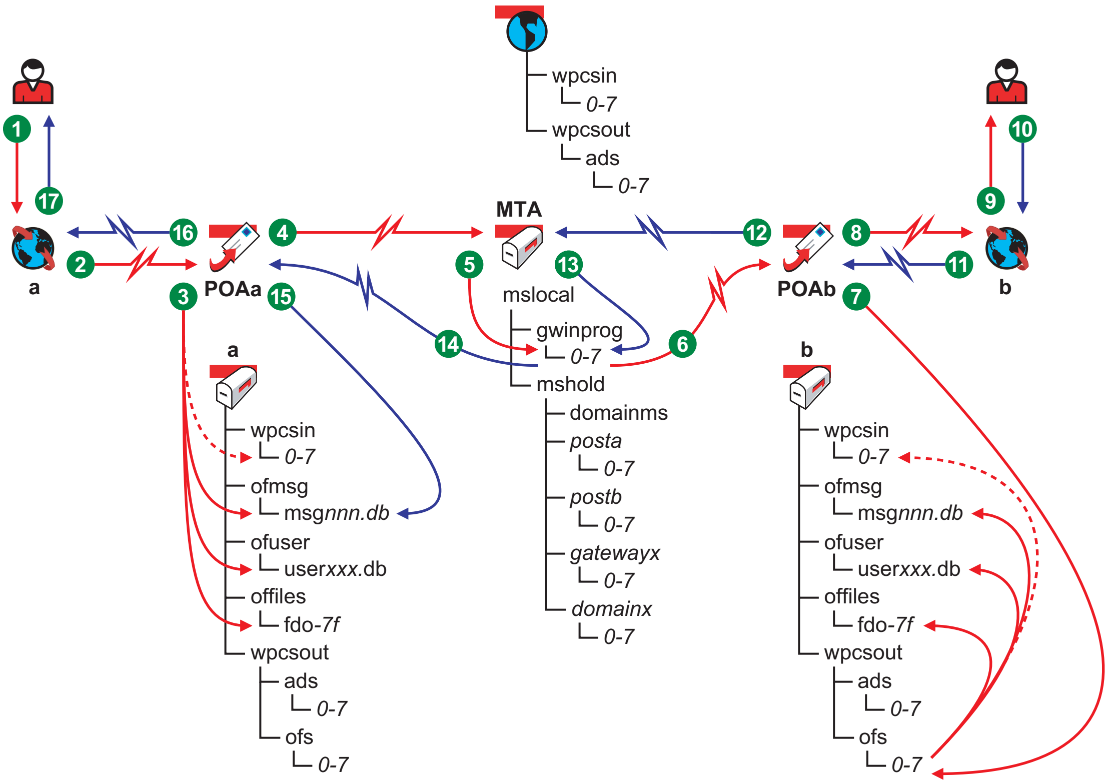

Table 2-1 *Message Flow When the TCP/IP Link is Open Stages*

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 (`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 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 22.
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 30 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 (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. 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.

Stage	Icon	Description
15 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).
16 POA for Local Post Office	POA	The POA communicates the 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 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.

Figure 2-2 *Message Flow When the TCP/IP Link is Closed*

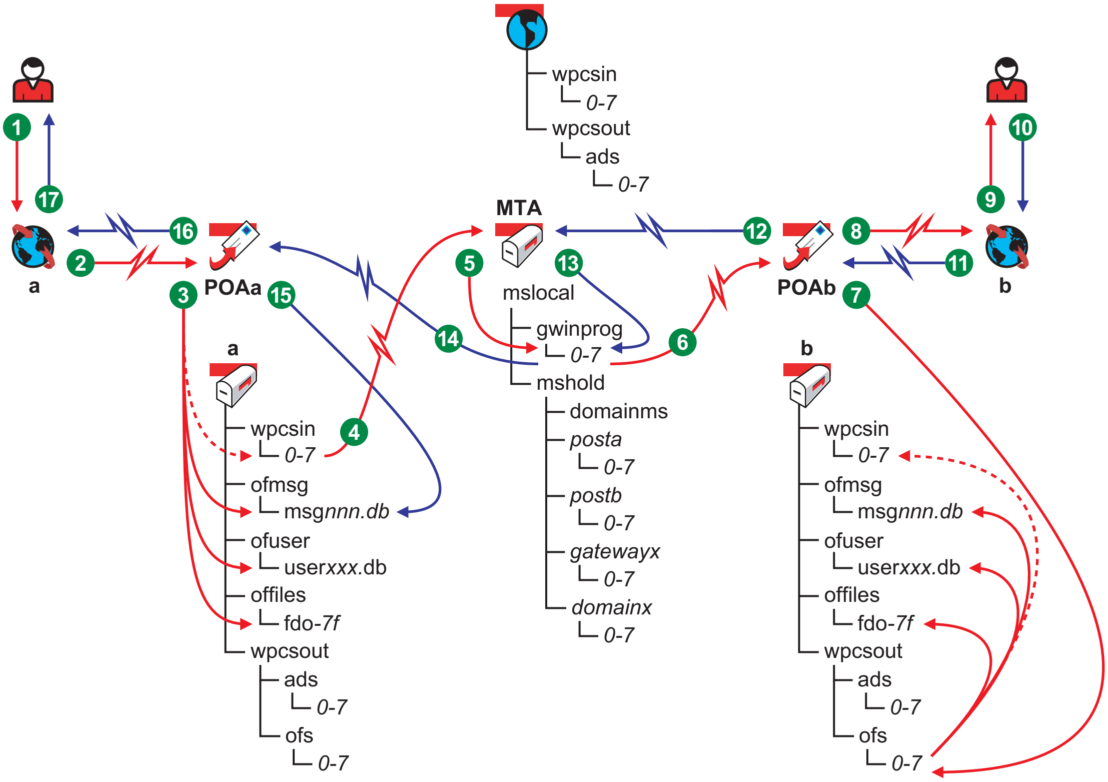

Table 2-2 *Message Flow When the TCP/IP Link is Closed Stages*

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 (`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 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 30.

Stage	Icon	Description
7 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.
8 POA for Local Post Office	POA	The POA communicates to the GroupWise client by way of TCP/IP that a new message has arrived.
9 Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
10 Recipient	👤	Each recipient opens the message in the GroupWise client.
11 Recipient's GroupWise Client	🌐	Each recipient's GroupWise client communicates the Opened status message to the POA by way of TCP/IP.
12 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.
13 MTA for Local Domain	MTA	The MTA places the status message into the MTA "in progress" (gwinprog) queue.
14 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.
15 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).

Stage	Icon	Description
⑯ POA for Local Post Office	POA	The POA communicates the Opened 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 a status of Delivered for each recipient (and possibly Opened as well if the recipient has opened the message).

Message Delivery to a Different Domain

3

The MTA handles message transfer between domains.

- Section 3.1, "TCP/IP Link Open: Transfer between Domains Successful," on page 27
- Section 3.2, "TCP/IP Link Closed: Transfer between Domains Delayed," on page 30

The message flow diagrams in *GroupWise 7 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*. For an explanation of link types and link protocols, see "Understanding Link Configuration" in "Domains" in the *GroupWise 7 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.

Figure 3-1 *Message Flow When the TCP/IP Link is Open*

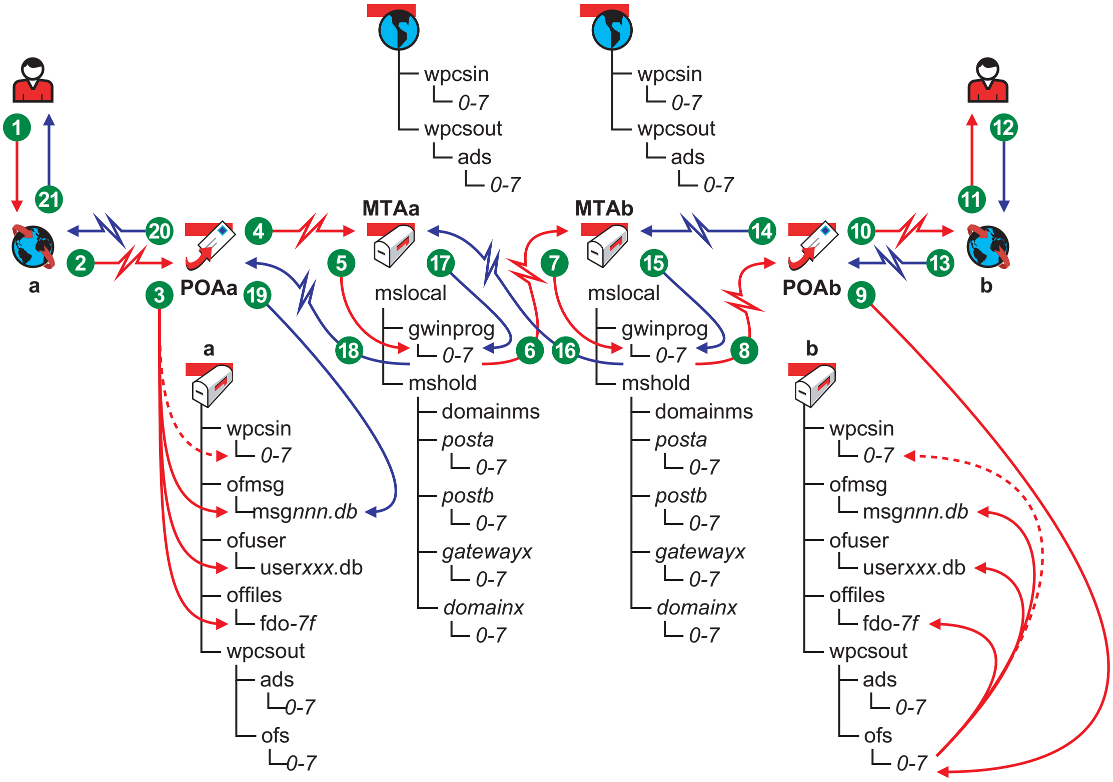

***Table 3-1** Message Flow When the TCP/IP Link is Open Stages*

Stage	Icon	Description
❶ 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.
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 (`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 22.
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 30.
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
⑨ 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.
⑩ 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.
⑪ Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
⑫ Recipient	👤	Each recipient opens the message in the GroupWise client.
⑬ Recipient's GroupWise Client	🌐	Each recipient's GroupWise client communicates the Opened status message to the POA by way of TCP/IP.
⑭ 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.
⑮ MTA for Recipient's Domain	MTA	The MTA for the recipient's domain places the status message into the MTA "in progress" (gwinprog) queue.
⑯ 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.

Stage	Icon	Description
⑰ MTA for Sender’s Domain	MTA	The MTA for the sender’s domain places the status message into the MTA “in progress” (`gwinprog`) queue.
⑱ 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.
⑲ 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).
⑳ 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.
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.

Figure 3-2 *Message Flow When the TCP/IP Link is Closed*

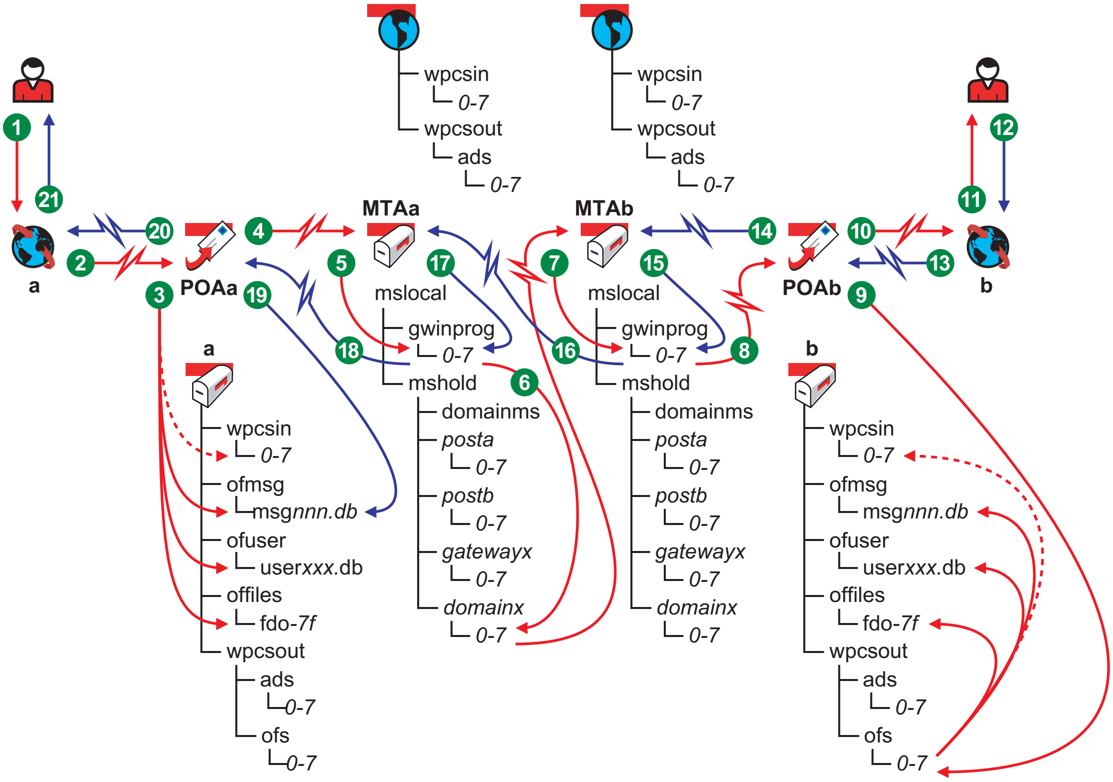

Table 3-2 *Message Flow When the TCP/IP Link is Closed Stages*

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 22.
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.

Stage	Icon	Description
⑰ MTA for Sender’s Domain	MTA	The MTA for the sender’s domain places the status message into the MTA “in progress” (`gwinprog`) queue.
⑱ 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.
⑲ 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).
⑳ 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.
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).

Message Delivery to and from the Internet

4

4.1 TCP/IP Link: 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 Internet Agent and when the Internet Agent can communicate successfully with the Internet host to which the message is addressed.

***Figure 4-1** Message Flow When the TCP/IP Link is Open*

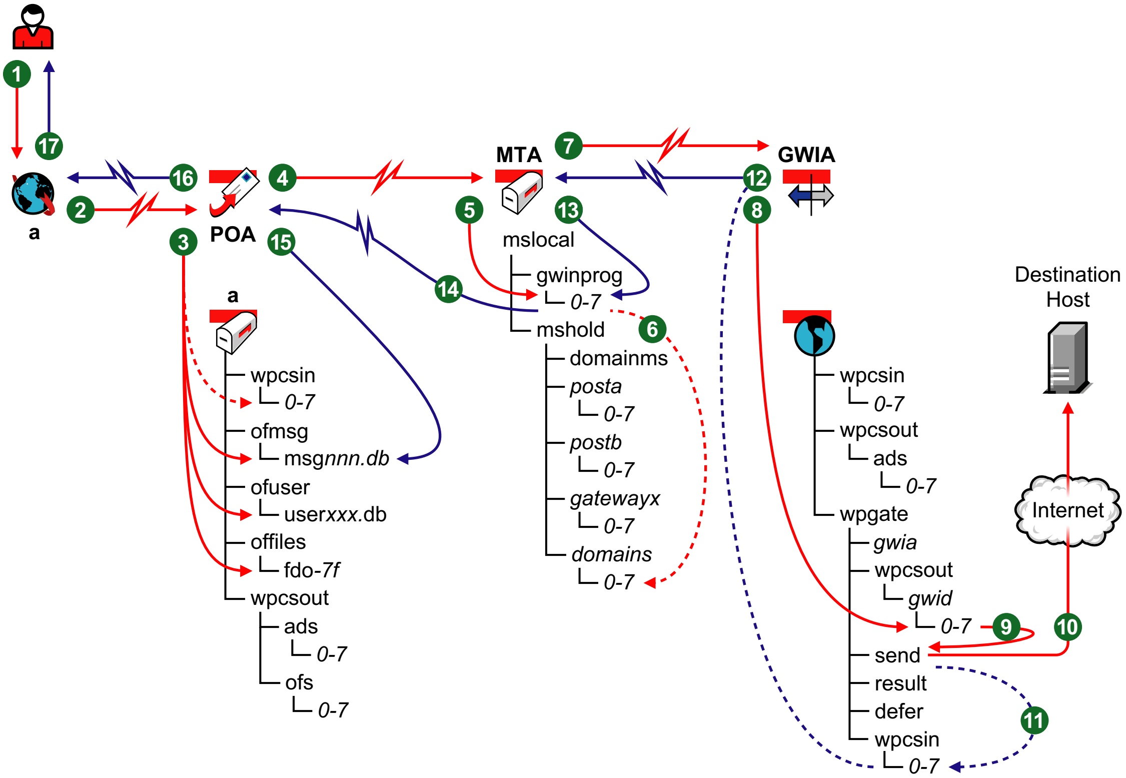

Table 4-1 *Message Flow When the TCP/IP Link is Open Stages*

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 (`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.
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 Internet Agent, the MTA creates a copy of the message in the appropriate priority `0-7` subdirectory of the Internet Agent hold queue (`mslocal\mshold\gatewayx\0-7`), in case the TCP/IP link to the Internet Agent is currently closed.
❼ MTA for Sender's Domain	MTA	The MTA then communicates the message to the Internet Agent for the sender's domain by way of TCP/IP, and deletes the copy in the Internet Agent holding queue because the TCP/IP transfer to the Internet Agent was successful.
❽ Internet Agent for Sender's Domain	GWIA	The Internet Agent 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 Internet Agent input queue.

Stage	Icon	Description
7 Internet Agent for Sender's Domain	GWIA	The Internet Agent scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the Internet Agent object in ConsoleOne®. The Internet Agent picks up the file in binary-encrypted format from the wpcsout\gwid\0-7 directory and converts it. The Internet Agent encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the Internet Agent saves it with S as the first character of the filename and places the message file in the domain\wpgate\gwia\send directory for processing.
10 Internet Agent for Sender's Domain	GWIA	While the Internet Agent is processing the message file in the send directory, it changes the first character of the filename to P. When processing is completed, the Internet Agent sends the message to the destination host across the Internet.
11 Internet Agent for Sender's Domain	GWIA	If the Internet Agent 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 Internet Agent is currently closed.
12 Internet Agent for Sender's Domain	GWIA	The Internet Agent 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.
13 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.
14 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.
15 POA for Sender's Post Office	POA	The POA for the sender's post office updates the sender's message database (msgnnn.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 Internet Agent was able to sent it successfully.

4.2 TCP/IP Link: 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 Internet Agent and when the Internet Agent cannot communicate successfully with the Internet host to which the message is addressed.

***Figure 4-2** Message Flow When the TCP/IP Link is Closed*

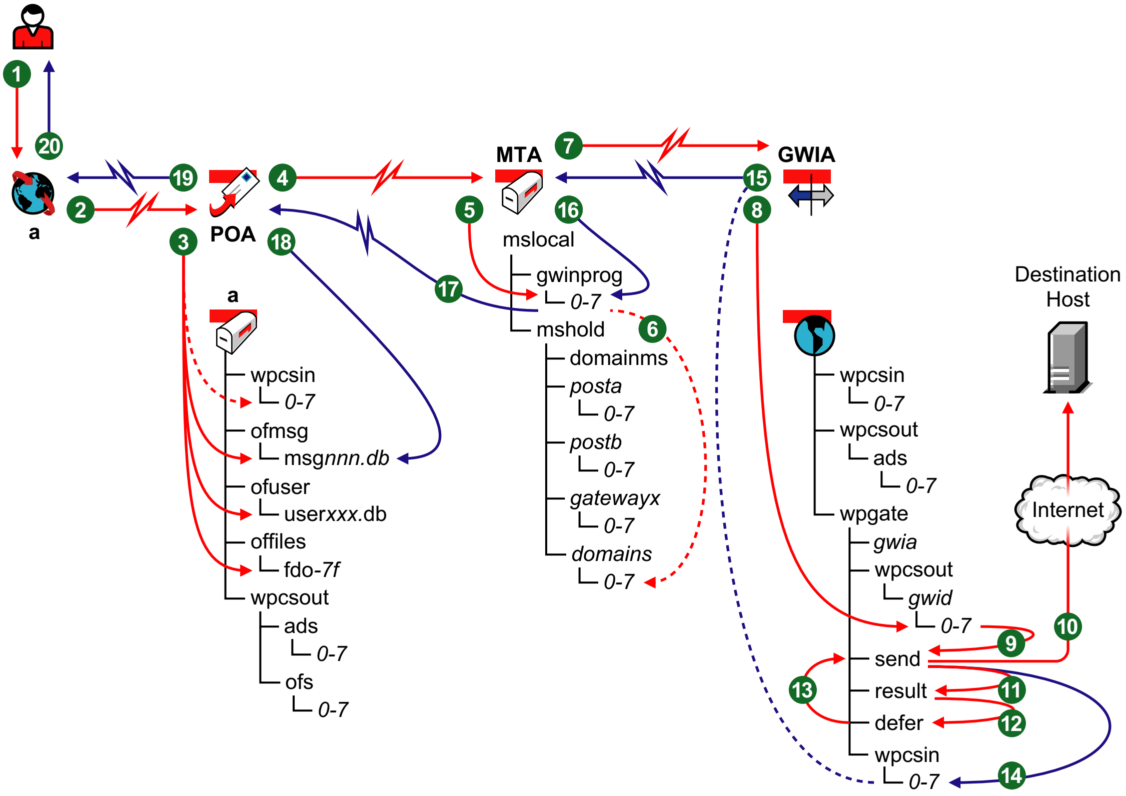

***Table 4-2** Message Flow When the TCP/IP Link is Closed Stages*

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.

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 (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 Internet Agent, the MTA creates a copy of the message in the appropriate priority 0-7 subdirectory of the Internet Agent hold queue (mslocal\mshold\*gatewayx**0-7*), in case the TCP/IP link to the Internet Agent is currently closed.
❼ MTA for Sender's Domain	MTA	The MTA then communicates the message to the Internet Agent for the sender's domain by way of TCP/IP, and deletes the copy in the Internet Agent holding queue because the TCP/IP transfer to the Internet Agent was successful.
❽ Internet Agent for Sender's Domain	GWIA	The Internet Agent 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 Internet Agent input queue.
❾ Internet Agent for Sender's Domain	GWIA	The Internet Agent scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the Internet Agent object in ConsoleOne. The Internet Agent picks up the file in binary-encrypted format from the wpcsout*gwid**0-7* directory and converts it. The Internet Agent encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the Internet Agent saves it with S as the first character of the filename and places the message file in the *domain*\wpgate*gwia*\send directory for processing.

Stage	Icon	Description
⑩ Internet Agent for Sender's Domain	GWIA	While the Internet Agent is processing the message file in the send directory, it changes the first character of the filename to P. When processing is completed, the Internet Agent sends the message to the destination host across the Internet.
⑪ Internet Agent for Sender's Domain	GWIA	If the Internet Agent does not receive a 250 OK SMTP reply code from the destination Internet host, the Internet Agent 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 Internet Agent 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.
⑫ Internet Agent for Sender's Domain	GWIA	The Internet Agent 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 Internet Agent moves the `S*.*` message file to the `defer` directory. Continue with Stage ⑬ If the `R*.*` file has a fatal error (meaning it has a 500-level SMTP reply code such as 550 Host Unknown), the Internet Agent deletes the `S*.*` file because it is undeliverable. Skip to Stage ⑭
⑬ Internet Agent 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 Internet Agent object in ConsoleOne, the Internet Agent requeues the `S *.*` message file back into the send directory for reprocessing.
⑭ Internet Agent 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 Internet Agent deletes all related schedule files from the defer directory because the message is undeliverable. The Internet Agent 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.
⑮ Internet Agent for Sender's Domain	GWIA	The Internet Agent 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.
⑯ 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.
⑰ 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.

Stage	Icon	Description
18 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.
19 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.
20 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 Internet Agent was able to send it successfully.

4.3 Mapped/UNC Link: 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 Internet Agent and when the Internet Agent can communicate successfully with the Internet host to which the message is addressed.

Figure 4-3 *Message Flow When the Mapped Link is Open*

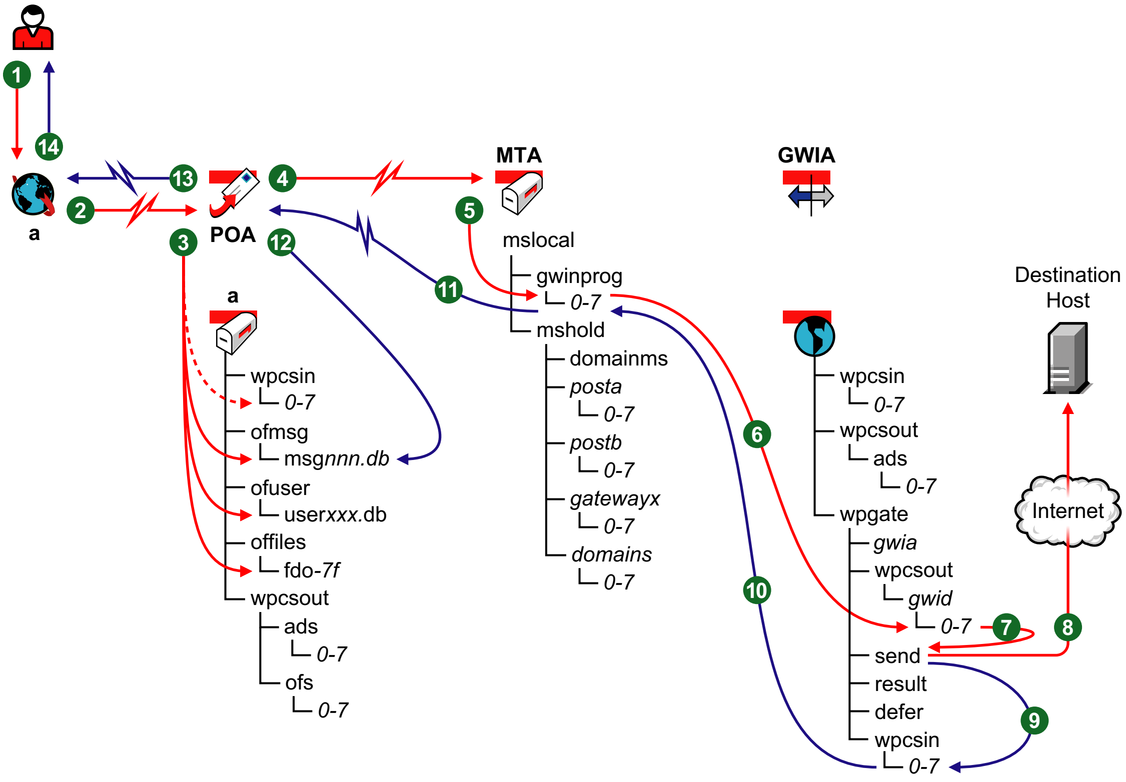

Table 4-3 Message Flow When the Mapped Link is Open Stages

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 mapped/UNC link between the MTA and the Internet Agent, the MTA places the message in its output queue in the Internet Agent's gateway directory (*domain*\wpgate*gwia*\wpcsout*gwid**0-7*).
❼ Internet Agent for Sender's Domain	GWIA	The Internet Agent scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the Internet Agent object in ConsoleOne. The Internet Agent picks up the file in binary-encrypted format from the wpcsout*gwid**0-7* directory and converts it. The Internet Agent encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the Internet Agent saves it with S as the first character of the filename and places the message file in the *domain*\wpgate*gwia*\send directory for processing.

Stage	Icon	Description
8 Internet Agent for Sender's Domain	GWIA	While the Internet Agent is processing the message file in the send directory, it changes the first character of the filename to P. When processing is completed, the Internet Agent sends the message to the destination host across the Internet.
9 Internet Agent for Sender's Domain	GWIA	If the Internet Agent 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.
10 MTA for Sender's Domain	MTA	Because of its mapped/UNC link with the Internet Agent, the MTA regularly scans its input queue in the Internet Agent'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.
11 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.
12 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.
13 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.
14 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 Internet Agent was able to sent it successfully.

4.4 Mapped/UNC Link: 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 Internet Agent and when the Internet Agent cannot communicate successfully with the Internet host to which the message is addressed.

Figure 4-4 Message Flow When the Mapped Link is Closed

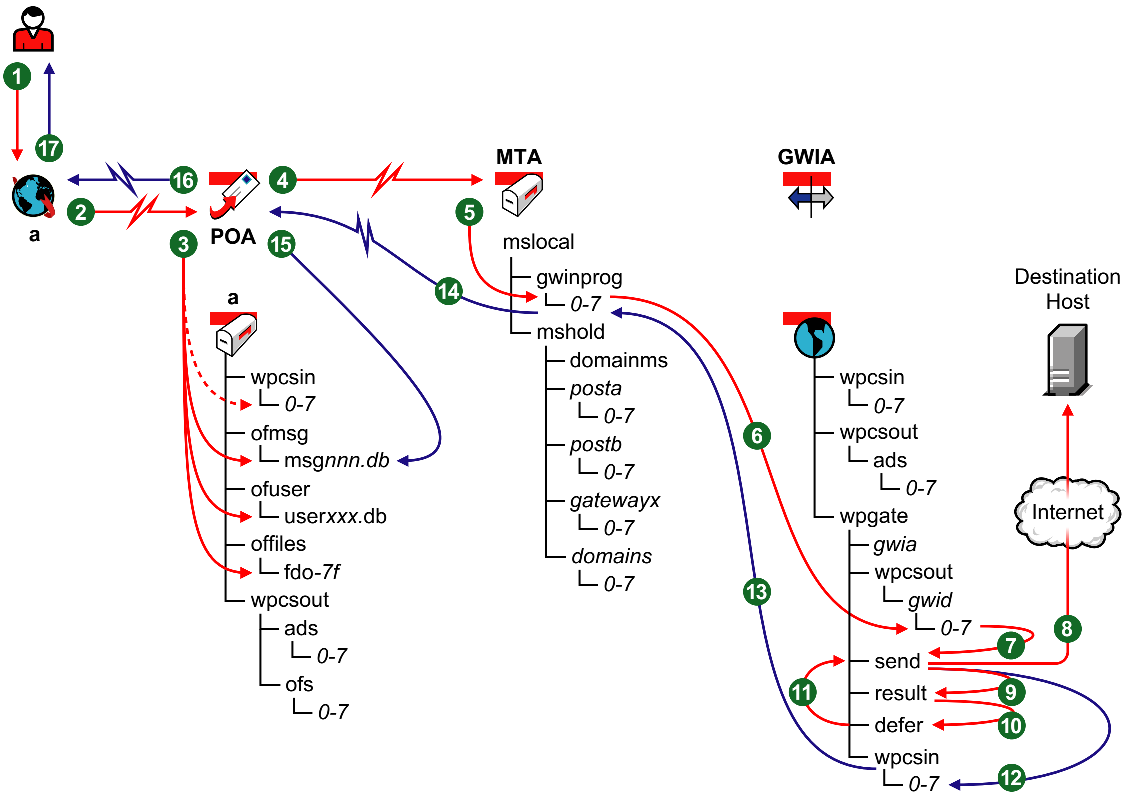

Table 4-4 Message Flow When the Mapped Link is Closed Stages

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.

Stage	Icon	Description
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.
4 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.
5 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.
6 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 Internet Agent, the MTA places the message in its output queue in the Internet Agent's gateway directory (*domain*\wpgate*gwia*\wpcsout*gwid**0-7*).
7 Internet Agent for Sender's Domain	GWIA	The Internet Agent scans its input queues according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the Internet Agent object in ConsoleOne. The Internet Agent picks up the file in binary-encrypted format from the wpcsout*gwid**0-7* directory and converts it. The Internet Agent encodes the message in MIME format with the appropriate encoding scheme. When the message file is built, the Internet Agent saves it with S as the first character of the filename and places the message file in the *domain*\wpgate*gwia*\send directory for processing.
8 Internet Agent for Sender's Domain	GWIA	While the Internet Agent is processing the message file in the send directory, it changes the first character of the filename to P. When processing is completed, the Internet Agent sends the message to the destination host across the Internet.
9 Internet Agent for Sender's Domain	GWIA	If the Internet Agent does not receive a 250 OK SMTP reply code from the destination Internet host, the Internet Agent 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 Internet Agent 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.

Stage	Icon	Description
⑩ Internet Agent for Sender's Domain	GWIA	The Internet Agent 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 Internet Agent moves the `S*.*` message file to the `defer` directory. Continue with Stage ⑪ If the `R*.*` file has a fatal error (meaning it has a 500-level SMTP reply code such as 550 Host Unknown), the Internet Agent deletes the `S*.*` file because it is undeliverable. Skip to Stage ⑫
⑪ Internet Agent 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 Internet Agent object in ConsoleOne, the Internet Agent requeues the `S *.*` message file back into the send directory for reprocessing.
⑫ Internet Agent 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 Internet Agent deletes all related schedule files from the defer directory because the message is undeliverable. The Internet Agent then creates an Undeliverable status message for the MTA to pick up and return to the sender.
⑬ MTA for Sender's Domain	MTA	Because of its mapped/UNC link with the Internet Agent, the MTA scans its input queue in the Internet Agent'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 Internet Agent was able to send it successfully.

4.5 TCP/IP Link: 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 Internet Agent and the MTA for the recipient's domain is a TCP/IP link.

Figure 4-5 *Message Flow in From the Internet*

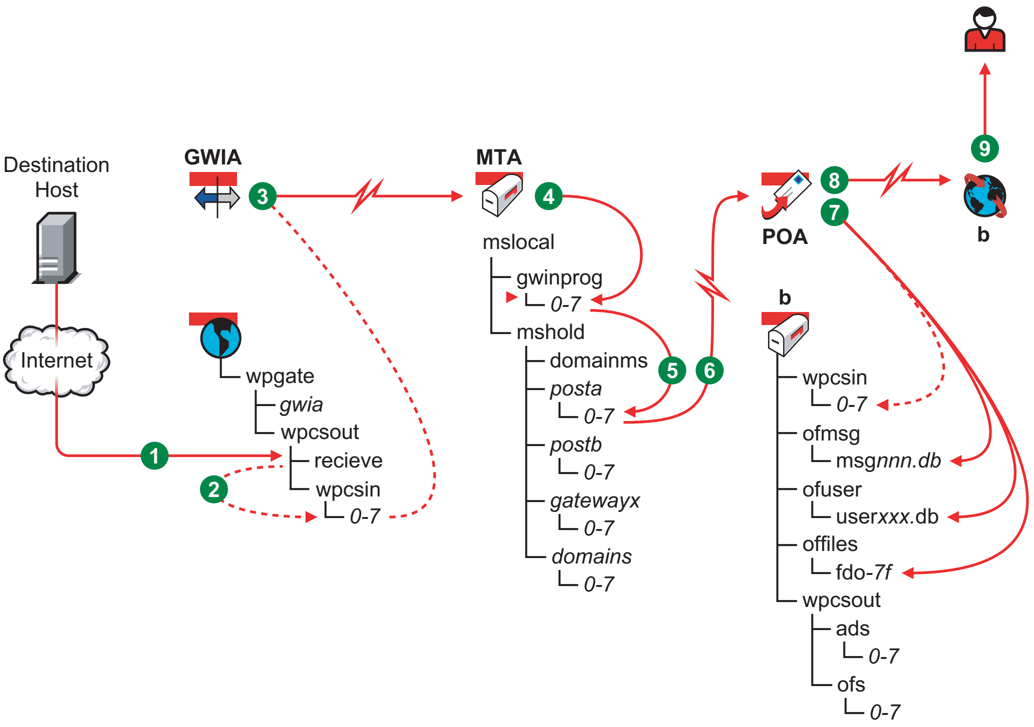

Table 4-5 *Message Flow in From the Internet Stages*

Stage	Actor	Action
❶ Internet Agent for Recipient's Domain	GWIA	An Internet user sends a message to a GroupWise user. The Internet Agent receives the message from the external Internet host and places the message file in the wpgate\gwia\receive directory.
❷ Internet Agent for Recipient's Domain	GWIA	The Internet Agent polls the receive directory according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the Internet Agent 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 Internet Agent puts messages only in the 4 directory, used for normal priority messages.
❸ Internet Agent for Recipient's Domain	GWIA	The Internet Agent 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.

Stage	Actor	Action
5 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`).
6 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.
7 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.)
8 Recipient's GroupWise Client	🌐	The Notify component of the recipient's GroupWise client notifies the recipient that a new message has arrived.
8 Recipient	👤	Each recipient opens the message in the GroupWise client.

4.6 Mapped/UNC Link: 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 Internet Agent and the MTA for the recipient's domain is a mapped/UNC link.

Figure 4-6 *Message Flow in From the Internet*

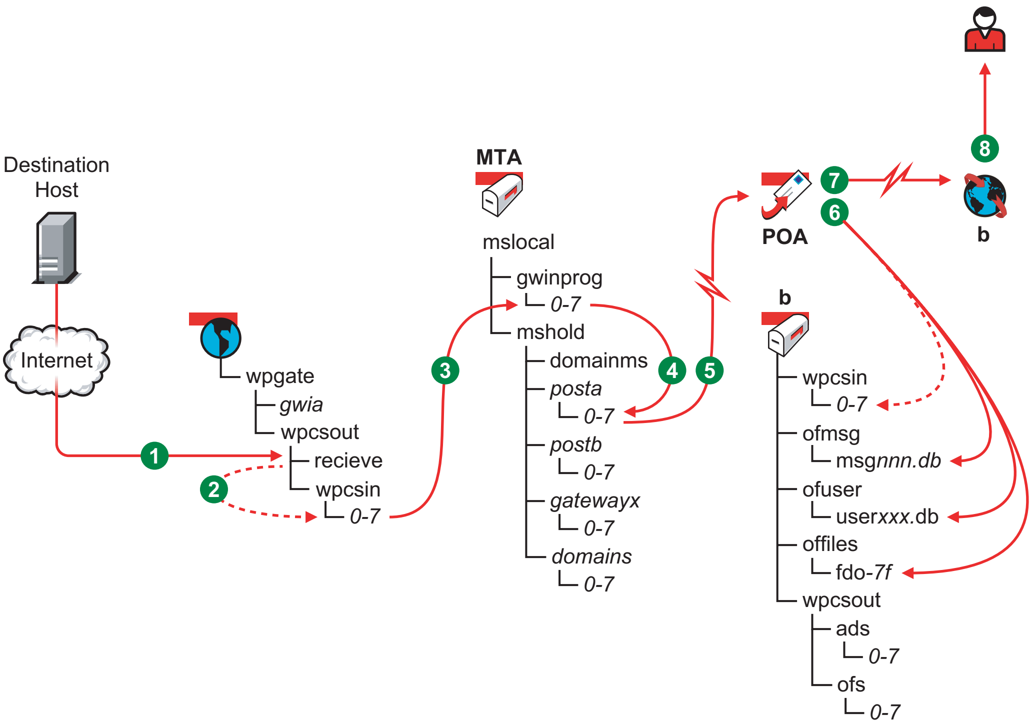

Table 4-6 *Message Flow in From the Internet Stages*

Stage	Actor	Action
❶ Internet Agent for Recipient's Domain	GWIA	An Internet user sends a message to a GroupWise user. The Internet Agent receives the message from the external Internet host and places the message file in the wpgate\gwia\receive directory.
❷ Internet Agent for Recipient's Domain	GWIA	The Internet Agent polls the receive directory according to the *Idle Sleep Duration* setting on the Gateway Time Settings page of the Internet Agent 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 Internet Agent 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\fd*0-7F* 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).

Stage	Actor	Action
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.
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.

Message Delivery through a Modem Connection

5

GroupWise® client users can still access information in their mailboxes even when not connected to the network by running the GroupWise client in Remote mode.

- Section 5.1, "“Hit the Road” Process in Online Mode," on page 51
- Section 5.2, "Modem Link through the Async Gateway in Remote Mode," on page 52

5.1 “Hit the Road” Process in Online Mode

This message flow diagram illustrates how a user who will be away from the network prepares to access GroupWise from a remote location by downloading mailbox contents.

Figure 5-1 *Message Flow for Hit the Road*

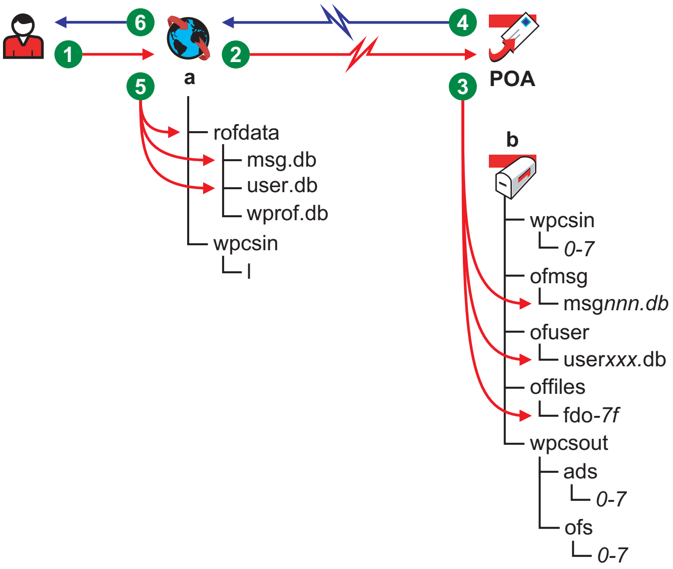

Table 5-1 *Message Flow for Hit the Road Stages*

Stage	Icon	Description
❶ Remote User	👤	The GroupWise user runs Hit the Road in Online mode in order to request items from the master mailbox to be downloaded to the Remote mailbox in preparation for disconnecting from the master GroupWise system. For example, the user could be preparing a laptop computer for use away from the network.

Stage	Icon	Description
❷ GroupWise Client in Online Mode	🌐	The GroupWise client prompts the user for the location to create the Remote mailbox and the types of items to download, then it transfers the request to the POA.
❸ POA for Remote User's Post Office	POA	The POA receives the request from the GroupWise client and performs the following actions: • Gathers all folders from the GroupWise user's master mailbox (`msgnnn.db` and `userxxx.db`) so that it has somewhere to put the items requested by the user. • Gathers the requested items from the GroupWise user's master mailbox. • Gathers any attachments for requested items from the `post_office\offiles\fd0-F6` subdirectories in the GroupWise user's post office. • Gathers any other types of information requested by the GroupWise user, such as rules, address books, documents, and junk mail lists.
❹ POA for Remote User's Post Office	POA	The POA compiles the information into a response file and transfers the response to the GroupWise client.
❺ GroupWise Client in Online Mode	🌐	The GroupWise client receives the response and performs the following actions for the GroupWise user: • Updates the GroupWise Remote message database (`msg.db`) with any items requested from the user's master mailbox. • Creates pointers in the GroupWise Remote user database (`user.db`) so the messages gathered from the master mailbox appear in the user's GroupWise mailbox when running in Remote mode. • Places any requested attachments larger than 2 KB in the `rofdata` directory 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.) • Updates the Remote mailbox with any other types of information requested by the GroupWise user. The user's GroupWise Remote mailbox now contains current copies of requested items from the user's master mailbox.
❻ Remote User	👤	The GroupWise client restarts in Remote mode and accesses the new Remote mailbox. The GroupWise user can now review current GroupWise mail in Remote mode after the connection to the master GroupWise system is no longer available.

5.2 Modem Link through the Async Gateway in Remote Mode

This message flow diagram illustrates how a GroupWise user in Remote mode can access the master GroupWise system through the GroupWise Async Gateway.

Figure 5-2 *Message Flow Through a Modem*

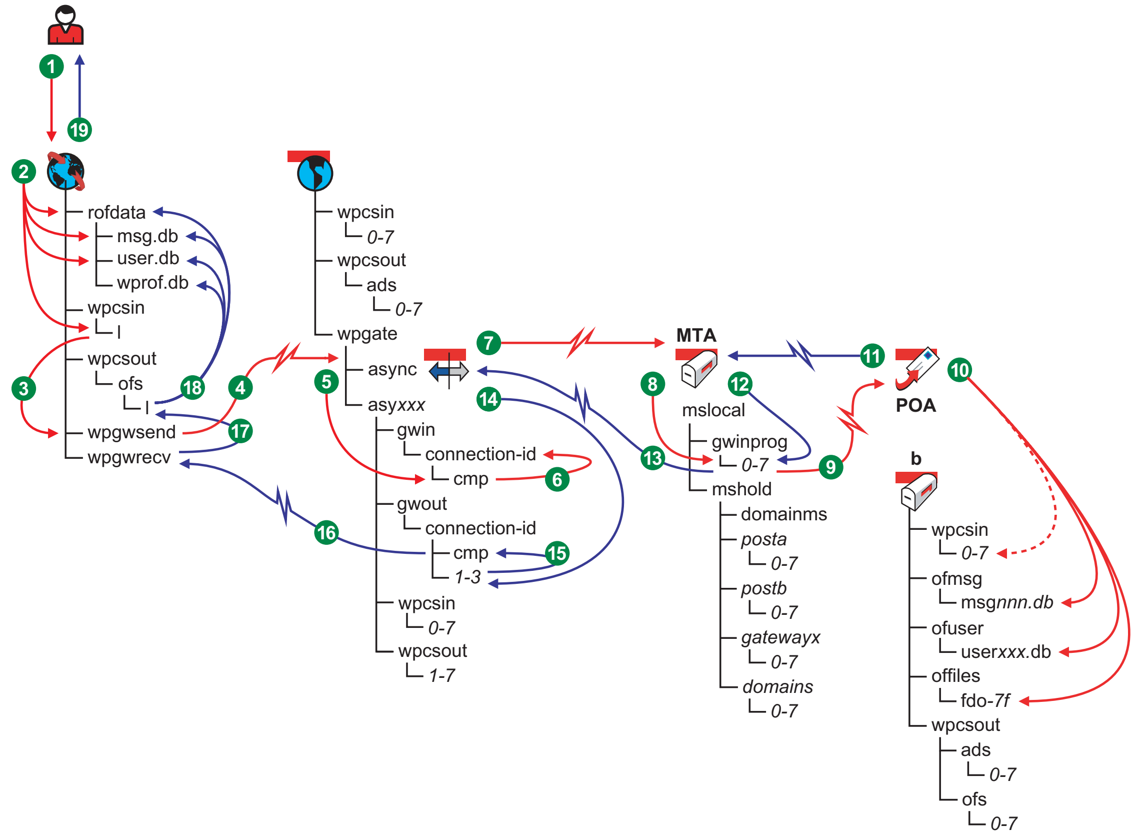

Table 5-2 *Message Flow Through a Modem Stages*

Stage	Icon	Description
1 Remote User		The GroupWise user, who is working in Remote mode, sends a message to another GroupWise user or creates a request for items from the master mailbox. The remote user's computer is not currently connected to the network or the user's master GroupWise system.

Stage	Icon	Description
❷ GroupWise Client in Remote Mode		When the remote GroupWise user sends a message to another GroupWise user, the GroupWise client performs the following actions in the user's Remote mailbox: • Adds the message to the message database (`msg.db`) on the user's remote computer. • Creates a pointer in the user database (`user.db`) so the message appears as a sent item in the Remote mailbox on the user's remote computer. • Places attachments larger than 2 KB in the `rofdata` directory and creates pointers from the message to its attachments on the user's remote computer. (For database efficiency, messages and distribution lists larger than 2 KB are also handled as attachments.) • Creates a copy of the message in the `wpcsin\1` subdirectory of the GroupWise client's remote input queue on the user's remote computer. When the remote GroupWise user sends a request for items from the master mailbox to be downloaded to the Remote mailbox, the GroupWise client places the request in the `wpcsin\1` subdirectory of the GroupWise client's remote input queue on the user's remote computer.
❸ GroupWise Client in Remote Mode		When the remote GroupWise user initiates the modem connection, the GroupWise client polls its input queue (`wpcsin\1`) and compresses the outgoing messages and/or requests into a file. If the compressed file totals over 50 KB, additional compressed files are created. The GroupWise client moves the compressed message/request files into its output queue (`wpgwsend`) directory.
❹: GroupWise Client with Modem Connection		The GroupWise client dials in to the gateway, logs in, then transmits the compressed message/request files across the modem connection to the GroupWise Async Gateway in the GroupWise system where the user's master mailbox is located.
❺ Async Gateway		The GroupWise Async Gateway receives the message/request files and places them into it input queue (`wpgate\async\asyxxx\gwin\`*`connection_id`*`\cmp`) for processing.
❻ Async Gateway		The Async Gateway decompresses the message/request files and moves them into the subdirectory for the remote user's connection with the master GroupWise system.
❼ Async Gateway		WIth a TCP/IP link, the Async Gateway transfers the decompressed files to the MTA for the remote user's domain in the master GroupWise system. With a UNC/mapped link, the Async Gateway places the message/request files into the MTA's input queue (`wpgate\async\wpcsin\1`), where the MTA picks up the files.
❽ MTA for Remote User's Domain	MTA	The MTA for the remote user's domain receives the message/request files and places them into the MTA "in progress" (`gwinprog`) queue.

Stage	Icon	Description
9 MTA for Remote User’s Domain	MTA	The MTA for the remote user’s domain then communicates the message/request files to the POA for the post office where the remote user’s master mailbox is located. **NOTE:** This message flow diagram illustrates only the simplest case where the recipient is in the same post office as the remote user’s master mailbox. If the remote user sends a message to a user in any other post office, the MTA routes the message to the appropriate destination.
10 POA for Remote User’s Post Office	POA	When the POA receives a message from the remote user, the POA performs the following actions to update the remote user’s master mailbox: • Adds the message to the remote user’s message database (`msg`*`nnn`*`.db`) in the Online mailbox. • Creates a pointer in the recipient’s user database (`user`*`xxx`*`.db`) in the Online mailbox so that 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 in the remote user’s 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.) • Creates a Delivered status message in the priority `1` subdirectory of the remote user’s MTA input queue (`wpcsin`) in the post office. When the POA receives a request for items from the remote user’s master mailbox, the POA performs the following actions: • Gathers the requested items from the remote user’s master mailbox (`msg`*`nnn`*`.db`). • Gathers any attachments for requested items from the *`post_office`*`\offiles\fd`*`0-F6`* subdirectory in the remote user’s post office. • Compiles the information into a response file and places it in the priority `1` subdirectory of the MTA input queue for return to the remote user.
11 POA for Remote User’s Post Office	POA	The POA for the remote user’s post office communicates the Delivered status for messages and the response file for requests to the MTA for the remote user’s domain. When the transfer is successful, the copies in the MTA input queue are deleted.
12 MTA in Remote User’s Domain	MTA	The MTA for the remote user’s domain places the status/response files into the MTA “in progress” (`gwinprog`) queue.

Stage	Icon	Description
⑬ MTA in Remote User's Domain	MTA	With a TCP/IP link, the MTA for the remote user's domain communicates the status/response files to the Async Gateway in the remote user's domain. With a UNC/mapped link, the MTA for the remote user's domain places the status/response files into the Async Gateway's input queue (`wpgate\async\asyxxx\wpcsout\1`), where the Async Gateway picks up the files. When the MTA in the GroupWise Remote user's domain detects the response for the GroupWise Remote user, the MTA picks it up from its post office input queue and transfers it to its output queue in the Async Gateway directory under `wpgate` in the GroupWise Remote user's domain. The MTA output queue in the Async Gateway directory is the input queue for the Async Gateway.
⑭ Async Gateway	⇆	The Async Gateway places the status/response files into the MTA output queue (`wpgate\async\wpcsout\asyxxx\connection_id\1`) of the Async Gateway directory.
⑮ Async Gateway	⇆	If the modem connection to the remote user is still active, the Async Gateway compresses the status/response files and moves them to the cmp directory. If the modem connection is no longer available, the status/response files wait in the `connection_id\1` subdirectory until a new connection is established by the remote user.
⑯ Async Gateway with Modem Connection	⇆	The Async Gateway transmits the status/response files through the modem connection to the input queue (`wpgwrecv`) for the GroupWise client on the user's remote computer.
⑰ GroupWise Client in Remote Mode	🌐	The GroupWise client on the remote computer decompresses the status/response files and places them in its input queue (`wpcsout\ofs\1`) on the user's remote computer.

Stage	Icon	Description
⓲ GroupWise Client in Remote Mode		Taking the items from its input queue, the GroupWise client performs the following actions to update the Remote mailbox on the user's remote computer: • Updates the message database (`msg.db`) with any items requested from the remote user's master mailbox. • Creates pointers in the user database (`user.db`) so the messages gathered from the master mailbox appear in the Remote mailbox. • Places any requested attachments larger than 2 KB in the rofdata directory 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.) • Updates the remote Address Book (`wprof.db`) to synchronize it with the Address Book in the remote user's master GroupWise system. The user's Remote mailbox now contains current copies of requested items from the remote user's master mailbox, plus any messages received in the user's master mailbox from other GroupWise users.
⓯ Remote User		The GroupWise user can now review current GroupWise mail when the modem connection to the master GroupWise system is no longer available.

Administrative Database Update

6

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.

***Figure 6-1** Administrative Message Flow through TCP/IP*

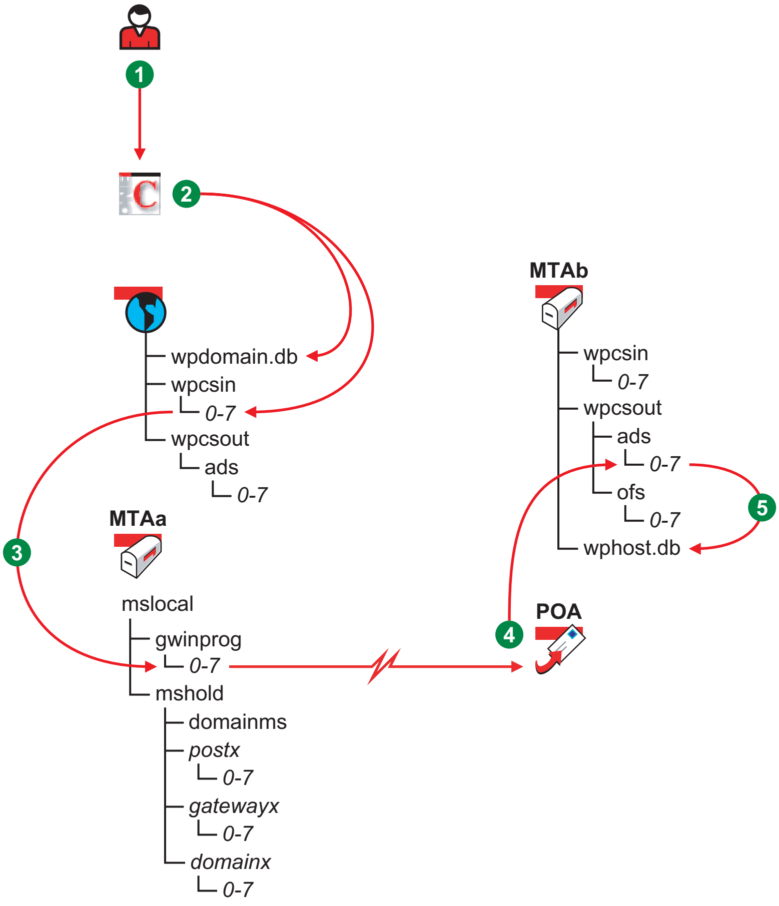

***Table 6-1** Administrative Message Flow through TCP/IP Stages*

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.

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.

7 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.

7.1 Domain Directory

domain	Domain 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
wptools	Supporting program directory
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 5.*x*, 6.*x*, and 7 domain databases
📄 gwpo.dc	Data dictionary for 5.*x*, 6.*x*, and 7post office databases
📄 viewcopy.log	Log file recording view file updates for post offices

7.1.1 *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).

7.1.2 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 7 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 7 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 7 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 7 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 7 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 7 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.

7.1.3 wptools directory

The wptools subdirectory in the domain contains programs that support GroupWise administration.

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

7.1.4 wpgate directory

The wpgate subdirectory in the domain contains a subdirectory for each GroupWise gateway you have installed in your GroupWise system. For a list of gateways, see GroupWise Gateways (http://www.novell.com/documentation/gwateways). GroupWise 5.5 gateways can be used with GroupWise 6.*x* and 7.

7.1.5 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 7 Troubleshooting 2: Solutions to Common Problems*.

7.1.6 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.

7.1.7 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 re-create 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 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 7/6.*x*/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.

7.1.8 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 on the GroupWise CD.

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.

7.1.9 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 on the GroupWise CD.

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.

7.1.10 gwdom.dc file

The `gwdom.dc` file in the domain is the data dictionary for creating and rebuilding GroupWise 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 7/6.*x*/5.*x* secondary domains. The original `gwdom.dc` file is located in the `domain` directory of the software distribution directory or on the GroupWise distribution media.

7.1.11 gwpo.dc file

The `gwpo.dc` file in the domain is the data dictionary for creating and rebuilding GroupWise 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 7/6/*x*/5.*x* post offices in that domain. The original `gwpo.dc` file is located in the `domain` directory of the software distribution directory or on the GroupWise distribution media.

7.1.12 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 7 Administration Guide*.

7.2 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
📁 gwdms	Document Management Services directory
🗋 dmsh.db	Shared Document Management Services database
📁 lib00*01-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
📁 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
🗋 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 7/6.*x*/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

7.2.1 *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.

7.2.2 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 7 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.

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 7 Administration Guide*.

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:

- Remote with a direct connection places requests here for routing to remote GroupWise users' master mailboxes.
- The POA places outgoing status messages to remote GroupWise users here for routing to the Async Gateway.

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 7 Administration Guide*.

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.

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 7 Administration Guide*.

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.

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 7 Administration Guide*.

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.

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 7 Administration Guide*.

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.

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.

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.

7.2.3 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 filenames represents the partition number, which is generated by a hashing algorithm to guarantee uniqueness.

The *01-ff* in the filenames 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.

7.2.4 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 "Standalone Database Maintenance Programs" in the *GroupWise 7 Administration Guide*.

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.

7.2.5 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).

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.

7.2.6 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.

7.2.7 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 gwview*xx*.ini and ofview*xx*.ini files configure the standard views on the menus where users select views. The gwview*xx*.ini file configures GroupWise 7, 6.*x*, and 5.5 standard views. The ofview*xx*.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.

7.2.8 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.

7.2.9 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.

7.2.10 oftemp directory

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

7.2.11 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

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. You can change the location if needed. See "Using POA Log Files" in "Post Office Agent" in the *GroupWise 7 Administration Guide* guide.

The first two digits of the filename 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 wprov50.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 7 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 7 Troubleshooting 2: Solutions to Common Problems*.

7.2.12 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 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 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.

7.2.13 gwpo.dc file

The gwpo.dc file in the post office is the data dictionary for creating and rebuilding GroupWise 7, 6.*x*, and 5.*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 7/6.*x*/5.*x* post offices in that domain. The original gwpo.dc file is located in the domain directory of the software distribution directory or on the GroupWise CD.

7.2.14 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 7 Administration Guide*.

7.2.15 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)

7.2.16 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 7 Administration Guide*.

7.2.17 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 7 Administration Guide*.

7.2.18 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.

7.3 MTA Local Queue Directory

📁 mslocal	MTA local working directory
🗋 *mmddxxx.nn*	MTA log files
📁 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

7.3.1 mslocal directory

The `mslocal` directory is the MTA local working directory. The /work startup 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.

Typical locations for the mslocal directory include:

- `sys:\system` on a NetWare® server
- `c:\` on a Windows server

To move the `mslocal` directory, 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 startup 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 7 Administration Guide*.

The first two digits of the filename 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.

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.

7.3.2 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 7 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 7 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 filename 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.

7.3.3 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 7 Administration Guide*.

7.3.4 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.

7.3.5 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.

7.3.6 *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 19.

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.

7.3.7 *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.

7.3.8 *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 27.

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.

7.3.9 *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 7.1, "Domain Directory," on page 63
- Section 7.2, "Post Office Directory," on page 70

7.3.10 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*.

7.3.11 gwvsscan directory

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

7.3.12 mtaconv directory

The mtaconv subdirectory is the working directory where the MTA converts GroupWise 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.

7.4 Internet Agent Queue Directory

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

▨ *domain*\wpgate\gwia	GroupWise Internet Agent home directory
▨ 000.prc	Internet Agent message processing directory
▨ cmd	Not currently used
▨ gwwork	Hold 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 Internet Agent operation
□ proc	Process lock file indicating that the Internet Agent is running
□ pulse.tmp	Temporary file to verify Internet Agent 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	Hold directory for damaged inbound messages
▨ gwchars	Directory for character conversion tables
▨ save	Directory for old configuration files from reinstalls or upgrades

gwia.cfg	Internet Agent configuration file for startup 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
statusxx.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	Hold directory for re-queued and deferred messages
work	Schedule files for SMTP service operations on deferred messages
dsnhold	Delivery Status Notification (DSN) hold directory

7.4.1 *domain*\wpgate*gwia* directory

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

000.prc directory

The Internet Agent uses the 000.prc directory to process messages.

gwwork directory

The gwwork directory stores temporary files created by the Internet Agent 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 Internet Agent. The Internet Agent 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 Internet Agent Log Files" in "Internet Agent" in the *GroupWise 7 Administration Guide*.

acct file

The `acct` file contains information about the messages the Internet Agent sends each day. It is e-mailed 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 7 Administration Guide*.

set file

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

stat file

The `stat` file stores statistics about the Internet Agent's functioning. For information about the statistics provided by the Internet Agent, see "Statistics" in "Internet Agent" in the *GroupWise 7 Administration Guide*.

proc file

The `proc` file is the lock file for the Internet Agent process. The proc file is opened and locked when the Internet Agent starts. This prevents multiple Internet Agents from being started for the same domain.

pulse.tmp file

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

wpcsin directory

For a mapped/UNC link, the Internet Agent 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 Message Transfer Agent retrieves the messages and delivers them to the proper destinations.

For a TCP/IP link, the Internet Agent 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 35.

wpcsout directory

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

For a TCP/IP link, the Internet Agent 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 35.

gwixxxx directory

The *gwixxxx* directory is a system-defined directory, where *gwi* represents the first three letters of the Internet Agent object name as defined during installation and displayed in ConsoleOne, and *xxxx* is a randomly-generated string. Here, the Message Transfer Agent places outbound messages in the appropriate 0-7 priority subdirectory for the Internet Agent 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 7 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 Internet Agent transfers them to the send directory for processing by the SMTP service.

gwprob directory

The Internet Agent 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 Internet Agent uses to convert message attachments between character sets.

save directory

If you reinstall or upgrade the Internet Agent, 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 Internet Agent configuration file that contains startup switches. Some switches are set during installation. You can set others as needed. For more information, see "Using Internet Agent Startup Switches" in "Internet Agent" in the *GroupWise 7 Administration Guide*.

NetWare:	The NetWare Internet Agent uses the *gwia*.cfg file created in sys:\system during installation. The *gwia*.cfg file under the domain is just a boilerplate file with no switches set during installation.
Linux:	The Linux Internet Agent 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 Internet Agent 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 7 Administration Guide*.

gwauth.cfg file

The gwauth.cfg file enables the Internet Agent to log in to SMTP hosts that require authentication. For more information, see "SMTP Host Authentication" in "Internet Agent" in the *GroupWise 7 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 7 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 sys:\system directory on NetWare, 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 Internet Agent object in ConsoleOne. For more information, see "Configuring How the Internet Agent Handles E-Mail Addresses" in "Internet Agent" in the *GroupWise 7 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 Internet Agent 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 7 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 7 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 7 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 7 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 7 Administration Guide*.

status*xx*.xml file

The `status`*`xx`*`.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 7 Administration Guide*.

7.4.2 *gwia* directory

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

send directory

The Internet Agent 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 Internet Agent SMTP service places inbound messages in the receive directory, converts them into GroupWise format, and then passes them to the Message Transfer Agent by placing them in the wpcsin directory.

result directory

When the Internet Agent 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 Internet Agent SMTP service has completed the transmission with the destination host, it moves another file, s*.* from the send directory to the result directory. The filenames 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 Internet Agent 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 7 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 7 Administration Guide*.

7.5 WebAccess Agent Queue Directory

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

📁 *domain*\wpgate\webac70a	GroupWise WebAccess Agent home directory
📁 000.prc	WebAccess Agent log file processing directory
📁 gwwork	Hold directory for temporary files used during processing
📄 *mmddweb.nnn*	WebAccess Agent log files
📁 wpcsin	MTA input queue directory
📁 *0-7*	Message priority subdirectories

wpcsout	MTA output queue
webxxxx	System-defined directory for the WebAccess Agent
0-7	Message priority subdirectories
problem	Hold directory for damaged outbound messages
gwhold	Hold directory for delayed delivery messages
gwprob	Hold directory for damaged inbound messages
template	Directory for templates for viewing documents
commgr.cfg	Communications Manager configuration file
comint.cfg	Communications initialization configuration file
mimetype.cfg	MIME encoding configuration file for various file types
gwac.db	Access control database
gwac.dc	Database dictionary file used to create the gwac.db file

7.5.1 *domain*\wpgate\webac70a directory

The *domain*\wpgate\webac70a directory is the WebAccess Agent home directory where WebAccess Agent configuration files and queue directories are located. The name is established when you install the WebAccess Agent. The default is wpgate\webac70a in the domain directory. You can change the location using the /home startup switch in the WebAccess Agent configuration file (*webac70a*.waa in the WebAccess Agent installation directory).

7.5.2 000.prc directory

The NetWare and Windows WebAccess Agents use the 000.prc directory to store log files.

On Linux, the 000.prc directory is located under /var/log.

*mmdd*log.*nnn* file

The *mmdd*log.*nnn* files hold error and status messages about the functioning of the WebAccess Agent. The WebAccess Agent 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 about log files, see "Controlling WebAccess Agent Logging" in "WebAccess" in the *GroupWise 7 Administration Guide*.

7.5.3 wpcsin directory

No longer used. The WebAccess Agent and the MTA communicate by way of TCP/IP and do not need queue directories.

7.5.4 wpcsout directory

No longer used.

7.5.5 gwhold directory

No longer used.

7.5.6 gwprob directory

No longer used.

7.5.7 template directory

The `template` directory holds the HTML templates used for viewing documents in HTML format.

7.5.8 commgr.cfg file

The `commgr.cfg` file in the WebAccess Agent queue directory contains information for communication between the WebAccess Agent and the WebAccess Application, including the IP address and port where the WebAccess Agent is running, the number of threads that are running, and the encryption key for the WebAccess Agent. This communications information is gathered during installation. For more information, see "Configuring the GroupWise Service Provider" in "WebAccess" in the *GroupWise 7 Administration Guide*.

As part of the installation process, the `commgr.cfg` file is automatically copied to the Web server installation (`sys:\novell\webaccess` on NetWare, `/opt/novell/groupwise/webaccess` on Linux, and `c:\novell\webaccess` on Windows). The copies are synchronized automatically by the WebAccess Application. The `commgr.cfg` file is also copied to the `webpublisher` subdirectory on the Web server.

7.5.9 comint.cfg file

The `comint.cfg` file in the WebAccess Agent queue directory is read by the WebAccess Agent on startup. It contains the same communications information as the `commgr.cfg` file and is synchronized with it automatically.

7.5.10 mimetype.cfg file

The `mimetype.cfg` file enables you to customize MIME content-type mappings for various attachment types. The WebAccess Agent handles this just as the Internet Agent does. For more information, see "Customizing MIME Content-Type Mappings" in "Internet Agent" in the *GroupWise 7 Administration Guide*

7.5.11 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 Database" in "WebAccess" in the *GroupWise 7 Administration Guide*.

7.5.12 gwac.dc file

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

7.6 Caching Mailbox Directory

\novell\groupwise\gwxxxxxx	GroupWise Caching mailbox
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

7.6.1 *\novell\groupwise*\gw*xxxxxx* directory

Your GroupWise Caching mailbox is a directory structure that functions similarly to a post office. Like a post office, it contains databases and input/output queues. It is created in the directory where the GroupWise client is installed, which is typically `\novell\groupwise`.

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.

7.6.2 rofdata directory

The `rofdata` directory contains the databases accessed by the GroupWise Windows client when running in Caching mode. The databases in `rofdata` are similar to the databases found in post offices. For comparison, see Section 7.2, "Post Office Directory," on page 70.

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 `user`*`xxx`*`.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 7 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.

7.6.3 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.

7.6.4 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.

7.6.5 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.

7.6.6 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.

7.6.7 remote*n*.log

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.

7.7 Remote Mailbox Directory

`remote_mailbox`	GroupWise mailbox on a remote computer
`rofdata`	Remote database directory
`msg.db`	Remote message database
`user.db`	Remote user database
`wprof.db`	Remote Address Book
`wprof.dc`	Data dictionary for Remote Address Book
`ngwguard.db`	Remote guardian database
`ngwguard.dc`	Data dictionary for Remote 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 Remote mailbox
📁 index	QuickFinder index for Remote mailbox
📁 index	QuickFinder index for messages in the Remote mailbox
📁 wpcsin	Input queue for the GroupWise client in Remote mode
📁 *0-7*	Priority subdirectories
📁 wpcsout\ofs	Output queue for the GroupWise client in Remote mode
📁 *0-7*	Priority subdirectories
📁 wpgwsend	Output queue to master mailbox
📁 wpgwrecv	Input queue from master mailbox
🗋 remote*n*.log	Remote connection log

7.7.1 *remote_mailbox* directory

The GroupWise remote mailbox is a directory structure that functions similarly to a post office. Like a post office, it contains databases and input/output queues.

7.7.2 rofdata directory

The rofdata subdirectory in the remote mailbox directory contains the databases accessed by the GroupWise client in Remote mode. The databases in rofdata are similar to the databases found in post offices.

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 in the remote data directory (rofdata) in the remote mailbox directory is the remote equivalent of the msg*nnn*.db files in the ofmsg directory in the post office where your master mailbox is located. The msg.db file contains messages you have downloaded from your master mailbox.

user.db file

The user.db file in the remote data directory (rofdata) in the remote mailbox directory is the remote equivalent of the user*xxx*.db files in the ofuser directory in the post office where your master mailbox is located. The user.db file contains user information you have downloaded from your master mailbox.

wprof.db file

The wprof.db file in the remote data directory (rofdata) in the Remote mailbox directory contains the remote version of the GroupWise Address Book if you have downloaded it.

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 in the remote data directory (rofdata) in the Remote mailbox directory is the data dictionary for the remote 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 in the remote data directory (rofdata) in the Remote mailbox directory is the guardian database for the remote GroupWise mailbox. It is parallel in function to the ngwguard.db file in the post office.

ngwguard.dc file

The ngwguard.dc file in the remote data directory (rofdata) in the Remote mailbox directory is the data dictionary for building the databases in the remote GroupWise 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 in the remote data directory (rofdata) in the Remote mailbox directory 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 Remote user's 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 Remote mode 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 Remote mailbox repair. For assistance with GroupWise Check errors, see "GroupWise Check Error Codes" in "Administration Error Messages" in the *GroupWise 7 Troubleshooting 1: Error Messages*.

gwdms directory

The gwdms subdirectory in the remote data directory (rofdata) in the remote mailbox directory is the Document Management Services directory. It contains the document libraries you have downloaded from your master 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 have been downloaded to your remote GroupWise mailbox. It 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 Remote mailbox.

index directory

The index directory contains the QuickFinder index for the documents contained in the library.

index directory

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

7.7.3 wpcsin directory

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

When the user initiates the connection to the master GroupWise system, the GroupWise client polls the wpcsin\1 directory and compresses the 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 Remote mailbox. Some naming conventions were originally preserved for backward compatibility.

0-7 directories

The priority 0-7 subdirectories in the connection input queue (wpcsin) in the remote mailbox directory parallel those found in the wpcsin directory in the remote user's post office on the master GroupWise system.

7.7.4 wpcsout\ofs directory

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

In Remote mode, the GroupWise client scans the wpcsout\ofs\1 subdirectory and updates the user.db and msg.db files with the information received from the master 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) in the remote mailbox directory parallel those found in the ofs subdirectory in the remote user's post office on the master GroupWise system.

7.7.5 wpgwsend directory

The wpgwsend subdirectory in the remote mailbox directory holds compressed files that contain outgoing messages, requests, or both. When a connection to your master GroupWise system is established, the GroupWise client uploads the files across the link from the remote computer to the master GroupWise system.

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.

7.7.6 wpgwrecv directory

The wpgwrecv subdirectory in the remote mailbox directory holds compressed files that contain messages or other information that have been downloaded from your master 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.

7.7.7 remote*n*.log

The remote*n*.log files in the connection output queue (wpcsout\ofs) in the remote mailbox directory are saved versions of the connection logs you can view in the GroupWise client in Remote mode. These log files can be useful for troubleshooting problems with your modem or with your connection to your master GroupWise system.

Agent Installation Directories

8

- Section 8.1, “GroupWise Agent Installation (POA and MTA),” on page 107
- Section 8.2, “Internet Agent Installation,” on page 119
- Section 8.3, “WebAccess Agent Installation,” on page 126
- Section 8.4, “Monitor Agent Installation,” on page 138
- Section 8.5, “Apache/Tomcat Installation,” on page 143

8.1 GroupWise Agent Installation (POA and MTA)

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

- “NetWare Installation Directory” on page 107
- Section 8.1.2, “Linux Installation Directory,” on page 110
- “Windows Installation Directory” on page 116

8.1.1 NetWare Installation Directory

sys:\system	NetWare® agent installation directory
gwenn5.nlm	GroupWise® 7 Agent Engine
gwpoa.nlm	Post Office Agent (POA) program
gwmta.nlm	Message Transfer Agent (MTA) program
strtup*xx*.poa	Boilerplate POA startup file
strtup*xx*.mta	Boilerplate MTA startup file
post_office.poa	POA startup file for a specific post office
domain.mta	MTA startup file for a specific domain
gwpoan*xx*.hlp	POA online help files
gwmtan*xx*.hlp	MTA online help files
gwpoan*xx*.fil	POA language information files
gwmtan*xx*.fil	MTA language information files
grpwise.ncf	Agent load file
x*5.nlm	XIS NLM™ programs
gwxis12.nlm	GroupWise XIS NLM program

agent.xml	XIS agent configuration files
gwtsa.nlm	GroupWise Target Service Agent (GWTSA) program
gwtsa*xx*.fil	GWTSA language information files
gwtsa.ncf	GWTSA load file
gwtmstmp.nlm	GroupWise Time Stamp program
help	Subdirectory for GroupWise agent Web console help files

NetWare agent installation directory

When you first install GroupWise Administration, the GroupWise agents are installed in platform-specific subdirectories in the agent directory in the software distribution directory. However, the agents cannot be run from this initial location. They must be installed on the servers where they will run.

On a NetWare® server, the GroupWise agents are typically installed in the sys:\system directory. You can choose a different location during installation.

gwenn5.nlm file

The gwenn5.nlm file is the GroupWise 7 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 loaded on a server automatically loads the GroupWise Agent Engine.

gwpoa.nlm file

The gwpoa.nlm file is the Post Office Agent program. You load this NLM program to start the Post Office Agent. See "Starting the POA" in "Post Office Agent" in the *GroupWise 7 Administration Guide*.

gwmta.nlm file

The gwmta.nlm is the Message Transfer Agent program. You load this NLM program to start the Message Transfer Agent. See "Starting the MTA" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

strtup*xx*.poa file

The POA startup file contains startup 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 filename 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 startup switch automatically set to the post office directory the POA will service. See "Using POA Startup Switches" in "Post Office Agent" in the *GroupWise 7 Administration Guide*.

strtup*xx*.mta file

The MTA startup file contains startup 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 filename 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 startup switch automatically set to the domain directory the MTA will service. See "Using MTA Startup Switches" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

nlm_agentnxx.hlp files

Online help is available by pressing F1 at the agent console on the server where it runs. See "Using the POA Server Console" in "Post Office Agent" and "Using the MTA Server Console" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*

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

nlm_agentnxx.fil files

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

grpwise.ncf file

The grpwise.ncf file is the NetWare configuration file that automatically loads the agents whenever the server is restarted. A typical grpwise.ncf file would look similar to the following:

```
load sys:\system\gwmta.nlm @corpdom.mta
load sys:\system\gwpoa.nlm @acctpo.poa
```

For more information, see "Starting the NetWare POA" in "Post Office Agent" and "Starting the NetWare MTA" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

x*5.nlm and gwxis12.nlm files

The x*5.nlm and gwxis12.nlm programs provide XIS capability for the GroupWise agents, so that you can create XML documents to monitor and change the agents as needed.

nlm_agent.xml files

The *nlm_agent*.xml files are specialized agent configuration files for use in the XIS environment.

gwtsa.nlm file

The gwtsa.nlm is the GroupWise Target Service Agent (GWTSA), which provides reliable backups of a running GroupWise system by successfully backing up open files and locked files, rather than skipping them as some backup software does. For more information, see "GroupWise Target Service Agent (GWTSA) for NetWare 5.1" in "Databases" in the *GroupWise 7 Administration Guide*.

gwtmstmp.nlm file

If you use the GroupWise Target Service Agent (GWTSA) of the GroupWise Target Server Agent for File Systems (GWAFSGW) to back up your GroupWise system, the time stamping is an automatic part of the backup process. However, if you choose not to use one of these TSAs, you must still make sure that user databases are time-stamped so that items are not prematurely purged. The gwtmstmp.nlm accomplishes this task. For more information, see "GroupWise Time Stamp Utility" in "Databases" in the *GroupWise 7 Administration Guide*.

gwtsa*xx.fil* file

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

gwtsa.ncf file

The gwtsa.ncf file is the NetWare configuration file that automatically loads the GWTSA whenever the server is restarted. A typical gwtsa.ncf file would look similar to the following:

```
load sys:\system\smdr
load sys:\system\gwtsa /home-sys:\gwsystem\corpdom
     /home-sys:\gwsystem\acctpo
```

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 7 Administration Guide*.

8.1.2 Linux Installation Directory

/opt/novell/groupwise/agents	Linux agent installation directory

📁 bin	Subdirectory for GroupWise agent executables
🗋 gwpoa	Post Office Agent (POA) executable
🗋 gwmta	Message Transfer Agent (MTA) executable
🗋 gwha	GroupWise High Availability service executable
🗋 gwpoa.xml	POA XIS configuration file
🗋 gwmta.xml	MTA XIS configuration file
🗋 strtup*xx*.poa	Boilerplate POA startup file
🗋 strtup*xx*.mta	Boilerplate MTA startup file
🗋 gwcsrgen	GroupWise Generate CSR utility
🗋 gwtmstmp	GroupWise Time Stamp utility
📁 lib	Subdirectory for GroupWise agent library files
🗋 gwpoa*xx*.fil	POA language information file
🗋 gwmta*xx*.fil	MTA language information file
🗋 libpoa.so*	POA shared library files
🗋 libpoa_ui.so*	POA shared library files for the optional user interface
🗋 libmta.so*	MTA-specific library files
🗋 libmta_ui.so*	MTA-specific library files for the optional user interface
🗋 libgwha.so*	GroupWise High Availability service library files
🗋 *.so*	Supporting library files
📁 image	Subdirectory for icon images
📁 share	Subdirectory for agent shared files
🗋 *post_office*.poa	POA startup file for a specific post office
🗋 *domain*.mta	MTA startup file for a specific domain
📁 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 POA and MTA

rc3.d	Standard Linux location for runlevel-3 symbolic links
S99grpwise	Symbolic link to the startup script for the POA and MTA
rc5.d	Standard Linux location for runlevel-5 symbolic links
S99grpwise	Symbolic link to the startup script for the POA and MTA

/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
post_office.poa	Post office-specific subdirectory for POA log files
*mmdd*poa.*nnn*	POA log files
domain.mta	Domain-specific subdirectory for MTA log files
*mmdd*mta.*nnn*	MTA 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.

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 POA" in "Post Office Agent" in the *GroupWise 7 Administration Guide*.

gwmta file

The gwmta file is the Message Transfer Agent executable. You run this executable file to start the Message Transfer Agent. See "Starting the MTA" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

gwha file

The gwha file is the GroupWise High Availability service executable. If the POA or the MTA goes down for any reason, the High Availability service automatically restarts it. See "Enabling the High Availability Service for the Linux Agents" in "Installing GroupWise Agents" in the *GroupWise 7 Installation Guide*.

*.xml files

The *agent*.xml files are specialized agent configuration files for use in the XIS environment.

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 filename represents a two-letter language code.

strtup*xx*.mta file

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

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" in the *GroupWise 7 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 (user*xxx*.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 7 Administration Guide*.

lib directory

The lib directory holds GroupWise library files.

*.fil files

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

.so files

These files are Linux library files that provide information to the GroupWise executables.

image directory

The image directory has subdirectories for agent icons for both color and monochrome displays.

share directory

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

post_office.poa

The POA startup file contains startup 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 strtup*xx*.poa file, named *post_office*.poa, is created in the share directory. This customized version has the --home startup switch automatically set to the post office directory the POA will service. See "Using POA Startup Switches" in "Post Office Agent" in the *GroupWise 7 Administration Guide*.

domain.mta

The MTA startup file contains startup 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 strtup*xx*.mta file, named *domain*.mta, is created in the share directory. This customized version has the --home startup switch automatically set to the domain directory the MTA will service. See "Using MTA Startup Switches" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

agtcon directory

The agtcon directory holds subdirectories and files used by the agent 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 7 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 7 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 and Private Key" in "Security" in the *GroupWise 7 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 POA and MTA. For more information about starting the agents, see "Linux: Installing and Starting the GroupWise Agents" in "Installing a Basic GroupWise System" in the *GroupWise 7 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 S99grpwise. It is created if you choose during installation to have the agents start automatically when the server boots. See "Starting the Linux POA" in "Post Office Agent" and "Starting the Linux MTA" in "Message Transfer Agent" in the *GroupWise 7 Administration 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 S99grpwise. It is created if you choose during installation to have the agents start automatically when the server boots. See "Starting the Linux POA" in "Post Office Agent" and "Starting the Linux MTA" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

/etc/opt/novell/groupwise directory

The /etc/opt directory is the standard location of application configuration files on Linux. 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 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 GroupWise agents if they go down unexpectedly. See "Enabling the High Availability Service for the Linux Agents" in "Installing GroupWise Agents" in the *GroupWise 7 Installation Guide*.

uid.conf file

The uid.conf file configures the GroupWise agents to run as a non-root user. See "Running the Agents as a Non-root User" in "Installing GroupWise Agents" in the *GroupWise 7 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.

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 7 Administration Guide* guide.

The first two digits of the filename 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.

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 7 Administration Guide*.

The first two digits of the filename 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.

8.1.3 Windows Installation Directory

c:\grpwise	Windows agent installation directory
gwenv1a.dll	GroupWise Agent Engine
gwpoa.exe	Post Office Agent (POA) program
gwmta.exe	Message Transfer Agent (MTA) program
strtup*xx*.poa	Boilerplate POA startup file
strtup*xx*.mta	Boilerplate MTA startup file
post_office.poa	Customized POA startup file
domain.mta	Customized MTA startup file
post_office.poa	POA startup file for a specific post office
domain.mta	MTA startup file for a specific domain
gwpoan*xx*.chm	POA online help file
gwmtan*xx*.chm	MTA online help file
gwpoan*xx*.dll	POA language information files
gwmtan*xx*.dll	MTA language information files
gwsnmp.dll	Customized DLL program for SNMP
gwwww1.dll	Customized DLL program for MIME
x*10a.dll	XIS DLL programs
gwxis10.dll	GroupWise XIS DLL program
agent.xml	XIS agent configuration files

help	Subdirectory for GroupWise agent Web console help files

c:\grpwise

When you first install GroupWise Administration, the GroupWise agents are initially installed in platform-specific subdirectories in the agent directory. However, the agents cannot be run from this initial location. They must be installed on the servers where they will run.

On a Windows server, the GroupWise agents can be installed in any directory you choose. The default is c:\grpwise. 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 both 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.

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 POA” in “Post Office Agent” in the *GroupWise 7 Administration Guide*.

gwmta.exe file

The gwmta.exe file is the Message Transfer Agent program. You run this executable file to start the Message Transfer Agent. See “Starting the MTA” in “Message Transfer Agent” in the *GroupWise 7 Administration 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
post_office.poa file

The POA startup file contains startup 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 filename 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 startup switch automatically set to the post office directory the POA will service. See "Using POA Startup Switches" in "Post Office Agent" in the *GroupWise 7 Administration Guide*.

strtup*xx*.mta file
domain.mta file

The MTA startup file contains startup 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 filename 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 startup switch automatically set to the domain directory the MTA will service. See "Using MTA Startup Switches" in "Message Transfer Agent" in the *GroupWise 7 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 filename 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.

x*10a.dll and gwxis10.dll files

The `x*10a.dll` and `gwxis10.dll` programs provide XIS capability for the GroupWise agents, so that you can create XML documents to monitor and change the agents as needed.

agent.xml files

The *`agent`*`.xml` files are specialized agent configuration files for use in the XIS environment.

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 7 Administration Guide*.

8.2 Internet Agent Installation

- Section 8.2.1, "NetWare Installation Directory," on page 119
- Section 8.2.2, "Linux Installation Directory," on page 120
- Section 8.2.3, "Windows Installation Directory," on page 124

8.2.1 NetWare Installation Directory

sys:\system	NetWare Internet Agent installation directory
gwenn5.nlm	GroupWise 7 Agent Engine
gwia.nlm	Internet Agent NLM program
gwia.cfg	Internet Agent configuration file
gwia*nxx*.fil	Internet Agent language information file
gwia.ncf	Internet Agent load file
x*5.nlm	XIS NLM programs
gwxis12.nlm	GroupWise XIS NLM programs
gwia.xml	XIS Internet Agent configuration file
help	Subdirectory for GroupWise agent console help files

See also Section 7.4, "Internet Agent Queue Directory," on page 87.

NetWare Internet Agent installation directory

On a NetWare server, the GroupWise Internet Agent is typically installed in the sys:\system directory. You can choose a different location during installation.

gwenn5.nlm file

The gwenn5.nlm file is the GroupWise 7 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 loaded on a server automatically loads the GroupWise Agent Engine.

gwia.nlm file

The gwia.nlm file is the Internet Agent program. You load this NLM program to start the Internet Agent. See “Starting the Internet Agent” in “Installing the GroupWise Internet Agent” in the *GroupWise 7 Installation Guide*.

gwia.cfg file

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

During installation, a customized version of the Internet Agent startup file is created in sys:\system. This customized version has the /home startup switch automatically set to the domain directory where the Internet Agent’s queue directories are located. See “Using Internet Agent Startup Switches” in “Internet Agent” in the *GroupWise 7 Administration Guide*.

gwia*nxx*.fil files

This file contains all language-specific information for the Internet Agent. The digit *n* is a version number. The last two characters *xx* are a language code.

gwia.ncf file

The gwia.ncf file is the NetWare configuration file that automatically loads the Internet Agent whenever the server is restarted. A typical gwia.ncf file would look similar to the following:

```
load sys:\system\gwia @gwia.cfg
```

x*5.nlm and gwxis12.nlm files

The x*5.nlm and gwxis12.nlm programs provide XIS capability for the GroupWise agents, so that you can create XML documents to monitor and change the agents as needed.

gwia.xml file

The gwia.xml file is a specialized Internet Agent configuration file for use in the XIS environment.

8.2.2 Linux Installation Directory

📁 /opt/novell/groupwise/agents	Linux agent installation directory
📁 bin	Subdirectory for GroupWise agent executables
🗋 gwia	Internet Agent executable
🗋 gwia.xml	Internet Agent XIS configuration file
🗋 gwha	GroupWise High Availability service executable

lib	Subdirectory for GroupWIse agent library files
gwiaxxx.fil	Internet Agent language information file
libgwia.so	Internet Agent shared library file
libgwia_ui.so	Internet Agent user interface shared library file
libgwha.so*	GroupWise High Availability service library files
*.so	Supporting shared library files
image	Subdirectory for icon images
share	Subdirectory for agent shared files
gwia.cfg	Internet Agent 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 Internet Agent
rc3.d	Standard Linux location for run-level-3 symbolic links
S99grpwise	Symbolic link to the startup script for the Internet Agent
rc5.d	Standard Linux location for run-level-5 symbolic links
S99grpwise	Symbolic link to the startup script for the Internet Agent

/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
domain.gwia	Domain-specific subdirectory for Internet Agent log files
*mmdd*gwia.*nnn*	Internet Agent log files

See also Section 7.4, "Internet Agent Queue Directory," on page 87.

agents directory

On a Linux server, the Internet Agent 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 Internet Agent executable. You run this executable file to start the Internet Agent. See "Starting the Internet Agent" in "Installing the GroupWise Internet Agent" in the *GroupWise 7 Installation Guide*.

gwia.xml file

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

gwha file

The `gwha` file is the GroupWise High Availability service executable. If the Internet Agent goes down for any reason, the High Availability service automatically restarts it. See "Enabling the High Availability Service for the Linux Agents" in "Installing GroupWise Agents" in the *GroupWise 7 Installation Guide*.

lib directory

The `lib` directory holds GroupWise shared library files.

gwia*xx*.fil files

This file contains all language-specific information for the Internet Agent. The last two characters *xx* are a language code.

*.so files

These files are Linux shared library files that provide information to the Internet Agent executable.

image directory

The `image` directory has subdirectories for agent icons for both color and monochrome displays.

share directory

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

gwia.cfg

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

During installation, the Internet Agent startup file is created in the share directory with the --home startup switch automatically set to the domain directory where the Internet Agent queue directory is located. See "Using Internet Agent Startup Switches" in "Internet Agent" in the *GroupWise 7 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 Internet Agent. For more information about starting the Internet Agent, see "Starting the Internet Agent" in "Installing the GroupWise Internet Agent" in the *GroupWise 7 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 S99grpwise. It is created if you choose during installation to have the Internet 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 script is S99grpwise. It is created if you choose during installation to have the Internet Agent start automatically when the server boots.

/etc/opt/novell/groupwise directory

The /etc/opt directory is the standard location of application configuration files on Linux. Files that configure how the GroupWise Internet Agent 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 Internet Agent if it goes down unexpectedly. See "Enabling the High Availability Service for the Linux Agents" in "Installing GroupWise Agents" in the *GroupWise 7 Installation Guide*.

uid.conf file

The uid.conf file configures the Internet Agent to run as a non-root user. See "Running the Agents as a Non-root User" in "Installing GroupWise Agents" in the *GroupWise 7 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.gwia directory

The *domain*.gwia directory is a domain-specific location for Internet Agent log files.

Within the *domain*.gwia directory, the Internet Agent 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 Internet Agent Log Files" in "Internet Agent" in the *GroupWise 7 Administration Guide*.

The first two digits of the filename 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 Internet Agent log file created on May 18.

8.2.3 Windows Installation Directory

c:\grpwise\gwia	Windows Internet Agent installation directory
gwenv1a.dll	GroupWise Agent Engine
gwia.exe	Internet Agent program
gwia.cfg	Internet Agent configuration file
gwia*nn*.chm	Internet Agent online help file
gwia*nn*.dll	Internet Agent language information file
gwwww1.dll	Customized DLL program for MIME
x*10a.dll	XIS DLL programs
gwxis10.dll	GroupWise XIS DLL program
gwia.xml	XIS agent configuration file

See also Section 7.4, "Internet Agent Queue Directory," on page 87.

c:\grpwise\gwia

On a Windows server, the Internet Agent can be installed in any directory you choose. The default location is c:\grpwise\gwia. The Internet Agent 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 both 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 Internet Agent program. You run this executable file to start the Internet Agent. See "Starting the Internet Agent" in "Installing the GroupWise Internet Agent" in the *GroupWise 7 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 Internet Agent configuration file is located.

gwia*nxx*.chm file

The .chm files contain the online documentation for the Internet Agent. Online help is available by clicking *Help* in the Internet Agent 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 Internet Agent. 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.

x*10a.dll and gwxis10.dll files

The x*10a.dll and gwxis10.dll programs provide XIS capability for the GroupWise agents, so that you can create XML documents to monitor and change the agents as needed.

gwia.xml file

The gwia.xml file is a specialized agent configuration file for use in the XIS environment.

8.3 WebAccess Agent Installation

- Section 8.3.1, "NetWare Installation Directory," on page 126
- Section 8.3.2, "Linux Installation Directory," on page 128
- Section 8.3.3, "Windows Installation Directory," on page 134
- Section 8.3.4, "Document Viewer Agent Working Directory," on page 137

8.3.1 NetWare Installation Directory

`sys:\system`	NetWare WebAccess Agent installation directory
`gwenn5.nlm`	GroupWise 7 Agent Engine
`gwinter.nlm`	WebAccess Agent program
`gwdva.nlm`	Document Viewer Agent program
`strtweb.ncf`	WebAccess Agent load file
`stopweb.ncf`	WebAccess Agent unload file
webac70a`.waa`	WebAccess Agent startup file
`gwdva.dva`	Document Viewer Agent startup file
`x*5.nlm`	XIS NLM programs
`gwxis12.nlm`	GroupWise XIS NLM program
`gwinter.xml`	WebAccess Agent XIS configuration file
`cache`	Subdirectory for WebAccess template file storage
`template\`*xx*	Subdirectory for HTML template files
`webacc.tpt`	WebAccess template files
`publish.tpt`	WebPublisher template files
`gwdva.dir`	Document Viewer Agent working directory
`help`	Subdirectory for GroupWise agent console help files

Additional WebAccess files are integrated into your Web server to support the WebAccess client and the WebAccess Agent Web console. See Section 8.5, "Apache/Tomcat Installation," on page 143. See also Section 7.5, "WebAccess Agent Queue Directory," on page 93.

NetWare WebAccess Agent installation directory

On a NetWare server, the GroupWise WebAccess Agent is typically installed in the `sys:\system` directory. You can choose a different location during installation.

gwenn5.nlm file

The `gwenn5.nlm` file is the GroupWise 7 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 loaded on a server automatically loads the GroupWise Agent Engine.

gwinter.nlm file

The `gwinter.nlm` file is the WebAccess Agent program.

gwdva.nlm file

The `gwdva.nlm` file is the Document Viewer Agent program.

strtweb.ncf file

The `strtweb.ncf` file is the NetWare configuration file that automatically loads the WebAccess Agent whenever the server is restarted. A typical `strtweb.ncf` file would look similar to the following:

```
load sys:\system\gwinter @webac70a.waa
```

It includes the WebAccess Agent startup file in the load command. For more information, see "NetWare and Windows: Starting GroupWise WebAccess" in "Installing GroupWise WebAccess" in the *GroupWise 7 Installation Guide*.

stopweb.ncf

The `stopweb.ncf` file is the NetWare configuration file that unloads the WebAccess Agent. A typical `stopweb.ncf` file would look similar to the following:

```
unload gwinter
```

webac70a.waa file

The WebAccess Agent startup file contains startup switches for the WebAccess Agent. Switch settings placed in the WebAccess Agent startup file override comparable options set for the WebAccess Agent in ConsoleOne. The startup file is named the same as the WebAccess Agent object in ConsoleOne. The default is `webac70a`.

During installation, a customized version of the WebAccess Agent startup file is created. This customized version has the /home startup switch automatically set to the domain directory where the WebAccess Agent's queue directories are located. See "Using WebAccess Startup Switches" in "WebAccess" in the *GroupWise 7 Administration Guide*.

gwdva.dva file

The `gwdva.dva` file is the Document Viewer Agent startup file. The Viewer Agent is currently configured completely by startup switches. For more information, see "Document Viewer Agent Startup Switches".

x*5.nlm and gwxis12.nlm files

The `x*5.nlm` and `gwxis12.nlm` programs provide XIS capability for the GroupWise agents, so that you can create XML documents to monitor and change the agents as needed.

gwinter.xml file

The `gwinter.xml` file is a specialized WebAccess Agent configuration file for use in the XIS environment.

cache\template\xx directory

The `cache\template\`*`xx`* directory holds language-specific HTML template files that the WebAccess Agent passes to the Document Viewer Agent for viewing documents in documents in WebAccess and WebPublisher. The *xx* is a two-letter language code.

gwdva.dir directory

See Section 8.3.4, "Document Viewer Agent Working Directory," on page 137.

help directory

The `help` directory holds help files and images used by the WebAccess Agent Web console. See "Using the WebAccess Agent Web Console" in "WebAccess" in the *GroupWise 7 Administration Guide*.

8.3.2 Linux Installation Directory

📁 `/opt/novell/groupwise`	Novell GroupWise installation directory
📁 `agents`	Linux agent installation directory
📁 `bin`	Subdirectory for GroupWise agent executables
🗋 `gwinter`	WebAccess Agent executable
🗋 `gwinter.xml`	WebAccess Agent XIS configuration file
🗋 *`webac70axx`*`.waa`	Boilerplate WebAccess startup file
📁 `template`	Subdirectory for HTML template files
🗋 `gwdva`	Document Viewer Agent executable
📁 `gwdva.dir`	Document Viewer Agent working directory

lib	Subdirectory for GroupWise agent library files
*.so	Supporting shared library files
image	Subdirectory for icon images
share	Subdirectory for agent shared files
webac70a.waa	WebAccess Agent startup file
gwdva.dva	Document Viewer Agent startup file
webcon	Subdirectory for agent Web console files
help	Subdirectory for agent Web console help files
webaccess	Subdirectory for WebAccess Application files
webacc.cfg	WebAccess Application configuration file
commgr.cfg	Communications Manager configuration file
spellchk.cfg	Novell Speller Application configuration file
ldap.cfg	LDAP service provider configuration file
default	Subdirectory for boilerplate configuration files
logs	Subdirectory for WebAccess Application log files
users	Subdirectory for users' WebAccess session information
webpublisher	Subdirectory for WebAccess Application files
webpub.cfg	WebPublisher Application configuration file
commgr.cfg	Communications Manager configuration file
default	Subdirectory for boilerplate configuration files
logs	Subdirectory for WebPublisher Application log files

/etc/init.d	Standard Linux location for application startup scripts
grpwise-wa	Startup script for the WebAccess Agent
rc3.d	Standard Linux location for run-level-3 symbolic links
S99grpwise-wa	Symbolic link to the startup script for the WebAccess Agent
rc5.d	Standard Linux location for run-level-5 symbolic links
S99grpwise-wa	Symbolic link to the startup script for the WebAccess Agent

/var/log/	Standard Linux location for application log files
novell/groupwise	Subdirectory for GroupWise agent log files

domain.webac70a	Domain-specific subdirectory for WebAccess Agent log files
000.prc	Processing directory for log files
mmddgwia.nnn	WebAccess Agent log files
gwdva	Subdirectory for Viewer Agent log files
mmdddva.nnn	Viewer Agent log files

Additional WebAccess files are integrated into your Web server to support the WebAccess client and the WebAccess Agent Web console. See Section 8.5, "Apache/Tomcat Installation," on page 143. See also Section 7.5, "WebAccess Agent Queue Directory," on page 93.

agents directory

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

bin directory

The bin directory holds GroupWise executable files.

gwinter file

The gwinter file is the WebAccess Agent executable. You run this executable file to start the WebAccess Agent. See "Linux: Setting Up GroupWise WebAccess" in "Installing GroupWise WebAccess" in the *GroupWise 7 Installation Guide*.

gwinter.xml file

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

webac70axx.waa file

The *webac70axx*.waa file is the boilerplate file from which a domain-specific *webac70a*.waa file is created in the share directory. The *webac70a* represents the name of the WebAccess Agent object in eDirectory. The *xx* represents a two-letter language code.

template\xx directory

The template\xx directory holds language-specific HTML template files that the WebAccess Agent passes to the Document Viewer Agent for viewing documents in WebAccess and WebPublisher. The *xx* is a two-letter language code.

gwdva file

The gwdva file is the Document Viewer Agent executable. The WebAccess Agent starts and stops the Viewer Agent automatically whenever it starts and stops.

gwdva.dir directory

See Section 8.3.4, "Document Viewer Agent Working Directory," on page 137.

lib directory

The lib directory holds GroupWise shared library files.

*.so files

The *.so files are Linux shared library files that provide information to the WebAccess Agent executable.

image directory

The image directory has subdirectories for agent icons for both color and monochrome displays.

share directory

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

webac70a.waa file

The WebAccess Agent startup file contains startup switches for the WebAccess Agent. Switch settings placed in the WebAccess Agent startup file override comparable options set for the WebAccess Agent in ConsoleOne. The startup file is named the same as the WebAccess Agent object in ConsoleOne. The default is webac70a.

During installation, a customized version of the WebAccess Agent startup file is created. This customized version has the /home startup switch automatically set to the domain directory where the WebAccess Agent's queue directories are located. See "Using WebAccess Startup Switches" in "WebAccess" in the *GroupWise 7 Administration Guide*.

gwdva.dva file

The gwdva.dva file is the Document Viewer Agent startup file. The Viewer Agent is currently configured completely by startup switches. For more information, see "Document Viewer Agent Startup Switches".

webcon directory

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

webaccess directory

The webaccess directory holds files used by the WebAccess Application.

webacc.cfg file

The webacc.cfg file holds the same configuration information that is stored on the WebAccess Application object (named GroupWiseWebAccess) in eDirectory. For more information, see "Configuring the WebAccess Application" in "WebAccess" in the *GroupWise 7 Administration Guide*.

commgr.cfg file

The commgr.cfg file holds the same configuration information that is stored on the GroupWise Provider object (named GroupWiseProvider) in eDirectory, including the IP address and encryption key for the WebAccess Agent. This information enables the WebAccess Application to communicate with the WebAccess Agent. For more information, see "Configuring the GroupWise Service Provider" in "WebAccess" in the *GroupWise 7 Administration Guide*.

A copy of the commgr.cfg file is automatically maintained in the WebAccess Agent queue directory (*domain/wpgate/webac70a*).

spellchk.cfg file

The spellchk.cfg file holds the same configuration information that is stored on the Novell Speller Application object (named NovellSpeller) in eDirectory. For more information, see "Configuring the Novell Speller Application" in "WebAccess" in the *GroupWise 7 Administration Guide*.

ldap.cfg file

The ldap.cfg file holds the same configuration information that is stored on the LDAP Provider object (named LDAPProvider) in eDirectory. For more information, see "Configuring the LDAP Service Provider" in "WebAccess" in the *GroupWise 7 Administration Guide*.

default directory

The default directory holds boilerplate versions of the various .cfg files.

logs directory

The logs subdirectory holds WebAccess Application log files. For more information, see "Controlling WebAccess Application Logging" in "WebAccess" in the *GroupWise 7 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. Session files are deleted when users log back in.

webpublisher directory

The webpublisher directory holds files used by the WebPublisher Application.

webpub.cfg

The webpub.cfg file holds the same configuration information that is stored on the WebPublisher Application object (named GroupWiseWebPublisher) in eDirectory. For more information, see "Configuring the WebPublisher Application" in "WebAccess" in the *GroupWise 7 Administration Guide*.

commgr.cfg file

The `commgr.cfg` file holds the same configuration information that is stored on the GroupWise Provider object (named GroupWiseProvider) in eDirectory, including the IP address, port number, number of threads, and encryption key for the WebAccess Agent. This information enables the WebAccess Application to communicate with the WebAccess Agent. For more information, see "Configuring the GroupWise Service Provider" in "WebAccess" in the *GroupWise 7 Administration Guide*.

A copy of the `commgr.cfg` file is automatically maintained in the WebAccess Agent queue directory (*`domain`*`/wpgate/`*`webac70a`*).

default directory

The `default` directory holds boilerplate versions of the `webpub.cfg` and `commgr.cfg` files.

logs directory

The `logs` subdirectory holds WebPublisher Application log files. For more information, see "Controlling WebAccess Application Logging" in "WebAccess" in the *GroupWise 7 Administration Guide*.

/etc/init.d directory

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

grpwise-wa file

The `grpwise-wa` script is created automatically during installation. You can use the script to start, restart, stop, and display status information about the WebAccess Agent. For more information about starting the WebAccess Agent, see "Linux: Setting Up GroupWise WebAccess" in "Installing GroupWise WebAccess" in the *GroupWise 7 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-wa` script is `S99grpwise-wa`. It is created if you choose during installation to have the WebAccess 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-wa` script is `S99grpwise-wa`. It is created if you choose during installation to have the WebAccess Agent start automatically when the server boots.

/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.webac70a directory

The *`domain`*`.`*`webac70a`* directory is a domain-specific location for WebAccess Agent log files.

000.prc directory

Within the `000.prc` directory, the WebAccess Agent creates log files (*mmdd*`web.`*nnn*) to inform you of its processing and any problems it encounters. For more information about log files, see "Controlling WebAccess Agent Logging" in "WebAccess" in the *GroupWise 7 Administration Guide*.

The first two digits of the filename 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, `0518web.002` is the second WebAccess Agent log file created on May 18.

gwdva directory

The `gwdva` directory stores Viewer Agent log files. The first two digits of the filename 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, `0518web.002` is the second Viewer Agent log file created on May 18.

8.3.3 Windows Installation Directory

c:\webacc	Windows WebAccess Agent installation directory
gwenv1a.dll	GroupWise Agent Engine
gwinter.exe	WebAccess Agent program
strtweb.bat	WebAccess Agent batch file
webac70a.waa	WebAccess Agent startup file
gwdva.exe	Document Viewer Agent program
gwdva.dva	Document Viewer Agent startup file
gwdva.dir	Document Viewer Agent working directory
gwwan*xx*.dll	WebAccess Agent language information file
gwwasnmp.dll	Customized DLL program for SNMP
gwwww1.dll	Customized DLL program for MIME
x*10a.dll	XIS DLL programs
gwxis10.dll	GroupWise XIS DLL program
gwinter.xml	WebAccess Agent XIS configuration file
cache\template\xx	Subdirectory for HTML template files
webacc.tpt	WebAccess template files
publish.tpt	WebPublisher template files
help	Subdirectory for GroupWise agent console help files

Additional WebAccess files are integrated into your Web server to support the WebAccess client and the WebAccess Agent Web console. See Section 8.5, "Apache/Tomcat Installation," on page 143. See also Section 7.5, "WebAccess Agent Queue Directory," on page 93.

c:\webacc

On a Windows server, the WebAccess Agent can be installed in any directory you choose. The default location is `c:\webacc`. The WebAccess 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.

gwinter.exe file

The `gwinter.exe` file is the WebAccess Agent program.

strtweb.bat file

The `strtweb.bat` file is used to start the WebAccess Agent. The batch file specifies the WebAccess startup file. A typical `strtweb.bat` file includes the following:

```
title Novell GroupWise WebAccess
@echo off
cls
gwinter.exe @webac70a.waa
```

You run this batch file to start the WebAccess Agent. See "NetWare and Windows: Starting GroupWise WebAccess" in "Installing GroupWise WebAccess" in the *GroupWise 7 Installation Guide*.

webac70a.waa file

The WebAccess Agent startup file contains startup switches for the WebAccess Agent. Switch settings placed in the WebAccess Agent startup file override comparable configuration options set for the WebAccess Agent in ConsoleOne. The startup file is named the same as the WebAccess Agent object in eDirectory. The default is `webac70a`.

During installation, the WebAccess Agent startup file is created with the /home startup switch automatically set to the domain directory where the WebAccess Agent queue directory (*domain*\wpgate*webac70a*) is located. See "Using WebAccess Startup Switches" in "WebAccess" in the *GroupWise 7 Administration Guide*.

gwdva.exe file

The gwdva.exe file is the Document Viewer Agent program. The WebAccess Agent automatically starts and stops the Viewer Agent whenever the WebAccess Agent starts and stops.

gwdva.dva file

The gwdva.dva file is the Document Viewer Agent startup file. The Viewer Agent is currently configured completely by startup switches. See "Document Viewer Agent Startup Switches".

gwdva.dir directory

See Section 8.3.4, "Document Viewer Agent Working Directory," on page 137.

gwwa*nxx*.dll files

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

gwwasnmp.dll file

The gwwasnmp.dll file provides interaction with the Windows SNMP Service, so that you can monitor the WebAccess Agent using an SNMP monitoring program. This file is not installed unless you enable SNMP during installation. It cannot be installed if the SNMP Service is not installed on the Windows server.

gwwww1.dll file

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

x*10a.dll and gwxis10.dll files

The x*10a.dll and gwxis10.dll programs provide XIS capability for the GroupWise agents, so that you can create XML documents to monitor and change the agents as needed.

gwinter.xml file

The gwia.xml file is a specialized agent configuration file for use in the XIS environment.

cache\template\xx directory

The cache\template\xx directory holds language-specific HTML template files that the WebAccess Agent passes to the Document Viewer Agent for viewing documents in WebAccess and WebPublisher. The *xx* is a two-letter language code.

help directory

The help directory holds help files and images used by the WebAccess Agent Web console. See "Using the WebAccess Agent Web Console" in "WebAccess" in the *GroupWise 7 Administration Guide*.

8.3.4 Document Viewer Agent Working Directory

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

NetWare:	`sys:\system\gwdva.dir`
Linux:	`/opt/novell/groupwise/agents/bin/gwdva.dir`
Windows:	`c:\webacc\gwdva.dir`

gwdva.dir	Document Viewer Agent working directory
cache	Subdirectory for caching converted documents for repeated display
000	Subdirectory for converted GroupWise library documents
tran	Subdirectory for converted attachment documents
problem	Subdirectory for information about documents that failed conversion
hold	Subdirectory for quarantined documents that failed HTML conversion
log	Subdirectory for Document Viewer Agent log files
temp	Subdirectory for the temporary files using during HTML conversion
template	(not currently used)

gwdva.dir directory

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

cache directory

The `cache` directory contains subdirectories for managing the documents that the Viewer Agent has already converted. See "Document Cache" in "WebAccess" in the *GroupWise 7 Administration Guide*.

000 directory

The `000` directory holds GroupWise library documents that have been converted to HTML. Because these documents belong to a GroupWise library, they are retained for a longer period of time. If the documents in the `000` directory exceed the established maximum cache size, a new directory, `001`, is created. Additional library cache directories are created as needed.

tran directory

The `tran` directory holds attachment documents that have been converted to HTML. Attached documents are more transitory than GroupWise library documents and are retained for a shorter period of time.

problem directory

The `problem` directory stores information about documents that have failed conversion. This prevents the Viewer Agent from needlessly attempting to convert the same failed documents.

hold directory

The `hold` directory is the document quarantine directory. Here the Viewer Agent places documents that have failed HTML conversion. You can configure the Viewer Agent to notify an administrator when a document fails conversion. See “Document Quarantine” in “WebAccess” in the *GroupWise 7 Administration Guide*

log directory

The `log` directory stores log files produced by the Viewer Agent. See “Agent Log Files” in “WebAccess” in the *GroupWise 7 Administration Guide*.

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

temp directory

The `temp` directory is used for the temporary files created during HTML conversion. See “Document Conversion” in “WebAccess” in the *GroupWise 7 Administration Guide*.

8.4 Monitor Agent Installation

- Section 8.4.1, “Linux Installation Directory,” on page 138
- Section 8.4.2, “Windows Installation Directory,” on page 141

8.4.1 Linux Installation Directory

📁 `/opt/novell/groupwise`	GroupWise installation directory
📁 `agents`	Linux agent installation directory
📁 `bin`	Subdirectory for GroupWise agent executables
🗋 `gwmon`	Monitor Agent executable
📁 `lib`	Subdirectory for GroupWise agent library files
🗋 `*.so`	Supporting shared library files
📁 `share`	Subdirectory for agent shared files
🗋 `monitor.xml`	Monitor Agent configuration file
📁 `gwmonitor`	Subdirectory for Monitor Application files

🗋 gwmonitor.cfg	Monitor Application configuration file
📁 default	Subdirectory for boilerplate configuration files
📁 logs	Subdirectory for Monitor Application log files
📁 users	Directory for Monitor users' session files

📁 /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
🗋 S99grpwise-ma	Symbolic link to the startup script for the Monitor Agent
📁 rc5.d	Standard Linux location for run-level-5 symbolic links
🗋 S99grpwise-ma	Symbolic 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

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 Monitor Agent" in "Monitor" in the *GroupWise 7 Administration Guide*.

lib directory

The lib directory holds GroupWise shared library files.

*.so files

These files are Linux shared library files that provide information to the Monitor Agent executable.

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.

gwmonitor directory

The `gwmonitor` directory holds files used by the Monitor Application.

gwmonitor.cfg file

The `gwmonitor.cfg` file holds the same configuration information that is stored on the Monitor Application object (named GroupWiseMonitor) in eDirectory. For more information, see "Configuring the Monitor Application" in "Monitor" in the *GroupWise 7 Administration Guide*.

default directory

The `default` directory holds boilerplate versions of the gwmonitor.cfg file and the monitor.xml file.

logs directory

The `logs` subdirectory holds Monitor Application log files. For more information, see "Modifying Monitor Application Log Settings" in "Monitor" in the *GroupWise 7 Administration Guide*.

users directory

The `users` subdirectory holds session files for Monitor users.

/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 Monitor Agent" in "Monitor" in the *GroupWise 7 Administration 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 `S99grpwise-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 S99grpwise-ma. It is created if you choose during installation to have the Monitor Agent start automatically when the server boots.

/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 7 Administration Guide*.

The first two digits of the filename 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.

8.4.2 Windows Installation Directory

c:\gwmon	Windows Monitor Agent installation directory
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
x*10a.dll	XIS DLL programs
gwxis10.dll	GroupWise XIS DLL program
gwmon.xml	Monitor Agent XIS configuration file
monwork	Monitor Agent working directory

Additional Monitor files are integrated into your Web server to support the Monitor Web console. For an example, see Section 8.5, "Apache/Tomcat Installation," on page 143.

c:\gwmon

On a Windows server, the Monitor Agent can be installed in any directory you choose. The default location is c:\gwmon. 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 "Starting the Monitor Agent" in "Monitor" in the *GroupWise 7 Administration 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 7 Administration Guide*.

gwmon*xx*.dll files

The gwmon*xx*.dll files contain all language-specific information for the WebAccess 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 7 Administration Guide*.

x*10a.dll and gwxis10.dll files

The x*10a.dll and gwxis10.dll programs provide XIS capability for the GroupWise agents, so that you can create XML documents to monitor and change the agents as needed.

gwmon.xml file

The gwmon.xml file is a specialized agent configuration file for use in the XIS environment.

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 7 Administration Guide*

8.5 Apache/Tomcat Installation

WebAccess, WebPublisher, 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 7 Installation Guide*. In this section, the Apache Web server and the Tomcat Servlet Engine used to illustrate how GroupWise integrates with your Web server.

- Section 8.5.1, “NetWare Installation Directory,” on page 143
- Section 8.5.2, “Linux Installation Directory,” on page 148

8.5.1 NetWare Installation Directory

📁 `sys:\apache2`	Apache installation directory
📁 `bin`	Standard Apache directory
📁 `cgi-bin`	Standard Apache directory
📁 `conf`	Standard Apache directory
🗋 `httpd.conf`	Main Apache configuration file
🗋 `GWApache2.conf`	Specialized Apache configuration file for use with GroupWise
📁 `error`	Standard Apache directory
📁 `htdocs`	Apache document root directory
🗋 `index.html`	Web server default Web page
📁 `icons`	Standard Apache directory
📁 `logs`	Standard Apache directory
📁 `man`	Standard Apache directory
📁 `manual`	Standard Apache directory
📁 `modules`	Standard Apache directory

📁 `sys:\tomcat\4`	Tomcat installation directory
📁 `bin`	Standard Tomcat directory
📁 `common`	Standard Tomcat directory
📁 `conf`	Standard Tomcat directory
📁 `examples`	Standard Tomcat directory
📁 `logs`	Standard Tomcat directory
📁 `server`	Standard Tomcat directory
📁 `shared`	Standard Tomcat directory
📁 `temp`	Standard Tomcat directory

webapps/ROOT	Standard Tomcat document root directory
webapps/gw	GroupWise-specific document root directory
index.html	GroupWise default Web services page
com/novell	Directory for Novell product files
collexion	Directory for Novell class files and properties files
GWMonitor	Directory for GroupWise Monitor files
help	Subdirectory for Monitor Web console help files
images	Subdirectory for Monitor Web console image files
images	Directory for Novell product image files
inso	Directory for Inso* viewer files for viewing attachments
webaccess	Directory for GroupWise WebAccess files
applets	Subdirectory for WebAccess client properties files
help	Subdirectory for WebAccess client help files
images	Subdirectory for WebAccess client image files
webconsole	Subdirectory for WebAccess Application Web console help files
webpublisher	Directory for GroupWise WebPublisher files
help	Subdirectory for WebPublisher help files
images	Subdirectory for WebPublisher image files
WEB-INF	Web server information directory
web.xml	Web server configuration file
classes/com/novell	Directory for Novell product files
collexion	Directory for language-specific morphology files
GWMonitor	Directory for GroupWise Monitor files
providers	Subdirectory for service provider files
templates	Subdirectory for template files
webaccess	Directory for GroupWise WebAccess files
providers	Subdirectory for service provider files
templates	Subdirectory for template files
webpublisher	Directory for GroupWise WebPublisher files
providers	Subdirectory for service provider files
templates	Subdirectory for template files

📁 `work`	Standard Tomcat directory

📁 `sys:\novell`	Directory for Novell product configuration files
📁 `gwmonitor`	Subdirectory for GroupWise Monitor
📄 `gwmonitor.cfg`	Monitor Application configuration file
📁 `logs`	Directory for Monitor Application log files
📁 `webaccess`	Subdirectory for WebAccess
📄 `webacc.cfg`	WebAccess Application configuration file
📄 `commgr.cfg`	Communications Manager configuration file
📄 `spellchk.cfg`	Novell Speller Application configuration file
📄 `ldap.cfg`	LDAP service provider configuration file
📁 `logs`	Directory for WebAccess Application log files
📁 `temp`	Temporary working directory
📁 `users`	Directory for WebAccess users' session files
📁 `webpublisher`	Subdirectory for WebAccess Application files
📄 `webpub.cfg`	WebPublisher Application configuration file
📄 `commgr.cfg`	Communications Manager configuration file
📁 `default`	Subdirectory for boilerplate configuration files
📁 `logs`	Subdirectory for WebPublisher Application log files

conf/httpd.conf file

The `httpd.conf` file is the main Apache configuration file. On NetWare, the GroupWise Installation program modifies it to include a reference to the `GWApache2.conf` file.

conf/GWApache2.conf file

The `GWApache2.conf` file includes the jkmount command that enables Apache and Tomcat to communicate with each other.

htdocs/index.html file

The `index.html` file in the `htdocs` directory is the default HTML file that displays in your Web browser when you go to the IP address or hostname of the server.

webapps/gw/index.html file

The `index.html` file in the `gw` directory is the default HTML file for GroupWise Web services (WebAccess, WebPublisher, and Monitor) and displays when you go to the following URL:

http://*web_server_address*/gw

From this default page, specific GroupWise URLs can be easily accessed:

http://*web_server_address*/gw/webacc
http://*web_server_address*/gw/webpub
http://*web_server_address*/gw/gwmonitor

WEB-INF/web.xml file

The `web.xml` file tells the Web server where the various GroupWise service providers and their configuration files are located. By default, they are located in subdirectories under `sys:\novell`.

GWMonitor/providers directory

The `providers` directory holds the Java* class and properties files for the Monitor provider. For more information, see "Adding or Removing Service Providers" in "Monitor" in the *GroupWise 7 Administration Guide*.

GWMonitor/templates directory

The `templates` directory holds subdirectories for the various file formats used in various browser environments. For more information, see "Modifying Monitor Application Template Settings" in "Monitor" in the *GroupWise 7 Administration Guide*.

webaccess/providers directory

The `providers` directory holds the Java class and properties files for the WebAccess providers. For more information, see "Adding or Removing Service Providers" in "WebAccess" in the *GroupWise 7 Administration Guide*.

webaccess/templates directory

The `templates` directory holds subdirectories for the various file formats used in various browser environments. For more information, see "Modifying WebAccess Application Template Settings" in "WebAccess" in the *GroupWise 7 Administration Guide*.

webpublisher/providers directory

The `providers` directory holds the Java class and properties files for the WebPublisher providers. For more information, see "Adding or Removing Service Providers" in "WebAccess" in the *GroupWise 7 Administration Guide*.

webpublisher/templates directory

The `templates` directory holds subdirectories for the various file formats used in various browser environments. For more information, see "Modifying WebPublisher Application Template Settings" in "WebAccess" in the *GroupWise 7 Administration Guide*.

sys:\novell directory

The `sys:\novell` directory holds subdirectories for each Novell product that uses application configuration files.

gwmonitor directory

The gwmonitor directory is created when you install GroupWise Monitor.

gwmonitor.cfg file

The gwmonitor.cfg file holds the same configuration information that is stored on the Monitor Application object (named GroupWiseMonitor) in eDirectory. For more information, see "Configuring the Monitor Application" in "Monitor" in the *GroupWise 7 Administration Guide*.

logs directory

The logs subdirectory holds Monitor Application log files. For more information, see "Modifying Monitor Application Log Settings" in "Monitor" in the *GroupWise 7 Administration Guide*.

webaccess directory

The webaccess directory holds files used by the WebAccess Application.

webacc.cfg file

The webacc.cfg file holds the same configuration information that is stored on the WebAccess Application object (named GroupWiseWebAccess) in eDirectory. For more information, see "Configuring the WebAccess Application" in "WebAccess" in the *GroupWise 7 Administration Guide*.

commgr.cfg file

The commgr.cfg file holds the same configuration information that is stored on the GroupWise Provider object (named GroupWiseProvider) in eDirectory, including the IP address, port number, number of threads, and encryption key for the WebAccess Agent. This information enables the WebAccess Application to communicate with the WebAccess Agent. For more information, see "Configuring the GroupWise Service Provider" in "WebAccess" in the *GroupWise 7 Administration Guide*.

A copy of the commgr.cfg file is automatically maintained in the WebAccess Agent queue directory (*domain*\wpgate*webac70a*).

spellchk.cfg file

The spellchk.cfg file holds the same configuration information that is stored on the Novell Speller Application object (named NovellSpeller) in eDirectory. For more information, see "Configuring the Novell Speller Application" in "WebAccess" in the *GroupWise 7 Administration Guide*.

ldap.cfg file

The ldap.cfg file holds the same configuration information that is stored on the LDAP Provider object (named LDAPProvider) in eDirectory. For more information, see "Configuring the LDAP Service Provider" in "WebAccess" in the *GroupWise 7 Administration Guide*.

logs directory

The `logs` subdirectory holds WebAccess Application log files. For more information, see "Controlling WebAccess Application Logging" in "WebAccess" in the *GroupWise 7 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.

webpublisher directory

The `webpublisher` directory holds files used by the WebPublisher Application.

webpub.cfg

The `webpub.cfg` file holds the same configuration information that is stored on the WebPublisher Application object (named GroupWiseWebPublisher) in eDirectory. For more information, see "Configuring the WebPublisher Application" in "WebAccess" in the *GroupWise 7 Administration Guide*.

commgr.cfg file

The `commgr.cfg` file holds the same configuration information that is stored on the GroupWise Provider object (named GroupWiseProvider) in eDirectory, including the IP address, port number, number of threads, and encryption key for the WebAccess Agent. This information enables the WebAccess Application to communicate with the WebAccess Agent. For more information, see "Configuring the GroupWise Service Provider" in "WebAccess" in the *GroupWise 7 Administration Guide*.

A copy of the `commgr.cfg` file is automatically maintained in the WebAccess Agent queue directory (*domain*`\wpgate\`*webac70a*).

default directory

The `default` directory holds boilerplate versions of the `webpub.cfg` and `commgr.cfg` files.

logs directory

The `logs` subdirectory holds WebAccess Application log files. For more information, see "Controlling WebAccess Application Logging" in "WebAccess" in the *GroupWise 7 Administration Guide*.

8.5.2 Linux Installation Directory

`/opt/novell/httpd`	Novell Apache installation directory
`modules`	Standard Apache directory

📁 /etc/opt/novell/httpd	Novell Apache configuration directory
📁 conf.d	Standard Apache directory
📁 gconf.d	Standard Apache directory
📁 sslconf.d	Standard Apache directory

📁 /etc/opt/novell/tomcat4	Novell Tomcat configuration directory
🗋 catalina.properties	Standard Tomcat file
🗋 jk2.properties	Standard Tomcat file
🗋 server.xml	Standard Tomcat file
🗋 tomcat-users.xml	Standard Tomcat file
🗋 tomcat4.conf	Standard Tomcat file
🗋 web.xml	Web server configuration file
📁 init.d	Standard Tomcat directory
📁 logrotate.d	Standard Tomcat directory

📁 /var/opt/novell	Novell subdirectory under /var
📁 gw	Apache document root directory for WebAccess and WebPublisher
🗋 index.html	Web server default Web page
📁 com/novell	Directory for Novell product files
📁 collexion	Directory for Novell class files and properties files
📁 images	Directory for Novell product image files
📁 inso	Directory for Inso viewer files for viewing attachments
📁 webaccess	Directory for GroupWise WebAccess files
📁 applets	Subdirectory for WebAccess client properties files
📁 help	Subdirectory for WebAccess client help files
📁 images	Subdirectory for WebAccess client image files
📁 webconsole	Subdirectory for WebAccess Application Web console help files
📁 webpublisher	Directory for GroupWise WebPublisher files
📁 help	Subdirectory for WebPublisher help files
📁 images	Subdirectory for WebPublisher image files

`WEB-INF`	Web server information directory
`web.xml`	Web server configuration file
`classes/com/novell`	Web server configuration file
`collexion`	Directory for language-specific morphology files
`webaccess`	Directory for GroupWise WebAccess files
`providers`	Subdirectory for service provider files
`templates`	Subdirectory for template files
`webpublisher`	Directory for GroupWise WebPublisher files
`providers`	Subdirectory for service provider files
`templates`	Subdirectory for template files
`gwmon`	Apache document root directory for Monitor
`index.html`	Web server default page
`com/novell`	Directory for Novell product files
`gwmonitor`	Directory for GroupWise WebPublisher files
`help`	Subdirectory for WebPublisher help files
`images`	Subdirectory for WebPublisher image files
`WEB-INF`	Web server information directory
`web.xml`	Web server configuration file
`gwmonitor`	Directory for GroupWise Monitor files
`providers`	Subdirectory for service provider files
`templates`	Subdirectory for template files
`httpd`	Standard Apache directory
`log`	Standard Apache directory
`novlwww`	Home directory for the novlwww user created at installation
`run`	Standard Apache directory

`tomcat4`	Standard Tomcat directory
`bin`	Standard Tomcat directory
`common`	Standard Tomcat directory
`conf`	--> `/etc/opt/novell/tomcat4`
`logs`	Standard Tomcat directory
`server`	Standard Tomcat directory
`shared`	Standard Tomcat directory
`temp`	Standard Tomcat directory
`webapps`	Standard Tomcat directory
`admin.xml`	Standard Tomcat file
`manager.xml`	Standard Tomcat file
`examples`	Standard Tomcat directory
`gw`	--> `/var/opt/novell/gw`
`gwmon`	--> `/var/opt/novell/gwmon`
`tomcat-docs`	Standard Tomcat directory
`webdav`	Standard Tomcat directory

gw/index.html file

The `index.html` file in the `gw` directory is the default HTML file for GroupWise Web services (WebAccess and WebPublisher) and displays when you go to the following URL:

http://*web_server_address*/gw

From this default page, specific GroupWise URLs can be easily accessed:

http://*web_server_address*/gw/webacc
http://*web_server_address*/gw/webpub

gw/WEB-INF/web.xml file

The `web.xml` file in the `gw/WEB-INF` directory tells the Web server where the various WebAccess service providers and their configuration files are located. By default, they are located in `/opt/novell/groupwise/webaccess`.

webaccess/providers directory

The `providers` directory holds the Java class and properties files for the WebAccess providers. For more information, see "Adding or Removing Service Providers" in "WebAccess" in the *GroupWise 7 Administration Guide*.

webaccess/templates directory

The `templates` directory holds subdirectories for the various file formats used in various browser environments. For more information, see "Modifying WebAccess Application Template Settings" in "WebAccess" in the *GroupWise 7 Administration Guide*.

webpublisher/providers directory

The providers directory holds the Java class and properties files for the WebPublisher providers. For more information, see "Adding or Removing Service Providers" in "WebAccess" in the *GroupWise 7 Administration Guide*.

webpublisher/templates directory

The templates directory holds subdirectories for the various file formats used in various browser environments. For more information, see "Modifying WebPublisher Application Template Settings" in "WebAccess" in the *GroupWise 7 Administration Guide*.

gwmon/index.html file

The index.html file in the gwmon directory is the default HTML file for the GroupWise Monitor Web service and displays when you go to the following URL:

http://*web_server_address*/gwmon

From this default page, the GroupWise Monitor login page can be easily accessed:

http://*web_server_address*/gwmon/gwmonitor

gwmon/WEB-INF/web.xml file

The web.xml file in the gwmon/WEB-INF directory tells the Web server where the Monitor service provider and its configuration file are located. By default, they are located in /opt/novell/groupwise/gwmonitor.

gwmonitor/providers directory

The providers directory holds the Java class and properties files for the Monitor provider. For more information, see "Adding or Removing Service Providers" in "Monitor" in the *GroupWise 7 Administration Guide*.

gwmonitor/templates directory

The templates directory holds subdirectories for the various file formats used in various browser environments. For more information, see "Modifying Monitor Application Template Settings" in "Monitor" in the *GroupWise 7 Administration Guide*.

Software Distribution Directory

9

- Section 9.1, "NetWare/Windows Software Distribution Directory," on page 153
- Section 9.2, "Linux Software Distribution Directory," on page 160

9.1 NetWare/Windows Software Distribution Directory

software_distribution directory	Master copy of GroupWise® software
agents	GroupWise agent software
install.exe	GroupWise Agent Installation program
nlm	NetWare® agent software
nt	Windows agent software
startups	Agent startup files
snmp	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 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 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 7, 6.*x*, and 5.*x* post office databases

client	GroupWise client software
win32	GroupWise client for Windows
setup.exe	GroupWise client installation program
setup.cfg	Configuration file for the AutoUpdate feature
addons	GroupWise client addon programs
gwmailto	Program to make GroupWise your browser e-mail program
gwtip	Program to display tips each time GroupWise starts
pdaconnect	Program to synchronize your PDA with GroupWise
gwcheck	GroupWise Check utility
ofviews	GroupWise client view files
ppforms	GroupWise client day planner forms
admin	GroupWise administrator software
c1admin	GroupWise snap-ins to ConsoleOne®
monitor	GroupWise Monitor software
utility	GroupWise administrative utilities
gwcheck	GroupWise Check utility
gwcsrgen	GroupWise Generate CSR utility
setupip	GroupWise TCP/IP Setup utility
internet	Internet connectivity software
gwia	GroupWise Internet Agent software
webaccess	GroupWise WebAccess software
license	GroupWise Software License Agreement in multiple languages
common	Java files used by GroupWise programs
docs	GroupWise *Installation Guide*

9.1.1 \grpwise\software directory

The GroupWise software distribution directory resides initially on the GroupWise distribution CD. GroupWise Administration is installed directly from the CD. 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`.

9.1.2 agents directory

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

- Post Office Agent (POA)
- Message Transfer Agent (MTA)
- GroupWise Target Service Agent (GWTSA)

install.exe file

The `install.exe` file in the `agent` subdirectory is the installation program you run to install the GroupWise agents on the servers where you will run the POA and/or MTA. For instructions, see "Installing and Starting the POA" in "Post Office Agent" and "Installing and Starting the MTA" in "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

nlm directory

The `nlm` subdirectory in the `agent` subdirectory contains the GroupWise agent NLM™ files installed on NetWare servers. See "NetWare Installation Directory" on page 107.

nt directory

The `nt` subdirectory contains the GroupWise agent files installed on Windows servers. See "Windows Installation Directory" on page 116.

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 "NetWare Installation Directory" on page 107 or "Windows Installation Directory" on page 116.

snmp directory

The `snmp` subdirectory contains GroupWise MIB files. For more information, see "Using an SNMP Management Console" in "Post Office Agent" and "Message Transfer Agent" in the *GroupWise 7 Administration Guide*.

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 7 Administration Guide*.

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 distribution media.

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 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 7/6.*x*/5.*x* secondary domains. The original gwdom.dc file is located in the domain subdirectory of the GroupWise distribution media.

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 distribution media. 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 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 7/6.*x*/5.*x* post offices in that domain. The original gwpo.dc file is located in the domain directory of the GroupWise CD. 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 CD.

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

9.1.5 client directory

The client subdirectory contains all files associated with GroupWise Windows client. See "Client" in the *GroupWise 7 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 171.

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 7 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 "Installing with the Configuration File (setup.cfg)" in "Client" in the *GroupWise 7 Administration Guide*.

addons directory

The addons directory contains subdirectories for programs that can be used along with the GroupWise client to enhance aspects of its functionality.

gwmailto directory

The gwmailto directory contains a program that makes the GroupWise Windows client the default e-mail application in your Web browser. For more information, see "[GWMAILTO]" in "Modifying the Configuration File" in "Client" in the *GroupWise 7 Administration Guide*.

gwtip directory

The gwtip directory contains a program that display a helpful tip about GroupWise each time the client starts. For more information, see "[GWTIP] " in "Modifying the Configuration File" in "Client" in the *GroupWise 7 Administration Guide*.

pdaconnect directory

The pdaconnect directory contains a program to synchronize data between GroupWise and a PDA device. For more information, see the PDA Connect Readme (http://www.novell.com/documentation/gw7/readmeen_pda_connect/readmeen_pda_connect.html).

9.1.6 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 "Managing Your Mailbox" in the *GroupWise 7 Windows Client User Guide*.

9.1.7 ofviews directory

The ofviews subdirectory contains platform-specific subdirectories of view files for use by the GroupWise client. In addition, the gwview*xx*.ini and ofview*xx*.ini files configure custom views on the menus where users select views. The gwview*xx*.ini file configures GroupWise 7, 6.*x*, and GroupWise 5.5 views. The ofview*xx*.ini file configures views from earlier versions of GroupWise.

9.1.8 ppforms directory

The ppforms subdirectory contains day planner forms for printing GroupWise calendars and tasks.

9.1.9 admin directory

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

c1admin directory

The admin\c1admin subdirectory contains the GroupWise snap-ins to ConsoleOne.

monitor directory

The admin\monitor subdirectory contains the GroupWise 7/6.*x* Monitor program, an SNMP monitoring program for use with the GroupWise agents. See "Monitor" in the *GroupWise 7 Administration Guide*.

utility directory

The admin\utility subdirectory contains helpful GroupWise utilities.

gwcheck directory

The `admin\utility\gwcheck` subdirectory contains the GroupWise Check utility. See "GroupWise Check" in "Standalone Database Maintenance Programs" in the *GroupWise 7 Administration Guide*.

gwcsrgen directory

The `admin\utility\gwcsrgen` subdirectory contains the GroupWise Generate CSR utility. See "Server Certificates and SSL Encryption" in "Security" in the *GroupWise 7 Administration Guide*.

setupip directory

The `admin\utility\setupip` subdirectory contains supporting programs for the client AutoUpdate feature. See "Enabling AutoUpdate" in "Client" in the *GroupWise 7 Administration Guide*.

9.1.10 internet directory

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

gwia directory

The `internet\gwia` subdirectory contains the GroupWise Internet Agent software, used to connect GroupWise systems across the Internet and to allow GroupWise users to exchange e-mail with users of various Internet e-mail programs. For information about Internet Agent files and directories, see Section 7.4, "Internet Agent Queue Directory," on page 87 and Section 8.2, "Internet Agent Installation," on page 119.

For information about the Internet Agent, see "Internet Agent" in the *GroupWise 7 Administration Guide*.

webaccess directory

The `internet\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 7.5, "WebAccess Agent Queue Directory," on page 93, Section 8.3, "WebAccess Agent Installation," on page 126, and Section 8.5, "Apache/Tomcat Installation," on page 143

For information about WebAccess, see "WebAccess" in the *GroupWise 7 Administration Guide*.

9.1.11 license directory

The `license` subdirectory contains the GroupWise Software License Agreement in multiple languages. Run `license.exe`, then select a language.

9.1.12 common directory

The `common` subdirectory contains Java files used by various components of GroupWise.

9.1.13 docs directory

The `docs` subdirectory contains language-specific subdirectories for the *Installation Guide* in PDF format. You can find the most up-to-date version of the *GroupWise 7 Installation Guide* on the GroupWise 7 Documentation Web site (http://www.novell.com/documentation/gw7).

9.2 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
novell-groupwise-agents-*ver*.*date*.i386.rpm	GroupWise agent package
novell-groupwise-gwha-*ver*.*date*.i386.rpm	GroupWise High Availability service package
novell-sms-*ver*.i586.rpm	Novell® Storage Management Services™ package
startups	Subdirectory for startup files
language	Language-specific files
strtup*xx*.poa	POA startup file
strtup*xx*.mta	MTA startup file
grpwise	GroupWise agent startup script (POA and MTA)
domain	Domain data dictionary files
wpdomain.dc	Data dictionary for GW 4.*x* domain databases
gwdom.dc	Data dictionary for GW 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 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 7, 6.*x*, and 5.*x* post office databases

📁 `client`	GroupWise client software
📁 `linux`	Subdirectory for Linux software
🗋 `install`	Cross-Platform client installation script
🗋 `novell-groupwise-gwclient-`*`ver.date`*`.i386.rpm`	Cross-Platform client package
🗋 `novell-groupwise-gwcheck-`*`ver.date`*`.i386.rpm`	GroupWise Check Utility package
📁 `ofviews`	GroupWise client view files
📁 `admin`	GroupWise administrator software
🗋 `novell-groupwise-admin-`*`ver.date`*`.i386.rpm`	GroupWise snap-ins to ConsoleOne package
🗋 `novell-groupwise-dbcopy-`*`ver.date`*`.i386.rpm`	GroupWise Database Copy Utility package
🗋 `novell-groupwise-gwcheck-`*`ver.date`*`.i386.rpm`	GroupWise Check Utility package
🗋 `NOVLc1Linuxjre-`*`ver`*`.i386.rpm`	ConsoleOne JRE update package
📁 `monitor`	GroupWise Monitor software
📁 `linux`	Subdirectory for Linux software
🗋 `novell-groupwise-gwmon-`*`ver.date`*`.i386.rpm`	Monitor Agent package
🗋 `novell-groupwise-monitor-`*`ver.date`*`.i386.rpm`	Monitor Application package
📁 `startup`	Subdirectory for startup files
🗋 `grpwise-ma`	Monitor startup script
🗋 `S99grpwise-ma`	Monitor startup script for automatic start on reboot
📁 `internet`	Internet connectivity software

gwia	GroupWise Internet Agent software
linux	Subdirectory for Linux software
novell-groupwise-gwia-*ver.date*.i386.rpm	GroupWise Internet Agent package
novell-groupwise-gwha-*ver.date*.i386.rpm	GroupWise High Availability service package
root	Subdirectory for boilerplate supporting files
gwia.cfg	Internet Agent startup file
gwauth.cfg	Host authentication configuration file
mimetype.cfg	MIME encoding configuration file
gwac.dc	Database dictionary for the access control database
preamble.txt	Message for non-MIME recipients
preamble.all	Preamble message in various languages
status*xx*.xml	File for customizing status messages
startup	Subdirectory for startup files
grpwise	Internet Agent startup script
webacces	GroupWise WebAccess software
linux	Subdirectory for Linux software
novell-groupwise-gwinter-*ver.date*.i386.rpm	GroupWise WebAccess Agent package
novell-groupwise-webaccess-*ver.date*.i386.rpm	GroupWise WebAccess Application package
root	Subdirectory for boilerplate supporting files
mimetype.cfg	MIME encoding configuration file
gwac.dc	Database dictionary for the access control database
gwgs.ini	Gateway identifier file
startup	Subdirectory for startup files
grpwise-wa	WebAccess Agent startup script
S99grpwise-wa	WebAccess Agent startup script for automatic start on reboot
webserver	Web server to support WebAccess and Monitor
linux	Subdirectory for Linux software
*.rpm	Apache, Tomcat, and related packages
license	GroupWise Software License Agreement in multiple languages
docs	GroupWise Readme, *Quick Start*, and *Installation Guide*
gwinst	Directories and files used by the Installation program

9.2.1 /opt/novell/groupwise/software directory

The GroupWise software distribution directory resides initially on the GroupWise distribution CD. GroupWise Administration is installed directly from the CD. 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. See "Linux: Setting Up a Basic GroupWise System" in the *GroupWise 7 Installation Guide*.

9.2.2 agents directory

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

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

novell-groupwise-agents.*version*.*date*.i386.rpm

The `novell-groupwise-agents.`*`version`*`.`*`date`*`.i386.rpm` file is the GroupWise agent package that is installed by the Installation script. It installs the POA and the MTA. You can install the agent package manually if necessary. To see what files are installed by the package, see Section 8.1.2, "Linux Installation Directory," on page 110.

novell-groupwise-gwha.*version*.*date*.i386.rpm

The `novell-groupwise-gwha.`*`version`*`.`*`date`*`.i386.rpm` file is the GroupWise High Availability service package that is installed by the Installation script. It is automatically installed along with the POA and the MTA. You can install the High Availability package manually if necessary. To see what files are installed by the package, see Section 8.1.2, "Linux Installation Directory," on page 110.

novell-sms-*version*.i586.rpm

The `novell-sms-`*`version`*`.i586.rpm` file is the Novell Storage Management Services package that is used in conjunction with the Service Agent for File Systems (TSAFS) and TSAFS for GroupWise (TSAFSGW) for the purpose of backing up GroupWise databases in a live GroupWise system. It is not installed by the Installation script but must be installed manually, as described in "Linux: Running TSAFS and TSAFSGW" in "Databases" in the *GroupWise 7 Administration Guide*.

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 Section 8.1.2, "Linux Installation Directory," on page 110.

grpwise file

The grpwise file is the GroupWise agent startup script. It is installed in /etc/init.d. If you choose to have the agents start automatically when the server reboots, the Installation script places symbolic links to it named S99grpwise in the rc3.d and rc5.d subdirectories of /etc/init.d. For more information, see “Linux: Installing and Starting the GroupWise Agents” in “Installing a Basic GroupWise System” in the *GroupWise 7 Installation 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 distribution media.

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 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 7/6.*x*/5.*x* secondary domains. The original gwdom.dc file is located in the domain subdirectory of the GroupWise distribution media.

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 distribution media. 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 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 7/6.*x*/5.*x* post offices in that domain. The original gwpo.dc file is located in the domain directory of the GroupWise CD. 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 CD.

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

9.2.5 client directory

The client subdirectory contains all files associated with GroupWise Windows client. See "Client" in the *GroupWise 7 Administration Guide*.

install file

The install file is the script that GroupWise Cross-Platform client users run to install and set up the GroupWise client on their Linux or Macintosh* workstations. See "Setting Up the GroupWise Cross-Platform Client on Linux" in the *GroupWise 7 Installation Guide*.

novell-groupwise-gwclient-*version.date*.i386.rpm

The novell-groupwise-client.*version*.*date*.i386.rpm file is the GroupWise client package that is installed by the Installation script. You can install the client package manually if necessary. To see what files are installed by the package, see Section 10.2, "Cross-Platform Client on Linux," on page 175.

novell-groupwise-gwcheck-*version.date*.i386.rpm file

The `novell-groupwise-gwcheck.`*`version`*`.`*`date`*`.i386.rpm` file is the GroupWise Check utility package. It is not installed by the Installation script but must be installed manually, as described in "Starting GWCheck on a Linux Workstation" in "Databases" in the *GroupWise 7 Administration Guide*.

9.2.6 ofviews directory

The `client\ofviews` subdirectory contains platform-specific subdirectories of view files for use by the GroupWise client. On Linux, the `win` subdirectory contains the view files for the Windows client, so that users on a Linux post office can use the Windows client if desired. The Cross-Platform client does not use view files like the Windows client does.

In addition, the `gwview`*`xx`*`.ini` and `ofview`*`xx`*`.ini` files configure custom views on the menus where users select views in the Windows client. The `gwview`*`xx`*`.ini` file configures GroupWise 7, 6.*x*, and 5.5 views. The `ofview`*`xx`*`.ini` file configures views from earlier versions of GroupWise.

9.2.7 admin directory

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

novell-groupwise-admin-*version.date*.i386.rpm file

The `novell-groupwise-admin.`*`version`*`.`*`date`*`.i386.rpm` file is the GroupWise administrator package that is installed by the Installation script. It contains the GroupWise Administrator snap-ins to ConsoleOne. You can install the admin package manually if necessary.

novell-groupwise-dbcopy-*version.date*.i386.rpm file

The `novell-groupwise-dbcopy.`*`version`*`.`*`date`*`.i386.rpm` file is the GroupWise Database Copy utility package. It is not installed by the Installation script but must be installed manually, as described in "Using DBCopy on Linux Servers" in "Databases" in the *GroupWise 7 Administration Guide*.

novell-groupwise-gwcheck-*version.date*.i386.rpm file

The `novell-groupwise-gwcheck.`*`version`*`.`*`date`*`.i386.rpm` file is the GroupWise Check utility package. It is not installed by the Installation script but must be installed manually, as described in "Starting GWCheck on a Linux Workstation" in "Databases" in the *GroupWise 7 Administration Guide*.

NOVLc1Linuxjre-*version*.i386.rpm file

The `NOVLc1Linuxjre-`*`version`*`.i386.rpm` file is the package that updates ConsoleOne so that it is compatible with the GroupWise Administrator snap-ins. It is installed for you automatically when you install GroupWise administration.

monitor directory

The monitor subdirectory contains the GroupWise Monitor software. See "Monitor" in the *GroupWise 7 Administration Guide*.

novell-groupwise-gwmon-*version*.*date*.i386.rpm file

The novell-groupwise-gwmon.*version*.*date*.i386.rpm file is the GroupWise Monitor Agent package that is installed by the Installation script. You can install the package manually if necessary. To see what files are installed by the package, see Section 8.4, "Monitor Agent Installation," on page 138 and Section 8.5, "Apache/Tomcat Installation," on page 143.

novell-groupwise-monitor-*version*.*date*.i386.rpm file

The novell-groupwise-monitor.*version*.*date*.i386.rpm file is the GroupWise Monitor Application package that is installed by the Installation script. You can install the package manually if necessary. To see what files are installed by the package, see Section 8.4, "Monitor Agent Installation," on page 138 and Section 8.5, "Apache/Tomcat Installation," on page 143.

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 for the First Time" in "Installing GroupWise Monitor" in the *GroupWise 7 Installation Guide*.

S99grpwise-ma file

The S99grpwise-ma file is the Monitor Agent startup script that is placed in the rc3.d and rc5.d subdirectories of /etc/init.d if you choose to have the Monitor Agent start automatically when the server reboots. For more information, see "Starting the Linux Monitor Agent on System Startup" in "Installing GroupWise Monitor" in the *GroupWise 7 Installation Guide*.

9.2.8 internet directory

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

gwia directory

The gwia subdirectory contains the Internet Agent software, used to connect GroupWise systems across the Internet and to allow GroupWise users to exchange e-mail with users of various Internet e-mail programs. See "Internet Agent" in the *GroupWise 7 Administration Guide*.

novell-groupwise-gwia-*version*.*date*.i386.rpm file

The novell-groupwise-gwia.*version*.*date*.i386.rpm file is the Internet Agent package that is installed by the Installation script. You can install the package manually if necessary. To see what files are installed by the package, see Section 8.2, "Internet Agent Installation," on page 119.

novell-groupwise-gwha.*version*.*date*.i386.rpm

The `novell-groupwise-gwha.`*`version`*`.`*`date`*`.i386.rpm` file is the GroupWise High Availability service package that is installed by the Installation script. It is automatically installed along with the Internet Agent. You can install the High Availability package manually if necessary. To see what files are installed by the package, see Section 8.2.2, "Linux Installation Directory," on page 120.

root directory

The `root` directory contains boilerplate versions of the configuration files that are installed to the Internet Agent root directory under the `domain` directory. For more information, see Section 7.4, "Internet Agent Queue Directory," on page 87.

grpwise file

The `grpwise` file is the Internet Agent startup script. During installation, it is placed in `/etc/init.d`. If you choose to have the Internet Agent start automatically when the server reboots, the Installation script places symbolic links to it named `S99grpwise-ia` in the `rc3.d` and `rc5.d` subdirectories of `/etc/init.d`. For more information, see "Linux: Starting the Internet Agent" in "Installing the GroupWise Internet Agent" in the *GroupWise 7 Installation Guide*.

webacces directory

The `internet\webacces` subdirectory contains the GroupWise WebAccess software, which allows users to access their GroupWise mailboxes from a Web browser. See "WebAccess" in the *GroupWise 7 Administration Guide*.

novell-groupwise-gwinter-*version*.*date*.i386.rpm file

The `novell-groupwise-gwinter.`*`version`*`.`*`date`*`.i386.rpm` file is the WebAccess Agent package that is installed by the Installation script. You can install the package manually if necessary. To see what files are installed by the package, see Section 8.3, "WebAccess Agent Installation," on page 126.

novell-groupwise-webaccess-*version*.*date*.i386.rpm file

The `novell-groupwise-webaccess.`*`version`*`.`*`date`*`.i386.rpm` file is the WebAccess Application package that is installed by the Installation script. You can install the package manually if necessary. To see what files are installed by the package, see Section 8.3, "WebAccess Agent Installation," on page 126 and Section 8.5, "Apache/Tomcat Installation," on page 143.

root directory

The `root` directory contains boilerplate versions of the configuration files that are installed to the WebAccess Agent root directory under the domain directory. For more information, see Section 7.5, "WebAccess Agent Queue Directory," on page 93.

grpwise-wa file

The `grpwise-wa` file is the WebAccess Agent startup script. During installation, it is placed in `/etc/init.d`. For more information, see "Starting the Linux WebAccess Agent for the First Time" in "Installing GroupWise WebAccess" in the *GroupWise 7 Installation Guide*.

S99grpwise-wa file

The `S99grpwise-wa` file is the WebAccess Agent startup script that is placed in the `rc3.d` and `rc5.d` subdirectories of `/etc/init.d` if you choose to have the WebAccess Agent start automatically when the server reboots. During installation, it is named `grpwise-ma` and placed in `/etc/init.d`. For more information, see "Starting the Linux WebAccess Agent on System Startup" in "Installing GroupWise WebAccess" in the *GroupWise 7 Installation Guide*.

webserver directory

On Linux, you can install Apache and Tomcat along with WebAccess and Monitor. This automatically integrates WebAccess and Monitor into Apache and Tomcat for you. If you are already running Apache and Tomcat, you can choose to integrate WebAccess and Monitor into that installation, or you can have two Apache/Tomcat installations on your server. WebAccess and Monitor cannot be used without a supporting Web server. For more information, see "Determining the WebAccess and WebPublisher Applications' Configuration" in "Installing GroupWise WebAccess" and "Determining the Monitor Application's Configuration" in "Installing GroupWise Monitor" *GroupWise 7 Installation Guide*.

9.2.9 license directory

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

9.2.10 docs directory

The `docs` subdirectory contains language-specific subdirectories for the *GroupWise 7 Installation Guide* and *Quick Start* in PDF format, along with the Readme in HTML format. You can find the most up-to-date version of the GroupWise documentation on the GroupWise 7 Documentation Web site (http://www.novell.com/documentation/gw7).

9.2.11 gwinst directory

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

GroupWise Client Installation Directories

10

- Section 10.1, “Windows Client,” on page 171
- Section 10.2, “Cross-Platform Client on Linux,” on page 175
- Section 10.3, “Cross-Platform Client on Macintosh,” on page 176

10.1 Windows Client

📁 c:\novell\groupwise	Installation directory for GroupWise® 7 Windows client software
🗋 grpwise.exe	GroupWise client program
🗋 gwtip.exe	GroupWise Tip of the Day 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
🗋 pdaconnect.exe	PDA Connect Installation program
🗋 ngwguard.dc	Data dictionary for databases
🗋 wprof.dc	Data dictionary for Remote Address Book
🗋 *.dll	DLL programs to support GroupWise Windows client
🗋 *.ocx	OLE custom controls
🗋 *.flt	Graphics filters for Inso viewers
🗋 *.chm	Help files to support the GroupWise client
📁 gwcheck	GroupWise Check utility
📁 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

10.1.1 c:\novell\groupwise

The GroupWise Windows client installation directory contains all files necessary to run the GroupWise client. The default location is `c:\novell\groupwise`.

- For a standard installation, the GroupWise client directory resides on the user's workstation.
- For a workstation installation, the GroupWise client directory resides on the network and is run by users on many workstations.
- For a remote GroupWise installation, the GroupWise directory contains the same files as would be installed on a workstation connected to the network. Whenever GroupWise cannot make a network connection, it checks for a Remote mailbox to access. See Section 7.7, "Remote Mailbox Directory," on page 100.

10.1.2 grpwise.exe file

The `grpwise.exe` file is the GroupWise 7 client program for use with Windows.

10.1.3 gwtip.exe file

The `gwtip.exe` file is the GroupWise Tip of the Day program, which displays helpful information about a GroupWise feature each time you start the GroupWise client.

10.1.4 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.

10.1.5 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 e-mail distribution lists, and so on.

10.1.6 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 7 Administration Guide*.

10.1.7 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 "Importing POP3 and IMAP4 Accounts" in "Working with Items in Your Mailbox" in the *GroupWise 7 Windows Client User Guide*.

10.1.8 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 e-mail program on your workstation. See "Setting Up the GroupWise Client" in "Installing the GroupWise Windows and Cross-Platform Clients" in the *GroupWise 7 Installation Guide*.

10.1.9 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 7 Windows Client User Guide*.

10.1.10 gwsync.exe file

The gwsync.exe file is the program that synchronizes your Online and Caching mailboxes. See "Using Caching Mode" in the *GroupWise 7 Windows Client User Guide*.

10.1.11 pdaconnect.exe file

The pdaconnect.exe file is the PDA Connect Installation program for synchronizing data between GroupWise and a PDA device. For more information, see the PDA Connect Readme (http://www.novell.com/documentation/gw7/readmeen_pda_connect/readmeen_pda_connect.html).

10.1.12 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.

10.1.13 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.

10.1.14 *.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.

10.1.15 *.ocx files

*.ocx files are OLE custom controls. For example, *.ocx files control the Address Book and name completion.

10.1.16 *.flt files

*.flt files are graphics filters used by the Inso viewers incorporated into the GroupWise client. The viewers enable you to view a wide variety of file types from within GroupWise.

10.1.17 *.chm files

*.chm files contain the online documentation for all components of GroupWise. *.chm files are typically located in the same directory with the executable programs they support.

10.1.18 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 "Managing Your Mailbox" in the *GroupWise 7 Windows Client User Guide*.

10.1.19 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.

10.1.20 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.

10.2 Cross-Platform Client on Linux

📁 /opt/novell/groupwise/client	GroupWise Cross-Platform client installation directory
🗋 gwclient.desktop	Desktop shortcut for the Cross-Platform client
🗋 gwclient.png	Desktop icon for the Cross-Platform client shortcut
📁 bin	Subdirectory for the Cross-Platform client executable
🗋 groupwise	Cross-Platform client executable
📁 lib	Subdirectory for Cross-Platform client library files
🗋 gwenlxxx.fil	Cross-Platform client language resource files
🗋 ngwguard.dc	Data dictionary for local databases on the user's workstation
🗋 wprof.dc	Data dictionary for the local copy of the Address Book
🗋 *.jar	Java archive (.jar) files used by the Cross-Platform client
🗋 *.so	Shared library files used by the Cross-Platform client
📁 jre	Java Runtime Environment (JRE) used by the Cross-Platform client
📁 logs	Log files generated by the Cross-Platform client

10.2.1 /opt/novell/groupwise/client directory

The /opt/novell/groupwise/client directory is the default location for the GroupWise Cross-Platform software. The Installation program does not provide the opportunity to specify a different location and typically there is no need to move the software.

gwclient.desktop file

The gwclient.desktop file create the GroupWise Cross-Platform client shortcut on your Linux desktop.

gwclient.png file

The gwclient.png file is the GroupWise icon used for Cross-Platform client shortcut.

10.2.2 bin directory

The bin directory is the standard Linux location for executable files.

groupwise file

The groupwise file is the GroupWise Cross-Platform client executable. See "Installing the GroupWise Cross-Platform Client" in the *GroupWise 7 Cross-Platform Client User Guide*.

10.2.3 lib directory

The `lib` directory is the standard Linux location for library files.

gwenl*xx*.fil file

The `gwenlxxx`.file provides the language-specific strings for the localized versions of the Cross-Platform client. The *xx* is a two-letter language code.

ngwguard.dc file

The `ngwguard.dc` file is the data dictionary for building the databases used by the Cross-Platform client. It is parallel in function to the `ngwguard.dc` file in the post office.

wprof.dc file

The `wprof.dc` file is the data dictionary used to create the local copy of the Address Book that resides on the user's workstation (`wprof.db`).

*.jar files

The `*.jar` files efficiently store information referenced by the Cross-Platform client. The `gwclient.jar` file contains the Cross-Platform online help information.

*.so files

The `*.so` files are Linux shared library files the provide information to the Cross-Platform client executable.

10.2.4 jre directory

The `jre` directory holds the version of the Java Runtime Environment (JRE) that is required by the Cross-Platform client.

10.2.5 logs directory

The `logs` directory stores log files generated by the Cross-Platform client when an error occurs.

10.3 Cross-Platform Client on Macintosh

`/Applications/GroupWise.app`	GroupWise Cross-Platform client installation directory
`Contents`	Standard Macintosh directory for application files
`PkgInfo`	Standard Macintosh package info file
`Info.plist`	Standard Macintosh package list file
`MacOS`	Standard Macintosh directory for operating system files

Resources	Standard Macintosh directory for resource files
GroupWise.icns	Icons used by the Cross-Platform client
lib	Subdirectory for Cross-Platform client library files
gwenl*xxx*.fil	Data dictionary for local databases on the user's workstation
ngwguard.dc	Data dictionary for the local copy of the Address Book
wprof.dc	Cross-Platform client language information files
Java	JRE used by the Cross-Platform client
*.jar	Java archive (.jar) files used by the Cross-Platform client
*.dylib	Dynamic libraries used by the Cross-Platform client

10.3.1 /Applications/GroupWise.app directory

The /Applications/GroupWise.app directory contains the GroupWise Cross-Platform client on Macintosh.

10.3.2 Contents directory

The Contents directory is the standard location on Macintosh for all files that make up an application.

10.3.3 MacOS directory

The MacOS directory contains the JavaApplicationStub file that enables the Cross-Platform client to invoke the Java Virtual Machine (JVM) on Macintosh.

10.3.4 Resources directory

The Resources directory is the standard location on Macintosh for files that support an application.

GroupWise.icns file

The GroupWise.icns file contains all the images that display in the Cross-Platform client.

10.3.5 lib directory

The lib directory holds library files that provide information required by the Cross-Platform client.

gwenl*xxx*.fil file

The gwenl*xxx*.file provides the language-specific strings for the localized versions of the Cross-Platform client. The *xx* is a two-letter language code.

ngwguard.dc file

The ngwguard.dc file is the data dictionary for building the databases used by the Cross-Platform client. It is parallel in function to the ngwguard.dc file in the post office.

wprof.dc file

The wprof.dc file is the data dictionary used to create the local copy of the Address Book that resides on the user's workstation (wprof.db).

10.3.6 Java directory

The Java directory holds the version of the Java Runtime Environment (JRE) that is required by the Cross-Platform client.

*.jar files

The *.jar files efficiently store information referenced by the Cross-Platform client. The gwclient.jar file contains the Cross-Platform online help information.

*.dylib files

The *.dylib files are dynamic library files the provide information to the Cross-Platform client program.