

CommuniGate Pro Configuration Guide



This Configuration Guide in Adobe Acrobat® format (PDF) is valid for CommuniGate Pro.

A Configuration Guide is also available on-line from the <http://www.stalker.com/CommuniGatePro/>

The on-line guide, which is updated more frequently, contains information for the most current release of CommuniGate Pro.

If there are any discrepancies between the PDF guide and the on-line guide, please use the information from the on-line guide.

CommuniGate Pro Quick Start 20

CHAPTER 1 **Introduction** 24

- [CommuniGate Pro Product Description](#)
- [Server Features](#)
- [Server Administrating](#)
- [Support and Discussions](#)
- [What Is New in CommuniGate Pro Updates?](#)
- [Download the Latest Versions](#)
- [Download the CommuniGate Pro Plugins](#)
- [Download the CommuniGate Pro MAPI Connector](#)

CHAPTER 2 **Installation** 27

- [Installation](#)
 - [Installing on a MS Windows 2000/NT/9x System](#)
 - [Installing on a Sun Solaris System](#)
 - [Installing on a Linux System](#)
 - [Installing on a MacOS X \(Darwin\) System](#)
 - [Installing on a MacOS X Server \(Rhapsody\) System](#)
 - [Installing on a FreeBSD System](#)
 - [Installing on a BSDI BSD/OS System](#)
 - [Installing on an OpenBSD System](#)
 - [Installing on an AIX System.](#)
 - [Installing on an HP/UX System](#)
 - [Installing on a Tru64 System](#)
 - [Installing on an SGI IRIX System](#)

- [Installing on an SCO UnixWare System](#)
- [Installing on an IBM OS/400 System](#)
- [Installing on a QNX System](#)
- [Installing on an IBM OS/2 System](#)
- [Installing on a BeOS System](#)
- [Initial Configuration](#)
- [Upgrading to a Newer Version](#)
- [Moving to a New Hardware Server](#)

CHAPTER 3

Migration

42

- [Supporting Network Users](#)
- [Supporting Local Users](#)
- [Using Legacy Mailboxes](#)
- [Converting Passwords](#)
- [Migrating from sendmail](#)
- [Migrating from Post.Office® Servers](#)
- [Migrating from Netscape®/iPlanet Messaging Servers](#)
- [Migrating from IMail® Servers](#)
- [Migrating from CommuniGate/MacOS and SIMS](#)
- [Copying Mailboxes from Other POP Servers](#)
- [Copying Mailboxes from Other IMAP Servers](#)
- [Copying All Mailboxes from Other Servers](#)
- [Migrating from an Arbitrary Server \("on-the-fly" migration\)](#)
- [Switching Servers](#)
- [Moving To Secondary Domains](#)

CHAPTER 4

System Administration

56

- [Sections and Access Rights](#)
- [Base Directory Structure](#)
- [General Settings](#)
- [Command Line Options](#)
- [Specifying Command Line Options under Windows NT/2000](#)
- [Customizing Unix Startup Scripts](#)
- [Shutting Down](#)
- [OS syslog](#)
- [Server Root Privilege](#)
- [Domain Administration](#)
- [Domains Administrators in other Domains](#)
- [Customizing Domain WebAdmin Interface](#)

- [Customizing Server Prompts](#)
- [Domain Name Resolver \(DNR\)](#)

CHAPTER 5

HTTP Module

70

- [Access to the WebAdmin Interface Pages](#)
- [Access to the Domain WebAdmin Interface Pages](#)
- [WebAdmin Settings \(Preferences\)](#)
- [Access to the WebUser Interface](#)
- [Access to Personal Web Sites](#)
- [Configuring the HTTP module](#)
- [Routing](#)

CHAPTER 6

Server Logs

75

- [Creating and Deleting Log Files](#)
- [Specifying a Time Interval](#)
- [Filtering Log Records](#)
- [Searching](#)

CHAPTER 7

Protection

79

- [Prohibiting Unauthorized Relaying](#)
 - [Specifying Client IP Addresses](#)
 - [Specifying Client IP Domains](#)
 - [Configuring the SMTP module](#)
- [Relaying for Mobile Users](#)
 - [The SMTP AUTH method](#)
 - [The Read-then-Send method](#)
 - [Account and Domain Settings](#)
- [Client-only Logins](#)
- [Relaying Rerouted Messages](#)
- [Return-Path Address Verification](#)
- [Blacklisting Offenders](#)
 - [Specifying Offender Addresses](#)
 - [Using DNS-based Blacklisting \(RBL\)](#)
 - [Un-listing Addresses \(White Hole Addresses\)](#)
- [Spam Traps](#)
- [Banning Mail by Header and Body Lines](#)
- [Filtering Mail](#)
- [Cluster Setup](#)

CHAPTER 8	<u>Security</u>	93
	<ul style="list-style-type: none"> • <u>Authentication Methods</u> • <u>Account Passwords</u> • <u>CommuniGate Passwords</u> • <u>OS Passwords</u> • <u>External Authentication</u> • <u>Account Name Harvesting</u> • <u>Account Password Attacks</u> • <u>Granting Access Rights to Users</u> • <u>Restricting Access</u> • <u>Using SSL/TLS Secure Connections</u> • <u>Certificates and Private Keys</u> • <u>Domain Security Settings</u> <ul style="list-style-type: none"> ○ <u>Assigning a Private Key</u> ○ <u>Assigning a Certificate</u> ○ <u>Assigning a Certificate Authority Chain</u> • <u>Using Default and Self-Signed Certificates</u> 	
CHAPTER 9	<u>Scalability</u>	112
	<ul style="list-style-type: none"> • <u>Serving Large Domains</u> • <u>Handling High-Volume Local Delivery</u> • <u>Supporting Many Concurrent Clients</u> <ul style="list-style-type: none"> ○ <u>Setting the TCP TIME_WAIT time</u> • <u>Handling High-Volume SMTP Delivery</u> • <u>Estimating Resources Usage</u> • <u>OS Limitations</u> 	
CHAPTER 10	<u>TCP/IP Listener</u>	118
	<ul style="list-style-type: none"> • <u>Multi-Socket Listening</u> • <u>Secure Sockets</u> • <u>Restrictions</u> • <u>Limiting Connections from the same Address</u> 	
CHAPTER 11	<u>Virus Scanner</u>	122
	<ul style="list-style-type: none"> • <u>Installing External Filter Software</u> • <u>Starting the External Filter</u> • <u>Using the External Filter</u> 	
CHAPTER 12	<u>Alerts</u>	126

- [Posting Alerts](#)
- [Storage Quota Alerts](#)

CHAPTER 13

SNMP Interface

129

- [Configuring SNMP Agent](#)
- [Accessing the Server MIB](#)
- [Monitoring the SNMP elements via Web](#)
- [Monitoring SNMP elements via CLI/API](#)
- [Sending SNMP Traps](#)

CHAPTER 14

Events

133

- [Configuring Event Handlers](#)
 - [Notification via E-mail](#)
 - [Notification via SNMP Traps](#)
 - [Notification via Account Alerts](#)
- [Specifying Events](#)

CHAPTER 15

Dialup

137

- [Mail Receiving](#)
- [TCP Activity Schedule](#)
- [Serving LAN Clients](#)

CHAPTER 16

Command Line Interface

140

- [Administrating the Server via the PWD module](#)
- [CLI Syntax](#)
- [Account Administration](#)
- [Group Administration](#)
- [Forwarder Administration](#)
- [Domain Administration](#)
- [Mailbox Administration](#)
- [Alert Administration](#)
- [Personal Web Site Administration](#)
- [Mailing Lists Administration](#)
- [Web Skins Administration](#)
- [Web Interface Integration](#)
- [Server Settings](#)
- [Monitoring](#)
- [Access Rights Administration](#)
- [Statistics](#)
- [Miscellaneous Commands](#)

CHAPTER 17

- [Web Interface Tuning](#)

Objects

189

- [Domains](#)
- [Accounts](#)
- [Groups](#)
- [Forwarders](#)
- [Mailing Lists](#)
- [Account Aliases](#)
- [Mailboxes](#)
- [Default Settings](#)

CHAPTER 18

Domains

196

- [Displaying the Domain List](#)
- [Creating a New Domain](#)
- [Specifying Domain Settings](#)
- [Enabling Messaging Services](#)
- [Multihoming and Dedicated IP Addresses](#)
- [Domain Aliases](#)
- [Directory Integration](#)
- [Domain Limits](#)
- [Processing Unknown Names](#)
- [Sending Mail To All Accounts in the Domain](#)
- [Sending Mail To All Accounts in All Domains](#)
- [WebUser Interface Settings](#)
- [Enabling Auto-Signup](#)
- [Relaying via a Dedicated IP Address](#)
- [Server OS Integration](#)
 - [Legacy \(Unix\) Mailer Compatibility](#)
- [Subdirectories for Large Domains](#)
- [Administrator Domain](#)
- [Renaming Domains](#)
- [Removing Domains](#)
- [Specifying Default Domain Settings](#)
- [Specifying Domain Security Settings](#)
- [Domain File Directories](#)

CHAPTER 19

Mapping

213

- [CommuniGate Pro Domains](#)

- [Direct Mapping \(Domain Aliases\)](#)
- [Modifying Mapping](#)
- [Unified Domain-Wide Accounts](#)

CHAPTER 20

[**Accounts**](#)

217

- [Displaying the Account List](#)
- [Creating a New Account](#)
- [Specifying Account Settings](#)
 - [Authentication Methods](#)
 - [Enabled Services](#)
 - [Resource Usage Limits](#)
 - [Processing Options](#)
 - [Miscellaneous Options](#)
- [Specifying Account Aliases](#)
- [Creating Mailing Lists](#)
- [Renaming Accounts](#)
- [Removing Accounts](#)
- [Specifying Default Account Settings](#)
- [Specifying Account Template](#)
- [Importing User Accounts Information](#)

CHAPTER 21

[**Groups**](#)

232

- [Creating a New Group](#)
- [Specifying Group Settings](#)
- [Group Member Processing](#)
- [Renaming Groups](#)
- [Removing Groups](#)

CHAPTER 22

[**Forwarders**](#)

236

- [Specifying Domain Forwarders](#)

CHAPTER 23

[**Mailboxes**](#)

238

- [Mailbox Names](#)
- [Mailbox Access Control Lists \(ACL\)](#)
- [Mailbox Formats](#)
- [Creating Mailboxes](#)
- [Mailbox Subscription](#)
- [Mailbox Aliases](#)

CHAPTER 24

[**Web Files**](#)

245

- [HTTP Access to Personal Web Sites](#)

- [Private Folder](#)
- [HTML-based Management](#)
- [HTTP-based Management](#)
- [FTP-based Management](#)

CHAPTER 25

[**Account Data**](#)

249

- [Domain Files](#)
- [Account Files](#)
- [Personal Web Site](#)
- [Preferences](#)
- [Netscape Roaming](#)

CHAPTER 26

[**Message Transfer**](#)

253

- [Submitting Messages](#)
- [Routing](#)
- [Enqueueing](#)
- [Delays and Suspensions](#)
- [Dequeueing](#)

CHAPTER 27

[**Router**](#)

259

- [Domain and Local Parts of E-mail Addresses](#)
- [Converting to the E-mail Address Type](#)
- [Main Domain Name](#)
- [Multiple Domains and MX Records](#)
- [Routing Table](#)
- [Domain-Level Routing Records](#)
- [Alias \(Address-Level\) Routing Records](#)
- [Special Addresses](#)
- [Routing by IP Addresses](#)
- [Routing via Modules](#)
- [Default Records](#)
- [Extending Non-Qualified Domain Names](#)
- [Cluster-wide Routing Table](#)

CHAPTER 28

[**Automated Mail Processing \(Rules\)**](#)

270

- [Specifying Account Rules](#)
- [Creating, Renaming and Removing Rules](#)
- [Rule Conditions](#)
 - [String Lists](#)
- [Rule Actions](#)

- [Auto-Reply Message](#)
- [Redirect All Simplified Rule](#)
- [Logging Rules Activity](#)
- [Using External Content/Virus Filtering](#)
- [Content Filtering API](#)
- [Cluster-wide Rules](#)

CHAPTER 29

SMTP Module

290

- [Simple Mail Transfer Protocol \(SMTP\) and DNS](#)
- [Configuring the SMTP module](#)
- [Sending Messages via the Internet](#)
 - [Sending via a Forwarding Mail Server](#)
 - [Sending Directly to the Recipients](#)
 - [Multi-channel Delivery](#)
 - [Sending via Dial-up Links](#)
 - [Secure \(encrypted\) Message Sending](#)
- [Receiving Messages](#)
 - [Waking up the Backup Server](#)
 - [On-demand Mail Relaying \(ATRN\)](#)
- [Serving Dial-up Client Hosts](#)
 - [Remote Queue Starting \(ETRN\)](#)
 - [On-demand Mail Relaying \(ATRN/TURN\)](#)
 - [Waking up via E-mail](#)
 - [Holding Mail in Queue](#)
- [Message Relaying](#)
 - [Relaying via Dedicated IP Addresses](#)
- [Processing Mail from BlackListed Addresses](#)
- [Routing](#)
- [Sending to Non-Standard Ports](#)

CHAPTER 30

Local Delivery Module

304

- [Configuring the Local Delivery Module](#)
- [Message Flow Control](#)
- [Routing](#)
- [Routing to Unknown Accounts](#)
- [Unified Domain-Wide Accounts](#)
- [Automated Mail Processing](#)
- [Storing Mail in Account Mailboxes](#)
- [Direct Mailbox Addressing](#)

- [Routing Settings](#)
- [Sending Mail to All Accounts](#)
- [All-Domain Aliases](#)

CHAPTER 31

RPOP Module

312

- [Post Office Protocol \(POP3\) and Mail Retrieving](#)
- [Configuring the RPOP module](#)
- [Specifying Unified Domain-Wide Accounts](#)
- [Special Headers and Mail Distribution](#)
 - [Mail Distribution without Special Headers](#)
- [Specifying Remote Accounts for Individual Users](#)
- [Appendix A. Configuring sendmail for Unified Domain-Wide Accounts](#)

CHAPTER 32

LIST Module

320

- [Mailing Lists](#)
- [Configuring the LIST module](#)
 - [Creating Mailing Lists](#)
 - [Configuring Mailing Lists](#)
 - [Renaming Mailing Lists](#)
 - [Removing Mailing Lists](#)
- [Composing Service Texts](#)
- [Subscription Processing](#)
 - [Subscription Modes](#)
 - [Confirmation Requests](#)
 - [Welcome and Good Bye Messages](#)
- [Posting Messages](#)
- [Processing Messages](#)
- [Bounce Processing](#)
- [FEED Mode Distribution](#)
- [Digesting and Archiving](#)
- [DIGEST/INDEX Mode Distribution](#)
- [Archiving](#)
- [Subscribers List](#)
 - [Adding Subscribers](#)
 - [Importing Subscriber Lists](#)
 - [Subscribing Lists to Lists](#)
- [Processing Service Requests](#)
- [Routing](#)

CHAPTER 33

PIPE Module

346

- [The Submitted Folder](#)
- [Delivering to External Applications](#)
 - [Serialized Delivery](#)
 - [Command Tags](#)
- [Configuring the PIPE module](#)
- [Foreign Queue Processing](#)

CHAPTER 34

[**Access to CommuniGate Pro Accounts**](#)

352

- [Access to Accounts](#)
- [Serving Multiple Domains](#)
- [Multihoming](#)
- [Routing](#)

CHAPTER 35

[**Mailbox Sharing**](#)

358

- [Simultaneous Access](#)
- [Foreign and Public Mailboxes](#)
- [External Mailboxes](#)

CHAPTER 36

[**CommuniGate Pro POP Module**](#)

362

- [Post Office Protocol \(POP3\)](#)
- [Configuring the POP module](#)
- [User Authentication](#)
- [Secure \(encrypted\) Access](#)
- [Special Features](#)
- [The XTND XMIT Extension](#)
- [Notification Alerts](#)
- [Accessing Additional Mailboxes](#)
- [Accessing Individual Mail in a Unified Account](#)

CHAPTER 37

[**CommuniGate Pro IMAP Module**](#)

369

- [Internet Message Access Protocol \(IMAP\)](#)
- [Configuring the IMAP module](#)
- [MultiAccess](#)
- [Access Control Lists](#)
- [Foreign \(Shared\) and Public Mailboxes](#)
- [User Authentication](#)
- [Notification Alerts](#)
- [Login Referrals](#)
- [Monitoring IMAP Activity](#)
- [Additional IMAP Extensions](#)

CHAPTER 38	<u>CommuniGate Pro Web User Interface</u>	374
	<ul style="list-style-type: none"> • <u>WebUser Interface to Multiple Domains</u> • <u>Account Access and WebUser Sessions</u> • <u>WebUser Interface Settings</u> • <u>WebUser Interface to Mailing Lists</u> • <u>Auto Sign-up</u> • <u>WebUser Interface Customization</u> 	
CHAPTER 39	<u>CommuniGate Pro MAPI Connector</u>	382
	<ul style="list-style-type: none"> • <u>MAPI Connector Overview</u> • <u>Installing the MAPI Connector</u> • <u>Creating a Mail Profile</u> • <u>Configuring the MAPI Connector</u> • <u>Enabling Mailbox Sharing</u> • <u>Free/Busy Information</u> <ul style="list-style-type: none"> ○ <u>Posting Free/Busy Information</u> ○ <u>Accessing Free/Busy Information for Other Users</u> • <u>WebMail Integration</u> • <u>Communicating with Microsoft Exchange users</u> 	
CHAPTER 40	<u>FTP Module</u>	393
	<ul style="list-style-type: none"> • <u>File Transfer Protocol</u> • <u>Configuring the FTP module</u> • <u>Providing Access to Personal Web Sites</u> 	
CHAPTER 41	<u>CommuniGate Pro LDAP Module</u>	395
	<ul style="list-style-type: none"> • <u>Lightweight Directory Access Protocol</u> • <u>Configuring the LDAP module</u> • <u>User Authentication</u> • <u>Central Directory</u> • <u>The mail Attribute processing</u> 	
CHAPTER 42	<u>CommuniGate Pro ACAP Module</u>	400
	<ul style="list-style-type: none"> • <u>Application Configuration Access Protocol</u> • <u>Configuring the ACAP module</u> 	
CHAPTER 43	<u>PWD Module</u>	402
	<ul style="list-style-type: none"> • <u>Password Modification Protocol (poppwd)</u> • <u>Configuring the PWD module</u> • <u>Providing Access to the Server CLI</u> 	

CHAPTER 44	<u>CommuniGate Pro: Directory</u>	404
	<ul style="list-style-type: none"> • <u>What is Directory?</u> • <u>Directory Storage Units</u> • <u>Local Units</u> • <u>Remote Units</u> • <u>Remote Directory Root</u> • <u>Binding to the Directory</u> • <u>Access Right Records</u> • <u>Access Right Specifications</u> • <u>Directory Browser</u> • <u>Importing Directory Data</u> 	
CHAPTER 45	<u>CommuniGate Pro: Directory Schema</u>	419
	<ul style="list-style-type: none"> • <u>Default Schema</u> • <u>Record Attributes</u> • <u>Object Classes</u> • <u>Object Class Descriptor</u> 	
CHAPTER 46	<u>CommuniGate Pro: Directory Integration</u>	423
	<ul style="list-style-type: none"> • <u>Directory Integration Concept</u> • <u>Attribute Renaming</u> • <u>Domains Subtree</u> • <u>Custom Account Settings</u> • <u>Integrating Regular Domains</u> • <u>LDAP-based Provisioning</u> • <u>Directory Integration in a Cluster</u> • <u>Directory-Based Domains</u> • <u>Shared (Multi-Server) Directory</u> • <u>Distributed Domains (Directory Routing)</u> 	
CHAPTER 47	<u>Data Formats</u>	465
	<ul style="list-style-type: none"> • <u>Strings, Dictionary, and Array Formats</u> 	
CHAPTER 48	<u>Clusters</u>	468
	<ul style="list-style-type: none"> • <u>Cluster Types</u> • <u>Supported Services</u> • <u>Frontend Servers</u> <ul style="list-style-type: none"> ○ <u>Withdrawing Frontend Servers from a Cluster</u> • <u>Cluster Server Configuration</u> • <u>Static Clusters</u> 	

- [Backend and Frontend Server Settings](#)
- [Adding Servers to a Static Cluster](#)
- [Withdrawing Servers from a Static Cluster](#)
- [Backend Failover in a Static Cluster](#)
- [Dynamic Clusters](#)
 - [Traditional File-Lock Approach](#)
 - [Cluster Controller](#)
 - [Cluster File Systems and Cluster OSes](#)
 - [Configuring Backend Servers](#)
 - [Adding a Backend Server to a Dynamic Cluster](#)
 - [Adding a Frontend Server to a Dynamic Cluster](#)
 - [Shared Settings](#)
 - [Shared Processing](#)
 - [Withdrawing Servers from a Dynamic Cluster](#)
 - [Upgrading Servers in a Dynamic Cluster](#)
- [Assigning IP Addresses to Shared Domains](#)
- [Security Issues](#)
- [Cluster Configuration Details](#)
- [Cluster Of Clusters](#)

CHAPTER 49

[Web Application Module](#)

491

- [Stateless and Session-based Processing](#)
- [Skins](#)
- [Skin Text Dataset](#)
- [Serving Regular Files](#)
- [Serving Web Application \(WSSP\) Files](#)
- [Creating and Managing Skins](#)

CHAPTER 50

[WSSP Scripting](#)

498

- [Scripting Elements](#)
- [Expressions](#)
- [Text Elements](#)
- [Structural Elements](#)

CHAPTER 51

[Web Application Code Components](#)

512

- [Code Components for Stateless Requests](#)
- [Error Pages](#)
- [Code Components for Session Requests](#)
- [Generic Code Components](#)

CHAPTER 52

[CommuniGate Pro WebMail Guide](#)

537

- [WebUser Interface Pages](#)
- [WebUser Interface Settings](#)
- [Password Modification](#)
- [Public Info Editor](#)
- [Automated Rules](#)
- [RPOP Accounts](#)

CHAPTER 53

CommuniGate Pro WebMail: MailBoxes

542

- [Access to Mailboxes](#)
- [Mailbox Browsing](#)
- [Mailbox Management](#)
- [Mailbox Subscription Management](#)
- [Mailbox Aliases Management](#)
- [Access to Mailboxes by Name](#)

CHAPTER 54

CommuniGate Pro WebMail: Messages

549

- [Message Browsing](#)

CHAPTER 55

CommuniGate Pro WebMail: Composing Messages

552

- [Opening the Composer Page](#)
- [Composer Settings](#)
- [Replying to Messages](#)
- [Forwarding Messages](#)
- [Attaching Files](#)
- [Delivery Status Notification](#)
- [Address Book](#)

CHAPTER 56

CommuniGate Pro WebMail: Secure Mail (S/MIME)

558

- [Public Key Infrastructure \(PKI\)](#)
- [Digital Signatures](#)
- [Certificates](#)
- [Private Key and Certificate Storage](#)
- [Private Key Activation](#)
- [Receiving Signed Messages](#)
- [Recording Certificates](#)
- [Sending Signed Messages](#)
- [Sending Encrypted Messages](#)
- [Receiving Encrypted Messages](#)

CHAPTER 57

Miscellaneous

567

- [Return Receipts](#)
- [Address Testing](#)
- [Adding Required Headers](#)
- [Legacy Mail Emulation](#)

CHAPTER 58

[CommuniGate Pro Licensing](#)

571

- [License Keys](#)
- [Single-Server Pricing](#)
- [Cluster Pricing](#)
- [Plugin Pricing](#)
- [Purchasing the License Keys](#)
- [Support](#)
- [Training](#)

CHAPTER 59

[How To](#)

577

- Routing
 - [How can I gradually migrate accounts from my old server?](#)
- SMTP Delivery
 - [How can I relay mail for certain domains?](#)
 - [How can I send mail to a remote host bypassing its DNS MX records?](#)
 - [How can I hold all client mail till their servers send ETRN?](#)
 - [How can I forward mail to the other SMTP MTA on the same server?](#)
 - [How can my customer servers receive mail if they have dial-up connections?](#)
 - [How can my customers release mail to all their domains with one ETRN or ATRN?](#)
 - [How can my customer servers receive mail if they have dynamic IP addresses?](#)
- Rules
 - [How can I store all outgoing mail sent by all my users?](#)
 - [How can I restrict to whom my users can send mail?](#)
 - [How can I create an autoresponder that sends files or HTML messages?](#)
- Mailboxes
 - [How can I create and use Shared Mailboxes?](#)
 - [How can an Administrator clean User Mailboxes?](#)
- Personal Web Sites
 - [How can I provide username.domain.com personal Web Sites?](#)

- WebAdmin
 - [I have rerouted the Postmaster account and now I cannot log in as the Postmaster.](#)
 - [I have deleted the Postmaster account.](#)
 - [I have created a secondary Domain and now I cannot log into WebAdmin.](#)
 - [When I try to log in, I get the "access from your network is denied" error.](#)
- SMTP Receiving
 - [My Server does not accept mail from my Web script/applet.](#)
- SMTP Sending
 - [My Server cannot send mail to some host using SSL/TLS.](#)
- Access
 - [WebUser connections return the pink page saying "we do not provide Web Access to this domain".](#)
 - [WebUser sessions are disconnected almost immediately after login.](#)
 - [What does the "unassigned local network address" error mean?](#)
- Directory
 - [Microsoft LDAP \(Outlook and Outlook Express\) users cannot find Directory records.](#)
 - [Attempts to update Account Settings result in the directory record with the specified DN is not found error.](#)
- Date and Time
 - [The time stamps in messages sent or received with CommuniGate Pro are several hours off.](#)
- Logs
 - [Every time I access the WebAdmin interface, a Failure-type ROUTER record appears in the Log.](#)
 - [What do these DNR-16538\(xxx.xx.x.xx.rss.mail-abuse.org\) A:host name is unknown records mean?](#)
- Misc
 - [What is that UDP port the CommuniGate Pro Server opens on my system?.](#)
 - [How can I make my formmail-type CGI work with CommuniGate Pro?.](#)

- [Kernel](#)
- [Security](#)

- [International](#)
- [SMTP \(Simple Mail Transfer Protocol\)](#)
- [IMAP \(Internet Message Access Protocol\)](#)
- [POP and RPOP \(Post Office Protocol\)](#)
- [HTTP \(HyperText Transfer Protocol\)](#)
- [FTP \(File Transfer Protocol\)](#)
- [LDAP \(Lightweight Directory Access Protocol\)](#)
- [Mailing Lists](#)
- [ACAP \(Application Configuration Access Protocol\)](#)
- [DNR \(Domain Name Resolver\)](#)
- [SNMP \(Simple Network Management Protocol\)](#)
- [Web User Interface](#)

CHAPTER 62

[Appendix B: CommuniGate Pro History](#)

605



CommuniGate Pro Quick Start

This section will cover the basic steps to download, install and configure the software. After completing steps 1-4, you will have a live email server.

Note: This section is for setting up a single server, trial version of CommuniGate Pro with one domain. This configuration keeps most of the default system values and is designed for evaluation purposes only. To take full advantage of CommuniGate Pro's many features, after licensing the software, you should read the entire online guide (<http://www.stalker.com/CommuniGatePro/>) and configure your system accordingly.

1. Download CommuniGate Pro

Go to the Stalker Website and download the latest version of CommuniGate Pro for your operating system - <http://www.stalker.com/CommuniGatePro/default.html#Current>

2. Install

Follow the installation instructions for your operating system - <http://www.stalker.com/CommuniGatePro/Install.html>

Restart the system or start the CommuniGate Pro Server manually.

3. Before you begin

A. Find the postmaster password:

- On a Unix or BeOS system, open the file:
`{CGateBase}/Accounts/postmaster.macnt/account.settings`

- On a MacOS X (Darwin) system:
open the Terminal application, type the following command:
`sudo more`
`/var/CommuniGate/Accounts/postmaster.macnt/account.settings`
the sudo command will ask you for the "root" password.
- On a MS Windows system, open the file:
`{CGateBase}\Accounts\postmaster.macnt\account.settings`
- On an OS/400 system, use any ASCII-capable text editor to open the file:
`{CGateBase}/Accounts/postmaster.macnt/account.settings`

This text file contains a random password assigned to the postmaster account. Remember this password, as you will use it to login as the postmaster in the next section.

DO NOT EDIT the password with a text editor or vi command while the CommuniGate Pro services are running.

B. Ensure that your server is connected to the network and the MX record for the domain (i.e. testcompany.com) points to the server.

4. Initial Configuration

Connect to the server with any Web browser, using the port number 8010. In your browser, type the following URL: `http://your.server.domain:8010`, or `http://localhost:8010`, or `http://127.0.0.1:8010`.

Start configuring the Server by opening the General page under the Settings menu. Use the default username postmaster, with the password from the file above to login.

On the General Settings screen, enter the name of the domain this server should process into the Main Domain Name field (i.e. testcompany.com)

Under the Monitors menu, click on the Log button. Open your log file to ensure that all listeners were created. If any were unable to be created because the port was already in use, you will need to shut down the other program using the port (i.e. sendmail).

Choose the Accounts menu (you might have to login again) and enter some user accounts for your domain.

To access Webmail you might want a user-friendly domain name such as mail.testcompany.com.

You will have to tell CommuniGate Pro that mail.testcompany.com will be an alias of the testcompany.com domain. To do this, open the Domain Settings page and find the Aliases table. In an empty field, enter the mail.testcompany.com name and click the Update button. Now the CommuniGate Pro Server will know that mail.testcompany.com domain name is just a different name for the testcompany.com Domain it serves.

Your users can connect to their accounts via Webmail by entering the URL <http://mail.testcompany.com> and then entering their username/password.

To test your email server, first send some local messages between accounts (user1@testcompany, user2@testcompany.com, etc). Next try sending to other internet domains. Finally, ensure that your users can receive mail from outside your domain.

5. Prevent Open Relay

While evaluating the software, you might want to "close" your server for unauthorized SMTP relaying, so that you are not an open relay. The easiest way to accomplish this is to restrict anyone who is not within your internal network from using your server to send messages.

First, from the Web Admin Interface under the Settings menu, choose SMTP. On the SMTP Settings screen, scroll down to the Relaying section. Change the If Received from parameter to clients and then hit the Update button.

Secondly, open the Protection page from the Settings menu and click the Client IP Addresses link. Enter the IP addresses on your LAN(s), as well as IP addresses of other systems that should be allowed to use your server as a mail relay. Read the online doc for more info:

<http://www.stalker.com/CommuniGatePro/AntiSpam.html#Relay>

6. Need Help?

Check out the on-line Guide - <http://www.stalker.com/CommuniGatePro/>

or read the CommuniGate Pro Mailing List archive -

<http://mail.stalker.com/Lists/CGatePro/List.html>

or e-mail technical support cgp-support@stalker.com

7. Get licensed

When you're ready to purchase the CommuniGate Pro license, simply go to www.stalker.com, and click on the Order tab. Under the CommuniGate Pro section, click on the arrow that says Order Form. Fill in the required fields and print out the form. Sign the form and fax back to Stalker at 1 415 383 7461. Once we receive the form, we will e-mail or fax you the appropriate license keys. Click here to read more information on how to enter the license keys:

<http://www.stalker.com/CommuniGatePro/Licensing.html>



CommuniGate Pro

Messaging Server

Description

Welcome to CommuniGate Pro, the Internet Messaging server application implementing:

- [Multi-domain](#) support with and without IP multihoming.
- [Multi-mailbox](#) accounts and [shared mailboxes](#).
- Internet mail exchange service using the [ESMTP](#) protocol.
- [Anti-Spam](#) and other [protection](#) mechanisms.
- Mailbox access with any [POP3](#) mail client.
- Mailbox access with any [IMAP4](#) mail client.
- Mailbox access using Microsoft® Windows [MAPI](#) interface (including the Microsoft Outlook in the "groupware" mode).
- Remote mailbox access using any [Web browser](#) (customizable "WebMail").
- Personal [Web Sites](#) for every account, with HTTP and [FTP](#) access.
- Mailing [List manager](#) with automatic bounce processing and a Web interface to list archives.
- Local and Distributed [Directory](#) and [LDAP](#) services.
- [Distributed Domains](#) for Distributed Multi-Server configurations.
- Multi-Server [Cluster](#) support for large sites (250,000 - 5,000,000 active accounts).
- Cluster of [Clusters](#) support for extra-large sites (more than 5,000,000 active accounts).
- [Dial-up](#) support for very small sites; [ETRN](#), [ATRN](#), and [UDWA](#) mail delivery.
- [Remote Accounts](#) polling using the POP3 protocol.
- Remote administration using any [Web browser](#).
- Remote administration using the [CLI](#) module.
- Remote monitoring using any [SNMP](#) Manager.
- Secure Authentication ([SASL](#)) and Secure Transfer ([SSL/TLS](#)) for SMTP, IMAP, POP, HTTP, LDAP, ACAP, PWD and Administration sessions.
- [Automated Mail Processing](#) Rules.
- High performance interface for external [anti-virus and content filtering](#) programs.

- Remote password modification using the [poppwd](#) protocol.
- Optional [LDAP-based provisioning](#) for all Domain types.
- [Mailboxes](#) in the form of text files, file folders, and databases.
- The [ACAP](#) address book and roaming support, and [Netscape® Roaming](#) via HTTP.
- The [Notification Alerts](#) mechanism.
- The [Events](#) mechanism for proactive server monitoring.

Features

The CommuniGate Pro Server is based on the Internet Standards (RFCs) and it has many additional features required for today's industrial-level messaging systems. The [Features](#) table can be used to compare the CommuniGate Pro with other systems available on the market today.

Administrating

CommuniGate Pro Server can be configured remotely (via the Internet) using any Web browser. CommuniGate Pro has a built-in HTTP (Web) server so it does not require any additional Web Server application to support remote configuration features, and it does not conflict with any other Web Server running on the same computer.

Remote administration features include:

- configuring the Server, Router, and all communication modules;
 - creating, removing and updating of user account information;
 - monitoring modules activity;
 - monitoring System Logs;
 - working with Server queues and individual messages in the Server queues.
-

Support and Discussions

Please subscribe to the CGatePro@stalker.com mailing list to discuss the CommuniGate Pro related issues. To subscribe to this mailing list, please send any message to CGatePro-on@stalker.com. Since the traffic on this mailing list is rather high, you may want to subscribe in the Digest mode. To subscribe to the List in the Digest mode, send a message to CGatePro-digest@stalker.com. A searchable archive of the CommuniGate Pro list is available at

<http://mail.stalker.com/Lists/CGatePro/List.html>

If you do not feel comfortable sending your questions to the mailing list, we will promptly answer your letters sent to cgp-support@stalker.com.

Stalker Software contact and general information is available at <http://www.stalker.com>.

What Is New in CommuniGate Pro Updates?

CommuniGate Pro is being updated on a regular basis.

You can review the history of updates and bug fixes at the [CommuniGate Pro History page](#).



CommuniGate Pro

Installation

You should download the CommuniGate Pro software either from the Stalker [Web/FTP](#) site, or from any authorized mirror site. Make sure you have the latest version of the software, and that the downloaded version is designed to work on the selected platform. Install the Server following the instructions below, and then proceed with [Initial Configuration](#).

Installation

On all systems, the CommuniGate Pro uses 2 directories (folders):

- the *application directory* containing the Server application and supplementary files (as Web Interface page templates). Files in this folder are not modified when the system runs.
- the *base directory* containing queued messages, account files, settings, and other server data.

Installing on a MS Windows XP/2000/NT/9x/ME System

- Use any "unzip"-type tool to unpack the `CGatePro-Win32-Intel-version.zip` file. **The tool should be able to use long file names.**
- Some "unzip" applications have the Install option, use that option if available. If the Install option is not available, unpack the archive. The unpacked directory should contain the CommuniGate Pro application folder and the `Installer.exe` application. Launch the `Installer.exe` application.
- If the CommuniGate Pro software is running, the Installer asks your permission to stop it.
- The Installer asks you where to place the CommuniGate Pro application folder, and where to create the "base" directory. If a previous version of CommuniGate Pro has been already installed, the Installer shows the locations used, and the Install button is renamed into the Update button.
- Click the Install/Update button to copy the CommuniGate Pro software into the selected location. If the CommuniGate "base" folder does not exist, the Installer creates an empty folder in the selected location.
- The information about the selected locations is stored in the System Registry.

Windows NT/2000/XP

The CommuniGate Pro Messaging Server (the `CGStarter.exe` application) is registered as a *service* that starts automatically when the system starts. This small application starts the `CGServer.exe` - the CommuniGate Server application. The Installer asks if you want to start the Server now.

Note: you should use the Services control panel to verify or change the Log On as name for the CommuniGate Pro service. That name should have the `Act as part of the operating system` Windows NT privilege. If the CommuniGate Pro Server does not have this privilege, not only it will fail to authenticate users using the Windows NT password system, but an attempt to use an incorrect password may cause the server to crash. This problem is fixed in the Windows NT Service Pack 4.

Note: if your server should serve 100 accounts or more, check the description of the [TIME_WAIT problem](#) and follow the instructions to decrease the NT `TIME_WAIT` time interval.

Windows 95/98/ME

The CommuniGate Pro (the `CGServer.exe` application) is added to the `RunServices` Registry key, so the server is started up when the system starts. Restart the Windows 9x system to start the server.

Note: Unlike Windows 98, the Windows 95 system does not have the "WinSock2" library installed. Download this library (.dll) from the <http://www.microsoft.com> and install it before you try to launch the CommuniGate Pro Server.

- Launch the Server and proceed with [Initial Configuration](#).

You can also start the CommuniGate Pro server manually, as a "console application", by launching the `CGServer.exe` file. If started without parameters, the Server creates the `C:\CommuniGatePro` folder and uses it as its "base" folder". If you want to use any other location, use the `--Base` command line parameter:

```
CGServer.exe --Base D:\OtherDirectory
```

Installing on a Sun Solaris System.

- Log in as a super-user (root).
- Unpack the CommuniGate Pro archive with the `gtar` command (or with the `gunzip` and `tar` commands):

```
gunzip CGatePro-Solaris-version.tar.gz
tar -xpf CGatePro-Solaris-version.tar
```
- Install the CommuniGate Pro package:

```
pkgadd -d .
```

The CommuniGate Pro software will be installed in the `/usr/local/sbin` directory.

- If your system was running `sendmail` or any other SMTP server, stop that server and remove that server start-up script from the `/etc/rcn/` directory, so the system will not start that other SMTP server automatically.
 - If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
 - The Installer creates a symbolic link `/bin/cgmail` for the command line mode mail program to use with the CommuniGate Pro system.
 - The Installer creates a startup script `/etc/init.d/STLKCGPro.init`, and the symbolic link `/etc/rc2.d/S88CommuniGate` for that script.
 - The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the `/etc/init.d/STLKCGPro.init` file and update it.
 - Restart the system or launch the start-up script manually:

```
/etc/init.d/STLKCGPro.init start
```
 - Proceed with [Initial Configuration](#).
-

Installing on a Linux System.

- Make sure you are running Linux kernel 2.0.34 or better.
- Log in as a super-user (root).
 - Using the Red Hat Package Manager (the .rpm file):

```
rpm -i CGatePro-Linux-version.rpm
```
 - Using other systems (the .tgz file):

```
tar -xzf CGatePro-Linux-version.tgz  
cd CGateProSoftware  
sh install.sh
```

The CommuniGate Pro software will be installed in the `/usr/local/sbin` directory.

- The installer will create the `/etc/rc.d/init.d/CommuniGate` startup script. To make the CommuniGate Pro Server start and stop automatically when the system starts up and shuts down, the installer adds the startup file links to the `/etc/rc.d/rcn.d` directories.
- If your system was running some standalone SMTP server/MTA (such as `sendmail`), stop that server (for example, you can use the `/sbin/chkconfig sendmail off` command and then restart the server computer).
- If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file (or put the hash sign (#) into the first position of those lines).

- The Installer renames the `/bin/mail` program into the `/bin/LegacyMail`. If you decide to uninstall the CommuniGate Pro system, the legacy mail program will be renamed back to `/bin/mail`.
- The Installer creates the new `/bin/mail` application - a drop-in substitution for the legacy mail program.
- The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the `/etc/rc.d/init.d/CommuniGate` file and update it.
- Restart the system or launch the start-up script manually:
`/etc/rc.d/init.d/CommuniGate start`
- Proceed with [Initial Configuration](#).

Note: Under Linux, each thread in a multi-threaded application is seen as a "process" when you use the `ps` and `top` system utilities. As a result, you may see 30+ CGServer "processes" when the server is just started, and more after it has been actively used. All those "processes" are actually CommuniGate Pro Server threads, and they share all their resources - VRAM, File Descriptors, etc.

Installing on a MacOS X (Darwin) System.

- Log in as a user with the administrative rights.
- Unpack the CommuniGate Pro archive using any uncompressing utility, or start the Terminal application and use the shell `tar` command:
`tar xzpf CGatePro-Darwin-platform-version.tgz`
the `CommuniGate.pkg package directory` will be created in the current directory.
- Install the software by double-clicking the `CommuniGate.pkg` icon.
The CommuniGate Pro software will be installed in the `/usr/sbin/` directory.
- Note: The Installer creates the startup directory
`/System/Library/StartupItems/CommuniGatePro`, so the CommuniGate Pro Server will auto-start when the MacOS X System starts.
- Note: The Installer packs the startup directory
`/System/Library/StartupItems/Sendmail` into the
`/System/Library/StartupItems/Sendmail.tar` startup archive, so the legacy `sendmail` daemon will not auto-start. If you decide to uninstall CommuniGate Pro, this archive will be unpacked and the `Sendmail` startup directory will be restored.
- If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
- The Installer renames the `/usr/bin/mail` program into the `/usr/bin/LegacyMail`. If you decide to uninstall the CommuniGate Pro system, the legacy mail program will be renamed back to `/usr/bin/mail`.
- The Installer creates the new `/usr/bin/mail` application - a drop-in substitution for the legacy

mail program.

- The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the `/System/Library/StartupItems/CommuniGatePro/CommuniGatePro` file and update it.
 - Restart the MacOS X System.
 - Proceed with [Initial Configuration](#).
-

Installing on a MacOS X Server (Rhapsody) System.

- Log in as a super-user (root).
 - Unpack the CommuniGate Pro archive using any uncompressing utility, or use the shell (Terminal.app) `guntar` command:

```
guntar xzpf CGatePro-Rhapsody-version.tgz
```

the `CommuniGate.pkg` package directory will be created in the current directory.
 - Install the software by double-clicking the `CommuniGate.pkg` icon. The CommuniGate Pro software will be installed in the `/Local/Servers` directory.
 - Note: The installer will create a file `/etc/startup/1950_CommuniGate`, so the CommuniGate Pro Server will auto-start when the System starts.
 - Note: The installer will disable the file `/etc/startup/1800_Mail` (`/etc/startup/1900_Mail` on DR Rhapsody versions) used to auto-start `sendmail` on your system.
 - If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
 - The Installer renames the `/usr/bin/mail` program into the `/usr/bin/LegacyMail`. If you decide to uninstall the CommuniGate Pro system, the legacy mail program will be renamed back to `/usr/bin/mail`.
 - The Installer creates the new `/usr/bin/mail` application - a drop-in substitution for the legacy mail program.
 - The Installer creates the "base directory" `/Local/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the `/etc/startup/1950_CommuniGate` file and update it.
 - Restart the system or launch the startup script manually:

```
/etc/startup/1950_CommuniGate
```
 - Proceed with [Initial Configuration](#).
-

Installing on a FreeBSD System.

- Make sure you are running FreeBSD version 4.0 or better.
 - Log in as a super-user (root).
 - Install the CommuniGate Pro package:
`pkg_add CGatePro-FreeBSD-version.tgz`
 The CommuniGate Pro software will be installed in the `/usr/local/sbin` directory.
 - If your system was running `sendmail` or any other SMTP server, stop that server and modify the OS start-up scripts so the system will not start that other SMTP server automatically.
 - If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
 - The Installer creates a script `/usr/local/etc/rc.d/CommuniGate.sh`, so the CommuniGate Pro Server is started automatically when the FreeBSD system starts.
 - The Installer creates a symbolic link `/bin/cgmail` for the command line mode mail program to use with the CommuniGate Pro system.
 - The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the start-up script file and update it.
 - Restart the system or launch the start-up script manually:
`/usr/local/etc/rc.d/CommuniGate.sh start`
 - Proceed with [Initial Configuration](#).
-

Installing on a BSDI BSD/OS System.

- Make sure you are running BSD/OS version 4.0.1 or better.
- Log in as a super-user (root).
- Unpack the CommuniGate Pro archive with the `gunzip` and `tar` commands:
`gunzip CGatePro-BSDI-version.tar.gz`
`tar -xpf CGatePro-BSDI-version.tar`
- Install the CommuniGate Pro package:
`installsw -c ../../CGatePro` (use the full path to the CGatePro directory)
 The CommuniGate Pro software will be installed in the `/usr/local/sbin` directory.
- If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
- If your system was running `sendmail` or other mail transfer agent, remove the lines describing those servers from the `/etc/rc` file.
- The Installer adds a reference to the CommuniGate Pro startup script to the `/etc/rc.local` file, so the CommuniGate Pro Server is started automatically when the BSDI BSD/OS system starts.
- The Installer creates a symbolic link `/usr/bin/cgmail` for the command line mode mail program to use with the CommuniGate Pro system.

- The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the start-up script file (`/usr/local/sbin/CommuniGate/Startup`) and update it.
 - Restart the system or launch the start-up script manually:
`/usr/local/sbin/CommuniGate/Startup start`
 - Proceed with [Initial Configuration](#).
-

Installing on an OpenBSD System.

- Make sure you are running OpenBSD version 2.8 or better.
 - Log in as a super-user (root).
 - Install the CommuniGate Pro package:
`pkg_add CGatePro-OpenBSD-version.tgz`
The CommuniGate Pro software will be installed in the `/usr/local/sbin` directory.
 - If your system was running `sendmail` or any other SMTP server, stop that server and modify the OS start-up scripts so the system will not start that other SMTP server automatically.
 - If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
 - The Installer adds a reference to the CommuniGate Pro startup script to the `/etc/rc.local` file, so the CommuniGate Pro Server is started automatically when the OpenBSD system starts.
 - The Installer creates the mail group if it was absent.
 - The Installer creates a symbolic link `/bin/cgmail` for the command line mode mail program to use with the CommuniGate Pro system.
 - The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the start-up script file (`/usr/local/sbin/CommuniGate/Startup`) and update it.
 - Restart the system or launch the start-up script manually:
`/usr/local/sbin/CommuniGate/Startup start`
 - Proceed with [Initial Configuration](#).
-

Installing on an AIX System.

- Make sure you are using the AIX System version 4.3 or better.
- Log in as a super-user (root).
- Unpack the CommuniGate Pro archive with the `compress` command:
`compress -d CGatePro-AIX-version.bff.Z`
- Use either the `installp` command, or the `smitty` utility, or the `smitty` utility to install the

software.

- The installation script creates a startup script `/etc/rc.cgpro` and updates the `/etc/inittab` file to start the CommuniGate Pro server on run level 2.
 - The startup script creates the "base directory" `/var/CommuniGate` and the Server uses this directory by default. You can move the "base directory" to any other location. In this case, open the `/etc/rc.cgpro` script file and update it.
 - If your system was running `sendmail` or any other SMTP server, stop that server and modify the OS start-up scripts so the system will not start that other SMTP server automatically.
 - If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
 - Restart the system or launch the start-up script manually:
`/etc/rc.cgpro start`
 - Proceed with [Initial Configuration](#).
-

Installing on an HP/UX System.

- Make sure the HP/UX version 11 is installed.
- Apply all patches available for HP-UX 11.00, or, at least the threads-related ones.
- Set the kernel parameters as follows:
 - set the `maxdsize` kernel parameter to 50MB or more.
 - set the `max_thread_proc` (max number of threads per process) to 128 or more.
 - set the `ncallout` parameter (max number of pending timeouts) to 128 or more.
 - set the `maxfiles` parameter to 512 or more.
- Log in as a super-user (root).
- Unpack the CommuniGate Pro archive with the `gunzip` and `tar` commands:

```
gunzip -xzf CGatePro-HPUX-HPPA-version.tar.gz
tar -xf CGatePro-HPUX-HPPA-version.tar
```

The `CGatePro.depot` directory should appear in the current directory.
- Install the CommuniGate Pro package:
`swinstall -s `pwd`/CGatePro.depot` (you should use the absolute path for the unpacked `CGatePro.depot` directory)
The CommuniGate Pro software will be installed in the `/opt/CommuniGate` directory.
- If your system was running `sendmail` or any other SMTP server, stop that server and modify the OS start-up scripts so the system will not start that other SMTP server automatically.
- If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
- The server startup script is created as `/sbin/init.d/CommuniGate` and its symbolic links `/sbin/rc2.d/S80CommuniGate` and `/sbin/rc1.d/K80CommuniGate` are automatically created.

- The Server uses `/var/CommuniGate` as its "base directory" by default. You can move the "base directory" to any other location. In this case, open the `/sbin/init.d/CommuniGate` script file and update its `BASEDIRECTORY` parameter.
 - The symbolic link `/bin/cgmail` is created for the CommuniGate "mail" program.
 - Restart the system or launch the start-up script manually:
`/sbin/init.d/CommuniGate start`
 - Proceed with [Initial Configuration](#).
-

Installing on a Tru64 (Digital Unix) System.

- Make sure the Tru64 version 4.0e with the update package #2 (Sept/98) or Tru64 version 5 or better is installed.
 - Log in as a super-user (root).
 - Unpack the CommuniGate Pro archive with the `gtar` command (or with the `gunzip` and `tar` commands):
`gtar -xzf CGatePro-Tru64-platform-version.tar.gz.`
 - Install the CommuniGate Pro package:
`/usr/sbin/setld -l CGatePro.pkg`
The CommuniGate Pro software will be installed in the `/usr/opt/` directory.
 - If your system was running `sendmail` or any other SMTP server, stop that server and modify the OS start-up scripts so the system will not start that other SMTP server automatically.
 - If your system was running POP, IMAP, or `poppwd` servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
 - The Installer creates the symbolic link `/sbin/init.d/CommuniGate` for the server startup script.
 - The Installer also creates the `/sbin/rc0.d/K10CommuniGate`, `/sbin/rc2.d/K10CommuniGate`, and `/sbin/rc0.d/S80CommuniGate` startup symbolic links.
 - The Server uses `/var/CommuniGate` as its "base directory" by default. You can move the "base directory" to any other location. In this case, open the `/usr/opt/CGPversion/startup` script file and update its `BASEFOLDER` parameter.
 - Restart the system or launch the start-up script manually:
`/sbin/init.d/CommuniGate start`
 - Proceed with [Initial Configuration](#).
-

Installing on an SGI IRIX System.

- Make sure you are using the IRIX System version 6.5 or better.
 - Log in as a super-user (root).
 - Install the CommuniGate Pro package:

```
inst -f CGatePro-IRIX-MIPS-version.tardist
```

or

```
swmgr -f CGatePro-IRIX-MIPS-version.tardist
```

The CommuniGate Pro software will be installed in the `/opt/CommuniGate` directory.
 - If your system was running POP, IMAP, or poppwd servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
 - If your system was running sendmail, use the `chkconfig` to disable it:

```
/sbin/chkconfig sendmail off
```

The CommuniGate Pro installation script writes the word `off` into the `/etc/config/sendmail` file.
 - The installation script creates a startup script `/etc/init.d/CommuniGate`, and the symbolic links `/etc/rc2.d/S75CommuniGate` and `/etc/rc0.d/K05CommuniGate` for that script.
 - The installation script creates the "base directory" `/var/CommuniGate` and the Server uses this directory by default. You can move the "base directory" to any other location. In this case, open the `/etc/init.d/CommuniGate` script file and update it.
 - Restart the system or launch the start-up script manually:

```
/etc/init.d/CommuniGate start
```
 - Proceed with [Initial Configuration](#).
-

Installing on an SCO UnixWare System.

- Make sure that UnixWare 7.1 or better is installed.
- Log in as a super-user (root).
- Create a new directory, and "cd" into that directory; download the `CGatePro-UnixWare-version.tar.gz` archive.
- Unpack the CommuniGate Pro archive with the `gunzip` and `tar` commands):

```
gunzip CGatePro-UnixWare-version.tar.gz
```

```
tar -xpf CGatePro-UnixWare-version.tar
```
- Install the CommuniGate Pro package:

```
pkgadd -d `pwd`
```

The CommuniGate Pro software will be installed in the `/usr/local/sbin` directory.
- If your system was running sendmail or any other SMTP server, stop that server and modify the OS start-up scripts so the system will not start that other SMTP server automatically.
- If your system was running POP, IMAP, or poppwd servers, remove the lines describing those servers from the `/etc/inetd.conf` file.
- The Installer creates a symbolic link `/bin/cgmail` for the command line mode mail program to

use with the CommuniGate Pro system.

- The Installer creates a startup script `/etc/init.d/STLKCGPro.init`, and the symbolic link `/etc/rc2.d/S88CommuniGate` for that script.
- The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the `/etc/init.d/STLKCGPro.init` file and update it.
- Restart the system or launch the start-up script manually:
`/etc/init.d/STLKCGPro.init start`
- Proceed with [Initial Configuration](#).

Note: UnixWare 7.1 has a very small per process limit for open listeners. To avoid the problem, the CommuniGate Pro ACAP server is disabled by default, and the LDAP server does not create a listener to accept secure connections. Do not try to create additional listeners before this limit is increased.

Installing on an IBM OS/400 System.

- Make sure you have OS/400 version V4R4M0 or later.
- Decide where to place the CommuniGate Pro *base directory*:
 - The files used by the CommuniGate Pro Software must reside in a thread-safe file system. Only the following file systems are thread-safe:
`/(root), QOpenSys, QNTC, QSYS.LIB, QOPT, QLANsrv, user-defined`.
For details see the Integrated File System concepts book in the [AS/400 Information Center](#).
 - If you need to have case-sensitive mailbox names you should place the CommuniGate Pro *base directory* into a case-sensitive file system. The QOpenSys file system is case-sensitive. User-defined file systems can be created as case-sensitive, too. The Installer program allows you to create a case-sensitive file system for the CommuniGate Pro *base directory*.
Note: If you are upgrading the CommuniGate Pro Server and the *base directory* already exists, then installer will ignore the case-sensitivity option you have specified, and will leave the *base directory* unmodified.
- If your OS/400 system was running SMTP, POP, IMAP, ACAP, or poppwd servers, stop those servers, and modify the start-up defaults, so the system will not start those legacy servers automatically. You can change the start-up defaults using the AS400 Operations Navigator.
- Make sure your OS/400 system has its TCP/IP stack active and the FTP server running.
- Download the current CommuniGatePro OS400 version (the CGatePro-OS400-AS400.exe file) to a computer running MS Windows OS and connected (via a TCP/IP network) to your OS/400 system.
- Run the CGatePro-OS400-AS400.exe installer program on that MS Windows system. Follow the instructions the installer program provides.
- On the OS/400 system, start the CommuniGate Pro Server job: QGPL/STRCGSRV. The

CGSERVER job will start in the QSYSWRK subsystem.

- You may want to create an autostart job entry for the CommuniGate Pro Server, or run it in a specially created subsystem. For details see OS/400 Work Management book in [AS/400 Online Library](#).
 - **Note:** the source code of the STRCGSRV and ENDCGSRV commands is included into the CGatePro distribution set, so you can modify those commands to meet your needs.
 - Proceed with [Initial Configuration](#).
-

Installing on a BeOS System.

- Make sure that BeOS R5 or better is installed.
- Download the CommuniGate Pro package: `CGatePro-BeOS-version.pkg`.
- Double-click the package file and install it.
The CommuniGate Pro software will be installed in the `/boot/Servers/` directory.
- If your system was running `sendmail` or any other SMTP server, stop that server and modify the OS start-up scripts so the system will not start that other SMTP server automatically.
- If your system was running POP, IMAP, or `poppwd` servers controlled with the `inetd` daemon, remove the lines describing those servers from the `/etc/inetd.conf` file.
- The Installer adds commands to the `/boot/home/config/boot/UserBootscript` and `/boot/home/config/boot/UserShutdownFinishScript` files. Those commands call the CommuniGate Pro startup script when the BeOS starts and shuts down.
- The Installer creates the "base directory" `/var/CommuniGate` and the Server uses it by default. You can move the "base directory" to any other location. In this case, open the `/boot/Servers/CommuniGate/Startup` file and update it.
- Restart the system or launch the start-up script manually:
`/boot/Servers/CommuniGate/Startup start`
- Proceed with [Initial Configuration](#).

Note: Under BeOS, each thread in a multi-threaded application is seen as a "process" when you use the `ps` system utility. As a result, you may see 30+ CGServer "processes" when the server is just started, and more after it has been actively used. All those "processes" are actually CommuniGate Pro Server threads, and they share all their resources - VRAM, File Descriptors, etc.

Initial Configuration

- Restart the system or start the CommuniGate Pro Server manually.

- On a Unix system: use either the super-user account or any account included into the "mail" group to access the CommuniGate Pro files in the "base directory" (by default that directory is /Local/CommuniGate for MacOS X Rhapsody, /var/CommuniGate/ for other Unix systems).

On a Unix or BeOS system, open the file:

`{CGateBase}/Accounts/postmaster.macnt/account.settings`

On a MacOS X (Darwin) system:

open the Terminal application, type the following command:

`sudo more`

`/var/CommuniGate/Accounts/postmaster.macnt/account.settings`

the sudo command will ask you for the "root" password.

On a MS Windows system, open the file:

`{CGateBase}\Accounts\postmaster.macnt\account.settings`

On an OS/400 system, use any ASCII-capable text editor to open the file:

`{CGateBase}/Accounts/postmaster.macnt/account.settings`

This text file contains a random password assigned to the postmaster account. Remember this password.

- Connect to the server with any Web browser, using the port number 8010. In your browser, type the following URL:
`http://your.server.domain:8010`
where your.server.domain is the domain name or the IP address of the computer running the CommuniGate Pro server.
- To open Setup, Accounts, Monitor, and Master configuration pages (realms), use the postmaster username and the password you have seen in the postmaster.account.settings file.
- Check the [General settings](#). Make sure the date/time settings of the Server operating system are set properly.
Check the Server (Main Domain) name, and modify it if needed. Read the [Domains](#) section of the manual to learn how CommuniGate Pro supports multiple Domains and Domain Aliases, and select the proper Main Domain name.
- Check the [Protection](#) section of the manual to learn how to "close" your Server for unauthorised SMTP relaying.
- Open the [Accounts](#) page and register your users.
- If you have any problem with configuration, check the [How To](#) and [Help Me](#) sections of the manual.

The [Migration](#) section can help you to schedule your CommuniGate Pro deployment process.

Upgrading to a Newer Version

When you upgrade to a new version, the files in the *application directory* are substituted with the

new files.

The *base directory* and all its files are not modified when you upgrade the CommuniGate Pro Server software, so all accounts, mailboxes, messages, settings, Personal Web Sites, Licenses, customized WebUser and WebAdmin files stay in place and continue to work with the new version of the CommuniGate Pro software.

Note: if you chose to manually modify WebUser and/or WebAdmin files right in the application directory, then you should save them before upgrading.

To upgrade:

- Download the new version of the CommuniGate Pro Software.
- Stop the CommuniGate Pro Server.
- Remove the previous version of the software using the same installation utility you used to install the software (the *base directory* will not be removed). This step is needed if the OS installer does not allow you to install a new version "on top" of the old one (Solaris, FreeBSD, Linux).

Note: If you are using the Linux rpm package manager, do not use its "update" option: uninstall the old version, then install the new one:

```
rpm -e CGatePro-Linux
```

```
rpm -i CGatePro-Linux-version.rpm
```

- Install the new version of the CommuniGate Pro Software
- Start the CommuniGate Pro Server.

Moving to a New Hardware Server

You may want to move your CommuniGate Pro server to a different computer - running the same or a different OS. All your module settings, account and domain settings, mailboxes, licenses, and other data can be preserved.

CommuniGate Pro keeps all its data in the [base directory](#). This is the only directory you need to copy to your new server computer.

CommuniGate Pro uses the same file formats on all hardware and software (OS) platforms, so usually you can just pack the entire CommuniGate Pro base directory into an archive file (using tar and gzip on Unix systems, zip on MS Windows systems), and unpack the archive on the new server computer.

Additional processing is needed when you move the CommuniGate Pro Server from a computer running any MS Windows operating system to a computer running any flavor of Unix, or vice versa. CommuniGate Pro files are text files, and text files on MS Windows and Unix have different EOL (end of line) symbols: CR-LF (return - linefeed) on MS Windows and bare LF (linefeed) symbol on Unix systems. To copy files properly, you may want to use any FTP software to copy files between those systems: when an FTP client is instructed to transfer files in the ASCII mode, it properly converts EOL symbols.

Note: CommuniGate Pro base directory can contain non-text (binary) files in the `WebUser` and `WebAdmin` directories inside the `Account` and `Domains` subdirectories - graphic files used in the customized WebUser and WebAdmin Interfaces. Binary files can also be stored in `Personal Web Sites` - `account.web` directories inside the `account` directories. When you move a CommuniGate Pro base directory between systems using different EOL conventions, check that those binary files are copied in the BINARY mode (i.e. without EOL recoding).

If the new server computer is running a Unix system, check that the copied directory and all its files and subdirectories have the same access rights as they had on the old system.

After the CommuniGate Pro base directory is copied, download and install the proper version of the CommuniGate Pro Server on the new server computer. There is no need to copy the content of [application directory](#) from the old server computer, even if both new and old computers are running the same operating system.

Check that the newly installed copy of the CommuniGate Pro Server (its startup script, if any) is configured to use the copied base directory, and then start the CommuniGate Pro Server on the new computer. Use the WebAdmin Interface to modify the computer-related settings on the new server. For example, you may need to update the Client IP Addresses table or re-assign IP addresses to CommuniGate Pro Domains.



Migrating to CommuniGate Pro

If your system was already running some type of mail server, you may want to integrate your existing E-mail environment into the CommuniGate Pro messaging system.

To migrate all your users, you need:

- to create all existing mail accounts on your new CommuniGate Pro Server, using already specified account settings, especially - account passwords
- to migrate all user E-mail from the old server to the new CommuniGate Pro Server
- provide services for "local users" - the users of "local" (Unix) mail programs that directly access their mailbox files, bypassing E-mail protocols (such as the POP or IMAP protocols).

Supporting Network Users

Users that access their mail accounts using any standard Internet Protocol (POP, IMAP) do not have to switch their mailer applications or change mailer settings - CommuniGate Pro supports not only the published mail access protocols standards, but most of unofficial protocol extensions, too.

Supporting Local Users

Some users registered with the server OS may access their mail accounts using legacy mailer applications that read mailbox files directly. Since these mailers bypass Internet protocols, the CommuniGate Pro server has no control over those mailers.

The CommuniGate Pro Server keeps all user accounts and mailboxes inside its "base directory". On a properly configured system direct user access to the base directory is prohibited to ensure that account mailboxes and other Server files stay intact.

When you create a CommuniGate Pro account for a local user who needs mail access via legacy mailers, select the [External INBOX](#) option. In this case, the account INBOX will not be created

inside the CommuniGate Pro *base directory*. Instead, the mailbox location will be taken from the [user domain settings](#). Usually, you specify `/var/mail/*` or `/var/spool/mail/*` - the "standard" location where legacy mailers expect to see user mailboxes.

When the Server accesses these *external mailboxes*, it uses the OS file-locking mechanism to synchronize its activity with legacy mailers.

Note: legacy mailers were not designed to support simultaneous access to mailboxes. They can destroy data in your mailbox if you open two legacy mailer sessions with the same mailbox and delete messages in one of the sessions. CommuniGate Pro cannot fix this, since these mailers bypass the server, it can only guarantee (using file locks) that legacy mailers do not destroy a mailbox while CommuniGate Pro is working with it.

For more details, see the [Sharing](#) section.

Using Legacy Mailboxes

The default format for CommuniGate Pro mailboxes is the BSD-type text mailbox format: a mailbox is a text file with messages separated with an empty line and a line starting with the symbols `From` .

Since most legacy mail systems use this format, the existing mailbox files can be used when migrating to CommuniGate Pro. You should either copy the old mailbox files into the proper places in the CommuniGate Pro account directories, or you can specify that some accounts have external INBOXes (see above), and the old mailbox files will be used "in place".

Note: when CommuniGate Pro stores a message in a BSD-type mailbox, it adds additional fields to the separator (`From`) line. These fields are ignored by legacy mailers and mail servers, but they allow the CommuniGate Pro Server to keep the message status and unique message ID information. When you make your CommuniGate Pro Server use a BSD-type mailbox composed with an old mail system, it issues warnings (Log records) about missing fields in the message separators. The Server still opens such mailboxes: it creates empty status flag sets for such messages, and it generates unique IDs on-fly. As users read, move, and/or delete old mail from their mailboxes, messages stored with the old mail system disappear, and the Server stops complaining when opening these mailbox files.

Converting Passwords

If your old mail server authenticated clients using the Unix OS account and passwords (the `passwd` and `shadow` files), you can simply enable the [Use OS Password](#) option for those accounts and the CommuniGate Pro Server will use the OS authentication procedures for them.

Since the OS Passwords are one-way encrypted, they cannot be used for secure SASL authentication methods. The following procedure can be used to allow migrated users to employ the secure SASL methods:

- enable the Use OS Password option either in the Default Account Settings or in the Account Template.
- specify an empty string for the CommuniGate Password in the Account Template.
- import all accounts without the Password field.

Users can connect to the newly created accounts using their old OS Passwords - i.e. the same passwords they used with the legacy mail system. When users try to modify their account passwords, the new passwords will be stored as CommuniGate Passwords. All users that have updated their passwords will be able to use the secure SASL authentication methods.

Sometimes you cannot use this method. For example, you migrate users from a different server, and you do not register them all with the Unix OS on the new server, but you do have the `passwd` file from the old server. In this case, you may want to enter the Unix-style (crypt-encrypted) passwords as the CommuniGate (internal) Passwords.

To make the CommuniGate Pro server process its internal password string as a U-crpt (crypt-encrypted) password, store it in the Account Settings with one-byte binary 002 prefix. If you want to create a user `test` using the [CLI](#) interface, and the Unix (crypt-encrypted) password for that user is `AslUzTlJkPsocc`, then use the following CLI command:

```
createaccount "test" {Password="\002AslUzTlJkPsocc"; }
```

If you create users by [importing](#) an account list from a text file, place the Unix passwords strings into the `UnixPassword` column, not into the `Password` column. In this case, the Loader will automatically add the binary 002 prefix to all password strings.

Account Settings of the new accounts should specify one of the CommuniGate Pro password encryption methods (clear text or A-crpt). Users will be able to log in using their old Unix passwords. When they update their passwords, new CommuniGate Password strings will be stored using the specified CommuniGate Pro password encryption method. All users that have updated their passwords will be able to use the secure SASL authentication methods.

Some servers produced by the Netscape and Software.com companies store user passwords using several encryption methods. When passwords are retrieved from those servers, they have the

following form:

`{method}eNcoDeD` where *method* specifies one of the standard encryption methods, and the *eNcoDeD* string is a base64-encoded encrypted password.

CommuniGate Pro can use these passwords in the same way it uses the Unix-encrypted passwords, and they should be entered in the same way: you should use the binary 002 prefix in the CLI commands and/or you should place those passwords into the UnixPassword field of the Import file.

The following encryption methods are supported:

- `{crypt}` - the standard Unix crypt method.
- `{WM-CRY}` - the standard Unix crypt method (same as `{crypt}`)
- `{MD5}` - the MD5 digesting method (Base-64 encoded MD5 digest of the password string).
- `{SHA}` - the SHA1 digesting method (Base-64 encoded SHA1 digest of the password string).
- `{NS-MTA-MD5}` - the MD5-based method used in the Post.Office servers and old Netscape Messaging servers (the eNcoDeD portion contains 64 hexadecimal digits).

Some BSD systems use the MD5 method of password encoding. Passwords of that type start with the `1` string, and the CommuniGate Pro Server checks them with its own internal routines. This feature allows you to migrate users (and their OS passwords) from BSD systems to a CommuiGate Pro Server running under an OS that does not support MD5-encrypted passwords.

Sample Import file:

Name	UnixPassword	Password Type
user2	YIdhkjeHDKbYsji	Unix-crypt
user1	{SHA}Ue4Erbim2TC7CmuukMOBejeytr2=	SHA1-digested
user3	{MD5}zverMUhsgJUIDjeytr2=	MD5-digested
user4	{crypt}YIdhkjeHDKbYsji	Unix-crypt, same as for the user1
user5	\$1\$VlPrB\$vNjOAYtB3W.j0bkkbaN2Z.	account BSD-type MD5-encrypted

You can use a CommuniGate Pro CLI script to automatically import all users and their passwords from the OS `/etc/passwd` file. See the [CommuniGate Perl Interface](#) site for a sample script.

Migrating from sendmail

If you are migrating from a sendmail-based mail system, you may find the following information useful:

The `aliases` file

The sendmail aliases file allows the administrator to redirect local mail to one or several addresses. Sendmail uses the term "alias" for too many different things, so you should select the proper CommuniGate Pro feature to implement different types of sendmail "aliases":

- Each account can have one or several [aliases](#). Mail sent to any account alias name is routed to the account itself. If the `domain.dom` account `john.smith` has `j.smith` and `smith` aliases, mail sent to `j.smith@domain.dom` and `smith@domain.dom` addresses is delivered to the `john.smith@domain.dom` account. When an account is renamed, its aliases automatically start to point to the new name, and when an account is removed, all its aliases are removed, too.
- Domain [Forwarder](#) objects allow you to redirect mail sent to a domain address to any other address. The `domain.dom` forwarder `susan.smith` can reroute all mail sent to `susan@domain.dom` address to `susan@otherisp.dom` address.
- Domain [Group](#) objects allow you to redirect mail sent to some domain address to any set of addresses.
- The [Router](#) module allows you to redirect mail sent to a certain address to any other address. The Router Alias Record `<*.smith@domain.dom> = Smith@domain.dom` will reroute mail sent to `john.smith@domain.dom` and to `susan.smith@domain.dom` to the `Smith@domain.dom` address.
- The [Account Rules](#) allow the administrator and/or the users themselves to redirect/forward/mirror all or certain mail to one or several addresses.
- The [Server-Wide Rules](#) allow the administrator to redirect/forward/mirror all or certain mail to one or several addresses.
- The shared or [foreign](#) mailboxes feature allows a user to grant access to a mailbox to other users; in many cases a shared mailbox is a much better alternative to mail distribution.
- The [LIST](#) module provides a very powerful mailing list distribution mechanism.

`procmail` processing

The Server-Wide, Domain-Wide, and Account-Level [Automated Rules](#) allow administrators and users to perform automatic mail processing and filtering using the powerful condition checking and processing operations built into the CommuniGate Pro Server.

For the situations where messages should be processed using an external filter or processor, the `Execute Automated Rules` operation can be used to start the specified external program as a separate OS task (for example, the Rules can be used to process all or certain incoming messages with the same `procmail` program).

Restricted Relaying

Unlike sendmail, CommuniGate Pro does not use syntax rules to prevent [unauthorized relaying](#).

Instead, it uses the [Router](#) and really checks if message delivery to a specified address would result in SMTP transfer to a "stranger" host.

Migrating from Post .Office® servers

The Post.Office product stores account names, passwords, and other information in its account database. The special [Post.Office Migration Utility](#) can be used to retrieve that information and to store it in a tab-delimited file that can be used with the CommuniGate Pro WebAdmin [Account Import](#) function.

Migrating from Netscape®/iPlanet Messaging servers

The Netscape (iPlanet) Messaging server stores account names, passwords, and other information in a Directory server "subtree". Use regular LDAP tools to export the Directory subtree into an LDIF file. The special [Netscape Migration Script](#) can be used to convert the account information from the LDIF format into a tab-delimited file that can be used with the CommuniGate Pro WebAdmin [Account Import](#) function.

Migrating from IMail® servers

The IMail product stores account names, passwords, and other information in its account database. The special [IMail Migration Utility](#) can be used to retrieve that information and to store it in a tab-delimited file that can be used with the CommuniGate Pro WebAdmin [Account Import](#) function.

Migrating from CommuniGate/MacOS and SIMS

If you want to move your users from a [CommuniGate for MacOS](#) server, you can build the account list file using the [CommuniGate/MacOS extractor](#) utility.

If you want to move your users from a Stalker Internet Mail Server ([SIMS](#)), you can build the account list file using the [SIMS extractor](#) utility.

Copying Mailboxes from Other POP Servers

When migrating from other mail servers, you may want to copy all messages from an account on the old server to the already created account on the new server.

The CommuniGate Pro software package includes the MovePOPMail program. This program connects to the old POP server, logs in, retrieves all messages, and submits it to the new SMTP server:

```
MovePOPMail [--verbose] [--delete] [--notimeout] POPserver POPname  
POPpassword SMTPserver SMTPrecipient
```

POPserver

The IP address of the old (source) POP3 server; if that POP server operates on a non-standard TCP port, you can specify the port number using the colon sign: 192.0.2.3:111 - POP server at the 192.0.2.3 address, port 111.

POPname

The POP account name - i.e. the name of the source account on the POP server.

POPpassword

The POP account password.

SMTPserver

The IP address of the new (target) SMTP server; if that SMTP server operates on a non-standard TCP port, you can specify the port number using the colon sign: 192.0.2.4:26 - SMTP server at the 192.0.2.4 address, port 26.

SMTPrecipient

The address to send the retrieved messages to. Usually - the account name on the new server.

--verbose

An optional parameter. When specified, the progress information is sent to the standard output.

--delete

An optional parameter. When specified, the program deletes all retrieved messages from the POP account.

--notimeout

An optional parameter. When specified, the program increases the SMTP and POP operation timeout values from 20 seconds to 1 hour.

Sample:

```
MovePOPMail --verbose 192.0.0.4 john "jps#dhj" 192.0.1.5 john
```

Copying Mailboxes from Other IMAP Servers

When migrating from other mail servers, you may want to copy all mailboxes and all messages from an account on the old server to the already created account on the new server.

The CommuniGate Pro software package includes the `MoveIMAPMail` program. This program connects to the old and new IMAP servers, logs into both, retrieves the list of mailboxes in the old account, creates all missing mailboxes in the new account, and copies all messages from mailboxes in the old account to the mailboxes in the new account. The program also copies the list of "subscribed mailboxes".

```
MoveIMAPMail [flags] OldServer oldName oldPassword NewServer  
newName newPassword
```

oldServer

The IP address of the old (source) IMAP4 server; if that IMAP server operates on a non-standard TCP port, you can specify the port number using the colon sign: `192.0.2.3:144` - IMAP server at the 192.0.2.3 address, port 144.

oldName, *oldPassword*

Strings to use when logging into the source IMAP server.

newServer

The IP address of the new (target) IMAP4 server; if that IMAP server operates on a non-standard TCP port, you can specify the port number using the colon sign: `192.0.2.5:145` - IMAP server at the 192.0.2.5 address, port 145.

newName, *newPassword*

Strings to use when logging into the target IMAP server.

Flags is zero, one or several optional parameters:

`--verbose`

An optional parameter. When specified, the progress information is sent to the standard output.

`--list search`

An optional parameter. When specified, the following *search* string is used to find all mailboxes in the source account. Some IMAP servers show the entire user directory or even system directory if the default search string `"*"` is used. Consult with your old IMAP server manual to learn the search string to use.

`--target prefix`

An optional parameter. When specified, the following *prefix* string is appended to all mailbox names sent to the target server. For example, if the target server is CommuniGate Pro, you can specify the `postmaster` credentials in the parameters (instead of the `username` credentials), and use the

`--target '~username/'`

parameter to copy mailboxes to the username account. This can be useful when you do not know the username account password.

`--timeout`

An optional parameter. When specified, the program increases the IMAP operation time-out value from 20 seconds to 1 hour. You may want to specify this option when your copy mail from slow servers.

`--delete`

An optional parameter. When specified, the program deletes the retrieved messages from the source account.

`--nosubscription`

An optional parameter. When specified, the program does not copy the mailbox subscription list to the target account.

`--subscribed`

An optional parameter. When specified, the program copies only those mailboxes that are listed in the source account mailbox subscription list.

Sample:

```
MoveIMAPMail --list "Mail/*" 192.0.0.4 john "jps#dhj" 192.0.1.5  
johnNew dummy
```

Note: if a mailbox name in the source account ends with symbols `.mbox` or `.mdir`, the mailbox name in the target account will end with the `-mbox` or `-mdir` symbols.

Copying All Mailboxes from Other Servers

After you have created the accounts on your new CommuniGate Pro Server, you may want to copy mail from all mailboxes on the old server to the accounts on the new server.

The CommuniGate Pro software package includes the `MoveAccounts` program. This program uses a tab-delimited text file that contains account names and passwords. It can be the same file you have used to [Import Accounts](#) to a CommuniGate domain.

The program scans the file and uses either the [MovePOPMail](#) or the [MoveIMAPMail](#) program to move messages for each account. These programs should be located in your current directory.

```
MoveAccounts [--POP | --IMAP] file sourceServer targetServer  
[suppl_parameters]
```

--POP, --IMAP

Parameters that specify whether MovePOPMail or MoveIMAPMail program should be used.

If none is specified, the MovePOPMail program is used.

file

The name of a tab-delimited file that contains account names and passwords.

sourceServer

The IP address of the old (source) server (POP or IMAP); can include the port number.

targetServer

The IP address of the new (target) server (SMTP or IMAP); can include the port number.

suppl_parameters

Optional parameters (such as --verbose, --delete, --notimeout, --list *search*, etc.) passed to the MovePOPMail or MoveIMAPMail program.

The first line of the *file* should contain the data field names. The fields with names Name and Password must be present.

If the field NewName exists, it is used as the *SMTPRecipient* parameter when the MovePOPMail program is started, or as the *newName* parameter for the MoveIMAPMail program. Otherwise the same Name field data is used.

If the --IMAP parameter is specified, the program checks if the NewPassword field exists. If it does, the data in that field is passed as the *newPassword* parameter to the MoveIMAPMail program. Otherwise, the same Password field data is used.

All fields with other names are ignored.

Sample:

AccountList file

Name	Limit	Password
john	10K	j27ss#45
jim	120K	dud-ee
george	31M	mia#hj!

```
MoveAccounts --POP AccountList 192.0.0.3 192.0.1.5
```

If you cannot obtain the clear-text passwords for all accounts you want to copy, and the old server is using the Unix /etc/passwd or /etc/shadow file, follow these steps:

- Find the OS file that contains the encrypted user passwords: `/etc/passwd`, `/etc/shadow`, or `/etc/master.passwd` (see your OS documentation). We will refer to that file as `/etc/shadow`.
- Create a backup copy of the `/etc/shadow` file.
- Find the `/etc/shadow` record that contains information for your own account or any other account you know the clear-text password for. Let us say that this known unencrypted OS account password is `mypassword`, and encrypted password you found in that account record is `/etc/shadow` record is `FU3jjF/gkJJdA`.
- Edit the `/etc/shadow` file so all account records will contain `FU3jjF/gkJJdA` in their encrypted password fields.
- Open the CommuniGate Pro Default Account Settings page for the domain you are migrating. Enable the `Use OS Passwords` option and check that the `OS UserName` option is set to `*`.
- Create the `AccountList` file that contains the account names in the `Name` field, and the string `mypassword` in the `Password` field.
- Copy all account mailboxes from the old server to the new server using the `MoveAccounts` command. The command will successfully log into both servers, since all accounts on both servers accept the string `mypassword` as the account password.
- Restore the `/etc/shadow` from the backup copy.
- Disable the `Use OS Password` setting in the CommuniGate Pro Default Account Settings, if you do not want to use OS Passwords for your CommuniGate Pro Accounts.

Migrating from an Arbitrary Server ("on-the-fly" migration)

Use this alternative migration method when the password switching method explained above cannot be used, and the plain-text passwords cannot be retrieved from the old server.

The method is based on the [External Authentication](#) feature of the CommuniGate Pro server.

When CommuniGate Pro Accounts are created for the old server users, the CommuniGate Pro password should not be specified at all (they can be left empty). Before creating those accounts, disable the `Use CommuniGate Password` option and enable the `Use External Password` option in the Account Template.

Download, install, and tailor the [migration script](#), and specify this script as the CommuniGate Pro External Authentication program.

When a user connects to the server, the user mailer sends the user name and the user password in the plain text form. Because the CommuniGate Pro Account has the Use CommuniGate Password option disabled, and the Use External Password option enabled, the External Authentication script is called. The script connects to the old server using the POP or IMAP protocol and checks if it can log into the old server with the provided (user login) credentials.

If the old server accepts the specified (user login) password for this Account name:

- the script uses the CommuniGate Pro [CLI](#):
 - to set the specified (user login) password as the CommuniGate Password for this Account
 - to switch off the Use External Password option for this Account
 - to switch on the Use CommuniGate Password option for this Account
- the script starts the MoveIMAPMail or MovePOPMail programs to copy the account mailboxes from the old server to the CommuniGate Pro server.

After the first successful login, the correct password will be set as the new CommuniGate Password, and all Account mail will be copied from the old server.

After all old server users have successfully connected to the CommuniGate Pro server at least once, all their Accounts will have the correct CommuniGate Passwords set. Then you can disable the External Authentication script and retire the old server.

Switching Servers

Very often it is essential to switch to the new server without any interruption in the services you provide.

If you install the new CommuniGatePro server on the same system as the old mail server, you should:

- switch CommuniGate Pro [SMTP receiving](#) to port 26, so it will not interfere with your old server smtp activity.
- switch CommuniGate Pro [POP service](#) to port 111, and [IMAP service](#) to port 143, so they will not interfere with your old server pop/imap activity.
- Configure CommuniGate Pro and create [domain aliases](#) and full featured Secondary [Domains](#).
- Create some test accounts in the main and secondary domains and check that you can log into those accounts using [WebUser Interface](#).

- Check that you can send mail from those accounts using the [WebUser Interface](#): mail to other test accounts in the created domains and mail to other servers should be delivered correctly.
- If you have a POP or IMAP client that allows you to specify a non-standard port number, check that you can log into the test accounts on the POP port 111 and IMAP port 143.
- Create tab-delimited file(s) with names, passwords, and other attributes of your existing accounts and use the [Account Import](#) feature to create all accounts on your new server.

All this can be done while your old server is still active.

Now, you should stop your old server activity by either changing its port numbers to non-standard values, or by disconnecting it from the external network.

Use the [AccountMove](#) program to copy all messages from your old server to CommuniGate Pro.

When all messages are copied, change the CommuniGate Pro SMTP port number back to 25, POP port number - to 110, and IMAP port number to 143. Now CommuniGate Pro will operate as your mail server, without any interruption in the services you provide.

You may want to keep the old server running for several hours - in case its queue contained some delayed outgoing messages. Just check that the old server does not try to use the standard ports.

Moving to Secondary Domains

If you create several [Secondary domains](#) in CommuniGate Pro, you may want to move accounts from your old server(s) to a Secondary CommuniGate Pro domain, not to its Main Domain.

In this case, you should add the NewName field to your account list file, and copy all names into that column, adding the @domainname string to each name.

If you use the IMAP protocol to move messages between the servers, you may use a simpler method:

- If the target domain has an IP address assigned to that domain, use that address in the

mail copying programs: all non-qualified names provided on connections established with that address are interpreted as names in that domain. See the [Access](#) section for the details.

- If the target domain does not have an assigned IP address, temporarily assign the IP address of the main domain to that secondary domain. Move the messages, and remove the IP address from the list of addresses assigned to that domain.



System Administration

When the CommuniGate Pro Server is up and running, it can be configured, monitored, and set up using any Web browser.

By default, the [HTTP module](#) provides access to the CommuniGate Pro administration pages (WebAdmin Interface) via the TCP port number 8010. To connect to the server, the administrator should type `http://serveraddress:8010` where serveraddress is either server IP address or the server domain name (A-record).

Note: If you use a Netscape® browser, check that its caching setting (Preferences->Advanced->Cache) is set to Every time.

Sections and Privileges

The Server administration pages are divided into four groups (realms). To access a page in any group, a user should be registered with the CommuniGate Pro Server (should have an Account on the Server), and the user should be explicitly granted access rights to that section.

- The Settings section contains pages that allow a system administrator to modify the Server kernel and module settings.
- The Accounts and Domains section contains pages that allow a system administrator to create and remove secondary domains and accounts, and to modify the domain and account settings.
- The Directory section contains pages that allow a system administrator to configure the Directory services of the CommuniGate Pro server.
- The Monitors section contains pages that allow a system administrator to monitor server and module queues, communication channels and their states, and to browse the Server Logs. If a user is granted an access right to the Monitors section, additional Monitor Access rights can be granted, too (rights to release and reject module queues, reconfigure the Log Manager, etc.)
- The Master section contains the pages that allow a system administrator to grant and revoke access rights, and to modify the Server License Keys.

Note: If a user is granted the Master access right, that user can access all other sections.

Note: These access rights can be granted to the accounts (users) in the main domain only. Accounts in secondary domains can be granted [domain administration](#) rights only.

When a Server is installed for the first time, it creates the `postmaster` account in the main domain, assigns a random password to that account, and grants the Master access right to the `postmaster` user.

Base Directory Structure

All CommuniGate Pro Server files - accounts, domains, mailboxes, settings, queues, etc. are stored in one place - in the Server *base directory*.

When the Server starts, it creates the following objects inside its base directory:

- The `Settings` directory. This directory contains files with module and kernel component settings.
- The `Queue` directory. This directory contains [Temporary and Message files](#). The Message files contain messages submitted to the Server, but still undelivered to all their recipients.
- `BadFiles` directory. This directory contains Message files the [Enqueuer kernel component](#) failed to parse. This directory should be empty.
- `Accounts` directory. This directory contains [account files](#) for the Main Server Domain.
- `Domains` directory. This directory contains directories for all [Secondary Domains](#).
- `Submitted` directory. This drop-in directory is used to submit messages to Server via the [PIPE module](#).
- `SystemLogs` directory. This directory contains [Server Logs](#).
- `ProcessID` file. This file exists only when the Server is running and contains the numeric identifier of the Server process.
- `Directory` file. This file contains or describes the Server [Central Directory](#).

For more information about the Account and Domain files and directories, see the [Account Data](#) section.

You can use symbolic links to move some of these directories to other locations (and other disks).

General Settings

Start configuring the Server by opening the General page in the Settings section.

Main Domain Name:	
System Internals Log:	
Crash Recovery:	
Server Time:	21:08:46 -0800
Server OS:	Sun Solaris
Server Hardware:	Intel
Server Version:	3.5
Server IP Address(es):	[216.200.213.118],[216.200.213.119],[10.0.0.5]
Name Server(s) IP Address(es):	[216.200.213.113],[216.200.213.114]

Main Domain Name

In this field you should enter the name that the CommuniGate Pro Server will interpret as its own *Main Domain Name*. All mail addressed to that domain will be treated as local, and (in the simplest case) that mail will be stored in local account mailboxes. Initially, this field contains the server computer name that CommuniGate Pro retrieves from the OS. If this name looks like `host12345hh.company.com`, you should change it to the name of the domain this Server should process.

Note: unless you create additional [Domains](#) ONLY the messages directed to addresses in the Main Domain will be processed as *local*. If the Main Domain Name is entered as `company.com`, then messages to `mail.company.com` will not be processed as local, and if such a message is received, the server will try to deliver it to the `mail.company.com` system over the network. If the DNS record for the `mail.company.com` points to the same Server computer, the *mail loop* error will be detected, and the message will be rejected.

If your server should process mail for several domains, enter the additional domain names as Main Domain Aliases (if those domain names should be [mapped](#) to the Main Domain), or create additional [Secondary Domains](#).

Sample configuration:

A server should process mail for the `company.com` and `client1.com` domains. In the DNS system, these domain names have only MX-records pointing to `mail.company.com` and `mail.client1.com` A-records, and these A-records point to IP address(es) belonging to the CommuniGate Pro Server system.

- set `company.com` as the Main Domain Name.
- open the Domains page, find the **company.com** record and click on its `Settings` link to open the `company.com` Domain Settings page. Scroll it down to find the Aliases fields.
- enter `mail.company.com` into an empty Aliases field, and click the Update button.
- open the Domains page. Enter `client1.com` into the text field and click the Create Domain button.
- the `client1.com` record should appear in the list; click its `Settings` link to open the `client1.com` Domain Settings page. Scroll it down to find the Aliases fields.
- enter `mail.client1.com` into an empty Aliases field, and click the Update button.

System Internals Log

Use this setting to specify what kind of information the server kernel module should put in the [Server Log](#). Usually you should use the `Major` (message transfer reports) level. But when you experience problems with the server kernel, you may want to set the `Log Level` setting to `Low-Level` or `All Info`: in this case low-level details will be recorded in the System Log as well. When the problem is solved, set the `Log Level` setting to its regular value, otherwise your System Log files will grow in size very quickly.

The kernel records in the System Log are marked with the `SYSTEM` tag.

Kernel problems are very unlikely to happen. If you see any problem with the Server, try to detect which component is causing it, and change the Log setting of that component (Router, SMTP, POP, etc.) to get more information.

Crash Recovery

If this option is enabled, the CommuniGate Pro Server uses special recovery techniques to proceed after various failures (including the crashing bugs in the Server software itself).

If you see "exception raised" messages in your CommuniGate Pro Log and/or in the OS `system.log` or `mail.log`, you may want to disable this option and force the Server to stop when an exception is raised again, and to produce a *core dump* file.

Core dump files can be uploaded to the Stalker ftp site for examination.

Stalker Software recommends you to disable this option if you are running any beta-version of the CommuniGate Pro software.

Information fields

Information fields on the General Settings page display the name of the Server Operating System,

the hardware platform, the version of the CommuniGate Pro Server, the Server network address(es), the Server Local Time and Time Zone. This information is useful for system administrators that have to examine Logs from remote locations, as all time stamps in the System Logs are specified in the Server local time.

Refresh

This button can be used after the Server OS local IP Addresses have been changed or the DNS settings for CommuniGate Pro Domains have been modified. When you click this button:

- the Server re-reads the list of Local IP Addresses from the OS;
- the Server re-reads the [Domain Name Server addresses](#) from the OS settings.
- the Server updates the "Assigned IP Addresses" for all Server Domains. If some domains have IP Addresses specified "Using DNS A/MX Records", the new addresses are retrieved from the DNS system;

Drop Root

This button is available on certain Unix platforms. It allows the System Administrator to tell the server to drop the "superuser" privileges. Certain functions (such as OS Authentication, Execute Rules operations, etc.) may become unavailable.

If the Server succeeds to drop the "superuser" privileges, the button title changes to Restore Root. Click the Restore Root button to restore the "superuser" privileges.

Command Line Options

The CommuniGate Pro Server supports the following command-line options (parameters):

`--CGateBase directory`

or

`--Base directory`

The next parameter string specifies the location of the CommuniGate Pro [base directory](#).

`--LogToConsole`

This option tells the Server to duplicate all its [System Log](#) records to the stdout (standard output). This option can be used for troubleshooting when the Web interface to System Logs is not available.

`--LogAll`

This option tells the Server to ignore all current Log Level settings and record all possible Log records.

`--NoWebCache`

This option tells the server not to cache the Web Interface files internally. Use this option when you modify the Web Interface files and you need to see the results without

restarting the server.

--Daemon

This option can be specified on Unix platforms only. It tells the server to fork and operate in the background, with stdin, stdout, and stderr redirected to /dev/null.

--CGateApplication *directory*

The next parameter string specifies the location of the CommuniGate Pro [application directory](#). You can use this option when the application itself cannot properly detect its own location, or if the CommuniGate Pro Server application file is not placed in the same location as other application directory files and subdirectories. For example, on OS/400 CommuniGate Pro Server is located in an OS/400 library, and this parameter is used to tell the server where the Unix-style directory with WebUser, WebAdmin, WebGuide, and other files is located.

--noLockFile

This option tells the Server not to create the ProcessID lock file. This option can be used if the file system hosting the *base directory* does not support file locks.

--dropRoot

This option can be specified on Unix platforms only (this does not include Linux). It tells the Server to drop the *root privilege* permanently. The server drops the privilege approximately 60 seconds after the end of its kernel initialization process, so all listenening sockets can be opened when the server is still running as the *root*. The root privilege cannot be restored later. See the [Server Root Privilege](#) section for more details.

--ThreadsScope

This option can be specified on platforms supporting p-threads (OS/400 and most Unix flavors). The next parameter string can be either "system" or "process". See your OS manual to learn how these "scheduling scopes" work. If this option is not specified, the default OS scheduling mode is used.

--BatchLogon

This option can be specified on Microsoft Windows NT/2000/XP platforms only. The option tells the Server to use 'batch logon' instead of the 'network logon' when an account password is verified using the Windows OS password system.

--SharedFiles

This option can be specified on Microsoft Windows platforms only. The option tells the Server to open all files with the FILE_SHARE_READ sharing attribute making it possible for other programs (such as backup daemons) read the CommuniGate Pro *base directory* files when the server is running. Command line option names are case-insensitive.

Specifying Command Line Options under Windows NT/2000/XP

You can specify the Command Line Options using the Services control panel "Startup Parameters" field. A non-empty set of Command Line Options is stored in the System Registry and it is used every time the CommuniGate Pro Messaging Server service is started without parameters. To clear the stored set of the Command Line Options, specify a single "-" sign using the Services control panel "Startup Parameters" field.

Customizing Unix Startup Scripts

You may need to add certain shell commands to the CommuniGate Startup script. Since the Startup script is a part of CommuniGate Pro application software, it is overwritten every time you upgrade your CommuniGate Pro system. Instead of modifying the Startup script itself, you can place a `Startup.sh` file into the CommuniGate Pro *base directory*. Startup scripts check if that file exists, and execute it before performing the requested start/stop operations.

Shutting Down

The CommuniGate Pro Server can be shut down by sending it a SIGTERM or a SIGINT signal.

On Unix platforms, you can use the startup script with the `stop` parameter, or you can get the Server process id from the `ProcessID` file in the base directory and use the `kill` command to stop the server.

On the Windows NT platform, you can use the Services control panel to stop and start the CommuniGate Pro server.

You can also use the `shutdown` [CLI API](#) command to stop the server.

When the Server receives a shutdown request, it closes all the connections, commits or rolls back mailbox modifications, and performs other shutdown tasks. Usually these tasks take 1-3 seconds, but sometimes (depending on the OS network subsystem) they can take more time. Always allow the server to shut down completely, and do not interrupt the shutdown process.

OS syslog

The CommuniGate Pro server can store as much as several megabytes of Log data per minute (depending on the Log Level settings of its modules and components), and it can search and

selectively retrieve records from the log. To provide the required speed and functionality, the Server maintains its own multithreaded [Log system](#).

The Server places records into the OS log (system.log or mail.log):

- when it starts up;
- when it shuts down;
- when it detects its own memory leaks;
- when it detects its own program error;
- when a program error exception (signal) is raised.

Server Root Privilege

The CommuniGate Pro is designed as a highly secure application. In order to perform certain operations, the Server runs as *root* on Unix platforms, and it carefully checks that no user can access restricted OS resources via the Server. Since many other servers do not provide the same level of security, system administrators preferred to run servers in a non-root mode, so a hole in the server security would not allow an intruder to access the restricted OS resources.

CommuniGate Pro can "drop" the *root privilege*. The privilege can be dropped in the "permanent" or "reversible" mode. When asked to drop the root (uid=0) privilege, the Server changes its UID:

- to the UID of the Unix user `cgatepro` (if exists), otherwise
- to the UID of the Unix user `nobody` (if exists), otherwise
- to the UID 1

When the root privilege is dropped, the following restrictions apply:

- No [Listener](#) port with number < 1024 can be opened. If you try to add a listener with the port number *n* (*n* < 1024), the port with the number 8000+*n* is opened instead.
- Remote applications started via Account-Level [Rule](#) EXECUTE command run in the current CommuniGate Pro Server environment (the effective UID and other OS-level process parameters are not changed).
- OS Passwords cannot be used.

If the root privilege was dropped in the "reversible" mode, the root privilege can be restored. For example, if you need to open a listener on the port 576, but the Server root privilege has been dropped, you should restore the root privilege first, then open the listener port, and then you can

drop the Root privilege again.

To drop the root privilege permanently, use a special [Command Line Option](#).

To drop the root privilege in the "reversible" mode, click the "Drop Root" button on the General page. The button should change to the "Restore Root" button - you can use it to restore the Server root privilege. This option is not available on those platforms that cannot drop the root privilege correctly (Linux).

Domain Administration

If your Server has several [Secondary Domains](#), you may want to grant some user(s) in that domain the [domain administrator access right](#).

A domain administrator can use the WebAdmin interface to access the pages in the Accounts section, but the access is limited to that domain only, and not all domain and account Settings can be modified.

When you grant the *domain administrator* access right to a user, you will see a list of specific access rights - the internal names of Domain and Account Settings. You should specify which settings the domain administrator can modify. Also, the list of enabling options allows you to grant the domain administrator rights:

- to create, rename, and remove [Accounts](#)
- to create, rename, and remove account aliases
- to create, rename, and remove [Groups](#) and [Forwarders](#)
- to create, rename, and remove [mailing lists](#)
- to modify the [WebUser Interface](#) files (HTML templates and graphic images)
- to send Alert messages to domain users
- to [administer other domains](#)
- to have full direct access to all [Mailboxes](#) in all domain Accounts.
- to have full direct access to all [Personal Web Sites](#) in all domain Accounts.

The domain administrator access right can be granted to users in *secondary domains* by a system administrator that has the Accounts (All Domains and Account Settings) access right.

A Domain administrator can control the domain using the same WebAdmin port (see [HTTP module](#) description for the details), or using the [Command Line Interface](#) commands.

Domains Administrators in other Domains

When a customer has several Domains, you may want to let an Account in one Domain administer other Domains. You should grant such an Account the CanAdminSubDomains Domain Admin right. Then you should open the [Domain Settings](#) page for the target Domain and specify the administrator Domain name in the Administrator Domain Name field.

Sample.

A customer has the company1.com, company2.com, company3.com Domains on your Server. You may want to specify "company1.com" as the Administrator Domain Name in the company2.com and company3.com Domain Settings. Now, any Account in the company1.com Domain that has the CanAdminSubDomains Domain Administrator right can administer all three Domains.

Note: when a Domain Administrator connects to the Domain WebAdmin Interface, the browser displays the Login Dialog Box. If the Administrator Account is in a different Domain, the full account name (*accountName@domainName*) should be specified.

Customizing Domain WebAdmin Interface

The Server Administrator can modify the look and feel of the Domain WebAdmin interface. For each CommuniGate Pro domain, a custom version of WebAdmin files can be created and uploaded to the domain storage.

To modify the Domain WebAdmin interface pages, connect to the server WebAdmin Interface as a Server Administrator, open the Domain Settings page and click the WebAdmin link. The list of WebAdmin files will appear. Click the Accounts link to open the subdirectory containing the files used to compose WebAdmin pages in the "Account" realm:

(Accounts subdirectory) [UP](#)

Marker	File Name	Size	Modified
default	AccountDefaults.html	1929	15-Feb-00
default	AccountList.html	2K	15-Feb-00
	AccountMain.html	4K	27-Feb-00
default	AccountRemove.html	489	15-Feb-00
default	AccountRights.html	2K	15-Feb-00
default	AccountSettings.html	2K	15-Feb-00
.....			
default	WebUserSettings.html	1194	15-Feb-00
default	WebUserSettingsMain.html	3K	15-Feb-00
Totals:	32	66K	

If the file exists in the Domain WebAdmin storage, its name is marked with a check box in the Marker field. You can select the check box and click the Delete Marked button to remove the custom file(s) and make the Server use the default WebAdmin files.

The Server Administrator can also upload custom files to the "default" WebAdmin storage. Those files will be used in all Domain WebAdmin Interfaces unless a Domain has the same file explicitly uploaded into its WebAdmin Interface storage.

To upload the "default" WebAdmin files, use the Server WebAdmin Interface as a Server Administrator, and open the WebAdmin link on the Domains page. If your server is a member of a [Cluster](#), an additional panel appears. This panel allows you to upload files either as the default Domain WebAdmin files for all non-shared (this-server-only), or for all shared (cluster-wide) Domains.

If the file does not exist in the Domain WebAdmin storage, the default file (server-wide or cluster-wide, depending on the Domain type) is used. If this file does not exist, the file from the *application directory* WebAdmin subdirectory is used.

To modify some element of the WebAdmin Interface:

- use the WebAdmin Interface Editor to open the directory that contains the file you want to modify;
- use the file link to open the file and/or to copy the file on your local disk;
- modify the file using any HTML editor program;
- on the same WebAdmin Interface Editor directory page, click the Browser button and select the modified file on your local disk;
- click the Upload File button to upload the modified file to the CommuniGate Pro WebAdmin directory opened in the WebAdmin Interface Editor.

If the WebAdmin directory/subdirectory did not contain a custom copy of the uploaded file, you will see the `default` file marker changing to a checkbox. If a custom version of that file already existed in the WebAdmin directory/subdirectory, the old version is replaced with the uploaded one.

To remove a custom version of a WebAdmin Interface file, select the checkbox on the left of that file name and click the Delete Marked button. If the file with that name exists in the *application directory* WebAdmin subdirectory, the file name does not disappear from the WebAdmin Interface Editor page, but the name gets the `default` marker indicating that the default (original) version of the file will be used again.

Note: The Server WebAdmin interface **always** uses the files located in the WebAdmin subdirectory of the *application directory*. If you modify the WebAdmin interface for the main domain, the modified pages will be used when a Domain Administrator of the main domain uses the WebAdmin Interface. The Server Administrator will see the framed version of the WebAdmin Interface (with the Settings, Domains, Directory, and Monitors realms) and the default WebAdmin files will be used to compose the Server WebAdmin Interface pages.

Customizing Server Prompts

The Server Administrator can modify the protocol prompts and other text strings the CommuniGate Pro Server sends to client mailers.

To modify the Server Strings, the administrator should follow the [Strings](#) link on the [General](#) Settings page. The Server Strings page appears (the actual page has much more strings):

Keyword	String
POPPrompt	CommuniGate Pro POP3 Server ^0 ready
SMTPByeBye	CommuniGate Pro SMTP closing connection
SMTPNoRelay	we do not relay
SMTPNonInternet	will leave the Internet
SMTPNormalPrompt	^1 ESMTP CommuniGate Pro ^0

To modify a Server String, enter the new text in the text field, and select the upper radio button. To change the string to its default value (displayed under the text field), simply select the lower radio button.

Click the Update button to update the Server Strings.

Domain Name Resolver (DNR)

The CommuniGate Pro server uses its own high-speed multithreaded Domain Name Resolver to convert domain names into network (IP) addresses. To convert names, the Domain Name Resolver sends requests to the specified Domain Name Servers.

Server Administrators with the Can Modify Settings access right can modify the Resolver settings. Open the Obscure page in the Settings section of the Server WebAdmin Interface:

Domain Name Resolver	
Log:	Concurrent Requests:
Initial Time-out:	Retry Limit:
DNS Addresses:	[209.1.58.247], [206.40.74.1]

Log

Use this setting to specify what kind of information the Domain Name Resolver should put in the Server Log. Usually you should use the Major or Problems levels. In the later case you will see the information about all failed DNS lookups. If you use the RBL services, you may see a lot of failed lookups in the Log. When you experience problems with the Domain Name Resolver, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well.

The Resolver records in the System Log are marked with the DNR tag.

Concurrent Requests

This setting limits the number of concurrent requests the Resolver can send to Domain Name Servers. On a heavily-loaded mail relay processing several hundred requests per second, this parameter should be selected after some testing: older DNS servers may crash if requested to process too many concurrent requests, also in certain cases the DNR traffic may start to compete with the mail transfer (SMTP) traffic.

Initial Time-out

Then Domain Name System uses a connectionless UDP protocol, and if there any network trouble,

a UDP request or response can be lost (TCP protocol automatically resends lost packets). The Domain Name Resolver waits for a response from a DNS server for the period of time specified with this option.

If a response is not received, the Resolver resends the request, and waits twice longer, if it times out again, it can resend the request again and wait three times longer.

If you have several Domain Name Servers specified, each time the resolver needs to repeat a request, it sends it to the next DNS server in the list.

Retry Limit

This option specifies how many times the Resolver should re-send the same request if it has not received any response from a DNS server.

Note: when a request is an [RBL](#) request, the Resolver sends the same request not more than twice, and both times it uses the same (Initial) response time-out.

DNS Addresses

This setting specifies how the CommuniGate Pro Server selects the DNS servers to use. If the OS-specified option is selected, the Server reads the DNS server addresses from the OS. To force the server to re-read those addresses, click the Refresh button on the General page in the Settings section.

If the Custom option is selected, the CommuniGate Pro server will use the DNS servers addresses listed in the text field next to this pop-up menu.

If no DNS server address is specified, the CommuniGate Pro server uses the 127.0.0.1 address, trying to connect to a DNS server that can be running on the same computer as the CommuniGate Pro server.

The Domain Name Resolver uses TCP connections if the server UDP response came back with the "Truncated" flag set. This feature allows the Resolver to retrieve very large records from DNS servers.



CommuniGate Pro HTTP Module

The CommuniGate Pro HTTP module implements the Hypertext Transfer Protocol via TCP/IP networks.

The CommuniGate Pro Server uses the HTTP Admin module:

- to provide access to the [Server WebAdmin](#) (Administration) Interface pages.
- to provide access to [Domain WebAdmin](#) (Administration) Interface pages.

The CommuniGate Pro Server uses the HTTP User module:

- to implement the [WebUser Interface](#) to [user Accounts](#) and [Mailing List](#) archives.
- to provide access to [Personal Web Sites](#).

Access to the WebAdmin Interface Pages

The Server Administrator can use any Web browser application to configure and to monitor the Server remotely, using the Web (HTML) *forms*.

The authentication schemes supported with the HTTP protocol protect the WebAdmin pages from an unauthorized access. In order to access the WebAdmin pages, the user should provide the name and the password of a CommuniGatePro account with required [Server Access Rights](#).

By default, the HTTP module accepts *clear text* TCP/IP WebAdmin connections on the port 8010 and *secure* (SSL/TLS) connections on the TCP port 9010.

To access the WebAdmin pages, the Server administrator should use the following URLs:
`http://domain.com:8010`

`https://domain.com:9010`

where *domain.com* is the name of the main server domain or its alias, or the IP address of the CommuniGate Pro Server.

Access to the Domain WebAdmin Interface Pages

If the CommuniGate Pro Server supports several [Secondary Domains](#), the same port can be used by the [Domain Administrators](#) to access the Secondary Domains settings and account lists.

A domain administrator should access the server using the following URL:

`http://sub.domain.com:8010`

`https://sub.domain.com:9010`

where *sub.domain.com* is the name of the secondary domain to administer.

The server will ask for the user name and a password, and if the specified account has the [Domain Administrator access right](#), the list of the domain accounts is displayed.

Sometimes this URL cannot be used. For example, a secondary Domain may have no DNS A-records (only MX records). To access such a domain, its domain administrator should use the following URL:

`http://domain.com:8010/Admin/sub.domain.com/`

where:

domain.com is the name of the main server domain or its alias, or the IP address of the CommuniGate Pro Server.

sub.domain.com is the name of the domain to access.

Server configuration errors can cut you off the Server WebAdmin Interface, if all your server IP addresses and DNS names are assigned to secondary Domains. To access the main Server WebAdmin Interface, use the following URLs:

`http://sub.domain.com:8010/MainAdmin/`

`https://sub.domain.com:9010/MainAdmin/`


where *sub.domain.com* is any name pointing to your server computer or any of its IP addresses.

Other Domains can specify your Domain as their [Administrtor Domain](#). Your Domain WebAdmin Login page provides a list of those Domains, so you can open their WebAdmin Interfaces.

Remember that you should login using your full Account name (*yourAccountName@yourDomainName*) when accessing other Domain WebAdmin pages.

WebAdmin Settings (Preferences)

Server and Domain administrators can customize the WebAdmin Interface parameters, including the initial number of Accounts to be displayed in the Account Lists, the refresh rate for the Monitor pages, etc.

Each CommuniGate Pro WebAdmin *realm* has its own WebAdmin Preferences page. Click the  icon on any of the WebAdmin pages to open the Preferences page.

The specified Settings (Preferences) are stored as one of the Administrator Account Setting attributes, so different administrators can have different Settings (Preferences).

Access to WebUser Interface

CommuniGatePro users can connect to the CommuniGate Pro Server with any Web browser (via the HTTP protocol) to manage their accounts, to browse their mailboxes, to read, copy, delete, forward, and redirect messages, to move messages between mailboxes, to compose and submit new messages, etc. This CommuniGate Pro component is called the [WebUser Interface](#).

Registered users and guests can also use this component to browse [mailing list archives](#).

By default, the HTTP module accepts *clear text* WebUser TCP/IP connections on the TCP port 8100, and the secure connections - on the TCP port 9100. If your Server does not have to coexist with some other Web Server on the same computer, it's recommended to change these port numbers to 80 and 443 - the standard HTTP and HTTPS port numbers.

In this case your users will not have to specify the port number in their browsers.

Access to Personal Web Sites

CommuniGatePro users can have their personal Web Sites. See the [Web Site](#) section for the details.

The URL for the account@domain personal Web site is:
`http://domain:port/~account`
where port is the WebUser port (8100 by default).

The list of files on that personal Web Site can be seen at:
`http://domain:port/~account/index.html`

Configuring the HTTP module

Use any Web Browser to connect to the Administration Port on your Server, and open the Access page in the Settings section.

Log:		
User Port:	Channels:	listener
Admin Port:	Channels:	listener

Log Level

Use this setting to specify what kind of information the HTTP module should put in the Server Log. Usually you should use the Major or Problems (non-fatal errors) levels. But when you experience problems with the HTTP module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well.

The HTTP Admin module records in the System Log are marked with the HTTPPA tag.

The HTTP User module records in the System Log are marked with the HTTPU tag.

Channels

This setting is used to limit the number of simultaneous TCP/IP connections the HTTP module can accept. Most browsers open several connections to load an HTML page and all embedded pictures, so do not set this limit to less than 5, otherwise some browsers may fail to download embedded graphic objects.

[listener](#)

Follow this link to open the HTTP Admin Port or HTTP User Port [listener](#) settings. There you can specify the TCP port number(s) the service should use, the interfaces to use, and other options.

If the CommuniGate Server computer runs some other Web Server application, you should specify a port number in the "secondary range" to avoid conflicts with that other Web Server application. Usually the "secondary" Web Servers use ports numbers in the 8000-8100 range. If you set the port number to 8010, you will be able to connect to your server by entering `http://xxx.yyy.zzz:8010` in your Web

browser, where *xxx.yyy.zzz* is the exact domain name (A-record) or the IP address of your server.

Routing

As any [Access](#) module, the HTTP module uses the [Router](#) to process all address. But unlike other modules, the HTTP module often deals not with the complete E-mail addresses, but with domain names only.

When a request is received on the WebAdmin port, the HTTP module should use the domain name or the IP address specified in the URL to decide which Domain Administration pages to display.

When a request is received on the WebUser port, the HTTP module should use the name specified in the URL to decide which Domain (its login page, mailing lists, personal Web Sites, etc.) to access.

In order to support all types of CommuniGate Pro Routing features (Router Table, Domain Aliases, IP Address to Domain Mapping, etc.), the HTTP module composes a complete E-mail address `LoginPage@domainname` (where *domainname* is the domain name specified in the request URL), and processes this address with the Router. If the addresses is routed to the LOCAL module, and the routed username is still LoginPage, then the domain part of the resulting address is used to open the proper CommuniGate Pro domain (the main domain or a secondary one).

If the address is routed to the LOCAL module, but the resulting username is not LoginPage, the [Personal Web Site](#) of the addressed Account is opened.



Server Logs

All components of the Server store messages in one System Log. Each record contains a time stamp, the log level, the tag identifying the component that created the record, and the record data itself.

System Logs are plain text files, and can be processed with any text-processing utility.

When sending a technical support request to [Stalker Technical Support](#), always include a portion of the Log that indicates the problem.

Creating and Deleting Log Files

You can use any Web browser to examine the Logs. Click the Logs button in the Monitor section and the list of the stored Logs appears. The current Log is marked with the asterisk (*) sign.

You should have the "Can Monitor" [Server Access Right](#) to view the Logs.

The options on the top of the page allow you to specify when the Logs files are created and deleted:

Log Engine Options	
Start New File every:	Delete Old Files in:
or if Larger than:	

A new file is created automatically every day (at midnight), or more often, as you specify with the first option. Additionally, a new Log file is created if the current Log file size exceeds the specified limit. Shortly after a new Log file is created, the Server checks all files in the SystemLogs directory, and removes all files that are older than specified in the third option.

You should have the "CanTuneLoggerSettings" [Monitor Access Right](#) to modify the Logs Engine

settings.

You can select one or several Logs in the list and then remove them using the Delete Marked Logs button. The active (current) Log file cannot be deleted.

You should have the "CanTuneLoggerSettings" [Monitor Access Right](#) to delete Logs.

If there are too many Log files on the Server, you can enter a string in the Filter field and click the Display button: only the Logs with names matching the Filter string will be displayed:

Filter:		11 selected
	Name	Size
*	2001-04-04	115K
	2001-04-03	204K
	2001-04-02	34K
	2001-04-01	34K
	2001-03-31	34K
	2001-03-30	25K
	2001-03-29	33K
	2001-03-28	35K
	2001-03-27	34K
	2001-03-26	33K
	2001-03-25	33K

Click the Log file name to open the selected Log.

Specifying a Time Interval

When the Log appears in your browser window, all Log records are displayed. Since Logs can have thousands of them, you may want to view only a portion of the Log. Interrupt the Log downloading process and specify the Log Level and the Time Range options:

Selection	
Filter:	
Level:	
Interval:	-

Only the records with time stamps in the specified interval are displayed.

Note: if you are viewing the current Log and specify "*" in the second field, all records placed in the Log by this moment are displayed.

Note: if you are viewing the current Log and specify some future time in the second field, the Server will keep the browser channel open, sending new Log records as they are placed in the Log. The channel is closed either when the specified time comes, or when the Server starts a new Log.

Filtering Log Records

Since the System Logs can be very big (several megabytes of data) on a heavily loaded server or on a server with low-level logging enabled, it is difficult to examine the entire Log.

You can set the `Level` setting to suppress displaying records that are more detailed than the specified value (have a higher level tag), and you can put a filter into the `Filter` field. Click the `Display` button to display only the records that contain the specified substring.

Example:

One of your users complains that sometimes his mailer application cannot retrieve messages from your server properly and it displays an error message about some protocol faults.

Since it does not occur often, you should run the POP module with its Log Level set to Low-Level, and this will make the System Log very big. Finally, the client contacts you and says that the mailer has displayed the same error.

You open the Log and set the Level to 3 (Problems). Now you may see all the problems with the POP module that occurred today. You find the record that indicates the problem your client is talking about, and that record has a tag POP-12357. So, you type POP-12357 into the Filter field, and change the Log Level to 5 (All Info). As a result, you see a clean log of that particular POP session.

Selection	
Filter:	
Level:	
Interval:	-

Searching in System Logs

Use your browser Find command to search for a string in the filtered portion of the System Log.
Use the Print command of your Web browser to print the System Log.



Protection

The Internet is flooded with soliciting E-mail messages distributed to millions of E-mail addresses. These messages are known as "spam".

Spammers fill your user mailboxes with a huge amount of unwanted messages, not only overloading the Internet and your Server resources, but making mail retrieval very slow and difficult for your users.

In order to distribute their messages to thousands and even millions of E-mail addresses, spammers try to use any SMTP mail server on the internet as a relay: they deliver one copy of the message to each mail server, requesting that the server then route it to a hundred addresses. This practice not only overloads your Server resources, but it places you at risk to be recognized as a spammer (since messages come from your server).

The CommuniGate Pro Server has Anti-Spam Options that can help you to deal with "spam".

Prohibiting Unauthorized Relaying

If your SMTP module can accept incoming TCP connections, your server can be used by spammers as a mail relay engine: they can distribute their messages all over the world using your server as an *open relay*.

To protect your site from spammers, you should restrict the Server relaying functionality. Basically, only your own users should be able to use your Server to relay mail to other places on the Internet. Messages coming from other sources should go only to your own Accounts, and should be relayed to other Internet sites only when you explicitly allowed that type of relaying.

Specifying Client IP Addresses

The simplest way to decide if an incoming SMTP message is coming from your own user is to look at the network address it is coming from. If all your users connect from one or several LAN(s), you can treat all messages coming from those networks as "messages from Clients", and your Server will relay them to the Internet.

Use the WebAdmin Interface to open the Protection pages inside the Settings section (realm), and click the Client IP Addresses link.

Enter the IP addresses on your LAN(s), as well as IP addresses of other systems that should be allowed to use your server as a mail relay:

Client IP Addresses

If you are an ISP and your mail server is used as a forwarding mail server for your client systems, enter the IP addresses of your client servers as well.

If you provide dial-up services, enter the IP address ranges you have allocated to your dial-up users.

A comment (separated with the semicolon (;) symbol) can be placed at the end of a line. A line starting with a semicolon symbol is a comment line.

Specifying Client IP Domains

You can specify your Client IP Addresses using the "reverse lookup domain names".

Detect Clients by DNS Name

When a client connects for an IP address not listed in the Client IP Addresses list, and the Detect Clients by DNS Name option is enabled, the server tries to get the domain name for that IP address (if the IP address is *aa.bb.cc.dd*, the Server tries to retrieve the PTR record for the *dd.cc.dd.aa.in-addr.arpa* name). If the PTR domain name is retrieved, it is checked against the strings specified in the table (these strings can include the wildcard (*) symbols). If the retrieved name matches one of the table strings, the server retrieves the DNS A record for the retrieved domain name, and checks that the IP address is included into the IP addresses in that record. If it is included, the address is considered to be a "Client IP Address", and it is processed in the same way as if it was entered into the Client IP Addresses list.

Note: while this method was popular with legacy mail servers, it can be very expensive for large-scale systems. It requires the server to make 2 DNS transactions for each incoming connection, and these transactions can take a lot of time. Use this method only when absolutely necessary, for example when your server needs to support a large (and unknown) set of campus networks, and the only thing known about

those networks is the fact that all their IP addresses can be "reversed-resolved" into some subdomain of the school domain. Even in this case, try to enter all known addresses and networks into the Client IP Addresses list, decreasing the number of required "reverse-resolving" operations.

Configuring the SMTP module

When a message is received with the [SMTP module](#), and the sender IP address is not found in the Client Addresses list, the message is marked as being received "from a stranger". If this message should be relayed by your server to some other host on the Internet, and that host is not listed in the Client IP Addresses list either, the message can be rejected.

As a result, servers and workstations included into the Client Addresses list can use your Server to send (relay) messages to any mail server on the Internet. But any message coming from an unlisted address and directed to some other unlisted system can be rejected. This will prohibit spammers from using your Server as an "open mail relay".

Since this functionality can affect your legitimate users if you do not specify their IP addresses correctly, the Relay to non-Clients option is available on the [SMTP Settings](#) page. Set that option to "if received from Clients", and "stranger-to-stranger" relay attempts will be rejected.

The Client IP Addresses list can include addresses of some other mail servers. The Server can relay mail sent by anybody and addressed to a server included into the Client IP Addresses list, but it can also check if the message address is a "simple" address.

On the [SMTP Settings](#) page you can set the Relay to Clients option. Set this option to "to simple addresses" to avoid auto-relaying of "complex addresses" (such as `username%somehost@otherserver`) to servers listed in the Client IP Addresses list. This setting will prevent spammers from using your servers for "two-server relays".

When the "two-server relay" method is used:

- a spammer sends a message with the `username%somehost@server2` address to the `server1` server;
- the `server1` server relays the message to the `server2` server, because the `server2` address is included into the `server1` Client IP Addresses list;
- the `server2` server relays the message to `username@somehost`, since it has received it from `server1`, which is included into the `server2` Client IP Addresses list.

When servers relay only "simple" E-mail addresses to each other, those servers cannot be used for "two-server relaying" even if they maintain "mutual trust" (i.e. list each other in the Client IP Addresses lists).

To avoid problems with old mail servers that ignore the quote marks in addresses, the addresses with the local part containing quotes cannot be relayed to Client IP Addresses servers if the "simple" option is selected.

If the Relay to Client IP Addresses option is set to "no", these addresses are not processed in any special way - messages sent to servers with Client IP Addresses are processed in the same way as messages sent to servers with non-Client IP Addresses.

Client-only Logins

Non-Client IP Addresses
Reject all Logins from Non-Client IP Addresses

If you do not plan to support [mobile users](#), you may want to select the Reject all Logins from from Non-Client IP Addresses option to allow any type of "login" operations from the Client IP Addresses only. Connections from other addresses are accepted, but only the services that do not require "login" operations will be available: SMTP mail transfer, Personal Web Sites and public Mailing List browsing, etc.

Note: Please check that your Client IP Addresses field is filled with your client addresses and read the [Security](#) section before you select this option.

Relaying for Mobile Users

If some of your users travel a lot, they may use various ISPs to connect to the Internet, and as a result they will connect to your Server from various IP addresses. If those users use your Server as the SMTP mail relay to which they submit all outgoing messages, Relay Restrictions will not allow them to send messages when their IP addresses are not in the Client IP Addresses list.

The SMTP AUTH method

Many E-mail clients (including Microsoft Outlook Express, Netscape Messenger, Qualcomm Eudora, and many others) now support "SMTP AUTH" - the standard SMTP Authentication method that allows a mailer to authenticate the user (the sender). If the SMTP module receives a message from an authenticated user, the message is marked as being "submitted from a local account", and this message can be relayed to the Internet.

The Read-then-Send method

To allow mobile users with older mailer applications (those not supporting SMTP AUTH) to send messages via the CommuniGate Pro server, the POP, IMAP, and other "access-type" modules check if an authenticated user has connected from an IP address not listed as one of the Client Addresses. During that POP/IMAP session, and for some time after the session is closed, that IP address is considered to be a "Client Address", so that users can send mail via your Server right AFTER they have checked their mailboxes.

Non-Client IP Addresses

Reject all Logins from Non-Client IP Addresses

Allow Mobile Users to Login and

Process as a Client IP Address for

after the user disconnects

Remember up to:

such addresses

The expiration time is used because of the "dynamic IP address" policies of most ISPs: when a user disconnects from an ISP modem pool, and some other user connects to the Internet via the same ISP, the same IP address can be assigned to that other user.

Inform your users about the expiration time. They should compose all their messages off-line, then they should connect to the Internet using any ISP, check their mailbox on your Server, and only then they can send the queued outgoing messages. If they want to reply to some messages they have just retrieved from the mailbox on your Server, they should use the Get Mail command in their mailer application again, and only then can they send their replies.

Since many mailer applications try to send queued messages first, the SMTP module checks the Return-Path (the address in the Mail From SMTP protocol command). If that address is an address of a registered user, a to-be-relayed message is not rejected with the "permanent failure" error code. Instead, a "temporary failure" code is returned (with the "try to authenticate first" comment). Many mailers do not interrupt the mail session when they receive such a code, and continue by authenticating the user, retrieving the user mail, and retrying to send the queued messages. The queued messages will be accepted this time, because the user is authenticated from the same address.

An SMTP (message submit) session should start either during a POP or IMAP session, or within the expiration time after the end of the POP/IMAP session. Then that SMTP session can last as long as needed (several hours), if the queued messages are large and the link is slow.

Account and Domain Settings

Support for the mobile users can be disabled on per-account and per-domain basis by disabling the Mobile option in the [Enabled Services](#) section on the Account Settings and Domain Settings pages. If this service is disabled for an account, the account user will not be able to connect to that account from an internet address not included into the Client IP Addresses list.

Mail relaying for the mobile users can be disabled on per-account and per-domain basis by disabling the Relay option in the [Enabled Services](#) section on the Account Settings and Domain Settings pages. If a user or a domain has this service disabled, the IP address from which they log in are not remembered as "temporary client IP addresses", and the SMTP Authentication will not allow those users to relay messages via your SMTP module. This setup is useful when you give users accounts on your server, but you do not want them to be able to relay SMTP mail through your server (they are forced to submit messages using the WebUser Interface or any other non-SMTP methods).

Relaying Rerouted Messages

Read this section if you need to provide special relaying features.

If you place an alias record into the Router table:

```
NoRelay:<user> = user@other.host
```

then all mail from strangers to that user will be rerouted to that other.host server. If that server address is not included into the Client IP Addresses list, these messages will be treated as messages "from a stranger to a stranger", and they will be rejected if the Relay for Clients Only option is switched on. To enable relaying, use the Relay: prefix or just use a record without any prefix:

```
Relay:<user> = user@other.host  
<user> = user@other.host
```

When an address is being converted with such a record, it gets a marker that allows the server to relay messages to that address. If an address is modified with a record that has the NoRelay: prefix, this marker is not set, but it is not reset either - if it has been set with some other Router record (see the example below).

The same situation exists if you want to reroute all mail for a certain domain to a different host (for example, if you back up that host), and that host address is not included into the Client IP Addresses list.

```
Relay:clienthost.com = client1.com  
Relay:<*@clienthost.com> = client1.com
```

When the address modified with the Router record is not a "simple address", i.e. it contains several routes, as in user%host1@host2, or <@host2:user@host1> - the Relay: prefix does not set the flag that allows message relaying. This is done because the host to which the rerouted message is relayed may "trust" all messages that come from your host, and relaying addresses with multiple routes would allow someone to relay messages to anybody through your host and that other host.

If the receiving server is well-protected, too, you may need a Router record that allows relaying of any address rerouted with that record. Use the RelayAll: prefix for those records:

```
RelayAll:<report-*@clienthost.com> = report-*@client1.com
```

Very often you do not want the Router records to be used for actual relaying - you provide them for your own clients only, to specify a special path for certain addresses/domains. For example, if you want mail to bigprovdier.com to be sent via a particular relay relay3.com, you should place the following record into the Router table:

```
NoRelay:bigprovdier.com = bigprovdier.com@relay3.com.smtp
```

Without the NoRelay prefix, any host on the Internet could send messages to bigprovdiar.com via your Server. The NoRelay prefix tells the Router not to add marker to addresses in the bigprovdiar.com domain, so only your own users (clients) can send mail to bigprovdiar.com domain using your Server.

Note: you may have an alias record in your Router:

```
<joe> = joe5@bigprovdiar.com
```

This record tells the server to reroute all mail addressed to joe@mydomain.com to joe5@bigprovdiar.com. Since this record has the (default) Relay: prefix, anybody in the world can send messages to joe@mydomain.com and those messages will be successfully relayed to the bigprovdiar.com domain. The joe5@bigprovdiar.com address will be converted to joe5%bigprovdiar.com@relay3.com.smtp and sent via relay3.com host: the second address transformation does not add the "can relay" marker, but it does not reset the "can relay" marker set during the first transformation:

Operation Applied	Address	Marker
Received (Original) address	joe@mydomain.com	NO
Main Domain (mydomain.com) cut-off:	joe	NO
Router Record: <joe> = joe5@bigprovdiar.com	joe5@bigprovdiar.com	YES
Router Record: NoRelay:bigprovdiar.com = bigprovdiar.com@relay3.com.smtp	joe5%bigprovdiar.com@relay3.com.smtp	YES
SMTP Module: accepted for the host relay3.com	joe5@bigprovdiar.com	YES

Return-Path Address Verification

If your SMTP module can accept incoming TCP connections, your server can be used by spammers as a mail relay engine: they can distribute their messages all over the world using your server. To protect your site from spammers, the SMTP module can verify the Return-Path address (specified with the Mail From SMTP command) of incoming messages.

When the Verify HELO and Return-Path option is selected in the SMTP Service Settings, the SMTP module parses the message Return-Path (Mail From) addresses, and the module refuses to receive a message if:

- the Return-Path domain name is an empty string (no domain specified);
- the Return-Path address is routed (via the Server Router) to the ERROR address;
- the Domain Name System does not have MX or A records for the Return-Path domain (an unregistered domain);
- the Domain Name System has an MX record for the Return-Path domain, but it points to an A-record that does not exist (a faked domain);

- the A-record or the highest-priority MX record for the Return-Path domain points to an IP address included into your Blacklisted IP Addresses list;
- the Return-Path domain name is specified as an IP address, and that address is not included into the Client Addresses list.

The SMTP module uses the [Router](#) after it parses the Mail From address. If that address is an address of a local user, or the address is known (rerouted) with the Router, the Mail From address is accepted. This eliminates Domain Name System calls for the addresses "known" to the Server.

The addresses routed to the ERROR address are rejected, so you can specify "bad" addresses and domains in the Router.

Examples:

If you do not want to accept mail from any address in the offenderdomain.com domain, put the following line into the Router settings:

```
offenderdomain.com = error
```

or

```
<*@offenderdomain.com> = error
```

If you do not want to accept mail from all addresses starting with "promo" in the offenderdomain.com domain, put the following line into the Router settings:

```
<promo*@offenderdomain.com> = error
```

If the Return-Path domain cannot be verified because the Domain Name Server that keeps that domain records is not available, the module refuses to accept the message, but instead of a "permanent" error code the module returns a "temporary" error code to the sending system. The sending system will try again later.

Blacklisting Offenders

Since your SMTP module can accept incoming TCP connections, your server can be used by spammers as a mail relay engine: they can try to distribute their messages all over the world using your server, and they can also send a lot of unwanted messages to your account users.

To protect your system from known spammer sites, CommuniGate Pro provides several methods to maintain "black lists" of offending hosts IP addresses.

When a "blacklisted" host connects to your server and tries to submit a message via SMTP, it gets an error message from your SMTP module and mail from that host is not accepted.

Use the WebAdmin Interface to open the Protection pages inside the Settings section (realm), and click the Blacklisted IP Addresses link.

Specifying Offender Addresses

Enter the IP addresses of offending hosts in the Blacklisted IP Addresses field:

Blacklisted IP Addresses	

Each line can contain either one address:

10.34.56.78

or an address range:

10.34.50.01-10.34.59.99

A comment can be placed at the end of a line, separated with the semicolon (;) symbol. A line starting with the semicolon symbol is a comment line, and it is ignored.

Using DNS-based Blacklisting (RBL)

It is difficult to keep the Server "blacklist" current. So-called RBL (Realtime Blackhole List) services can be used to check if an IP address is known as a source of spam.

Some ISPs have their own RBL servers running, but any RBL server known to have a decent blacklist can be used with your CommuniGate Pro server. Consult with your provider about the best RBL server available.

To use RBL servers, select the Use Blacklisting DNS option and enter the exact domain name (*not* the IP address!) of the RBL server. Now, when the SMTP module accepts a connection from an IP address `aaa.bbb.ccc.ddd` and this address is not listed in the Blacklisted, Unblacklistable, or Client Addresses lists, the module composes a fictitious domain name `ddd.ccc.bbb.aaa.rbl-server-name` where `rbl-server-name` is the domain name of the RBL server you have specified.

The SMTP module then tries to "resolve" this name into an IP address. If this operation succeeds and the retrieved IP address is in the 127.0.0.2-127.1.255.255 range, then the aaa.bbb.ccc.ddd address is considered to be blacklisted.

Note: this option results in an additional DNS (Domain Name System) operation and it can cause delays in

incoming connection processing.

Use Blacklisting DNS

You can specify several RBL Servers using the last (empty) field in the RBL Server table. To remove a server from the list, enter an empty string into its field. The more servers you use, the larger the incoming connection processing delay. If you really need to use several RBL servers, but do not want those additional delays, make your own DNS server retrieve the RBL information from those servers (using daily zone updates) and use your own DNS server as an RBL server.

Note: An RBL server failure can cause very long delays for incoming connections. To avoid these situations, the requests to RBL servers are sent not more than twice, each time with the minimal time-out.

Blacklisting IP Domains

When a client connects from an IP address not listed in the Client IP Addresses and Blacklisted IP Addresses lists, and the Blacklist by DNS Name option is enabled, the server tries to get the domain name for that IP address (if the IP address is *aa.bb.cc.dd*, the Server tries to retrieve the PTR record for the *dd.cc.dd.aa.in-addr.arpa* name). If the PTR domain name is retrieved, it is checked against the strings specified in the table (these strings can include the wildcard (*) symbols). If the retrieved name matches one of the table strings, the address is processed as a blacklisted one.

Blacklist by DNS Name

Note: if the Blacklist by DNS Name option is enabled, the server has to make an additional reverse-lookup DNS operation (unless the [Detect Clients by DNS Name](#) has been already enabled). This additional DNS operation can cause additional delays when processing incoming SMTP connections, so enable this option only when needed, and only when you cannot specify all blacklisted addresses explicitly - in the Blacklisted IP Addresses list.

Un-listing Addresses (White Hole Addresses)

When you use external RBL Servers, you may want to avoid blacklisting certain sites, even if they are included into the RBL Server tables, or into your own Blacklisted IP Addresses list.

Enter those "unblacklistable" addresses using the same format you use for Blacklisted IP Address list:

UnBlacklistable (White Hole) IP Addresses

Note:Addresses listed in your Client IP Addresses list are never checked using any Blacklisting method, so there is no reason to include the Client IP Addresses into the UnBlacklistable IP Addresses table once again.

You can "unblacklist" addresses using their DNS (PTR) names:

UnBlacklist by DNS Name

Select the checkbox to enable this option and enter the DNS domain names you do not want to be blacklisted. This can be useful if some "good" addresses are blacklisted with the RBL services you use.

Note: The explicitly specified Blacklisted IP Addresses cannot be "unblacklisted" using the DNS Names.

Spam Traps

You can protect your site from incoming spam by creating and advertising one or several "spam-trap" E-mail addresses. The CommuniGate Pro Router detects a special local address, `spamtrap`. If your server receives a message, and at least one of its recipients is `spamtrap@yourhost` or at least one of its recipients is routed to `spamtrap`, the Server rejects the entire message.

You may want to create one or several alias records for "nice-looking" fictitious E-mail addresses and route those addresses to `spamtrap`:

```
<misterX> = spamtrap  
<johnsmith@subdomain.com> = spamtrap
```

You do not have to create fictitious accounts, you should create the Router alias records only.

Then you should do your best to help these addresses (`misterX@yoursite.com`, `johnsmith@subdomain.com`) to get to the bulk mailing lists used by spammers. Since most of those lists are composed by robots scanning Web pages and Usenet newsgroups, place these fictitious addresses on Web pages and include them into the signatures used when you and your users post Usenet messages. To avoid confusion, make the fictitious E-mail addresses invisible for a human browsing your Web pages and/or attach a comment explaining the purpose of these addresses.

Many bulk mailing lists are sorted by the domain name, and as a result many spam messages come to your site addressed to several recipients. These recipients are the E-mail addresses in your domain(s) that became known to spammers. When the fictitious, "spam-trap" addresses make it to those databases, most of spam messages will have these addresses among the message recipients. This will allow the Server to reject the entire messages, and they will not be delivered to any real recipient on your site.

Banning Mail by Header and Body Lines

You can specify a set of message Header and Body lines to be used to detect spam. When the server receives mail in the RFC822 format (via [SMTP](#), [RPOP](#), [POP XTND XMIT](#), [PIPE](#) modules), it compares each received header and body line with the specified lists. If a message contains one of the specified lines, the message is rejected.

You can use the wildcard ('*', asterisk) symbols in the Banned Lines you specify. Usually you should not use them, since you are expected to compose the "banned" lists by copying header or body lines from the known spam messages.

Message lines are compared to the specified Banned lines in the **case-sensitive** mode.

Each Header line can include the end of line symbols if the header field was "wrapped".

If a message header or body is encoded (using MIME or UU encoding), the lines are **not** decoded before they are compared to the Banned line sets.

To specify the set of Banned Lines, open the Protection page in the Settings section of the WebAdmin Interface, and click the RFC822 Receiver link.

[illegible]

To add a new line, enter it in the empty field, and click the Update button.

To remove a line, delete it from its field, and click the Update button.

Filtering Mail

When a message is received with the Server, a set of [Server-Wide Rules](#) is applied. These Rules can be used to detect unwanted messages and reject, discard, or redirect them.

For example, the following Rule can be used to reject all messages that have an empty string in their To : header fields:

Data	Operation	Parameter
Action	Parameters	

You can create various filtering rules using all features of CommuniGate Pro Automated Mail Processing, including external filter programs started with the Execute Rule Action.

Cluster Setup

When a Server is a member of a [Dynamic Cluster](#), the WebAdmin Protection Settings pages provide links that allow you to switch between the local (server-wide) and the cluster-wide Settings.

The cluster-wide Address Tables (Client IP Addresses, Blacklisted IP Addresses, Unblacklistable IP Addresses) are processed as extensions of the server-wide tables: an address is considered to be listed if it is included into either the server-wide or into the cluster-wide table.

The cluster-wide "Client By Name" list is processed as an extension of the individual server-wide list of "client domains" (if the Detect Clients by DNS Name option is enabled on the cluster-wide page).

The cluster-wide list of "Blacklisted" RBLs is processed as an extension of the individual server-wide RBL server lists. Each server will consult with the locally-specified RBL servers first, then it will consult with

the RBL servers specified in the cluster-wide settings.

The cluster-wide "Banned" settings are processed as extensions of the server-wide settings: a message is banned if its header or body line is listed in the server-wide or in the cluster-wide settings.



Security

The CommuniGate Pro Server ensures that only certain users are allowed to access certain resources.

The CommuniGate Pro Server can authenticate the users, and can also reject connections from outside of the "client networks".

Authentication Methods

The CommuniGate Pro Server supports both clear-text and secure SASL authentication methods for the following protocols:

- POP (as specified in RFC1734).
- IMAP (as specified in RFC2060).
- LDAP (as specified in RFC2251).
- ACAP (as specified in RFC2244).
- SMTP (as specified in RFC2554).

These secure methods allow mail clients to send encrypted passwords over non-encrypted and insecure links. If anybody can monitor your network traffic, SASL methods ensure that the real passwords cannot be detected by watching the client-server network traffic.

As an alternative to SASL methods, secure links (SSL/TLS) can be used between the client mailer and the server. When an SSL link is established, the entire network traffic between the server and the client is encrypted, and passwords can be sent in clear text over these secure links.

You can force an account user to use either a SASL authentication method or SSL/TLS links if you enable the `Secure Method Required` option in the Account Settings. When this option is enabled, the Server rejects all authentication requests that send passwords in the clear text format over insecure links.

The CommuniGate Pro Server supports the following secure SASL authentication methods:

- CRAM-MD5
- DIGEST-MD5
- NTLM (non-standard method used in Microsoft® products)

The CommuniGate Pro Server supports the following insecure SASL authentication methods:

- PLAIN
- LOGIN

Besides, the CommuniGate Pro supports the secure APOP authentication method (used mostly for the POP protocol), and the insecure "regular login" method for the protocols that support Clear Text Login.

Use the WebAdmin Interface to open the Obscure page in the Settings realm to switch the Advertise Secure SASL Methods and the Advertise NTLM SASL Method options:

Login Security
Advertise Secure SASL methods
Advertise NTLM SASL method

Advertise Secure SASL methods

In certain situations (for example, when all Accounts use External Authentication or OS Passwords), you may want to disable this option, so your Server will not advertise the supported secure SASL methods.

Advertise NTLM SASL method

Some Microsoft products send incorrect credentials when they detect that the server supports the NTLM SASL method. While those products then resend the correct credentials, the failed login attempts produce Failure-level Log records and may increase the "failed logins" counter too quickly, so the account becomes "temporarily locked". Disable this option if you do not want your Server to advertise the NTLM SASL method.

Account Passwords

The CommuniGate Pro Server supports several passwords for each account.

One password is the CommuniGate Pro's "own password ". This password is stored as an element of the Account Settings, and it can be used with the CommuniGate Pro Server only.

The other password is the "OS password". The user may be registered with the Server OS and the CommuniGate Pro Server can check the supplied password against the password set in the Server OS registration information for this user.

An account can have the External Password option enabled. In this case, user authentication is done using any custom authentication program running as a separate process (see below).

The system administrator can [enable](#) any set of passwords for any user account. On larger sites, it is better to enable these options using the Server-wide or Domain-wide Default Account Settings.

When several passwords are enabled for an account, the Server first checks the CommuniGate (internal) password, then the OS password, and then tries to use the External Authentication program. If at least one of these passwords matches the password presented with the client application, the application is granted access to that account.

CommuniGate Passwords

CommuniGate passwords are strings stored in the Account Settings. Password strings can be stored in the clear-text format or in encoded format. The Password Encryption Account Setting specifies the encryption to use when **updating** account passwords.

When the U-crpt Password Encryption option is selected, the CommuniGate passwords are stored using the standard Unix crypt routine. Since this is a one-way encryption, such passwords cannot be used for secure ([SASL](#)) Authentication Methods. Use this option only if you need compatibility with legacy password strings, but cannot use the OS passwords.

The U-crypted passwords can have some special format, so they are recognized as MD5- and SHA1-digested passwords, not as Unix-crypted passwords. See the [Migration](#) section for more details.

If the CommuniGate Password is absent or empty, it cannot be used to log into the account even if the Use CommuniGate Password option is enabled. But if the user has logged in using the OS Password or the External Authentication method, the user can specify (update) the account CommuniGate Password. This feature can be used to [migrate](#) users from legacy mail systems where you can not compose the list of accounts with non-crypted user passwords.

OS Passwords

When the Server checks the OS password, it composes the *username* string using the account [OS User Name setting](#). When the default setting * is used, the composed OS user name is the same as the account name. By changing the OS User Name settings you can use different OS usernames for accounts in different CommuniGate Pro domains.

Server Operating System	Notes about OS Passwords
Microsoft Windows 9x	OS does not support passwords, the Use OS Password option does not work.
Microsoft Windows NT/2000	<p>The Windows NT domain authentication system is used. The Windows account used to run the CommuniGate Pro Messaging Server should have the Act as part of the operating system privilege.</p> <p>The --BatchLogon command line option can be used to tell the Server to use the LOGON_BATCH authentication method (if the option is not present, the LOGON_NETWORK method is used).</p> <p>The Server checks if the composed OS user name contains the percent (%) symbol. If the symbol is found, the part of the name before that symbol is used as the Windows account name, and the part after that symbol is used as the Windows domain name.</p> <p>If Accounts in the company1.dom CommuniGate Pro domain have the OS User Name setting set to *%comp1, then the OS user name for the CommuniGate Pro Account joe will be joe%comp1, and the CommuniGate Pro Server will use the Windows LogonUser API to try to authenticate the mail client as the Windows user joe in the Windows domain comp1.</p>
Unix-based systems	The passwd and shadow or any other OS-supported authentication mechanisms are used.
OS/400 systems	The user profile authentication mechanisms are used.
BeOS	OS does not support passwords, the Use OS Password option does not work.

The OS passwords are one-way-encrypted passwords. As a result, they cannot be used for [Secure Authentication Methods](#).

External Authentication

The CommuniGate Pro Server can use an external application (program) for user authentication. That program should be created by your own technical staff and it can implement authentication

mechanisms required at your site but not supported directly with the CommuniGate Pro Server.

The program name and its optional parameters should be specified using the WebAdmin Helpers page. Open the General page in the Settings realm, and click the Helpers link:

External Authentication	
Log:	Program Path:

When the External Authenticator option is enabled, the Server starts the specified program as a separate OS process.

The Server then sends commands to the External Authenticator process via the process *standard input* and reads responses from the process *standard output*. Each command starts with a sequence number, and the response produced with the External Authenticator process starts with the same number. This method allows the External Authenticator program to process several requests simultaneously, and it can return responses in any order.

Commands and responses are text lines, ending with the EOL symbol(s) used in the Server OS.

Note: communication between the Server and an External Authentication program takes place via OS *pipes*, and many programming libraries buffer output data sent to pipes. Check that your External Authentication program uses some form of the *flush* command after it sends its response to its standard output, otherwise the response will never reach the Server.

The Interface Version command is used to provide compatibility between different versions of External Authenticator programs and different versions of the CommuniGate Pro server. The server sends this command specifying the protocol version it implements:

```
nnnnnn VRFY name@domain password EOL
```

where:

```
nnnnnn
```

a unique sequence number for this request

```
serverInterfaceVersion
```

the version of the External Authenticator protocol implemented by this version of the CommuniGate Pro Server.

The External Authenticator program should return the OK response and the supported protocol version.

```
nnnnnn OK programInterfaceVersion
```

If the returned number is smaller than the Server protocol version, the Server will use this (older) protocol version:

When a user should be authenticated using the *clear text* method, the Server sends the following command:

```
nnnnnnn VRFY name@domain password
```

or

```
nnnnnnn VRFY (mode) name@domain password
```

where:

nnnnnnn

a unique sequence number for this request

mode

the name of the service (IMAP, POP, FTP, etc.) that requested this authentication operation. This parameter can be absent if the request has been received from an unnamed Server component. If the service name is specified, it is enclosed into the parenthesis. If the protocol version supported with the External Authenticator program is less than 1, the server never sends the *mode* parameter.

name

user account name

domain

user account domain name

password

password string to verify

When a user should be authenticated using a secure [SASL](#) method, the following command is sent:

```
nnnnnnn SASL(method) name@domain password key EOL
```

or

```
nnnnnnn SASL(method) (mode) name@domain password key EOL
```

where:

method

the name of the secure SASL method used (CRAM-MD5, APOP)

key

the *challenge* string sent to the client mailer.

If the password is accepted, the External Authenticator should return a positive response:

```
nnnnnnn OK
```

If the password was not accepted, a negative response should be returned:

```
nnnnnnn ERROR optional-error-message
```

Sample session (I: - server commands sent to the program standard input, O: - responses the program writes to its standard output):

```
I: 00001 INTF 1
```

```
O: 00001 OK 1
I: 00010 VRFY user1@domain1.com dsyui134
O: 00010 OK
I: 00011 VRFY (IMAP) user2@domain2.com jskj23#45
O: 00011 ERROR incorrect password
I: 00012 SASL(CRAM-MD5) user4@domain2.com hdkj547812329394055 <pop-23456@mydomain.com>
O: 00012 ERROR unsupported SASL method
```

The External Authentication program can process several requests simultaneously: it can read the next request before sending responses for all previously received requests. The program can return responses in any order, specifying the correct sequence numbers:

Sample session:

```
I: 00010 VRFY (POP) user1@domain1.com dsyui134
I: 00011 VRFY (POP) user2@domain2.com jskj23#45
O: 00011 ERROR incorrect password
O: 00010 OK
```

The External Authentication program can be used to process unknown names, too. For example, the program can consult an external database, check if the user exists in that database, create an Account, Alias, Group, Mailing List, or Forwarder using the CommuniGate Pro [CLI/API](#), and return a positive response to the Server. In this case, CommuniGate Pro will re-try to open a domain object with the specified name.

To check an unknown name, the Server sends the following command:
nnnnnn NEW name@domain

Sample session:

```
I: 00010 NEW user1@domain1.com
O: 00010 ERROR this account is not known
I: 00011 NEW user2@domain2.com
O: 00012 OK
```

The Consult External Authenticator [Domain Setting](#) tells the Server to use the External Authenticator program when an unknown name is addressed.

When the Server shuts down or when it needs to stop the External Authentication program, it closes the process *standard input*. The External Authentication program detects the end-of-file condition and it should quit within 5 seconds.

Log

Use this setting to specify what kind of information the External Authentication module should put in the Server Log. The module records in the System Log are marked with the EXTAUTH tag.

Sample External Authentication programs and scripts can be found at the <http://www.stalker.com/CGAUTH/> site.

Account Name Harvesting

Some spammers use 'brute force' attacks on mail systems, sending random names and passwords to system POP, IMAP, and other access ports. If the system sends different error messages for the "unknown account" and "incorrect password" situations, the attacker can harvest a large portion of the system account names and then use those names for spam mailings.

To prevent this type of attack, you may want to enable the Hide Unknown Account messages option, located on the Obscure page in the WebAdmin Settings realm:

Login Security	
Hide Unknown Account messages	
Suspend Account after	failed logins within

Hide Unknown Account messages

If this option is enabled, the Server does not send the Unknown Account and Incorrect Password error messages. Instead, both messages are replaced with the Incorrect Account Name or Password error message.

Account Password Attacks

The CommuniGate Pro server can temporarily disable all types of login operation for an Account that has seen too many incorrect login attempts. The Login Security panel shown above allows you to specify a time period and the number of incorrect login attempts that a user or users can make before the Account is disabled for login operations. The Account is re-enabled after the same period of time.

Granting Access Rights to Users

In order to control, monitor, and maintain the CommuniGate Pro Server, one Postmaster account is usually enough. But you may want to allow other users to connect to the CommuniGate Pro Server: for example, you may want to allow an operator to monitor the Logs, but you do not want to grant that operator all Postmaster access rights.

You should be logged in as the Postmaster, or you should have the "Can Modify Access Rights" right in order to assign access rights.

To grant access rights to a user and/or to revoke those rights, open that user Account ([the Account Setting page](#)), and click the Access Rights link. The Access Rights page will appear.

The page lists all Access Rights and the rights granted to the selected user are marked.

The following access rights can be granted only to the users (accounts) in the main domain:

Can Modify Access Rights (unlimited access):

This setting specifies if the user is allowed to modify Access Rights of CommuniGate Pro users. If some users are granted this right, they can access all Server settings and pages (i.e., all other rights are granted, too).

Can Modify User Accounts

This setting specifies if the user is allowed to create, remove and delete Accounts and Domains, and to change Account and Domain Settings.

Can Modify Server Settings

This setting specifies if the user is allowed to change configurations of CommuniGate Pro services (SMTP, POP, Router, etc.).

Can Monitor Server

This setting specifies if the user is allowed to view Server Logs, to monitor Server Queues, etc.

The following access rights can be granted to users in any domain:

Can Modify This Domain and its Account Settings:

This setting specifies if the user is allowed to create, remove and delete Accounts within its own domain, and to change some of the Domain Settings. You usually assign this right to a user ("domain master") who will manage the domain.

Initially, the user Postmaster in the main domain has the Unlimited Access right.

Select the desired Access Rights and click the Update button.

The Access Rights are stored in one file for each domain, the Access.settings file stored in

the Settings subdirectory of the domain directory. This makes it easy to check to whom the Server administration rights are granted.

Restricting Access

If you do not plan to support mobile users, you may want to restrict access to the Server accounts. Use the following option on the Protection page:

Reject all Logins from Non-Client Addresses

When this option is selected, all "login" operations (needed for POP, IMAP, WebUser Interface, ACAP, PWD, etc.) are accepted only from the Server computer itself, and from the [Client IP Addresses](#).

When an access module accepts a connection from an unlisted network address, and this option is selected, the module sends an error code to the client application, and the connection is closed immediately. Links with the rest of the Internet will be used only for mail [Transfer](#) and access to [Personal Web Sites](#).

When this option is selected, the SMTP AUTH operation can be used only if a client mailer or server connects from the network address included into Client Addresses list.

Note: Before you enable this option, make sure that the address you are using is included into the Client Addresses list: otherwise you will immediately lose access to the Server.

You can also specify the access restrictions on the lower (TCP) connection level. For each service (module), open the [Listener](#) page and specify the addresses the service (module) should or should not accept connections from. If a connection comes from an address that is not included into the Grant list or is included into the Deny list, the connection is closed immediately, and no module-level operations are performed.

Using SSL/TLS Secure Connections

The CommuniGate Pro Server supports secure (encrypted) connection for all its services and modules. Secure connections can be established in two ways:

- A client mailer uses a special port to connect to the Server and starts to establish a TLS (encrypted) connection right after a TCP connection with that port is established. This method is used in Netscape and Microsoft mail clients released before 2000. To support these clients

with the CommuniGate Pro, configure the [listeners](#) for POP, IMAP, ACAP, SMTP, and LDAP modules: the Listeners should accept TCP connections on those special ports (see the module descriptions for those port numbers) and the Security option should be enabled for those ports, so the CommuniGate Pro server starts to establish a secure connection as soon as a TCP connection to those ports is established.

- A client mailer uses a standard port to connect to the Server, and then issues a special (STARTTLS) command over the established clear-text TCP connection. When the Server receives such a command it starts to establish a secure connection. To support these clients you do not have to configure additional ports with the module Listener. At this moment, this method is implemented in the Netscape mailer only, and only for SMTP connections.

To specify the server SSL/TLS processing parameters, open the Obscure page in the Settings section of the WebAdmin Interface:

TLS Sessions	
Log:	Time To Live:

Log

Use this setting to specify what kind of information the TLS module should put in the Server Log. The TLS module records in the System Log are marked with the TLS tag.

Time To Live

This setting specifies the *cache time* for TLS sessions. When all connections using the same TLS session are closed, the Server waits for the specified time before deleting the TLS session parameters. This feature allows clients to open new connections *resuming* the old TLS sessions. It increases connection speeds and decreases the Server CPU load. This feature is especially important for HTTP clients that open and close connections very often.

Certificates and Private Keys

When a secure connection is being established (using the TLS/SSL protocol), a public key cryptography is used to:

- ensure that the server is really the server the client wants to connect to
- safely exchange the random data used to compose a "shared secret" - a key for "regular", symmetric cryptography used to encrypt actual data transferred via a secure connection.

A server must possess a so-called "private key" and a "certificate" that contains a public key. When a client starts to establish a secure connection, the server sends its certificate to the client. The client:

- uses the information in the Certificate to ensure that it has connected to the proper server;

- uses the public key to encrypt its portion of the "shared secret" data and sends that data to the server, which decrypts that data using its Private key.

The Certificates themselves use the Public Key cryptography, too. The Certificate contains the information about the server and the information about the organization or entity that has issued the Certificate. The entire Certificate is digitally signed using the Private Key of the issuer, and it is practically impossible to forge a certificate without make the digital signature invalid.

Modern browsers and mail clients have the Public Keys of several "known authorities" (issuers) built-in. As a result, they can verify the digital signatures of the Certificates issued by those "known authorities". It is assumed that such "known authorities" take reasonable steps to ensure that they issue a Certificate for abc.def domain to the legal owner of that domain.

Domain Security Settings

CommuniGate Pro allows you to specify Private Keys and Certificates on the per-domain basis. The Keys and Certificates can be assigned only to the CommuniGate Pro [Domains](#) that have one or several assigned network (IP) addresses. This limitation comes from the secure versions of the mail protocols used today: when a client wants to initiate a secure connection, the Server has no information about the Domain the client wants to connect to, and only the local IP address to which the client has connected is known.

You can configure a CommuniGate Pro Domain to use a Server-wide Private Key and a Certificate - but you should use this feature for testing purposes only.

To enter the domain Private Key and Certificate, use the WebAdmin Interface and follow the Security link on the Domain Settings page. The Domain Security page will open:

Secure Certificate To Use:

This option allows you to specify which private keys and certificates the CommuniGate Pro Server should use when a client wants to establish a secure connection with this domain.

No

if this option is selected, secure communications with this domain are prohibited

Server

if this option is selected, the Server-wide Private key and Certificate are used for secure connections to this domain. You do not have to configure any other Domain Security options in this case. Use this mode for testing only.

Custom

if this option is selected, the Domain Private Key and Certificate are used to establish secure connections with this domain.

Assigning a Private Key

Initially CommuniGate Pro Domains do not have any Private Keys assigned. You should select the size of the key and click the Generate Key button to create a random Private Key and assign it to the Domain.

Private Key
Size:

Note: depending on your server hardware platform, it can take up to several minutes to generate a 2048-bit Key.

Only after you assign a Private Key, the Certificate-related fields will appear on the Security page.

You can use any third-party program (such as OpenSSL) to generate a Private Key in the so-called PEM format (as shown below), and assign that Key to the Domain. Select "Custom" in the Size: field and click the Generate Key button. A multi-line text field appears. Copy the PEM-encoded Key into that text field, and click the Generate Key button:

Private Key
Size:
Enter a PEM-encoded Private Key:

Note: Because of the export regulations, some US-made products (such as Netscape 4.x) disable strong (more than 512 bit) cryptography for SSL/TLS "shared secret" exchange. Those products expect a server to send them a temporary 512-bit key instead, with the generated longer key being used for certificate validation only. If your users employ software products with disabled strong cryptography, generate and use 512-bit keys only: in this situation longer keys will only increase the server load without any increase in the security level. Alternatively, you can use special Certificates (see below) issued by VeriSign and other companies. Those Certificates contain supplementary attributes that tell "weak" products to lift the restrictions and use strong cryptography instead. If you want to use those special certificates, you should generate 1024-bit or 2048-bit keys.

If the Private Key was set correctly, and the Key can be used for public/private key cryptography, you will see the following panel:

Private Key
MIIBPAIBAAJBAKrbekuRk8V...
Key Test: Verification String is OK

If the Key Test field indicates an error, the generated Private Key cannot be used for public/private key cryptography.

Use the Remove button if you want to remove the entered Domain Private Key. Since the Domain Certificate can be used with one and only one Private Key, it becomes useless when you delete the Private Key, so the existing Domain Certificate will be removed, too.

Assigning a Certificate

To accept secure connections, the Domain must have a certificate issued for that domain. Please note that the clients compare the name in the Certificate to the name they used to connect to the Server. If a CommuniGate Pro Domain has domain aliases, attempts to connect to the Server using a domain alias name will result in warning messages on the client workstations notifying users about the name mismatch. Since the certificate can contain only one name, select the name (real Domain name or one of the Domain Aliases) that your users will use in the mail client settings. If your CommuniGate Pro Domain name is `company.dom`, and that domain name does not have a DNS A-record, but the Domain has an Alias `mail.company.dom` that has an A-record pointing to the CommuniGate Pro Server, your users will use the `mail.company.dom` name in their client settings and WebUser Interface URLs, so the Domain Certificate should be issued for the `mail.company.dom` name rather than the `company.dom` name.

To create a Certificate, fill the fields in the Certificate Attributes table:

Certificate
Common Name:
Country:
Province/State:
City/Town:
Organization Name:
Organization Unit:
Contact E-mail:

Common Name

Each certificate can contain only one Common Name, while a CommuniGate Pro domain can have many aliases. Client applications check that the Certificate Common Name matches the name the user has specified in the URL and/or in the mailer settings.

If the `domain.dom` domain users have to specify the domain alias `mail.domain.dom` name to connect to that CommuniGate Pro domain (because the `domain.dom` name does not have a DNS A-record), then select the `mail.domain.dom` name as the Certificate Common Name.

Contact E-mail

This field must contain a valid E-mail address, though that address does not have to be inside this CommuniGate Pro Domain.

All other fields are optional.

You can create a Self-Signed Certificate if you do not want to use any external authority. Click the Generate Self-Signed button and the CommuniGate Pro Server creates a so-called self-signed certificate: the issuer will be same entity you have specified, and the entire certificate will be signed using the Domain Private Key. When a Domain has a Self-Signed Certificate, client applications will warn user that the addressed server has presented a certificate "issued by an unknown authority". Users can "[install](#)" self-signed certificates to avoid these warnings.

To receive a Certificate from an external source ("trusted authority"), click the Generate Signing Request button. A text field containing the PEM-encoded CSR (Certificate Signing Request) will appear:

Certificate
Common Name:
Country:
Province/State:
City/Town:
Organization Name:
Organization Unit:
Contact E-mail:

Certificate Signing Request (CSR)

submit this request to a Certification Authority and paste the result below

Enter a PEM-encoded Certificate

Copy the CSR text and submit it to the Certification Authority (CA) of your choice. You can submit via E-mail or using a Web form on the CA site. The Certification Authority should send you back the signed Certificate in the PEM-format. Enter that Certificate into the bottom field and click the Set Certificate button.

If the Certificate is accepted, the Certificate information is displayed:

Certificate

Issued to:	Country: US Province: CA City: Sausalito Organization: ACME Yacht Rentals, Inc. Unit: On-line Services Common Name: d1.stalker.com Contact: bill@domain.company	
Issued by:	Country: ZA Province: FOR TESTING PURPOSES ONLY Organization: Thawte Certification Unit: TEST TEST TEST Common Name: Thawte Test CA Root	
Valid:	From: 06-Jun-00	Till: 06-Jan-02

The Certificate panel shows the information about the issuer (the Certificate Authority), the information about the "subject" (the data you have entered and the domain name) and the validity period of this Certificate.

Note: the entered Private Key and Certificate will be used for domain-related communications ONLY if the Secure Certificate To Use option is set to Custom.

Note: the Certificate contains the domain name as a part of the "Subject" data. If you rename the CommuniGate Pro domain, the domain name in the certificate does not change, and the client applications may start to warn users about the name mismatch.

Click the Remove Certificate button to remove the Domain Certificate.

Assigning a Certificate Authority Chain

If the Certificate issuer is known to the users client software (mailers and browsers), the warning message does not appear on the user screen when the client software receives a Certificate from the Server. In many cases, the "trusted authority" does not issue certificates itself. Instead, it delegates the right to issue certificates to some other, intermediate authority. When your Server uses a Certificate issued by such an authority, the Server should also present the Certificate of that authority issued by the "trusted authority". The client software would check your Certificate first, then it will detect that the issuer of your Certificate is not a "trusted authority" and it will check the additional Certificate(s) the Server has sent. If that additional Certificate is issued by a "trusted authority", and it certifies the issuer of your Domain Certificate, your Certificate is accepted.

When you receive a Certificate from a Certificate Authority that is not listed among the "trusted authorities" in the client software settings, that intermediate Certificate Authority (CA) should also

give you its own Certificate signed with a "trusted authority". That Certificate should be in the same PEM format as your Domain Certificate:

Certificate Authority Chain (Optional)

Click the Set CA Chain button to assign the Certificate Authority Chain to the Domian. If the decoded CA Chain format is correct, the list of certificates is displayed:

Certificate Authority Chain (Optional)

Issued to	Issued by	Valid From	Valid Till
Organization: VeriSign Trust Network Unit: VeriSign, Inc. VeriSign International Server Unit: CA - Class 3 www.verisign.com/CPS Unit: Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign	Country: US Organization: VeriSign, Inc. Class 3 Public Unit: Primary Certification Authority	16-Apr-97	07-Jan-04

When set, the Certificate Authority Chain is sent to clients together with the Domain Certificate.

Click the Remove CA Chain button to remove the Certificate Authority Chain from the Domain Security Settings.

Using Default and Self-Signed Certificates

Web browsers and mailer programs check if the server certificate is issued by a "known authority". The list of "known authorities" is built into those programs. If a domain on your CommuniGate Pro server uses either a "Server" certificate, or a custom self-signed certificate, or a custom certificate signed by an authority not known to client browser or mailer, the client application will display an alert message every time it establishes a secure connection with your Server domain.

Your users can "install" your Server domain certificates into their mailers and browsers. Once installed into the client software, a certificate becomes a "trusted" one. For some programs (such as Mac versions of Microsoft Outlook and Outlook Express) installing an "untrusted" certificate is the only way to enable secure communications.

To install a domain certificate, the user should use a browser application and connect to the login page of the [WebUser Interface](#) for the selected domain. If the domain has an enabled Certificate, the Secure Certificate link appears. The user should click on that link to download the domain certificate and "open" it. The browser should allow the user to verify the certificate and install it as a "trusted" certificate.



Scalability

This section explains how CommuniGate Pro and the Server OS can be configured to serve large (10,000-100,000 accounts) sites.

For carrier-level sites (from 100,000 up to several million accounts) the multi-server [Cluster](#) configurations should be used.

Serving Large Domains

If some domains you serve have a large number of accounts (10,000 are more), you should consider storing accounts in [subdirectories](#) rather than in a flat domain directory.

You can move domain subdirectories to other disks, just replace the moved subdirectories with their symbolic links.

You can also move domain directories from the Domains directory and replace them with symbolic links.

Handling High-Volume Local Delivery

When the number of messages to be delivered to local CommuniGate Pro accounts is expected to be higher than 1 message/second, you should allocate more "processors" in the [Local Delivery](#) Module. This is especially important for environments that process heavy inbound SMTP traffic (often used as a performance test environment). Insufficient number of Local Delivery module processors (threads) may result in excessive Queue growth and large latency in message delivery. You should watch the Local Delivery module Monitor and allocate more processors (threads) to that module if you see that the module Queue size grows to more than 200-300 messages. Do not allocate additional threads if, for example, you have 10 Local Delivery processors and see the waiting Local Delivery queue of 200 messages: this Queue size introduces only 1-2 seconds delivery latency. Increase the number of Local Delivery threads only if you see that Queue growing.

Administrators of high-end mail servers may want to disable the User Conservative Updates option (located in the Local Account Options on the WebAdmin Obscure Settings page). This decreases the load on file i/o subsystem.

Supporting Many Concurrent Clients

For ISP and large corporate installations, the number of users that can be served simultaneously is an issue of a very high concern.

In order to estimate how many users you can serve at the same time, you should realize what type of service your clients will use.

POP3 Clients

POP3 mailers connect to the server just to download new messages. Based on the average connections speeds, expected mail traffic, and your user habits, you can estimate how much time an average session would take. For example, if you are an ISP and you estimate that an average your "check mail" operation will take 15 seconds, and they mostly check their accounts during 12 peak hours, then with 100,000 POP3 users you can expect to see $100,000 * 15 \text{ sec} / (12 * 60 * 60 \text{ sec}) = 35$ concurrent POP3 sessions.

This number is not high, but POP3 sessions put a high load on your disk I/O and network I/O subsystems: after authentication, a POP3 session is essentially, a "file downloading" type of activity.

IMAP4 Clients

The IMAP protocol allows a much more sophisticated processing than POP3. Mail is usually left on the server, and some unwanted messages can be deleted by users without downloading them first.

But since the IMAP protocol is "mail access", not "mail downloading" protocol, IMAP users spend much more time being connected to the server. In corporate environments, users can leave their IMAP sessions open for hours, if not days. While such inactive sessions do not put any load on your disk or network I/O subsystems, or CPU, each session still requires an open network connection and a processing thread in the server. Since the IMAP protocol allows users to request search operations on the server, IMAP users can also consume a lot of CPU resources if they use this feature a lot.

WebUser Clients

The CommuniGate Pro WebUser interface provides the same features provided by IMAP mailer clients, but it does not require an open network connection (and processing thread) for each user session. When a client (a browser) sends a request, a network connection is established, the request is processed with a server thread, and the connection is closed.

This allows the server to use just 100 HTTP connections to serve 3,000 or more open sessions.

When you know the type and number of clients you plan to serve, you can estimate the resources they will need on your Server.

Setting the TCP `TIME_WAIT` time

When you expect to serve many TCP/IP connections, it is important to check the time your Server OS waits before releasing a logically closed TCP/IP socket. If this time is too long, those "died" sockets can consume all OS TCP/IP resources, and all new connections will be rejected on the OS level, so the CommuniGate Pro Server will not be able to warn you.

This problem can be seen even on the sites that have just few hundred accounts. This indicates that some of

the clients have configured their mailers to check the server too often. If client mailers connect to the server every minute, and the OS TIME_WAIT time is set to 2 minutes, the number of "died" sockets will grow, and eventually, they will consume all OS TCP/IP resources.

It is recommended to set the TIME_WAIT time to 20-30 seconds.

The TIME_WAIT problem is a very common one for Windows NT systems. Unlike most Unix systems, Windows NT does not have a generic setting for the TIME_WAIT interval modification. To modify this setting, you should create an entry in the Windows NT Registry (the information below is taken from the <http://www.microsoft.com> site:

- Run Registry Editor (RegEdit.exe).
- Go to the following key in the registry:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\tcpip\Parameters
- Choose Add Value from the Edit menu and create the following entry:
Value Name:
 TcpTimedWaitDelay
Data Type:
 REG_DWORD
Value:
 30-300 (decimal) - time in seconds
 Default: 0xF0 (240 decimal) not in registry by default
- Quit the Registry Editor
- Restart the computer for the registry change to take effect.

Description: This parameter determines the length of time that a connection will stay in the TIME_WAIT state when being closed. While a connection is in the TIME_WAIT state, the socket pair cannot be reused. This is also known as the "2MSL" state, as by RFC the value should be twice the maximum segment lifetime on the network. See RFC793 for further details.

Handling High-Volume SMTP Delivery

To handle high-volume (more than 50 messages/second) SMTP delivery load you need to ensure that your DNS server(s) can handle the load CommuniGate Pro generates and that the UDP packet exchange between CommuniGate Pro and the DNS servers does not suffer from excessive packet loss. You may want to re-configure your Routers to give UDP traffic a higher priority over the TCP traffic.

You may want to try various values for the Concurrent Requests setting in the Domain Name Resolver panel on the Obscure Settings page: depending on your DNS server(s) setup, increasing the number of Concurrent Requests over 10-20 can result in DNS server performance degradation.

If an average size of the messages sent via SMTP is higher than 20K, you should carefully select the number of SMTP sending channels (threads), too. Too many concurrent data transfers can exceed the available network bandwidth and result in performance degradation. 500 channels sending data to remote sites with a relatively slow 512Kbit/sec connectivity can generate 250Mbit/sec outgoing traffic from your site. Usually the traffic is much lighter, since outgoing channels spend a lot of time negotiating parameters and exchanging envelope information. But as the average message size grows channels spend more time sending actual message data and the TCP traffic generated by each channel increases.

Estimating Resources Usage

Each network connection requires one network socket descriptor in the server process. On Unix systems, the total number of sockets and files opened within a server process is limited.

When the CommuniGate Pro server starts, it tries to put this limit as high as possible, and then it decreases it a bit, if it sees that the limit set can be equal to the system-wide limit (if the CommuniGate Pro consumes all the "descriptors" available on the server OS, this will most likely result in the OS crash). The resulting limit is recorded in the CommuniGate Pro Log.

To increase the maximum number of file and socket descriptors the CommuniGate Pro Server process can open, see the instructions below.

Each network connection is processed by a server *thread*. Each thread has its own *stack*, and the CommuniGate Pro threads have 100Kbyte stacks on most platforms. Most of the stack memory is not used, so they do not require a lot of real memory, but they do add up, resulting in bigger *virtual memory* demand. Most OSes do not allow the process virtual memory to exceed a certain limit. Usually, that limit is set to the OS swap space plus the real memory size. So, on a system with just 127Mbytes of the swap space and 96Mbytes of real memory, the maximum virtual memory that can be allocated is 220Mbytes. Since the swap space is shared by all processes that run under the server OS, the effective virtual memory limit on such a system will be around 100-150MB - and, most likely, the CommuniGate Pro Server will be able to create 500-1000 processing threads.

On 32-bit computers 4GB of virtual memory is the theoretical memory size limit, and allocating more than 4GB of disk space for page swapping does not change anything.

During a POP3 or IMAP4 access session one of the account mailboxes is open. If that mailbox is a text file (BSD-type) mailbox, the mailbox file is open. During an incoming SMTP session a temporary file is created for an incoming message, and it is kept open while the message is being received. So, on Unix systems, the total number of open POP, IMAP, and SMTP connections cannot exceed 1/2 of the maximum number of socket/file descriptors per process.

While a WebUser session does not require a network connection (and thus a dedicated socket and a thread), it can keep more than one mailbox open.

On Unix systems, when the Server detects that the number of open network sockets and file descriptors is coming close to the set limit, it starts to reject incoming connections, and reports about this problem via the

OS Limitations

This section explains how you can check and increase the limits imposed by various server Operating Systems. The most important limits are:

- The maximum number of files and network sockets a process can open.
- The maximum size of virtual memory available to a process.

Solaris

Solaris `ncsizekernel` parameter has to be *decreased* on the large systems, especially - on Dynamic Cluster backends. The cache this parameter controls cannot keep any usable subset of file paths, but the large cache size causes the system to waste a lot of CPU cycles checking this cache table (symptoms: more than 50% CPU utilization, most CPU time is spent in the kernel). Decrease the `ncsizekernel` parameter value down to 1000-2000.

Windows 9x/NT/2000

The Windows system limits the maximum number port number assigned to outgoing connections. By default this value is 5000. You may want to increase that value to 20,000 or more, by adding the `MaxUserPort` DWORD-type value to the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`, key.

For more details, check the [Microsoft Support Article Q196271](#).

Linux

The pre 2.2.x Linux kernels allowed a process to open 256 files descriptors only. If you want your server to handle more than 100 TCP/IP connections, use the Linux kernel 2.2.x or a later version to avoid the "out of file descriptors" problem.

The Linux threads library uses the one-to-one model, so each CommuniGate Pro thread is a kernel thread (actually, a "process"). This may be not the best solution for very large systems that should run several thousand threads.

In spite of the fact that the Linux threads are handled within the kernel, the Linux thread library has its own scheduler, too. By default, that scheduler uses a static table with 1024 entries, so no more than 1024 threads can be created. This is enough for even large sites serving may POP and WebMail users, but can cause problems for sites that need to serve several hundred concurrent IMAP users. To increase this number, the Linux threads library has to be recompiled with the `PTHREAD_THREADS_MAX` parameter increased.

The Linux threads library allocates thread stacks with 2MB steps. This does not allow the system to start more than 1000 threads on 32-bit machines. CommuniGate Pro threads do not need stacks of that size. You may want to recompile the Linux threads library decreasing the `STACK_SIZE` parameter to 128K.

Mac OS X Server (Rhapsody)

The Mac OS X Server has 1600-1800 descriptors/process "hard limit" set by default.

The Mac OS X sets a 6MB limit on "additional" virtual memory an application can allocate. This is not enough for sites with more than 2,000 users, and you should increase that limit by specifying:

```
ulimit -d 100000
```

command in the CommuniGate Pro startup file.



TCP/IP Listeners

The CommuniGate Pro Server accepts SMTP, IMAP, POP, LDAP, and other TCP/IP connections using *Listeners*. Listener open one or several *listener sockets*, each accepting TCP/IP connections on a specified port number, and, optionally on a specified local IP address.

The Listener settings allow the Server administrator to specify the type of connections to accept (regular, *clear text* connections or *secure* connections), and the optional remote address restrictions - that grant access to the listener sockets only to the computers inside the specified networks.

CommuniGate Pro Listeners can limit the number of incoming connections that come from the same IP address. This can help to prevent some of the Denial of Service (DoS) attacks.

Multi-Socket Listening

For each service that uses a Listener, several *listener sockets* can be created. The socket accepts incoming TCP connections on the TCP port number you specify. You should also specify if the socket should accept connections on all IP addresses your server computer has, or only on a selected address.

For example, you may want to create a socket that accepts all connections on one local IP address, while the other socket is used to accept connections on the other local IP address - and only from the specified networks.

Because of the nature of TCP/IP sockets, you cannot have two listener sockets that use the same port number and the same local IP address: if you create a listener socket on the port N that works with ALL local IP addresses, you cannot create a different socket on the same port N. If you create a listener socket on the port N and a specific local address xx.yy.zz.tt, then you can create a different listener socket on the same port N and a different local address xx.yy.zz.tt.

If your CommuniGate Pro server coexists with some other server software, such as a third-party Web server, you may want to configure that Web server to use one local IP address, while your CommuniGate Pro server would provide its HTTP services on a different local IP address - but on the same port number. If that port number is 80, and the domain name `www.company.com`

resolves into the first IP address, while `mail.company.com` resolves into the second IP address, then typing `http://www.company.com` in a client browser will bring up the third-party Web Server home page, while typing `http://mail.company.com` will bring up the CommuniGate Pro Login page - with both servers running on the same server computer.

Port	Local Address	SSL/TLS	Remote Address Restrictions

To create a new listener socket, change the value in the last table element from 0 to the desired TCP port number and click the Update button.

To remove a listener socket, change its port number to 0 and click the Update button.

Even if your server has only one IP address, you may want to create two listener sockets for most of your services: one for regular, *clear text* connections and one (on a different port number) for secure connections (see below).

Secure Sockets

If you enable the `Init SSL/TLS` listener socket option, the Listener component initiates SSL/TLS negotiations as soon as a connection from a remote site is accepted. Only when a secure connection is established, the Listener allows the communication module to initiate its own protocol (IMAP, HTTP, etc.) - on top of the secure SSL/TLS protocol.

Note: Please read the [Security](#) section and configure your domain certificates before you enable any Secure Socket.

Note: When a Listener accepts a connection on a Secure Socket, it tries to detect the CommuniGate Pro Domain the client has connected to. At this time no information has yet been transferred from the client to the server, so the local server IP address the client has connected to is the only data CommuniGate Pro can use to detect the target Domain. If you want a Domain to have its own Security Certificate and to use it for Secure Socket connections, that Domain **must** have an IP address assigned to it.

When the Domain is selected, the Listener retrieves the Domain Certificate and initiates a secure (SSL/TLS) session. If the selected Domain does not have a Certificate, the connection is dropped and an error message is placed into the CommuniGate Pro Log.

Note: The current versions of the Internet protocols support the STARTTLS/STLS or equivalent commands. These commands are used to provide secure communications **without** creating a special Secure Socket on an additional port. Instead, a regular port is used, and a regular, non-secure connection is established, and then the client sends the STARTTLS or an equivalent command, and the client and server initiate the SSL/TLS session. If the software you use employs the STARTTLS command (as most SMTP software packages do these days), then you do not need to create any special Secure Socket for secure (SSL/TLS) communications.

Restrictions

You may want a listener socket to accept connections only from the certain remote network (IP) addresses.

If you set the Restriction setting to Grant, and list the IP addresses in the text field, the socket will accept connections from the specified addresses only.

If you set the Restriction setting to Deny, and list the IP addresses in the text field, the socket will deny access to all clients trying to connect from the specified (blacklisted) addresses.

The IP addresses should be specified in the multi-line format: each line should contain either one IP address or a pair of addresses (separated with the minus (-) sign) that specify an address range. A line can contain a comment text after a comment separator. The semicolon (;) sign, the percent (%) sign, or the minus (-) sign can be used as comment separators.

There is a difference between the Access Restrictions on the listener socket level, and the restrictions set in the SMTP module. When a remote site connects to your server SMTP port and the site IP address is not accepted by the listener socket, the connection is closed immediately. As a result, the remote site will try all other IP addresses of your server, and then it will try to relay mail via your back-up server.

On the other hand, if the remote site address is included into the Server [Protection](#) Black-List, SMTP sessions are not closed immediately. Instead, the SMTP session starts and the remote (blacklisted) server is allowed to send the addresses of message recipients. But the SMTP module rejects each address with a "fatal error" code, thus stopping the blacklisted host from trying to relay those messages via your back-up servers.

There is a difference between the Access Restrictions on the listener socket level, and the restrictions set by the [Grant Access to the Clients Only](#) option. When a remote site connects to your Server POP, IMAP, WebUser, or other access-type port and the site IP address is not accepted by the listener socket, the connection is closed immediately. As a result, the remote site may try all other IP addresses of your server (and you may have different access restrictions on listener sockets serving those addresses).

On the other hand, if the remote site address is not included into the Server [Client IP Addresses](#) list, sessions are not closed immediately. Instead, an access-type session starts, and, if the Grant Access to Clients Only option is enabled, an error message is sent to the remote site before the module closes the connection.

Limiting Connections from the same Address

To prevent various Denial of Service (DoS) attacks you may want to limit the number of connections a Listener can accept (on all sockets) from the same IP address:

Max Connections from same Address:

When you set this settings, you should remember that:

- IMAP clients usually open several connections to the server. If you set this setting for the IMAP listener to less than 5, you can cause problems for your users.
- Web browsers can open several simultaneous connections to retrieve embedded graphic files and other HTML page elements.
- Many Web clients can connect to your server via the same proxy, and they all appear as connecting to your server from the same IP address.

Note: to avoid problems with inter-server communication, this setting has no effect on connections that come from other servers in the same CommuniGate Pro [Cluster](#).



Virus Scanner

This section explains how CommuniGate Pro can use third-party products (plugins) to scan all messages (virus protection, content filtering).

The CommuniGate Pro Filter plugins provide a much more solid solution than various stand-alone SMTP-based "mail scanners":

- Stand-alone "scanner" SMTP servers usually implement only the basic SMTP functions. Since all SMTP connections have to be established to those servers, and not to the CommuniGate Pro SMTP module, the CommuniGate Pro SMTP extended functionality becomes unavailable to users and other SMTP servers.
- Stand-alone "scanner" SMTP servers usually provide much weaker performance and reliability compared to the CommuniGate Pro Servers. When the "scanner" server goes down, the CommuniGate Pro SMTP functionality becomes unavailable, too.
- Stand-alone "scanner" SMTP servers usually cannot scan several messages simultaneously, so when a large message is being scanned, the SMTP traffic to the CommuniGate Pro Server stops.
- Stand-alone "scanner" SMTP servers cannot scan messages not submitted via SMTP. For example, messages composed using the WebUser Interface and directed to a user on the same CommuniGate Pro Server are delivered bypassing any SMTP transfer operations.

The CommuniGate Pro Plugins run alongside the CommuniGate Pro Server. They do not deal with message transfer protocols. Instead, the CommuniGate Pro Server passes them a message file right before the message is being enqueued into module queues. As a result, all messages can be scanned, not only the messages sent via a particular mail transfer protocol.

If the CommuniGate Pro [ENQUEUER](#) is configured to use several processors (threads), several messages can be scanned simultaneously. As a result, long messages that require several seconds of scanning time do not stop the message flow.

The third-party plugins usually require an additional License Key.

The following filtering programs are available now:

- McAfee Virus Scanner for CommuniGate Pro

Installing External Filter Software

External Filter Software usually consists of several files - the filter program, the libraries that program uses, and various data files (virus definition files, content specification files, etc.).

The External Filter Software should be downloaded either from the Stalker Software Web and/or FTP servers, or directly from the manufacturer servers. The downloaded archive should be unpacked and moved to the appropriate location on the CommuniGate Pro Server system. The CommuniGate Pro *base directory* is a recommended location for the External Filter Software directories.

After the software directory is stored on the CommuniGate Pro Server system, learn and remember the path to the Filter Program. If you stored the Filter Directory outside the CommuniGate Pro *base directory*, you should use the full path name.

McAfee Virus Scanner

The Filter Directory is called CGPMcAfee, the Filter Program is called CGPMcAfee (CGPMcAfee.exe for the MS Windows platforms).

See the [McAfee Plugin](#) Guide for more details.

Starting the External Filter

When the External Filter Software is installed, you should tell CommuniGate Pro to start the Filter Program and to establish a communication link with it. Open the General page in the Settings section of the WebAdmin Interface, and click the Helpers link. The Helpers page is displayed:

Content Filtering 	
Log:	Program Path:
Time-out:	Auto-Restart:

Log

Use this setting to specify the type of information the External Filtering module should put in

the Server Log. Usually you should use the Problems Log level (status change and non-fatal errors). But when you experience problems with the External Filter program, you may want to set the Log setting to Low-Level or All Info: in this case the inter-program protocol-level details will be recorded in the System Log as well.

The External Filter records in the System Log are marked with the EXTFILTER tag.

Program Path

Use this setting to specify the file name path for the External Filter program. If the External Filter Software has been installed inside the CommuniGatePro *base directory*, you can use the relative path (CGPMcAfee\CGPMcAfee.exe, for example). Otherwise, use the full path (such as D:\Programs\CGPMcAfee\CGPMcAfee.exe).

Note: always use the backslash (\) path separators if the CommuniGate Pro Server runs on one of the Microsoft Windows platforms.

Select the check box and click the Update button to start the External Filter program. If the program cannot be started, an error message appears on the Helpers page.

Time-out

Certain conditions and/or errors in the External program code can make it enter a loop and stop responding to CommuniGate Pro Server requests. If a response for any of the Server requests is not received within the specified period of time, the Server sends a termination signal to the External Program.

Auto-Restart

Certain conditions and/or errors in the External program code can crash that program. Also, the Server itself can send a termination signal to the External program if the program does not respond to requests within the specified period of time (see above).

If the Auto-Restart parameter is not set to Disabled, the CommuniGate Pro server detects the External Program termination, waits for the specified period of time, and then restarts the External Program automatically. Then it resends all pending requests to the newly started External Program and resumes normal request processing.

If the Auto-Restart parameter is set to Disabled, you need to open the Helpers WebAdmin page and click the Update button to force the Server to restart the External program.

Using the External Filter

An enabled External Filter is not used for scanning mail messages by default. To make the Filter scan all or some messages, create a Server-Wide [Rule](#).

If you want to scan all messages that are transferred with your CommuniGate Pro Server, do not specify any condition in that Rule. Alternatively, you can specify a Rule condition that checks the message size (for example), and performs message scanning only if the message size is larger than the specified limit:

Data	Operation	Parameter
Action	Parameters	

External Filters are contacted from the [ENQUEUEUR](#) threads. Since it can take several seconds to process a large message, increase the number of ENQUEUEUR processors (threads) using the Obscure page in the WebAdmin Settings section. This allows the CommuniGate Pro Server to proceed with the message enqueueing process even when a large message scan is in progress.



Alerts

The CommuniGate Pro Server can send Alert messages to its users.

Alerts are displayed to the users when they connect to [POP module](#) or when they work with their accounts using the [IMAP module](#) or the [WebUser Interface](#).

Alerts can be posted by the Server and/or Domain Administrator, and some alert messages can be generated automatically by the CommuniGate Pro Server software.

The [Event Handlers](#) can use Account Alerts to notify system administrators about certain system events.

Posting Alerts

The Server Administrator can send Alert messages to all CommuniGate Pro users. Server and Domain administrators can also send Alert messages to all CommuniGate Pro domain users.

To send an Alert Message, the administrator should follow the [Alerts](#) link either on the Domains page, or on the Domain Settings page.

The Alerts page appears and lists the already posted Alerts:

Posted Alerts	
2-Aug-2001	Server will be shut down on Aug,3 from 1:00pm till 1:30pm
22:57:13	Please check your mailer - we will enforce secure authentication starting Aug, 5th
23:53:28	The next service shut down is scheduled on Aug, 15th
23:54:10	The IMAP service is now availble

The Alerts page for a CommuniGate Pro Domain contains both Server-wide and Domain-wide alerts. The Server-wide Alerts have highlighted (bold) time stamps, they cannot be removed from this Domain Alerts page.

To post an Alert message, enter the message text in the text field and click the Post Alert button.

To remove some Alert messages, mark them using the checkboxes and click the Remove Marked button.

A domain administrator can add and remove Domain Alerts only if the `CanPostAlerts` access right is granted to the administrator account.

In a Dynamic [Cluster](#) the system maintains Server-wide and Cluster-wide Alert sets. The Server-wide Alerts are displayed to all users with the Accounts in non-Shared (Local) Domains, while the Cluster-wide Alerts are displayed to all Shared Domain Account users.

Alerts sent to an individual Account are removed as soon as they are delivered to the Account user. Old and outdated Domain-wide, Server-wide, and Cluster-wide Alerts should be explicitly removed by administrators.

Storage Quota Alerts

The Server checks the account storage quota for every connected user. If the account storage is limited, and the specified percent of that limit is already used, the Server generates an alert message for that user.

The [Local Delivery module](#) settings specify if and when the Storage Quota Alerts should be generated.

After a Storage Quota Alert is sent to the account user, the Server does not generate Storage Quota Alerts for that account for 10 minutes.

Note: if a user connects to his/her account using the [POP module](#), the Storage Quota Alert is displayed as an error message, and the user should try to connect again. If the user does not retry immediately, but makes the next connection attempt more than 10 minutes later, and the account is still over its storage quota, the Storage Quota Alert is generated again and the connection is refused

again. Instruct your POP3 users to retry immediately if they see the Storage Quota Alert messages.



SNMP Interface

The CommuniGate Pro Server internal information can be accessed via the built-in SNMP server ("agent").

The SNMP agent receives requests from SNMP clients ("managers") and either returns the information about internal states, counters, problems, or modifies the internal settings by a client request.

The [Setting up MRTG for CommuniGate Pro](#) document should help you configure the popular freeware SNMP manager.

By default, the CommuniGate Pro SNMP agent is not activated.

Configuring SNMP Agent

To configure the SNMP agent, use the WebAdmin Interface. Open the Obscure page in the Settings section and find the SNMP Agent panel:

SNMP Agent	
Port:	Log:
Local Address:	Password:
Restriction:	

Port

Use this setting to specify the UDP port the SNMP agent should use.

By default, SNMP agents are expected to receive incoming requests on the port 161.

If your server computer is already running some other SNMP agent, you may want to specify a non-standard port number here and reconfigure your SNMP Manager software to use that port number.

If you set the port number to 0, the CommuniGate Pro SNMP agent software deactivates itself.

Local Address

Use this setting to specify the local network address that SNMP agent should use to receive requests. By default, the SNMP Agent processes requests that come to any of the Server network (IP) addresses.

Restriction

Use this setting to specify the network addresses that can be used to send requests to the SNMP agent. If this setting is set to None, requests are accepted from any network (IP) address. If this setting is set to Grant, requests are accepted only if they come from the network addresses specified in the Restriction text field. If this setting is set to Deny, requests are accepted if they come from any network address except those listed in the Restriction text field.

Each line of the Restriction text field can contain either one IP address or a pair of IP addresses, separated with the minus sign and specifying an inclusive range of network addresses.

Empty lines and lines starting with the ';' or '%' symbol are ignored. A comment starting with the same symbols can be placed on any line after the IP address or IP address pair.

Log

Use this setting to specify what kind of information the SNMP agent should put in the Server Log. Usually you should use the Major or Problems (non-fatal errors) levels. But when you experience problems with the SNMP agent, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well.

The SNMP agent records in the System Log are marked with the SNMP tag. Please note that SNMP is a binary protocol, so all low-level data is presented in the hexadecimal form.

Password

Use this setting to specify the SNMP "community name". The CommuniGate Pro SNMP agent accepts only those SNMP requests that contain the proper "community name" data.

Accessing the Server MIB

The MIB (Management Information Base) is a text file describing the internal objects the SNMP agent can display, monitor, and/or modify. You need the CommuniGate Pro MIB file to properly configure the SNMP client ("manager") you want to use for server monitoring.

Different versions of the CommuniGate Pro software support different sets of internal objects, and CommuniGatePro Server generates its MIB by a user request, presenting the most current information.

To access the CommuniGate Pro MIB file, use the WebAdmin interface to access the CGatePro-MIB.txt file:

`http://yourservername:8010/CGatePro-MIB.txt`

Save this file to a monitoring workstation disk and use it to configure your SNMP manager

software.

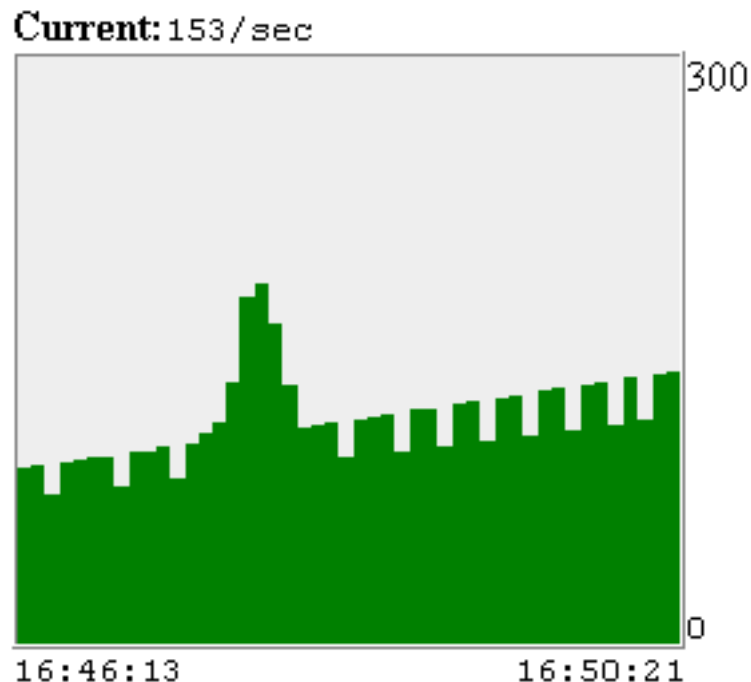
Monitoring the SNMP elements via Web

The SNMP module allows a Server Administrator to monitor the server internal parameters via Web. Open the SNMP page in the Monitors section of the [Web Admin](#) Interface. You need to have the Can Monitor Server [Server Access Right](#) to open this page.

smtpInputActive 0	smtpInputTotal 0
smtpInputJobs 0	smtpInputMessagesReceived 0
smtpInputMessageBytesReceived 0	smtpInputRecipientsAccepted 0
.....	

The page contains the list of the SNMP (MIB) elements and their current values. Each element name is a link. Click the element link to open the Element Monitor page.

The Element Monitor page contains a histogram allowing you to monitor how the element value changes over time:



There are time stamps at the bottom of the histogram (the time stamp on the right is the latest sampling time). On the right side of the histogram the graphic scale is indicated.

The histogram traces the current value of the INTEGER-type elements. For the COUNTER-type

elements, the histogram traces the difference between the last two sample values, divided by the number of seconds passed between samples.

The WebAdmin Preferences can be used to change the parameters of the SNMP Web Monitor system.

Monitoring SNMP elements via CLI/API

The [CLI/API](#) GetSNMPElement command allows a Server Administrator to monitor the server internal parameters via the CLI/API interface and via various CLI "wrappers".

Sending SNMP Traps

The SNMP module can send Traps on certain Events. See the [Events](#) section of the manual for more details. Traps can be sent to the network addresses explicitly specified in the Event Handler settings and/or to all remembered network addresses - addresses from which SNMP requests with the "Trap Password" *community name* have been received.



Events

The CommuniGate Pro Server can detect certain situations and process them as *events*, invoking *Event Handlers*. Event Handlers notify System Administrators about the event.

The Event Manager monitors the values of selected [SNMP](#) elements. It generates an *event* when the value of an INTEGER-type element crosses the specified threshold, or when the value of a COUNTER-type element has increased more than the specified limit over the specified period of time. For example, an event can be generated when there are more than 10,000 messages in the CommuniGate Pro Queue (the value of the numQueuedMessages INTEGER-type SNMP Element is over 10,000) or when the POP server has to process more than 100 connections per second (the value of the popInputJobs COUNTER-type SNMP Element increased for more than 2000 during the last 20 seconds).

Configuring Event Handlers

An Event Handler specifies how a system administrator or system operator should be notified. Several notification methods are supported, and each Event Handler can use any number of supported methods.

To configure the Event Handlers, open the Events page in the Settings realm of the CommuniGate Pro WebAdmin Interface. The list of existing Event Handlers will appear. Each Event Handler has a name and the notification method parameters:

Handler Name:

.....methods....

There is an empty Event Handler at the bottom of the page. Enter a new Handler name and click the Update button to create a new Handler.

To remove a Handler, empty its Handler Name field and click the Update button.

To rename a Handler, change the value of its Handler Name field and click the Update button.

Notification via E-mail

To send Event Notifications via E-mail, select the Send Email option:

Send Email	Subject: To: Text:
------------	--------------------------

Subject

This field specifies the Subject for E-mail messages sent with this Event Handler.

To

This field specifies the recipient(s) for E-mail messages sent with this Event Handler. Multiple recipients should be separated with the comma (,) symbol.

Body

This field specifies the text for E-mail messages sent with this Event Handler.

The Subject and Body parameters can include the special symbol combinations:

^0

When an E-mail message is being composed, this combination is replaced with the name of the SNMP Element that generated the Event.

^1

When an E-mail message is being composed, this combination is replaced with the

Notification via SNMP Traps

To send Event Notifications as [SNMP](#) Traps, select the Send SNMP Trap option:

Send SNMP Trap	To: Active Monitors
----------------	------------------------

To address field

This field specifies the addresses for computers to which SNMP Traps should be sent. Multiple addresses should be separated using the comma (,) symbol. Addresses can be specified as network (IP) addresses or as DNS domain names. If you want to send SNMP Traps not to the standard UDP port 162, but to a different port, specify the port number after the address, using the colon (:)

symbol.

To *Active Monitors*

If this option is selected, the SNMP Trap is sent to all "Active SNMP Monitor" computers - all computers that have recently sent requests to the CommuniGate Pro [SNMP](#) module using the Trap Password "community name".

Notification via Account Alerts

To send Event Notifications as an [Account Alert](#), select the Send Account Alert option:

Send Account Alert	To: Text:
--------------------	--------------

To

This field specifies the names of CommuniGate Pro Accounts the Alert should be sent to. Multiple names should be separated using the comma (,) symbol.

Text

This field specifies the text of the Alert. This text can contain the same special symbol combinations used in E-mail Notifications (see above).

Specifying Events

You can specify an Event by setting a "threshold" for some [SNMP Element](#).

For an INTEGER-type Element you specify the threshold as an integer value: if the Element value becomes larger than the threshold value, an Event is generated.

For a COUNTER-type Element you specify the threshold as an integer value and a time period: if the Element value has increased for more than the threshold value during the specified period of time, an Event is generated.

Use a Web browser to open the Events page in the Settings realm of the WebAdmin Interface. Click the Elements link to specify Events:

Element	Handler	Threshold
smtpInputActive	
smtpInputTotal		
smtpInputJobs		in
smtpInputMessagesReceived		in
	

The Handler field specifies which Event Handler should be used to process this Event.

To remove an Event, reset its Threshold value (set it to ---).

If you want to disable an Event without removing its threshold value, reset its Handler field (set it to ---).



Dialup

The CommuniGate Pro Server can work on a site that only has a dial-up Internet connection. It allows LAN users to access their mail and to submit messages without generating any Internet TCP traffic, and it allows the System Administrator to specify the Schedule for Internet (dialup) TCP/IP activity.

Mail Receiving

Dial-up systems are not connected to the Internet all the time. As a result, most of the time other systems cannot send mail directly to your dial-up server. You can use three methods to receive incoming mail:

- store all mail in a Unified Domain-Wide Account on your ISP server, and retrieve it from there periodically, using the [RPOP](#) module.
- specify your ISP mail server as your mail backup server, and use the SMTP *Remote Queue Starting (ETRN)* feature to retrieve mail using the [SMTP](#) module.
- specify your ISP mail server as your mail server (main MX), and use the SMTP *ATRN* feature to retrieve mail using the [SMTP](#) module. The ISP mail server should be able to handle the ATRN requests. If the ISP server is a CommuniGate Pro server, too, it should be configured to hold mail for your domain and to accept [ATRN](#) commands from your server.

Your Server should have a *static IP* address to be able to receive mail via SMTP (using ETRN). If your ISP assigns you an IP address dynamically, and each time your Server can get a different IP address, retrieving mail using the SMTP ATRN module or the RPOP module are the only choices.

TCP Activity Schedule

The Server Administrator can specify when and how often the Server is allowed to generate outgoing TCP/IP traffic. This helps to limit the time your Internet dial-up link is up.

The TCP Activity Schedule is checked within the [SMTP](#) and [RPOP](#) modules and can be used to limit their activities.

Use a Web browser to open the TCP Activity Schedule page. Open the General page in the Settings realm of the WebAdmin Interface, and then follow the Schedule link.

Log:

Day(s) of Week	When	Pause
	-	
	-	
	-	

Log Level

Use this setting to specify what kind of information the TCP Activity Schedule component should put in the Server Log. Usually you should use the `Major` (session starts) levels. But when you experience problems with the TCP Activity Schedule component, you may want to set the Log Level setting to `Low-Level` or `All Info`: in this case the schedule calls and schedule processing details will be recorded in the System Log as well.

The TCP Activity Schedule component System Log records are marked with the TCP tag.

Day of Week

This setting specifies the week days when this TCP Scheduler element should be used.

When

This setting specifies the time period when outgoing TCP Activity is allowed.

If the first time setting is larger than the second one, it specifies an "over-the-midnight" time period: `19:30 - 07:30` means from 19:30 till midnight and from midnight till 7:30 in the morning.

Pause

This setting specifies the minimal time interval between successive outgoing "TCP sessions".

You can remove elements from the Schedule by settings the Day of Week option to `Never`, and you add elements to the Schedule by changing the Day of Week option value in the last dummy (`Never`) element.

Serving LAN Clients

Your LAN client mailers can generate outgoing Internet activity when they submit messages via SMTP, and this can force your dial-up link to go up every time they send a message. To avoid this:

- make sure that your own CommuniGate Pro Server, not the ISP mail server is specified in the client mailer settings as the "outgoing mail server";
- make sure that IP addresses of all your LAN Clients are included into the Client Hosts list;
- make sure that SMTP module [Verify Return-Path for](#) option is set to `non-clients` or `nobody`.



Command Line Interface

The CommuniGate Pro Server provides a Command Line Interface (CLI) for server administrating. This interface can be used as an alternative for the standard Web Administrator interface.

CLI can also be used as the Application Program Interface (API), so the server can be managed via scripts and other programs that issue the CLI commands to the server.

The CommuniGate Pro Server provides several methods to access its CLI.

The [CommuniGate Pro Perl Interface](#) document contains the set of the [Perl language](#) utilities that allow a Perl script to access the CommuniGate Pro CLI API. The document also contains links to several useful sample Perl scripts (automated account registration and removal, etc.).

The [CommuniGate Pro Java Interface](#) document contains the set of the [Java language](#) classes that allow a Java program to access the CommuniGate Pro CLI API. The document also contains links to several sample Java programs.

Administrating the Server via the PWD module

The CommuniGate Pro Server CLI is available as an extension to the [PWD](#) protocol.

As soon as a PWD user is authenticated, the CLI commands are accepted. For each CLI command the server checks the access rights of the authenticated user.

If a command produces some data, the data is sent after the protocol line with the positive response. The CR-LF combination is sent after the data.

Here is a sample PWD session with CLI commands:

```
C: telnet servername.com 106
S: 200 CommuniGate Pro at mail.stalker.com PWD Server 3.5 ready
C: USER postmaster
```

```
S: 300 please send the PASS
C: PASS postmasterpassword
S: 200 login OK
C: CreateAccount "user1"
S: 200 OK
C: CreateAccount "user1"
S: 501 Account with this name already exists
C: RenameAccount "user1" into "user2"
S: 200 OK
C: CreateDomain "client1.com"
S: 200 OK
C: CreateAccount "user1@client1.com" TextMailbox
S: 200 OK
C: QUIT
S: 200 CommuniGate Pro PWD connection closed
```

CLI Syntax

The CommuniGate Pro CLI uses the [Dictionary Format](#) to parse commands and to format the output results.

Note: These Dictionary format syntax rules allow you to specify a string without the quotation marks if the string contains alphanumerical symbols only. You should use the quotation marks if a string contains the dot (.), comma (,), and other non-alphanumerical symbols.

In spite of the fact that the Dictionary format is multi-line, all arrays and dictionaries you specify as CLI parameters should be stored on one command line.

If a CLI command produces some output in the array or dictionary format, the output data can be presented on several lines.

Account Administration

A user should have the [Account Settings access right](#) or the [Domain Administration access right](#) to use the Account Administration CLI commands.

```
ListAccounts [ domainName ]
lstacnt [ domainName ]
```

Use this command to get the list of all accounts in the domain. The command produces output

data - a *dictionary* with the keys listing all accounts in the specified (or default) domain.

domainName : *string*

This *optional* parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

```
CreateAccount accountName [accountType] [ external ] [settings]  
cracnt accountName [accountType] [settings]
```

Use this command to create new accounts.

accountName : *string*

This parameter specifies the name for the new account.

The name can contain the @sign followed by the domain name, in this case the account is created in the specified domain. If the domain name is not specified, the command applies to the administrator domain.

accountType : MultiMailbox | TextMailbox | MailDirMailbox

This optional parameter specifies the type of the account to create. If no account type is specified a MultiMailbox-type account is created.

external

This optional flag tells the system to create an account with an external (visible for legacy mailers) INBOX.

settings : *dictionary*

This optional parameter specifies the initial account settings. Account is created using the settings specified in the Account Template for the target domain. If the *settings* parameter is specified, it is used to modify the Template settings.

This command can be used by a domain administrator only if the domain administrator has the CanCreateAccounts access right.

If this command is used by a domain administrator, it will use only those account settings that the domain administrator is allowed to modify.

```
RenameAccount oldAccountName into newAccountName  
rncant oldAccountName into newAccountName
```

Use this command to rename accounts.

oldAccountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

newAccountName : *string*

This parameter specifies the new account name. The name can include the domain name (see above).

This command can be used by a domain administrator only if the domain administrator has the `CanCreateAccounts` access right.

```
DeleteAccount oldAccountName  
dlacnt oldAccountName
```

Use this command to remove accounts.

oldAccountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

This command can be used by a domain administrator only if the domain administrator has the `CanCreateAccounts` access right.

```
GetAccountSettings accountName  
gtacnt accountName
```

Use this command to get the account settings. The command produces an output - a *dictionary* with the account settings. Only the explicitly set (not the default) account settings are included into the dictionary.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

You can also specify the single asterisk sign (*) instead of an account name. This will indicate the current authenticated account.

Note: All users can send the `GetAccount` command for their own accounts.

```
UpdateAccountSettings accountName newSettings  
updacnt accountName newSettings
```

Use this command to update the account settings.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

newSettings : *dictionary*

This dictionary is used to update the account settings dictionary. It does not have to contain all settings data, the omitted settings will be left unmodified. If a new setting value is specified as the string `default`, the account setting value is removed, so the default account setting value will be used.

If this command is used by a domain administrator, it will update only those account settings that the domain administrator is allowed to modify.

`GetAccountEffectiveSettings` *accountName*
`gtacnt` *accountName*

Use this command to get the effective account settings. The command produces an output - a *dictionary* with the account settings. Both the explicitly set and the default account settings are included into the dictionary.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

You can also specify the single asterisk sign (*) instead of an account name. This will indicate the current authenticated account.

Note: All users can send the `GetAccount` command for their own accounts.

`SetAccountPassword` *accountName* `To` *newPassword*

Use this command to update the account password.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

newPassword : *string*

This string is used to specify the new account password. The new password will be stored using the effective Password Encryption setting of the target account.

To use this command, the user should have the "Basic Settings" Domain Administration right for the target account domain.

`VerifyAccountPassword` *accountName* `PASSWORD` *password*

Use this command to verify the account password.

accountName : string

This parameter specifies the name of an existing account. The name can include the domain name (see above).

password : string

This string is used to specify the password to check (in the clear text format)

To use this command, the user should have any Domain Administration right for the target account domain.

`GetAccountAliases accountName`

`gtacntals accountName`

Use this command to get the list of account aliases. The command produces an output - an *array* with the account alias names.

accountName : string

This parameter specifies the name of an existing account. The name can include the domain name (see above).

`SetAccountAliases accountName newAliases`

`stacntals accountName newAliases`

Use this command to set the account aliases.

accountName : string

This parameter specifies the name of an existing account. The name can include the domain name (see above).

newAliases : array

This array should contain the account alias name strings. All old account aliases are removed.

This command can be used by a domain administrator only if the domain administrator has the `CanCreateAliases` access right.

`GetAccountRules accountName`

`gtacntrl accountName`

Use this command to get the list of account Rules. The command produces an output - an *array* of the Rules specified for the account.

accountName : string

This parameter specifies the name of an existing account. The name can include the domain

name (see above).

```
SetAccountRules accountName newRules  
stacntrl accountName newRules
```

Use this command to set the account Rules.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

newRules : *array*

This array should contain the account Rules. All old account Rules are removed.

This command can be used by a domain administrator only if the domain administrator has the CanModifyRules access right.

```
GetAccountRPOP accountName
```

Use this command to get the list of account RPOP records. The command produces an output - an *array* of the RPOP records specified for the account.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

```
SetAccountRPOP accountName newRecords
```

Use this command to set the account RPOP records.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

newRecords : *array*

This array should contain the account RPOP records. All old account RPOP records are removed.

This command can be used by a domain administrator only if the domain administrator has the CanModifyRPOP access right.

```
GetAccountRights accountName  
gtacntrghtaccountName
```

Use this command to get the array of the Server or Domain access rights granted to the specified user. The command produces output data - an *array* listing all account Server Access rights.

accountName : string

This parameter specifies the name of an existing account. The name can include the domain name.

```
GetAccountInfo accountName Key keyName  
gtacntinf accountName Key keyName
```

Use this command to get an element of the account "info" dictionary. The command produces an output - the content of the "info" element retrieved. If the element is not found, the output is an empty string - two quotation marks (" ").

accountName : string

This parameter specifies the name of an existing account. The name can include the domain name (see above). You can also specify the single asterisk sign (*) instead of an account name. This will indicate the current authenticated account.

keyName : string

This parameter specifies the name of the requested "info" element. Note that when accounts "info" data are stored in .info dictionary files, the "info" elements have dictionary names starting with the hash sign. You should NOT include the hash sign into the keyName parameter of the GetAccountInfo command.

Sample:

```
GetAccountInfo "user1@domain1.com" Key LastLogin
```

Note: the "info" element names are case-sensitive.

Note: All users can use the GetAccountInfo command to retrieve elements from their own account "info" data.

```
GetWebUser accountName  
gtwusr accountName
```

Use this command to get the account WebUser Settings. The command produces an output - a *dictionary* with all user (account) WebUser Settings.

accountName : string

This parameter specifies the name of an existing account. The name can include the domain name (see above).

```
SetWebUser accountName newSettings
```

`stwusr accountName newSettings`

Use this command to set the account WebUser Settings.

accountName : string

This parameter specifies the name of an existing account. The name can include the domain name (see above).

newSettings : dictionary

This dictionary should contain the new account WebUser Settings. All old account WebUser Settings are removed.

Group Administration

A user should have the [Can Modify All Domains and Account Settings](#) access right or the [Domain Administration access right](#) to use the Groups Administration CLI commands.

`ListGroups [domainName]`

Use this command to get the list of all Groups in the Domain. The command produces output data - an *array* with the names of all Groups in the specified (or default) domain.

domainName : string

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

`CreateGroup groupName [settings]`

Use this command to create new groups.

groupName : string

This parameter specifies the name for the new group.

The name can contain the @sign followed by the domain name, in this case the group is created in the specified domain. If the domain name is not specified, the command applies to the administrator domain.

settings : dictionary

This optional parameter specifies the initial group settings and the members list.

This command can be used by a domain administrator only if the domain administrator has the `CanCreateGroups` access right.

RenameGroup *oldGroupName* into *newGroupName*

Use this command to rename groups.

oldGroupName : *string*

This parameter specifies the name of an existing group. The name can include the domain name (see above).

newGroupName : *string*

This parameter specifies the new group name. The name can include the domain name (see above).

This command can be used by a domain administrator only if the domain administrator has the CanCreateGroups access right.

DeleteGroup *groupName*

Use this command to remove groups.

groupName : *string*

This parameter specifies the name of an existing group. The name can include the domain name (see above).

This command can be used by a domain administrator only if the domain administrator has the CanCreateGroups access right.

GetGroup *groupName*

Use this command to get the group settings. The command produces an output - a *dictionary* with the group settings and members.

groupName : *string*

This parameter specifies the name of an existing group. The name can include the domain name (see above).

SetGroup *groupName* *newSettings*

Use this command to set the group settings.

groupName : *string*

This parameter specifies the name of an existing group. The name can include the domain name (see above).

newSettings : *dictionary*

This dictionary is used to replace the group settings dictionary.

Forwarder Administration

A user should have the [Can Modify All Domains and Account Settings](#) access right or the [Domain Administration access right](#) to use the Forwarders Administration CLI commands.

`ListForwarders [domainName]`

Use this command to get the list of all Forwarders in the Domain. The command produces output data - an *array* with the names of all Forwarders in the specified (or default) domain.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

`CreateForwarder forwarderName TO address`

Use this command to create new forwarders.

forwarderName : *string*

This parameter specifies the name for the new forwarder.

The name can contain the @sign followed by the domain name, in this case the forwarder is created in the specified domain. If the domain name is not specified, the command applies to the administrator domain.

address : *string*

This parameter specifies the E-mail address the forwarder should reroute mail to.

This command can be used by a domain administrator only if the domain administrator has the `CanCreateForwarders` access right.

`DeleteForwarder forwarderName`

Use this command to remove forwarders.

forwarderName : *string*

This parameter specifies the name of an existing forwarder. The name can include the domain name (see above).

This command can be used by a domain administrator only if the domain administrator has the `CanCreateForwarders` access right.

`GetForwarder forwarderName`

Use this command to get the forwarder address. The command produces an output - a *string* with the E-mail address this forwarder reroutes all mail to.

forwarderName : string

This parameter specifies the name of an existing forwarder. The name can include the domain name (see above).

Domain Administration

A user should have the [Can Modify All Domains and Account Settings](#) access right or the [Domain Administration access right](#) to use the Domain Administration CLI commands.

`GetDomainSettings [domainName]`
`gtdmn [domainName]`

Use this command to get the domain settings. The command produces an output - a *dictionary* with the `domainName` settings. Only the explicitly set (not the default) settings are included into that dictionary.

domainName : string

This optional parameter specifies the name of an existing domain.

`GetDomainEffectiveSettings [domainName]`

Use this command to get the domain settings. The command produces an output - a *dictionary* with the `domainName` settings. Both the explicitly set and the default settings are included into that dictionary.

domainName : string

This optional parameter specifies the name of an existing domain.

`UpdateDomainSettings [domainName] newSettings`
`upddmn [domainName] newSettings`

Use this command to update the Domain settings.

domainName : *string*

This optional parameter specifies the name of an existing domain.

newSettings : *dictionary*

This dictionary is used to update the domain settings dictionary. It does not have to contain all settings data, the omitted settings will be left unmodified. If a new setting value is specified as the string `default`, the domain setting value is removed, so the default domain settings value will be used.

If this command is used by a domain administrator, it will update only those Domain Settings that this domain administrator is allowed to modify.

```
GetAccountDefaults [ domainName ]
```

```
gtacndfl [ domainName ]
```

Use this command to get the default account settings for the specified domain. The command produces an output - a *dictionary* with the default settings.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

```
UpdateAccountDefaults [ domainName ] newSettings
```

```
updacndfl [ domainName ] newSettings
```

Use this command to modify the Default Account settings for the specified domain.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

newSettings : *dictionary*

This dictionary is used to modify the domain Default Account settings. The dictionary does not have to contain all settings data, the omitted settings will be left unmodified. If a new setting value is specified as the string `default`, the setting value is removed, so the global Server Default Account Settings will be used.

If this command is used by a domain administrator, it will update only those Default Account settings that the domain administrator is allowed to modify.

```
GetWebUserDefaults [ domainName ]
```



```
gtwusrdf1 [ domainName ]
```

Use this command to get the default account WebUser Interface settings for the specified domain. The command produces an output - a *dictionary* with the default settings.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

```
SetWebUserDefaults [ domainName ] newSettings  
stwusrdf1 [ domainName ] newSettings
```

Use this command to change the Default WebUser Interface settings for the specified domain.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

newSettings : *dictionary*

This dictionary is used to replace the domain Default WebUser Interface settings. All old Default WebUser Interface settings are removed.

```
GetAccountTemplate [ domainName ]  
gtacntmp [ domainName ]
```

Use this command to get the account template settings. The command produces an output - a *dictionary* with the template settings.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

```
UpdateAccountTemplate [ domainName ] newSettings  
updacntmp [ domainName ] newSettings
```

Use this command to modify the account template settings.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

newSettings : *dictionary*

This dictionary is used to modify the domain Account Template. All new accounts in the specified domain will be created with the template settings. The dictionary does not have to

contain all settings data, the omitted settings will be left unmodified. If a new setting value is specified as the string `default`, the Template setting value is removed.

If this command is used by a domain administrator, it will update only those Template settings that the domain administrator is allowed to modify.

```
GetDomainAliases domainName  
gtdmnals domainName
```

Use this command to get the list of domain aliases. The command produces an output - an *array* with the domain alias names.

domainName : *string*

This parameter specifies the name of an existing domain.

The following commands are available for the System Administrator only:

```
ListDomains  
lstdmn
```

Use this command to get the list of domains. The command produces output data - an *array* with the names of all server domains.

```
MainDomainName
```

Use this command to get the name of the Main Domain. The command produces output data - a *string* with the Main Domain name.

```
GetDomainDefaults  
gtdmndfl
```

Use this command to get the server-wide default Domain Settings. The command produces an output - a *dictionary* with the default Domain Settings.

```
UpdateDomainDefaults newSettings  
upddmndfl newSettings
```

Use this command to change the server-wide default Domain settings.

newSettings : *dictionary*

This dictionary is used to update the default Domain settings dictionary. It does not have to

contain all settings data, the omitted settings will be left unmodified.

`SetDomainDefaults newSettings`

Use this command to change the server-wide default Domain settings.

newSettings : *dictionary*

This dictionary is used to replace the server-wide default Domain settings dictionary.

`GetClusterDomainDefaults`

`UpdateClusterDomainDefaults newSettings`

`SetClusterDomainDefaults newSettings`

These commands are available in the Dynamic Cluster only.

Use these commands instead of the [Get|Update|Set]DomainDefaults commands to work with the cluster-wide default Domain Settings.

`GetAllAccountsDefaults`

`gtalacndfl`

Use this command to get the server-wide Default Account settings. The command produces an output - a *dictionary* with the global default account settings.

`UpdateAllAccountsDefaults newSettings`

`updalacndfl newSettings`

Use this command to update the server-wide Default Account settings.

newSettings : *dictionary*

This dictionary is used to update the Default Account settings dictionary. It does not have to contain all settings data, the omitted settings will be left unmodified.

`SetAllAccountsDefaults newSettings`

Use this command to set the server-wide Default Account settings.

newSettings : *dictionary*

This dictionary is used to replace the server-wide Default Account settings dictionary.

`GetClusterAccountDefaults`

`UpdateClusterAccountDefaults newSettings`

`SetClusterAccountDefaults newSettings`

These commands are available in the Dynamic Cluster only.

Use these commands instead of the `[Get|Update|Set]AllAccountsDefaults` commands to work with the cluster-wide Default Account settings.

`CreateDomain domainName [settings]`
`crdmn domainName [settings]`

Use this command to create a new secondary domain.

domainName : *string*

This parameter specifies the domain name to create.

settings : *dictionary*

This optional parameter specifies the domain settings.

`RenameDomain oldDomainName into newDomainName`
`rndmn oldDomainName newDomainName`

Use this command to rename a domain.

domainName : *string*

This parameter specifies the name of an existing secondary domain.

newDomainName : *string*

This parameter specifies the new domain name.

`DeleteDomain oldDomainName [force]`
`dldmn oldDomainName [force]`

Use this command to remove a domain.

domainName : *string*

This parameter specifies the name of the domain to be removed.

force

This optional parameter specifies that the domain should be removed even if it is not empty.
All domain accounts and mailing lists will be removed.

`CreateSharedDomain domainName [settings]`

Use this command to create a new shared secondary domain in a Dynamic Cluster.

domainName : *string*

This parameter specifies the domain name to create.

settings : *dictionary*

This optional parameter specifies the domain settings.

CreateDirectoryDomain *domainName* [*settings*]

Use this command to create a new directory-based domain.

domainName : *string*

This parameter specifies the domain name to create.

settings : *dictionary*

This optional parameter specifies the domain settings.

This operation is allowed only when the Directory-based Domains are enabled.

ReloadDirectoryDomains

Use this command to tell the server to scan the Domains Directory subtree so it can find all additional Directory-based Domains created directly in the Directory, bypassing the CommuniGatePro Server.

This operation is allowed only when the Directory-based Domains are enabled.

SetDomainAliases *domainName* *newAliases*

stdmnals *domainName* *newAliases*

Use this command to set the domain aliases.

domainName : *string*

This parameter specifies the name of an existing domain.

newAliases : *array*

This array should contain the domain alias name strings. All old domain aliases are removed.

GetDirectoryIntegration

Use this command to get the server-wide Directory Integration settings. The command produces an output - a *dictionary* with the Directory Integration settings.

SetDirectoryIntegration *newSettings*

Use this command to set the server-wide Directory Integration settings.

newSettings : dictionary

This dictionary is used to replace the server-wide Directory Integration settings dictionary.

`GetClusterDirectoryIntegration`

`SetClusterDirectoryIntegration newSettings`

These commands are available in the Dynamic Cluster only.

Use these commands instead of the [Get|Set]DirectoryIntegration commands to work with the cluster-wide Directory Integration settings.

`SetDomainSettings domainName newSettings`

`upddmn domainName newSettings`

Use this command to change the Domain settings.

domainName : string

This parameter specifies the name of an existing domain.

newSettings : dictionary

This dictionary is used to replace the domain settings dictionary. All old domain settings are removed.

`SetAccountSettings accountName newSettings`

`stacnt accountName newSettings`

Use this command to change the account settings.

accountName : string

This parameter specifies the name of an existing account.

newSettings : dictionary

This dictionary is used to replace the account settings dictionary. All old account settings are removed.

`SetAccountDefaults [domainName] newSettings`

`stacndfl [domainName] newSettings`

Use this command to change the Default Account settings for the specified domain.

domainName : string

This parameter specifies the domain name.

newSettings : dictionary

This dictionary is used to replace the domain Default Account settings. All old Account

Default settings are removed.

```
SetAccountTemplate [ domainName ] newSettings  
stacntmp [ domainName ] newSettings
```

Use this command to change the account template settings.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

newSettings : *dictionary*

This dictionary is used to update the domain Account Template. All new accounts in the specified domain will be created with the template settings. All old Account Template settings are removed.

```
GetAccountLocation accountName
```

Use this command to get the account file directory path (for multi-mailbox accounts) or the account INBOX mailbox path (for single-mailbox accounts). The command produces an output - a *string* with the account file path. The path is relative to the account domain directory.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name (see above).

Mailbox Administration

A user should be the mailbox owner, or should have the [Can Modify All Domains and Account Settings](#) access right or the CanAccessMailboxes [Domain Administration](#) access right to use the Mailbox Administration CLI commands.

```
LISTMAILBOXES accountName [ FILTER filter] [ AUTH authAccountName ]  
lstmbx accountName [ FILTER filter] [ AUTH authAccountName ]
```

Use this command to get the list of account mailboxes. The command produces an output - a *dictionary*.

each dictionary key specifies a mailbox name;

if the *authAccountName* user is not specified or if the specified user has the `Select` access right for this mailbox, the key value contains a *dictionary* with mailbox information;

if the specified `authAccountName` does not have the `Select` access right, the key value contains an empty *array*;
 if there is a 'mailbox folder' with the dictionary key, but there is no 'regular' mailbox with that name, the key value is an empty *array*;
 if there is a 'mailbox folder' with the dictionary key, and there is also a 'regular' mailbox with that name, the key value is an *array* with one element - the information for the 'regular' mailbox (either a dictionary or an empty array).

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

filter : *string*

This optional parameter specifies the filter string to apply to account mailboxes. The filter can use the same wildcard symbols "*" and "%" as the IMAP LIST command. If the filter is not specified, the filter string "*" is assumed, and all account mailboxes are returned.

authAccountName : *string*

This optional parameter specifies the name of an account on whose behalf the LIST operation should be executed. If this name is specified, the output includes only those mailboxes for which the specified account has the `Lookup` mailbox access right.

```
CREATEMAILBOX accountName MAILBOX mailboxName [ AUTH
authAccountName ]
```

Use this command to create a mailbox in the specified account.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

mailboxName : *string*

This parameter specifies the name for the new mailbox.

authaccountname : *string*

This optional parameter specifies the name of an account on whose behalf the operation should be executed. If this name is specified, the mailbox is created only if the specified account has the `Create` access right for the 'outer' mailbox (this means that an account should have the `Create` access right for the `Archive` mailbox in order to create the `Archive/March` mailbox).

```
DELETEMAILBOX accountName MAILBOX mailboxName [ AUTH
authAccountName ]
```

```
DELETEMAILBOX accountName MAILBOXES mailboxName [ AUTH
authAccountName ]
```


Use this command to remove a mailbox from the specified account. If the keyword MAILBOXES is used, all nested mailboxes (submailboxes) are deleted, too.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

mailboxName : *string*

This parameter specifies the name of the mailbox to be deleted.

authaccountname : *string*

This optional parameter specifies the name of an account on whose behalf the operation should be executed. If this name is specified, the mailbox is deleted only if the specified account has the Create access right for the 'outer' mailbox (this means that an account should have the Create access right for the Archive mailbox in order to delete the Archive/March mailbox), and the specified account should have the DELETE right for the specified mailbox.

```
RENAMEMAILBOX accountName MAILBOX mailboxName INTO newMailboxName [
AUTH authAccountName ]
RENAMEMAILBOX accountName MAILBOXES mailboxName INTO newMailboxName
[ AUTH authAccountName ]
```

Use this command to rename a mailbox in the specified account. If the keyword MAILBOXES is used, all nested mailboxes (submailboxes) are renamed, too.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

mailboxName : *string*

This parameter specifies the name of the mailbox to be renamed.

newMailboxName : *string*

This parameter specifies the new name for the mailbox.

authaccountname : *string*

This optional parameter specifies the name of an account on whose behalf the operation should be executed. If this name is specified, the mailbox is created only if the specified account has a right to perform the DELETEMAILBOX operation with the original mailbox name and the CREATEMAILBOX operation with the new mailbox name (see above).

```
GETMAILBOXINFO accountName MAILBOX mailboxName [ AUTH
authAccountName ]
```

Use this command to get the internal information about the account mailbox. The command produces an output - a *dictionary* with the mailbox internal information.

accountName : string

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

mailboxName : string

This parameter specifies the name of an existing mailbox in the specified account.

authaccountname : string

This optional parameter specifies the name of an account on whose behalf the operation should be executed. If this name is specified, the mailbox info is returned only if the specified account has the `Select` mailbox access right.

```
GETMAILBOXACL accountName MAILBOX mailboxName [ AUTH  
authAccountName ]
```

Use this command to get the access control list for the account mailbox. The command produces an output - a *dictionary* with the mailbox access elements.

accountName : string

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

mailboxName : string

This parameter specifies the name of an existing mailbox in the specified account.

authaccountname : string

This optional parameter specifies the name of an account on whose behalf the operation should be executed. If this name is specified, the ACL info is returned only if the specified account has the `Admin` access right for the specified mailbox.

```
SETMAILBOXACL accountName MAILBOX mailboxName [ AUTH  
authAccountName ] newACL
```

Use this command to modify the access control list for the account mailbox.

accountName : string

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

mailboxName : string

This parameter specifies the name of an existing mailbox in the specified account.

authaccountname : string

This optional parameter specifies the name of an account on whose behalf the operation should be executed. If this name is specified, the ACL info is updated only if the specified account has the `Admin` access right for the specified mailbox.

newACL : dictionary

This parameter specifies the access right elements to be modified. Each dictionary key specifies an *identifier*, and the key value should be a string with access right symbols. If the key value string starts with the minus ("-") symbol, access rights specified in the string are removed from the access right element. If the key value string starts with the plus ("+") symbol, access rights specified in the string are added to the access right element. In other cases, access rights specified in the string replace the set of rights in the access right element. If the access right element for the specified key did not exist, it is created. If the new access right element has empty set of access rights, the element is removed.

```
GETMAILBOXRIGHTS accountName MAILBOX mailboxName AUTH  
authAccountName
```

This command produces an output - a *string* with the effective mailbox access rights for the given *authAccountName*.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

mailboxName : *string*

This parameter specifies the name of an existing mailbox in the specified account.

authaccountname : *string*

This parameter specifies the name of an account whose effective access rights should be retrieved.

```
GETACCOUNTSUBSCRIPTION accountName
```

This command produces an output - an *array* with the list of Account "subscribed mailboxes".

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

```
SETACCOUNTSUBSCRIPTION accountName newSubscription
```

Use this command to set the Account "subscribed mailboxes" list.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

newSubscription : *array*

The list of subscribed mailboxes. Each array element should be a *string* with a mailbox name.

GETMAILBOXALIASES *accountName*

This command produces an output - a *dictionary*. Each dictionary key is the name of an existing mailbox alias, and the key value is a *string* with the name of mailbox this alias points to.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

SETMAILBOXALIASES *accountName newAliases*

Use this command to set the Account "subscribed mailboxes" list.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

newAliases : *dictionary*

The set of new mailbox aliases.

Alert Administration

A user should have the [Can Modify All Domains and Account Settings](#) access right or the CanPostAlerts [Domain Administration access right](#) to use the [Alert](#) Administration CLI commands.

GetDomainAlerts [*domainName*]

gtalrt [*domainName*]

Use this command to get the domain Alerts. The command produces an output - a *dictionary* with the domain alert strings and time stamps.

domainName : *string*

This optional parameter specifies the name of an existing domain.

SetDomainAlerts [*domainName*] *newAlerts*

stalrt [*domainName*] *newAlerts*

Use this command to change the Domain alerts.

domainName : *string*

This optional parameter specifies the name of an existing domain.

newAlerts : *dictionary*

This dictionary is used to replace the domain alert dictionary. All old domain alerts are removed.

PostDomainAlert [*domainName* ALERT] *newAlert*

Use this command to post a Domain-wide alert message.

domainName : *string*

This optional parameter specifies the name of an existing domain. If this parameter is used, it must be followed with the ALERT keyword.

newAlert : *string*

This string specifies the Alert text.

RemoveDomainAlert [*domainName* ALERT] *timeStamp*

Use this command to remove a Domain-wide alert message.

domainName : *string*

This optional parameter specifies the name of an existing domain. If this parameter is used, it must be followed with the ALERT keyword.

timeStamp : *string*

This string specifies the time stamp of the Alert message to be removed.

The following commands are available for the System Administrator only:

GetServerAlerts

Use this command to get the list of the server-wide Alerts. The command produces an output - a *dictionary* with the server alert strings and time stamps.

SetServerAlerts *newAlerts*

Use this command to change the server-wide Alerts.

newAlerts : *dictionary*

This dictionary is used to replace the server-wide Alert dictionary. All old server-wide alerts are removed.

PostServerAlert *newAlert*

Use this command to post a server-wide Alert message.

newAlert : *string*

This string specifies the Alert text.

RemoveServerAlert *timeStamp*

Use this command to remove a server-wide Alert message.

timeStamp : *string*

This string specifies the time stamp of the Alert message to be removed.

GetClusterAlerts

SetClusterAlerts *newAlerts*

PostClusterAlert *newAlert*

RemoveClusterAlert *timeStamp*

These commands are available in the Dynamic Cluster only.

Use these commands instead of the [Get|Set|Post|Remove]ServerAlert[s] commands to work with the cluster-wide Alerts.

Personal Web Site Administration

The following commands allow an authenticated user to deal with files in the account [Personal Web Site](#) area.

GETWEBFILE *accountName* FILE *fileName*

Use this command to retrieve a file from the account Personal Web Site. This command produces an output - a *array* of 2 *strings*. The first string contains the base64-encoded content of the specified file, the second string contains the file modification date (in the ACAP time format).

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

fileName : string

This parameter specifies the name of the Personal Web Site file to be retrieved.

The authenticated user should be the account owner, or should have the [Can Modify All Domains and Account Settings](#) access right or the WebSite [Domain Administration](#) access right to use the personal WebSite Administration CLI commands.

```
PUTWEBFILE accountName FILE fileName DATA encodedData
```

Use this command to store a file in the account Personal Web Site. If a Personal WebSite file with the specified name already exists, the old file is removed.

accountName : string

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

fileName : string

This parameter specifies the name for the Personal Web Site file.

encodedData : string

This parameter contains the Base64-encoded content of the Personal Web Site file.

```
RENAMEWEBFILE accountName FILE oldFileName INTO newFileName
```

Use this command to rename a file in the account Personal Web Site.

accountName : string

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

oldFileName : string

This parameter specifies the name of an existing Personal Web Site file.

newFileName : string

This parameter specifies the new name for the Personal Web Site file.

```
DELETEWEBFILE accountName FILE fileName
```

Use this command to remove a file from the account Personal Web Site.

accountName : string

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

oldFileName : *string*

This parameter specifies the name of an existing Personal Web Site file.

LISTWEBFILES *accountName* [*PATH filePath*]

Use this command to list all files in the Personal Web Site top directory or in one of its subdirectories. This command produces an output - a *dictionary*, where each key is a name of the Web Site file, and the key value is a *dictionary* for regular file and an empty *array* for subdirectories.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

filePath : *string*

This optional parameter specifies the name of the Personal Web Site subdirectory. You can omit this parameter along with the PATH keyword, in this case the command returns the list of the WebSite files in the top Web Site directory.

GETWEBFILESINFO *accountName*

Use this command to get the statistical information about all files in the Personal Web Site area. This command produces an output - an *array* with 2 *string* elements. The first element contains the total size of all Web Site files, the second element contains the number of files in the Web Site area.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

Mailing Lists Administration

A user should have the [Account Settings access right](#) or the [Domain Administration access right](#) to use the Mailing List Administration CLI commands.

ListLists [*domainName*]

Use this command to get the list of all mailing lists in the domain. The command produces output data - an *array* of strings. Each string is the name of a mailing list in the specified (or default) domain.

domainName : *string*

This optional parameter specifies the domain name.

GetDomainLists [*domainName*]

Use this command to get the list of all mailing lists in the domain. The command produces output data - an *dictionary*. Each dictionary key is the name of a mailing list in the specified (or default) domain. The key value is a numeric string with the actual number of the list subscribers ("-1" if the current number of subscribers is not known).

domainName : *string*

This optional parameter specifies the domain name.

GetAccountLists *accountName*

Use this command to get the list of all mailing lists belonging to the specified account. The command produces output data - a *dictionary*. Each dictionary key is the name of a mailing list belonging to the specified (or default) domain. The key value is a numeric string with the actual number of the list subscribers ("-1" if the current number of subscribers is not known).

accountName : *string*

This parameter specifies the list's owner account name.

CreateList *listName* for *accountName*

Use this command to create a mailing list.

listName : *string*

This parameter specifies the name of an existing mailing list. It can include the domain name. If the domain name is not specified, the user domain is used by default.

accountName : *string*

This parameter specifies the name of the mailing list owner. It should be the name of an already existing account in the mailing list domain.

RenameList *listName* into *newName*

Use this command to rename a mailing list.

listName : *string*

This parameter specifies the name of an existing mailing list. It can include the domain name. If the domain name is not specified, the user domain is used by default.

newName : string

This parameter specifies the new name for the mailing list (without the domain part).

DeleteList *listName*

Use this command to remove a mailing list.

listName : string

This parameter specifies the name of an existing mailing list. It can include the domain name. If the domain name is not specified, the user domain is used by default.

The following commands can also be used by the mailing list owner.

GetList *listName*

Use this command to retrieve list settings. The command produces an output - a *dictionary* with the *listName* mailing list settings.

listName : string

This parameter specifies the name of an existing mailing list. It can include the domain name. If the domain name is not specified, the user domain is used by default.

UpdateList *listName newSettings*

Use this command to retrieve list settings. The command produces an output - a *dictionary* with the *listName* mailing list settings.

listName : string

This parameter specifies the name of an existing mailing list. It can include the domain name. If the domain name is not specified, the user domain is used by default.

newSettings : dictionary

This dictionary is used to update the mailing list settings dictionary. It does not have to contain all settings data, the omitted settings will be left unmodified.

List *listName operation [silently] [confirm] subscriber*

Use this command to update the subscribers list.

listName : string

This parameter specifies the name of an existing mailing list. It can include the domain name.

If the domain name is not specified, the user domain is used by default.

operation : subscribe | feed | digest | index | null | banned | unsubscribe

This parameter specifies the operation (see the [LIST module](#) section for the details).

silently

This optional parameter tells the server not to send the Welcome/Bye message to the subscriber.

confirm

This optional parameter tells the server to send a confirmation request to the subscriber.

subscriber : *E-mail address*

The subscriber address. It can include the *comment part* used as the subscriber's real name.

Sample:

```
LIST MyList@mydomain.com FEED confirm "Bill Jones"
<BJones@company.com>
```

ListSubscribers *listName* [*filter* [*limit*]]

Use this command to retrieve list subscribers. The command produces an output - an *array* with subscribers' E-mail addresses.

listName : *string*

This parameter specifies the name of an existing mailing list. It can include the domain name. If the domain name is not specified, the user domain is used by default.

filter : *string*

If this optional parameter is specified, only the addresses that contain the specified string are returned.

limit : *number*

This optional parameter limits the number of subscriber addresses returned.

SetPostingMode *listName* FOR *subscriberAddress* [UNMODERATED | MODERATEALL | PROHIBITED | SPECIAL | *numberOfModerated*]

Use this command to set the posting mode for the specified subscriber.

listName : *string*

This parameter specifies the name of an existing mailing list. It can include the domain name. If the domain name is not specified, the user domain is used by default.

subscriberAddress : *string*

This parameter specifies the E-mail address of the list subscriber.

postingMode : *number*

This optional parameter limits the number of subscriber addresses returned.

The command sets the posting mode the specified subscriber. If *numberOfModerated* (a number) is specified, the posting mode set requires moderation of the first *numberOfModerated* messages from this subscriber.

Web Skins Administration

The following commands can be used to manage CommuniGate Pro [Skins](#) used for the CommuniGate Pro WebUser Interface.

A user should have the [Account Settings access right](#) or the CanModifySkins [Domain Administration access right](#) to manipulate with the domain Skins.

ListDomainSkins [*domainName*]

Use this command to list custom Domain Skins. The command produces an output - an *array* with Skin names.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain.

CreateDomainSkin [*domainName* SKIN] *skinName*

Use this command to create a custom Domain Skin.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain. If it is specified, it should be followed with the SKIN keyword.

skinName : *string*

This parameter specifies the name of the new Skin.

RenameDomainSkin [*domainName* SKIN] *skinName* INTO *newSkinName*

Use this command to rename a custom Domain Skin.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain. If it is specified, it should be followed with the SKIN keyword.

skinName : *string*

This parameter specifies the name of an existing Skin.

newSkinName : *string*

This parameter specifies the new name for the Skin.

DeleteDomainSkin [*domainName* SKIN] *skinName*

Use this command to delete a custom Domain Skin.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain. If it is specified, it should be followed with the SKIN keyword.

skinName : *string*

This parameter specifies the name of the Skin to be deleted.

ListDomainSkinFiles [*domainName* SKIN] *skinName*

Use this command to list files in a custom Domain Skin. The command produces an output - a *dictionary* with Skin file names as keys. The dictionary element values are dictionaries with file attributes.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain. If it is specified, it must be followed with the SKIN keyword.

skinName : *string*

This parameter specifies the name of an existing Domain Skin.

ReadDomainSkinFile [*domainName* SKIN] *skinName* FILE *fileName*

Use this command to read a file from a custom Domain Skin. The command produces an output - an *array*. The first array element is a string with the BASE64-encoded Skin file content, the second array element is a string with the file modification date in the ACAP date format.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain. If it is specified, it must be followed with the SKIN keyword.

skinName : *string*

This parameter specifies the name of an existing Domain Skin.

fileName : *string*

This parameter specifies the name of an existing file in the specified Domain Skin.

```
StoreDomainSkinFile [ domainName SKIN] skinName FILE fileName DATA  
fileContent
```

```
StoreDomainSkinFile [ domainName SKIN] skinName FILE fileName DELETE
```

Use this command to store a file into a custom Domain Skin, or to delete a file from a custom Domain Skin.

domainName : *string*

This optional parameter specifies the domain name. If the domain name is not specified, the command applies to the administrator domain. If it is specified, it must be followed with the SKIN keyword.

skinName : *string*

This parameter specifies the name of an existing Domain Skin.

fileName : *string*

This parameter specifies the Skin file name.

fileContent : *string*

This string contains the BASE64-encoded file content. This parameter is specified only if the DATA keyword is used.

If the DATA keyword is specified and the Skin contains a file with the same name, the old file is deleted.

The file with the specified name is removed from the Skin Cache (in the Dynamic Cluster the file is removed from Skin caches on all cluster members).

The following commands are available for the System Administrator only:

ListServerSkins

Use this command to list custom Server Skins. The command produces an output - an *array* with Skin names.

CreateServerSkin *skinName*

Use this command to create a custom Server Skin.

skinName : *string*

This parameter specifies the name of the new Skin.

RenameServerSkin *skinName* INTO *newSkinName*

Use this command to rename a custom Server Skin.

skinName : *string*

This parameter specifies the name of an existing Skin.

newSkinName : *string*

This parameter specifies the new name for the Skin.

DeleteServerSkin *skinName*

Use this command to delete a custom Server Skin.

skinName : *string*

This parameter specifies the name of the Skin to be deleted.

ListServerSkinFiles *skinName*

Use this command to list files in a custom Domain Skin. The command produces an output - a *dictionary* with Skin file names as keys. The dictionary element values are dictionaries with file attributes.

skinName : *string*

This parameter specifies the name of an existing Server Skin.

ReadServerSkinFile *skinName* FILE *fileName*

Use this command to read a file from a custom Server Skin. The command produces an output - an *array*. The first array element is a string with the BASE64-encoded Skin file content, the second array element is a string with the file modification date in the ACAP date format.

skinName : *string*

This parameter specifies the name of an existing Server Skin.

fileName : *string*

This parameter specifies the name of an existing file in the specified Server Skin.

```
StoreServerSkinFile skinName FILE fileName DATA fileContent
StoreServerSkinFile skinName FILE fileName DELETE
```

Use this command to store a file into a custom Server Skin, or to delete a file from a custom Server Skin.

skinName : *string*

This parameter specifies the name of an existing Server Skin.

fileName : *string*

This parameter specifies the Skin file name.

fileContent : *string*

This string contains the BASE64-encoded file content. This parameter is specified only if the DATA keyword is used.

If the DATA keyword is specified and the Skin contains a file with the same name, the old file is deleted.

The file with the specified name is removed from the Skin Cache (in the Dynamic Cluster the file is removed from Skin caches on all cluster members).

```
ListClusterSkins
CreateClusterSkin skinName
RenameClusterSkin skinName INTO newSkinName
DeleteClusterSkin skinName
```

These commands are available in the Dynamic Cluster only.

Use these commands instead of the [List|Create|Rename|Delete]ServerSkin[s] commands to work with the cluster-wide Skins.

```
ListClusterSkinFiles skinName
ReadClusterSkinFile skinName FILE fileName
StoreClusterSkinFile skinName FILE fileName DATA fileContent
StoreClusterSkinFile skinName FILE fileName DELETE
```

These commands are available in the Dynamic Cluster only.

Use these commands instead of the [List|Read|Store]ServerSkinFile[s] commands to

work with files in the cluster-wide Skins.

Web Interface Integration

The following commands can be used to integrate the CommuniGate Pro WebUser Interface with third-party applications.

`CreateWebUserSession` *accountName* ADDRESS *ip-address*

Use this command to create a WebUser session for the specified account. The command produces an output - a *string* that contains the WebUser Session ID. This string can be used to compose a URL that will allow the client browser to "enter" the WebUser Session. That URL can have the following format:

`http://cgateproserver:port/Session/rrrrrrrrrrrr/Mailboxes.html`
where `rrrrrrrrrrrr` is the Session ID string returned.

account : *string*

This parameter specifies the Account name.

ip-address : *string*

This parameter specifies the IP address of the client browser. If the Account has the "Fixed IP" WebUser Preference setting enabled, connections to the session will be allowed from that IP address only.

The authenticated user should have the [Can Modify All Domains and Account Settings](#) access right or the CanAccessMailboxes [Domain Administration](#) access right to create WebUser Sessions.

`GetWebUserSession` *sessionID*

Use this command to retrieve the WebUser Session data. The command produces an output - a *dictionary* with the *session dataset* (specified in the [WSSP](#) section of this manual).

sessionID : *string*

This parameter specifies the WebUser Session ID.

The authenticated user should have the [Can Modify All Domains and Account Settings](#) access right to retrieve WebUser Session data.

This operation resets the WebUser session inactivity timer.

`KillWebUserSession sessionID`

Use this command to terminate a WebUser Session.

sessionID : *string*

This parameter specifies the WebUser Session ID.

The authenticated user should have the [Can Modify All Domains and Account Settings](#) access right to retrieve WebUser Session data.

Server Settings

A user should have the [Server Settings access right](#) to use the Server Settings CLI commands.

`GetModule moduleName`

Use this command to get the module settings. The command produces an output - a *dictionary* with the module settings.

moduleName : *string*

This parameter specifies the name of a CommuniGate Pro Server module.

`SetModule moduleName newSettings`

Use this command to set the module settings.

moduleName : *string*

This parameter specifies the name of a CommuniGate Pro Server module.

newSettings : *dictionary*

This dictionary is used to set the module settings dictionary.

`UpdateModule moduleName newSettings`

Use this command to update the module settings.

moduleName : *string*

This parameter specifies the name of a CommuniGate Pro Server module.

newSettings : *dictionary*

This dictionary is used to update the module settings dictionary. It does not have to contain all settings data, the omitted settings will be left unmodified.

GetBlacklistedIPs

Use this command to retrieve the set of Blacklisted IP Addresses. The command produces an output - a (multi-line) *string* with Blacklisted IP addresses and address ranges.

GetClientIPs

Use this command to retrieve the set of Blacklisted IP Addresses. The command produces an output - a (multi-line) *string* with Client IP addresses and address ranges.

GetWhiteHoleIPs

Use this command to retrieve the set of WhiteHole IP Addresses. The command produces an output - a (multi-line) *string* with White Hole IP addresses and address ranges.

GetProtection

Use this command to retrieve the Protection settings. The command produces an output - a *dictionary* with the server Protection settings.

GetBanned

Use this command to retrieve the Banned Message Lines settings. The command produces an output - a *dictionary* with the server Banned Message Lines settings.

SetBlacklistedIPs *newAddresses*

Use this command to update the set of Blacklisted IP Addresses.

newAddresses : *string*

This (multi-line) string parameter contains the set of addresses and address ranges forming the new set of Blacklisted IP Addresses.

SetClientIPs *newAddresses*

Use this command to update the set of Client IP Addresses.

newAddresses : string

This (multi-line) string parameter contains the set of addresses and address ranges forming the new set of Client IP Addresses.

`SetWhiteHoleIPs newAddresses`

Use this command to update the set of WhiteHole IP Addresses.

newAddresses : string

This (multi-line) string parameter contains the set of addresses and address ranges forming the new set of WhiteHole Addresses.

`SetProtection newSettings`

Use this command to set the server Protection Settings.

newSettings : dictionary

New server Protection settings.

`SetBanned newSettings`

Use this command to set the server Banned Message Line Settings.

newSettings : dictionary

New server Banned settings.

`GetClusterBlacklistedIPs`

`GetClusterClientIPs`

`GetClusterWhiteHoleIPs`

`GetClusterProtection`

`GetClusterBanned`

`SetClusterBlacklistedIPs newAddresses`

`SetClusterClientIPs newAddresses`

`SetClusterWhiteHoleIPs newAddresses`

`SetClusterProtection newSettings`

Use these commands to retrieve and update the Cluster-wide IP Address lists and Protection settings.

`GetServerRules`

Use this command to read the Server-Wide Automated Mail Processing Rules. The command produces an output - an *array* of the Server Rules.

SetServerRules *newRules*

Use this command to set the Server-Wide Automated Mail Processing Rules.

newRules : *array*

An array of new Server Rules.

GetClusterRules

Use this command to read the Cluster-Wide Automated Mail Processing Rules. The command produces an output - an *array* of the Cluster Rules.

SetClusterRules *newRules*

Use this command to set the Cluster-Wide Automated Mail Processing Rules.

newRules : *array*

An array of new Cluster Rules.

GetRouterTable

Use this command to read the Router Table. The command produces an output - a (multi-line) *string* with the Router Table text.

SetRouterTable *newTable*

Use this command to set the Router Table.

newTable : *string*

A (multi-line) string containing the text of the new Router Table

Note: multiple lines should be separated with the \e symbols.

GetClusterRouterTable

SetClusterRouterTable *newTable*

These commands are the same as the GetRouterTable and SetRouterTable commands, but they deal with the Cluster-Wide Router Table.

RefreshOSData

Use this command to set make the Server re-read the IP data from the server OS: the set of the local IP addresses, and the set of the DNS addresses.

A user should have the [Server Settings access right](#) or the [Account Settings access right](#) to use the following CLI commands.

Route *address*

Use this command to get the routing for the specified address.

address : *string*

This parameter specifies the E-mail address to be processed with the CommuniGate Pro Router.

This command produces an output - an array of three strings:

module

the name of the CommuniGate Pro module the address is routed to, or SYSTEM if the address is routed to a built-in destination (like NULL).

host

the object/queue handled by the specified module: an Internet domain name for the SMTP module, a local account name for the Local Delivery module, etc.

address

the address inside the queue (E-mail address for SMTP, Real-To: address for Local Delivery, etc.)

Monitoring

A user should have the Monitoring [access right](#) to use the Server Monitoring CLI commands.
getsnmpelement 1.3.6.1.4.1.5678.2.1.1.5.15

GetSNMPElement *ObjectID*

Use this command to retrieve the current value of a server state (SNMP) element.

ObjectID : *string*

The object ID of the server state element (see the [SNMP](#) section for more details).

This command produces an output - a *string* with the server state element value.

Shutdown

Use this command to stop the CommuniGatePro Server.

Access Rights Administration

A user should have the [unlimited access right](#) to use the Access Rights Administration CLI commands.

```
SetAccountRights accountName newRights  
stacntrght accountName newRights
```

Use this command to set the account Server Access rights.

accountName : *string*

This parameter specifies the name of an existing account. The name can include the domain name.

newRights : *array*

This array should contain the Access Right codes. All old account access rights are removed.

To set access rights for an account in a secondary domain (i.e. Domain Administration Rights), the user may have only the [All Account and Domains](#) administration access right.

Statistics

The Account-Level Statistics data is collected if the Account Statistics option is enabled on the Obscure page in the Settings realm of the CommuniGate Pro WebAdmin Interface.

```
GETACCOUNTSTAT accountName [ KEY keyName ]
```

Use this command to retrieve statistics data about the specified account.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

keyName : *string*

This optional parameter specifies the name of the statistical entry to retrieve.

This command produces an output - a *string* with the specified statistical information, or (if the KEY keyword and the *keyName* parameter are not specified) a *dictionary* with all available statistical data.

If the statistical data for the specified key does not exist, an empty *string* is returned.

To use this command, the user should have the Domain Administration right for the target account domain. All users can retrieve the Account statistics data for their own accounts.

```
RESETACCOUNTSTAT accountName [ KEY keyName ]
```

Use this command to reset statistics data about the specified account.

accountName : *string*

This parameter specifies the name of an existing account. The asterisk sign (*) can be used to specify the current authenticated account.

keyName : *string*

This optional parameter specifies the name of the statistical entry to reset.

If the KEY keyword and the *keyName* parameter are not specified, all Account statistical entries are reset.

To use this command, the user should have the "Basic Settings" Domain Administration right for the target account domain.

The following Account statistics data keys are implemented:

Key Name	Value
StatReset	The date & time when the last parameterless RESETACCOUNTSTAT command was sent to this Account
MessagesReceived	The total number of messages delivered to the Account
BytesReceived	The total size of all messages delivered to the Account

GETDOMAINSTAT *domainName* [KEY *keyName*]

Use this command to retrieve statistics data about the specified domain.

domainName : *string*

This parameter specifies the name of an existing Domain. The asterisk sign (*) can be used to specify the domain of the current authenticated account.

keyName : *string*

This optional parameter specifies the name of the statistical entry to retrieve.

This command produces an output - a *string* with the specified statistical information, or (if the KEY keyword and the *keyName* parameter are not specified) a *dictionary* with all available statistical data.

To use this command, the user should have the Domain Administration right for the target Domain.

RESETDOMAINSTAT *domainName* [KEY *keyName*]

Use this command to reset statistics data about the specified domain.

domainName : *string*

This parameter specifies the name of an existing Domain. The asterisk sign (*) can be used to specify the domain of the current authenticated account.

keyName : *string*

This optional parameter specifies the name of the statistical entry to reset.

If the KEY keyword and the *keyName* parameter are not specified, all Domain statistical entries are reset.

To use this command, the user should have the "Basic Settings" Domain Administration right for the target Domain.

The following Domain statistics data keys are implemented:

Key Name	Value
StatReset	The date & time when the last parameterless RESETDOMAINSTAT command was sent to this Domains
MessagesReceived	The total number of messages delivered to the Domain
BytesReceived	The total size of all messages delivered to the Domain

Miscellaneous Commands

`WRITELOG logLevel logRecord`

Use this command to store a record into the Server Log.

logLevel : *number*

This parameter specifies the record log level.

logRecord : *string*

This parameter specifies the string to be placed into the Server Log.

Log records generated with this command have the `SYSTEM` prefix.

To use this command, the user should have the "Can Monitor" Server Administration right.

`RELEASESMTPQUEUE queueName`

Use this command to release an SMTP queue.

queueName : *string*

This parameter specifies the queue (domain) name to release.

In a Dynamic Cluster environment, this command releases the specified SMTP queue on all servers.

To use this command, the user should have the "Can Monitor" Server Administration right.

Web Interface Tuning

Note: These command deal with the old WebUser Interface. The Interface is depreciated, and it will be removed from the CommuniGate Pro system. Do not build new applications using these commands, use the [Web Skin](#) administration commands instead.

A user should have the [Account Settings access right](#) or the [Domain Administration access right](#) to tune the domain Web Interfaces.

`GetWebUserInterface domainName FILE path`

Use this command to read a custom domain WebUser Interface file. The command produces

output data - a strings that contains the base64-encoded content of the interface file.

domainName : *string*

This parameter specifies the domain name. The asterisk sign (*) can be used to specify the domain of the currently authenticated user.

path : *string*

This parameter specifies the name of the WebUser Interface file to retrieve.

`PutWebUserInterface` *domainName* `FILE` *path* `DATA` *filedata*

Use this command to upload a custom domain WebUser Interface file.

domainName : *string*

This parameter specifies the domain name. The asterisk sign (*) can be used to specify the domain of the currently authenticated user.

path : *string*

This parameter specifies the name of the WebUser Interface file to upload.

fileData : *string*

This parameter contains base64-encoded content of the Interface file.

`DeleteWebUserInterface` *domainName* `FILE` *path*

Use this command to delete a custom domain WebUser Interface file.

domainName : *string*

This parameter specifies the domain name. The asterisk sign (*) can be used to specify the domain of the currently authenticated user.

path : *string*

This parameter specifies the name of the WebUser Interface file to delete.

`ListWebUserInterface` *domainName* [`PATH` *path*]

Use this command to list all custom domain WebUser Interface files in the specified directory. The command produces output data - a dictionary where the keys specify file names, and the key values are file attributes.

domainName : *string*

This parameter specifies the domain name. The asterisk sign (*) can be used to specify the domain of the currently authenticated user.

path : *string*

This optional parameter specifies the name of the WebUser sub-directory.

ClearWebUserCache *domainName*

Use this command to clear the internal WebUser Interface file cache.

domainName : *string*

This parameter specifies the domain name. The asterisk sign (*) can be used to specify the domain of the currently authenticated user.



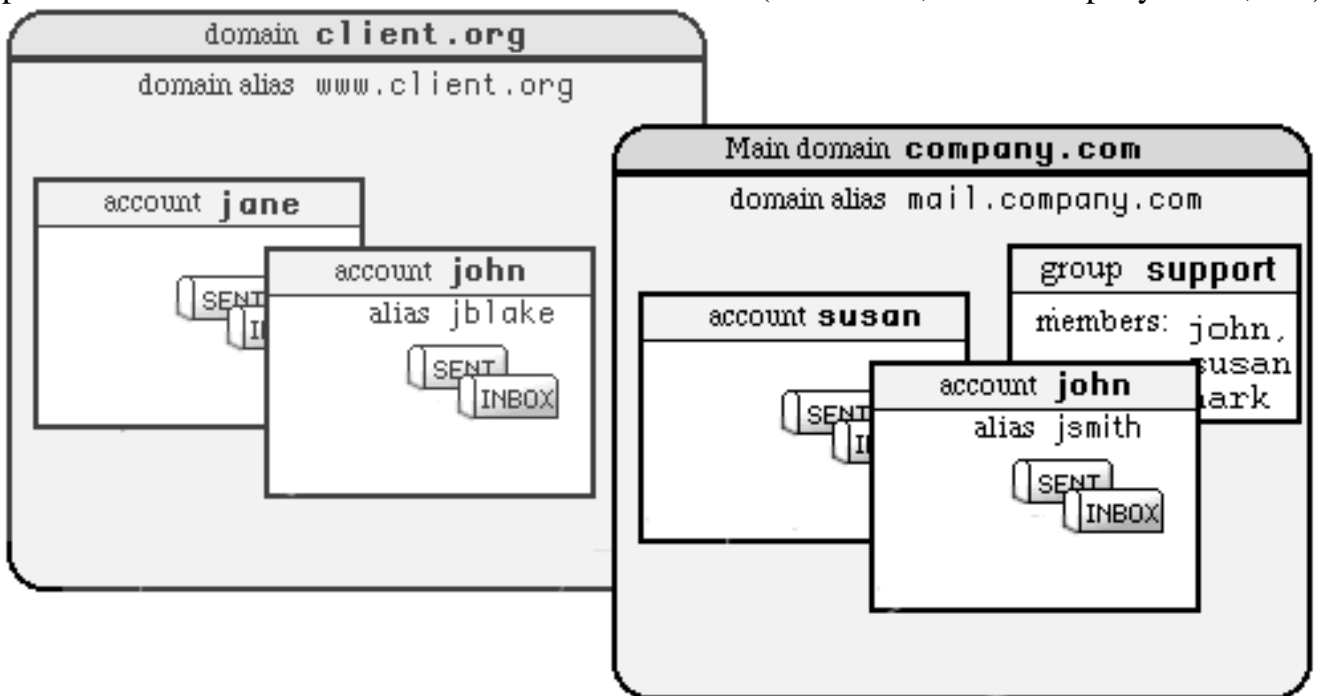
CommuniGate Pro Objects

CommuniGate Pro has a hierarchy of *objects* it serves on each installation. On the topmost level there is a set of Domains, and each Domain contains accounts, groups, mailing lists and forwarders. Each account contains one or several mailboxes. Each mailbox contains some number of E-mail messages.

Besides these basic objects, CommuniGate Pro supports supplementary objects: accounts can contain Web Sites and Preference Data, domains can contain certificates and WebUser Interface files, etc.

Domains

Domains are the CommuniGate Pro objects that contain other objects: accounts, mailing lists, groups and forwarders. Each domain has a domain name (client.com, www.company1.com, etc.):



While each CommuniGate Pro Domain has its domain name, it is not necessary to create a separate CommuniGate Pro Domain for each domain name you want to serve. CommuniGate Pro Domains

can have domain aliases, that allow you to assign several names to the same CommuniGate Pro Domain. For example, the CommuniGate Pro Domain `company.com` may have a domain alias `mail.company.com`. In this case all references to the domain name `mail.company.com` will be processed as references to the `company.com` CommuniGate Pro Domain.

There is a special CommuniGate Pro Domain, called the *Main Domain*. Other CommuniGate Pro Domains are called *secondary domains*. The Main Domain is created as soon as the Server is installed, and its name is specified in the [General Settings](#). If your Server should serve only one Domain, the Main Domain is all you need and there is no need to create secondary domains. The Main Domain name is used as the Server Name.

Each CommuniGate Pro Domain has its own settings and a set of objects - Accounts, Groups, Forwarders, Aliases, Mailing Lists.

See the [Domains](#) section for more information about CommuniGate Pro Domains.

Accounts

An account is the basic *service unit*: every user served with a CommuniGate Pro server should have an account on that server.

Each account is protected with a password, so only the account owner (and, optionally, system and domain administrators) can have unrestricted access to account data.

When the CommuniGate Pro Server is installed, the `postmaster` account is automatically created in the main domain. The Master (unlimited) [access right](#) is granted to that account.

Accounts are created inside CommuniGate Pro *domains*.

Each CommuniGate Pro domain has its own set of accounts. Accounts should have unique names within their domain, but two accounts in different domains can have the same name.

Account E-mail address is `accountname@domainname` address where *accountname* is a name of a CommuniGate Pro account, and *domainname* is the name of the CommuniGate Pro domain in which this account is created. Messages directed to this account address are delivered to the account using the [Local Delivery](#) module.

An Account may have several names (for example, `john.smith` and `jsmith`). An administrator can create [account aliases](#) to assign several names to one Account.

Each CommuniGate Pro Account has its own settings and a set of [Mailboxes](#).

Each CommuniGate Pro Account has its own [Personal Web Site](#).

Accounts can also store additional information and data. See the [Account Data](#) section for the details.

See the [Accounts](#) section for more information about CommuniGate Pro Accounts.

Groups

CommuniGate Pro Domains can contain Groups. Groups are essentially lists of account names and/or other groups and sending a message to a group results in sending it to all group members.

See the [Groups](#) section for more information about CommuniGate Pro groups.

Forwarders

CommuniGate Pro Domains can contain Forwarders. Each forwarder has a name and contains an E-mail address for redirection. If mail is sent to *name@domain.com* where *name* is a forwarder object in the domain.com CommuniGate Pro domain, then mail is re-routed to the E-mail address specified in that forwarder object.

Group and Forwarder Objects are different:

- a forwarder can contain only one address, while a group can contain several addresses;
- a forwarder works on the [Router](#) level, substituting its own address with the specified address, while a group object actually processes a message sent to the group and generates a new message copy to be sent to the group members.

See the [Forwarders](#) section for more information about CommuniGate Pro Forwarders.

Mailing Lists

CommuniGate Pro Domains can contain Mailing Lists. Each Mailing List has a name and it always belongs to some account in the same domain - the Mailing List owner.

Mailing list contains a list of subscribers, and it maintains several mailboxes in the list owner account. Those mailboxes are used to store and archive postings, generate digests, store subscription requests and error reports.

Groups and Mailing Lists are different:

- groups are designed for a small number of members, mailing lists are designed to handle several hundred thousand subscribers per list;
- groups are designed mostly for local members (accounts) and if the subscriber account is renamed or removed, it is also renamed in or removed from all the groups in its domain;
- mailing lists provide a lot of features beyond basic mail distribution: automatic subscribing, bounce processing, archiving and digesting, browsing, posting policies, moderating, etc.

See the [LIST](#) section for more information about CommuniGate Pro Mailing Lists.

Mailboxes

A mailbox is the basic *storage unit*: messages sent to accounts are stored in account mailboxes. Messages can be read from mailboxes, they can be marked with various flags, they can be copied to other mailboxes, and they can be removed from mailboxes.

Each account can have one or several mailboxes. The INBOX mailbox is special: it exists in every account, and it is used to store incoming messages. The INBOX mailbox is created automatically when an account is created. A user cannot remove the INBOX mailbox, but a user can rename it. In this case, a new empty INBOX is immediately created.

CommuniGate Pro allows administrators to create *single-mailbox* accounts. These accounts contain only the INBOX mailbox.

The CommuniGate Pro Server provides [access](#) to account mailboxes via POP, IMAP, WebUser Interface and other modules.

CommuniGate Pro mailboxes can have various formats. Administrators and users can select the mailbox format when they create a new mailbox.

See the [Mailboxes](#) section for more information about CommuniGate Pro Mailboxes.

Account Aliases

An Account Alias is an alternative name assigned to a CommuniGate Pro Account. Each Account can have zero, one, or several Account Aliases.

For example, the Account `j.smith` in the `domain2.com` Domain can have aliases `smith` and `jsmith`. Mail sent to the `smith@domain2.com` address will be stored in the `j.smith` Account, and attempts to login as `jsmith@domain2.com` will open the same `j.smith` Account.

You can use [Forwarders](#) to assign alternative name for Accounts, too. If you create the Forwarder `js` in the `domain2.com` Domain, and make it point to the `j.smith` address, it will work as yet another alias for the `j.smith` Account.

If you rename the account `j.smith` into `james.smith`, all Account Aliases will "move" with it - `smith` and `jsmith` will remain the Aliases for the `james.smith` Account. If you remove the Account, the Account Aliases will be removed, too.

Renaming and removing of Accounts has no effect on the Forwarders: if you rename or remove the `j.smith` Account, the Forwarder `js` will continue to point to the `j.smith` address.

As a result, it is not recommended to use Forwarders where you can use Aliases. Forwarders should be used to create "objects" that redirect mail to other Domains or to other mail servers.

Default Settings

Each CommuniGate Pro Account has individual *settings*. Settings are specified by the system or domain administrator and most of them cannot be modified by the account owner.

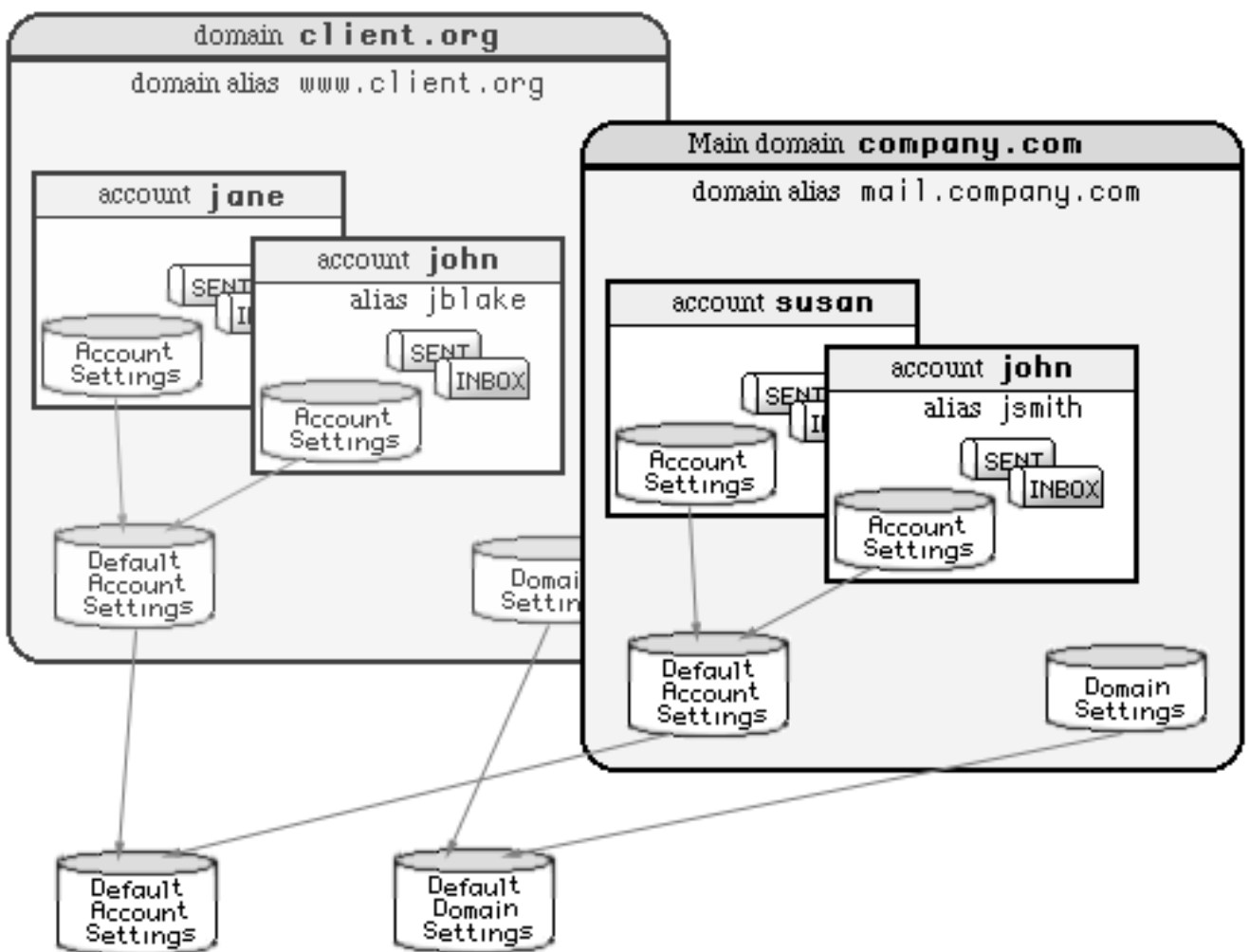
The Account Settings specify the user password and resource limits (maximum mailboxes size, number of files in the Personal Web Site, etc.), authentication methods, and other individual options.

It is convenient not to specify the same setting value explicitly for each account, but let all accounts in a domain, or even all accounts on the CommuniGate Pro server use the same setting value that can be modified for all accounts at once. If you specify the `default` value for an account setting,

the system will use the setting value retrieved from the *Default Account Settings* for the account domain. The domain Default Account Settings can themselves be specified as having the default value, in this case the setting value is retrieved from the global *Default Account Settings* specified for the entire Server.

Each CommuniGate Pro Domain has individual *settings*. Settings are specified by the system or domain administrator.

It is convenient not to specify the same setting value explicitly for each domain, but let all domains use the same setting value that can be modified for all domains at once. If you specify the default value for a domain setting, the system will use the setting value retrieved from the *Default Domain Settings*.



Example:

The global (Server)	Default Account Settings:	Storage Limit = 10Mbytes
The company.com	Default Account Settings:	Storage Limit = 30Mbytes
The client.com	Default Account Settings:	Storage Limit = default

Now:

- If you create an account in any domain, and set its `Storage Limit` to some value, that value will be used.
- If you create an account in the `company.com` domain, and set its `Storage Limit` value to `default`, the account will be able to keep up to 30Mbytes of mail (the Default Account Setting for that domain).
- If you create an account in the `client.com` domain, and set its `Storage Limit` value to `default`, the account will be able to keep up to 10Mbytes of mail (the global Default Account Setting for the Server).

When you serve many accounts, you should try to specify most of the setting values as `default`, so you can easily change those settings for all accounts. If some account should be treated differently, you explicitly specify the required setting value for that account.



Domains

CommuniGate Pro Server can serve accounts in its Main Domain, and, optionally, in multiple Secondary Domains, each with its own set of user accounts (and other objects such as mailing lists, groups, and forwarders).

Every domain can have one or several *aliases* (alternative names). All domain names and domain aliases should be unique, and they should be registered with the Domain Name System (DNS).

In many cases, a mail domain should not have a separate set of user accounts, but should rather be a domain name alias for an already existing CommuniGate Pro domain. You may also want to serve some mail domains using account mapping and/or Unified Domain-Wide Accounts. In all these cases, you do not have to create a new CommuniGate Pro Secondary domain to serve a mail domain.

See the [Mapping](#) section for the details.

When a client application (a mailer) connects to your CommuniGate Pro server, and specifies an account name, the Server has to detect in which domain to look for that account.

You can use Secondary domains if:

- your Server has multiple IP addresses (i.e. it uses *multihoming*), and each secondary domain has its own (dedicated) IP address on your Server computer. In this case the server will detect the IP address to which a client connects and look for accounts in the domain associated with that IP address.

or

- your clients use the [Web User Interface](#)

or

- your clients use mailers that can specify the full account name (i.e. an account name and the domain name) when connecting to a server.

See the [Access section](#) for the details.

Displaying the Domain List

To display the list of all Domains served with your server, use a Web browser and enter the Domains section. You should be connected as the Postmaster or any other user with the Can Modify All Accounts and Domains access rights.

Filter:						
2083 Accounts	3 of 3 Domains selected	Show Aliases			3 selected	
Domain	IP Address	Accounts	Open	Hits	Last Hit	Refs
client1.com	192.0.0.2	645	24	2891	21:29:17	14 Settings
client2.com		78	5	3456	21:30:32	7 Settings
mycompany.com	192.0.0.1	1380	89	15890	21:30:29	35 Settings
mail.client1.com		client1.com				Settings
mail.client1.com		client2.com				Settings
webmail.client1.com		client1.com				Settings

To select domains by name, type a string into the `Filter` field, and click the Display button: only the domains with names containing the specified string will be displayed.

Each entry in the domain list contains the Domain name, the assigned network address (if any), and the number of Accounts in the Domain. If the Domain is a shared Domain served by a [Dynamic Cluster](#), the Domain name has the [+] prefix. If the Domain is a Directory-based Domain, its name is displayed with the [D] prefix.

A list entry also displays the number of currently opened domain accounts, the total number of times domain accounts have been opened (since the Server last restart), and the last time any domain account was opened.

Select the `Show Aliases` option to include domain aliases into the list. Each domain alias list element contains the link to its "real" domain account list and settings pages.

Click a domain name to view the [accounts](#) in that domain.

Click the word Settings in the last column to view and update the domain Settings.

Creating a New Domain

Type a new domain name into the field on the right side of the Create Domain button.

Domain created

Click the Create Domain button. When a new domain is created, its name appears in the Domain List.

If a server is a member of a [Dynamic Cluster](#), the additional Create Shared Domain button appears. Click that button to create a domain that will be served by all members of the Cluster. The domain created using the Create Domain button are created as "local" domains and are served by this server only.

Specifying Domain Settings

Main domain and all Secondary Domains have domain-level settings.

To open the Domain Settings page in your browser, either click the Domain Settings link in the Domains List, or click the Domain Settings link on the domain Accounts List page.

Account Log:

Mailbox Log:

The Account Log option allows you to specify how the account-level operations (account open/close, password verifications, mailbox creating/removing, size updates, etc.) are recorded. Log records created for account-related events have the ACCOUNT tag.

The Mailbox Log option allows you to specify how the mailbox-level operations (message storing/removing, message status updating, etc.) are recorded. Log records created for mailbox-related events have the MAILBOX tag.

Most of Domain Settings can be set to the [Default](#) value. In this case the actual setting value is taken from the global, Server-wide [Default Domain Settings](#).

When the Domain Settings are modified, click the Update button. The page should appear again, displaying the *Updated* marker.

You can click the Accounts link to switch to the domain Account List.

Enabling Messaging Services

Main domain and all Secondary Domains have domain-level settings that specify which CommuniGate Pro services can be used with the domain accounts:

Enabled Services					
Mail	POP	IMAP	PWD	ACAP	WebMail
WebSite	Relay	Mobile	FTP	MAPI	TLS

Mail

If this option is disabled, incoming mail is not delivered to Domain Accounts. Incoming messages are suspended in the [Local Delivery](#) module queue, and they are rejected if this option is not re-enabled within the specified period of time. See the [Local Delivery](#) module settings for the details.

POP, IMAP, PWD, ACAP

If a protocol option is disabled, Accounts in this Domain cannot be opened using that protocol.

WebMail

If this option is disabled, the Domain Accounts cannot be opened using the WebUser Interface, and the Domain mailing lists cannot be browsed.

WebSite

If this option is disabled, the Personal Web Sites in this Domain cannot be accessed via HTTP.

FTP

If this option is disabled, FTP access to this Domain is disabled.

MAPI

If this option is disabled, MAPI access to this Domain is disabled.

Mobile

If this option is disabled, Domain users will not be able to connect to their Accounts from Internet Addresses not included into the [Client Addresses](#) list. This can be useful if you provide free Accounts in this Domain and you want Domain users to connect to those Accounts only from the dial-up addresses your own site provides.

Relay

If this option is disabled, Domain users will not be able to use the [Mobile Users Support](#) features. This can be useful if you provide free WebMail Accounts in this Domain and you do not want spammers to use these Accounts to enable SMTP relaying.

TLS

If this option is disabled, secure (SSL/TLS) access to Accounts in this Domain is disabled.

Services can also be disabled for individual Domain [Accounts](#).

A service is available for an Account only if that service is enabled for the Account itself AND for the Account Domain. Disabling a service in the Domain Settings disables that service for *all* Domain Accounts.

Note: This is different from disabling a service in the Domain Default Account Settings: disabling a service in the Default Account Settings disables that service only for those Domain Accounts that have the Enabled Services option set to default.

Multihoming and Dedicated IP Addresses

You should read this chapter only if you plan to support multihoming, if your system is behind a firewall, or if you have a non-standard Domain Name System setup.

When the Server starts, it detects its own network address(es). Your Server system is "multihomed" if it has more than one network (IP) address.

If the Server system has several IP addresses, some of them can be assigned (dedicated) to secondary domains. Accounts in such domains can be [accessed](#) using any POP and IMAP mailer without explicitly specifying the full account name.

The Assigned IP Addresses option allows you to assign network addresses to the main and secondary domains.

Assigned IP Addresses
[206.40.74.198]

All Available

This option can be selected for the Main Domain only, and it is the default setting for the Main Domain. All Server's network addresses not assigned to other Domains are assigned to this Domain.

Manually Defined

This option is selected by default for all secondary Domains.

If you want to assign (dedicate) an IP address to this Domain, type the address into the text field on the right of the pop-up menu.

Only the Server computer's own addresses are accepted, and all specified addresses should not be already assigned to any other CommuniGate Pro Domain.

If you select this option and leave the text field blank, the Domain will not have any IP addressed assigned to it. In this case, to access the Domain accounts, users should specify the full account name (*account@domain*) in their mailer settings. See the [Access](#) section for the details.

by DNS A-Record

When this option is selected, the Server sends a request to the Domain Name System and tries to resolve the Domain name. If an A-Record for this CommuniGate Pro Domain is found in the Domain Name System, the

addresses from that record are assigned to the Domain. The system checks that all addresses retrieved from the A-record belong to the Server computer and that these addresses have not been already assigned to any other Domain.

This setting is useful if you have several secondary Domains with dedicated IP addresses and you want to redistribute the Server addresses from time to time. Instead of reconfiguring both DNS and Server settings, you may reconfigure the DNS records only, and the Server will take the updated data from the DNS.

by DNS MX-Record

When this option is selected, the Server retrieves the highest-priority MX record (relay name) for this CommuniGate Pro Domain, and then processes addresses in the A-record for that relay name.

For each Domain in the Domain List, the assigned network (IP) addresses are displayed. This can be used to check the DNS and Server setup for systems with multihoming.

Because of setup errors or due to a non-standard network and DNS setup, the Server's own IP address(es) may be left unassigned to any of the Server domains. Open the [General Settings](#) page to see the list of the Server own IP addresses. The unassigned addresses are marked in red.

When a client mailer connects to the Server via an unassigned address and the full account name is not specified, the Server does not allow the user to log in.

Domain Limits

The System Administrator can specify some limits on the resources available for the domain: the domain administrator is not allowed to create more accounts or more mailing lists than specified with these domain settings.

A Domain Administrator can see, but cannot modify these limits.

Resources	Limits	Usage
Accounts:		390
Mailing Lists:		5
RPOP Accounts:		15
MAPI Connections:		37

Domain Aliases

Each CommuniGate Pro domain can have aliases (alternative names). If the domain `client.dom` has the `mail.client.dom` and `www.client.dom` aliases, mail directed to

user@mail.client.dom and to user@www.client.dom will be routed to the user@client.dom account. Also, to access the user@client.dom account via POP, IMAP, and other mailer applications the account names user@mail.client.dom and user@www.client.dom can be specified in the mailer settings.

This is especially useful for [WebUser](#) clients. Users specify the domain name in their browser URLs, and users of the client.dom domain tend to use www.client.dom in the browser URLs. You may want to register the www.client.dom domain with the DNS, assigning it the same IP address as the address assigned to the client.dom domain, and then you should create the www.client.dom alias for the client.dom domain.

Aliases

You can modify existing aliases, add an alias by typing a new name in the empty field, and remove an alias by deleting it from its field. Use the Update button to update the list of domain aliases.

The Domain Aliases are stored in the DomainAliases [database](#) located in the Settings directory inside the CommuniGate Pro *base directory*.

Directory Integration

The System Administrator can specify if the domain accounts should be included into the Central Directory.

Directory Integration
Usage:

This panel is not displayed for Directory-Based Domains, since those domains are always completely integrated with the [Directory](#).

See the [Directory Integration](#) section for the details.

Processing Unknown Names

Addresses used in mail messages and in client "login names" can contain unknown names. If the Server cannot find an object (an account, a mailing list, an alias, or a forwarder) with the specified name, the Domain Unknown Names settings are used.

Unknown Names
Consult External Authenticator:
Mail to Unknown Names is:

Consult External Authenticator

When an unknown name is supplied and this option is enabled, the CommuniGate Pro Server sends a command to the [External Authentication](#) Helper application. That application can check an external database (or any other data source) and optionally create a new object (account, alias, etc.) with the specified name. If the program returns a positive response, the Server makes one more attempt to find a domain object.

Mail to Unknown Names

This setting specifies what the Server should do when unknown account/object names are encountered in message addresses.

Rejected

The address is rejected; if the message is being received via SMTP, the address is not accepted, and if it was the only message recipient address, the message is not received at all.

Discarded

The address is routed to NULL. The message is considered "delivered" immediately (it is discarded).

Rerouted to:

the address is changed to the E-mail address specified in the text field, and the [Router](#) restarts trying to route this new address.

Note: you specify an E-mail address, not an account name there. So, if you specify Rerouted To: Postmaster for the client1.com domain, messages sent to unknown names will be routed to the Postmaster account in the Main Domain, not to the postmaster Account in that client1.com Secondary Domain. Specify postmaster@client1.com to direct those messages to the postmaster Account in the client1.com Domain.

Note: you can use the asterisk (*) symbol in the E-mail address field. This symbol will be replaced with the original (unknown) name.

Sample:

The domain client1.com Mail to Unknown Name option is set to
Rerouted to: Bad-*@support.company.com

A message comes addressed to jjones@client1.com, and the Account jjones does not exist in the client1.com Domain.

The message is rerouted to bad-jjones@support.company.com

Accepted and Bounced

The Router accepts E-mail addresses with unknown names, routing them to the Local Delivery module. When the message is enqueued into the Local Delivery module queue, the module fails to find the addressed account/object, the message is rejected, and an error report is sent back to the sender.

Sending Mail To All Accounts in the Domain

The administrator can enable the special virtual list (address) "all" that can be used to send messages to all Accounts created in this Domain.

Mail to <all@client1.com>	
is distributed for	Send to Forwarders:

Messages sent to the <all@domainname> address are stored directly in the Account INBOX mailboxes, bypassing any Account [Rules](#).

Messages sent to the <all@domainname> address are not stored in the Accounts that have the Accept Mail to All setting disabled.

Mail access to the <all@domainname> address can be restricted.

anybody

Any message sent to the <all@domainname> is distributed to all accounts in this Domain.

Clients

A message sent to the <all@domainname> address is distributed only if it has been received via SMTP from an Internet address included into the [Client IP Addresses](#) list, or if the message was received using one of the *trusted* methods (Web User Interface, via RPOP, via POP using the XTND XMIT method, etc.).

Authenticated Users

A message sent to the <all@domainname> address is distributed only if it has been received from a Server user (Account) using one of the *trusted* methods.

Authenticated Domain Users

A message sent to the <all@domainname> address is distributed only if it has been received (using one of the *trusted* methods) from an Account in this Domain or from any other Server Account that has the [domain administration](#) right for this domain.

Authenticated Administrator

A message sent to the <all@domainname> address is distributed only if it has been received (using one of the *trusted* methods) from a Server Account that has the [domain administration](#) rights for this domain.

nobody

The <all@domainname> address is disabled. In this case it is possible to create the real Account, Forwarder, Group, or Mailing List with the All name.

Messages to <all@domainname> can be sent to all Forwarder addresses, too:

Send to Forwarders:

When this option is enabled, a new message is composed. Its envelope contains the addresses from all Forwarder objects in this Domain. The message body is a copy of the message sent to the <all@domainname>.

Sending Mail To All Accounts in All Domains

If the administrator has enabled mail distribution to all accounts in the main domain, a message can be sent to all accounts in all domains.

To send a message to all accounts in all server domains, it should be sent to the alldomains@main_domain_name address.

For each domain, the message source is checked and the message is distributed to the domain accounts only if it passes that domain "Mail to All" distribution checks.

WebUser Interface Settings

Each domain has several [WebUser Interface](#) options:

WebUser Interface	
Cache Files:	
Mail Trailer	
Text:	
Web Banner	
Text:	

Cache Files

If this option is enabled, the CommuniGate Pro server maintains a memory cache for files (HTML templates, images, etc.) in the domain [WebUser directory](#) and its subdirectories. When you upload a modified file to the domain WebUser directory using the CommuniGate Pro uploading methods (HTML PUT command, form uploading, etc.), the Server automatically removes the old version of that file from the Webuser cache.

If you prefer to modify the domain WebUser files directly, you may want to disable the WebUser Caching.

Flush Cache

Click this button to remove all domain WebUser Interface files from the memory cache. It also causes the WebUser module to reload the `Strings.data` file from the domain WebUser directory (if that file exists).

Mail Trailer Text

The text in this field is automatically appended to all messages the domain users compose via the WebUser Interface.

Web Banner Text

The text in this field is automatically inserted into the beginning of all HTML files retrieved from the domain user Personal Web Sites.

Enabling Auto-Signup

You can allow users to create domain accounts themselves, via the [Web User](#) Interface:

Auto Sign-up
Enabled

If the Auto Sign-up option is enabled, the Sign-up link appears on the domain Login Web page. This link allows new users to open the Sign-up page, where they can enter a new account name, the user real name, and the desired password.

The Server checks that an account with the specified name does not exist and creates a new account. The Server uses the Account Template settings for the newly created account, overriding its Password and Real Name settings with the data specified by the new user.

Relaying via a Dedicated IP Address

You may want to tell the CommuniGate Pro Server to use a particular Local IP Address to send all mail originated from this Domain. For example, if there is a risk that some of the Domain users can be involved in spamming activity, you may want to send all the Domain mail via an IP address dedicated to that Domain, so your Server other IP Addresses will not be blacklisted.

Use the SMTP Sending panel to specify the Relaying IP Address:

SMTP Sending	
Use:	IP Address

The Use IP Address setting values:

any

The SMTP module should not use any particular IP Address when sending messages received by this domain. The module allows the Server OS to select some Server Local IP Address for outgoing SMTP connections.

first

For all messages received for or generated in this Domain, the SMTP module should use the first IP Address from the set of IP Addresses assigned to this Domain.

same

The SMTP module should try to use the same Local IP Address that was used to receive the message. If the message was received using the SMTP protocol or using the XTND XMIT extension of the POP3 protocol via a Server Local IP Address assigned to this Domain, the same Server Local IP Address will be used to send the message out (to relay it). If the message was composed during a WebMail Session opened for some Account in this Domain, the SMTP module will allow the Server OS to select the Local IP Address to use.

Note: this option should NOT be used if you have a firewall and some of the Server Local IP Addresses belong to an internal LAN: users can submit messages via internal LAN IP addresses, and the SMTP module will fail to send those messages to the Internet if it has to use an internal LAN IP address.

Note: in most cases, the **First** option (see above) provides better results.

Server OS Integration

CommuniGate Pro Accounts may be "mapped" to the accounts (registered users) of the Server OS. See the [Accounts](#) section for more details.

Legacy (Unix) Mailer Compatibility

The CommuniGate Pro allows you to create Accounts with [external INBOX](#) mailboxes. These mailboxes are stored not inside the CommuniGate *base directory*, but in the system file directory

known to the legacy mailer applications.

If you have to support Local Mailer compatibility for all or some accounts in a domain, you should specify the External INBOX settings:

Server OS Integration		
External INBOX	location:	default ()
	synchronize using:	Locks

location

This setting specifies where the external INBOX files should be located. For each Account that has an external INBOX, the Server substitutes the asterisk sign (*) with the CommuniGate Account name. Consult with your OS manuals to see where your legacy mailers expect to find user mailboxes. On most Unix systems, the `/var/mail/` directory is the correct location, but some systems may use `/var/spool/mail/` or some other directory.

synchronize using

This setting specifies the file locking method to use for updates synchronization.

See the [Sharing](#) section for the details.

Subdirectories for Large Domains

When a CommuniGate Pro Domain contains many Accounts (more than 10,000), you may want to place account files in several subdirectories:

- many operating systems have limits on the number of files in one directory;
- subdirectories can speed up the account files access operations;
- subdirectories can be moved to additional storage devices.

Domain subdirectories are directories inside the domain directory. A subdirectory name should have the `.sub` file path extension (suffix).

Subdirectories can be nested.

Note: When the CommuniGate Pro server starts, it scans all domain directories and all their subdirectories, and it collects the names of all domain Accounts. This feature allows the system administrator to move accounts between subdirectories at any time when the server is stopped. It also allows you to change the foldering method (see below) without stopping the server and without relocating already created accounts.

For each Account, the CommuniGate Pro server remembers the name of the subdirectory that contains the account files.

When a new Account is being created (or when an existing Account is being renamed), the Server composes a name for the subdirectory in which the Account files should be created.

Account Storage
Foldering Method:
Rename In Place:
Generate Index:

Foldering Method

This option allows you to specify the subdirectory name construction method. The following methods are supported:

flat

This is the default method. All new Accounts are placed into the domain directory itself.

2 Letters 1 Level

The first two letters of the Account name are used to form the name of the subdirectory, the Account `jsmith` will be placed into the `domain/js.sub/` subdirectory. If the account name has just one letter, that letter is used as the subdirectory name.

2 Letters 2 Levels

The first two letters of the Account name are used to form the name of a nested subdirectory, the Account `jsmith` will be placed into the `domain/j.sub/s.sub/` subdirectory. If the account name has just one letter, that letter is used as the subdirectory name.

Hashed 1 Level

A numeric hash function is applied to the Account name, the result is used to form a subdirectory name: the Account `jsmith` will be placed into the `domain/pf.sub/` subdirectory.

Hashed 2 Levels

A numeric hash function is applied to the Account name, the result is used to form a nested subdirectory name: the Account `jsmith` will be placed into the `domain/lu.sub/y.sub/` subdirectory.

Note: many other systems serving large domains use domain subdirectories, too. Every time an account is to be opened, those systems form the account subdirectory name using some built-in method. As a result, the built-in method cannot be changed "on the fly", and accounts cannot be moved between subdirectories. The CommuniGate Pro Server uses its subdirectory name forming methods only when a new account is being created, and it always remembers in which subdirectory every account is located. The Server does not have to form the subdirectory name every time an account is to be opened. As a result, the CommuniGate Pro domain "foldering" methods can be

changed at any moment, and the Accounts can be moved between the subdirectories when the server is not running.

Note: if you cannot store all Domain Accounts on one disk volume, you can copy some `xx.sub` directories to other volumes, and replace them with symbolic links.

Rename in Place

If this option is not enabled, and you rename an Account, the CommuniGate Pro Server uses the currently set Foldering method to compose a new file path for the renamed Account and moves the account data there. If you have replaced the `xx.sub` directories with symbolic links to directories on different disk volumes, such a rename operation may require moving data from one volume to a different one, and it will fail. If you enable this option, the CommuniGate Pro Server will move (rename) the renamed Account data within the same directory, so the "cross-volume link" problem will be avoided.

Generate Index

If this option is enabled, the CommuniGate Pro Server creates the `Index.data` file in the Domain file directory. This file contains the names of all Domain Accounts, the Account types, and the location of the Account files. When the Server starts and finds the `Index.data` file in the Domain directory, it reads that file instead of scanning the Domain file directory tree. On some file systems scanning a directory tree with 100,000 files can take up to 10 minutes.

Note: if you have stopped the Server and manually moved/removed some Domain Account directories, delete the `Index.data` file from the Domain directory before you start the Server again.

Note: if you want to keep only symbolic links in the Domain file directory, you can create the `Index` subdirectory inside the Domain directory (or an `Index` symbolic link to some other directory). If this subdirectory exists, the Server stores the `Index.data` file inside that subdirectory rather than in the Domain file directory itself.

Administrator Domain

Domains can be controlled by the [Server Administrators](#) and by the [Domain Administrators](#) - Accounts in the same Domain that are granted some Domain Administrator Access Rights. You may choose to grant administration rights for this Domain to Domain Administrators created in a different Domain. In this case the name of that other Domain should be entered into the Administrator Domain Name field:

Administrator Domain

If this field is not empty, the Domain Administrator Accounts created in this Domain and the Domain Administrator Accounts created in the specified Domain can be used to administer this Domain.

See the [System Administrator](#) section for more details.

Renaming Domains

If you want to rename a Secondary Domain, open its Domain Settings page with a Web browser, and enter a new account name into the New Domain Name field. Click the Rename Domain button.

If there is no other domain with the same name as the specified new domain name, the domain is renamed and its Domain Settings page should reappear on the screen under the new name.

You cannot rename a domain when any of its accounts is in use.

Removing Domains

If you want to remove a Secondary Domain, open its Domain Settings page with a Web browser, and click the Remove Domain button. The confirmation page should appear. If the Empty Domains Only option is selected, a Secondary Domain is removed only if there are no accounts in it. Otherwise, all Domain Accounts are permanently removed, too.

If you confirm the action, the selected domain, its settings, and all its accounts will be permanently removed from the Server disks.

You cannot remove a domain when any of its accounts is in use.

Specifying Default Domain Settings

A domain setting can have the `default` value. In this case the actual setting value is taken from the global Default Domain Settings. You can modify these Default values by clicking the Domain Defaults link on the Domains (Domain List) page.

The Default Domain Settings page resembles a regular Domain Settings page.

A Dynamic [Cluster](#) installation maintains separate server-wide Default Domain settings for all non-Shared (Local) Domains, and cluster-wide Default Domain settings for all Shared Domains. In the Cluster environment, the Default Domain Settings page displays links that allow you to switch between the Server-wide and Cluster-wide Default Settings.

Specifying Domain Security Settings

Domains can have security settings (Private Keys and Certificates) that can be used for [secure communications](#) with that domain.

Use the Security link on the Domain Settings page to open the Domain Security settings.

See the [Security](#) section for more details.

Domain File Directories

Account files for the main domain accounts are stored in the `Accounts` directory inside the CommuniGate Pro *base directory*.

For each secondary domain, a directory with the domain name is created inside the `Domains` directory. The domain directory contains files for all secondary domain accounts.

When a domain contains many Accounts, [subdirectories](#) inside the domain directory can be used.



Mapping

Many domain names can have DNS records pointing to your Server. But the Server automatically processes mail sent only to its Main Domain, or to one of its [Secondary Domains](#).

To let the Server accept and locally process mail sent to any other domain, that domain should be mapped to any existing CommuniGate Pro domain. Otherwise, a message sent to an unlisted domain will be directed to the SMTP module for relaying, and that module, detecting that it has to send the message to itself, will reject the message reporting about a 'DNS loop'.

CommuniGate Pro provides a variety of methods to serve multiple domains. The Server Administrator should decide how to serve each domain and either create a new full-scale [Secondary Domain](#) or just map a mail domain name onto one of the existing CommuniGate Pro Domains.

CommuniGate Pro Domains

The CommuniGate Pro Server allows you to create [Secondary Domains](#) in addition to the Main Server Domain. In this section the Main and Secondary domains are referenced as *real* or *CommuniGate Pro* domains.

Each CommuniGate Pro domain has its own set of accounts, own settings, own WebUser Interface, etc. If `client1.com` and `client2.com` are CommuniGate Pro domains, both domains can have the account `info`, and these Accounts are different - each one with its own settings, mailboxes, passwords, etc.

Each CommuniGate Pro Domain can have its own [Domain Administrator\(s\)](#).

Direct Mapping (Domain Aliases)

Very often one real Domain should have several aliases (additional names). For example, if your CommuniGate Pro has the `company.com` Domain, you may want to process the `mail.company.com` and `relay.company.com` domain names as aliases for the `company.com` Domain.

To create a Domain Alias, open the `company.com` [Domain Settings](#) page and enter the `mail.domain.com` name into the [Aliases](#) table. Click the Update button to see the alias name accepted, and enter the `relay.company.com` name into the new empty field. Click the Update button again.

Created aliases will tell the [Router](#) that all references to these two domain names should be substituted with the `company.com` name. As a result, messages sent to `info@company.com`, `info@mail.company.com`, and to `info@relay.company.com` will all end up in the `info` Account in the `company.com` CommuniGate Pro Domain.

The Router and Domain Aliases are also used for [account access](#), so when a user tries to access the `info@mail.company.com` account, the `info` Account in the `company.com` CommuniGate Pro Domain is opened.

You can use the [Router](#) *domain records* to achieve the same results:

```
mail.company.com = company.com
relay.company.com = company.com
```

You can also use the [Router](#) *alias records*:

```
<*@mail.company.com> = *@company.com
<*@relay.company.com> = *@company.com
```

Modifying Mapping

Sometimes you do not want to create a separate CommuniGate Pro Domain for a mail domain (for example, if that mail domain will have only few accounts), but you do not want the account names in that mail domain to interfere with the account names in the CommuniGate Pro Domain you map it on.

For example, your client with `client.com` CommuniGate Pro Domain wants to accept mail for the `info@shop.client.com` and `order@shop.client.com` addresses, but these accounts should not be the same as the `info` and `order` Accounts in the `client.com` Domain.

Use the Router *alias record* to map the `shop.client.com` mail domain:

```
<*@shop.client.com> = shop-*@client.com
```

This Router record will redirect mail sent to `info@shop.client.com` to `shop-info@client.com`. Mail sent to `order@shop.client.com` will be redirected to `shop-order@client.com`, etc.

The Accounts `shop-info` and `shop-order` must exist in the `client.com` Domain.

To access these accounts, the users should specify the `client.com` name as the name of their mail server, and `shop-info`, `shop-order`, etc. as their account names.

You can use this method to modify just some of the additional domain account names, while mapping other names directly to the names in the `client.com` CommuniGate Pro domain:

```
<info@shop.client.com> = shop-info@client.com
```

```
<order@shop.client.com> = shop-order@client.com
```

```
<*@shop.client.com> = *@client.com
```

Please note that the generic alias record must be specified *after* the first two records.

Unified Domain-Wide Accounts

You can tell the CommuniGate Pro Server to store all mail for a mail domain in one Account. This method is useful if:

- the mail domain belongs to a dial-up client system that does not have a static IP address and thus cannot receive its mail via SMTP;
- the mail domain has only few POP3 users and you do not want to create a full-featured CommuniGate Pro domain to serve them.

To create a Unified Domain-Wide Account for the `client.com` domain, use the following Router *domain record*:

```
client.com = client.local
```

or a Router *alias record*:

```
<*@client.com> = *@client.local
```

All messages to the `client.com` mail domain will be stored in the `client` Account in the CommuniGate Pro Main Domain. The `.local` suffix explicitly tells the [Local Delivery Module](#) to

accept this address and direct it to the `client` account.

The Local Delivery Module uses the *local part* of the address to form the `X-Real-To:` header fields in the stored messages. These fields will allow the client software to retrieve messages from the CommuniGate Pro Unified Domain-Wide Account and distribute them locally to the proper users of that mail domain. See the [Local Delivery Module](#) section for more details.

If users of such mail domain do not have their own mail server that retrieves mail from the Unified Domain-Wide Account, but connect to your CommuniGate Pro Server directly, the [POP module](#) will show each user only the messages directed to that user, rather than all messages stored in the Unified Domain-Wide Account. See the [POP Module](#) section for more details.

Please note that while the following Router record will also store all mail sent to the `client.com` domain in one `client` account:

```
<*@client.com> = client
```

But this Router record discards the information about the original user name (the part before the `@` sign). As a result, no `X-Real-To:` header fields will be added to the messages stored in the `client` account.

You can mix the mapping methods with the Unified Domain-Wide Account method. For example, if you want messages sent to the `jim@client.com` address to be stored in the `client-jim` account in the Main CommuniGate Pro domain, while directing the rest of the `client.com` mail to the Unified Domain-Wide Account `client`, use the following Router records:

```
<jim@client.com> = client-jim  
<*@client.com> = *@client.local
```

If you are serving many small mail domains, providing Unified Domain-Wide Accounts can be a very effective alternative to real CommuniGate Pro domains.



Accounts

An account is the basic *service unit*: every user served with a CommuniGate Pro server should have an account on that server.

Each account is protected with a password, so only the account owner (and, optionally, system and domain administrators) can have access to account data.

The Postmaster account is automatically created in the main Server domain. The Master (unlimited) access right is granted to that account.

Displaying the Account List

To browse the list of CommuniGate Pro accounts or to create a new account, use a Web browser to access the [WebAdmin Interface](#) and enter the Accounts section.

If you want to view accounts in a [Secondary Domain](#), enter the Domains section, and follow the link for that domain. You should have the Can Modify All Domains And Accounts [access right](#) to browse, create, and remove accounts and to modify account settings.

If you are a [Domain Administrator](#), then the list of accounts in your domain appears on the main [Domain administration](#) page.

The Accounts page lists accounts and other [objects](#) in the selected domain.

Filter:				
Groups (1 of 4)	Forwarders (0 of 10)	Accounts (2 of 345)	Show Account Info	
Account	Type	Current Size	Last Access	
lsmith	MultiMailbox	140K	19-12-1998	[206.40.74.198]
fsmith	Text Mailbox	34K	20:34:56	[206.40.74.195]
smith-L	GROUP	15		

To select users by name, type a string into the Filter field, and click the Display button: only the objects with names containing the specified string will be displayed.

The pop-up menu allows you to limit the number of objects to be displayed.

The options allow you to specify the type of objects you want to display: Account, Groups, Forwarders, and show the selected and total number of those objects in the domain.

Each line in the list contains an object name and its type.

- If the object is a Group, the group members number is displayed.
- If the object is a Forwarder, the forwarding address is displayed.
- If the object is an Account, the account type is displayed. If the Show Account Info option is selected, the line displays the information about the last user session with this Account: the time when the Account was opened and the network address from which it was accessed.

This info retrieval operation can be resource-consuming.

Creating a New Account

To create a new Account, type a new account name into the field on the right side of the Create Account button.

external INBOX in /var/mail/*

Use the pop-up menu to specify the account type:

MultiMailbox

A folder-type Account that can contain several mailboxes of various types. The INBOX mailbox

is automatically created within the new Account. All incoming mail is stored in the INBOX mailbox by default. The user can create additional mailboxes using any IMAP client software, or using the CommuniGate Pro Web E-mail Interface.

Text Mailbox, MailDir Mailbox, ...

An Account that contains a single INBOX mailbox. You can select any supported [mailbox format](#). If the user plans to use just POP3 client software, only one mailbox is needed, and you may want to create a Single-Mailbox type Account for that user.

By default, the account name becomes the person's E-mail name, so account names should contain only letters, digits, dash and point signs - some mail systems cannot send mail to accounts if account names contain other symbols.

external INBOX

Select this option if you want the new account INBOX to be created as an [external mailbox](#), so new account can be used with legacy *local mailers*. This option is enabled only if the external mailbox location is specified in the Domain Settings.

Click the Create Account button. When a new Account is created, its name appears in the Accounts list. The Server automatically displays the [Settings page](#) for the new Account.

The Settings of a newly created Account are automatically set to the [Account Template](#) values.

You can create several Accounts at once, by preparing an Account List file and using the [Import](#) option.

Specifying Account Settings

To specify Account Settings, click the account name in the Accounts list. The Account Settings page appears.

Account Type: MultiMailbox Access Rights

Account Type

This field indicates the account type (single-mailbox or multi-mailbox).

Access Rights

This link is used to open a page and grant [Access Rights](#) to the user.

Real Name: organization: city: CommuniGate Password: telephoneNumber:	
---	--

Real Name

This field is used to specify the real-life user name. The Server uses this information to compose the default 'From' address in Web Mailer.

additional *System* fields

If the Server [Directory Integration](#) settings contain some System Custom Account Setting fields, these fields appear in this panel where they can be set and modified.

CommuniGate Password

The account password. When authenticating a user, the Server can check either this password or OS password, or both (see below).

additional *Public Info* fields

If the Server [Directory Integration](#) settings contain some Public Info Custom Account Setting fields, these fields appear in this panel where they can be set and modified.

The modified values of the Real Name and additional fields are updated in the Directory if the Domain has the [Directory Integration](#) setting set to Keep In Sync.

After the Account Settings are modified, click the Update button.

Authentication Methods

Use the following settings to specify the account authentication methods.

CommuniGate Password	Server OS Integration
Allow to Use:	default (*)
Allow to Modify:	OS UserName:
Encryption:	Enable OS Password:
Secure Login	External Authentication
Required:	Allow to Use:

CommuniGate Password

Allow to Use

This setting tells the Server if it should use the CommuniGate Password string when

authenticating a user. The user may use the CommuniGate Password, or the Server OS password (see below) to connect to the CommuniGate Pro Server.

Allow to Modify

This option allows the user to modify the CommuniGate Password via either the [PWD module](#) or via the [WebUser Interface](#) for Account Settings.

Encryption

This option specifies how the Server should store the CommuniGate Password. If the `clear` option is selected, the password is stored as a clear-text string. All other options specify various encryption methods. In most cases, you will not specify this setting on a per-account basis, but rather using the Domain Account Defaults or global Account Defaults.

The `U-crypt` password encryption is available on Unix platforms only. It is used for compatibility with the Unix "crypt" encryption method and it should be used for migrating users from other mail servers only. The U-crypt-encrypted passwords can not be used for Secure (SASL) Authentication methods.

See the [Security section](#) for the details.

Server OS Integration

CommuniGate Pro Accounts can be "mapped" onto the accounts (registered users) of the Server OS. When a CommuniGate Pro user is being authenticated using a Server OS password, or when a separate process (program) should be launched on the user behalf, the CommuniGate Pro Server constructs an OS *username* (OS account name) to be used for that CommuniGate Pro user (account).

Server OS user name

This setting specifies how to compose the Server OS *username*. The asterisk (*) symbol is substituted with the CommuniGate Pro Account name. If this setting contains just one symbol - the asterisk sign, then the CommuniGate Pro Account is "mapped" onto the OS account with the same name: when the CommuniGate Pro Server checks the OS password for the Account `jmsmith`, it checks if the specified password can be used to log into the OS account `jmsmith`. If the setting contains `*.dj`, the OS username for the CommuniGate Pro Account `jmsmith` is `jmsmith.dj` - and the `jmsmith.dj` name is used for all OS-level operations initiated on behalf of the CommuniGate Pro Account `jmsmith`.

Enable OS Password

This setting allows the user to work with the Account using the password set in the Server OS registration information for this user. If both OS and CommuniGate Pro passwords are enabled, and if at least one of those passwords matches the password provided by a user, the user is allowed to connect to the Account.

See the [Security](#) section for the details.

Allow to Use External Authentication

This setting allows the user to work with the account using the password verified with the External Authenticator program.

See the [Security section](#) for the details.

Secure Method Required

This option requires use of [secure authentication methods](#) (APOP or non-clear-text SASL methods) with this account. If a user mailer application connects to the Server and supplies a password for this account using an unsecure ("clear text") authentication method, the server will reject the connection even if the supplied password is correct. Clear-Text password are still accepted if they are passed through a secure (SSL/TLS) communication channel.

Note: Since OS passwords can be checked only using the clear-text authentication method, the Allow to Use CommuniGate Password option should be enabled and a CommuniGate password should be specified if the user is forced to use a secure authentication method.

If Allow to Use CommuniGate Password, Allow to Use External Authentication, and Enable OS Password options are disabled, the user will not be able to access the account.

Any of Authentication Setting can be set to the default value, in this case the setting value is taken from the domain Default Account Settings or the global Default Account Settings.

Enabled Services

There is a set of settings that specify which CommuniGate Pro services can be used with the account:

Enabled Services						
default	Mail	POP	IMAP	PWD	ACAP	WebMail
WebSite	Relay	Mobile	FTP	MAPI	TLS	

The Server checks the account and the account [domain](#) settings. Only if the service is enabled for both the account and the account domain, that service can be used with this account.

See the [Domains Settings](#) section for more details.

If you select the default option, the Enabled Services for this account are defined using domain Default Account Settings or the global Default Account Settings.

Please note a difference between the Default Account settings and the Enabled Services specified for the domain: while you can override the default account settings for some account by explicitly specifying the enabled services for that account, you cannot override the Enabled Services specified for the Domain. If the Default Account Settings disable POP and IMAP access, you can explicitly enable POP and IMAP access for a particular account. But if POP and IMAP access is disabled in the Domain Settings, no account in that domain can be accessed via these protocols.

Resource Usage Limits

Limits	
Current	
Mail Storage:	134K
Mailboxes:	33
Web Storage:	89K
Web Files:	6

Mail Storage

This option is used to specify the maximum total size of the all Account mailboxes. If a new incoming message cannot be stored in an account, because the account size would exceed the specified limit, the message is rejected and the message sender receives an error report.

Mailboxes

This option is used to specify the maximum number of mailboxes that can be created in this Account.

Web Storage

This option is used to specify the maximum total size of the all files in the account personal Web site. If this option is set to zero, the account Web Site is disabled.

Web Files

This option is used to specify the maximum number of all files in the account personal Web site.

Any of the Limits Settings can be set to the default value, in this case the setting value is taken from the domain Default Account Settings or the global Default Account Settings.

Processing Options

Processing	Allow to Modify
RPOP Accounts	
Automated Rules	

RPOP Accounts

This setting tells the Server if the user is allowed to specify remote host (RPOP) accounts that the [RPOP module](#) should poll on the user's behalf.

If this option is disabled, only the administrator can specify the RPOP accounts for this user.

Click the [RPOP Accounts](#) link to specify the remote accounts to be polled on behalf of this user.

Automated Rules

This setting tells the Server if the user is allowed to specify automated Rules that instruct the Server how to process incoming messages.

No

If this option is selected, only the administrator can specify the automated rules for this user.
Filter Only

If this option is selected, the user can specify only the following actions: Discard, Reject, Stop Processing, Mark, Add Header, and Store in.

All But Exec

If this option is selected, the user can specify any action, but the Execute action.

Any

If this option is selected, the user can specify any action.

Click the [Automated Rules](#) link to specify the rules to be applied to all incoming messages directed to this account.

If an administrator creates an Automated Rule containing actions the account user is not allowed to specify, the user will be able to view that Rule, but will not be able to modify any part of it.

Any of these Settings can be set to the default value, in this case the setting value is taken from the domain Default Account Settings or the global Default Account Settings.

Miscellaneous Options

Miscellaneous	
Accept Mail to all:	New Mailboxes:
Add Mail Trailer:	Add Web Banner:

Accept Mail to all

This setting tells the Server to store messages directed to the `all@domain` address in the account INBOX.

New Mailboxes

This setting is displayed for multi-mailbox accounts only. It specifies the default [format](#) for all new mailboxes created in this account.

Add Mail Trailer

This setting tells the Server to append the trailer text (specified in the [Domain Settings](#)) to all messages this user composes using the [WebUser Interface](#).

Add Web Banner

This setting tells the Server to insert the Web banner code (specified in the [Domain Settings](#)) to all HTML files retrieved from the account [Personal Web Site](#).

Any of these Settings can be set to the default value, in this case the setting value is taken from the domain Default Account Settings or the global Default Account Settings.

Specifying Account Aliases

Each account can have aliases (alternative names). If the account JohnSmith has the jsmith and j.smith aliases, mail directed to jsmith and to j.smith will be stored in the JohnSmith account. Also, to access the JohnSmith account via POP, IMAP, and other mailer application the user names jsmith and j.smith can be specified in the mailer settings.

Aliases

You can modify existing aliases, add an alias by typing a new name in the empty field, and remove an alias by deleting it from its field. Use the Update button to update the list of account aliases.

Alias names should not be the same as the name of some other account, alias, or a mailing list in the same domain.

Creating Mailing Lists

Every CommuniGate Pro Mailing List has an owner - an account in the main or one of the secondary domains. To create a Mailing List, you should create the Owner account first. For each list, the Mailing List manager creates several mailboxes inside the owner account, so the owner account should be of the MultiMailbox type.

Mailing Lists	
List Name	Subscribers
RD-List	1025

To create a mailing list, type the list name and click the Create List button. To modify the list settings, to rename and remove the mailing lists use the links to the [Mailing List Settings](#) pages.

Renaming Accounts

If you want to rename an account, open its Settings page with a Web browser, and enter a new account name into the New Account Name field. Click the Rename Account button.

If there is no other account with the same name as the specified new account name, the account is renamed and its Account Settings page should reappear on the screen under the new name.

You cannot rename an account when it is in use.

New Account Name:

Removing Accounts

If you want to remove an account, open its Settings page with a Web browser, and click the Remove Account button. The confirmation page should appear.

If you confirm the action, the selected account, all its mailboxes, settings, and other account-related data files will be permanently removed from the Server disks.

The account aliases and all mailing list owned by this account will be removed, too.

You cannot remove an account when it is in use.

Specifying Default Account Settings

An account setting can have the `default` value. In this case the actual setting value is taken from the Default Account Settings for this domain. You can modify these default values by clicking the Account Defaults link on the Account List or Domain Settings page.

The Default Account Settings page resembles a regular Account Settings page. Any setting on that page can also be set to the `default` value, in this case the actual value is taken from the server-wide Default Account Settings, which specify the default setting values for all Accounts in all Server Domains.

You can modify the server-wide Default Account Settings by clicking the Account Defaults link on

the Domains (Domain List) page.

A Dynamic [Cluster](#) installation maintains separate server-wide Default Account Settings for all Accounts in non-Shared (Local) Domains, and cluster-wide Default Account Settings for all Accounts in the Shared Domains. In the Cluster environment, the Default Account Settings page displays links that allow you to switch between the Server-wide and Cluster-wide Default Settings.

Specifying Account Template

When you have to create many accounts, you may want to specify some non-default setting for all new accounts. Each domain has its own Account Template, and you can modify it by clicking the Template link on the Account List page.

The Accounts Template page resembles a regular Account Settings page.

All the settings set there will be copied to all newly created accounts in this domain.

Note: The Default Account Settings and Account Template are quite different. The Account Template is used only when an account is being created. All template settings with non-default values are copied to the new account settings. If you modify the template settings after an account has been created, those account settings will not change.

Besides the initial, non-Default setting values, the Account Template can be used to instruct the server to create several mailbox in each new account (by default only the INBOX mailbox is created), to subscribe the account to certain mailboxes, and to create mailbox aliases in all newly created accounts.

Additional Mailboxes



Enter a name into the empty field to add a mailbox name to the list. In this sample, when a new multi-mailbox account is created in this domain, the mailboxes `Sent` and `Drafts` will be created in that account, along with the `INBOX` mailbox.

Initial Subscription

See the [Mailboxes](#) section to learn about Mailbox Subscriptions.

Creating initial non-empty subscription:

- simplifies the initial set-up of some client mailers that can access only those account mailboxes that are included into the Mailbox Subscription list;
- helps new users to subscribe to public mailboxes containing administrative information, news, etc.

Initial Mailbox Aliases

Alias Name	Foreign Mailbox Name

See the [Mailboxes](#) section to learn about Mailbox Aliases.

Specifying a non-empty list of mailbox aliases simplifies the initial set-up for Microsoft Outlook users that need access to public mailbox and other [foreign mailboxes](#), but cannot use their mailers to access foreign mailboxes directly.

Initial Greeting Message

This field can contain a mail message in the RFC822 format. If this field is not empty, then the specified message is stored in the INBOX mailbox of every newly created account.

The `Date` : header field is automatically added to the stored messages.

If the message contains non-ASCII symbols, then the message text should start with `[charsetName]` string, and the Content-Type header field should include the `charset=` parameter:

Initial Greeting Message

Templates can be used to generate an initial Personal WebSite page for all newly created accounts:

Initial Personal WebSite Home Page

This field can contain an HTML text. If this field is not empty, then the specified text is stored as the `default.html` file in the Personal Web Site area of every newly created account.

Importing User Account Information

The built-in Account Loader allows the administrator to register sets of users. The user names and account attributes should be placed into a tab-delimited text file on the administrator (client) computer, and that file should be uploaded to the server using the Import field.

Click the browse button to select a file on your local system, and then click the Import Accounts button to create accounts listed in the selected file.

Below is a sample IMPORT file:

Name	Type	Ignore	Storage	Aliases
johnd	MultiMailbox	sales dept	50M	
susan	MultiMailbox	mgmnt	10M	susan.s,susan_smith
sales	MultiMailbox	dummy	30M	
info	MultiMailbox	dummy	50M	help

Note: The import file must be prepared on the client computer (on the computer you use to run your browser). The browser allows you to upload files from disks connected to that computer, not to the CommuniGate Pro Server computer.

Note: When using Netscape and some other Unix browsers, make sure that the file name ends with the .txt suffix - otherwise the browser won't upload the file as a text one, and the file will be ignored.

Note: The 4.5 and later Macintosh versions of the Microsoft Internet Explorer upload Macintosh files in the encoded x-macbinary format if the file contains a *resource fork*. Most text files created with Macintosh text editor applications contain resource forks that keep the information about the file fonts, file window position, and other Macintosh data. Such files cannot be used as import files with the Microsoft Internet Explorer browser. Either use a text editor application that saves text files without resource forks or use a browser that uploads Macintosh files without encoding.

The first file line describes the file contents. It should contain tab-delimited names of account attributes. The following names are supported:

Name

This column contains the account names. This attribute is not required to be in the first column, but it must exist. All other attributes are optional.

RealName

This column contains the account user "real name".

Type

This column contains the account type (MultiMailbox, Text Mailbox, etc.). If the file does not contain this column, or this field is empty, the Account type selected on the Account List WebAdmin page is used.

Password

This column contains the account password. If the file does not contain this column, or this field

is empty, the CommuniGate Password and the Use CommuniGate Password settings are taken from the domain Account Template.

UnixPassword

This column can be used instead of the Password column. If it exists, it should contain crypt-encrypted account passwords. The Account Loader will add the binary prefix to those strings, so these CommuniGate passwords will be used as U-crpt encrypted passwords. See the [Migration](#) section for more details.

Storage

This column contains the maximum account size (in bytes, or in kilobytes, if the number is followed with K, or in megabytes, if the number is followed with M). The column data can contain -1 or unlimited to specify unlimited storage.

Aliases

This column contains the account aliases; several aliases may be specified in one field if they are separated with the comma signs.

Rules

This column contains the account [Rules](#). Rules should be represented in the internal format, as an [array](#) of individual Rules. Each Rule is an array, where the first element is the Rule priority, the second element is the Rule Name [string](#), the third element is the Rule conditions [array](#), and the last element is the Rule Actions [array](#).

Ignore

This column is ignored. An account list file can contain several Ignore columns.

setting name

You can use columns that contain initial values for various additional account settings (WebSite file and size limit, type or Rule actions enabled, etc.). The column should have the same name as that account setting name (keyword). For example, you can use the column named MaxWebSize to specify the storage limit for the account Web Site, and you can also use the column named MaxAccountSize instead of the Storage column.

Custom Setting

You can use columns that contain initial values for various [Custom Account Settings](#). For example, if the Directory Integration page contains the Custom Setting city, you can include a column named city in your Account Import file.

If the first line is parsed, all other lines are processed. Each line should contain tab-delimited fields, with the field contents specified in the first line. A line can contain less fields than the first line, in this case missing fields are processed as empty fields.

Attribute values for empty and missing fields are taken from the [Account Template](#).

If an error occurs while processing some file line (missing name field, duplicate name, etc.), all accounts created while processing previous lines are removed, and the number of the line that caused the problem is displayed. You can fix the file and try again.



Groups

CommuniGate Pro Domains can contain Group objects. A Group is a set of group member names. Group members are other objects in the same domain and/or external E-mail addresses. Each Group has its own Settings.

Messages sent to a group object are processed with the LIST module, which copies them without any modification of any header field, and resubmits them specifying group members in the envelope recipient list.

Administrators can create Group objects to simplify mailing to logical groups of users ("marketing", "sales", etc.). A message can be addresses to a simple `groupname@domain.dom` address, and the server will distribute it to all groupname Group members.

Creating a New Group

Groups in any domain can be created by the Server Administrator if the administrator account has the Can Modify All Accounts And Domain Settings access right.

A Domain Administrator can create, remove, and rename groups only if the CanCreateGroups access right is granted to the administrator account.

To create a new group, type a new group name into the field on the right side of the Create Group button.

Click the Create Group button. When a new group is created, its name appears in the list. The Server automatically displays the [Settings page](#) for the new group.

Specifying Group Settings

To specify Group Settings, click the group name in the Accounts list. The Group Settings page appears.

Real Name:	
Report Delivery to Group	Set Reply-To to Group
Expand Other Groups	
Remove Author from Distribution	Remove To and Cc from Distribution
Members	

Members

This is the list of all group members. If the member name does not contain a domain part, it specifies an object in the same domain: a domain account or some other group. The last empty element of the table allows the administrator to add a new member to the group. To delete a member from the group, delete the member name and click the Update button.

RealName

A brief description of the group. This string is used to compose the comment for this group E-mail address.

Report Delivery to Group

If this option is selected, then a delivery report (if requested) is generated as soon as a message is copied and re-submitted for delivery to all group members. If subsequent delivery to any group member fails, error reports are not generated.

If this option is not selected, delivery to this group is processed as "relaying", and the delivery notification options are copied to addresses of all group members.

If delivery to any group member fails, the sender gets an error message.

If a message was sent with delivery notification requested, the sender will get notification delivery from all group members.

Set Reply-To to Group

If this option is selected, the Reply-to: header pointing to the group address is added to the message copy before it is sent to group members. This ensures that replies to a message sent to this group will go back to the group, not to the message author.

Expand Other Groups

If this option is selected, the group members are checked before a message is copied and sent to member addresses. If a group member is a group in the same domain, then that group members are extracted and inserted into the address list. If that group also has this option enabled, the extracted members are checked, too. This option allows to process group delivery more efficiently (only one message copy is created for all recipients) and it also helps to avoid duplicates and mail loops.

If the group contains 2 other groups (sub-groups) as members and those sub-groups contain the same address, then only one copy of the message is delivered to that address if the Expand option is enabled. If this option is disabled, the copy of the original message will be delivered to both sub-groups, and each sub-group will send its copy of the original message to that address.

Remove Author from Distribution

If this option is selected, the message From: address is removed from the (optionally expanded) members list.

Remove To and Cc from Distribution

If this option is selected, all addresses from the message To and Cc fields are removed from the (optionally expanded) members list.

The sender address is always removed from the members list so the message sender does not get back his/her own messages.

Group Members Processing

If a group member is a name of an account, or some other group in the same Domain, then the group member list is automatically updated if that account or group is renamed or removed.

Renaming Groups

If you want to rename a group, open its Settings page with a Web browser, and enter a new group name into the New Group Name field. Click the Rename Group button.

If there is no other object with the same name as the specified new group name, the group is renamed and its Group Settings page should reappear on the screen under the new name.

New Group Name:

Removing Groups

If you want to remove a group, open its Settings page with a Web browser, and click the Remove Group button.



Forwarders

CommuniGate Pro domains can contain forwarder objects. A Forwarder is just some E-mail address associated with the Forwarder name.

Forwarders work exactly as the Router alias records: a Forwarder "jim" in the domain client1.com that contains the E-mail address "john@otherdomain.dom" acts as the Router record:

Relay: <jim@client1.com> = john@otherdomain.com

Forwarders allow the Server administrator to keep the Router table small, and they can be set by a Domain Administrator, while the Router can be modified by the Server Administrator only.

Specifying Domain Forwarders

Forwarders in any domain can be created by the Server Administrator if the administrator account has the Can Modify All Accounts And Domain Settings access right.

A Domain Administrator can create and remove forwarders only if the CanCreateForwarders access right is granted to the administrator account.

To specify Forwarders, click the Forwarders link on the domain Account List page. The list of Forwarders and their names appears:

Name	Forward to

To create a new Forwarder enter the new forwarder name into the left column and the address to re-route that mail to - into the right column, then click the Update button.

To delete a Forwarder delete its name or its forwarding address and click the Update button.



Mailboxes

CommuniGate Pro [Accounts](#) contain one or several *mailboxes*. Each mailbox has its own unique name and can contain messages. The [POP](#), [IMAP](#), and [WebUser Interface](#) modules provide [access](#) to Accounts and their mailboxes.

Several storage formats can be used for CommuniGate Pro mailboxes. A multi-mailbox Account can contain mailboxes stored in different formats.

Each Account always has the INBOX mailbox. Any message delivered to a CommuniGate Pro Account is stored in its INBOX mailbox - unless some Automated Processing Rules instruct the Server to store the message in a different mailbox.

Mailbox Names

When an Account is created, its INBOX mailbox is automatically created. The system and/or domain administrator can specify additional mailboxes to be created at that time.

A user can create a mailbox using an IMAP mailer application or using the [WebUser Interface](#).

Mailboxes can be "nested": for any mailbox "A" you can create a sub-mailbox "B" - in the same way as you can create a file directory inside some other file directory. The CommuniGate Pro Server uses the slash (/) symbol as the hierarchy separator:

INBOX/important

is the name of the submailbox important "inside" the INBOX mailbox.

Unlike many other servers, CommuniGate Pro allows you to store messages in some mailbox X and at the same time create submailboxes X/Y, X/Z for that mailbox. This feature is implemented by providing two "invisible" mailbox entities - one for storing messages, one - for serving as a "directory" for the nested mailboxes. The "directory" entity is created automatically, as soon as you try to create the first submailbox. You can, though, create the "directory" entity without creating the

"mail storage" entity: use the ABCDEF / name as the new mailbox name to create only the directory entity with the ABCDEF name. The name ABCDEF will be listed, but will not be "selectable" - and you will not be able to store messages in the ABCDEF mailbox. You can later create the regular ABCDEF mailbox and the "storage" entity for your ABCDEF mailbox name will be added.

It is impossible to delete the INBOX mailbox. You can rename the INBOX mailbox, though. In this case a new empty INBOX mailbox will be created automatically.

Mailbox names are case-sensitive. Some file systems (NTFS, for example) provide case-insensitive file naming conventions. When these file systems are used for CommuniGatePro account/mailbox storage, the mailbox names are still case-sensitive, but you cannot create two mailboxes with names that differ in case only. The INBOX mailbox name is an exception: it is always a case-insensitive name.

Mailbox Access Control Lists

The CommuniGate Pro Server maintains an Access Control List (ACL) for every mailbox it creates. Each element of the Access Control List contains a name and a set of mailbox access rights granted to that name.

The Access Control Lists are used to control the [Foreign Mailbox Access](#) feature that allows one account user to access mailboxes in other accounts.

The name in an ACL element can be:

Some other account name (as *accountname*)

This ACL element specifies the access rights granted to that Account user.

full account name (as *accountname@domain*)

This ACL element specifies the access rights granted to the user of an Account in a different CommuniGate Pro Domain.

anyone

This ACL element specifies the access rights granted to everybody.

-accountName

the ACL element specifies the access rights revoked from the specified Account user.

Account owners always have all access rights to all mailboxes in their own Accounts. For any other *someaccount* user, the effective access rights are checked. The effective access rights are the mailbox access rights granted to the *anyone* name, plus the rights granted to that *someaccount* name, minus the rights "granted" to *-someaccount* name.

A Server Administrator with the All Accounts and Domains [access right](#) has all access rights for all server mailboxes. A Domain Administrator with the CanViewMailboxes access right has all access rights for all mailboxes in his/her Domain.

The following mailbox access rights are supported:

l (Lookup)

If you grant a user the Lookup access right, that user will be able to see this mailbox when it asks the Server to list all mailboxes in your Account.

r (Read/Select)

If you grant a user the Read access right, that user will be able to open (select) this mailbox and see (read) the messages in this mailbox.

s (Seen)

If you grant a user the Seen access right, that user will be able to mark messages as read (seen). Usually a message is automatically marked as seen when a user reads it. But if this access right is not granted to a user reading the mailbox, the mailbox message "seen" status will not be changed.

w (Write/Flags)

If you grant a user the Write access right, that user will be able to set message flags: i.e. to mark messages as answered or "flagged", and to reset the message flags.

d (Delete)

If you grant a user the Delete access right, that user will be able to mark messages as deleted and to compress the mailbox, removing all its messages marked as deleted.

i (Insert)

If you grant a user the Insert access right, that user will be able to append messages to this mailbox and to copy messages from other mailboxes into this one.

p (Post)

This access right is not used by modern mailers.

c (Create)

If you grant a user the Create access right, that user will be able to create new submailboxes "inside" this mailbox.

a (Administer)

If you grant a user the Administer access right, that user will be able to modify the ACL for this mailbox.

When a submailbox is created, it inherits the ACL of the "parent" mailbox. This means that if you create the INBOX/sales mailbox, it is created with the same ACL as specified for the INBOX mailbox.

The Access Control Lists can be set and modified using either the [WebUser Interface](#) or using a decent [IMAP](#) client.

In order to be able to delete a foreign mailbox, a user should have:

- the Create access right for the outer (parent) mailbox, and
- the Delete access right for the specified mailbox.

In order to be able to rename a foreign mailbox, a user should have:

- the Create access right for the outer (parent) mailbox of the original mailbox, and
- the Delete access right for the specified original mailbox, and
- the Create access right for the outer (parent) mailbox of the new mailbox (mailbox name).

When granting access rights, the real Account names, not Account Aliases should be used. If an Account `j.smith` has two aliases `john.smith` and `jonny`, the access rights should be granted to the name `j.smith`.

Mailbox Formats

CommuniGate Pro stores received messages in Account mailboxes. The server supports several mailbox formats, and the mailbox type is defined by the mailbox file (or directory) name extension.

For single-mailbox accounts, the mailbox type is specified when the account is created.

Each multi-mailbox account has a [setting](#) that specifies the default type for all new mailboxes created in this account. A user can explicitly specify the mailbox type creating a mailbox in a multi-mailbox account: if the mailbox name is specified as *name.extension*, then the mailbox *name* of the *extension* type is created.

The .mbox Format

The mailbox files with this extension store messages in the legacy BSD mailbox format. Each message in the mailbox is preceded with the a *From-line*:

```
From <return-path>(flags-UID) time stamp
```

This is the same format as one used in legacy mail systems, but with a "comment" added after the return-path part. The .mbox format remains compatible with legacy applications (local mailers), and at the same time it allows the CommuniGate Pro Server to store the required message information (message status flags and the unique mailbox message ID).

If a mailbox file has been copied from an old system, or when it is used as an [External INBOX](#) and old applications can add messages to this mailbox, some messages may have no "comment;" part. CommuniGate Pro allows a user to work with such messages, but it does not store message flags if they were modified, and it does not remember the message UIDs between sessions. The simplest solution is to copy such messages to a different mailbox and then copy them back to the original mailbox - the copy operation places the correct information into the *From-line*.

When a message is being stored in the .mbox-type mailbox, all message lines are checked. If there is an empty line followed with the line starting with the letters From, the '>' symbol is inserted before the letter F,

The .mdir Format

Mailboxes with this extension are file directories. Each mailbox message is stored as a separate file in the mailbox directory.

The message file name has the following format:

iiii-flags-timestamp

where *iiii* is the message unique ID, *flags* are the message status flags, and the *timestamp* is the message *internal time stamp* - the time (GMT) when the message was added to the mailbox, in the *yyyymmddhhmmss* format.

Note:

On the Unix platforms, the .mdir mailboxes implement the *shared storage model*: if the same message is directed to many accounts/mailboxes, only one message file is created, and a *hard link* to that file is placed into each mailbox directory. When a message is removed from all mailboxes, the file is automatically deleted by the OS.

Note: most of freeware mail systems use either the mbox-like or mdir-like formats, and designers of those systems make various claims about the advantages of the formats they have selected. It is very important to remember that:

- CommuniGate Pro does not use OS or file system features (locks) to provide multi-access to mailboxes. CommuniGate Pro mailboxes in all formats can be accessed by several clients at the same time, and all synchronization is implemented in the CommuniGate Pro Mailbox Manager that works in the same way for all mailbox formats.
- CommuniGate Pro uses efficient mechanisms to parse mailboxes, so many claims about various mailbox formats being 'slower' or 'faster' than other formats usually do not apply to CommuniGate Pro installations.

- CommuniGate Pro allows users to create mailboxes in different formats, even within the same Account.

Note: the `.mbox` format is more efficient than `.mdir` in most cases, this is why this format is used as the default one. The `.mdir` format is recommended only for those mailboxes that contain many (20 or more) large (100K or more) messages. If a user has a `Proposals` mailbox where she stores all messages with attached documents, each 50-70K in size, then this mailbox may work faster if it is created in the `.mdir` format.

Creating Mailboxes

Every account has a [setting](#) that specifies the default format for new mailboxes that can be created in this account.

The account user can explicitly specify the storage format for a new mailbox by adding the format extension to the new mailbox name. If a user tells the CommuniGate Pro Server to create the `newmailbox.mdir` mailbox, the `.mdir`-formatted mailbox `newmailbox` is created.

Mailbox Subscription

The CommuniGate Pro Server allows an account user to *subscribe* to some mailboxes. The account mailbox subscription is a simple list of mailbox names. This list is not used by the Server itself - the Server just stores one subscription list for each account.

Many [IMAP](#) mailers use the account subscription list and show only the mailboxes the account is subscribed to. The [WebUser Interface](#) can also be configured to show only the subscribed mailboxes.

You can modify the account subscription either via a decent IMAP mailer, or using the WebUser Interface.

You can use the account mailbox subscription to make some not-so-decent IMAP mailers access foreign mailboxes: make sure that your IMAP client is configured to use the account mailbox subscription, and add the desired [foreign mailbox](#) name into the subscription list.

Note: Some IMAP mailers tend to rebuild account subscription lists: they empty the subscription, and then subscribe you to all mailboxes in your own account.

The account mailbox subscription is stored in the account `.info` service file.

Mailbox Aliases

Many IMAP clients (such as Microsoft Outlook and Outlook Express) cannot handle [foreign mailboxes](#) directly, and they cannot use the Account [mailbox subscription](#) to access foreign mailboxes.

Mailbox aliases can be used to let these IMAP clients access foreign mailboxes.

Mailbox alias is a name associated with some [foreign] mailbox name. For example, you can create a mailbox alias `salesBox` for the `~sales/INBOX` mailbox name. You will see the `salesBox` mailbox in your IMAP mailer, but in reality this will be the `INBOX` mailbox in the `sales` account.

Mailbox aliases can be created only on the topmost level of the account mailbox hierarchy, that means that the mailbox alias name cannot contain the slash (`" / "`) sign.

Mailbox aliases can contain just the name of the foreign account (`~accountName`). Such an alias provides access to all accessible mailboxes in that foreign account. The mailbox alias itself is presented as an unselectable mailbox name.

Sample configuration:

The owner of the account `chief` has granted "lookup" and other access rights for his mailboxes `INBOX` and `Pending` to the `assistant` account.

The user `assistant` has created the mailbox alias `boss` pointing to `~chief`.

When the user `assistant` connects to her account using any IMAP client or the WebUser Interface, she sees all her own mailboxes, the unselectable mailbox `boss`, and also the `boss/INBOX` and `boss/Pending` mailboxes.

If the user `chief` creates a new mailbox `Urgent` in his account and grants access rights for that mailbox to the `assistant` account, the user `assistant` will immediately see the new mailbox as the `boss/Urgent` mailbox.



Web Files

CommuniGate Pro [Accounts](#) can contain *WebFiles* - a set of HTML, JPEG, and other files that are available via the HTTP protocol. This file set is used as the *Personal Web Site* for the Account user.

The Account owner or an administrator can add files to the Personal Web Site, rename them and remove them.

If the Account and its Domain have the WebSite Service enabled, anybody can retrieve Personal Web Site files using any HTTP browser.

A Personal Web Site can contain nested folders (file directories).

The total number of files and folders and the total size of all Personal Web Site files is limited by the special Account Limits settings.

HTTP Access to Personal Web Sites

CommuniGate Pro allows each user to be presented on the World Wide Web with a personal Web Site. The URL for the *accountname@domainname* Account Web Site is:

`<http://domainname:port/~accountname>` where the *port* is the [WebUser port](#).

For example, the *jsmith@client1.com* account has the Personal Web Site at:

`<http://client1.com:8100/~jsmith>`

Personal Web Sites use the same HTTP port as the [WebUser](#) Interface (the port 8100 by default).

The ~ is the default prefix for the Personal Web Sites. It can be changed to a different string, and it can be changed to an empty string.

To modify the HTTP Access options, use the WebAdmin Interface and open the WebUser page in the Settings realm:

Personal Web Sites

Site Prefix:

All Routing Rules discussed in the [Access](#) section apply to the Personal Web Site URLs, so account and domain aliases can be used in the Personal Web Site URL.

Personal Web site can be accessed without a prefix, using just the server part of the URL string. When the CommuniGate Pro server receives an HTTP connection on the its [WebUser port](#), it uses the special [Domain Routing](#) procedure.

If the domain name `user.domain.com` has a DNS A-record pointing to the IP address of the CommuniGate Pro server, and the CommuniGate Pro Router has the following record:
`<LoginPage@user.domain.com> = userA@domainB.com`
and the Account `userA` exists in the CommuniGate Pro Domain `domainB`, then the URL `http://user.domain.com/` can be used to access the Personal Web Site of the `userA@domainB.com` Account.

The home (default) page of a Personal Web Site should have the `default.html` name. This means that when the file name is not specified explicitly, the `default.html` name is assumed. There can be no file with the `index.html` name - that name is used to access the Personal WebSite Management forms.

Private Folder

If the Personal Web Site contains the folder with the name `private`, then files in that folder are available only to the Account owner and Administrators.

The `private` folder can be used as a repository for any type of documents - the user can access them from anywhere, using any browser.

HTML-based Management

Users can manage their Personal Web Sites using any browser. There are two methods to access the Personal WebSite administration pages:

- By opening a [WebUser Session](#), and using the WebSite link in the WebUser Interface navigation panel.
- By opening the `Index.html` file in their own Personal WebSite:
`http://domain.dom:8100/~username/Index.html`

The browser will present a Login dialog box, and the user should enter her account name and password in order to open the WebSite administration page.

Server administrators with the All Domains and Accounts [Access Right](#) and Domain administrators with the CanAccessWebSites access right can access WebSite belonging to other users. They can use the same URL, opening the Index.html file, but they should provide their own account names and passwords.

Server and Domain administrators can access other users Personal WebSites using the WebAdmin Interface: the Account management pages have the WebSite link in their navigation panels.

All management methods use similar HTML pages for WebSite administration:

Subdirectory documents			UP
Marker	File Name	Size	Modified
	report.txt	488	20:52:49
	myDocs	-->	
This Folder:	2	488	
Totals:	5	976	
Limits:	Unlimited	30720	

Click the Browse button and select a file you want to upload to the WebSite. Click the Upload File button to upload the file. Its name should appear in the list.

Select the checkboxes to mark the files you want to remove from the Personal WebSite and click the Delete Marked button. The selected files will be removed.

Type in a name and click the Create Folder button to create a folder (sub-directory) in the Personal Web Site.

Click the file name link to open the file. Click the folder name link to open the subdirectory. When a subdirectory is opened, its name is displayed on the top of the file list. Click the UP link to open the parent subdirectory.

The This Folder line displays the total number of files and folders, and the total size of all files in the opened folder. The Totals line displays the total number of files and folders, and the total size of all files in the Personal Web Site. The limits line displays the specified maximum number of files and folders and the specified maximum total file size for this Personal Web Site.

HTTP-based Management

Personal Web Sites can be modified using the HTTP 1.1 PUT, DELETE, and MOVE methods. Some HTML design tools (such as Netscape Composer) can use these methods to upload files to the server. These HTTP requests should contain the Authentication information: the Account name of the Personal Web Site owner or the Account name of a Server/Domain Administrator, and the password for that Account.

FTP-based Management

Personal Web Sites can be modified using the CommuniGate [FTP module](#). When an Account user connects to the FTP module, the FTP "root directory" as well as the "current directory" are set to the topmost directory of the Account



Account Data

The CommuniGate Pro server stores account information in several places. Most of the information is stored in the account service files, while some account data is grouped in the domain files.

Domain Files

For each CommuniGate Pro domain, a file directory is created in the `Domains` subdirectory inside the *Server base directory*. The directories have the same names as the domains.

For the main domain, the `Accounts` file directory is created inside the base directory.

Inside each domain file directory, a `Settings` file directory is created. This directory contains the following files:

`Access.settings`

This file has the dictionary format, and contains the names of the users that have administrative access rights to the server or to the domain, and the list of the granted rights. By storing all administrative access rights in one location the CommuniGate Pro Server makes it easier to maintain server security. Only the `Access.settings` file stored in the main domain `Settings` directory can contain the server-level access rights. All other files can contain only the domain-level administrative access rights.

`Domain.settings`

This file contains the [domain settings](#).

`RPOP.data`

This file contains the information about all individual [RPOP](#) (remote POP) accounts that should be polled on behalf of the domain users.

`Template.settings`

This file contains the Account Template for this domain and provides the default account

settings for new accounts in this domain.

`Aliases.data`

This file contains the list of all account-level aliases specified for the domain accounts.

`LISTS`

This directory contains files with the information about the [mailing lists](#) created in the domain.

`WebUser`

This directory contains HTML files that customize the [WebUser Interface](#) to the domain accounts and mailing lists.

The [Domain Administrator](#) can place HTML and other files into this directory (*publish* them) using any HTML composer application that supports the POST, DELETE, and MOVE HTTP methods.

Account Files

Every CommuniGate Pro account contains at least one (INBOX) mailbox file, and at least two service files. Service files have special file name extensions, and `.settings` and `.info` service files always exist. The `.settings` file contains account settings, while the `.info` file contains volatile account information, such as mailbox sizes, last UIDs used in each mailbox, etc. Since the `.info` file is being modified rather often, the CommuniGate Pro server is built to survive `.info` file corruptions. For example, if the mailbox last UID information is corrupted, the server rescans the mailbox and restores the correct mailbox info.

The account files are located in the domain file directory or in its subdirectory (see the [Domains](#) section for the details).

For a multi-mailbox account, a directory with the account name and `.macnt` extension is created, and all account files are stored in that directory. The account service files are stored as `account.extension`. The INBOX mailbox is stored as the `INBOX.mailboxType` file. Example: for the multi-mailbox account John, the `john.macnt` directory is created, and the files `INBOX.mbox`, `account.settings`, `account.info` are placed in that directory.

For a single-mailbox account, the INBOX mailbox is created as a file in the domain file directory or its subdirectory, and it has the `accountName.mailboxType` file name. The account service files are stored in the same directory as `accountName.extension`.

Example: for the single-mailbox account John, the `john.mbox`, `john.settings`, and `john.info` are placed into the domain file directory.

Personal Web Site

The Personal Web Site files are stored in the account service directory with the `.web` file name extension.

Public Information

The CommuniGate Pro Server allows a user to specify a set of attributes (such as phone number, home Web page, job title, etc.) that becomes available to the public via the [LDAP](#) interface to the Central Directory.

The account Public Information is stored in the dictionary format in an optional account service file with the `.public` extension.

Users can update their Public Information attributes via the [WebUser Interface](#), or via the [ACAP](#) protocol.

Preferences

The CommuniGate Pro Server allows a user to store various application preferences in the account and to retrieve them from any computer. This service, known as *roaming service* is available via both [LDAP](#) and [ACAP](#) protocols.

Account preferences are stored in the optional `.prefs` account service file.

Netscape Roaming

The [Netscape](#)® Communicator product can use any advanced HTTP server to store and restore its settings.

To use this Netscape Roaming service, the user should specify the following URL as the *Roaming Server URL*:

`http://domain[:port]/Settings/`

where *domain* is the user domain (or the main CommuniGate Pro domain), and *port* is the CommuniGate Pro [User HTTP](#) port.

The actual account name is not specified in this URL. Access to the `/Settings/` realm requires authentication, and the CommuniGate Pro Server opens the account specified in the browser username/password dialog box.

If the URL used contains the correct domain name of the target account, the account name can be specified as a simple name (i.e. without the domain part), but if the URL contains the name of some other CommuniGatePro domain (because the target domain does not have any A-record), the account name should be specified along with the domain name, i.e. instead of the `jsmith` string, the `jsmith@domain.com` string should be used in the Netscape username/password dialog box.

The Netscape Settings are stored as separate files inside the account service directory with the `.roaming` extension.



Message Transfer

One of the main functions of the CommuniGate Pro Server is message transfer. Acting as an MTA (Message Transfer Agent), the server accepts messages from various sources (modules, internal kernel components, etc.), and delivers (transfers) them to remote or local destinations using the same or different modules.

While all submitted messages are stored as individual files in the Queue directory inside the CommuniGate Pro "base directory", each message can be enqueued into several different queues (if it has several recipients). Each communication module can maintain one or several logical queues. For example, the SMTP module maintains one queue for each Internet domain.

The CommuniGate Pro Server has the following set of message sources:

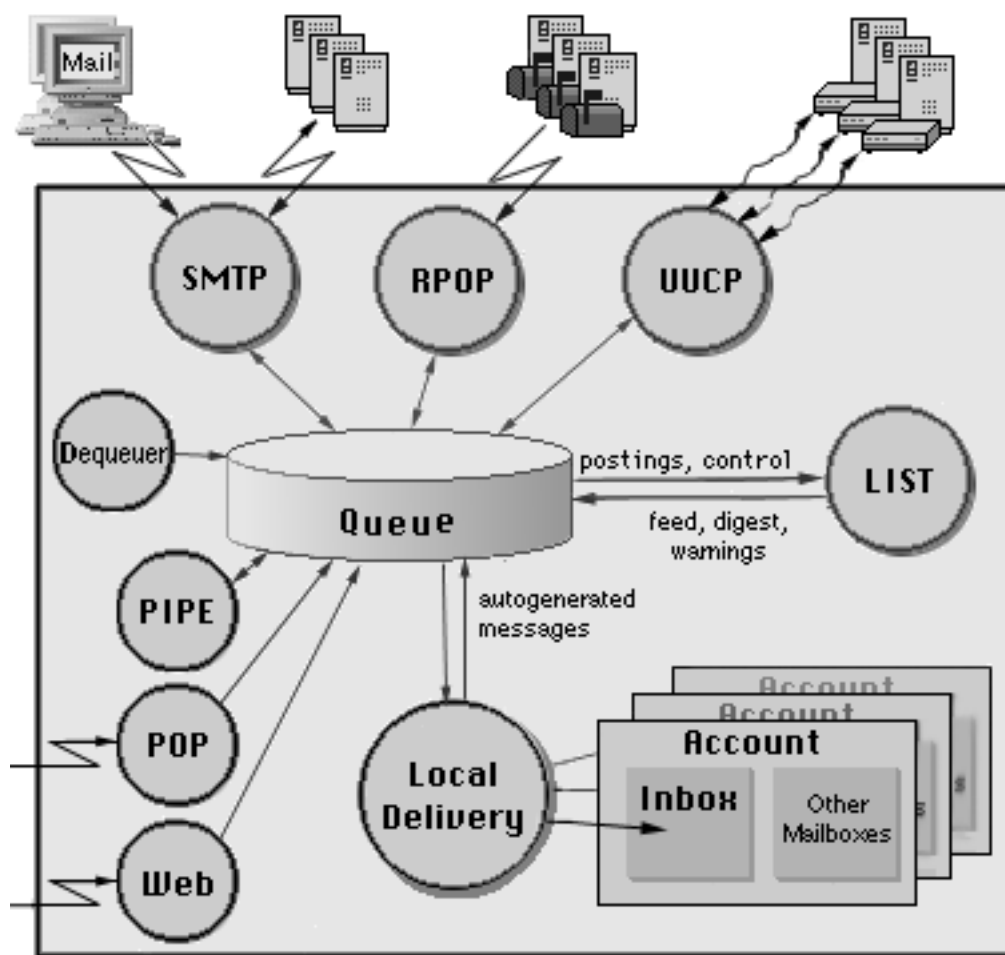
- the [SMTP module](#) submits the messages received from mailer applications and from other mail servers via the Internet;
- the [UUCP module](#) submits the messages received from "uucp neighbors" via modem and/or TCP/IP connections;
- the [RPOP module](#) submits the messages retrieved from remote POP servers;
- the [LIST module](#) submits the messages to be distributed to mailing list subscribers, along with various list administration messages;
- the [Local Delivery module](#) submits the messages generated with the [Automated Mail Processing Rules](#);
- the [PIPE module](#) submits the messages received from external applications via interprocess communication channels, and the messages generated and stored in the special Submitted directory;
- the [POP module](#) submits the messages received from certain mailer applications employing the XTND XMIT protocol extension;
- the [Web E-mail module](#) submits the messages composed within Web browsers;
- the [Dequeuer kernel component](#) generates and submits various delivery notification messages.

The CommuniGate Pro Server transfers messages to the following destinations:

- the [SMTP module](#) transfers messages to other SMTP mail servers via the Internet;

- the [UUCP module](#) transfers messages to "uucp neighbors" via modem and/or TCP/IP connections;
- the [LIST module](#) accepts and processes messages with mailing list postings, and with various list administration requests;
- the [Local Delivery module](#) transfers messages to local user mailboxes;
- the [PIPE module](#) transfers messages to external applications via interprocess communication channels;
- the kernel itself transfers (discards) messages routed to NULL or ERROR addresses;

The following diagram illustrates the message flow inside the CommuniGate Pro server.



Submitting Messages

All messages are created as temporary files. They are stored in the `Queue` directory as files with the `.tmp` extension. A module or a kernel component stores message envelope and the message itself in such a file and then submits it to the kernel for processing.

The message envelope is a set of text lines. Each line specifies either the message return-path, or one message recipient address, or message delivery options.

If a module fails to compose a message (for example, an SMTP connection breaks during message transfer), the module discards the temporary file, and the kernel deletes it.

When a message is completely composed and submitted to the kernel, the file extension is changed to `.msg` and the message is scheduled for processing.

When a system restarts, a kernel module checks all files with the `.msg` extension stored in the `Queue` directory and resubmits them for processing.

You can use a Web browser to configure the Temporary Files manager. Open the `Obscure` page in the `Settings` section.

Temp File Options	
TempFiler Log:	Recycle Temp Files:

TempFiler Log

This setting specifies what kind of information the Temporary Files manager should put in the Server Log. Usually you should use the `Failures` (file system error reports) level. But when you experience a problem with some module submitting messages, you may want to set this setting to `Low-Level` or `All Info`: in this case file i/o operations will be recorded in the System Log as well. When the problem is solved, set the TempFiler Log setting to its regular value, otherwise your System Log files will grow in size very quickly.

Recycle Temp Files

Enable this option to improve performance of your system under heavy load.

The Temporary Files manager Log records are marked with the `TEMPFILE` tag.

Routing

When a message is submitted for processing, a kernel component examines its envelope information. Each recipient address is parsed and passed to the [Router](#) component. The Router component decides which module or kernel component should process each recipient address.

Enqueueing

When all recipient addresses are parsed and routed, the Enqueuer component applies the [Server-Wide Rules](#) to the message. Then it passes the message to the modules specified with the Router component.

Communication modules do not process messages immediately, but enqueue them into the module-specific queues. The SMTP module creates and maintains a queue for each Internet domain, the UUCP module maintains one queue for each "uucp neighbor" system, the Local Delivery module creates and maintains a queue for each local account, etc.

You can use a Web browser to configure the Enqueuer component. Open the Queue page in the Settings section of the WebAdmin Interface.

Message Queue	
Message Enqueuer	
Log:	Processors:

Enqueuer Log

Use this setting to specify what kind of information the Enqueuer component should put in the Server Log. Usually you should use the `Failures` (file system error reports) level.

The Enqueuer component Log records are marked with the `ENQUEUER` tag.

The records created when applying the Server-Wide Rules are marked with the `ENQUEUERRULES` tag

Processors

Use this setting to specify the number of Enqueuer processors (threads). Usually one Enqueuer thread is enough even for a heavy-loaded server. You should increase the number of Enqueuer processors if:

- you have specified many Server-Wide [Rules](#);
- Server-Wide Rules use the `Execute` action to start external programs;
- the [content filtering](#) or anti-virus engine is enabled.

Delays and Suspensions

When a communication module fails to transfer a message, it uses the kernel queue management component to delay processing.

- a module can delay an entire queue: for example, the SMTP module can delay a queue created for an Internet domain, if it cannot connect to that domain or its relays;
 - a module can delay an individual queued message: for example, the SMTP module can delay a message if the receiving host rejects this particular message (transition failure);
 - a module can delay an individual recipient address in a queued message: for example, the SMTP module can delay an address if the receiving host rejects that particular address (transition failure).
-

Dequeuing

When a communication module transfers a message or when it rejects a message because of a fatal error, it removes the message from the module queue. The module composes a delivery report and passes it to the Dequeuer kernel component.

The Dequeuer component processes delivery information. If requested, it composes Delivery Status Notification (DSN) messages and submits them back to the system for delivery to the original message sender. When a message has several recipients, the Dequeuer module may choose to delay DSN generation, so each DSN message can contain reports about several recipients.

When all message recipients are processed and the message is dequeued from all queues, the Dequeuer component removes the message file from the Queue directory.

You can use a Web browser to configure the Dequeuer component. Open the Queue page in the WebAdmin Settings section.

Message Dequeuer	
Log:	Reporting Delay:
Processors:	On Failure, Return:
Copy Failure Reports to:	

Dequeuer Log

Use this setting to specify what kind of information the Dequeuer component should put in the Server Log. Usually you should use the Major & Failures (delivery reports) level.

The Dequeuer component Log records are marked with the DEQUEUER tag.

Processors

Use this setting to specify the number of Dequeuer processors (threads). Usually one Dequeuer

thread is enough even for a heavy-loaded server. Only if your Server performs some kind of special message processing and has to generate a lot of DSN messages, should you use several Dequeueer threads.

Reporting Delay

Use this setting to specify the maximum delay between the moment when a message was transferred or failed and the moment when a delivery report is generated. The more the delay, the more reports can be placed in one DSN message. A DSN message is generated immediately after the last message recipient is processed.

On Failure, Return

Use this setting to specify what portion of a failed message should be included into the DSN (error report) message.

- If the sender has not specified this option explicitly, and the `headers by default` option is selected, only the failed message headers will be returned;
- If the sender has not specified this option explicitly, and the `body by default` option is selected, the entire failed message will be returned;
- If the `always headers` option is selected, only the message headers are included into the DSN message, even if the message sender has specified that the entire message should be returned on failure.
- If the `always body` option is selected, the entire message is included into the DSN message, even if the message sender has specified that only the message headers should be returned on failure.

Copy Failure Reports

When this option is enabled, all error messages generated with the CommuniGate Pro Dequeueer are sent to both the failed message return-path and to the specified E-mail address.



Router

This chapter is for advanced administrators only. In most situations the default routing methods are enough. Only if your Server is working as a message relay for other systems, or it has to support several domains, or if you want to use sophisticated routing schemes, should you read this chapter.

When the Server processes a submitted message, it extracts the information about recipients from the message "envelope" and places the message into the queues of the communication modules. The part of the Server kernel that does this job is called the Router.

Domain and Local Parts of E-mail Addresses

Each E-mail address consists of two strings: a domain name and a local part. Usually, an address looks like `xxxx@yyyyy`, where `yyyyy` is the domain name (the unique name of the recipient mail system) and `xxxx` is the local part, i.e. a user name or an account name in that system.

An E-mail address can be more complicated, for example, an address can include some path information:

`<@zzzz:xxxx@yyyyy>` or `zzzz!yyyyy!xxxx` or `xxxx%yyyyy@zzzz`

These addresses specify that a message should be sent to the system `zzzz` first, and then that system should deliver it to `xxxx@yyyyy` (to the account `xxxx` on the system `yyyyy`).

When the Router parses an address, it extracts the name of the system the message should be delivered to. It becomes the domain name part of the address. The rest of the address is placed into the local part, i.e. the local part defines the recipient when the message is delivered to the system specified with the domain name. In the examples above, the domain name part is `zzzz`, while the local part is `xxxx@yyyyy`.

See the RFC822 and related documents for details on E-mail address formats.

Note: if the local part contains a complex address (i.e. it also contains domain name(s) and a local part), the local part is presented in the '%' notation: `local%domain1%domain2`. This information can be used for sophisticated [aliasing](#) methods.

Mail Domain Name

When the domain name is extracted from an address, the Router compares it against the domain name of the Server (set in the [General](#) settings). If they match, the domain name is set to an empty string. When the domain part becomes an empty string, the Router restarts processing with the local part, trying to divide it into the domain and local parts again.

For example, if the Main Domain name of your Server is stalker.com, then the following addresses will be converted as shown:

E-mail address	local part	domain part
support@stalker.com	support	stalker.com
	-- routed -->	support
<@stalker.com:sales@gamma.com>	sales@gamma.com	stalker.com
	-- routed -->	sales@gamma.com
	-- routed -->	sales gamma.com

Multiple Domains. MX records

Your server may receive and process E-mail messages directed not only to the Main Server Domain set in the [General settings](#), but sent to other domains, too. If your server has to process mail for a certain domain, you should ensure that the messages sent to that domain comes to your server.

For example, your server (mycompany.com) may act as an Internet-uucp relay for the client uucp systems client1, client2, and client3. Each client has its own domain name (client1.com, client2.com, and client3.com), and you have configured your Router to ensure that all messages sent to the client1.com domain will be routed to the uucp host client1, etc. But you should also ensure that when a message is sent to the client1.com domain, that message is routed to your server (mycompany.com).

Internet mail routing is controlled with the DNS - Domain Name System. Domain Name Servers contain the information about each domain name. So, when the client registers the client1.com domain name, the client should ask to create an MX (mail exchange) DNS record pointing to your domain - mycompany.com.

Routing Table

When an address is parsed and the domain part is extracted, and that name is not the Main Domain Name of your Server, the Router checks the routing records in the Routing Table.

Open the Router page in the Settings section of the WebAdmin Interface to manage the Routing Table:

```
fax.stalker.co      = stalker.com          ; -> to the same domain
hq.stalker.com      = newhq.stalker.com ; -> to some other server
Relay:*.test.com    = stalker.com          ; aaa.test.com, bbb.test.com
; just a comment line
<sales>             = john                 ; simple alias
<sales@client1.com> = sales-client1        ; simple foreign alias
<info@client1.com>  = info@otherhost.com; account -> other account
<*@client2.com>     = *.cl2                ; sales@.. -> sales.cl2
test.com            = Unified.local        ; unified Domain-Wide Account
```

Add to Non-Qualified Domain Names

Each line in the Routing Table is a routing record. A routing record contains the left part, the equals sign (=) and the right part. The semicolon sign (;) can be used to place a comment after the right part of a routing record. A comment line can be added to the Table by inserting a line starting with the semicolon sign.

The Router takes a parsed E-mail address (i.e. the domain and local parts of the address) and uses the Table, scanning its records from top to bottom. If an applicable record is found, it is applied as described below and the modified local and/or domain parts are processed with the Router from the beginning.

Any Routing record can contain the Relay: prefix (can be shorten to R:), the NoRelay: prefix (can be shorten to N:) or the RelayAll: prefix. See the [Protection](#) section for the details. If no prefix is specified, the Relay: prefix is assumed.

Log

The CommuniGate Pro Router has the Log Setting that you can modify using a Web browser. Use this setting to specify what kind of information the CommuniGate Pro Router should put in the System Log. Usually you should use the Major (address routing) level. But when you experience problems with the Router, you may want to set the Log setting to Low-Level or All Info: in this case more low-level information about the Router activity will be recorded in the System Log as well. The Router component records in the System Log are marked with the ROUTER tag.

Domain-Level Routing Records

If the left part of a routing record contains a domain name, the record specifies domain-level routing.

When some address is being processed and the domain name matches a domain name specified in such a record, the domain part is substituted with the right part of the routing record.

Example:

```
hq.stalker.com = twisted.stalker.com
```

All messages directed to the domain name hq.stalker.com will be redirected to the domain twisted.stalker.com. The Router restarts, trying to find a path to deliver messages to twisted.stalker.com.

A routing path can specify relays.

Example:

```
hq.stalker.com = hq.stalker.com@relay.stalker.com
```

All **m**essages directed to the domain name hq.stalker.com will be redirected to the domain relay.stalker.com, and then, from that machine, to the domain hq.stalker.com.

If mail to several domains should be routed in the same or similar way, you may use the asterisk sign as the wild-card symbol.

Example:

```
*.old_company.com = new_company.com
```

In this case messages to all the domains ending with .old_company.com will be routed to the domain new_company.com, with the local parts (user names) unchanged.

Very often this type of routing is used to process all subdomains of the some domain.

Example:

```
*.mycompany.com = mycompany.com
```

If the mycompany.com is the Server's Main domain name, then this routing record makes the server process messages sent to all subdomains of its Main Domain as messages sent to the Main domain, the address user@mail.mycompany.com will be processed as the user@mycompany.com address.

The asterisk symbol can be used in the right path, in this case it is substituted with the symbols matching the wildcard symbol in the left part.

Example:

```
*.old_company.com = *.new_company.com
```

When such a routing line is entered and a message comes for the domain

host5.old_company.com, it is routed to host5.new_company.com.

In most cases, the wildcard symbol is the first symbol in the domain name, but it is allowed to be used anywhere:

```
system-*.mycompany.com = uu*.uucp
```

This routing line will redirect system-abc.mycompany.com to uuabc.uucp.

Only one wildcard symbol is allowed in one routing record.

Besides domain-level routing records, routing for domains can be specified using [Aliasing](#) records (see below). Records for [Unified Domain-Wide Accounts](#) are domain-level routing records, too.

Account-Level (Alias) Routing Records

If the left part of a routing record contains an E-mail address in the angle brackets (< and >), the record specifies an alias - a routing rule for a specific address.

When an E-mail address is parsed and the Router scans the Table records, it compares the address domain part with the domain part of all alias records met. If the domain parts match, the Router compares the local part of the address with the local part of the alias record address. The local part of the alias record address can contain a wildcard symbol (*). If both domain and local parts match, the right part of the alias routing record is used as the new address. The Router restarts from the beginning, parsing and processing this new address.

Note: Because the Server Main Domain Name in the parsed address is immediately replaced with an empty string, alias records that should apply to addresses in the Main Domain should not contain any domain part at all.

In the all examples below mycompany.com is the Server Main Domain name.

Example:

```
<sales> = Bill
```

in this case, all messages to sales@mycompany.com will go to Bill, as if they were sent to Bill@mycompany.com

Note: if there is an alias for the local name xxxxx, there is no need to actually register the real xxxxx account with the Server. Additionally, that real account would be useless, since no message will ever be stored in that account: everything directed to the xxxxx name is routed elsewhere.

Note: the Router Alias record:

```
<sales@mycompany.com> = Bill
```

will never work: the alias records should NOT contain the name of the Server Main Domain.

The right side of an alias record can be any E-mail address.

Example:

```
<sales> = Bill@thatcompany.com
```

All messages directed to sales@mycompany.com will be directed to Bill@thatcompany.com.

The Router takes the new address, extracts the domain name (thatcompany.com) and local (Bill) parts, then the Router restarts trying to find a route to thatcompany.com.

You can use the wildcard symbol (*) in the local part of the alias records. The same symbol can be used in any part of the right-side address to specify substring substitution.

Example:

```
<dept-*> = postmaster@*-dept.mycompany.com
```

This record will redirect all messages sent to dept-sales@mycompany.com to the user postmaster at the sales-dept.mycompany.com department mail server.

You can use Router Alias records to reroute mail sent to some of the your Server Secondary Domains. In the following example, the client.com is a local Secondary Domain.

Example:

```
<sales@client.com> = Bill@client.com
```

All messages directed to sales@client.com will be directed to Bill@client.com.

Example:

```
<sales@client.com> = Bill
```

All messages directed to sales@client.com will be directed to Bill@mycompany.com (i.e. to the address Bill in the Main Domain).

In most cases you do not have to use Router Alias Records: if you need to provide an alternative name for an account in the main domain, use [Account Aliases](#) instead. If you need to re-route all mail sent to some name in a local domain to some external address, use [Forwarders](#) instead.

Sometimes it is necessary to create an alias for a specific account on a foreign system. For example, all mail sent to some domain should be routed to a specific mail host or to a unified account, but certain accounts in that domain should be routed to accounts on your or other systems.

Example:

```
<sales@client1.com> = sales-client1
```



```
client1.com = new.client1.com
```

These records route all messages directed to the account sales at the domain client1.com to the Account sales-client1 in your Server Main Domain, while messages to all other accounts in the client1.com domain are routed to the new.client1.com system.

The wildcard symbol (*) can be used only in the local part of the full account name (i.e. it can be used before the @ sign).

You can use the wildcard feature to host several domains in one CommuniGate Pro Domain creating a unique "address space" for each domain name.

Example:

```
<*@client5.com> = c15-*
```

```
<*@client7.com> = c17-*
```

Mail to sales@client5.com address will be stored on your server in the c15-sales Main Domain account, messages to info@client5.com address will be stored in the c15-info Main Domain account, while messages to sales@client7.com address will be stored in the c17-sales Main Domain account.

This method can be used when you do not want to create full-scale CommuniGate [Domains](#) for many domains that should host 1-2 accounts.

Special Addresses

If the domain part of an address is NULL, or if the domain name part is empty, and the local part is NULL, the address is marked as "delivered" without any processing. This allows you to use a local name NULL or the domain NULL as a "black hole" address: all messages sent to that address are just discarded. The MAILER-DAEMON address is automatically rerouted to NULL.

Example:

```
bad.company.com = null
```

```
<junk> = null
```

With these records in the Routing Table, the Server will discard all mail sent to the domain bad.company.com, as well as all mail sent to the Main Domain address junk.

If the domain name part of an address is ERROR, or if the domain name part is empty, and the local name part is ERROR, the address is rejected without processing, generating the "Blacklisted Address" error report.

Example:

`bad.company.com = error`

`<junk> = error`

With these records entered, the Server will reject all mail sent to the domain `bad.company.com`, as well as all mail sent to the Main Domain address `junk`.

If the domain name part of an address is `BlackListed`, or if the domain name part is empty, and the local name part is `BlackListed`, the address is rejected without processing, generating the "Blacklisted Address" error report. See the [SMTP module](#) description for the details.

If the domain name part is empty, and the local name part is `spamtrap`, routing stops. Addresses of that type are rejected as the `ERROR` addresses, but the [SMTP module](#) processes them in a special manner. See the [Protection](#) section for the details.

If the domain name part ends with the symbols `.here`, this suffix is removed, and the remaining part of the domain name is used as the name of a local CommuniGate Pro domain. This suffix allows you to avoid routing loops in certain situations.

Example:

`dept1.xyz.com = dept1.xyz.com.here`

`dept2.xyz.com = dept2.xyz.com.here`

`*.xyz.com = *.abc.com`

Mail to all subdomains of the `xyz.com` domain is rerouted to the subdomains of the `abc.com` domain, except for mail to `dept1.xyz.com` and `dept2.xyz.com` subdomains which is routed to the local `dept1.xyz.com` and `dept2.xyz.com` CommuniGate Pro domains.

Routing by IP Addresses

After all Routing Table records are applied, the Router checks if the domain name is actually an IP address. If the IP-address domain name is not enclosed into the square brackets, the Router encloses it: `user@10.34.45.67` is converted into `user@[10.34.45.67]`. This allows you to specify Routing Table records for IP addresses assuming that the address is always enclosed into square brackets.

For IP addresses enclosed in square brackets, the Router checks if the IP address is a dedicated IP address of some secondary domain. If a secondary domain is found, the IP address is substituted with that domain name. If the IP address is the IP address of the server Main Domain, an empty string is placed into the domain name part, and the Router makes the next iteration after parsing the local name part of the address.

Routing via Modules

If no Routing Table record can be applied to an address, and the address is not a special address or an IP address of a local domain, the Router calls each communication module requesting a routing operation.

Each module looks at the address passed and can:

- ignore the address if the module does not know how to handle it;
- modify the address (for example, the LIST module converts addresses `listname-admin@listdomain` into the real address of the mailing list owner);
- reject the address (for example, the Local Delivery module rejects `username@domainname` addresses if the domain name is a name of a local domain, and there is no username account or alias in that domain);
- accept the address.

If a module has modified an address, the Router makes a new iteration, repeating all steps for the new, modified address.

If the Router is called from the [Message Enqueuer](#) component, and a module has accepted an address, the message is enqueued to this module for delivery.

Each module is called twice. First, the Router calls each module asking to process "obvious" addresses. On this call the modules process only the addresses that are definitely directed to that module: the SMTP module processes addresses with the domain part ending with `.smtp`, the UUCP module processes addresses with the domain part ending with `.uucp`, the LIST module processes the addresses of the created mailing lists, etc.

If all modules have ignored an address, the Router calls each module again, asking for a "final" attempt. On that stage, the Local Delivery module processes all addresses directed to local domains, the SMTP module processes all addresses with domain names that have at least one dot, etc.

This two-step method allows several modules to correctly process E-mail addresses without relying to a particular module call order. If each module would process an address in one step, `listname@domainname` addresses (that look like Local account addresses), would be rejected with the Local Delivery module if it is called before the LIST module, `user@uucphost.uucp` addresses would be taken with the SMTP module instead of the UUCP module, etc.

See the module descriptions for details.

Default Records

When the server is first installed, the following records are placed into the Routing Table:

`<root> = postmaster`

This record reroutes all mail to the user "root" to the postmaster account. This is useful on Unix systems, where many logging utilities are preconfigured to mail reports to the user "root".

`localhost =`

On many systems the domain name "localhost" is a synonym for the local IP address of this computer, and some mailer programs use this name as a domain name. This record routes addresses within the "localhost" domain to the main server domain.

`mailhost =`

Some mailer programs use the "mailhost" name as the domain name of the local mail server. This record routes such addresses to the accounts in the main server domain.

`<blacklist-admin*@blacklisted> = postmaster`

This record implements ["white hole" processing](#) for blacklisted hosts.

All these default records can be modified or removed, if needed.

Extending Non-Qualified Domain Names

Users working on sites that have many different mail servers (`server1.myorg.org`, `server2.myorg.org`, `server3.myorg.org`) tend to use addresses with non-qualified domain names (`user@server1`, `user@server2`, `user@server3`). When you have only few servers in your `myorg.org` "upper level" domain, you can "fix" those E-mail addresses by specifying several Router Table records:

`server1 = server1.myorg.org`

`server2 = server2.myorg.org`

`server3 = server3.myorg.org`

If you have many servers in your `myorg.org` "upper level" domain, it becomes impossible to provide Router Table records for all of them. In this case you may want to enable the Add `myorg.org` to Non-Qualified Domain Names option. If this option is enabled, and an E-mail address cannot be routed using CommuniGate Pro Router Table and Modules, and the domain part of the address does not contain a dot symbol, the specified string (`myorg.org`) is added to the address domain name

(separated with the dot symbol). The address `user@someserver` will be converted to the `user@someserver.myorg.org` address and the Router will try to route this new, corrected address.

Note: It is a very bad practice to use non-qualified domain names in E-mail addresses. Enable this option only if you can not enforce a policy that requires your users to specify correct, fully-qualified domain names in E-mail addresses.

Cluster-wide Routing Table

The CommuniGate Pro [Dynamic Cluster](#) maintains the Cluster-Wide Routing Table. When you open the Router WebAdmin page on any Cluster member, you see the link that opens a Cluster-Wide Routing Table page. All modifications made to this Table are automatically propagated to all Cluster Members.

The Cluster-Wide Router Table is processed as an extension of the Server Router Table: the Cluster-Wide Router Table records are checked when no Server Router Table record can be applied.



Automated Mail Processing (Rules)

The CommuniGate Pro server can automatically process messages using several sets of Automated Rules.

The Server-Wide Rules are applied to all messages submitted to the server. These Rules are applied by the [Enqueuer](#) kernel component, before it enqueues a message into the transfer module queue(s).

When a message is directed to an Account on the CommuniGate Pro Server, the [Local Delivery module](#) applies the Rules specified for that Account.

Each Rule has a name, priority, a set of conditions, and a set of "actions". The higher priority Rules are checked first: a Rule with the priority level of 9 is applied before a Rule with the priority level 1.

If a message meets all Rule conditions, the Rule actions are performed, and automated processing either stops, or proceeds checking other, lower-priority Rules.

Specifying Account Rules

The System administrator can specify Server-Wide Rules using the Rules Settings page.

The System administrator can specify Account Rules using a link on the [Account Settings](#) page.

Account users can specify their Account-Level Rules themselves, using the [WebUser Interface](#). The System or Domain administrator can limit the set of Rule actions a user is allowed to specify.

Creating, Renaming and Removing Rules

When the list of Rules appears in a browser window, the Rule names and priorities can be modified:

Priority	Name	EditDelete
		Edit
		Edit
		Edit

After you have modified the Rule names and/or priorities, click the Update button. The list is displayed re-sorted by priority.

Rules with the disabled priority are not applied to the messages, but they are not deleted from the Account Rules set, and they can be reenabled at any moment.

To create a new Rule, enter its name in the field on the top and click the Add Rule button.

To remove a Rule, select the checkbox in the Delete column and click the Update button.

To modify the Rule conditions and actions, click the Edit link.

Rule Conditions

Each Rule can have zero, one, or several conditions. The conditions are checked in the same order they are specified. If a message meets all the Rule conditions, the Rule actions are performed.

The condition operations `is` and `is not` process their parameters as "pictures": the asterisk (*) symbols in parameters are processed as wildcards that match zero or more symbols in the tested string. To check that a string contains the `@thatdomain` substring, the `is *@thatdomain*` operation should be used, and to check that a string does not end with the `somedomain.com` substring, the `is not *somedomain.com` operation should be used.

The condition operations `in` and `not in` process their parameters as sets of one or more "pictures" separated with the comma (,) symbols. The tested string is compared to all picture strings. The `in` condition is met if the tested string matches at least one picture string. The `not in` condition is met if the tested string does not match any picture string in the specified set.

Note: do not use excessive spaces around the comma signs: spaces before the comma sign become trailing spaces of the previous picture, and spaces after the comma sign become leading spaces of the next picture.

The following Rule conditions are implemented:

From [is | is not | in | not in] *string*

This condition checks that the message From address is (or is not) equal to the specified *string*.

Sample:



This condition will be met for all messages coming from any account on any of stalker.com subdomains.

Sender [is | is not | in | not in] *string*

Reply-To [is | is not | in | not in] *string*

To [is | is not | in | not in] *string*

Cc [is | is not | in | not in] *string*

Reply-To [is | is not | in | not in] *string*

The same as above, but the message Sender, Reply-To, To, or Cc address is checked.

If a message has several addresses of the given type, the condition is met if it is true for at least one address. If a message has no addresses of the specified type, the condition is not met.

Any To or Cc [is | is not | in | not in] *string*

The same as above, but all message To AND Cc addresses are checked. If the message has no To/Cc addresses, the condition is not met.

Each To or Cc [is | is not | in | not in] *string*

All message To AND Cc addresses are checked. The condition is met if it is true for each To and Cc address of the message, or if the message has no To/Cc addresses.

Sample:

This condition will be met for messages where all To and CC addresses are addresses in the mycompany.com domain or addresses in the mydept.mycompany.com domain.

Return-Path [is | is not | in | not in] *string*

This condition compares the message "Return-Path" (a.k.a. MAIL FROM) envelope address with the specified string.

'From' Name [is | is not | in | not in] *string*

The same as above, but the instead of the address, the "address comment" (the real name) included in the From address is checked.

Sample:

This condition will be met for messages with the following From: addresses:

From: jsmith@company.com (John J. Smith)

From: "Bill J. Smith" b.smith@othercompany.com

From: Susan J. Smith <susan@thirdcompany.com>

Subject [is | is not | in | not in] *string*

This condition checks if the message subject is (or is not) equal to the specified string.

Sample:

This condition will be met for messages with the following Subject fields:

Subject: we urgently need your assistance

Subject: Urgent!

Message-ID [is | is not | in | not in] *string*

This condition checks if the message ID is (or is not) equal to the specified string.

Sample:

This condition will be met for all messages without the Message-ID flag and for messages that have Message-ID without the @ sign.

Message Size [is | is not | less than | greater than] *number*

This condition checks if the message size is less than (or greater than) the specified number of bytes.

Sample:

This condition will be met for messages larger than 100 kilobytes.

Human Generated

This condition checks if the message is not generated by some automatic message generating software.

It actually checks that the message header does not contain any of the following fields:

Precedence: bulk
Precedence: junk
Precedence: list
X-List*
X-Mirror*
X-Auto*
X-Mailing-List

This condition also checks that the message has a non-empty Return-Path.

Header Field [is | is not | in | not in] *string*

This condition checks if the message RFC822 header contains (or does not contain) the specified header field. The additional fields added using the Add Header operation (see below) are checked, too.

Sample:

Any Recipient [is | is not | in | not in] *string*

This condition compares message "Envelope" addresses and the specified *string*. If this condition

is used in an Account-level Rule, only the addresses routed to that account are checked.

The addresses are processed in the form they had before the Router Table and other routing methods have modified them. If an account has several aliases, this condition allows you to check if a message was sent to a specific account alias.

Messages can be submitted to the server using the ESMTP ORCPT parameter. This parameter specifies how the address was composed on the sending server, before the relaying/forwarding server has converted it to a different address. In this (rare) case, that server can use the ESMTP ORCPT parameter to specify the original address.

Sample:

- a message was composed somewhere and sent to the address user1@domain1.com;
- the domain1.com server received the message and converted that envelope address to user2@domain2.com (mail forwarding);
- the domain1.com server relayed the message to your CommuniGate Pro server domain2.com;
- the domain2.com CommuniGate Pro server received a message;
- the domain2.com CommuniGate Pro server found that the user2 is an alias of the user3 account, and the server routed the message to that user3 account.

If the domain1.com server is an advanced server and informed the domain2.com CommuniGate Pro server that the original address was user1@domain1.com, the string <user1@domain1.com> is used when the Recipient condition is checked.

If the domain1.com server has not informed your server about the original address, the <user2@domain2.com> string is used when the Recipient condition is checked.

The condition is met if it is met for at least one envelope address.

Each Recipient [is | is not | in | not in] *string*

The same as above, but the condition is met only if it is met for all message envelope addresses (if used in an Account-Level Rule - for all message addresses routed to that account).

Time Of Day [is | is not | less than | greater than] *time string*

This condition checks the current time of day in the Server time zone. This condition allows you to compose rules that are applied to messages only at certain times of day.

A time string should be specified as hh:mm or hh:mm:ss, where hh is the hour, mm - minutes, ss - seconds. Time strings can contain the am or pm suffix.

Sample:



Current Date [is | is not | less than | greater than] *date string*

This condition checks the current time and date. This condition allows you to compose rules that are applied to messages only before or after the specified date and time.

A date string should be specified in one of the following formats:

- DD MM YYYY
- DD MM YYYY hh:mm
- DD MM YYYY hh:mm:ss
- DD MM YYYY hh:mm:ss +ZZZZ
- DD MM YYYY hh:mm:ss -ZZZZ

where:

DD is the day of month

MM is month specified as 3-letter English abbreviation:

Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

YYYY is the year

hh is the hour

mm is the minute

ss is the second

+ZZZZ or -ZZZZ is the time zone; if the time zone is not specified, the Server time zone is used.

Sample:

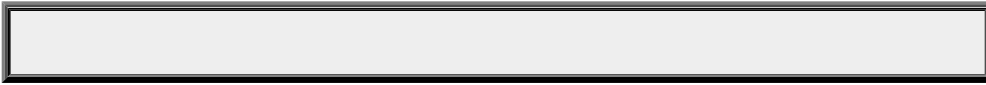


Current Day [is | is not | in | not in] *day string*

This condition checks the current day of week (using the Server local time zone). This condition allows you to compose rules that are applied to messages only on certain days of week.

Days should be specified either as numbers (0 for Sunday, 6 for Saturday), or as RFC822 abbreviations (Sun, Mon, Tue, Wed, Thu, Fri, Sat).

Sample:



The following conditions can be used in Server-Wide Rules only:

Any Route [is | is not | in | not in] *string*

This condition checks that a message "Envelope" address is routed to the specified *string*.

The condition is met if it is met for at least one envelope address.

Note: only the "local part" of the parsed and routed address is checked. If, for example, an envelope address <user@client.com> was processed with the [Router](#) record

`client.com = relay.host.smtp`

then the envelope address (to be sent to a remote host relay.host) will have the local part user, not user@relay.host.

If you plan to use this type of Rule condition, use the Test button on the WebAdmin Interface Router page to see how various addresses are routed.

Each Route [is | is not | in | not in] *string*

The same as above, but the condition is met only if it is met for all message envelope addresses.

String Lists

The CommuniGate Pro Server can store named lists of strings in the Account datasets. Each list can contain zero, one, or several strings. The Rule Condition operations can refer to those lists, if:

- the condition operation is `in` or `not in`
- the operation parameter is specified as a *string*
- the operation parameter starts with the hash (#) sign.

For example, the Condition operation

Sender in #BlockedSenders

checks if the message sender's address is included into the String List called BlockedSenders.

Rule Actions

Each Rule can have zero, one, or several actions. If a message meets all the Rule conditions, the Rule actions are performed.

The following Rule actions are implemented:

Stop Processing

This action should be the last one in a Rule. Execution of this Rule stops and no other (lower-priority) Rules are checked for that message. The message is stored in the INBOX.

Discard

This action should be the last one in a Rule. Execution of this Rule stops and no other (lower-priority) Rules are checked for that message. The message is not stored in the INBOX, but the positive Delivery Notification is sent back to the message sender (if requested).

Sample:

```
IF From is *that_annoying_guy*  
THEN  
Discard
```

Reject [*error message text*]

This action should be the last one in a Rule. Execution of this Rule stops and no other (lower-priority) Rules are checked for that message. The message is rejected, and a negative Delivery Notification is sent back to the message sender.

If the action parameter text is not empty, it is used as the error message text.

You can still store the rejected message using the Store action before the Reject action.

Sample:

```
IF Subject is *UCE*  
THEN  
Reject    please do not send such messages here
```

Mark *operation* [, *operation...*]

This action sets or resets the specified flag(s) for the message. Initially, the set of message flags is empty.

- The Read operation adds the Read (Seen) flag to the message flag set, the Unread

operation removes the Read (Seen) flag.

- The Flagged operation adds the Flagged flag to the message flag set, the Unflagged operation removes this flag.
- The Answered operation adds the Answered flag to the message flag set, the Unanswered operation removes this flag.

When a message is stored in a mailbox as a result of the `Store in` action, as well as when a message is stored in the INBOX after all Rules are applied, the message is stored with the specified flag set.

Sample:

```
IF Sender is *list*
THEN
  Mark Flagged
```

Add Headers *header fields*

This action adds RFC822 header fields to the message. Initially, the set of additional message header field contains the Retrurn-Path field generated using the return-path in the message envelope.

When a message is stored in a mailbox as a result of the `Store in` action, as well as when a message is stored in the INBOX after all Rules are applied, the message is stored with the additional header fields.

Sample:

```
IF Subject is *purchase*order*
THEN
  Add Headers X-Special-Processing: order
```

Note: the following actions are not implicit "Discard" actions, and they do not prevent the original message from being stored in the INBOX. If you want, for example, to redirect a message without keeping a copy in your INBOX, specify the Redirect action followed with the Discard action.

Store in *mailbox name*

The message is copied to the specified mailbox in your account. The mailbox should already exist.

If the mailbox name is specified as `~user_name/mailbox_name`, the message is stored in the mailbox_name mailbox in the user_name account. You should have the Insert access right to that mailbox.

Sample:

```
IF Subject is *Make*$$*
THEN
  Store in ~postmaster/abuse
  Discard
```

Redirect to *addresses*

The message is redirected to one or several specified E-mail addresses. If several addresses are specified, they should be separated with the comma (,) sign.

Forward to *addresses*

The message is forwarded to the specified addresses. The From address is changed to this account address.

Mirror to *addresses*

The message is mirrored (redirected) to the specified addresses. Unlike the Redirect to operation, the Mirror-to operation does not change the message headers, only the Return-Receipt-to: and Errors-to: header fields (if any) are removed, and the X-Mirrored-by header field is added to the "mirrored" messages.

Reply with *message text*

The specified text is used to compose a reply message. The reply is sent to the address specified in the Reply-To address of the original message. If the Reply-To header is absent, the reply is sent to the original message From address.

The header fields *Subject: Re: original message subject* and *In-Reply-To: original message-ID* are added to the reply message.

The specified message text can contain macro symbols that are substituted with actual data when a reply message is composed:

- ^S is substituted with the Subject of the original message;
- ^F is substituted with the From address of the original message;
- ^T is substituted with the Date field of the original message;
- ^I is substituted with the Message-ID field of the original message.

Sample:

Reply with

If the specified text starts with the '+' sign, the lines following this sign are added to the message header. The text should specify the Subject field, since the system will not automatically add the *Subject: Re: original subject* and *In-Reply-To: original message-ID* fields into the reply message.

The specified header portion can contain additional To, Cc, and Bcc fields and the reply message will be sent to those addresses (the Bcc fields will be removed from the message header).

If the specified header does not contain the **From** field, the account address is added as the **From** field. If the **From** field is specified, the account address is added as the **Sender** field.

The **^S** and other macro symbols can be used in the additional header fields, too.

An empty line should separate the message body from the additional header fields:

```
Reply with
```

If the specified text starts with the `[charsetName]` string, the text is converted to the specified charset (all non-ASCII texts are stored in the UTF-8 charset).

If the text does not start with the '+' sign, the header fields

```
MIME-Version: 1.0
```

```
Content-Type: text/plain; charset=charsetName
```

are added to the message headers.

If the text starts with the '+' sign, the '+' sign must be specified after the `[charsetName]` string, and you should specify the MIME-Version and Content-type fields yourself.

Reply to All with *message text*

The same as above, but the reply is sent to all addresses listed in the original message To: and Cc: fields.

React with *message text*

The specified message text should contain a header, an empty line, and the message body. The header should contain any number of To, Cc, and Bcc fields, the Subject field, as well as any number of additional fields. The composed message is sent to the specified addresses. The system uses the account address to compose the From field for these reaction messages.

If the specified header already contains the From field, the account address is added as the Sender field.

The specified message header and the message body can contain macro symbols listed above.

Sample:

```
React with
```

The message text can start with the `[charsetName]` string (see above), in this case you need to specify the MIME-Version and the Content-Type header fields:

Sample:

```
React with
```

Execute *command line*

The specified command is executed in a separate OS process (task).

The message text (the header and the body) is sent to the task as that task *standard input (stdin)*.

Note: the task must read the entire *stdin* data stream, otherwise the Execute command fails.

A command text can be prefixed with the `[FILE]` tag:

```
[ FILE ] myprogram parm1
```

When this prefix is used, the task standard input will be empty (closed), and the string

```
-f Queue/fileid.msg
```

(the `-f` flag and the Message file name, relative to the *base directory*) will be appended to the end of the command text:

```
-f Queue/12002345.msg
```

Note: usually access to the *base directory* is not granted to regular users, so the `[FILE]` prefix can be used in the Server-Wide Rules only.

A command text can be prefixed with the `[RETPATH]` tag:

```
[ RETPATH ] myprogram parm1
```

When this prefix is specified, the string `"-p"` followed by the message return-path address is added to the end of the command text:

```
-p address@domain.com
```

A command text can be prefixed with the `[RCPT]` tag:

[RCPT] myprogram parml

When this prefix is specified the string "-r" followed by the list of message recipient addresses is added to the end of the command text:

```
-r address1@domain1.com address2@domain2.com
```

Note: the [RCPT] prefix can be used in the Server-Wide Rules only.

A command text can be prefixed with the [STDERR] tag (see below).

A command text can have several prefix strings, and they can be specified in any order. If several of [FILE], [RETPATH], and [RCPT] prefix strings are specified, the -f flag and its parameter are added first, followed with the -p flag and its parameter, followed with the -r flag and its parameters.

When the task completes, the task exit code is checked. If the code is zero, the Rule action is considered as executed successfully, and the next Rule action is executed.

If the task exit code is non-zero, the message is rejected with the error code "automated processing failed", and the data from the task standard error channel is recorded in the Log along with the task exit code.

If the [STDERR] prefix was specified on the command line, the data written to the standard error channel (if any) is used to compose the error report text.

The data from the task standard output, if any, should not exceed 4Kbytes in size. It is recorded in the Log and discarded.

The CommuniGate Pro Server monitors the task during its execution, and it interrupts the task if it does not complete within 2 minutes.

When a task is to be executed as a part of Account-level Rule processing, the OS User Name is composed using the Account [OS User Name](#) setting, and the task is executed in that OS User Environment.

When the CommuniGate Pro runs under control of a Unix system, the task is assigned the specified Unix User ID, group ID, and the set of groups; the task current directory is set to the Unix User home directory.

The Execute action cannot be used in Account-level Rules if the CommuniGate Pro Server runs under MS Windows, AS/400, or BeOS operating systems.

When a task is to be executed as a part of a Server-Wide Rule, it is launched in the CommuniGate

Pro Server own environment (with the base directory being the current directory).

Sample:

```
Execute
```

`FingerNotify [address]`

The Server connects to the computer at the specified network address, port 79 (the *finger* port), and sends the `nm_notifyuser` string to that computer. If the address is not specified, and the action is executed as a part of an Account-Level Rule, the network address of the last user Login is used.

This action should be used with the [NotifyMail®](#) utility installed on client computers.

Sample:

```
FingerNotify
```

Users are allowed to specify this action only if they are allowed to specify `execute`-type actions.

You can configure the Notifier settings using the *Obscure* page in the Settings WebAdmin realm.

`ExternalFilter`

This action tells the Server to pass the message to the [External Content Filtering](#) program. This action can be specified only in the Server-Wide Rules.

`Write To Log string`

A Major-Level (Level 2) record with the message ID and the specified string is placed into the [System Log](#).

Only the Server Administrator is allowed to specify this action.

Auto-Reply Message

Each Account has one built-in rule to generate Auto-Reply messages. When enabled, the Rule checks

that the message is not an auto-generated one, and that the message author (the 'From' address) has not be placed into the `RepliedAddresses` string list. It then composes and sends an auto-reply message and adds the message author address into the `RepliedAddresses` string list, so the auto-reply message will be sent to each message author only once.

This Rule conditions are:

Human Generated

From not in `#RepliedAddresses`

The Rule actions are:

Reply with *Reply Text*

Remember 'From' in `RepliedAddresses`

Only the text of the Reply message can be modified:

Auto-Reply

Auto-Reply

If this option is not selected the Auto-Reply Message Rule is disabled. If this option is selected, the Auto-Reply Message Rule is enabled with a low priority (the rule priority is set to 2).

Even if the Administrator has not allowed the user to specify Automated Rules, the Auto-Reply Message can be enabled by the user herself, and the user can always modify the Auto-Reply Message text.

The Rules page of the WebUser Interface contains the "Clear 'Replied Addresses' List" button. When the user clicks this button, the `RepliedAddresses` string list is removed from the Account dataset.

Redirect All Simplified Rule

Each Account can have a simplified rule to redirect all incoming mail to a different address or addresses.

This Rule condition is either empty (the Rule action is applied to all messages) or, optionally, human generated, the Rule actions are `Redirect To` or `Mirror To`, and, optionally, `Discard`.

Only the list of redirection addresses can be modified:

Redirect All Mail to:		
Keep a Copy	Do not Redirect Automatic Messages	Preserve To/Cc fields

`Redirect All Mail to`

If this option is not selected the `Redirect All` Rule is disabled. If this option is selected, the `Redirect All` Rule is enabled with the lowest priority (the rule priority is set to 1).

`Keep a Copy`

If this option is not selected, the action `Discard` is added to the Rule and all redirected messages are NOT stored in the account INBOX.

`Do not Redirect Automatic Messages`

If this option is selected, the condition `Human Generated` is added to the Rule and messages from non-human sources (mailing list messages, error messages, redirected and mirrored messages) are not processed with this Rule.

`Preserve To/Cc fields`

If this option is selected, the `Mirror To` action is used for this Rule. If this option is not selected, the `Redirect To` action is used.

The account user can set this Rule only if the Account is granted a right to specify the *redirecting* Rule actions. Otherwise only the Administrator can set this Rule for the user account.

Logging Rules Activity

The [Enqueuer](#) component records Server-Wide Rules activity in the Log. Set the Enqueuer Log Level to `Low-Level` or `All Info` to see the Rules checked and the actions executed.

The [Local Delivery](#) module records Domain-Wide and Account-Level Rules activity in the Log. Set the Local Delivery module Log Level to `Low-Level` or `All Info` to see the Rules checked and the actions executed.

Using External Content/Virus Filtering

You may want to use an external program (Helper) to filter all messages the Server processes. This is useful for employing third-part anti-virus and content filtering products.

CommuniGate Pro starts the external Content Filtering program alongside the Server and sends it the names of the message files it should scan. This design allows the Server to avoid the overhead associated with program launching and initializing.

To use an external Content Filtering program, open the General page in the Settings section of the WebAdmin Interface and click the Helpers link. The Helpers page is displayed:

Content Filtering	
Log:	Program Path:

Log

Use this setting to specify what kind of information the Content Filtering Interface should put into the Server Log. The Content Filtering Interface records in the System Log are marked with the `EXTFILTER` tag.

Program Path

Use this setting to specify the full name of the external Content Filtering program.

The Content Filtering option is enabled only when the checkbox in the panel header is selected. If it is not selected, the Content Filtering operation is a dummy operation.

When the Content Filtering program is successfully specified, you can scan messages with that program using the `ExternalFilter` action in the Server-Wide Rules. If you want to scan all messages, create a Rule without any condition (so the Rule will apply to all messages the Server processes), and specify one action - `ExternalFilter`.

Content Filtering API

Any third-party program can be used for Content Filtering if it implements the following interface:

- The program should read requests from its *standard input*. Each request contains exactly

one text line and has the following format:

seqNum FILE *fileName*

where *seqName* is a positive number - request identifier, and *fileName* is the name of the file the program should scan.

- For each request, the program should produce exactly one text response line, and the response line should not be larger than 4096 bytes.
- If a message should not be rejected, the response line should have the following format:

seqNum OK

where *seqName* is the request identifier.

- If a message should be rejected the response line should have the following format:

seqNum ERROR *report*

where *seqName* is the request identifier, and the *report* is a text string explaining why the message is rejected. This (optionally multi-line) text should be placed into the response line in the CommuniGate Pro [String format](#)

- If a message should be postponed (because of the license limitations, for example), the response line should have the following format:

seqNum REJECTED *report*

where *seqName* is the request identifier, and the *report* is a text string explaining why the message should be postponed.

- The program SHOULD be ready to process several requests simultaneously (using several threads). Since the Content-Filtering program is used with the Server-Wide Rules (processed with the [Enqueuer](#) Server component), the program should be ready to handle N concurrent requests, when N is the number of Enqueuer "processors" (threads).
- The program MAY be implemented as a single-threaded one, so it reads the next request only after the previous request has been processed. But this design can result in severe performance degradation of the entire Server: when a single-threaded Content Filtering program is scanning a large message, other messages are not being enqueued.

The program is started with the CommuniGatePro *base directory* as its current directory.

If the external program crashes, CommuniGate Pro suspends the Enqueuer processes until the external program is restarted.

Cluster-Wide Rules

The [Dynamic Cluster](#) Administrators can see an additional link on the Rules page in the Settings section of the WebAdmin Interface. This link allows them to open the list of Cluster-wide Rules.

When you modify the Cluster-wide Rules set on any Cluster Member, the set is automatically updated on all Cluster members.

The effective set of "server-wide" rules for each Cluster member is a union of the Server-Wide Rules explicitly set on that Cluster member and the Cluster-wide Rules.

Rules from both sets are applied together, in the order specified with the Rule priority attribute. For example, messages can be processed with a high-priority Cluster-wide Rule, then with a medium-priority Server-wide Rule, then with a low-priority Cluster-wide Rule.



SMTP Module

The CommuniGate Pro SMTP module implements E-mail message transfer using the [SMTP and ESMTP Internet protocols](#) via TCP/IP networks.

The Simple Mail Transfer Protocol allows computers to transfer messages using network connections. A computer that has a message to send connects to the recipient's computer and establishes a network link. Then it sends one or several messages and closes the network connection.

Mailer applications (such as Eudora®, Netscape® mailer, and many others) use the SMTP protocol to submit messages to the mail servers, and mail servers then forward the submitted messages to the recipients. All mailers have a setting called SMTP Host Address that specifies the network address of the mail server computer. Mailer applications open connections to that address when they have a message to submit.

The CommuniGate Pro SMTP module supports special "secure ports" and the STARTTLS SMTP extension, and it can receive and send mail via secure (encrypted) connections.

The CommuniGate Pro SMTP module supports the AUTH extension and allows remote users to authenticate themselves before submitting messages.

Simple Mail Transfer Protocol (SMTP) and DNS

The mail servers use the global Domain Name System to find the network address of the recipient computer or the recipient mail server. Each domain (part of the E-mail address after the @ symbol) should have a special so-called MX-record in the Domain Name System. That record specifies the name of the computer that actually receives mail for that domain. For example, MX records can specify that mail for the domain `company.com` should be sent to the computer `mail.company.com`, and mail to the domain `enduser.com` should be sent to the computer `provider.com`.

There can be several MX-records for one domain (with different priority values). If one (high-priority or primary) computer cannot receive mail, mail is sent to lower-priority computers (called Back-up Mail Servers). Back-up mailer servers then try to deliver the message to the primary server.

When the name of the recipient computer is retrieved from the DNS, the sending mail server consults the DNS again. Now it uses the DNS to convert the receiving mail server name into its network address. The so-called DNS A-records contain the pairs that link a computer name to its global Internet network (IP) address.

When the network address of the recipient mail server is received from the DNS, the sending mail server opens an SMTP connection to that server and transfers the message(s). When all messages to that domain are transferred, the connection is closed.

When a message contains several addresses within the same domain, the SMTP module can transfer only one copy of the message to the mail server serving that domain, and that server delivers messages to all recipients in that domain. But if there are too many addresses, the SMTP module can break them in several portions and send several copies, each containing only a portion of the address set.

If there are several messages to one domain, the SMTP module can open several connections to the mail server serving that domain and send those messages simultaneously.

If you want to receive messages from the Internet with your own mail server, you should register your domain name, and ask your provider to register that name with the Domain Name System. The DNS records should point to the computer running your mail server.

Configuring the SMTP module

To configure the SMTP module, use any Web browser to connect to the CommuniGate Pro Server, and open the SMTP page in the Settings realm. To configure the SMTP module, you should have the Can Modify Settings access right.

Log:

Use the Log setting to specify what kind of information the SMTP module should put in the Server Log. Usually you should use the Major (message transfer reports) or Problems (message transfer and non-fatal errors) levels. But when you experience problems with the SMTP module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly.

The SMTP module records in the System Log are marked with the SMTP tag for incoming connections, with the SMTPO tag for outgoing connections, and with SMTPW tag for connections used to wake up the back-up server.

Sending Messages via the Internet

If you want to send messages over the Internet, your server should have a TCP/IP link to the Internet. When a message should be transferred to some remote host, the SMTP module connects to that host via the TCP/IP network, and it transfers the message using the SMTP protocol .

Sending		Channels:
Send Directly to Recipients		
Forward to		
Send AUTH:	Password:	
Retry Every:	Keep Trying for:	
then Every:	After:	
Send Warnings	After:	
Channels/Host:	Add Channels after:	

Channels

This setting limits the number of outgoing SMTP connections the module is allowed to open simultaneously.

Retry Every

Use this setting to tell the SMTP module when it should retry to send a message if a connection fails. If you use a

forwarding mail server, this option specifies when the module should retry to connect to that server if the previous connection failed for any reason. If you use the Directly to Recipients method, and a connection to some remote host (domain) fails, all messages directed to that domain will be suspended for the specified time. Usually SMTP systems suspend messages for 30 minutes.

Keep Trying

Use this setting to limit the number of attempts to deliver a message. If a message cannot be delivered within the specified period of time, the message is rejected and an error report saying that the host is unavailable is sent back to the message sender.

then Retry ... After

Use these settings to tell the SMTP module when and how it should change the retry interval. Usually, you would tell the SMTP module to increase the retry interval to 1-2 hours after a message has spent more than 2 hours in the queue.

Send Warnings After

Use these settings to tell the SMTP module when warning messages should be sent back to the message senders, notifying them about delivery delays.

When sending messages over the Internet, the SMTP module can forward them to some other mail server, or it can deliver messages directly to the recipients, using the DNS MX-records to find the recipient hosts on the Internet.

Sending via a Forwarding Mail Server

Forward to

When this option is selected, the SMTP module connects to the specified *forwarding mail server* (also called a *smart host*) and sends all queued messages to that server. Since the forwarding mail server is usually "close" to your server, messages leave your system quickly. But this method can cause additional delays in message delivery, since messages are queued on the forwarding server and those queues can be processed slowly. This method is recommended when your server is connected to the Internet using a slow link, or when you use a dial-up link and you want messages to leave your server as soon as possible to keep the connection time short.

Select the Forward to option and specify the name or the IP address of the forwarding server. The SMTP module will forward all outgoing messages to that mail server for delivery.

Note: the name of the forwarding mail server should be the name of the real computer (as specified in an A-type DNS record), not a mail domain (MX-type) name. While your provider domain name can be `provider.com`, the name of the provider mail server can be something like `mail.provider.com`. Consult with your provider to get the exact name of the forwarding mail server you can use.

Note: you can specify the IP address of the forwarding server instead of its name. You can also specify several IP addresses, separated with the comma (,) symbol. If SMTP connection to the first specified address cannot be established, the SMTP module will try the next specified address.

Note: when configuring a [Cluster backend](#) server, you can specify the asterisk (*) sign in the Forward To setting. In this case, all cluster frontends (specified using the Cluster Settings page) will be used as forwarding mail servers.

Note: when a recipient domain name is specified as an IP address (as in `user@[12.34.56.78]`), the SMTP module delivers messages directly to the host with the IP address 12.34.56.78, even if the Forward to option is selected. You may use this feature for message exchange between several mail servers on a LAN that does not have its own Domain Name Server.

Send AUTH

The forwarding mail server should be configured to enable relaying from your server to any other server on the Internet. Some forwarding mail servers may require your server to use the AUTH command with a valid name and password parameters before transferring messages that need to be relayed. In this case you should enter the AUTH login name and password in the Send AUTH fields.

This type of configuration is used when your server has a "dynamic IP address", and receives mail from the same forwarding server using the [ATRN method](#). Usually the username and password used for mail forwarding are the same as the username and password used for ATRN receiving.

Sending Messages Directly to Recipients

Directly to Recipients

When this option is selected, the SMTP module uses the DNS (Domain Name System) to convert message recipient addresses into the names and addresses of the receiving hosts. A receiving host can be the recipient host itself, or a relay host. The information about the proper relay host is stored in so-called MX records on Domain Name Servers. For each destination host several records can exist, each record having a priority value. If the SMTP module fails to connect to the relay host with the highest priority, other MX records are used and other relay hosts are tried. If no relay host is available, the message remains in the SMTP queue, and more attempts to deliver it (and all other messages to the same host) are made later.

This method allows the system to deliver a message either directly to the recipient computer or to a relay host that is "very close" to the recipient computer. Recipients can read your messages almost immediately, and your messaging system does not rely on any "forwarding mail server" performance.

Multi-channel Delivery

When the Server queue contains several messages to be directed to the same domain, the SMTP module opens a connection to that domain mail server and sends messages one by one. If the established connection is slow and there is a large message in the Queue, other messages would wait too long before being delivered. You may want to allow the SMTP module to open additional connections to the same mail server and send other messages in parallel.

Channels/Host

Use this setting to limit the number of TCP connection the SMTP module is allowed to establish with one domain mail server.

Add Channel after

Use this setting to specify the "reasonable wait" time period.

Additional connections to a domain mail server are opened if:

- the Server Queue for that domain contains more messages than the number of already opened channels;
- neither channel succeeds to complete a message transfer within the specified period of time or the number of messages in the domain queue is more than 50.

Sending via Dial-up Links

The SMTP module sending activity can be limited using the [TCP Activity Schedule](#). Outgoing messages wait in the SMTP queue till the TCP Activity Schedule allows the Server to initiate outgoing network connections.

When outgoing activity is allowed, the SMTP module tries to send all submitted messages accumulated in its queue.

Secure (encrypted) Message Relaying

You can configure your CommuniGate Pro Server SMTP module to use secure (encrypted) connections when sending messages to certain remote sites. This feature is especially useful if your company has several offices and E-mail traffic between the offices is sent via the public Internet.

You should simply list the domain names that should receive mail from your server via secure connections:

Send Encrypted (SSL/TLS)

To Domains:
(high security)

wherever possible (low security)

The specified names can contain a wildcard - the asterisk (*) symbol.

When the CommuniGate Pro SMTP module connects to a relay of one of the listed domains, it checks if that relay supports the STARTTLS protocol extension command. Then the SMTP module uses this command to initiate a secure connection with that relay.

The CommuniGate Pro SMTP module checks the validity of the remote relay Certificate. The Certificate *subject* must contain the *cn* (*Common Name*) field that matches either the domain name of the remote site, or the name of this relay. This can often cause a problem, since the domain `company.dom` may have the MX record `relay1.company.dom`, but the computer with the `relay1.company.dom` address has the "main" DNS name `smtp.company.dom` and its Certificate is issued to that name (its Certificate *subject* contains `smtp.company.dom` in the *cn* field).

To solve this problem, you should explicitly route all traffic to the `company.dom` domain via the `smtp.company.dom` relay, using the following [Router](#) record:

```
company.dom = company.dom@smtp.company.dom.smtp
```

See the [Routing](#) section for more details about SMTP routing.

Note: this feature ensures that messages between your server and a remote relay are transferred securely. To provide complete end-to-end security, you should verify that:

- users submit messages to servers either using a private network, or using TLS/SSL connections over the public Internet (secure SMTP or secure WebMail);
- all mail servers and relays exchange messages either using a private network, or using TLS/SSL connections over the public Internet (secure SMTP);
- users read messages either using a private network, or using TLS/SSL connections over the public Internet (secure POP, IMAP, WebMail).

If the domain is listed in the Send Secure To Domains list, and the receiving server does not support the STARTTLS command, or the remote server certificate Subject does not match the domain or domain relay name, all messages to that domain are **rejected**, ensuring that no message is sent via a potentially insecure link.

If your server sends all outgoing mail via a forwarding server, you can enter the asterisk(*) symbol into the [Send Encrypted](#) field to encrypt all communications with the forwarding server.

The CommuniGate Pro SMTP module does not check the *Subject* of the forwarding server certificate.

wherever possible

Select this option if you want the SMTP module to try to use SSL/TLS connections with all remote SMTP servers that support this feature. If the remote domain is **not** listed in the Send Secure To Domains list, but the remote server supports the STARTTLS command, the SMTP module tries to establish a secure (SSL/TLS) connection with that server. It does not check the remote server certificate Subject in this case. If the STARTTLS command or secure connection negotiations fail, the server defaults back to plain-text communication and sends message via an unencrypted channel.

If an outgoing connection is made to the port 465 (see [Sending to Non-Standard Ports](#) section), then the SMTP module initiates the secure (SSL/TLS) protocol immediately after establishing a TCP/IP connection.

Receiving Messages

The SMTP protocol is used to receive messages from the Internet and from the client mailer applications. If you want to receive messages from the Internet, you need a TCP/IP link to the Internet, and your server domain name and the IP address should be included into the DNS records.

Receiving		<u>listener</u>	Channels:
Message Limits:	Size:	Recipients:	
Advertise:	AUTH capability to:	8BITMIME to:	
Verify HELO and Return Paths for:		Force AUTH for:	
Send Wakeups	Every:	to:	
Use ATRN	login name:	password:	

Channels Limit

When you specify a non-zero value for this setting, the SMTP module creates a so-called "listener". The module starts to accept all SMTP connections that other mail servers establish in order to send mail to your Server. This setting is used to limit the number of simultaneous connections the SMTP module can accept. If there are too many incoming connections open, the module will reject new connections, and sending mail systems will retry later.

listener

This link allows you to tune the SMTP [Listener](#). You can specify which TCP ports to use for SMTP incoming connections (by default, the port 25 is used), which local IP addresses to use for incoming connections (all available addresses are used by default), and which remote addresses should be granted access to your CommuniGate Pro SMTP server (by default, all addresses can connect to the SMTP port).

Note: to allow Microsoft® Outlook Express 4.x users to submit messages using secure connections, you should configure the SMTP listener to accept connections on the TCP port 465, and enable the SSL/TLS option for that port.

Note: Netscape® Messenger and modern versions of Microsoft Outlook and Outlook Express products do not need any special port for secure communications, since these products use the STARTTLS command to initiate secure communications after establishing a regular, clear text SMTP connection to the standard port number 25.

Message Size Limit

This settings tells the module to reject all incoming messages that are larger than the specified limit.

Message Recipients Limit

This settings limits the number of message recipients the module can accept. Specifying a lower value makes your server less attractive for spammers.

Verify HELO and Return-Path

This option specifies if the parameters of HELO/EHLO commands and the Return-Path (MAIL FROM) addresses should be verified. You can set the module to verify these addresses only in the messages sent from hosts not included into the Client Hosts list. If Return-Paths are not verified, the domain names specified in the HELO/EHLO commands are not verified either. This is useful if:

- many of your clients use mailers that send bogus names in the HELO/EHLO commands;
- your Internet connection is a dial-up one, and you do not want any outgoing (DNS) traffic to be generated when receiving mail from your own client computers (Client Hosts).

See the [Protection](#) section for more details.

Advertise AUTH

If a server reports (in its initial EHLO prompt) that it supports SMTP Authentication, some mailer clients (including Netscape® Messenger 4.x) force users to authenticate themselves before sending messages. If you select the Non-Clients value, and a connection is accepted from an address included into the Client IP Addresses list, the SMTP module will not report that it supports the AUTH command. If the Nobody value is selected, the SMTP module never reports that it supports SMTP AUTH. This option does not disable the SMTP Authentication feature itself.

Advertise NTLM AUTH

The SMTP module supports the non-standard NTLM AUTH method used in Microsoft products (Outlook, Outlook Express, etc.) where it is called "Secure Password Authentication (SPA)". Unfortunately, when some of those products see that the NTLM method is supported, they first try to authenticate using the Windows workstation name and "domain", fail, and only then try to use the user-supplied authentication parameters. Since this can create an enormous number of support problems, the NTLM AUTH is disabled by default. This setting is used to specify when the NTLM AUTH should be advertised to the clients. Remember that the NTLM AUTH method can be advertised only when the SMTP AUTH is advertised (see [above](#)).

Advertise 8BITMIME

If your Server does not report the 8BITMIME capability, some mailers and servers will MIME-encode all non-ASCII messages that they send to your Server. This server-side encoding can cause troubles for many old mail clients. To avoid these troubles, your Server should report the 8BITMIME capability.

Note: The CommuniGate Pro SMTP module never converts non-ASCII messages into the MIME form itself, and (according to RFC1652) it should not advertise the 8BITMIME capability. But the modern Internet is completely 8-bit transparent and clean, so it is safe to enable the Advertise 8BITMIME option, preventing other servers from doing unneeded 8bit-to-MIME message conversion.

Force AUTH

This option can be used to force all "local" users to use the SMTP AUTH feature. If the message Return-Path is an address of one of the CommuniGate Pro Accounts, the message will be rejected if the client mailer has not sent the SMTP AUTH command first. The option value specifies for which sending mailer addresses this feature should be used.

Note: this option checks for the "fixed" Client IP Addresses only - it does not pay attention to the "temp-client" addresses added with the [Process as a Client Address](#) feature.

Note: use the Force AUTH option carefully. Some users may use different mail relays to submit their messages with their CommuniGate Pro Account names as the message Return-Paths. If this option is enabled and those messages are directed to your Server, they will be rejected, because mail relay servers are not able to authenticate the senders on your server.

Note: most mailers will send the AUTH command only when the server advertises its SMTP AUTH capability. Make sure that your server does advertise it (see [above](#));

Waking up the Backup Server

If your Server has a dial-up link, its domain name should have at least one additional DNS MX record, specifying a "back-up" mail server (usually, your ISP mail server). When your Server is off-line, all messages directed to your domain(s) are sent to that back-up mail server.

The back-up mail server tries to deliver collected messages to your server. Usually, the retry period is 30 minutes, so your system should stay on-line for at least that period of time in order to receive messages from the back-up server.

To avoid this delay, the SMTP module can be configured to send the Remote Queue Starting ("ETRN") command to the back-up server. When the back-up server receives that command, it immediately starts to send the collected messages to your

Server.

Send Wakeups

Use these settings to specify the address of the Back-up Server, and to specify how often the Remote Queue Starting command should be sent.

Note: the name of the back-up server should be the name of the real computer (as specified in an A-type DNS record), not a mail domain name. While your provider domain name can be `provider.com`, the name of the provider mail server can be something like `mail.provider.com`. Consult with your provider to get the exact name of your back-up server, or just examine the DNS MX records for your domain: your back-up server is specified with the MX record that has the priority next to your own Server MX Record priority.

The SMTP module wake-up activity is limited with the [TCP Activity Schedule](#).

On-demand Mail Relaying (ATRN)

The ETRN command can be used to release your domain queue on a remote backup server only if your server has a static IP address.

If your server has a dynamic IP address, the ETRN method does not work, since the backup server does not know the IP address your server is using, and the backup server is not able to open a connection to your server.

If your server uses a dynamic IP address, it should use the On-demand Mail Relaying method to retrieve mail from the backup server.

When On-demand Mail Relaying method is used, your server connects to the backup server, authenticates itself, and then it issues the ATRN command. Then the servers exchange their roles and the backup server starts to send your server your domain mail via the same channel. This eliminates a need for the backup server to open a connection to your server.

Since the backup server does not open a connection itself, it has to verify that the server that sends the ATRN command and wants to retrieve your domain mail is really your server. Your server should provide some name and password that should be accepted by the remote server and that should allow your server to issue the ATRN command.

Consult the remote server administrator to learn the name and the password your server should send before sending the ATRN command.

Use ATRN

select this option to use the ATRN (On-demand Mail Relaying) method instead of the ETRN method.

login name and password

this pair of strings is sent to the remote server using the AUTH command. The access rights granted to this login name on the remote server should allow your server to use the ATRN command.

The CommuniGate Pro SMTP module uses the AUTH CRAM-MD5 authentication method to send passwords in an encrypted form. If the remote server does not support the CRAM-MD5 method, the clear-text AUTH LOGIN method is used.

If your backup server does not support On-demand Mail Relaying, you should use the Unified Domain-Wide Account method implemented with the [RPOP](#) module.

The RFC2645 suggests to use the special TCP port number 366 to provide the ATRN services. If your backup mail server provides the ATRN services on that port (or on any port other than the standard SMTP port 25), you should specify the port name in the Send Wakeups To setting field. Use the colon symbol to separate the server name and the port number:
`mail.provider.dom:366`

You can use secure communications with the backup server if you include the backup server name into the [Send Encrypted](#) list.

When the backup server name is specified, the SMTP Settings page displays the Wake Up Now button. Click that button to initiate a wakeup session immediately.

Serving Dial-up Client Hosts

The CommuniGate Pro Server can be used as a back-up mail server for dial-up systems. Dial-up systems receiving mail via SMTP expect their back-up servers to receive and keep all their messages when these systems are off-line. When a dial-up system connects to the Internet again, it connects to its back-up mail server and either issues the special Remote Queue Starting command (ETRN, RFC1985), or sends a dummy E-mail message to a special address on the back-up server.

Remote Queue Starting (ETRN)

When your server receives the ETRN command, it tries to send out all messages collected for the host specified as the ETRN command parameter. This method allows a dial-up system to get its messages immediately, instead of waiting for your server to make the next attempt to deliver the collected messages.

The SMTP module supports the ETRN command, so CommuniGate Pro can be used as a back-up mail server. No special setting is required, since this feature is always enabled.

The SMTP module uses the [Router](#) to process the ETRN parameter (domain name). It adds the wakeup fictitious user name to that domain to get a regular E-mail address `wakeup@etrn-parameter` and runs it through the Router. If the address is routed to an SMTP host, the SMTP module releases (wakes up) that host queue.

If you have routed the domain `client.com` to `mail.client.com` in your Router Table, all mail to the `client.com` domain will be kept in the `mail.client.com` queue. Since the ETRN command parameter is processed with the Router, too, the `ETRN client.com` command will correctly release the `mail.client.com` queue.

In a [Dynamic Cluster](#) environment, the ETRN command received by any cluster member releases domain queues on all cluster members.

On-Demand Mail Relaying (ATRN/TURN)

When the CommuniGatePro SMTP module receives the ATRN command, it checks that the connected party has authenticated itself. Then the module releases the specified domain queue and sends all its messages directly via this (already established) connection. No special settings is required to enable the ATRN feature of the CommuniGate Pro SMTP module. There are some notes about the ATRN implementation:

- Only one ATRN command parameter is allowed.
- The name of the domain queue to be released should match the name of the authenticated user. If you want to allow a dial-up client host to release the `domain.dom` queue, you should create the `domain.dom` Account in the CommuniGate Pro Main Domain, and the client host should authenticate itself using the `domain.dom` as the login name and the `domain.dom` Account password as the password.
- If the ATRN command does not have a parameter, the name of the authenticated user is used as the name of the queue to be released.
- The domain name used in the ATRN command must be included into the `Hold Mail for Domains` list.

The ATRN command parameter (if any) is processed in the same way the [ETRN](#) command parameter is processed.

The RFC2645 suggests to use the special TCP port number 366 to provide the ATRN services. CommuniGate Pro SMTP module provides the ATRN services on all ports its [Listener](#) is using. To comply with the RFC2645 standard, you may want to add the port 366 to the SMTP Listener settings.

For compatibility with legacy Microsoft Exchange servers, the TURN SMTP command is supported, too. It is processed in the same way as the ATRN command without a parameter, and it requires authentication, too. The name of the queue to release is the same as the name of the authenticated user.

Waking up via E-mail

The SMTP module supports an alternative wakeup method: a dial-up system can send any message to *domain name-wakeup@serverdomain* to release the domain name message queue. The servername should be the main domain name of the CommuniGate Pro Server.

In a [Dynamic Cluster](#) environment, the Wakeup E-mail received by any cluster member releases domain queues on all cluster members.

Holding Mail in Queue

You can ask the SMTP module to hold mail for certain hosts in its queue, and not to try to deliver that mail until the receiving server issues the ETRN command or sends a wake-up E-mail. This can be useful if the receiving server is on a symmetric dial-on-demand line and its provider brings the link up automatically when there is any traffic for that receiving server.

Message Relaying

The situation when the SMTP module receives a message from a remote system and then sends that message to some other host is called *relaying*.

To avoid Server abuse, some relay restrictions can be specified.

Relaying		
Relay to non-Clients:	If Received from:	
Relay to Clients:	to:	addresses
Relay to Hosts We Backup:		
Accept Wakeups from:		
Hold Mail for Domains:		

- Relay to non-Clients
- Relay to Clients
 - See the [Protection](#) chapter for the information about these settings.
- Relay to Hosts We Backup

This option allows the SMTP module to relay messages to any domain, if the MX records for that domain includes this CommuniGate Pro Server (its main domain) as a back-up mail relay.

Accept Wakeups

This option tells the SMTP module to accept ETRN commands and wake-up E-mail messages either from anybody, or only from the hosts included into the Client IP Addresses list. Since the ATRN command requires authentication, it is always accepted from any address.

Hold Mail for Domains

When the SMTP module builds a queue for one of the domains (hosts) listed in this field, it immediately places that queue "on hold", waiting for the ETRN command or any other external action that releases that queue. This method should be used for the sites that receive mail via your server, and that want to receive it only when they issue the ETRN command.

Relaying via Dedicated IP Addresses

You may want to send messages for some of your CommuniGate Pro Domains via Local IP Addresses assigned to those Domains. See the [Domain Settings](#) section for more details.

If a message is to be delivered to the *hostName* host via a particular *12.34.56.78* Local IP Address, the message is not placed into the *hostName* SMTP queue. Instead, the Server places it into the *@12.34.56.78:hostName* SMTP queue.

This technique allows the Server to process messages from different Domains independently. If the IP Address of one of your Domains is blacklisted by remote hosts (because that Domain users have abused the mail system), messages to the same remote hosts from other Domains will not be delayed or rejected.

Processing Mail from BlackListed Addresses

When a [BlackListed](#) host connects to the SMTP module, the module does not reject a connection. Instead, it receives the `MAIL FROM SMTP` command, and starts to process the recipient (`RCPT TO`) addresses sent from the blacklisted host. The module adds the domain blacklisted to each recipient address received from a blacklisted host, i.e. the received address `user@domain` is converted into `user%domain@blacklisted`. Then the address is processed with the Router as usual. If the Router Table does not contain special rules for the blacklisted domain, the address is rejected with a special error code.

The default Router Table contains the following line:

```
<blacklist-admin*@blacklisted> = postmaster
```

All messages from blacklisted hosts sent to the `blacklist-admin` address in any domain, are routed to the postmaster, so these messages are accepted. This "white hole" feature allows the blacklisted host users to contact the postmaster on your server if they want to discuss the blacklisting issue. If you remove this line from the Router Table, no address will be accepted from blacklisted hosts.

When rejecting addresses sent from blacklisted hosts, the SMTP module verifies if the `blacklist-admin@blacklisted` address can be routed with the Router. If the Router Table contains such records (a default one or a different one), the error code sent back to the blacklisted host explains that mail to `blacklist-admin@serverdomain` name is accepted even from that blacklisted site.

If you want to provide a "white hole" feature, but you do not want the information about the white-hole address to be included into the error code, simply use a different name for the "white hole" address.

For example:

```
<abuse*@blacklisted> = postmaster
```

The following table contains samples of SMTP sessions established from a blacklisted host. The host commands are marked with C:, the SMTP module responses are marked with S:.

Router Table	
SMTP protocol	C: MAIL FROM: user@host S: 250 user@host sender accepted C: RCPT TO: somebody@somehost S: 591 Your host is in our Black List. No mail will be accepted C: RCPT TO: abuse@somehost S: 591 Your host is in our Black List. No mail will be accepted C: RCPT TO: blacklist-admin@somehost S: 591 Your host is in our Black List. No mail will be accepted
Router Table	<abuse*@blacklisted> = postmaster
SMTP protocol	C: MAIL FROM: user@host S: 250 user@host sender accepted C: RCPT TO: somebody@somehost S: 591 Your host is blacklisted. No mail will be accepted C: RCPT TO: abuse@somehost S: 250 abuse%somehost@blacklisted will leave Internet C: RCPT TO: blacklist-admin@somehost S: 591 Your host is blacklisted. No mail will be accepted
Router Table	<blacklisted-admin*@blacklisted> = postmaster
SMTP protocol	C: MAIL FROM: user@host S: 250 user@host sender accepted C: RCPT TO: somebody@somehost S: 591 Your host is blacklisted. Send your questions to blacklist-admin@mycompany.com. C: RCPT TO: abuse@somehost S: 591 Your host is blacklisted. Send your questions to blacklist-admin@mycompany.com. C: RCPT TO: blacklist-admin@somehost S: 250 blacklist-admin%somehost@blacklisted will leave Internet

Routing

The SMTP module immediately (on the first [Router](#) call) accepts messages addresses to *domain name*-wake up local addresses. When these messages are enqueued into the SMTP module queue, they are processed as [wake-up requests](#) for the domain name domain message queue.

The SMTP module also immediately accepts all addresses with IP-address domains, i.e. with domain names like [xx.yy.zz.tt]. Please note that the [Router](#) adds brackets to the IP-address domain names that do not have them, and the Router changes the IP addresses of local domains to those domain names. The Router performs these operations before calling the modules.

The SMTP module immediately accepts addresses that have domain names ending with .smtp. The .smtp suffix is removed, the domain name is used as the target host name, and the address "local part" is used as the envelope address to

pass to that host.

Sample 1:

The Server main domain is `company.com`.

Mail for the `sales.company.com` domain should be sent to a separate `sales.company.com` server via SMTP, while mail to all other subdomains of `company.com` should be processed as mail addresses to the main domain, i.e.

`user@subdomain.company.com` should be the same as `user@company.com`.

You can specify this routing as:

```
sales.company.com = sales.company.com.smtp ; explicitly direct to SMTP
*.company.com     = company.com             ; all other subdomains are rerouted
```

You can also specify this routing using IP addresses:

```
sales.company.com = [192.0.0.1]           ; explicitly direct to the IP address via
SMTP
*.company.com     = company.com             ; all other subdomains are rerouted
```

Sample 2:

All mail to the domains `client1.com`, `client2`, and `client3.com` should be sent to the site `host.com`.

You can specify this routing as:

```
client1.com = client1.com@host.com.smtp
client2.com = client2.com@host.com.smtp
client3.com = client3.com@host.com.smtp
```

or, in a more flexible way:

```
client1.com = client1.com@relay
client2.com = client2.com@relay
client3.com = client3.com@relay
relay = host.com.smtp
```

Note: You can specify just `host.com` instead of `host.com.smtp` here (given there is no other router record for the `host.com`), but in this case mail to `user@client1.com` will be sent to the `host.com` as `user%client1.com@host.com`. By specifying the `.smtp` suffix you not only tell the SMTP module to accept an address immediately, but you also force the SMTP module to send only the "local part" of the address to the remote host.

Address Processing without the `.smtp` suffix

<code>user @ client1.host</code>	Router converts to	<code>user%client1.host @ relay</code>
<code>user%client1.host @ relay</code>	Router converts to	<code>user%client1.host @ host.com</code>
<code>user%client1.host @ host.com</code>	Router stops	<i>no rule for host.com</i>
<code>user%client1.host @ host.com</code>	SMTP accepts	for <code>host.com</code> as <code>user%client1.host@host.com</code>

Address Processing with the `.smtp` suffix

<code>user @ client1.host</code>	Router converts to	<code>user%client1.host @ relay</code>
<code>user%client1.host @ relay</code>	Router converts to	<code>user%client1.host @ host.com.smtp</code>
<code>user%client1.host @ host.com.smtp</code>	SMTP accepts	for <code>host.com</code> as <code>user@client1.host</code>

On the final call, the SMTP module accepts mail to any domain if that domain name contains at least one dot (.) symbol. If the Forward option is selected, all these addresses (except those with IP-address domains) are rerouted to the specified Forwarding Server domain before the addresses are accepted.

Before accepting an address, the SMTP module checks if the address does not contain any @ symbol, but contains one or several % symbols. In this case, the rightmost % symbol is changed to the @ symbol.

Sending to Non-Standard Ports

Some mail servers can be configured to receive incoming SMTP mail on a non-standard port. The CommuniGate Pro SMTP module can send messages to those servers, if the domain part of an E-mail address contains the port number or is routed to an address that includes the port number.

There are two methods to include the port number into an E-mail domain:

- Use the IP-address notation: `[xx.yy.zz.tt:port]` where *xx.yy.zz.tt* is the IP address, *port* is the port number. Sample Router record:
`local.com = [192.0.0.5:26]`
- Use the `.smtp` suffix (see above) with the port number: `domain.port.smtp`. The SMTP module will not use the *domain* MX records in this case, it will try to resolve the *domain* name directly into an IP address. Sample Router record:
`secret.stalker.com = mail.stalker.com.26.smtp`



Local Delivery Module

The Local Delivery module processes messages routed to Accounts on your Server. It uses the Mailbox Manager to store messages in account mailboxes.

The Local module applies Account Automated Mail Processing Rules to all messages directed to that Account. Rules can instruct the module to store a message in a different mailbox, to redirect a message to different address(es), etc.

After a message is stored in an Account mailbox, it can be retrieved using any of Access modules.

The Local Delivery module can support Direct Mailbox addressing and Account Detail addressing.

The module can limit the number of messages each account can receive during the specified period of time. This feature allows the Server to minimize damage caused by mail loops.

Configuring the Local Delivery Module

Use a Web browser to connect to the CommuniGate Pro administrator port and open the LOCAL page in the Settings section.

Log:

Processing		Processes:
Keep for	if the account is	% full
Keep for		
	if Mail Service is disabled	
Send Warnings after		
Send Alertsif Account is	% full	
Alert Text:		

Log

Use the Log setting to specify what kind of information the LOCAL module should put in the Server Log. Usually you should use the Major (message transfer reports) or Problems (message transfer and non-

fatal errors) levels. But when you experience problems with the LOCAL module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly. The LOCAL module records in the System Log are marked with the LOCAL tag.

Processes

When you specify a non-zero value for this setting, the LOCAL module starts to process queued messages directed to local accounts. The module can use several simultaneous threads (processes) to deliver messages to several accounts at the same time. If you have more than 1000 accounts, or if you have many accounts with time-consuming automated [Rules](#), you should allow the module to use more than 1 process for message delivery.

Keep if the Account is full

When you specify a non-zero value for this setting, the LOCAL module checks the account mail storage before trying to deliver a message to that account. If the account storage is limited, and the specified percent of that limit is already used, the Local Delivery module delays all messages directed to the account. The module checks the account mail storage periodically and resumes message delivery when some messages are deleted from the account mailboxes.

This parameter specifies how long incoming messages should be kept in the module queue before they are rejected with the account is full error message.

Keep if the Mail Service is disabled

When you specify a non-zero value for this setting, mail sent to an account with disabled [Mail Service](#) option is not rejected immediately, but is left in the module queue for the specified period of time. When the administrator re-enables the account/domain Mail Service, the queued messages are delivered to the account.

Send Warnings after

If a message is delayed in the module queue (because the addressed Account is full or the Mail Service is disabled for that Account), the module can generate a warning message and send it back to the message sender. Use this setting to specify when the warning message should be generated.

Send Alerts

This option specifies if and when the [Account Quota Alerts](#) should be sent to account users.

Alert Text

Use this setting to specify the text of the alert message that will be sent to the account user when the account is over its quota.

Message Flow Control

While CommuniGate Pro employs many built-in techniques to prevent mail loops, in some situations (usually involving other servers) mailing loops still can occur. To minimize the damage caused by those loops, the Local Delivery Module counts all messages received by each Account. If this number exceeds the specified

limit, the incoming messages queue for that Account is suspended.

Note: The module counts the number of messages to be delivered to the Account, not the number of messages stored: even if an incoming message is not stored in the Account INBOX because an Account Rule has discarded it, the messages is still counted.

Flow Control	
Suspend Account Queue if Received	messageswithin

Routing

When the [Router](#) passes an address to the Local Delivery module, the module checks the domain name: if the domain name ends with the string `.local`, the Local Delivery module accepts the address, removes the `.local` suffix from the domain name, and stores the message in the Main Domain account with that name. This feature is used to create [Unified Domain-Wide Accounts](#).

Example:

a message sent to the address
`abcdef@nnnnn.local`

will be accepted with the Local Delivery module, and the message will be stored in the `nnnnn` account.

Sometimes, a Unified Domain-Wide Account should be created in a Secondary Domain, rather than in the Server Main Domain. Use the `.domain` suffix to direct mail to an account in a secondary domain. The last component of the address "local part" will be used to specify the name of the Secondary Domain account:

Example:

a message directed to the address
`abcdef%xyz@nnnnn.domain`

will be accepted with the Local Delivery module, and the message will be stored in the `xyz` account in the `nnnnn` domain.

When the Router calls the Local Delivery module for "first attempt", the module does not process any other addresses.

When the Router calls the Local Delivery module for "final delivery" attempt, it accepts all addresses with an empty domain name part or with the domain part equal to the name of a [secondary domain](#), and it routes the messages to the account specified with the "local part" of the address.

Examples:

a message sent to the address
`abcdef`

will be accepted with the Local Delivery module, and this message will be stored in the `abcdef` account in the main domain.

if subdomain.com is one of the secondary domains, a message sent to the address xyz@subdomain.com

will be accepted with the Local Delivery module, and this message will be stored in the xyz account in the subdomain.com secondary domain.

To provide the *domain-only* routing feature used within the [HTTP module](#), the Local Delivery module accepts all addresses with the LoginPage local part, and an empty domain part or a domain part equal to the name of a secondary domain or their aliases.

Routing to Unknown Accounts

When the Local Delivery module decides that an E-mail address is a local address, it checks that the account with the specified name exists. Each domain (the main one and each [secondary domain](#)) has a setting that instructs the Local Delivery module on what to do if a specified account does not exist.

If the selected option is "Rejected", all messages sent to unknown accounts are rejected, and the error message "unknown account" is returned to the sender.

If the selected option is "Discard", all messages sent to unknown accounts are rerouted to the NULL address, and the Server discards them without generating any error messages.

If you select the "Reroute to" option, all messages sent to unknown accounts will be rerouted to the specified address. That address can be a name of a registered local account, or it can be an E-mail address of an account on another server: the unknown account address is substituted with the specified address, and the Router restarts the address processing procedure.

The specified "rerouting" address may contain the asterisk sign. In this case the name of the unknown local account is used to substitute the asterisk sign.

Example:

the Reroute address is:

bad-*@monitoring.department.com

a message is sent to

james@mycompany.com

where mycompany.com is the domain name of your Server, and there is no account james on your Server.

The message is rerouted to:

bad-james@monitoring.department.com

Unified Domain-Wide Accounts

The Router can route an entire domain to a certain local account, if the .local domain suffix is used (see

above).

Example:

The Router line:

```
client1.com = C11.local
```

All messages sent to the client1.com domain are directed to the C11 local account.

Unified domain-wide accounts are useful if the client systems retrieves messages from your server using the [CommuniGate POP](#), the [CommuniGate Pro RPOP](#), or similar software that distributes retrieved messages locally. Alternatively, the client system can use a regular single-user mailer and then distribute retrieved messages manually.

While the information in the local part of the client1.com addresses is not used for routing, it is not discarded. When the Local Delivery module stores the message in the C11 account, it stores the local parts of the addresses in the X-Real-To: message header field (or other field specified in the Local Delivery module settings).

Example:

The Router line:

```
client1.com = C11.local
```

A message is sent to:

```
abcdef@client1.com, xyz@client1.com
```

It is stored in the C11 account, and a header field:

```
X-Real-To: abcdef, xyz
```

is added to the stored message

Note: the

```
<*@client1.com>= C11
```

foreign alias record also stores all messages sent to the client1.com domain in the C11 account, but if such a record is used, the information about the local part (account name) would be lost, and no X-Real-To: head fields would be generated. The client software that retrieves messages from this Unified account would have to rely on the To: and Cc: message header fields. Those fields do not always contain the correct information, and they never reflect any change in the local part of the address you could have done with some additional routing records.

The [POP module](#) allows individual users to retrieve mail from a Unified Account, by hiding out all messages that do not contain the specified username in the X-Real-To header field.

You usually create Unified Domain-Wide Accounts in the Main Domain. Use the .domain suffix to create an UDWA in a Secondary domain.

Messages routed to *xxxx%accountname@domainname.domain* will be stored in the *accountname* account in the *domainname* domain, with the *xxx* address being added to the message headers as the X-Real-To field.

For example, a Domain Administrator for the company.com domain may use the setting:

Mail To Unknown Addresses is Redirected to: **%Unknowns@company.com.domain*

and messages sent to unknown Domain Accounts will be stored in the account Unknowns, with all those unknown addresses stored in the message X-Real-To header fields.

Automated Mail Processing

After an address is accepted with the Local Delivery module, the message is queued to the Module queue. Each Module process takes messages from that queue, opens the addressed Account, and checks if the Account has Automated Rules specified.

If the Account has the Automated [Rules](#) specified, these Rules are applied: for each rule its conditions are checked, and if they are met, the specified Rule actions are performed. As a result of those actions, the message can be copied to some mailbox, a copy of the message can be redirected to some other addresses, an automatic reply can be generated, etc.

You can use a more detailed Log Level for the LOCAL module to see which Rules are applied to messages, why some conditions are not met, and what actions have been taken when all Rule conditions have been met.

Storing Mail in Account Mailboxes

After account [Rules](#) (if any) have been applied, and these Rules have not specified that the message should be discarded, the message is stored in the account INBOX.

The Local Delivery module checks the current size of the account mailboxes and rejects a message if the account storage quota would be exceeded.

Direct Mailbox Addressing

The Local Delivery Module can deliver messages directly to non-INBOX mailboxes. If the local part of the address is specified as *box#name*, then the message will be stored in the *box* mailbox in the *name* account.

The Account-Level rules are NOT applied if such an address is used.

You can use Direct Mailbox Addressing in the [Router](#) Table:

```
; store messages to sales@maindomain
; in the sales mailbox in the account public@maindomain
<sales> = sales#public
;
; store messages to support@client.com
; in the requests mailbox in the account staff in the hq.client.com domain
<support@client.com> = "requests#staff"@hq.client.com
```

Note: remember that mailbox names are case-sensitive.

Note: the Direct Mailbox Addressing feature can be used via the [POP module](#), too. With the sample Router records listed above, when a user logs in using the name `sales`, the client POP mailer is connected to the mailbox `sales` in the `public` account (if the user has provided the correct password for the `public` account).

Routing Settings

Routing
Envelope Recipients field:
Direct Mailbox (mailbox#account):
Account Detail (account+detail):

Envelope Recipients field

This setting specifies the name of the message header field the Local module generates when it stores messages in [Unified Accounts](#).

Direct Mailbox Addressing

This setting specifies if [direct mailbox addressing](#) is enabled.

Account Detail

This setting controls account detail addressing. Account detail address is an account name followed with the plus sign (+) and some string. You can set this settings to:

Disabled

The Local Delivery module will not process the plus signs in account names.

Enabled

The Local Delivery module will check for the plus sign in account names and delete the first plus sign and all following symbols from the address, then it will re-route the address. Users can use account detail addresses (`john+jokelists`) to subscribe to mailing lists. Messages sent to account detail addresses will be routed to the user accounts, and Account-level [Rules](#) (the Recipient condition) can be used to process those message automatically - for example, to store them in some jokes mailbox dedicated to these list messages.

Direct Mailbox

The Local Delivery module will check for the plus sign in account names and process the string after the plus sign as the [Direct Mailbox](#) address. The `john+jokelist` address will be processed as the `jokelist#john` address and the message will be routed directly to the `jokelist` mailbox of the `john` account, bypassing Account-Level Rules (if the Direct Mailbox Addressing option is enabled).

Sending Mail to All Accounts

The [Domain Settings](#) can be used to enable the virtual object All. Messages sent to the `all@domainname` address are stored in INBOXes of all Domain Accounts that have the Accept Mail To All option enabled.

Note: the individual Account Rules are not applied to messages sent to the `all` address.

The `alldomains@maindomain` address can be used to send messages to all Accounts in all Domains.

All-Domain Aliases

All-Domain Aliases	
Local Address	Reroute To

This table allows you to specify aliases that will work for all local domains. When the CommuniGate Pro Server detects that a message should be directed to some name in one of the Server local domains, these records are checked. If the local part of the address matches the Local Address field in one of these records, the message is rerouted to the address specified in the Reroute To field.

If, for example, the `abuse` and `postmaster@maindomain.dom` addresses are entered into the All-Domain Aliases table (as shown above), then all messages directed to any `abuse@domain.dom` address (where `domain.com` is one of the CommuniGate Pro Domains) are rerouted to the `postmaster@maindomain.com`.

Note: it's easy to create routing loops using these records: if you enter

```
postmaster -> postmaster@maindomain.dom
```

into this table, you will create a loop that will make it *impossible to connect to the Server as postmaster*. If you want mail to all postmaster names in all domains to go to the postmaster account in the main CommuniGate Pro domain, you should use:

```
postmaster -> anyname@postmaster.local
```

or, if the Direct Mailbox Addressing option is enabled:

```
postmaster -> mailboxName#postmaster
```



RPOP Module

The CommuniGate Pro RPOP implements E-mail message retrieval using the POP3 Internet protocol (STD0053) via TCP/IP networks. While the [POP](#) module allows the CommuniGate Pro users to retrieve mail from their Server mailboxes, the RPOP module retrieves messages from other (remote) hosts and delivers them to user mailboxes or to other destinations.

For each registered user, the RPOP module can retrieve messages from several remote mailboxes. The RPOP module can retrieve mail for your entire domain using "Unified Domain-wide accounts" and distribute retrieved messages to their recipients.

Post Office Protocol (POP3) and Mail Retrieving

The RPOP module can be used when the CommuniGate Pro Server has a dial-up connection with dynamically assigned IP address, and thus the Server cannot receive mail via SMTP. The RPOP module polls the specified remote host (ISP) accounts, retrieves messages and stores them in the Server mailboxes.

The RPOP module is useful even if the CommuniGate Pro Server has a full-time Internet connection. A user that has several accounts on several hosts can instruct the RPOP module to poll those accounts, so all their mail is collected in their CommuniGate Pro account.

The RPOP module supports Domain-Wide Accounts. A Domain-wide account is an account on the ISP or any other host that collects all messages sent to your domain. The RPOP module retrieves all messages from such an account and distributes them based on the addressing information in the message headers. The RPOP module can poll several Unified Domain-Wide Accounts.

The RPOP module activity can be limited using the [TCP Activity Schedule](#). The module does not poll any remote account till the TCP Activity Schedule allows the Server to initiate outgoing network connections.

Configuring the RPOP module

Use a Web browser to configure the RPOP module.

Log:

Polling	Channels Limit:
Delay Failed Hosts for:	Use APOP
Delay Failed Accounts for:	
Minimum Poll Period for Users:	Allow Self-Poll
Maximum Number of Accounts per User:	

Log

Use the Log setting to specify what kind of information the RPOP module should put in the Server Log. Usually you should use the Major (message transfer reports) or Problems (message transfer and non-fatal errors) levels. But when you experience problems with the RPOP module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly.

The RPOP module records in the System Log are marked with the RPOP tag.

Channels Limit

When you specify a non-zero value for the Channels Limit setting, the RPOP module starts to connect to the remote hosts and retrieve mail from accounts on those hosts. The setting is used to limit the number of simultaneous connections the RPOP module can initiate.

Use APOP

The RPOP can use the secure APOP authentication method when connecting to hosts that support this feature. If for any reason you want the RPOP module to always use the "clear text" passwords, disable the Use APOP option.

Delay Failed Hosts

When the RPOP module fails to connect to an external host, it marks the host as "failed" and stops polling all accounts on that host. The option specifies when the RPOP module should try to poll the failed host again.

Delay Failed Account

When the RPOP module fails to open a mailbox (wrong password, remote mailbox is locked, etc.), or if the connection fails when the module retrieves messages from a remote account, the module marks an account as "failed". The option specifies when the RPOP module should make the next attempt to poll the failed account.

Allow Self-Poll

Very often CommuniGate Pro users misunderstand the concept of remote account polling and specify their own CommuniGate Pro accounts as the "remote" accounts to be polled. This creates message loops and wastes Server resources. If this option is not selected, the RPOP module checks the network address of the remote POP server it has to connect to. If that address is one of the CommuniGate Pro Server own network addresses, the "remote" account is not polled.

Minimum Poll Period for Users

If some users are allowed to specify their own [individual RPOP accounts](#), they may select too short Poll Periods, generating a lot of network traffic and consuming the server resources. Use this option to set the minimum value the Server users can specify in their Poll Every remote account settings. This limit applies to users only. The administrator can always specify any Poll Period for the Unified Accounts and for individual RPOP accounts.

Maximum Number of Accounts per User

If some users are allowed to specify their own [individual RPOP accounts](#), they may specify too many accounts, generating a lot of network traffic and consuming the server resources. Use this option to limit the number of RPOP accounts the Server users can specify. This limit applies to users only. The administrator can always specify any number of Unified and individual RPOP accounts.

Click the Update button to modify the RPOP module settings.

Specifying Unified Domain-Wide Accounts

If a mail account on an external host collects mail directed to all users of your domain, the RPOP module can be instructed to retrieve mail from that account and distribute it to local users.

Unified Domain-wide Accounts				
Poll Every	Account	at Host	Password	LeaveAPOP Special Header

Poll Every

This option specifies how often the RPOP module should poll the remote account.

Account

This option specifies the name of the mail account on the remote host. For Unified Domain-Wide Accounts, this name is usually your domain name or part of your domain name.

at Host

This option specifies the exact name of the POP server that should be polled. Please note that this could be the name of a specific computer (as specified in DNS A-records), not just a generic domain name of the provider system. For example, if the provider has the domain name provider.com, its POP server is usually named mail.provider.com or pop.provider.com. Consult with your provider.

Standard POP servers accept incoming connections on the TCP port 110. If you need to poll an account on a remote POP server that uses a non-standard port, specify the port number after the host name, using the colon (:) symbol as the separator:

```
pop.provider.com:111
```

Password

The password to use to log into the remote account.

Leave

If this option is selected, the RPOP module does not delete messages from the remote account mailbox. Instead, it remembers the UID (Unique Identifier) of the retrieved messages, and the next time the RPOP module polls this remote account, it does not retrieve messages that have the same UIDs.

If you want to use this option, verify that the remote POP server supports the UIDL command.

APOP

If this option is selected AND the UseAPOP module option is enabled AND the target host advertises APOP capability in its initial prompt, the RPOP module uses the secure APOP method for authentication on that remote host.

Special Header

The name of the messages header (RFC822) field that the provider host inserts into the messages stored in the Unified Domain-Wide Account (see below).

There is always an empty row in the Unified Accounts table. Use it to specify a new Unified Account. To remove an account, set the Poll Every option to never.

Click the Update button to modify the RPOP module list of the Unified Domain-Wide Accounts.

Special Headers and Mail Distribution

When a message is sent via the Internet, the information about the sender and the message recipients is sent in the so-called mail envelope. If mail is sent via SMTP, the envelope is sent as a sequence of the protocol commands, if mail is sent via UUCP, the envelope is sent using additional files. The information in the envelope is usually the same as the information in the message headers, but it is not always true. The most important exceptions are:

- the message headers do not contain the addresses of the Bcc recipients
- the headers of a mailing-list message do not contain the addresses of the mailing list subscribers.

When a message is stored in a mailbox, the envelope information about the sender is added to the message headers as the Return-Path header field. Usually, the envelope information about the recipients is not added to the message headers.

When the RPOP module retrieves a message from a Unified Domain-Wide Account, it has to recompose the message envelope and deliver the message to its final recipient. If the message contains the Return-Path header field, the address in that field is placed in the new envelope as the sender's address, and the header field is removed from the message (it will be recreated when the message is delivered to its final destination).

If a Unified Domain-Wide Account is created with the mail system that can copy the recipient addresses from the envelope into some message header field, then the delivery via RPOP is as reliable as SMTP delivery.

Enter the name of that header field into the Unified Account settings, and the RPOP module will look for that field in all messages retrieved from that account. The addresses from that field will be placed into the new envelope and the messages will be directed to those addresses. The header field itself is removed from the message. All accepted addresses get the 'report on failure' flags, so if message delivery fails, the original message sender (the address in the message Return-Path field) will receive an error report.

All Stalker mail servers can be used to provide Unified Domain-Wide Accounts. For those accounts, the envelope recipients are added to the message headers as the X-Real-To fields. To learn how to provide Unified Domain-Wide Accounts with CommuniGate Pro, check the [Local Delivery module](#) section.

A legacy sendmail system can be configured to add X-Real-To header fields, too. See the [Appendix A](#) below.

Mail Distribution without Special Headers

Many ISPs still use various legacy mail systems that cannot store envelope recipients in message headers. If you have to host your Unified Domain-Wide Account on such a system, leave the Special Header field empty.

The RPOP module will search for all `To :`, `Cc :`, and `Bcc :` header fields in retrieved messages. It will use the addresses from those header fields only if that address is routed to any existing local CommuniGate Pro Account.

If an address is routed to the SMTP or some other module, or an address cannot be routed at all (unknown user name error, etc.), the RPOP module does not send any error messages to the sender. The module simply ignores that address.

All accepted addresses get the 'do not report failures' flags, so if the message delivery fails for any reason, no error report is sent to the original message sender.

If none of the message `To :`, `Cc :`, or `Bcc :` addresses has been accepted, the RPOP module sends that message to the `postmaster` Account in the Main Domain.

As explained above, the method based on `To :/Cc :` header field parsing can cause problems when the actual envelope addresses are not the same as the header field addresses. Besides, some systems do not process the Unified Accounts correctly, so if a message is sent to three users in your domain, those systems may store three copies of the message in the Unified Domain-Wide Account mailbox. Since each message header contains the addresses of all three users, the RPOP module will deliver three copies of the message to each user.

The problems with Bcc, mailing lists, and duplicated message can be very annoying, so we strongly recommend you to ensure that the provider's mail system adds envelope information to the messages stored in your Unified Domain-Wide Account, so you can use the Special Header feature.

Specifying Remote Accounts for Individual Users

The CommuniGate Pro RPOP module can poll POP accounts on remote hosts on behalf of the CommuniGate Pro users (Accounts). For each CommuniGate Pro user several external POP accounts can be specified. External accounts can be specified by the Server administrator, via a link on the [Account settings](#) page, or by the users themselves, via the WebUser Interface, if the right to specify remote POP accounts has been granted to the user.

Poll Every	Account	at Host	Password	Leave	Last
					12:34:56

The settings are the same as for the [Unified Accounts](#), but the Special Header field is not presented.

All messages retrieved on the user behalf are directed to that user, regardless of the message header contents.

Last

If the last attempt to retrieve mail from the remote account was successful, this field tells when (in the server local time) this attempt took place.

If the last attempt was not successful, the field contains the error code.

All messages retrieved for individual CommuniGate Pro accounts are sent to those accounts via the CommuniGate Pro [Queue](#), so all Server-Wide and Account-Level [Rules](#) are applied to those messages.

All messages retrieved for individual CommuniGate Pro accounts get the 'do not report failure' flags, so if delivery was unsuccessful, no error report is sent to the original message sender.

Appendix A. Configuring `sendmail` for Unified Domain-Wide Accounts

The following file can be used to force `sendmail` to store the envelope information in the messages headers.

```
# This file should be placed into the directory cf/feature from
# the sendmail.8.X.XX.cf.tar.Z archive.
# To add special headers, the macros `FEATURE(xrealto)' should be
# added to the main configuration file in the directory cf/cf,
# and the flag T should be added to the mailer description.
#
# This file adds special headers with the `X-Real-To' keyword.
# The special headers will be added to all messages routed to the
# mailer marked with the `T' flag in the sendmail configuration.
divert(0)
VERSIONID(`@(#)xrealto.m4 0.1 1/4/96')

divert(9)
# add the X-Real-To: header field to the message
# if the mailer is marked with the `T' flag
H?T?X-Real-To: $u
divert(0)
```

After these updates are applied, make sure that `sendmail` delivers all mail for your domain to one account on the `sendmail` system. The `sendmail` configuration for that unified account should list the 'mailer' marked with the 'T' flag.



LIST Module

The CommuniGate Pro LIST module implements the mailing list mechanism.

Mailing Lists

The system administrator can create one or several mailing lists. Users from the same or any other mail system can subscribe to these mailing lists using the Web interface or by sending E-mail. They can post messages on mailing lists by sending E-mail to the list addresses, and posted messages are delivered to all subscribers. All posted messages are stored in the mailing list mailbox that serves as an archive.

If a user subscribes in the FEED mode, all posted messages are redirected to that user immediately after they are received by the LIST module.

When a user subscribes in the DIGEST mode, the user starts to receive list digests: a multi-part messages generated with the LIST module for each mailing list. Each digest message contains all messages posted on the list since the last digest was generated, prefixed with an index of these messages.

When a user subscribes in the INDEX mode, the user starts to receive messages containing the indexes of newly posted messages. If the user wants to read some of the posted messages, they can use a Web browser to access the mailing list archive.

Configuring the LIST module

To configure the LIST module, use any Web browser to connect to the CommuniGate Pro Server, and open the LIST page in the Setting section. To open pages in the Settings section, you should have the Can Modify Settings access right.

Log:

Log

Use the Log setting to specify what kind of information the LIST module should put in the Server

Log. Usually you should use the Major (message processing reports) level. But when you experience problems with the LIST module, you may want to set the Log Level setting to Low-Level or All Info: in these cases more details will be recorded in the System Log. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly.

The LIST module records in the System Log are marked with the LIST tag.

Processing	Processes:

Processes

When you specify a non-zero value for the Processes setting, the LIST module starts to process queued messages directed to mailing lists, starts to generate digests, and starts to clean the mailing list archives. The module can use several simultaneous threads (processes) to process several mailing lists at the same time. If you have more than 50 mailing lists, or if you have many lists with extremely large (10,000+) subscriber lists, you should allow the module to use more than 1 process.

Mailing Lists		
Name	Domain	Owner
SIMS	node5.stalker.com	Ali
CGPro	test.stalker.com	kwa

This table shows all mailing lists created. By following the links in the table, you can open an individual list setup page, the [Domain Settings](#) page for the list domain, or the [Account Settings](#) page for the mailing list owner.

Creating Mailing Lists

Each mailing list is created inside the main or one of the secondary [domains](#), and each mailing list belongs to its owner - an Account in the same domain as the mailing list.

To create a mailing list, create an Account or choose an existing one - the Account of the mailing list owner. Open the [Account Settings](#) page, enter the name of the mailing list to create and click the New List button.

The Server checks that there is no Account, Group, Forwarder or Mailing List with the same name in the same Domain, and creates a new Mailing List.

For a mailing list *listname*, several mailboxes are created in the list owner Account:

listname this mailbox is the mailing list archive: it contains the messages posted on the mailing list

listname/requests this mailbox contains the messages with subscription requests

listname/reports this mailbox contains bounce and other DSN (Delivery Status Notification) messages generated for the messages distributed via this mailing list.

listname/approval this mailbox contains postings that require the list owner approval (moderated postings). To post these messages, the list owner should redirect them back to the mailing list using a secure submit method: the CommuniGate Pro [Web User](#) Interface, the XTND XMIT [POP3](#) method, a local "mail" command, the [PIPE](#) module, etc.

Configuring Mailing Lists

To configure a mailing list, open the Mailing List Settings page. You should select a link to the Settings page either from the list of all mailing lists located in the LIST module Settings page, or from the list owner [Account Settings page](#).

You should have the [All Accounts and Domains](#) or the [Domain Administrator](#) access right in order to open the Mailing List Settings pages via the WebAdmin Interface.

The list owners can access the Mailing List settings pages using the WebUser Interface with their Accounts: the page that lists the account mailboxes also provides the links to the account mailing lists.

Log:	Subscribers	Owner: ListMaster
-------------	-----------------------------	--

Log

Use the Log setting to specify what kind of information about this mailing list should be put in the Server Log. Usually you should use the Major (message posting, subscription, digest, and clean-up operations) level. But when you experience problems with this particular mailing list, you may want to set the list Log Level setting to Low-Level or All Info: in this case more details will be recorded in the System Log. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly.

The mailing list records placed in the System Log are marked with the `List(listname)` tag. Only the system administrator can change the mailing list Log setting.

Description:

Preferred Character Set:

Digesting and Archiving:

Verify Owner Using:

[Settings](#)

Description

Use the Description setting to specify the full name of the list. It will be used as a "comment" in the list E-mail address as the Real Name setting is used in account E-mail addresses.

Preferred Character Set

This option specifies how the List module should handle non-ASCII texts. It is used when displaying list messages via the Web interface: messages in the list can be composed using different character sets, and to display them all on one page, the module should know which character set is preferred.

Digesting and Archiving

Posted messages can be stored in a mailbox created in the list owner Account. Such a mailbox is used to collect messages for message digests. If messages are not removed from that mailbox after a digest is created, this mailbox can be used as a mailing list archive.

Enabled

All posted messages are stored in the owner Account mailbox. Use the link next to the pop-up menu to open the Digesting and Archiving Settings page.

Disabled

The posted messages are not stored in the owner Account mailbox, archiving and digesting options are disabled. All digest and index-mode subscribers are processed as feed-mode subscribers. All new attempts to subscribe to the mailing list in the digest or index mode are rejected.

Verify Owner

When the LIST module receives an E-mail message, it checks if the message is sent by the List Owner. First, the message Return-Path is compared to the list owner E-mail address. The Return-Path should be *ownerName@listdomain*, where the ownerName is the owner Account name (not one of its aliases), and the listdomain is the list and owner Account Domain name (not one of its domain aliases).

The Verify Owner setting specifies the additional checks to be made:

Return-Path

When this option is selected, no additional check is made.

IP Addresses

When this option is selected, the LIST module checks that the message has been submitted either using one of the *authenticating* methods (see below), or via SMTP, from a computer with an IP address included into the [Client IP Addresses](#) list.

Authentication

When this option is selected, the LIST module checks that the message has been submitted using one of the *authenticating* methods:

- via [SMTP](#), using the AUTH authentication;
- via [WebUser Interface](#);
- via [POP](#) with XTND XMIT extension;
- via the [PIPE](#) module.

The Settings page contains a set of options, settings, and text areas that controls the subscription, postings, message distribution, and bounce processing. See the sections below for the details.

Renaming Mailing Lists

To rename a mailing list, open that list Settings page via the System Administrator Web interface (see above).

You should have the [All Accounts and Domains](#) or the [Domain Administrator](#) access right in order to open the Mailing List Settings pages.

Enter the new name for the list in the Mailing List Settings page, and click the Rename button. The Server checks if there is no Account or Mailing List with the new name in the same Domain and renames the mailing list. Then the Mailing List Settings page is reopened.

Removing Mailing Lists

To remove a mailing list, open that list Settings page via the System Administrator Web interface (see above).

Click the Remove List button. A confirmation page appears. If you click the Remove button on the confirmation page, the list is removed, the files with the list subscribers and list settings are deleted, and the mailing list mailboxes are removed from the list owner Account.

Composing Service Texts

Mailing lists have settings that specify text to be sent to the subscribers - message header and trailers, confirmation requests, etc. A specified text can contain special symbol combinations to be substituted with actual data before the text is inserted into a message.

The following symbol combinations can be used in all textual settings:

Combination Substituted with

^N	the <i>listname</i> string
^D	the <i>domain</i> string
^E	the Description setting
^X	the sequence number of the current digest

If there is a number after a special string combination, as in ^N80, the number specifies the maximum length of the substitution string. If a substitution is longer than specified, its last symbols are cut off.

If there is a number after a special string combination, and the number starts with 0, as in ^N040, the number specifies the length of the substitution string. If a substitution string is longer than specified, its

last symbols are cut off, and if a string is shorter than specified, space symbols are added to the beginning of the string.

Subscription Processing

Each mailing list is a list of subscribers, i.e. a list of E-mail addresses receiving messages posted on the mailing list.

In order to subscribe, unsubscribe, and change their [subscription mode](#) via E-mail, users of the *listname@domain* mailing list should send any message to the following addresses:

Send to address:	New user	Existing subscriber
<i>listname-on@domain</i> or <i>listname-subscribe@domain</i>	to subscribe to the list in the default mode	to confirm the current subscription mode
<i>listname-feed@domain</i>	to subscribe to the list in the FEED mode	to change the subscription mode to FEED
<i>listname-digest@domain</i>	to subscribe to the list in the DIGEST mode	to change the subscription mode to DIGEST
<i>listname-index@domain</i>	to subscribe to the list in the INDEX mode	to change the subscription mode to INDEX
<i>listname-null@domain</i>	to subscribe to the list in the NULL mode	to change the subscription mode to NULL
<i>listname-off@domain</i> or <i>listname-unsubscribe@domain</i>		to unsubscribe from the list
<i>listname-confirm@domain</i>		to get the confirmation ID; this ID can be used as the password for other subscription operations and for list archive browsing

When subscription request messages are sent to these addresses, the message **From:** header fields are used as requesters E-mail addresses.

You can specify who can subscribe to the mailing list, and how they should subscribe.

Subscription Policy

Subscribe:	Save	Requests
Default Mode:	Request Confirmations	
Confirmation Request Message		
Subject:		
Text:		

- Subscribe
- nobody users cannot subscribe to this mailing list and/or change the subscription mode by themselves. Only the system administrator and the list owner can do these operations. Users still can unsubscribe by themselves.
 - this domain only only users from the mailing list domain can subscribe.
 - locals Only only users registered with this server can subscribe.
 - anybody any user on any system can subscribe to this list.
 - moderated all subscription requests will be stored in the *listname/requests* mailbox (see below) and should be approved by the list owner. The List owner should redirect these requests to the *listname-on*, *listname-feed* and other addresses to subscribe new users and to let them change the subscription mode. Users still can unsubscribe by themselves. When user requests are stored in the *listname/requests* mailbox for approval, they are "flagged" (get the Flag marker).

Save Requests

This setting specifies which subscription requests (i.e. messages sent to the *listname-on*, *listname-off* and other addresses listed above) should be stored in the *listname/requests* mailbox in the list owner Account. Messages stored in this mailbox can be examined if some user reports problems when trying to subscribe to, unsubscribe from, or change the subscription mode to this list.

You can specify if none, all, only the accepted, or only the rejected request messages should be stored in the *listname/requests* mailbox.

Requests waiting for the list owner approval are always stored in the *listname/requests* mailbox.

All stored request messages get the X-List-Report additional header field . This field contains the list manager report (Delivery Status Notification message). It is recommended that the List owners configure their mailer applications so they display the X-List-Report fields.

Default Mode

When a new user sends a subscription request in to the *listname-on@domain* or *listname-subscribe@domain* address, i.e. when the subscription mode is not specified, the mode specified with the Default Mode setting is used.

Subscription Modes

Each list user (E-mail address) is subscribed in one of the following modes:

FEED

In this mode, the subscriber receives list messages as they are posted. See the [FEED Mode Distribution](#) section for more details.

DIGEST

In this mode, the subscriber periodically receives *digest* messages. A digest message starts with the Table Of Content - the list of the messages posted, followed by the posted messages themselves. See the [DIGEST/INDEX Mode Distribution](#) section for more details.

INDEX

In this mode, the subscriber periodically receives *index* messages. An index message is the same as the digest Table Of Content, but it does not contain the posted messages themselves. INDEX subscribers can see if they are interested in any posted messages, and use the Web interface to read those messages in the mailing list archive. See the [DIGEST/INDEX Mode Distribution](#) section for more details.

NULL

In this mode, the subscriber does not receive any messages from the list. This mode can be used by the "posters" - the list users that only post messages on the list.

BANNED

The "banned" users do not receive messages from the list, and they cannot change their subscription mode themselves.

You can use this method to make it impossible for certain users to subscribe to your mailing lists, though usually the more generic anti-spam and other system-wide protection methods should be used.

Confirmation Requests

There are several very common problems with most of publicly available mailing lists:

- subscription messages are sent from incorrect addresses, those incorrect addresses are inserted into the subscribers list, and mailing list messages are repeatedly sent to those incorrect or unused addresses.
- list abusers (such as spammers) subscribe nonexistent E-mail addresses just to allow themselves to post on the lists that accept posts from subscribers only.
- using easily forged message headers, some persons can subscribe and unsubscribe another person E-mail addresses.

These and some other problems can be solved using confirmation requests.

Request Confirmations

When this option is selected, the List manager does not fulfill subscription requests immediately. Instead, a confirmation request message is composed and sent to the address that is about to be included or excluded from the subscribers list. The confirmation request contains a unique identifier (Confirmation ID) in its Subject field. When the user receives such a confirmation request, they can simply use the mailer Reply command to confirm the requested operation.

Confirmation Request

This text settings allows you to specify the text of confirmation requests sent to the subscribers. In addition to common "symbol combinations", this service text can contain the following combinations:

Combination	Substituted with
^O	the requested operation
^P	unsubscribe for the unsubscribe and subscribe operations and subscribe(<i>operation</i>) for other operations
^A	the subscriber address
^I	the confirmation identifier

Welcome and Good Bye Messages

Welcome/Policy Message	
Subject:	
Text:	
Good Bye Message	
Subject:	
Text:	

When a new user is subscribed, the Policy Text message is sent to that new user. When a user unsubscribes, the Good Bye message is sent.

Besides the generic "symbol combinations", these service texts can also contain the following combinations:

Combination	Substituted with
<code>^A</code>	the subscriber address
<code>^I</code>	the confirmation identifier

Posting Messages

To post a message on the mailing list, the author should send it to the *listname@domain* address.

Posting Policy	
Accept Postings:	New Subscribers:
Allowed Format:	Maximum Size:
Prohibit:	Non-matching Character Sets
	Unmodified Digest Subjects
Hide 'From' Addresses	

Accept Postings

This setting specifies who can post messages on this mailing list:

<code>from owner only</code>	Only messages submitted by the list owner (using any secure method) will be posted
<code>moderated</code>	Messages from everybody but the list owner are redirected to the list owner for approval; if the list owner redirects a message back to the list, the message is posted. Note: this mode can be used to change the list posting policy when the list owner wants to postpone all postings. If you use the <code>from subscribers</code> mode, the New Subscribers setting (see below) provides more advanced moderation options.
<code>from subscribers</code>	Messages from the list subscribers are accepted for posting; some messages can be moderated (see below).
<code>from anybody</code>	Messages from any address are accepted for posting.

New Subscribers

This settings is effective only if the Accept Postings setting is set to `from subscribers`. When a user subscribes and starts to post messages, the messages are stored in the list owner mailbox waiting for approval. After the specified number of messages is approved and posted, all messages sent by this user are posted on the list directly, without the list owner approval.

Note: this is a very effective way to enforce the list policies and to protect the mailing list from "spamming".

If you set this option to `Moderate All`, then all messages from new subscribers will be stored

for approval (the LIST module will not update the posted messages counter).

If you set this option to `Prohibited` new subscribers won't be able to post on the list (their postings will be rejected).

If you set this option to `Special`, new subscribers will be able to post auto-generated messages.

This is useful if you want to subscribe this mailing list to other mailing lists.

Note: when an auto-generated message should be posted on the list, and the message is not a message generated by the list owner, the message `Sender` address (if exists) is used instead of the `From` address. If that address is subscribed to the list, and the subscriber posting mode is set to `Special`, the message is posted.

You can open the [subscribers list](#) and change this setting for individual subscribers. You may want to change this setting to `Unmoderated` for some users, letting only those users post messages on the list, while posting from all other users (and new users) will be either stored for approval or rejected.

Allowed Format

This settings specifies the allowed MIME format for postings.

<code>plain text only</code>	only messages in the text/plain format can be posted
<code>text only</code>	only messages in the text (text/plain, text/html, etc.) formats can be posted
<code>text alternative</code>	only messages in the text format or multipart/alternative messages that contain a part in the text format can be posted (for example, a message can contain a text/html part and the same text as a image/gif part/variant)
<code>anything</code>	messages in any MIME format can be posted

The list owner can always post messages in any format.

Maximum Size

This settings restricts the size of messages that can be posted on this mailing list. The list owner can always post messages of any size.

Prohibit Unmodified Digest Subjects

When this option is selected, the Subject fields of all postings are checked. If the Subject is a "reply prefix" (such as `Re:`, `Re>`, etc.) followed by this list Digest String (see [below](#)), the message is rejected.

Prohibit Non-matched Character Set

When this option is selected, the character set used to compose the posting is checked. If the character set is explicitly specified, and it does not match the Preferred Character Set for this list, the message is rejected.

Hide 'From' Addresses

When this option is selected, the 'From' address of the posted messages is modified, so it contains the list address instead of the message author address. If the original 'From' address contained a comment (a real name of the message author), it is preserved.

The 'From' addresses in the message archives, digests, and indices are converted, too.

You can use this option to 'hide' the real E-mail addresses of those people who post messages on this mailing list.

Processing Messages

When a posted message is being sent to subscribers, the original message header is modified. Usually, only the From, Date, Message-ID, and Subject fields are copied from the original message.

You can specify additional header fields to be copied from original postings to the messages sent to subscribers - directly or as parts of digests.

RFC822 Fields to Keep
To and Cc

RFC822 Fields to Keep

Use these fields to specify the names of additional RFC822 header fields to be copied from the original postings to the distributed messages.

If you want to remove a name, enter an empty string into the name field.

If you want to copy all header fields, enter the asterisk (*) sign into a field.

If the To And Cc option is selected, all To and Cc addresses from the original messages are included into the distributed messages (as its Cc-addresses).

Bounce Processing

Incorrect and expired E-mail addresses create the most annoying problems for mailing list administrators. The CommuniGate Pro LIST module automates processing of incorrect, expired, and temporarily unavailable addresses.

All messages sent to subscribers (in all modes) have message envelope information that routes all error reports back to the LIST module. When a report is received, the LIST module:

- stores the report in the *listname/reports* mailbox in the owner Account (optional);
- parses the report text;
- if the report has a correct delivery-report ([RFC1892](#)/[RFC894](#)) structure, processes all records in the report.

Most of the problems can be detected immediately when sending a list message from the CommuniGate Pro Server, so most of the error reports are generated locally, on the same CommuniGate Pro server. The CommuniGate Pro server generates reports in the proper format, so most of the delivery problems are handled automatically.

If a list message has been sent to a remote site without a problem, but then that remote site fails to deliver the message to the recipient, the delivery report is generated on that remote site. Most of the modern mail servers generate delivery reports in the correct format, so in this case many problems are handled automatically, too.

And, finally, it is still possible to receive unformatted delivery reports from other sites. The LIST module can store those unformatted reports in the *listname/reports* mailbox, so the list owner can process them manually.

Each record in the delivery report contains information about one E-mail address, and indicates if the original message was or was not delivered to that address. It can also specify if the delivery problem is fatal (as when account is removed from the system), or if it is a non-fatal, temporary problem (as when a remote site is down or when the account disk space quota is exceeded).

If a non-fatal report is received, the E-mail address in question is suspended, and no list messages are sent to that address, and all additional error reports about that address are ignored. When the LIST module performs periodic list clean-ups, it sends a warning message to all suspended addresses. The warning message notifies that user that some list messages have not been delivered to the user address, and it also asks the user to confirm subscription (by replying to the warning message).

There are two ways to specify the suspension period:

- An address can be suspended for a fixed period of time. When that period ends, the LIST module resumes sending list messages to this address. This can result in new bounces which will increment the bounce counter, and the address is unsubscribed after the bounce counter exceeds the specified limit. The warning messages are sent to the failed address after the suspension is over, too - till the user confirms the subscription by replying to the warning message.
- An address can be suspended till the user confirms subscription by replying to the warning message. If no confirmation is received during the specified period of time, the address is unsubscribed.

When a user confirms the subscription by replying to the warning message, the suspension period ends, and the bounce counter associated with the user E-mail address is cleared.

Bounce Processor		
On a Non-Fatal Bounce:	suspend subscription for:	
	unsubscribe after:	bounces
	suspend till confirmation	
	unsubscribe after:	
Process a Fatal Bounce as:		
When Unsubscribing:	Notify Owner	
Cleanup List every:		
	Save	Bounce Reports
Warning Message		
Subject:		
Text:		

suspend subscription for

If this option is selected, it specifies for how long a subscriber should be suspended if the system receives a non-fatal problem report about the subscriber's E-mail address.

The `unsubscribe after` option specifies the number of unconfirmed suspension periods after which the user is unsubscribed.

suspend till confirmation

If this option is selected, a non-fatal error report suspends the address till the subscriber sends a confirmation message.

The `unsubscribe after` option specifies the time period to wait for a confirmation message.

Process a Fatal Bounce as

This setting specifies how the system should process fatal problem reports: as non-fatal, as several non-fatal, or as a fatal problem. If you specify that the system should unsubscribe a user after receiving 10 non-fatal problem reports about the user address, and you specify that a fatal problem report should be processed as 5 fatal, this will tell the system to unsubscribe the user after 2 fatal reports. If you specify that a fatal problem report should be processed as fatal, the system will unsubscribe a user immediately upon receiving a fatal problem report.

Notify Owner When Unsubscribing

If this option is selected, an E-mail message is sent to the List Owner every time an address is unsubscribed because of mail bouncing.

Cleanup List every

This setting specifies how often the system should scan the subscription list. When scanning the list,

the system:

- sends warning messages to the subscribers with non-zero bounce counter;
- removes subscribers who sent subscription requests more than 2 days ago and who have not confirmed the subscription requests;
- removes unsubscribed addresses from the list.

Save

This setting specifies which delivery reports should be saved in the *listname/reports* mailbox in the list owner Account. If you specify *unreadable*, only the messages that the LIST module fails to parse and process are stored in that mailbox.

Warning Text

This service text setting specifies the text of a warning message that is sent to a subscriber when the suspension period ends. In addition to common "symbol combinations", this service text can contain the following combinations:

Combination Substituted with

- | | |
|-----------------|-----------------------------|
| <code>^A</code> | the subscriber address |
| <code>^I</code> | the confirmation identifier |

FEED Mode Distribution

After a message is posted, it is distributed to all users subscribed in the FEED mode.

The `To` header field of a distributed message contains the mailing list address. The `From`, `Date`, and `Message-ID` fields (and specified [additional fields](#)) are copied from the original posting.

The body of the distributed message is a copy of the original message body. If the original message was not in the MIME format, or if it was in the MIME text/plain or multipart/mixed format (the most common formats), the FEED Mode Header text is inserted before and the FEED Mode Trailer text is inserted after the body of the original posting.

Feed Mode Format

Subject Prefix:

Direct Replies:

insert after Reply Prefix

Header

Trailer

Subject Prefix

This setting specifies the string that is inserted into the beginning of the Subject field of all messages distributed in the FEED mode.

When a message is distributed, the system checks the Subject field. If the subject prefix found after the reply prefix (Re:, Re>, etc.), then the subject prefix is deleted.

insert after Reply Prefix

Select this option to insert the Subject Prefix after the Reply prefix (this helps client mailers to group related messages into *discussion threads*).

Sample:

the Subject Prefix setting	[R&D]
a posted message	Subject: test
the distributed message	Subject: [R&D] test
the posted reply	Subject: Re: [R&D] test
the distributed reply	
(insert after Reply Prefix is not selected)	Subject: [R&D] Re: test
the distributed reply	
(insert after Reply Prefix is selected)	Subject: Re: [R&D] test
the composed digest	1) test
	2) Re: test

Direct Replies

If the `to List` option is selected, the Reply-To header field with the E-mail address of the list is added to all distributed messages. As a result, when subscribers answer to distributed messages, their replies are directed to the mailing list by default.

If the `to Sender` option is selected, the Reply-To header is not inserted, and user replies are directed to the From address of a distributed messages, i.e. to the From address of the message author (sender).

Header

This setting specifies the text to be included in the beginning of messages distributed in the FEED mode.

Trailer

This setting specifies the text to be included at the end of messages distributed in the FEED mode.

Digesting and Archiving

Posted messages can be stored in a mailbox created in the list owner Account. Such a mailbox is used to collect messages for message digests. If messages are not removed from that mailbox after a digest is created, this mailbox can be used as a mailing list archive.

Select the Enabled value for the Digesting and Archiving option and follow the link next to that setting to modify the Digesting and Archiving settings.

DIGEST/INDEX Mode Distribution

When the Digesting and Archiving option is enabled, all posted messages are stored in a mailbox created in the list owner Account. The LIST module periodically checks that mailbox and creates list digests and indices.

A digest message contains a set of the messages posted on the mailing list since the time when the previous digest was composed.

A digest message body contains the digest header, the table of contents (TOC) listing all the messages in the digest, the TOC trailer, the posted messages, and the digest trailer.

A list index message contains the same digest header, TOC, and TOC trailer, but it does not contain the posted messages themselves and it does not contain the digest trailer.

List index messages are created at the same time the list digest messages are created. Digest messages are sent to the digest-mode subscribers, and index messages are sent to the index-mode subscribers.

Digest Generator

Generate Every:

or if Larger than:

or if has:

messages

First Digest at:

Generate Every

This setting specifies how often the LIST module should generate digest (and index) messages for this list.

First Digest at

This setting specifies the time of the day when the first digest should be generated.

Note:if the Generate Every setting value is not more than 1 day, every day the first digest is generated at the specified time. If this setting is set to 4:00, the Generate Every setting is set to 1 Day, and the last digest was generated at 23:00 on Monday, the next digest will be generated at 4:00 on Tuesday.

If the the Generate Every setting value is set to N days ($N > 1$), the first digest is generated at the specified time N days later: if this setting is set to 4:00, the Generate Every setting is set to 5 Days, and the last digest was generated at 23:00 on Monday, the next digest will be generated at 4:00 on Friday.

if Larger than

This setting specifies the maximum size of the messages to be included into one digest. If the total size of all messages posted since the last digest was generated exceeds this limit, a new digest (and index) is generated immediately, overriding the Generate Every and First Digest at settings.

if has X messages

This setting specifies the maximum number of messages to be included into one digest. As soon as the specified number of messages is posted, a new digest is generated immediately.

Digest Format

Subject:

Body Format:

Header

Index Line

Index Trailer

Trailer

Digest Subject

This text setting specifies the Subject header field for digest and index messages created for this mailing list.

If the [Prohibit Unmodified Digest Subjects](#) option is selected, the mailing list manager rejects all postings with a reply prefix (Re :, Re>, etc.) followed by an unmodified Digest Subject text.

Body Format

This setting specifies how the digest body should be formatted.

plain text

Digests are composed as plain text messages; individual messages are separated with a line containing minus signs.

standard MIME

Digests are composed in the MIME multipart/digest format, where the first part has the text/plain Content-type and contains the digest TOC, and other parts contain the messages (postings).

embedded MIME

Digests are composed in the MIME multipart/mixed format, where the first part has the text/plain Content-type and contains the digest TOC, and the second part uses the standard multipart/digest format and contains all messages (postings).

Header

This text setting specifies the text to be included into all digest and index messages before the Index Lines (Table of Contents).

Index Line

This setting specifies the format for an index entry. Like the service text settings, the Index Line can (and should) have special symbol combination, but these combinations are different.

For each posted message, values from the message are substituted into the Index Line text and the resulting line is stored in the digest or index message being composed.

Combination Substituted with

- `^X` the message sequence number in the digest being composed
- `^F` the `From` header field of the message
- `^T` the `Date` header field of the message
- `^S` the `Subject` header field of the message
- `^I` the `Message-Id` header of the message

Index Trailer

This text setting specifies the text to be included into all digest and index messages after the Index Lines.

Trailer

This text setting specifies the text to be included after the last message in the digest.

Archiving

All messages posted on a *listname* mailing list are stored in the *listname* mailbox in the list owner Account. When a digest is being composed, the posted messages are retrieved from that mailbox. This mailbox also serves as an archive for all posted messages, and this archive can be searched via the user Web interface.

Each time after a digest is composed, the mailbox is checked and the oldest posted messages are removed to keep the archive mailbox size within the specified limits.

Archiving		
Maximum Archive Size:	Messages to Keep:	messages
Start New Archive every:	Who can Browse:	

Maximum Archive Size

This settings specifies the maximum size of the archive mailbox. After a digest is generated, the

new archive file can be generated or the oldest messages can be removed from the mailbox to keep the mailbox size below the specified limit.

Messages to Keep

This setting specifies the maximum number of messages that can be left in the archive mailbox after a digest is generated.

Start New Archive

This setting specifies when the new archive mailbox should be created. The old archive mailbox becomes a submailbox with the name YYYY-MM-DD, where YYYY specifies the year, MM specifies the month, and DD specifies the day when the first archive message was stored.

Unless the Start New Archive option is set to *never*, a new archive is created when the archive mailbox size limit or the archived message number limit is exceeded.

If the Start New Archive option is set to *never* and the Maximum Archive Size option is set to zero, all messages are removed from the archive mailbox as soon as a digest is generated.

Who can Browse

This setting specifies if this mailing list should appear in the Mailing Lists section of the Server Web Interface.

nobody

The mailing list will not be available via the Web Interface.

anybody

The mailing list will be displayed as a browsable mailing list, and its archive can be used by anybody.

subscribers

The mailing list will be displayed as a browsable mailing list, but in order to browse its archive, users should enter their E-mail address and the Confirmation ID (as the password). To retrieve a forgotten Confirmation ID, a user can always send a message to *<listname-*

confirm@domain> and get the confirmation ID even if the list subscription mode does not require confirmations.

clients

The mailing list will be displayed as a browsable mailing list, and can be browsed from the Internet addresses included into the server [Client IP Addresses](#) list. All users trying to view the list from outside the specified addresses/networks this mode works as the *subscribers* mode and they have to enter the name/password pair (E-mail and Confirmation ID) to browse the list.

Subscribers List

If you are a system administrator or the list owner, you can access the subscribers list page following the [Subscribers](#) link on the Mailing List Settings page.

The Subscribers page contains the list of all E-mail addresses subscribed to the mailing list. For each address additional information (such as the subscriber's real name, number of bounces from this address, etc.) is listed. Each address can be marked, and you can use the Mark All button to mark all list subscribers. You can use the Filter field to display the subscribers with matching addresses only.

first		Filter:			
2 of 289 selected					
E-mail Address	Mode	Subscr	Posts	Bounces	Real Name
andy@vax.stalker.com	null	15:54:51	3		Andy
test@mail.stalker.org	feed	18:18:11	2 mod		Test Account
postings					
mode					

Unsubscribe

Mark some subscribers and click this button to unsubscribe them from the list. Depending on the current FeedBack setting value, the LIST module will either unsubscribe them immediately or just send them confirmation requests. If the FeedBack setting (see below) value is Send Welcome, the Good Bye messages are sent to unsubscribed addresses.

Mark Failed

Mark some subscribers and click this button to tell the LIST module that mail to those addresses bounced. This can be useful in situations when the LIST module fails to process bounce reports automatically, because they come in a non-standard format. Clicking the Mark Failed button will result in the same actions (increased bounce counter, suspension, and warning generation) as caused by receiving a non-fatal bounce from the marked address.

Set Postings

These controls allow you to change the moderation mode for the selected users. You can select some subscribers and set their posting mode to moderated, prohibited, unmoderated, or special. See the [Posting Messages](#) section for more details.

Set Mode

These controls allow you to change the subscription mode for the selected users. See the [Subscription Processing](#) section for more details. If the FeedBack setting (see below) value is set to ask Confirmation, the subscription mode is not changed, but a confirmation request for the mode change operation is sent to the marked subscribers.

Adding Subscribers

You can manually add subscribers to the list. Enter the new subscriber E-mail address press the Subscribe button.

Single User:
Import:

Feedback

Feedback

If this option is set to ask `Confirmation`, all operations performed result in confirmation requests being sent to the specified subscriber address.

If this option is set to `Send Welcome`, confirmation requests are not generated, and the subscription modes are changed immediately, but the `Welcome` and `Good Bye` messages are sent to subscribers when you unsubscribe a user or subscribe a new user.

If this option is set to `silently`, no messages are sent to subscribers.

Single User

Use this field to enter an E-mail address of the user you want to subscribe to this list. A new subscriber address can be specified as an E-mail address with a comment, as `John Smith <johns@company.com>` or `johns@company.com (John Smith)`, in this case the comment is stored the subscriber's real name.

Click the `Subscribe` button to subscribe the specified address.

Importing Subscriber Lists

You can use text files with E-mail addresses to add subscribers to mailing lists. Open the [Subscribers](#) page, and use the `Import` control to select a file with E-mail addresses. Click the `Subscribe` button to add the addresses to the mailing list.

The text file should have one E-mail address per line, with several optional fields on each line. If a line contains several fields, they should be separated with the tabulation (TAB) symbol.

- The first and the only required field is the E-mail address.
- The second field specifies the subscription mode. The first field symbol is checked. The symbols `d` and `D` require the `DIGEST` mode, the symbols `I` and `i` - the `INDEX` mode, and the letters `f` and `F` - the `FEED` mode. All other field symbols are ignored. If the first symbol is not recognized or the field is absent, the new user is subscribed in the Mailing List default mode.
- The last field (if a line contains more than 2 fields) specifies the real name of the new user.

The Mailing List manager checks the file format first. If the file format is incorrect, no new user is subscribed. This allows you to fix the file format and to try the same file: either all addresses are added, or none is added.

Note: The import file must be prepared on the client computer (on the computer you use to run your browser). The browser allows you to upload files from disks connected to that computer, not to the CommuniGate Pro Server computer.

Note: When using Netscape and some other Unix browsers, make sure that the file name ends with the `.txt` suffix - otherwise the browser won't upload the file as a text one, and the file will be ignored.

Note: Some versions of the Netscape® browser for "classic" MacOS® do not convert the MacOS text files (that use the CR symbol as the line separator) into CR-LF delimited text files. You may see the "format error" messages if you try to import a subscriber list from a MacOS computer using that browser. You should either use a different browser, or you should convert the subscriber list into a CR-LF delimited text file before importing it with that browser.

Note: If you are moving users from a different mailing list system, make sure you have set the Feedback option to Silently - otherwise all inserted subscribers will receive confirmation requests and/or Welcome messages.

Subscribing Lists to Lists

You may want to subscribe List1 to List2, so all messages posted on the List2 list are sent to the List1 subscribers, too.

After you have added the qualified address of the List1 (`list1@domain1.dom`) to the List2 subscribers list, add the qualified address of the List2 (`list2@domain2.dom`) to the List1 subscribers list. Set the subscription mode to `null`, so messages will not go back to the List2 list, and set the posting mode to `special`, so messages from List2 (which are auto-generated list messages) will be allowed for posting on List1.

Processing Service Requests

The LIST module processes messages sent to `listserver@localdomainname`. The module takes the List Server commands from the message body, processes those commands, and composes a response message with the command execution results.

All commands sent to the above address apply to the mailing lists in the specified domain only (to the virtual list server in that domain).

The messages with the List Server commands should be in the plain text format, or in the

multipart/alternative format containing a part in the plain text format.

Each List Server command is stored on one text line. Line starting with the %, *, #, and ; symbols are not processed (comment lines).

The following commands are supported:

```
SUBSCRIBE listname [mode [confirmation ID]]
```

```
SUB listname [mode [confirmation ID]]
```

These commands subscribe the message author to the *listname* mailing list. If the *mode* is not specified, the default subscription mode is used.

This is equivalent to sending a message to the *listname-on@localdomainname* address.

```
UNSUBSCRIBE listname [confirmation ID]
```

```
UNSUB listname [confirmation ID]
```

These commands unsubscribe the message author from the *listname* mailing list.

This is equivalent to sending a message to the *listname-off@localdomainname* address.

```
CONFIRM listname
```

```
GETID listname
```

These commands send the message author his/her *listname* subscription ConfirmationID (password).

This is equivalent to sending a message to the *listname-confirm@localdomainname* address.

```
WHICH
```

```
CHECK
```

These commands list the mailing lists (in this *localdomainname* domain) the message author is subscribed to.

```
HELP
```

This command displays the list of supported commands.

```
QUIT
```

```
FINISH
```

These commands tell the LIST module to stop processing. The rest of the message text is ignored.

Routing

The LIST module routes to itself all addresses in the form

listname@domain

if the list *listname* exists in the *domain* domain, or

listname

addresses, if the mailing list *listname* exists in the main domain.

Messages to these addresses are processed as submissions to the specified mailing lists.

The LIST module detects addresses in the form

listname-request@domain and *listname-admin@domain*

or

listname-request and *listname-admin*.

Messages sent to these addresses are rerouted to the mailing list owner.

The LIST module also routes to itself the following addresses:

listname-xxx@domain

and

listname-xxx.

The xxx suffix can be one of the following:

suffix	Action
on, off, subscribe, unsubscribe, feed, digest, index	messages sent to these addresses are processed as subscription requests .
report	messages sent to these addresses are processed as delivery reports about the distributed messages
anything else	messages are rejected



PIPE Module

The PIPE module allows external applications running on the Server computer to submit messages to the CommuniGate Pro Server bypassing TCP/IP connections and Internet protocols, and it allows the Server to deliver messages to external applications.

The PIPE module is also used to submit messages composed with the `mail` and `sendmail` programs that come with the CommuniGate Pro server software. These programs are designed as drop-in substitutions for legacy `mail` and `sendmail` programs.

The Submitted Folder

The CommuniGate Pro PIPE module creates the `Submitted` folder inside the CommuniGate Pro *base folder*.

The PIPE module scans this folder periodically, and processes the files with the `.sub` file name extension. When such a file is found, the module copies it into a *message queue* file and submits that file to the Server kernel for processing.

The `.sub` text files should contain messages in the RFC822 format. The module uses the data in the RFC822 header fields to compose the message envelope.

- The RFC821 "channel format" should NOT be used: submitted files should use the system native "End of Line" character(s), the dot (.) symbol in the first position should not be doubled, and the file should not end with the dot (.) symbol.
- Addresses specified in the `To :`, `Cc :`, `Bcc :` header fields are used to create the message envelope (recipient addresses).
- The `Bcc :` header fields are removed from the submitted message.
- If at least one `Envelope-To :` header field is detected, message envelope (recipient) addresses are formed using this header, and addresses in all remaining `To`, `Cc`, and `Bcc` headers are not placed into the envelope. The `Envelope-To :` header fields are removed from the submitted message.

- If no `Envelope-To:` header field exists, and the `Envelope-Ignore` field or fields exist, the addresses specified in `To/Cc/Bcc` header fields and also listed in the `Envelope-Ignore` fields are NOT included into the message envelope.
- If the `Return-Path:` header field exists, the address specified in that header is used to compose the `Return-Path` envelope address, and this header field is removed from the submitted message.
- If the `Sender:` and/or `From:` header fields contain addresses without the domain part, the Server domain name is added to those addresses.
- If the `Return-Path:` header field does not exist, the address specified in the `From:` or `Sender:` header field is used to compose the envelope `Return-Path` address.
- If the `Envelope-ID:` header field exists, its content is used as the message `Envelope ID`.

If processing of a `.sub` file fails (for example, if the file does not have any recipient address), the module places a record into the System Log, and changes the file extension to `.bad`.

If the `.sub` file is submitted successfully, the file is deleted from the Submitted folder.

Because of the way the PIPE module processes the Submitted folder, it is recommended to compose messages in a different folder and then move the composed `.sub` files to the Submitted folder, or to compose messages in the Submitted folder, in files with the `.tmp` file name extension, and then change the file name extension to `.sub`.

Messages submitted via the PIPE module are marked as "received from a trusted source", so they can be relayed without restrictions.

The Submitted folder is used for [Legacy Mail Emulation](#).

Delivering to External Applications

The PIPE module accepts all messages directed to the `pipe` domain.

The local part of the message address specifies the external application to launch. The part can contain parameters, and can be enclosed into the quotation marks.

Example:

A message directed to the `"execjoe -l store"@pipe` address will be sent to the application `execjoe` started with the `-l store` parameters.

You usually use the PIPE delivery via the [Router](#):

```
<*@somedomain> = exec*@pipe
```

this Router record will direct messages sent to the joe@somedomain address to the execjoe application.

```
<*@somedomain> = "execall\ -u\ *"@pipe
```

this Router record will direct messages sent to the joe@somedomain address to the execall application started with the -u joe parameters.

To limit the set of applications that can be started via the PIPE module, the *external application directory* is specified as one of the PIPE module settings. The application names specified in message addresses can not include the slash (/) or the backslash (\\) symbols, and they cannot start with the dot (.) symbol, and it specified the name of the application (program) file in the *external application directory*.

The message text (including the message headers and the message body) is passed to the external application as its *standard input*.

Note: the application must read the entire *stdin* data stream, otherwise message processing fails.

The PIPE module discards the external application *standard output*.

Serialized Delivery

In order to allow several PIPE processors to deliver messages simultaneously, the PIPE module creates a separate queue for each message it has to deliver. If you want to serialize processing, you can use the following form of the PIPE address:

```
"queue[name] application parameters"@pipe
```

All messages directed to these addresses will be placed into the *name* queue, and a single PIPE processor will send the enqueued messages to the application(s) specified in those addresses. You can use any alphanumeric string as a queue *name*, and you can specify as many queues as you need.

The following [Router](#) records can be used to maintain two serialized PIPE queues (the PROC1 and ARCH queues):

```
<incoming> = "queue[PROC1] procin -mark"@pipe
```

```
<control> = "queue[PROC1] procin1 -control"@pipe
```

```
<archiver> = "queue[ARCH] appendfile /var/archive"@pipe
```

All messages sent to the <incoming@maindomain.com> and <control@maindomain.com> will be processed one-by-one using one PIPE processor.

For PIPE addresses that do not have the `queue[name]` prefix, the PIPE module creates separate queues with numeric names.

Command Tags

The message text (the header and the body) is sent to the task as that task *standard input* (*stdin*).

If the external application has written any text to its *standard error* channel, the message delivery fails, and the standard error text is sent to the message sender.

An application name can be prefixed with the [FILE] tag:

```
[FILE] application parameters
```

When this prefix is used, the application standard input will be empty (closed), and the string

```
-f Queue/fileid.msg
```

(the -f flag and the Message file name, relative to the *base directory*) will be appended to the end of the application parameters:

```
-f Queue/12002345.msg
```

An application name can be prefixed with the [RETPATH] tag:

```
[RETPATH] application parameters
```

When this prefix is specified, the string "-p" followed by the message return-path address is added to the end of the application parameters:

```
-p address@domain.com
```

An application name can be prefixed with the [RCPT] tag:

```
[RCPT]application parameters
```

When this prefix is specified the string "-r" followed by the original recipient address is added to the end of the application parameters:

```
-r address1@domain1.com
```

An application name can be prefixed with the [STDERR] tag (see below).

An application name can have several prefix strings, and they can be specified in any order. If several of [FILE], [RETPATH], and [RCPT] prefix strings are specified, the -f flag and its parameter are added first, followed with the -p flag and its parameter, followed with the -r flag and its parameter.

If the [STDERR] prefix is specified and the external application completes sending some data to its *standard error* channel, the *standard error* data is used to compose the error report text.

Configuring the PIPE module

Use a Web browser to open the PIPE page in the Settings realm of the WebAdmin Interface.

Log:

Log

Use the Log setting to specify what kind of information the PIPE module should put in the Server Log. Usually you should use the Major (message transfer reports) or Problems (message transfer and non-fatal errors) levels. But when you experience problems with the PIPE module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly.

The PIPE module records in the System Log are marked with the PIPE tag.

Submitting
Check Submitted Directory Every:

Check Submitted Directory

This option specifies how often the PIPE module should scan the Submitted directory and deliver the .sub files stored there.

Delivering	Processes:
Application Directory:	
Processing Time Limit:	

Processes

This option specifies the number of threads used to deliver messages. If some of your external applications are slow, you may want to use several PIPE delivery threads, so several messages can be processed at the same time.

Application Directory

This option specifies the directory with the applications the PIPE module can launch. If an empty string is specified for this option, all messages directed to the PIPE module are rejected.

Processing Time Limit

This option limits the time an external application uses to process a message. If an external

application does not complete within the specified period of time, the application process is interrupted and the message is rejected.

Foreign Queue Processing

In emergency situations you may need to process an additional Queue directory without stopping your server. These situations include a server hardware failure when the rescued Queue files should be processed with some other, already running server.

To process an additional Queue directory, move it to the *base directory* of a running server as the ForeignQueue directory. If you prefer to use symbolic links, make sure that the Queue and ForeignQueue directories are created on the same file system.

Every 3 minutes the PIPE module checks if the ForeignQueue directory exists in the server *base directory*. If the ForeignQueue directory is found, the module reads it, and creates a special processing thread. That processor moves all .msg files from the ForeignQueue to the Queue directory (it can rename the files in the process), The processor deletes all found .tmp files. Files with other extensions are left in the ForeignQueue directory.

All moved .msg files are submitted to the Server kernel, into the [ENQUEUER](#) queue, and the Server starts to process them in the same way it processes all submitted messages.

After the processing thread has completed, the ForeignQueue directory can be removed from the *base directory*.



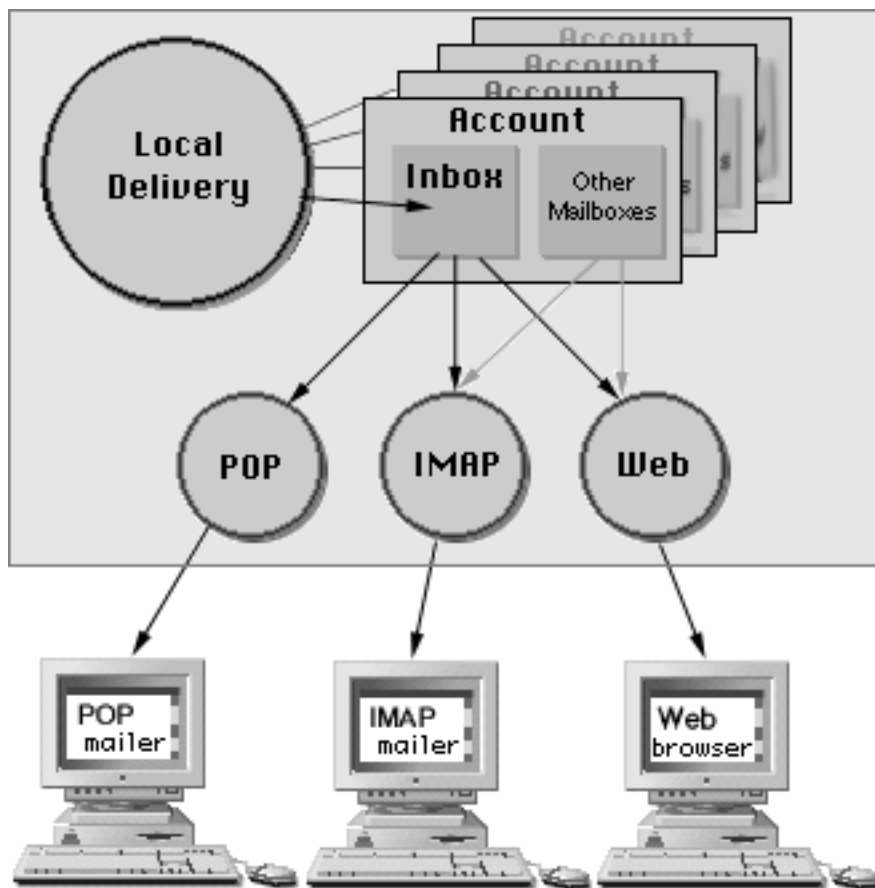
Access to CommuniGate Pro Accounts

The CommuniGate Pro Server allows users to use various mailer applications to access their accounts and mailboxes.

- The [POP module](#) is a POP3 server that allows users to retrieve mail from their INBOX mailboxes using the POP-based mailers.
- The [IMAP module](#) is an IMAP4rev1 server that allows users to process mail messages in all account mailboxes using IMAP-based mailers.
- The [WebMail module](#) is an HTTP (Web) server that allows users to process mail messages in all account mailboxes using any Web browser.
- The [MAPI module](#) allows users to access their accounts and mailboxes the Microsoft® Windows MAPI (Mail API) and use Microsoft Outlook in the "groupware" mode.
- The [FTP module](#) is an FTP server that provides access to [Personal Web Sites](#).
- The [LDAP module](#) is an LDAP server that provides access to various directories and databases.
- The [ACAP module](#) is an ACAP server that allows users to manage their Accounts.
- The [PWD module](#) is a poppwd server that allows users to change the account passwords using certain POP and IMAP mailers.

Access to Accounts

Every CommuniGate Pro Account can be accessed via Access modules - POP, IMAP, Web Email, etc. Several client applications can use the same CommuniGate Pro Account at the same time, via the same, or different access modules.



Any mailbox in any CommuniGate Pro Account can be [shared](#): an Account mailbox can be accessed not only by the Account owner, but by other Account users - if the Account owner or an administrator grants those users access rights for that mailbox.

Serving Multiple Domains

The main problem of serving multiple domains on one server is to provide access to accounts in several domains. In order to do this, the server should get the name of the domain name in which to look for the specified account. As for mail delivery, the server needs the "full account name" i.e. an address in the form *accountname@domainname*.

There are several methods to pass the domain name to the server:

- A client application explicitly specifies the domain name.
 - If a user accesses the server via the HTTP (Web interface), this happens automatically: the user first specifies the server URL (`http://domainname:port`), and then enters the account name in the Login form.
Since all modern browsers pass the original URL to the server, the domain name

becomes known, and the HTTP module immediately appends that domain name to a simple user name specified in the Login form.

○ If a user accesses the server via a POP or IMAP mailer, it is possible to specify the full account name in the mailer "account name" settings. Since many mailers do not like to see the @ symbol in the account name, the % symbol can be used instead. The user john that has an account in the secondary domain client1.com should specify the account name as john%client1.com, not just as john.

- The domain name can be detected using multihoming. If it is impossible to force users to access the server via the Web interface or to make them enter full account names in their POP/IMAP mailers, multihoming can be used.

A server is using multihoming if the server computer has more than one Internet (IP) address. Using the Domain Name System (DNS) the secondary domains can be assigned different IP addresses.

If a secondary domain has a dedicated IP address assigned to that domain, and a user tries to connect to the server via that IP address, all simple account names specified with the user mailer are processed as account names in that domain.

Additional IP addresses can be rather expensive, so this method should be used only if it is impossible to make users specify domain names explicitly.

All methods can be mixed in one server: a limited number of domains can be served using dedicated additional IP addresses, while other domains are served using explicit domain name specifications.

Multihoming

Every access session begins with the authentication procedure: a mailer application passes a user (Account) name and a password.

The CommuniGate Pro Server tries to detect which domain it should use to look for the specified Account name.

- If the specified name contains the @ symbol or the % symbol, the Server assumes that the user has specified a "full account name", i.e. an Account name with its Domain name: username@domainname or username%domainname (see above).
- If the specified name does not contain the @ symbol or the % symbol, the Server looks at the IP address on which it has received this connection. Systems with multihoming (i.e. systems that have several local IP addresses) may have certain IP addresses [dedicated to some secondary domains](#). If a connection IP address is dedicated to a secondary domain, that domain name is appended to the account name to get the full account name. If the address is dedicated to the main domain, the specified name is processed as an account in the main

domain.

Sample:

The server computer has 2 IP addresses: 192.0.0.1 and 192.0.0.2.

The server main domain is `company.com`, and the secondary domains are `client1.com` and `client2.com`.

The DNS A-records for `company.com` is pointing to the IP address 192.0.0.1, the A-record for the `client1.com` points to a dedicated IP address 192.0.0.2, while the A-records for the `client2.com` domain point to the same "main" IP address 192.0.0.1.

Each domain has an account `info`.

Three users configure their POP and IMAP mailers to access an account `info`, but they specify different names in their "mail server" settings: the first user specifies `company.com`, the second - `client1.com`, and the third user specifies `client2.com`.

When the first user starts her mailer:

- The mailer takes the specified "mail server" setting `company.com`, and it uses the Domain Name System A-records to resolve (convert) that name to the IP address 192.0.0.1.
- The mailer establishes a connection with that address (which is one of 2 addresses of the server computer), and it passes the user name `info`.
- The server detects a simple user name `info` and detects that this connection is established via the server address 192.0.0.1.
- The server detects that the main domain name points to that IP address, so it adds the main domain name `company.com` to the specified simple name.
- The server gets the correct full account name `info@company.com`.

When the second user starts checking mail:

- The mailer takes the specified "mail server" setting `client1.com`, and it uses the Domain Name System A-records to resolve (convert) that name to the IP address 192.0.0.2.
- The mailer establishes a connection with that address (which is one of 2 addresses of the server computer), and it passes the user name `info`.
- The server detects a simple user name `info` and detects that this connection is established via the server address 192.0.0.2.
- The server detects that the `client2.com` secondary domain name points to that IP address, so it adds the secondary domain name `client1.com` to the specified simple name.
- The server gets the correct full account name `info@client1.com`.

When the second user starts checking mail:

- The mailer takes the specified "mail server" setting `client2.com`, and it uses the Domain Name System A-records to resolve (convert) that name to the IP address 192.0.0.1.
- The mailer establishes a connection with that address (which is one of 2 addresses

of the server computer), and it passes the user name `info`.

- The server detects a simple user name `info` and detects that this connection is established via the server address `192.0.0.1`.
- The server detects that the main domain name points to that IP address, so it adds the main domain name `company.com` to the specified simple name. The `client2.com` domain name also points to that IP address, but the server dedicates IP addresses to only one domain, and it always assigns it to the first domain it processes, the main domain `company.com` in our case.
- The server gets the **incorrect** full account name `info@company.com`.

This happens because the mailer has not passed the information about the "mail server" name from its settings, and the only information the server has is the IP address. But since the IP address is the same for both main domain `company.com` and the secondary domain `client2.com`, the server is unable to detect which domain is needed and defaults to the main domain.

In order to solve this problem, the third user should specify the account name as `info%client2.com`, not just `info`. In this case, when this users starts the mailer:

- The mailer takes the specified "mail server" setting `client2.com`, and it uses the Domain Name System A-records to resolve (convert) that name to the IP address `192.0.0.1`.
- The mailer establishes a connection with that address (which is one of 2 addresses of the server computer), and it passes the user name `info%client2.com`.
- The server detects a full user name `info%client2.com` and it does not look at the IP addresses. It just converts the `%` symbol into the `@` symbol.
- The server gets the correct full account name `info@company.com`.

This problem does not appear if the third user uses the Web Interface: the server always gets the addressed domain name via the HTTP protocol, so it does not have to detect that name by looking at the IP addresses.

Note: FTP clients work in the same way as the POP/IMAP mailers do, so FTP users are required to supply qualified Account names unless they connect to an IP Address assigned to their Domain.

Note: the MAPI Connector always sends a qualified Account Name: if users specify names without the `@` or `%` signs, the Connector adds the `'@'` sign and `Server Name` setting value to the specified account name

Routing

When the full account name is composed, this name (address) is passed to the [Router](#).

- If the Router reports an error, the client is not authenticated and an error message is returned to the client application. An error is usually the Unknown user error, but it can be any error that Router or modules can generate when routing an address.
- If the Router has successfully routed the address, but the address is not routed to the [Local Delivery](#) module, the client is not authenticated and an error message is returned to the client application. This happens if a user has specified a mailing list name, not an account name, or if the specified name is rerouted to some other host via SMTP or UUCP.
- If the Router routed the address to some account in some local domain, that account is opened, and the account passwords are checked.

This means that all routing applied to E-mail addresses is also applied to the account names specified with mailer applications.

Sample:

The account John has an alias John_Smith;
all E-mail messages addressed to John_Smith will be stored in the account John;
the user can specify either John or John_Smith as the "account name" setting in his mailer - in both cases the account John will be opened when his mailer starts a session.

Sample:

The account John has been moved from the main domain company.com to the domain client1.com, and it was renamed in j.smith. The administrator has created an alias record with the Router:
<John> = j.smith@client1.com;
all E-mail messages addressed to John@company.com will be stored in the account j.smith in the secondary domain client1.com;
the user can still specify just John as the "account name" setting, and the same company.com "mail server" setting in his mailer - but the server will open the account j.smith in the client1.com domain.

Note:

do not create an alias record that redirects the Postmaster account in the main domain. You will not be able to administer the server, if the postmaster account is redirected to a nonexistent Account or to an Account that does not have the postmaster access rights.

If you want the postmaster mail to be directed to some other user, do not use the Router, but use the postmaster account [Rules](#) instead.



Mailbox Sharing

The word "shared" is used to describe several different features a messaging server can provide. Because of that, this CommuniGate Pro Guide uses the following terms:

- *Simultaneous Access* - a situation when several client applications work with the same mailbox at the same time.
- *Foreign Mailbox Access* - a situation when an Account user works with a mailbox created in a different Account.
- *External Mailbox* - a CommuniGate Pro mailbox located outside the CommuniGate Pro *base directory* and modified directly by other programs that do not use messaging protocols and bypass the CommuniGate Pro Server.

Simultaneous Access

The CommuniGate Pro Server allows several client applications to connect, open the same mailbox, and read and modify the mailbox data at the same time.

The CommuniGate Pro *multithreaded* design allows the Server to synchronize client activities without using OS-level *file locks* and it does not require a client to wait till all other clients close the mailbox.

Simultaneous Access means that:

- several clients (POP, IMAP, Web, etc.) can have simultaneous access to the same mailbox;
- new messages can be added to a mailbox while mailer clients are working with that mailbox;
- messages can be deleted from a mailbox while mailer clients are working with that mailbox.

Clients accessing the same mailbox can use the same or different mailbox access protocols - [POP](#), [IMAP](#), or [WebUser](#) Interface.

Simultaneous Access is supported for all [Mailbox types](#) implemented in the CommuniGate Pro

software.

This feature allows you to work with your mailbox from several workstations, and it lets a group of people (i.e. the sales department) process messages in one centralized mailbox.

Foreign and Public Mailboxes

The CommuniGate Pro access system allows an account user to access mailboxes in other accounts. Access to these foreign mailboxes (also called shared mailboxes) is controlled via the mailbox [Access Control Lists](#).

To access a mailbox in a different account, the mailbox name should be specified as `~accountname/mailboxname`. For example, to access the INBOX mailbox in the Boss account, the mailbox name should be specified as `~Boss/INBOX`,

If there are several local domains on the Server, mailboxes in a different domain can be accessed by specifying full account names. To access the LIST/reports mailbox in the account ListMaster in the client.com domain, the mailbox name should be specified as `~ListMaster@client.com/LIST/reports`.

Account names specified after the "~" sign are processed with the [Router](#), so account alias names can be used instead of the real account names, and all Routing Table rules are applied.

Very often Foreign mailboxes are used:

- to let a secretary view and mark messages in your INBOX;
- to let several sales persons see and process a single "sales maildrop" - the INBOX of the sales account;
- to let several engineers see and process a single "technical support maildrop" - the INBOX of the support account.

CommuniGate Pro can provide "public" mailboxes, too. This can be done by creating an account `public`, and assigning public Access rights to its mailboxes. Usually, each group of public mailboxes is managed by some administrator, who is not required to be a CommuniGate Pro administrator.

A CommuniGate Pro Server administrator should create the `public` Account, log into that Account using the Web User Interface or a decent IMAP client, create some public mailboxes, and

grant administration rights to regular users that will administer these public mailboxes. Those users will then grant access rights to other users, create submailboxes, and perform other administrative tasks.

For example, a public mailbox administrator can use [Automated Rules](#) to copy certain incoming messages directly into some public mailbox.

Some IMAP clients (such as Microsoft Outlook and Outlook Express) do not support foreign mailboxes at all. To let those clients access shared mailboxes in other Accounts, [Mailbox Aliases](#) can be used.

External Mailboxes

On some systems users have direct (login) access to the mail server computer, and some of them get used to *Local Mailers* - mail, elm, and others. *Local Mailers* do not use any network protocol to access account mailboxes. Instead, those programs read and modify mailbox files directly, via the file system.

The CommuniGate Pro allows you to create accounts with *external* INBOX mailboxes. These mailboxes are stored not inside the CommuniGate *base directory*, but in the system directory known to the legacy mailer applications.

Since these INBOX files can be read and modified directly, bypassing the CommuniGate Pro protocols and modules, the Server needs to synchronize its activity with legacy mail applications using OS *file locking* features - either FileLevel locks or FileRange locks.

On Unix systems the FileLevel locks are known as `flock` operations, and RangeLevel locks are known as `fcntl` operations. Check with your OS manual to see the which method the legacy mailers use on your system, and configure the CommuniGate Pro Server to use that method. For systems that support only one file locking mechanism (MS Windows, Sun Solaris, and some other systems), selecting either method selects that mechanism.

You should use external mailboxes only when absolutely necessary, because:

- access to external mailboxes is less effective because of the resources needed for OS *file locking*.
- *local mailer* applications do NOT synchronize correctly, and **there is always a chance that a local mailer destroys a mailbox**. If some of your users have to work via local mailers, warn them about this fact, and ask them to avoid using *local mailers* and modern (protocol-based) mailers at the same time. These bugs in most local mailers do not show up if

messages are only added to the mailbox when a local mailer is active. Local mailers may corrupt mailboxes if messages have been deleted from a mailbox during the time a local mailer was active.

The bugs in the local mailers *have nothing to do with the CommuniGate Pro Server*, and they can result in mailbox corruption on any system. You may check for yourself - [sample shell sessions](#) show you how the Unix `mail` programs can destroy your mailbox.

If you have to support Local Mailer compatibility for all or some accounts in a domain (usually - in the main domain), you should specify the [External Mailboxes settings](#) for that domain.

When you create an account that has an external INBOX, the Server checks if the account INBOX file already exists in the specified location and creates one if the mailbox file is absent.

When you delete an account that has an external INBOX, the Server does NOT remove the INBOX mailbox file.



CommuniGate Pro POP Module

The CommuniGate Pro POP module implements a POP3 server. POP3 servers allow client applications (mailers) to retrieve messages from account mailboxes using the POP3 Internet protocol (STD0053, RFC1939, RFC1734, RFC1725) via TCP/IP networks.

The CommuniGate Pro POP module implements several protocol extensions, including the XTND XMIT extension. Some mailers can employ this feature to submit messages to the CommuniGate Pro server.

The POP module activity statistics is available via the CommuniGate Pro [SNMP](#) agent.

Post Office Protocol (POP3)

The Post Office Protocol allows computers to retrieve messages from mailboxes on mail servers. A computer running a mailer (mail client) application connects to the mail server computer and provides account (user) name and the password. If access to the specified user account is granted, the mail application sends protocol commands to the mail server. These protocol commands tell the server to list all messages in the mailbox, to retrieve certain messages, or to delete them. When a server receives a request to retrieve a message, it sends the entire message to the mail client. The mail client may choose to retrieve only the first part of the message.

The POP3 protocol does not support multi-mailbox accounts. If a client application specifies a multi-mailbox (folder) account, the INBOX mailbox is opened.

When the client application sends a request to delete a message from the mailbox, the message is not deleted immediately, but it is marked by the server. Only when the client application ends the session properly and closes the connection, the marked messages are then removed.

The POP module supports the XTND XMIT extension of the POP protocol. This extension allows users to submit messages via the POP protocol instead of the SMTP protocol.

Configuring the POP module

Use a Web browser to configure the POP module. Open the Access page in the WebAdmin Settings section (realm):

Serving POP Clients	
Log:	
Channels:	<u>listener</u>

Use the Log Level setting to specify what kind of information the POP module should put in the Server Log. Usually you should use the Major (message transfer reports) or Problems (message transfer and non-fatal errors) levels. But when you experience problems with the POP module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly.

The POP module records in the System Log are marked with the POP tag.

When you specify a non-zero value for the Maximum Number of Channels setting, the POP module creates a so-called "listener". The module starts to accept all POP connections that mail clients establish in order to retrieve mail from your server. The setting is used to limit the number of simultaneous connections the POP module can accept. If there are too many incoming connections open, the module will reject new connections, and the mail client should retry later.

By default, the POP module Listener accepts clear text connections on the TCP port 110. The standard TCP port number for secure POP connections is 995, but it is not enabled by default. Follow the [listener](#) link to tune the POP [Listener](#).

The POP module supports the STARTTLS command that allows client mailers to establish a connection in the clear text mode and then turn it into a secure connection.

User Authentication

The POP module allows users to employ all [authentication methods](#) supported with the CommuniGate Pro Server, as well as the APOP method.

Secure (encrypted) Access

The POP module can be used to accept SSL/TLS (encrypted) connections from user mailers (see the Listener configuration note above). Additionally, the POP module supports the STLS command that allows client mailers to establish plain text, unencrypted connections (using the regular TCP port 110), and then start encrypted communications on those connections.

Special Features

Unlike many other POP servers, the CommuniGate Pro POP module does not "lock" the mailbox it opens on a mail clients behalf. The open mailbox can be used by other client applications at the same time. See the [Sharing](#) section for the details.

Since the POP3 protocol was not designed to support these features, the CommuniGate Pro POP module:

- shows only the messages that existed in the mailbox when the mailbox was opened with the client mailer; all new messages received during this session will be seen by the POP client only when it connects to the mailbox again;
- keeps *zombies* for the messages deleted during the current session; the module shows them as messages of a zero size, and the module reports an error when a client application tries to retrieve a deleted message;

When a client mailer retrieves a message with the RETR command, the message is marked with the "Seen" flag (this change is noticed when using an IMAP client with the same mailbox). The TOP command that allows a client POP mailer to retrieve only the first part of the message does not set the Seen flag.

In order to support Microsoft® E-mail clients, the POP module supports the non-standard "empty AUTH" command (the AUTH command without parameters), returning the list of supported [SASL methods](#).

The XTND XMIT Extension

The CommuniGate Pro POP module implements the XTND XMIT protocol extension. Mailer applications that support this extension (like Eudora®) can submit messages to the Server via a POP connection.

This feature can be useful for mobile users that would be otherwise unable to send their messages via CommuniGate Pro SMTP due to the Server anti-spam protection. Submitting messages via POP can be more convenient than using the ["address-remembering" scheme](#), since this method does not have time restrictions.

Notification Alerts

The POP3 protocol does not provide any method to send a notification alert to the client mailer. If an account has any pending [alert message](#), the CommuniGate Pro POP module simply rejects the connection request after the user is authenticated. The returned error code contains the alert message text:

```
ALERT: alert message text
```

When the user repeats a connection attempt to the same account, the next pending alert message is returned as an error - till all alert messages are sent to that user.

Accessing Additional Mailboxes

Unlike the [IMAP](#) protocol, the POP3 protocol was designed to access only one account mailbox - the INBOX mailbox.

The POP module allows users to access any account mailbox by specifying the mailbox name as a part of the account name. To access the mailbox *mailboxname* in the *accountname* account, a user should specify the account name as: *mailboxname#accountname*:

Account name	Accessed Mailbox
(specified in the mailer settings)	
jsmith	mailbox INBOX in the jsmith account
private#jsmith	mailbox private in the jsmith account
lists/info#jsmith@client1.com	mailbox lists/info in the jsmith account in the client1.com domain

The POP module allows a user to access any mailbox in any other account (a *foreign* or *shared* mailbox), as well as public mailboxes. See the [Sharing](#) section for the details.

If a user can log into the *accountname* account and wants to access the mailbox *mailboxname* in the *otheraccount* account, that user should specify the account name as:

~otheraccount/mailboxname#accountname:

Account name	Accessed Mailbox
(specified in the mailer settings)	
jsmith	mailbox INBOX in the jsmith account
~public/announces#jsmith	the public mailbox announces
~boss/INBOX#jsmith	mailbox INBOX in the boss account

In all samples above, the user is authenticated as *jsmith*, using the *jsmith* account password.

If the authenticated user does not have a right to delete messages in the selected mailbox, the DELE protocol operations fail and an error code is returned to the user mailer.

The POP module can also use the [Direct Mailbox Addressing](#) feature to open additional mailboxes.

Accessing Individual Mail in a Unified Account

The POP module implements [Unified Domain-Wide Account](#) filtering. As all [access modules](#), the POP module uses the [Router](#) to process the specified username.

If a client mailer specifies the *abcdef@client1.com* username (as used in the [example](#)), the Router routes this address to the Local account *CL1*, and it returns *abcdef* as the *local part* of the resulting address.

The POP module checks the local part returned by the Router, and if this string is not empty, it performs filtering on the open mailbox: the module hides all mailbox messages that do not have the *X-Real-To* header field (or other field specified in the Local Delivery module settings), or do not have the specified string (individual name) listed in that header field.

So, if the user has specified the *abcdef@client1.com* username, only the messages originally routed to that particular address will be shown in the *CL1* account mailbox.

If a user connects as C11, the same account mailbox will be opened, but since the local part string will be empty in this case, all mailbox messages will be shown.

Example:

The Router line:

client1.com = C11.local

The first message is sent to:

abcdef@client1.com

It is stored in the C11 account INBOX with unique ID 101, and a header field is added:

X-Real-To: abcdef

The next message is sent to:

xyz@client1.com

It is stored in the C11 account INBOX with unique ID 102, and a header field is added:

X-Real-To: xyz

After these 2 messages are stored, POP sessions will show different *views* depending on the user name specified:

S: +OK CommuniGate Pro POP Server is ready

C: USER C11

S: +OK, send pass

C: PASS mypassword

S: +OK 2 message(s)

C: UIDL

S: +OK

S: 1 101

S: 2 102

S: .

C: QUIT

S: +OK bye-bye

S: +OK CommuniGate Pro POP Server is ready

C: USER abcdef@client1.com

S: +OK, send pass

C: PASS mypassword

S: +OK 1 message(s)

C: UIDL

S: +OK

S: 1 101

S: .

C: QUIT

S: +OK bye-bye

S: +OK CommuniGate Pro POP Server is ready

```
C: USER xyz@client1.com
S: +OK, send pass
C: PASS mypassword
S: +OK 1 message(s)
C: UIDL
S: +OK
S: 1 102
S: .
C: QUIT
S: +OK bye-bye

S: +OK CommuniGate Pro POP Server is ready
C: USER blahblah@client1.com
S: +OK, send pass
C: PASS mypassword
S: +OK 0 message(s)
C: UIDL
S: +OK
S: .
C: QUIT
S: +OK bye-bye
```




CommuniGate Pro IMAP

Module

The CommuniGate Pro IMAP module implements an IMAP server. IMAP servers allow client applications (mailers) to retrieve messages from account mailboxes using the IMAP4rev1 Internet protocol (RFC2060) via TCP/IP networks.

The IMAP protocol allows client applications to create additional account mailboxes, to move messages between mailboxes, to mark messages in mailboxes, to search mailboxes, to retrieve MIME structure of stored messages, and to retrieve individual MIME components of messages stored in account mailboxes.

The CommuniGate Pro IMAP module supports both *clear text* and *secure (SSL/TLS)* connections.

Internet Message Access Protocol (IMAP)

The Internet Message Access Protocol allows computers to work with messages stored in mailboxes on mail servers. A computer running a mailer (mail client) application connects to the mail server computer and provides account (user) name and the password. If access to the specified user account is granted, the mail application sends protocol commands to the mail server. These protocol commands tell the server to list all messages in the mailbox, to retrieve certain messages, to delete messages, to search for messages with the certain attributes, to move messages between mailboxes, etc.

CommuniGate Pro IMAP supports [various Internet standards](#) (RFCs) and has many [additional unique features](#).

Configuring the IMAP module

Use a Web browser to configure the IMAP module. Open the Access page in the WebAdmin

Settings section (realm):

Serving IMAP Clients	
Log:	
Channels:	<u>listener</u>

Use the Log setting to specify the type of information the IMAP module should put in the Server Log. Usually you should use the Major (message transfer reports) or Problems (message transfer and non-fatal errors) levels. But when you experience problems with the IMAP module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well. When the problem is solved, set the Log Level setting to its regular value, otherwise your System Log files will grow in size very quickly.

The IMAP module records in the System Log are marked with the IMAP tag.

When you specify a non-zero value for the Maximum Number of Channels setting, the IMAP module creates a so-called "Listener". The module starts to accept all IMAP connections that mail clients establish in order to retrieve mail from your server. The setting is used to limit the number of simultaneous connections the IMAP module can accept. If there are too many incoming connections open, the module will reject new connections, and the mail client should retry later.

By default, the IMAP module Listener accepts clear text connections on the TCP port 143, and secure connections - on the TCP port 993. Follow the listener link to tune the IMAP [Listener](#).

The IMAP module supports the STARTTLS command that allows client mailers to establish a connection in the clear text mode and then turn it into a secure connection.

MultiAccess

While many other IMAP servers "lock" opened mailboxes, the CommuniGate Pro IMAP is designed to provide simultaneous access to any mailbox for any number of clients.

In reality, the IMAP module uses the CommuniGate Pro Mailbox Manager that provides this feature for all types of clients. See the [Sharing](#) section for the details.

Access Control Lists

The IMAP module supports [RFC2086 \(IMAP4 ACL extension\)](#). This protocol extension allows IMAP users to grant access to their mailboxes to other users.

See the [Mailboxes](#) section for the detailed description of mailbox ACLs.

In order to set Access Rights, a client should use a decent IMAP client that supports the ACL protocol extension. If such a client is not available, mailbox access rights can be set using the [WebUser](#) Interface.

Foreign (Shared) and Public Mailboxes

CommuniGate Pro allows account users to access mailboxes in other accounts. See the [Sharing](#) section for the details.

Many popular IMAP clients do not support foreign mailboxes. There is a workaround for IMAP mailers that use the "subscription" scheme. Subscription is a list of mailbox names that the mailer keeps on the server (in the account settings). Usually, mailers build the subscription list when you configure them for the first time. Later, they show only the mailboxes included into the subscription list.

By using a different IMAP client or the Web User Interface, a user can add a foreign mailbox name (such as `~sales/processed` or `~public/news/company`) to the subscription list. This will make the old IMAP client show the foreign mailbox along with the regular account mailboxes, and the user will be able to work with that foreign mailbox.

Some IMAP clients (such as Microsoft Outlook and Outlook Express) do not support foreign mailboxes at all. To let those clients access shared mailboxes in other Accounts, [Mailbox Aliases](#) can be used.

User Authentication

The IMAP module allows users to employ all [authentication methods](#) supported with the CommuniGate Pro Server.

Notification Alerts

The CommuniGate Pro IMAP module checks for any pending [alert message](#) sent to the authenticated Account. The alert messages are transferred to the client mailer using the standard IMAP [ALERT] response code.

The CommuniGate Pro IMAP module checks for alert messages right after the user is authenticated, and it can detect and send alert messages at any time during an IMAP session.

Login Referrals

The IMAP module supports [RFC2221 \(Login Referrals\)](#). As explained in the [Access section](#) all user addresses provided with mail clients are processed with the [Router](#). If the specified user name is routed to an external Internet address (handled with the SMTP module) the IMAP module returns a negative response and provides a login referral. If an IMAP client supports login referrals, it will automatically switch to the new address.

Sample:

A user account `j.smith` has been moved from your server to the account `John` at the `othercompany.com` server. In order to reroute the user mail you have created an alias record in the Router:

```
<j.smith> = John@othercompany.com
```

Now, when this user tries to connect to his old `j.smith` account on your server, the server rejects the user name, but provides a login referral:

```
1234 NO [REFERRAL IMAP://John;AUTH=*@othercompany.com/] account  
has been moved to a remote system
```

If the mail client supports login referrals, it will automatically try to connect to the server `othercompany.com` as the user `John`.

Monitoring IMAP Activity

You can monitor the IMAP module activity using the WebAdmin Interface.

Click the Access link in the Monitors realm to open the IMAP Monitoring page:

IMAP Server					<input checked="" type="checkbox"/> Channels:3
ID	Address	Account	Connected	Status	Running
9786	[216.200.213.116]	user1@domain2.dom	3min	selecting a mailbox	2sec
9794	[216.200.213.115]	user2@domain1.dom	34sec	Connected to mailbox	
9803	[216.200.213.115]		2sec	Authenticating	

ID

This field contains the IMAP numeric session ID. In the CommuniGate Pro Log, this session records are marked with the IMAP-*nnnnn* flag, where *nnnnn* is the session ID.

Address

This field contains the IP address the client has connected from.

Address

This field contains the name of the client Account (after successful authentication).

Connected

This field contains the connection time (time since the client opened this TCP/IP session).

Status

This field contains either the name of the operation in progress or, if there is not pending operation, the current session status (Authenticating, Selected, etc.).

Running

If there is an IMAP operation in progress, this field contains the time since operation started.

The IMAP activity statistics is available via the CommuniGate Pro [SNMP](#) agent.

Additional IMAP Extensions

The CommuniGate Pro IMAP module provides several protocol extensions that are not part of the IMAP standard and are not included into the existing IMAP Extension standards.

UNSELECT

This IMAP command is equivalent to the CLOSE command, but it does not expunge any message marked as \Deleted

Additional STATUS data items.

The STATUS command can use the following additional data item names:

INTERNALSIZE

The data item included into the response is a number. This number specifies the size of the mailbox as it is stored on the server. This size is close to, but not exactly the same as the summ of the RFC822.SIZE attributes of all messages stored in the mailbox.

OLDEST

The data item included into the response is a `date_time` string. It specifies the INTERNALDATE of the oldest message stored in the mailbox. If the mailbox has no messages, this data item is not included into the response.

Example:

```
A001 STATUS mailbox (UNSEEN OLDEST INTERNALSIZE)
* STATUS mailbox (UNSEEN 14 OLDEST "23-Feb-2002 07:59:42
+0000" INTERNALSIZE 2345678)
A001 OK completed
```

Additional LIST extensions.

The LIST command can use additional options along with the options specified in the LISTEXT extension standard:

UIDVALIDITY, MESSAGES, UIDNEXT, UNSEEN, INTERNALSIZE, OLDEST

The data items included into the response have the following format:

`\option_name(option_value)`

Example:

```
A001 LIST (CHILDREN UNSEEN INTERNALSIZE) " " "ma%"
* LIST (\HasNoChildren \UNSEEN(14) \INTERNALSIZE(2345678)
\Unmarked) mailbox
A001 OK completed
```



CommuniGate Pro Web User Interface

The CommuniGate Pro Server provides Web (HTTP/HTML) access to user Accounts. The WebUser component works via the [HTTP module](#) and allows users to read and compose messages and to perform Account and Mailbox management tasks using any Web browser.

Even if a user prefers a regular POP or IMAP mail client, the WebUser Interface can be used to access the features unavailable in some mailers. For example, the WebUser Interface can be used to specify Subscriptions and Access Control Lists for Account Mailboxes - the features many IMAP clients do not support yet. The WebUser Interface can be used to specify Mailbox Aliases, RPOP (External POP) accounts, Account Rules, etc.

This section describes the WebUser Interface from the administrator point of view. See the [WebMail](#) section for more detailed user-level information.

WebUser Interface to Multiple Domains

When a user points a browser to the CommuniGate Pro server (to the WebUser port specified in the [HTTP module](#) settings), the Login page is displayed. The user can enter his or her name and password and start a WebUser Session.

The WebUser module checks for the domain name specified in the URL and presents the Login page for the addressed domain. If the CommuniGate Pro server `provider.com` has a secondary domain `client.com`, then the `<http://provider.com:port>` URL will display the `provider.com` Login page in a user browser, and the `<http://client.com:port>` URL will display the `client.com` Login page, even if the `client.com` has no [dedicated IP address](#).

When the WebUser module retrieves the domain name from a URL, it runs it through the [Router](#) domain-level records. So, if the Router Table has a record:

```
www.client.com = client.com
```

the `<http://www.client.com:port>` URL will be processed as the

`<http://client.com:port>` URL and will display the `client.com` Login page, too.

If the URL specifies a domain that is not among the main and secondary server domains, an error page is displayed. This usually indicates an error in your Server setup: the specified domain name has a DNS A-record that points to your server (otherwise the server will not get this request), but that name is not routed to any of the secondary domains on your server. You should either create a secondary domain with that name, or route this domain to one of the existing CommuniGate Pro domains.

If a URL specifies an IP address instead of a domain name, the WebUser module tries to find a secondary domain to which the specified address is dedicated. If no secondary domain is found, the main domain Login page is displayed.

Users can open any account in any domain from any Login page, if they specify the complete account name: if the Login page of the main Server domain is displayed (`<http://provider.com:port>`) and the `username@client.com` name is entered in the username field, the account `username` will be opened in the `client.com` domain (if the correct password is provided).

If a domain has some mailing lists, its Login page contain a link to the [Mailing List](#) archive pages.

If the Domain has the Auto-Signup option enabled, a link to the [Auto Sign-up](#) page is displayed on the domain Login page.

If the Domain has a custom Security Certificate, a `Certificate` link is displayed. If a user clicks that link, the Domain Certificate can be installed as a *trusted Certificate* in the user browser.

Account Access and WebUser Sessions

IMAP and POP are session-oriented protocols: a client mailer establishes a connection with a server, provides the data needed to authenticate the user, processes the data (mailboxes, settings, etc.) in the user account, and then closes the connection. The HTTP protocol is not session-oriented: a Web browser establishes a connection, sends one or several page requests, receives the requested data, and closes the connection.

To provide the session-type functionality, the WebUser module implements a so-called *application server*: when a user is authenticated via the "login page", a *virtual session* is created. The virtual session is an internal server data structure keeping the information about the user, open mailboxes, and other session-related data, but it is not linked to any particular network connection. When the user is working with an account using a browser, the WebUser module routes browser requests to one of the already opened virtual sessions.

In order to route requests properly, the WebUser module creates a unique session identifier (session ID) for each virtual session created and makes user browsers include the session ID into every request they send.

To avoid "hijacking" of WebUser sessions, the WebUser module remembers the network (IP) address from which the login request was received, and routes to the session only the requests received from the same IP address.

Note: Sometimes, when a user works via a proxy server, the user requests may come to the Server from different IP addresses (if the proxy server uses several network addresses). In this case, the user should disable the address-controlling option on the WebUser Interface [Settings](#) page. Usually, users of large providers (such as AOL, WebTV) access the Internet via the provider's proxy servers, so their accounts should have the address-controlling option disabled.

WebUser Interface Settings

To configure the WebUser Interface module, use any Web browser to connect to the CommuniGate Pro Server, and open the WebUser page in the Settings realm. To configure the WebUser Interface module, you should have the Can Modify Settings access right:

Sessions		Limit:
Log:		
Inactivity Time-Out:	Session Time Limit:	

Limit

Use this setting to specify the maximum number of concurrent WebUser Interface Settings.

Note: remember that browser (HTTP) connections are not the same as WebUser Sessions. It is usually enough to support 100 concurrent [HTTP channels](#) to serve 5000 Sessions.

Log

Use this setting to specify what kind of information the WebUser Interface module should put in the Server Log. Usually you should use the Major (message transfer reports) level.

The WebUser Interface module records in the System Log are marked with the WEB tag.

Inactivity Time-Out

Use this setting to specify the maximum time interval between client (browser) connections for a particular Web Session. This settings allows to disconnect the users who did not log out correctly, but simply closed their browsers or moved to a different Web site. Do not set this setting to a too small value, otherwise users can get disconnected while they are composing

letters.

Session Time Limit

Use this setting to specify the maximum "login" time for a WebUser session. The limit is checked when a browser connects and retrieves a page from the session, so it is useless to set this setting to a value that is smaller than the Inactivity Time-Out setting value.

WebUser Interface to Mailing Lists

The WebUser module presents a link to the Mailing Lists page on a domain Login Page.

The Mailing Lists page displays all [mailing lists](#) created in the domain that have the `allow anybody to browse` option enabled. Each name is a link that can be used to open a page listing messages in the mailing list. Since mailing lists are archived in mailboxes, the mailing list WebUser interface is similar to the [Mailbox Browsing](#) interface.

The mailing list Web User Interface does not require any authentication, so no virtual session is created for list users, and each browser request is processed independently.

Auto Sign-up

If a domain has the [Auto-Signup](#) option enabled, the WebUser Interface Login page contains a link to the Auto-Signup page. This page allows a new user to enter a user name, a password, the "real-life" name, and to create a new account.

When a new account is created, its options and settings are taken from the domain Account Template.

WebUser Interface Customization

The Web User Interface files are stored in the WebUser directory inside the [application directory](#). Also, inside each [domain directory](#), an empty WebUser directory is created.

The WebUser directory contains the basic HTML files and all the graphic files. Its Account

subdirectory contains all HTML files used to provide WebUser Interface to user accounts. The `List` subdirectory of the WebUser directory contains the HTML files used to provide WebUser interface to mailing lists. The WebUser directories inside the domain directories should have the same layout.

The `Strings.data` file stored in the WebUser directory contains a dictionary with all customizable HTML elements used to compose WebUser Interface HTML pages.

When the WebUser module needs to retrieve any file, it looks into the WebUser directory inside the domain directory first. If the requested file is not found there (those WebUser directories are initially empty), the module retrieves the file from the WebUser directory inside the application directory.

To customize the WebUser Interface, you should place your version of a WebUser Interface file into the proper location in the WebUser directory inside a domain directory. Your version of the file will be used for all accounts and lists in that domain.

Note: avoid modifying the original files in the WebUser directory inside the CommuniGate Pro *application directory*: when you update the software, all files in the application directory are rewritten, while files in the *base directory* (including the files inside the domain WebUser directories) are left intact.

The [Domain Administrator](#) can place HTML and other files into the WebUser directory (*publish* them):

- by manually transferring files to the appropriate location inside the domain WebUser directory;
- by using the CommuniGate Pro WebUser Interface Editor page (it can be opened by following the WebUser link on the Domain Administration pages);
- by using any HTML composer application that supports the PUT, DELETE, and MOVE HTTP methods.

The WebUser Interface Editor is the preferred method. Click the WebUser link on the top of any Domain Administration page, and the the list of all available WebUser files will appear. The list contains files found in the *application directory* WebUser subdirectory and the custom files already stored in the domain WebUser directory:

Marker	File Name	Size	Modified
	Account/		
default	AnsweredLetter.gif	890	27-Feb-99
default	AttachedFile.gif	1147	27-Feb-99
	DeletedLetter.gif	896	27-Feb-99
default	Denied.html	306	26-Mar-99
	Directory.html	1195	06-Aug-99
default	Disconnected.html	306	27-Feb-99
		
	Lists/		
		
default	UserSiteIndex.html	1019	24-Aug-99
Totals:	30	29K	

If the file does not exist in the domain WebUser directory, the file from the *application directory* WebUser subdirectory is shown, and the default marker is displayed. If the file exists in the domain WebUser directory, that file is shown and a check box is displayed in the Marker field. The subdirectories of the WebUser directory (Account, List) are listed too, and you can open them by following the subdirectory link.

To modify some element of the WebUser Interface:

- use the WebUser Interface Editor to open the directory that contains the file you want to modify;
- use the file link to open the file and/or to copy the file on your local disk;
- modify the file using any HTML editor program;
- on the same WebUser Interface Editor directory page, click the Browser button and select the modified file on your local disks;
- click the Upload File button to upload the modified file to the CommuniGate Pro WebUser directory opened in the WebUser Interface Editor.

If the WebUser directory/subdirectory did not contain a custom copy of the uploaded file, you will see the default file marker changing to a checkbox. If a custom version of that file already existed in the WebUser directory/subdirectory, the old version is replaced with the uploaded one.

To remove a custom version of a WebUser Interface file, select the checkbox on the left of that file name and click the Delete Marked button. If the file with that name exists in the *application directory* WebUser subdirectory, the file name does not disappear from the WebUser Interface Editor page, but the name gets the default marker indicating that the default (original) version of

the file will be used again.

To modify WebUser Interface files using an HTML editor that supports the PUT HTTP method (Netscape® Composer or similar product):

- use the WebUser Interface Editor to open the directory that contains the file you want to modify;
- use the file link to "open the file in the Composer";
- modify the file using the HTML editor (composer) program;
- use the Post/Publish command to upload the modified version of the page to the same location; you may have to reenter your server/domain administrator username and password into the composer program settings.

To serve heavily loaded sites, the WebUser module uses an internal cache for the WebUser Interface files. When you upload the custom versions of the WebUser Interface files using the HTML Upload File form method or HTTP PUT method, the CommuniGate Pro server automatically clears the internal domain cache (on all servers in the cluster if you employ a Dynamic Cluster), so the new file version becomes effective immediately.

If you modified the WebUser Interface files bypassing the CommuniGate Pro server (i.e. you have modified those files "in place" or uploaded them and moved into the WebUser directory using the Server OS commands), you should click the Flush Cache button on the Domain Settings page, or you can completely switch WebCaching off for that domain. See the [Domains](#) section for the details.

If you choose to modify the original files in the application directory, you may want to restart the CommuniGate Pro Server with the `--NoWebCache` option to completely disable the WebUser Interface caching. **When you upgrade to the new version of the CommuniGate Pro Server, the *application* directory is completely replaced with the new files. If you choose to modify files in the application WebUser directory, save them to a different location before you update your CommuniGate Pro Server software.**

Note: The Strings.data file is always cached. You need to use the [Flush Cache](#) button to reload the Strings.data file from the domain WebUser directory, and you need to restart the Server after you have updated the Strings.data in the application directory.

The HTML files used in the WebUser module are, in fact, the "macro files" - these text files contain macro-symbols (two-symbol combinations starting with the caret symbol ^), that are substituted with the actual account data. You should use the same macro-symbols in your versions of the WebUser pages, but you can remove some of them.

For the session-based WebUser Account Interface the Session ID (see above) is a required

parameter. The WebUser module substitutes the macro symbol ^# with the current Session ID, and it expects to get the Session ID from the SID URL parameter. Check that your versions of the WebUser Account Interface pages ensure correct passing of a Session ID within a session.

Sometimes you want to add non-HTML and non-Image files to your customized WebUser Interface design (css style sheets files, pdf documents, etc.). You should place these files into the upper level of the domain WebUser folder (not inside the Account and List subfolders). The CommuniGate Pro WebUser Interface may not serve those files though - it can retrieve only files with .html, .gif, and .jpg extensions specified in the top-level URLs

(`http://yourserver:8100/filename.extension`). This is done to support Personal WebSites without a special prefix, when the string `abc.xyz` is interpreted as the reference to the `abc.xyz` user WebSite, not as a reference to the `abc.xyz` file. To overcome this problem, refer to those files as `http://yourserver:8100/Files/filename.extension`. The server will remove the `Files` "realm prefix" and will treat the rest of the URL as the file name.



CommuniGate Pro MAPI Connector

The CommuniGate Pro Server can be used as a "service provider" for Microsoft Windows applications supporting the MAPI (Microsoft Messaging API). To use this service, a special Connector library (CommuniGate MAPI Connector.dll) should be installed on client Microsoft Windows workstations.

The CommuniGate Pro MAPI Connector acts as a "MAPI provider". It accepts Messaging API requests from Microsoft Outlook (Outlook 97, Outlook 2000, Outlook 2002, Outlook XP and later) running in the "groupware" mode, and from other Windows applications. The MAPI Connector converts these requests into extended IMAP commands and sends them to the CommuniGate Pro Server.

The CommuniGate Pro MAPI Connector also performs data conversion between proprietary Microsoft "objects" data formats and the standard Internet data formats.

The CommuniGate Pro MAPI Connector uses TCP/IP networks and should be configured to connect to any non-TLS (clear text) IMAP port of your CommuniGate Pro server (the port 143 is the standard IMAP port).

The CommuniGate Pro MAPI Connector supports both *clear text* and *secure (STARTTLS)* connections, and it can use plain text and secure CRAM-MD5 login methods.

The CommuniGate Pro MAPI Connector contains two code parts (shared libraries). Main functionality is implemented in the library stored in the Server application directory. When the MAPI Connector connects to the CommuniGate Pro Server, the Server sends the "second part" of the MAPI Connector code to the client computer. This method allows you to deploy "regular" MAPI Connector updates without running the MAPI Connector Installer on all client workstations.

The CommuniGate Pro MAPI Connector requires a special [License Key](#).

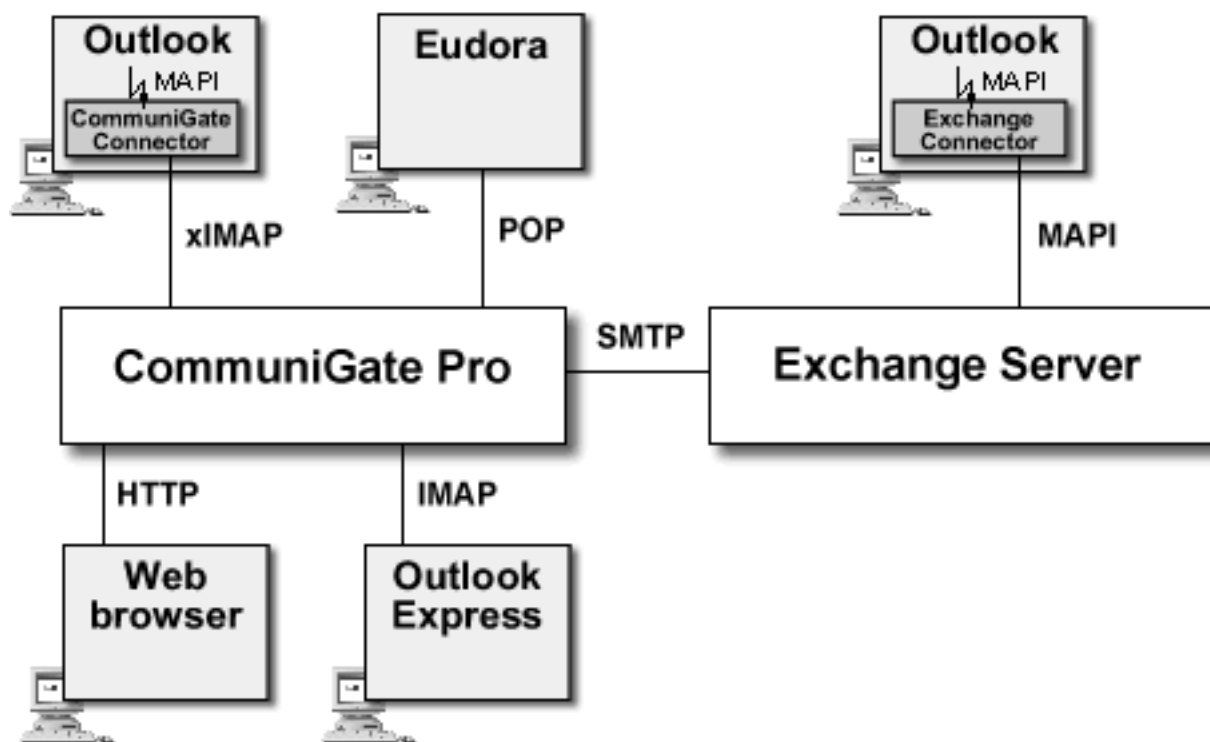
MAPI Connector Overview

MAPI stands for **M**essaging **A**pplication **P**rogramming **I**nterface, the system component that the Microsoft corporation has included into its Windows® operating system and the API to use that component with Windows applications.

The MAPI infrastructure provides an additional level of abstraction. Windows applications do not deal directly with a groupware server (or any other "data store"). Instead, applications send Messaging requests (such as "list my mailboxes", "retrieve message number X", etc.) to the MAPI component, and the MAPI component uses the installed "Connector" modules to send those requests to an Exchange® server, to locally stored "personal folders", to a fax server, etc.

The expandable nature of the MAPI architecture allows for creation of additional "Connectors" that can interact with various server products. One of the problems that such a Connector has to solve is data format: Windows applications send data objects via MAPI to Connector modules in the so-called "MAPI object" format that has nothing in common with any Internet format. The CommuniGate Pro MAPI Connector converts the MAPI data into one of the standard Internet formats and stores the converted "messaging objects" as standard Internet messages in a CommuniGate Pro mailbox. When reading those mailboxes, the CommuniGate Pro MAPI Connector converts messages back into the "MAPI object" format and passes the converted objects back to MAPI and Windows applications (such as Outlook).

Because the standard Internet formats are used, messages stored with the CommuniGate Pro MAPI Connector can be read using any standard POP3 or IMAP mail client or the CommuniGate Pro WebUser Interface:



Installing the MAPI Connector

You need to install the MAPI Connector shared library (.dll) on Microsoft Windows workstations. Download the [MAPI Connector](#) archive and unpack it. The unpacked folder contains the Setup.exe file.

Start the unpacked Setup.exe application to install or update your CommuniGate Pro MAPI Connector software. After successful install, the application may ask you to re-create your Mail Profile.

Creating a Mail Profile

When the CommuniGate Pro MAPI Connector is installed on a client workstation, you can create a Mail Profile that will tell Outlook and other applications to use the CommuniGate Pro MAPI services.

If you use Outlook 98 or Outlook 2000 check that it is configured to run in the "groupware mode". Start Outlook, and select the Options item from the Tools menu. The Options dialog box appears. Select the Mail Services Tab and click the Reconfigure Mail Support button to open the E-mail Service Options dialog box. Check that the Corporate or Workgroup option is selected.

Open the Mail Control Panel and click the Show Profiles button. The list of Mail profiles appears. If the CommuniGate Pro MAPI Installer has instructed you to re-create your existing Profile, select the old Profile and click the Remove button.

Click the Add button to create a new Profile. Depending on the version of the Outlook and the Mail control panel installed, you will see several dialog boxes. If you see a dialog box with the Additional Server Types option, select that option. Select CommuniGate Pro Server as the "service" or "additional server type".

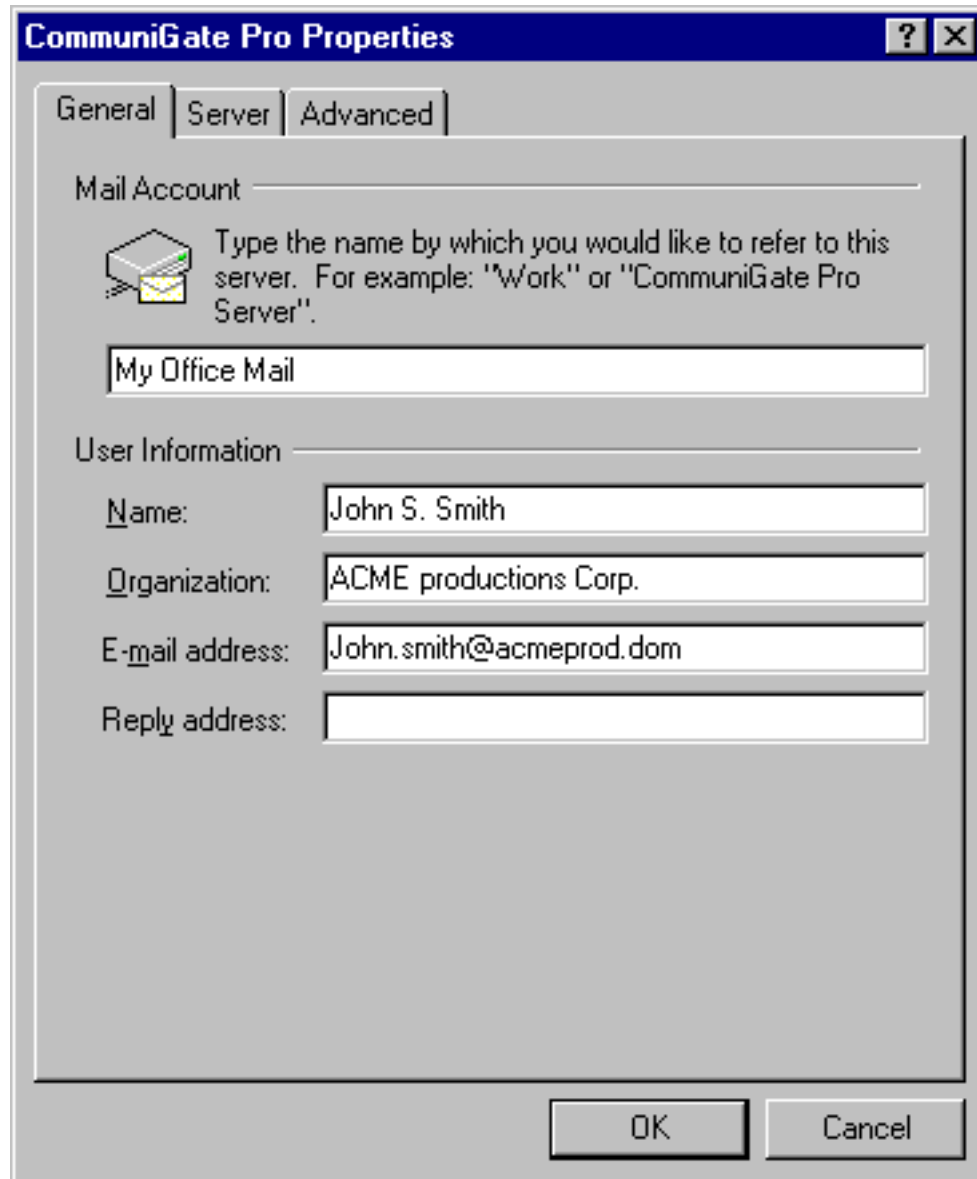
You can add other services into the same Profile.

Configuring the MAPI Connector

When the CommuniGate Pro service is added to a Mail Profile, the service settings can be

configured. Later you can open the Mail control panel, open this Profile, and open the CommuniGate Pro Server settings. You can also use the Services item in the Outlook Tools menu to open the service settings.

The General panel allows you to specify the MAPI Account name and other general data:



The screenshot shows the 'CommuniGate Pro Properties' dialog box with the 'General' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are three tabs: 'General', 'Server', and 'Advanced'. The 'General' tab contains the following fields:

- Mail Account:** A text field with the value 'My Office Mail'. Above this field is a small icon of a mail folder and a text instruction: 'Type the name by which you would like to refer to this server. For example: "Work" or "CommuniGate Pro Server".'
- User Information:** A section containing four text fields:
 - Name:** 'John S. Smith'
 - Organization:** 'ACME productions Corp.'
 - E-mail address:** 'John.smith@acmeprod.dom'
 - Reply address:** An empty text field.

At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

The Server panel allows you to specify the CommuniGate Pro Server and Account data:

The screenshot shows the 'CommuniGate Pro Properties' dialog box with the 'Server' tab selected. The 'General' tab is also visible. The 'Server Information' section contains fields for 'Server Name' (mail.acmecorp.dom) and 'Server port number' (143), with a 'Use Defaults' button. Below this is a checkbox for 'Use a secure (SSL/TLS) connection'. The 'Account Information' section contains fields for 'Account name' (john.smith) and 'Password' (masked with asterisks), with a 'Remember password' checkbox. At the bottom of the account section is a checkbox for 'Use Secure Authentication'. The dialog has 'OK' and 'Cancel' buttons at the bottom right.

Section	Field/Option	Value/State
CommuniGate Pro Server Information	Server Name	mail.acmecorp.dom
	Server port number	143
	Use a secure (SSL/TLS) connection	<input type="checkbox"/>
Account Information	Account name	john.smith
	Password	xxxxxxxx
	Remember password	<input checked="" type="checkbox"/>
	Use Secure Authentication	<input checked="" type="checkbox"/>

Server Name

The name of your CommuniGate Pro Server. This should be a domain (DNS) name that has an A-record pointing to the network (IP) address of the server.

Note: the MAPI Connector adds this name to the Account Name (see below) to send fully-qualified account names to the Server. This feature simplifies multi-domain support using a single IP address. Make sure that the specified name is either a name of some CommuniGate Pro [Domain](#), or a name of some CommuniGate Pro Domain Alias, otherwise the Server will report the account has been moved to a remote system error.

Server Port

The network port the CommuniGate Pro Server uses for MAPI clients. This is the same port as the port used for [IMAP](#) clients.

Use a Secure (SSL/TLS) connection

If this option is supported, the MAPI Connector establishes a network connection to the

specified Server port, and uses the STARTTLS command to encrypt all data sent between the workstation and the Server. See the [Security](#) section for more details.

Account Name

The name of the CommuniGate Pro Account to work with. This name can be a qualified name in the *accountName@domainName* form. If the simple name form is used (the name does not contain the @ symbol), the MAPI Connector adds the `Server Name` setting value to the specified account name.

Password

The password for the specified CommuniGate Pro Account.

Remember Password

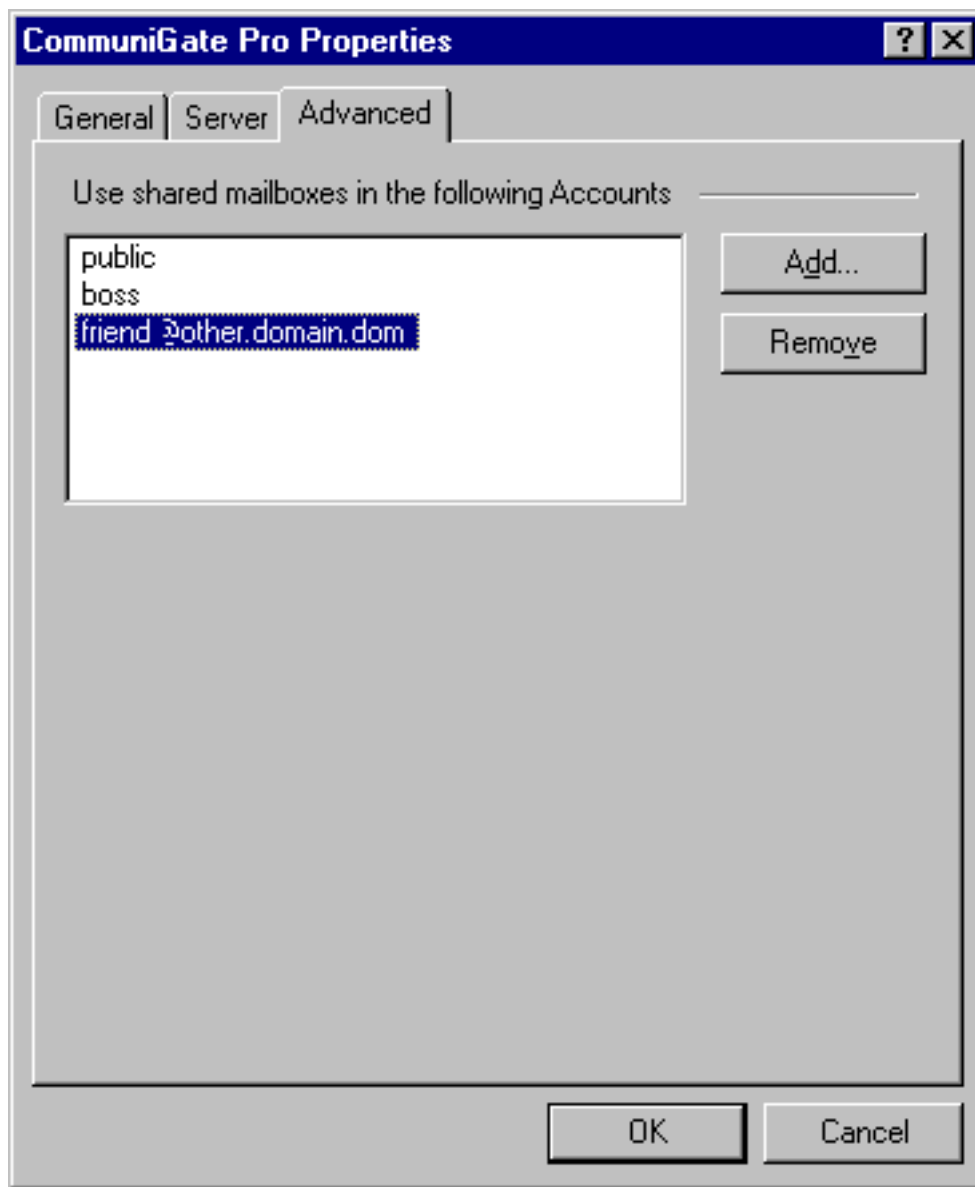
If this option is not selected, the MAPI Connector will present a Login dialog box every time it needs to connect to the Server. If this option is selected, the supplied password is stored in the MAPI Connector settings data.

Use Secure Authentication

If this option is selected, the MAPI Connector sends passwords using secure (encoded) SASL CRAM-MD5 method. The secure method does not work if passwords are stored on the Server using a one-way encrypted method (see the [Security](#) section for more details). In this case this option should be disabled, and the MAPI Connector will send passwords in clear text.

Note: if you need to send passwords in clear text while connecting to the Server via public networks, enable the `Use a Secure connection` option, so all information is encrypted.

The Advanced panel allows you to specify other CommuniGate Pro Accounts you want to work with.



Use the Add and Remove buttons to specify the names of other CommuniGate Pro Accounts. If you want to access an Account in a different Domain, specify the full name:

accountName@domainName.

The Account owners must grant you Mailbox Access Rights, otherwise you won't be able to see and open mailboxes in those Accounts.

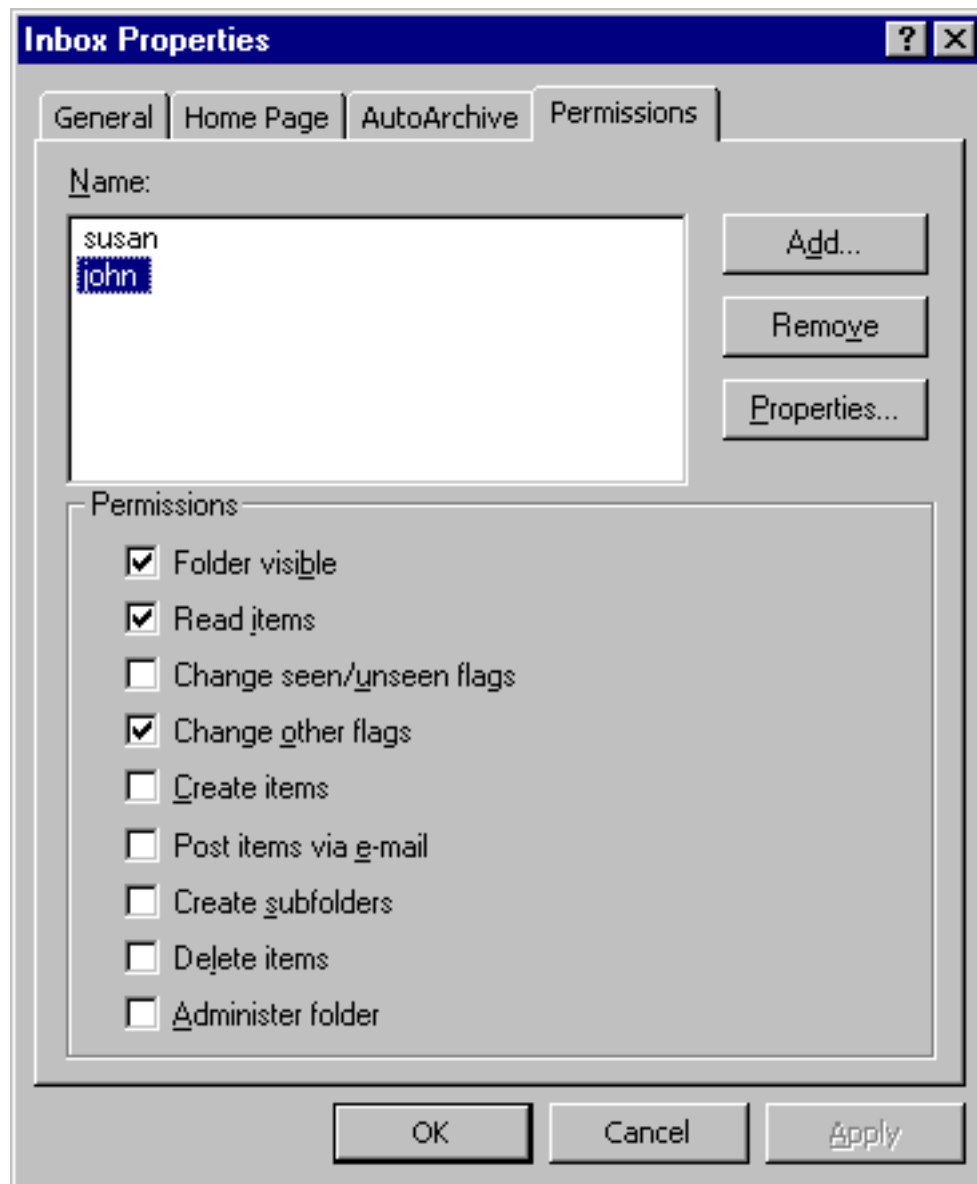
Note: Outlook needs to have access to the Deleted Items mailbox in each foreign Account you try to open. Make sure that such a mailbox exists there and that the Account owner has granted you the Lookup, Read, Insert, and Delete rights for that mailbox.

See the [Sharing](#) section for more details on foreign mailbox access.

Enabling Mailbox Sharing

You can specify Access Control List for your mailboxes to grant access to those mailboxes to other CommuniGate Pro users.

Select a mailbox in the Outlook Folder List, and use the **Properties** menu item to open the Properties dialog box. Open the Permissions panel:



Use the Add and remove buttons to specify the Accounts and other *identifiers* to specify those who should have access to this mailbox.

Select an identifier in the list and use the checkboxes to grant required access rights to this identifier. See the [Mailboxes](#) section for more details on mailbox ACLs.

Free/Busy Information

The Free/Busy information is a file specifying when the person is busy, free, out of the office, etc. This information is usually made publicly available, so other users can access it when planning their meetings, scheduling appointments, etc. To compose the Free/Busy data, the groupware client application collects data from user Calendar(s), and merges it into one Free/Busy schedule.

Posting Free/Busy Information

The MAPI Connector stores your Free/Busy information in the [Personal Web Site](#) area. Publicly available information in the standard vCalendar format is stored as the `freebusy.vfb` file in the topmost directory of your Personal Web Site. The private-type information in the Microsoft Object data format is stored as the `freebusy.eml` file in the Personal Web Site `private` directory.

This feature allows users of Outlook and other calendaring clients to access your Personal Web Site via HTTP and retrieve your Free/Busy information. The URL for the CommuniGate Pro Connector user Free/Busy information is

```
http://domainName:port/~accountName/freebusy.vfb
```

Note: Make sure your CommuniGate Pro Account limits allow the MAPI Connector to store these two files with your Free/Busy information in your Personal Web Site area.

Accessing Free/Busy Information for Other Users

In order to process Appointments and Meetings, the Outlook application on the client machine should be able to access the Free/Busy information of other users. This operation is not implemented via the MAPI Connector and should be done by the Outlook application itself. To configure your Outlook application:

- Install the Microsoft Web Publishing Wizard (it can be downloaded from the www.microsoft.com Web site).
- Select Options from the Tools menu to open the Options dialog box.
- Click the Calendar Options button to open the Calendar Options dialog box.
- Click the Free/Busy Options button to open the Free/Busy Options dialog box.
- Enter the `http://%SERVER%/~%NAME%/freebusy.vfb` URL string into the Search field. (read all the notes below).
- Click the OK buttons to close all dialog boxes.

This option will be used by the Outlook application when it needs to retrieve the Free/Busy

information for an E-mail user. The application substitutes the %SERVER% symbols with the domain part of the user E-mail, and the %NAME% symbols with the username part of the user E-mail, so for the E-mail address john@myserver.dom the Outlook will use the http://myserver.dom/~john/freebusy.vfb URL to retrieve John's Free/Busy schedule.

Note: the suggested Search URL will work only if your CommuniGate Pro Server accepts WebUser Interface connections on the port 80. If it accepts them on the default port 8100, or on any other non-standard port, the Search URL must include that port:

```
http://%SERVER%:8100/~%NAME%/freebusy.vfb
```

Note: the suggested Search URL will work only if your CommuniGate Pro Domains have names that have A-records pointing to the CommuniGate Pro server. Often, the DNS system does not contain any A-record for your mydomain.dom Domains, or those records point to a different system (company Web server), while the CommuniGate Pro Server addresses are specified as mail.mydomain.com, or cgate.mydomain.com, or mx.mydomain.com or similar DNS A-record(s). In this case the Search URL must be modified to use the proper domain names:

```
http://mail.%SERVER%/~%NAME%/freebusy.vfb
```

Note: if your CommuniGate Pro server is serving only one Domain, then you can specify the Search URL as:

```
http://mail.mydomain.com/~%NAME%/freebusy.vfb
```

where mail.mydomain.com is the name of the CommuniGate Pro Domain or its alias, that has a DNS A-record pointing to the CommuniGate Pro Server.

In this case this Search URL will work correctly only for the users of the same CommuniGate Pro Server.

The Search URL may work to retrieve the Free/Busy Information for users of other CommuniGate Pro Servers, as long as the Search URL correctly represents their Free/Busy file URLs. To overwrite the Search URL and specify a different (correct) path to some remote user Free/Busy file, create a Contact record for that user, click on the Details tab and enter the correct FreeBusy file URL into the Internet Free-Busy Address field. See the Microsoft Outlook manual for more details on these settings.

WebMail Integration

The Microsoft Outlook application moves the deleted messages to the Deleted Items mailbox,

while the WebMail Interface moves them into the Trash mailbox. To use the same mailbox in both cases:

- Login into your Account using the WebMail Interface;
- Remove the Trash mailbox;
- Open the Subscription page and create the Trash mailbox alias pointing to the Deleted Items mailbox.

Now both Outlook (and other MAPI applications) and the WebMail Interface will place deleted items in the same mailbox.

Communicating with Microsoft Exchange users

Outlook users that work with Exchange servers may have problems sending meeting requests to your users working with the CommuniGate Pro MAPI Connector. Meeting requests sent via an Exchange server may come in as plain text messages instead. The Exchange users should adjust the configuration of their Outlook applications:

Using Outlook Tools menu, Exchange users should open the Options dialog box. After they click the Calendaring Options button, the dialog box appears and they should enable the Send meeting requests using iCalendar by default option.



FTP Module

The CommuniGate FTP module implements an FTP server for TCP/IP networks.

The FTP protocol allows an FTP client application to connect to the Server computer and specify the user (Account) name and the password. If access to the specified user Account is granted, the client application can retrieve and update data inside that Account [Personal Web Site](#).

File Transfer Protocol

The File Transfer Protocol allows client computers to work with files stored on remote servers. A computer running an FTP client application connects to the server computer and provides account (user) name and the password. If access to the specified user account is granted, the client application sends protocol commands to the FTP server. These protocol commands tell the server to list all files in the current directory, to change the current directory, to retrieve, upload, rename, and remove files stored on the FTP server.

The CommuniGate Pro FTP module supports [various Internet standards](#) (RFCs).

The CommuniGate Pro FTP module supports the REST command and it can resume broken file transfer operations.

Configuring the FTP module

Use a Web browser to configure the FTP module. Open the Access page in the WebAdmin Settings section.

Serving FTP Clients	
Log:	
Channels:	<u>listener</u>

Log

Use this setting to specify what kind of information the FTP module should put in the Server Log. Usually you should use the `Major` (password modification reports) or `Problems` (non-fatal errors) levels. But when you experience problems with the FTP module, you may want to set the `Log Level` setting to `Low-Level` or `All Info`: in this case protocol-level or link-level details will be recorded in the System Log as well. Since the FTP clients send passwords in the clear text format, setting the Log to these setting for long periods of time can become a security hole if the Log file can be copied from the Server computer.

The FTP module records in the System Log are marked with the FTP tag.

channels

When you specify a non-zero value for the `TCP/IP Channels` setting, the FTP module creates a so-called "listener" on the specified port(s). The module starts to accept FTP connections from FTP clients. This setting is used to limit the number of simultaneous connections the FTP module can accept. If there are too many incoming connections open, the module will reject new connections, and the users should retry later.

If the number of channels is set to zero, the FTP module closes the listener and releases (unbinds from) the TCP port(s).

listener

By default, the FTP module Listener accepts clear text connections on the TCP port 8021.

Follow the [listener](#) link to tune the FTP [Listener](#).

If the server computer does not have any other FTP server software running, you may want to switch the FTP Listener to the port 21 (the standard FTP port).

Providing Access to Personal Web Sites

As soon as an FTP user is authenticated, the current directory is set to the topmost directory of the user [Personal Web Site](#). The FTP module allows a user to upload, download, rename and remove Personal Web Files. The FTP module also allows a user to create, remove and rename directories inside the Account Personal Web Site.



CommuniGate Pro LDAP

Module

The CommuniGate LDAP module implements an LDAP server for TCP/IP networks.

The LDAP protocol allows a client application (a mailer or a search agent) to connect to the Server computer and retrieve information from the server Directory. The LDAP protocol can also be used to modify data in the Directory.

The CommuniGate Pro LDAP module supports both clear text and secure (SSL/TLS) connections.

The LDAP module supports the "Start TLS" command (RFC2830) that allows client applications to establish a connection in the clear text mode and then turn it into a secure connection.

Lightweight Directory Access Protocol

The CommuniGate Pro LDAP module provides access to the CommuniGate Pro [Directory](#) tree and its records.

It is important to understand that the CommuniGate Pro LDAP module itself does not provide any Directory services. It just implements an access protocol, and the functionality it provides depends on the CommuniGate Pro Directory Manager and its units.

Very often LDAP services are used to look for names and E-mail addresses of Server users. But since the LDAP module provides access to the entire [Directory](#) tree, it can be used to work with any type of data placed into the CommuniGate Pro Directory. While the CommuniGate Pro Directory can be stored in several Storage Units - both local and remote, the LDAP clients see the entire Directory as one large tree.

To browse and modify the Directory, system administrators can use either LDAP clients and utilities, or the [Directory Browser](#) interface built into the CommuniGate Pro WebAdmin Interface.

Note: while the LDAP module implements an LDAP server functionality, the CommuniGate Pro Server can also work as an LDAP client, using the LDAP protocol to access external LDAP servers and their databases. Those external Directories are presented as subtrees of the CommuniGate Pro Directory tree. See the [Remote Units](#) Directory section for more details.

Configuring the LDAP module

Use a Web browser to configure the LDAP module. Open the Access page in the WebAdmin Settings section.

Serving LDAP Clients	
Log:	
Channels:	<u>listener</u>

Log

Use this setting to specify what kind of information the LDAP module should put in the Server Log. Usually you should use the Major or Problems (non-fatal errors) levels. But when you experience problems with the LDAP module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well.

The LDAP module records in the System Log are marked with the LDAP tag. Please note that LDAP is a binary protocol, so all low-level data is presented in the hexadecimal form.

Channels

When you specify a non-zero value for the TCP / IP Channels setting, the LDAP module creates a so-called "listener" on the specified port. The module starts to accept all LDAP connections that mail clients establish in order to update password data. This setting is used to limit the number of simultaneous connections the LDAP module can accept. If there are too many incoming connections open, the module will reject new connections, and the user should retry later.

If the number of channels is set to zero, the LDAP module closes the listener and releases (unbinds from) the TCP port(s).

listener

By default, the LDAP module Listener accepts clear text connections on the TCP port 389, and secure connections - on the TCP port 636. Follow the [listener](#) link to tune the LDAP [Listener](#).

Note: The pre-4.7 Netscape ® LDAP clients crash if they communicate with a very fast server

returning more than 90 records. Ask your users to update to the 4.7 or later version of Netscape browser/mailer product.

Note: The Netscape® LDAP client (version 4.7) does not correctly process the "properties" command - it always tries to connect to the port 389, even if the search was successfully made on a different (for example, secure) port.

Sometimes you need to specify the Directory Tree Root element (an empty string) as the "search base DN". Some LDAP clients do not process this situation correctly (for example, Microsoft LDAP client silently replaces an empty Search Base string with the `c=your_country` string). In these cases you should specify the string `top` as your Search Base string. The LDAP module interpretes this string as an empty string (Directory Root DN).

User Authentication

The Directory Access Rights are based on the so-called Bind DN's rather than on CommuniGate Pro account names and account rights. See the Directory Manager [Access Rights](#) section for more details.

The Directory Access Rights set by default do not require Directory (LDAP) clients to authenticate in order to retrieve any information from the Directory tree.

When an LDAP client tries to authenticate as a certain DN, the LDAP server retrieves the Directory record with the specified DN and compares that record `userPassword` attribute with the password supplied by the LDAP client. If the record exists, and it contains the `userPassword` attribute, and the attribute value matches the supplied password, the LDAP client authentication succeeds.

The LDAP module provides an alternative authentication method, with a CommuniGate Pro account name specified instead of a DN. In this case, the CommuniGate Pro Server opens the specified account and compares the account password with the supplied password. If the passwords match, the Server builds a DN for the account record using the [Directory Integration](#) settings, and uses it as the Bind DN.

Sample:

If the Directory Integration settings are:

Base DN:	<code>o=myCompany</code>
Domain RDN attribute:	<code>cn</code>

and the client has submitted the `user@domain.dom` name and the correct password for the `user@domain.com` account, then the LDAP client is authenticated with the following Bind DN: `uid=user,cn=domain.dom,o=myCompany` and this client can access the Directory information available for that Bind DN.

Note: If a user tries to authenticate using the explicitly specified `uid=user,cn=domain.dom,o=myCompany` Bind DN, only that Directory record `userPassword` attribute is checked - even if the CommuniGate Pro account `user@domain.dom` exists, its password is not checked. If a user tries to authenticate using the `user@domain.dom` string instead of a DN, only the `user@domain.dom` account password is checked, even if the `uid=user,cn=domain.dom,o=myCompany` Directory record exists and contains the `userPassword` attribute.

Note: The LDAP module uses the alternative authentication method if the specified string does not contain any equals (=) sign, or if it starts with the `mail=` symbols and does not contain any other equals (=) signs:

string specified	Method used
<code>uid=user,cn=domain.dom,o=myCompany</code>	<code>userPassword</code> record attribute
<code>ou=human resources,o=myCompany</code>	<code>userPassword</code> record attribute
<code>user@domain.com</code>	account password
<code>mail=user@domain.com</code>	account password

The LDAP module allows users to employ all [authentication methods](#) supported with the CommuniGate Pro Server.

If the account password authentication method is used, and the specified account has the [Directory Administrator](#) access right, the LDAP client can access and modify all Directory data ("master"-type access).

Central (users) Directory

Very often the LDAP services are used to retrieve information about the CommuniGate Pro [Accounts](#) and other Domain [Objects](#).

To search the Directory for CommuniGate Pro Domain Objects (Accounts, Groups, Mailing Lists), the LDAP clients should be tuned to point to the proper Subtree (this parameter is called "Search Base" in many LDAP clients). The Directory Subtree for the `company.com` domain is

`cn=company.com,o=MyCompany`, where `cn` is the Domain RDN attribute, and `o=MyCompany` is the Base DN for CommuniGate Pro Domains. The Base DN and Domain RDN attribute are the [Directory Integration](#) settings and can be modified. If these settings are modified, the locations of domain subtrees are changed, and the LDAP clients should be reconfigured to specify the new locations in their "Search Base" settings.

The `mail` Attribute processing

Most of the LDAP clients expect to see the `mail` attribute in Account and other Domain Object records. But, by default, CommuniGate Pro does not store such an attribute in those directory records.

If the LDAP module has to return such a record (a record of the `CommuniGateAccount`, `CommuniGateMailList`, or `CommuniGateGroup` object class), and that record does not contain the `mail` attribute, the LDAP module can compose that attribute on-the-fly, using the Object record DN: it takes the `uid` value from the DN (Account/Object name), the `cn` attribute value (Domain name), and merges them using the `@` sign to build the `uidValue@cnValue` `mail` attribute value. As a result, when an object is renamed (its record `uid` attribute is changed), or when the domain is renamed (the `cn` attribute in the object DN is changed), the `mail` attribute is automatically updated.

All search filters that use the `mail` attribute are modified internally to use the `uid` attribute instead.

These two features can be enabled or disabled using the Domain Integration page in the Domains realm of the WebAdmin Interface:

mail Attribute Processing	
Substitute with <code>uid</code> in conditions	Compose using <code>uid</code>



CommuniGate Pro ACAP

Module

The CommuniGate ACAP module implements an ACAP server for TCP/IP networks.

The ACAP protocol allows a client (mailer) application to connect to the Server computer and upload and download the application preferences, configuration settings and other datasets (such as personal address books).

Application Configuration Access Protocol

The Application Configuration Access Protocol allows mailers and other application store any type of structured data on an ACAP server. That data can be the application configuration data, so when the application is started on any workstation, it can connect to the ACAP server and configure itself using the configuration stored in the ACAP 'dataset' belonging to the current user.

ACAP 'datasets' can also be used to store address books, and there are several mailer applications like Mulberry® that can work with the address books stored on an ACAP server. The CommuniGate Pro [WebUser Interface](#) module uses the same datasets to store its address books. This feature allows CommuniGate Pro users to use the same personal address books via the WebUser Interface and via an ACAP-enabled mailer.

Configuring the ACAP module

Use a Web browser to configure the ACAP module. Open the Access page in the WebAdmin Settings section.

Serving ACAP Clients	
Log:	
Channels:	<u>listener</u>

Log

Use this setting to specify what kind of information the ACAP module should put in the Server Log. Usually you should use the Major (password modification reports) or Problems (non-fatal errors) levels. But when you experience problems with the ACAP module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well.

The ACAP module records in the System Log are marked with the ACAP tag.

channels

When you specify a non-zero value for the TCP/IP Channels setting, the ACAP module creates a so-called "listener" on the specified port. The module starts to accept all ACAP connections from mail clients and other applications. This setting is used to limit the number of simultaneous connections the ACAP module can accept. If there are too many incoming connections open, the module will reject new connections, and the user should retry later.

listener

By default, the ACAP module Listener accepts clear text connections on the TCP port 674. Follow the listener link to tune the ACAP [Listener](#).

The ACAP module supports the STARTTLS command that allows client mailers to establish a connection in the clear text mode and then turn it into a secure connection.



PWD Module

The CommuniGate PWD module implements a poppwd server for TCP/IP networks.

The poppwd protocol allows a client (mailer) application to connect to the Server computer and specify the user (account) name and the password. If access to the specified user account is granted, the mailer application sends the new password to the Server, and the server update the user password in the user account information data.

The PWD module also provides access to the Server [Command Line Interface](#) (CLI)

Password Modification Protocol (poppwd)

Since CommuniGate Pro supports 2 passwords per account, the PWD module can be used to modify them both. If the "old" password specified by a mail client matches the user's Server OS password on the server computer, the "new" password is stored as the user's Server OS password (*this feature is not available on some platforms*).

If the specified "old" password matches the password set in the user's Account Settings, the new password is stored in the Account Settings.

The PWD module checks the Can Modify Password [Account Settings](#) option and refuses to modify an account password if this option is disabled.

The PWD module supports the *clear text* authentication method, and it also supports the secure APOP and SASL AUTH authentication methods.

When used in a [Cluster environment](#), the PWD module can update passwords on other servers.

Configuring the PWD module

Use a Web browser to configure the PWD module. Open the Access page in the WebAdmin Settings section.

Serving PWD Clients	
Log:	
Channels:	<u>listener</u>

Log

Use this setting to specify what kind of information the PWD module should put in the Server Log. Usually you should use the Major (password modification reports) or Problems (non-fatal errors) levels. But when you experience problems with the PWD module, you may want to set the Log Level setting to Low-Level or All Info: in this case protocol-level or link-level details will be recorded in the System Log as well. Since the poppwd sends passwords in the clear text format, setting the Log to these setting for long periods of time can become a security hole, if the Log file can be copied from the Server computer.

The PWD module records in the System Log are marked with the PWD tag.

channels

When you specify a non-zero value for the TCP/IP Channels setting, the PWD module creates a so-called "listener" on the specified port. The module starts to accept all PWD connections that mail clients establish in order to updates password data. This setting is used to limit the number of simultaneous connections the PWD module can accept. If there are too many incoming connections open, the module will reject new connections, and the user should retry later. If the number of channels is set to zero, the PWD module closes the listener and releases (unbinds from) the TCP port.

listener

By default, the PWD module Listener accepts clear text connections on the TCP port 106. Follow the listener link to tune the PWD [Listener](#).

Providing Access to the Server CLI

As soon as a PWD user is authenticated, the Server Command Line Interface (CLI) commands are accepted. See the [Command Line Interface](#) chapter for the details.



CommuniGate Pro: Directory

The CommuniGate Pro Server includes the Directory Manager implementing high-performance standards-based Directory storage.

The Directory can contain records about the CommuniGate Pro Accounts, Domains, and other objects. It can also contain any other type of records, and it can be used as a stand-alone Directory Server serving any LDAP-based applications.

The Directory Manager is not the same as an LDAP server. The CommuniGate Pro LDAP module provides access to the Directory for LDAP clients, but various CommuniGate Pro components (Account and Domain Managers, WebUser Interface, etc.) access the Directory data directly, bypassing LDAP communications.

The Directory Manager implements Meta-Directory: it can store directory data in one or several sets of the server files (Local Units), and it can also use external LDAP servers as Directory Remote Units. Many different configurations are possible. The following simplest configurations are used most often:

- All Directory data is stored in a single Local Unit. In this case Meta-Directory is the same as a Directory implemented with a regular LDAP server.
- All Directory data is stored in a single Remote Unit. In this case Meta-Directory is used just as a method to access records stored on an external LDAP server.

What is Directory?

attribute

a name (*attribute name*) and one or several *attribute values*.
Usually an attribute is presented in the *name=value* form.

Attribute names are case-insensitive.

Samples:

```
userName=john  
eyeColor=blue
```

object class

an attribute with the `objectClass` name; this attribute is used to specify a nature of the object it belongs to.

Samples:

```
objectClass=person
objectClass=organization
```

distinguished name (DN).

a sequence of attributes presented in the *name=value* form and separated with the comma (,) sign.

DNs are used as unique names for objects (records).

Sample:

```
userName=john,server=BigIron,realm=Internet
```

DNs are used to build object name trees, with the rightmost attribute specifying the most generic name, and the leftmost attribute specifying the unique object name itself.

The leftmost attribute is called *Relative Distinguished Name* (RDN) - it provides a unique name for the object among all objects with DNs having the same parent DN.

Sample:

```
userName=jim,server=BigIron,realm=Internet
this is a different DN, but it has the same "parent DN"
(server=BigIron,realm=Internet)
```

Sample:

```
userName=john,server=SmallCopper,realm=Internet
this is a different DN, with a different "parent DN"
(server=SmallCopper,realm=Internet)
```

directory record or object

set of *attributes* with a *distinguished name*

Usually a record is presented as several lines starting with the name presenting the record DN, followed by the lines presenting the record attributes. Several records are usually separated with an empty line.

Sample:

```
DN: userName=jim,server=BigIron,realm=Internet
objectClass=person
eyeColor=blue
mailboxLimit=1024000

DN: userName=john,server=BigIron,realm=Internet
objectClass=person
eyeColor=green
mailboxLimit=2048000
```

Note: the LDAP standard recommends to include the RDN attribute into the set of attributes making up a directory record. CommuniGate Pro Directory Manager enforces this rule.

directory

a set of *directory records*; this can be a very large set (millions of records). The set is organized as a tree using DNs. Records are removed automatically when the record with the parent DN is removed. Record DNs are updated automatically when the parent DN is changed (renamed).

directory schema

a set of directory restrictions, including:

- a set of *attribute names* that can be used in the Directory (`userName`, `mail`, `city`, `eyeColor`, ...);
- a set of *objectClass* attribute values that can be used in the Directory (`person`, `organization`, `device`, `printer` ...);
- for each *objectClass* - names of the attributes that must be present in the object record; for records with `objectClass=person` a schema may require attributes with `cn` (canonical name) and `sn` (surname) names;
- for each *objectClass* - names of the attributes that may be present in the object record; for records with `objectClass=person` a schema may allow attributes with `driverLicense` and `eyeColor` names.

Directory Storage Units

While the entire CommuniGate Pro Directory is presented to its clients as one large tree of directory records, its subtrees can be stored in separate *storage units*. This type of "virtual" directories is often called Meta-Directory.

CommuniGate Pro Directory supports two types of storage units:

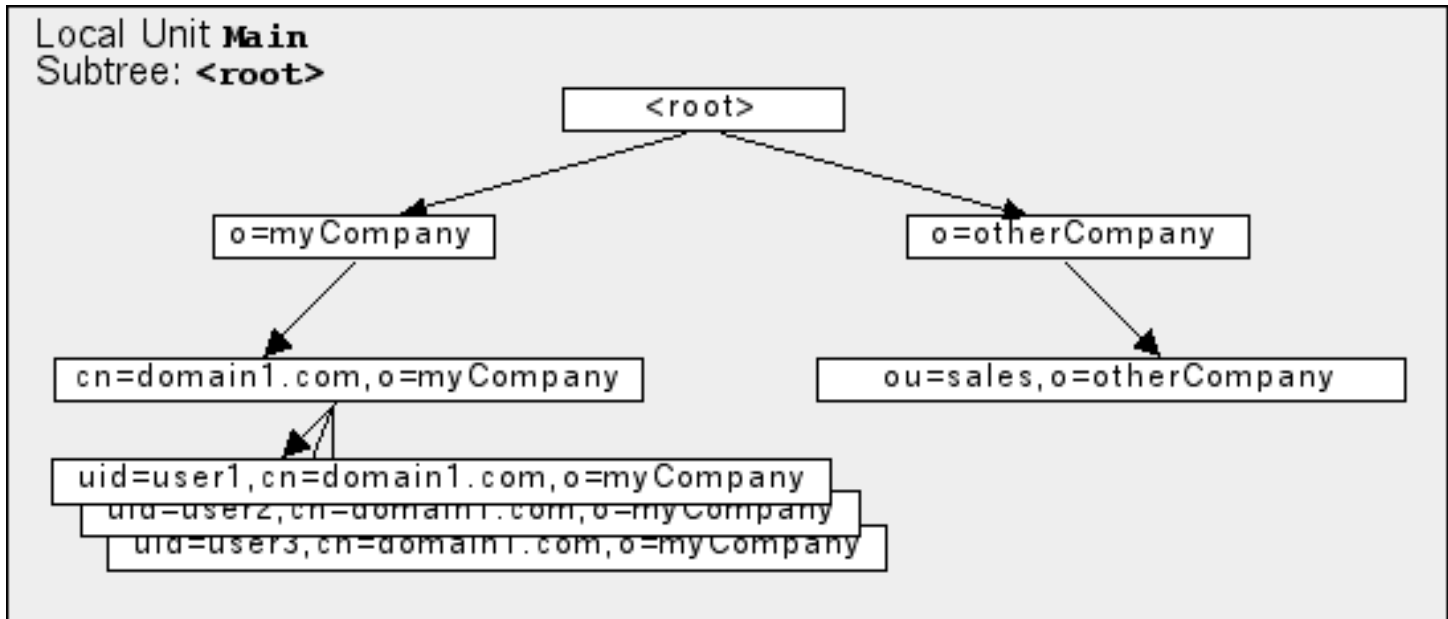
- *local* units - sets of files managed with the CommuniGate Pro File Directory Manager. These files contain directory records, replication information, and subtree schemas.
- *remote* units - descriptors managed with the CommuniGate Pro LDAP Directory Manager. These descriptors contain the information about remote Directories accessed via LDAP.

As a result, the CommuniGate Pro Directory may include subtrees located on remote servers. If an LDAP server `ldap.server.dom` provides access to some directory tree, you can create a remote unit in the CommuniGate Pro Directory that points to the `ldap.server.dom` server and the entire `ldap.server.dom` directory tree or one of its subtrees will be seen in the CommuniGate Pro Directory as some subtree.

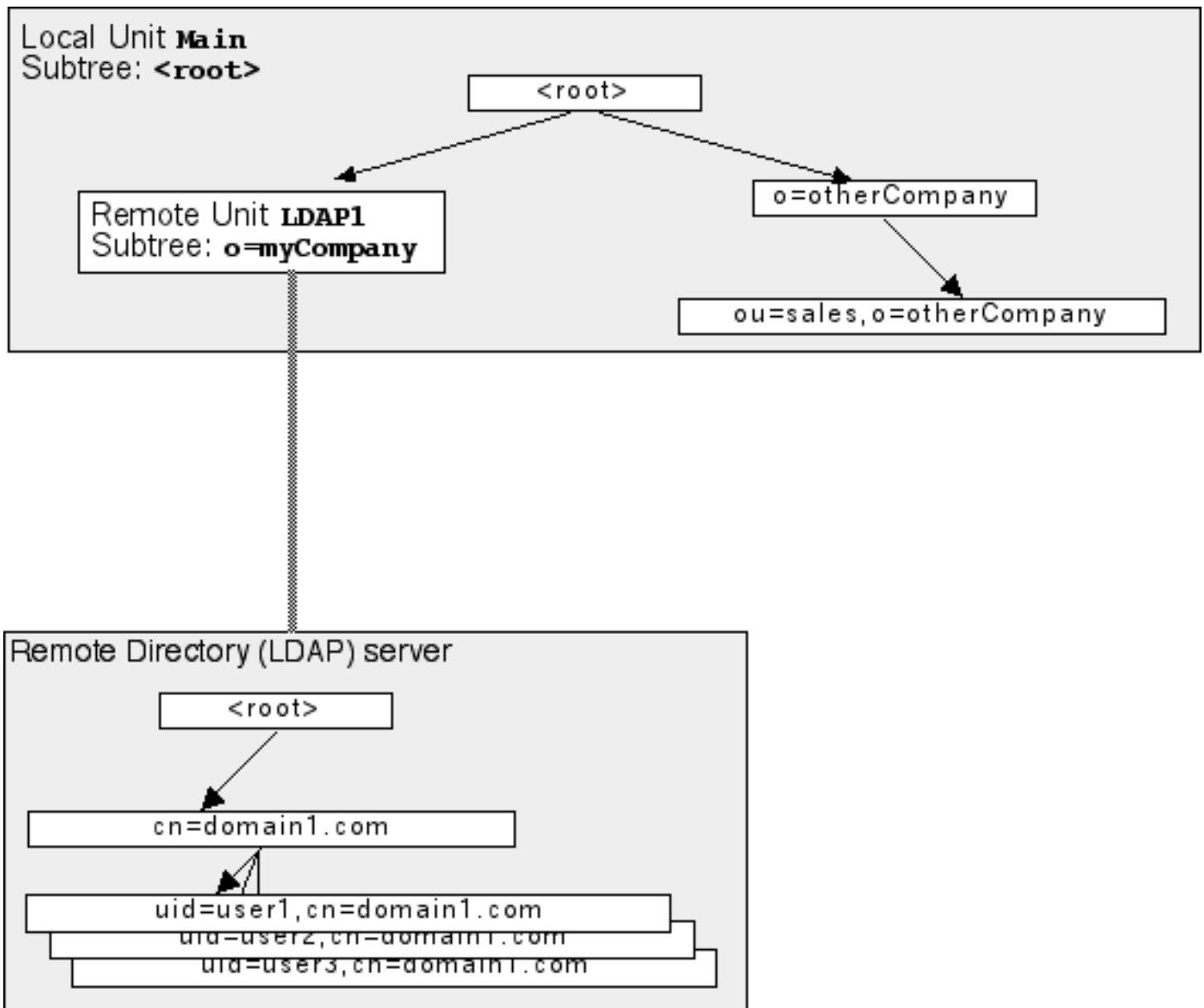
Initially, the CommuniGate Pro server creates one Local Storage Unit `Main` that contains the entire

directory. You may add additional storage units using the WebAdmin Interface:

The diagram below shows a directory stored in one Local Unit Main:



The diagram below shows the same directory stored in 2 Storage Units, with the entire o=MyCompany subtree stored in a separate Storage Unit LDAP1:



Example 1:

An external LDAP server `ldap1.com` has a subtree `o=company.com`, and you want to store all CommuniGate Pro Domain and Account records in that subtree. You can use the following settings:

- In the [Integration](#) settings, specify Base DN as `o=MyCompany,o=ldap1`
- Create a Remote Unit MYLDAP for the `o=ldap1` subtree.
- Enter `ldap1.com` as the server name, and an empty string as the Server Subtree in the MYLDAP Unit Settings.

Now, when the CommuniGate Pro Server tries to access a directory record for the account `john` in the `domain1.com` domain:

- The `uid=john,cn=domain1.com,o=MyCompany,o=ldap1` DN is formed.
- The Directory Manager detects that this record should reside on the MYLDAP Unit, and it asks that Unit to perform the requested operation on the record with the `uid=john,cn=domain1.com,o=MyCompany` DN (the Unit "mount point", `o=ldap1`

is removed from the DN).

- The MYLDAP Unit sends the request for the `uid=john,cn=domain1.com,o=MyCompany` DN to the remote server `ldap1.com`.

Example 2:

An external LDAP server `ldap1.com` has a subtree `o=company.com`, and you want to store all CommuniGate Pro Domain and Account records in that subtree (the same situation as in the Example 1). You can use the following settings:

- In the [Integration](#) settings, specify Base DN as `o=ldap1`
- Create a Remote Unit MYLDAP for the `o=ldap1` subtree.
- Enter `ldap1.com` as the server name, and `o=MyCompany` as the Server Subtree in the MYLDAP Unit Settings.

Now, when the CommuniGate Pro Server tries to access a directory record for the account `john` in the `domain1.com` domain:

- The `uid=john,cn=domain1.com,o=ldap1` DN is formed.
- The Directory Manager detects that this record should reside on the MYLDAP Unit, and it asks that Unit to perform the requested operation on the record with the `uid=john,cn=domain1.com` DN (the Unit "mount point", `o=ldap1` is removed from the DN).
- The MYLDAP Unit adds the Server Subtree suffix (`o=MyCompany`) to the DN, and then it sends the request for the `uid=john,cn=domain1.com,o=MyCompany` DN to the remote server `ldap1.com`.

Click the Directory link in the left frame of the CommuniGate Pro Server WebAdmin Interface. The Directory Storage Units page opens. You need to have the Directory access right to open this page.

Filter:	
<i>2 of 2 Storage Units selected</i>	
Subtree	Unit Name
<root>	Main
o=node6	node6

The `<root>` string is used to specify the name of the default Storage Unit (i.e. the Storage Unit that stores the root of the Directory Tree).

To create a new Storage Unit, type a name for the new unit (this name will be used for administrating only), the Distinguished Name (DN) for the subtree that should be stored in that unit, and click the Add Remote Unit or Add Local Unit button.

Local Units

Local Directory unit is a set of files containing unit data (records/entries), unit schema, unit settings, and unit modification journal.

Open the Directory WebAdmin page and click the name of a Local Storage Unit. The unit Settings page opens.

Settings			
Log:			
Enforce Schema	Search Result Limit:	records	

Log

Use this setting to specify what kind of information the Local Storage Unit Manager should put in the Server Log.

Enforce Schema

When this option is selected, the Local Storage Unit Manager compares the structure of all new and updated records with the Unit Schema. If this option is disabled, then the Manager checks only the names of the record attributes and ensures that those attributes are included into the Unit Schema, but it does not enforce the objectClass-related restrictions.

Search Results Limit

This setting limits the maximum number of records a Directory Search operation can return.

Each Local Unit has its own Schema. See the [Schema](#) section for more details on Unit Schemas.

You can browse and modify the Local Unit Schema by retrieving and modifying the virtual record with the cn=schema DN. If the Local Unit is mounted on some Directory subtree, the DN for the Unit Schema record is cn=schema , subtree.

Remote Storage Unit

Click a Remote Storage Unit name on the Directory WebAdmin page to open the unit Settings page:

Settings

Log:

LDAP Server Name:

Server Subtree:

BIND DN:

BIND Password:

Protocol Version:

Channel Cache:

Log

Use this setting to specify what kind of information the Remote Unit Manager should put in the Server Log.

LDAP Server Name

This field specifies the name or the IP address of the remote LDAP server that hosts the Storage Unit subtree. If the remote LDAP server uses non-standard TCP port, you can specify the Server Name as `servername:port`.

Server Subtree

This field specifies the remote LDAP server subtree to be "mounted". When this setting is left empty, the entire remote directory becomes visible as a CommuniGate Pro Directory subtree.

BIND DN, BIND Password

These fields specify the Distinguished Name and Password to use for "binding" (logging into) the remote LDAP server. If these fields are left blank, the CommuniGate Pro will use anonymous access to the remote LDAP server.

Protocol Version

Use this field to specify the LDAP protocol version supported with the remote LDAP server.

Channel Cache

Use this field to specify the number of "cached" TCP connections used to connect to the remote LDAP server.

Remote Directory Root

In certain situations a CommuniGate Pro server should not keep any Directory data in its Local Storage units. Instead, all Directory records should be stored on a remote LDAP server (some other

CommuniGate Pro server or a third-party Directory server). In this case, the CommuniGate Pro "root" should be stored in a Remote Storage Unit pointing to that external server. By default, the Directory "root" is stored in the Main Local Storage Unit. To tell the CommuniGate Pro server that the Directory "root" and the entire Directory tree is stored on a remote server, follow these steps:

- open the Main Storage Unit settings.
 - relocate the Unit to a fictitious subtree o=dummy
 - create a Remote Storage Unit RemoteRoot specifying an empty string as its Subtree.
 - configure the RemoteRoot Storage Unit so it will access the proper remote LDAP server.
 - check that the remote directory is available (using the CommuniGate Pro Directory Browser).
 - open the Main Storage Unit settings and click the Remove Unit button to remove this Storage Unit.
-

Binding to the Directory

Directory records can be (and usually are) protected from unauthorized access. When users want to access protected Directory data, they should authenticate themselves first. This process is called *binding* and successful authentication "binds" the user to a certain DN (distinguished name) in the Directory.

When a user tries to read or modify the Directory data, the binding DN is used to check the Directory [Access Rights](#).

When a user accesses the Directory from a CommuniGate Pro [Web User Interface](#) session, the binding DN is the DN of the user Account record:

`uid=accountname , cn=domainname , o=MyCompany.`

See the [Directory Integration](#) chapter for the details.

When the Directory is accessed using the [LDAP](#) module, the client can authenticate itself using the CommuniGate Pro Account name and the Account password. In this case, the binding DN is the DN of the Account record.

Before converting the user account name into the account Directory record DN, the user account [Server Access Rights](#) are checked. If the account has the Directory access right, the special "master" bind DN is used instead of the user account record DN. Clients with the "master" bind DN have unlimited Directory access rights.

Any Directory DN can be used for LDAP binding. The directory record with the specified DN must

exist, the record should contain the `userPassword` attribute, and the attribute value must match the supplied password string.

If a client has not authenticated itself, the special `anyone` bind DN is used.

Access Right Records

CommuniGate Pro Directory restricts client rights to read, search, and modify Directory records. The Directory contains a set of the Access Right records that allow and prohibit directory operations depending on the target directory subtree and on the client binding DN.

Open the Directory Access Rights page to set the Access Right records:

Name	Target	Bind DN	Type	Rights
				specifications
				specifications
				specifications

Each Access Right record has:

- a name.
- a target: the DN the record applies to; wildcard characters ("*") can be used in Target strings.
- a Bind DN: the client binding DN the record applies to.
- the record type: enabling or disabling.
- a link to the Record specific access rights.

The Up and Down buttons allow you to move the records in the table, increasing and decreasing record priorities.

When a client requests to perform a search, read, modify, or any other operation on a record or a subtree with a certain DN, the Access Right records are checked from top to the bottom. The server looks for an Access Right record that:

- has the Target field matching the DN specified in the client request.
- has the Bind DN field matching the client binding DN.

- has the operation (delete, create) matching the requested operation, or has the attribute matching the attribute used in the operation.

When such an Access Right record is found, the record type specifies if the operation is allowed or prohibited. If no Access Right record is found, the operation is prohibited.

If the client binding DN is "master" (see above), all operations are allowed.

When a client requests a "read"-type operation, the procedure is repeated for all attributes the client wants to retrieve. If the operation is prohibited for all specified attributes, the read operation fails. Otherwise, the operation is performed, and the attributes the client has a right to retrieve are returned to the client.

If a client requests a "search"-type operation, the procedure is repeated for all attributes used in the search filter. If the search operation is prohibited for at least one of those attributes, the search operation fails.

If a client requests a "rename"-type operation, the procedure is used twice: to learn if the client has a right to delete the original Directory record, then to learn if the client has a right to create a Directory record in the new location.

Special strings can be used in the Bind DN field:

brother

the Access Right record is applied if the Bind DN and the Target DN has the same parent DN. For example, the `uid=someuser,cn=domain1.com` and `uid=otheruser,cn=domain1.com` DN's are "brothers". This type of bind DN specifications is useful to grant CommuniGate Pro users access to the Directory records of other users in the same CommuniGate Pro domain.

parent

the Access Right record is applied if the Bind DN is a parent of the Target DN. For example, the `cn=domain1.com` DN is a parent of `uid=user1,cn=domain1.com` and `id=book1,uid=user1,cn=domain1.com` DN's.

child

the Access Right record is applied if the Target DN is a parent of the Bind DN.

self

the Access Right record is applied if the Target DN is the same as the Bind DN. This type of bind DN specification is useful to grant CommuniGate Pro Account users a right to modify their own directory record attributes.

To create an Access Right record, enter the record name, target DN, and bind DN into the last empty element of the Access Rights table and click the Update button. Use the Up buttons to set the record

priority.

To remove an Access Right record, delete the record name and click the Update button.

Access Right Specifications

To specify Directory Access Rights, open the Access Rights page and click the "specifications" link to open the Access Right record:

Entry-Level	
delete entry	create entry

These options specify if clients with the given Bind DN can create or delete records with the given Target DN.

Readable Attributes

This field lists the data attributes that clients with the given Bind DN can read from the records with the given Target DN. The attribute names should be comma-separated. To allow clients read all record attributes, use the asterisk ("*") sign.

Searchable Attributes

This field lists the attributes that clients with the given Bind DN can use in filters when searching Target DN subtrees.

Modifiable Attributes

This field lists the data attributes that clients with the given Bind DN can modify in the Target DN records.

Sample Access Right record:

Target DN

`uid=*,cn=domain1.com`

Bind DN

`brother`

Type

`allow`

Readable Attributes

`objectClass,officeEmail,roomNumber,cn,uid`

This record allows all `domain1.com` users to read the `objectClass`, `cn`, `uid`, `officeEmail`, and `roomNumber` attributes from Directory records of other domain users.

Sample Access Right record:

Target DN

`cn=domain1.com`

Bind DN

`child`

Type

`allow`

Readable Attributes

`objectClass,officeEmail,roomNumber`

Searchable Attributes

`cn,uid`

This record allows all `domain1.com` users to search the domain Directory subtree by `cn` (canonical name) and `uid`, but not by other readable attributes.

Directory Browser

The CommuniGate Pro WebAdmin Interface includes a Directory Browser. Open the Directory page and click the Browser link to open the Directory Browser page.

The Browser page includes the DN field:

Distinguished Name

Use this field to type the DN of the Directory record/subtree you want to view and click the Go

button. Click the Up button to remove the leftmost DN element and open the parent Directory record.

The next panel displays the Directory record with the specified DN:

Attribute	Value
businesscategory	"Software development/technical support"
description	"\"subsidiary for Stalker Software, Inc.\""
o	"Stalker Labs"
objectclass	organization
seealso	"o=Stalker Software, c=US"
telephonenumber	676-555-1212

If the record with the specified DN could not be retrieved, this panel will contain the error message.

The next panel displays all record children.

Subtree	
Filter:	
RDN	objectClass
cn=aaa.com	CommuniGateDomain
cn=bbb.com	CommuniGateDomain

Use the pop-up menu to limit the number of records displayed on the subtree panel.

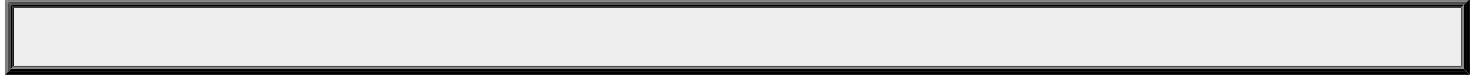
To search for specific records, enter an LDAP filter string (in the RFC 2254 format) into the Filter field and click the Display button.

The table elements display children RDNs and object classes.

Click the child element RDN link to open the child record in the Directory Browser.

Importing Directory Data

The CommuniGate Pro WebAdmin Interface allows the Server Administrator to import directory modifications from text files in the LDIF and "replug" formats:



To import data into the CommuniGate Pro Directory, click the Browse button and select an LDIF file on your workstation. Click the LDIF Import button to insert all records from the selected LDIF file.

To apply a set of record modifications to the CommuniGate Pro Directory, click the Browse button and select a "replug" file on your workstation. Click the LMOD Import button to apply all modifications from the selected file.



CommuniGate Pro: Directory Schema

The CommuniGate Pro Local Storage Units support expandable Directory *Schema*.

Every Unit has its own Schema that specifies the data (object classes and attributes) that can be stored in that Directory Unit.

You can view and modify the Unit Schema using the Web Administration Interface. Open the Local Unit Settings page and click the Schema link. The Schema page will open and it will display all attributes and object classes defined for this Storage Unit.

Default Schema

When a new Local Storage Unit is created, the Default CommuniGate Pro Schema is automatically created for that Local Storage Unit.

Record Attributes

The first part of the Schema page is the list of all record Attributes that can be used in this Storage Unit.

Attributes		
Name:		
ObjectID:		
Name	ObjectID	Syntax
objectClass	2.5.4.0	
aliasedObjectName	2.5.4.1	
cn	2.5.4.3	
sn	2.5.4.4	
c	2.5.4.6	
l	2.5.4.7	
st	2.5.4.8	
.....		
serverAccessRights	2.5.4.10103	
webUserSettings	2.5.4.10110	

This table lists all attributes defined in the Local Unit Schema.

You can add new attributes to the Unit Schema. Type the new attribute name and (optionally) new attribute Object ID (OID) into the text fields and click the Add Attribute button.

Object Classes

The second Schema page table lists all object classes defined in this Local Unit Schema:

Object Classes				
Name:				
ObjectID:				
Parent:				
Name	ObjectID	Parent Class	Required Attributes	Optional Attributes
top	2.5.6.0		objectClass	
alias	2.5.6.1	top	aliasedObjectName	
country	2.5.6.2	top	c	
organization	2.5.6.4	top	o	street, postOfficeBox, telephoneNumber, facsimileTelephoneNumber, userPassword, dc
organizationalUnit	2.5.6.5	top	ou	street, postOfficeBox, telephoneNumber, facsimileTelephoneNumber, userPassword, dc
person	2.5.6.6	top	cn, sn	description, telephoneNumber, facsimileTelephoneNumber, userPassword
organizationalPerson	2.5.6.7	person		
inetOrgPerson	2.16.840.1.113730.3.2.2	organizationalPerson		uid, mail
CommuniGateDomain	2.5.1000.0	organization		accessModes, autoSignup, RPOPLimit, accountsLimit, storageLimit, listsLimit, trailerText, webBanner, mailRerouteAddress, foldering, mailToAllAction, mailToUnknown, centralDirectory, accountsLogLevel, mailboxesLogLevel, domainAccessModes, IPMode, IPAddresses, webUserCache, externalLocation, externalLockType, osUserName

CommuniGateAccount	2.5.1000.1	inetOrgPerson	1	maxAccountSize, externalINBOX, hostServer, maxWebSize, maxWebFiles, accessModes, rulesAllowed, RPOPAllowed, PWDAAllowed, mailToAll, addMailTrailer, addWebBanner, passwordEncryption, defaultMailboxType, useAppPassword, useSysPassword, useExtPassword, requireAPOP, recoverPassword, storageLocation
CommuniGateAccountTemplate	2.5.1000.2	CommuniGateAccount		initialMailboxes, initialSubscription
CommuniGateAlert	2.5.1000.20	top		alertTimeStamp, alertText
CommuniGateAccess	2.5.1000.21	top		serverAccessRights
CommuniGateWebUser	2.5.1000.22	top		webUserSettings

To add an objectClass to the Local Unit Schema, enter the new class name, (optionally) class object ID (OID), and select the parent objectClass from the pop-up menu listing all existing classes. Click the Add Class button to add a new class to the Schema.

Click the class name link to open the Class [Descriptor](#) page.

Object Class Descriptor

You can click the class name on the Local Unit Schema page to open the Object Class Descriptor page:

Required Attributes	Optional Attributes
cn, sn	description, telephoneNumber, facsimileTelephoneNumber, userPassword
objectClass	

This table lists the Class attributes - required and optional.

The first part of the table lists the attributes defined for the class itself, while the second part of the table lists the attributes defined in the class parent classes.

You can extend your Schema by adding more attributes to a Schema Class. Select the attribute name from the pop-up

menu and click either the Add Required Attribute or Add Optional Attribute button.



CommuniGate Pro: Directory Integration

CommuniGate Pro [Directory](#) can be used to store any information. One of the Server features is integration of the CommuniGate Pro Directory and CommuniGate Pro [Domains](#). The integration level is selected on a per-domain basis.

A Server Administrator can control how CommuniGate Pro Domains are integrated with the Directory. Open the Domains page and follow the Directory Integration link on that page.

Central Directory Concept

CommuniGate Pro Domains can use the following levels of Directory integration:

- No integration; the Domain and the Domain Accounts settings are stored in `.settings` files, and when an Account is created, updated, renamed, or removed, Directory is not updated.
- Synchronized; the Domain and Domain Account settings are stored in the `.settings` files; each Account has a Directory record that stores *some* of the Account settings as attributes:
 - the Account name in the `uid` attribute
 - the Account Real Name in the `cn` attribute
 - the hosting server main domain name in the `hostServer` attribute (this attribute is needed to implement Directory-based [Static Clusters](#))
 - an optional set of custom settings/attributes.

When an Account is updated, renamed, removed, or updated, the Directory is automatically updated.

- Directory-based. The Domain and the Domain Account settings are stored in the Directory records. The `.settings` files are not created, and the Server retrieves all settings information from the Directory.

Finally, the CommuniGate Pro can use regular (non Directory-Based) domains, but still allow account provisioning via LDAP, so it looks like the Directory-Based Domains are used and LDAP commands can create and update Account. This feature is called LDAP-based Provisioning.

Attribute Renaming

Some Domain and Account settings names may not match the standard attribute names used in the Directory Schema. For example, the Account setting `Real Name` has to be stored in the Directory as the `cn` (common name) attribute, and the custom settings `surname` and `city` (see below) should be stored as attributes `sn` and `l`.

When you need to add an attribute to your Directory Schema, always try to use attribute names specified in one of the LDAP Internet Standards (RFCs). If this attribute should be used for Directory Integration (i.e. it will be used to store some Domain or Account setting value), you may want to use the Attribute Renaming capability to "map" CommuniGate Pro Domain or Account setting name on some Directory Attribute name.

Use the Attributes Renaming table to specify the name translation rules:

Attributes Renaming	
Name in CommuniGate	Name in Directory

Note: The Attributes Renaming feature works only for the Directory Integration component of the CommuniGate Pro Server. If you access the CommuniGate Pro Directory directly (via the [LDAP module](#), for example), no renaming takes place: LDAP clients should specify the Directory Attribute names, and the returned records have Directory Attribute names, not CommuniGate Pro Domain and Account setting names - "`cn`", not "`RealName`" and "`userPassword`", not "`Password`".

Domains Subtree

For each regular CommuniGate Pro Domain with the Directory setting set to `Keep in Sync`, and for each Directory-Based Domain a Directory Subtree is created. This Subtree has the Domain record as its root, and all Domain Account records as the Subtree elements ("leaves"). For Directory-based Domains, additional elements are created to store Account aliases, Domains settings, etc.

The Domain Subtree panel allows you to specify the location of Subtrees created for each CommuniGate Pro Domain:

Domains Subtree
Base DN
Domain RDN attribute
Domain objectClass
UID subtree

Base DN

This field specifies the "base" DN for all domains in the Central Directory. You may want to set it as:

o=your company name

so each CommuniGate Pro Domain will have the following DN:

cn=domain name,o=your company name

When a domain is placed into the Directory, a record with its DN is created. If the Base DN does not exist, the Directory Manager may return an error. Use the Create It button to create an empty record with the Base DN.

If you are an ISP you may want to give each domain you host the top-level DN:

cn=domain name

In this case, specify an empty string in the Base DN field.

Domain RDN attribute

This field specifies the attribute name to use for Domain record RDNs. In most cases, the default value (cn) is the best choice. However, you can change that to the name of any other attribute defined in the Directory schema. If you set this name to o, the CommuniGate Pro Domain records will have the following DNs:

o=domain name,base DN

Note: If you specify the string dc as the Domain RDN attribute, then the DN for a CommuniGatePro domain mail.domain.dom will be composed as dc=mail,dc=domain,dc=dom.

Domain objectClass

This field specifies the *objectClass* for CommuniGate Pro Domain records in the Directory. The CommuniGateDomain objectClass defined in the CommuniGate Pro Directory Manager schema is the default value. If you choose to select a different objectClass, make sure it exists in your Directory schema.

For regular domains, the domain Directory record is empty. As a result, you may use any objectClass that can store the cn attribute (or the attribute you have specified in the Domain RDN attribute setting).

For Directory-based Domains, the domain Directory record contains all domain settings, so the objectClass for these records should support all attributes included into the CommuniGateDomain objectClass.

UID subtree

If this field is empty, then the domain object (account, groups, lists, forwarders) records are stored in the directory using the following DNs: uid=*objectName*,*domain DN*.

If the base DN is o=mycompany, and the Domain RDN attribute is cn, then the directory record for the user1 account in the domain1.dom domain will have the following DN:

`uid=user1,cn=domain1.dom,o=mycompany`

The UID subtree parameter allows you to place the objects "below" the domain tree. If the UID subtree is set to `ou=People`, then the record for the same account will have the following DN:

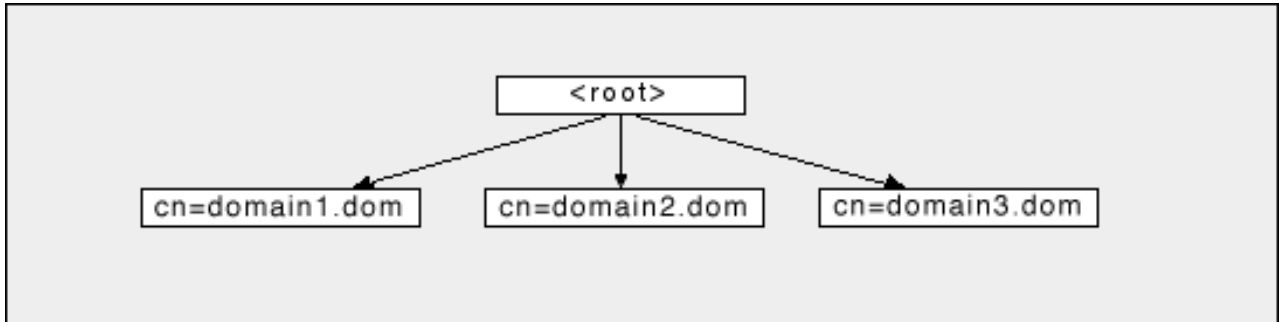
`uid=user1,ou=People,cn=domain1.dom,o=mycompany`

If the Domain RDN attribute is set to `dc`, then the record for the same account will have the following DN:

`uid=user1,ou=People,dc=domain1,dc=dom,o=mycompany`

Example:

BaseDN is an empty string, Domain RDN Attribute is `cn`, and three CommuniGate Pro domains (`domain1.dom`, `domain2.dom`, and `domain3.dom`) have been created:



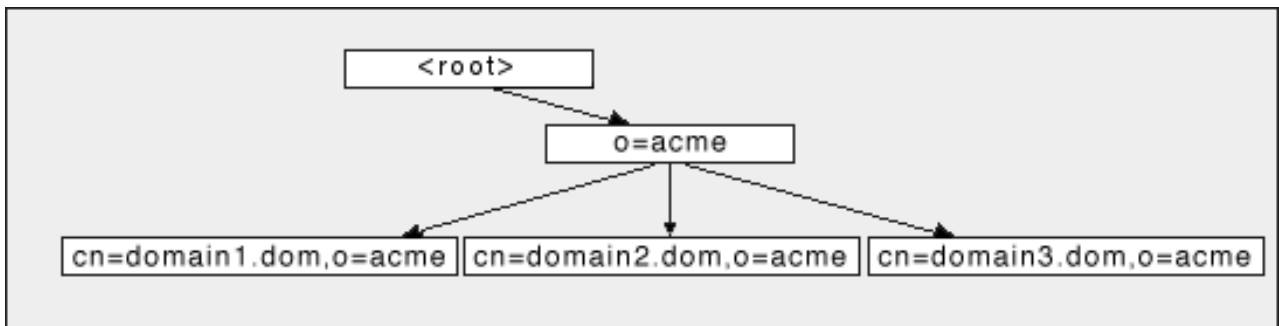
To search for accounts in these domain subtrees, LDAP clients should have the string

`cn=domainN.dom`

specified in their "Base Object" or "Search Base" settings.

Example:

BaseDN is `o=acme`, Domain RDN Attribute is `cn`, and three CommuniGate Pro domains (`domain1.dom`, `domain2.dom`, and `domain3.dom`) have been created:



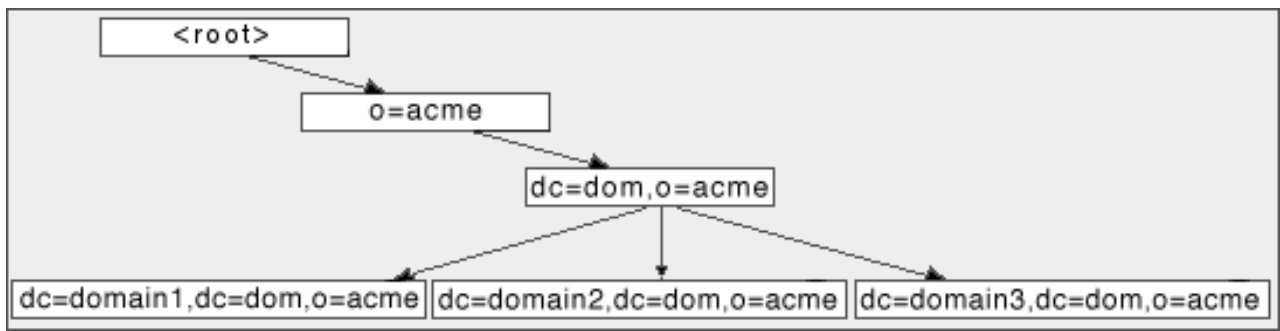
To search for accounts in these domain subtrees, LDAP clients should have the string

`cn=domainN.dom,o=acme`

specified in their "Base Object" or "Search Base" settings.

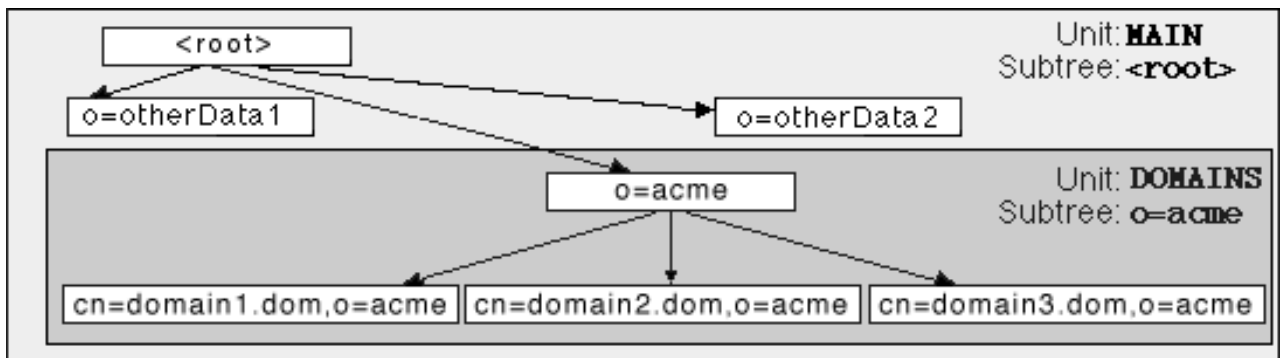
Example:

BaseDN is `o=acme`, Domain RDN Attribute is `dc`, and three CommuniGate Pro domains (`domain1.dom`, `domain2.dom`, and `domain3.dom`) have been created:



To search for accounts in these domain subtrees, LDAP clients should have the string
`dc=domainN,dc=dom,o=acme`
 specified in their "Base Object" or "Search Base" settings.

After you have decided how to organize your Domains Subtree, you can create additional Directory [Storage Units](#) to store your Domain and Account data in several units (if necessary). For example, if you want to use your CommuniGate Pro Directory Manager to store information not related to CommuniGate Pro Accounts, and you want all Domain and Account information to be stored either on a remote LDAP server or in a dedicated Local Storage Unit, you can create a Storage Unit `MyDomains` for the Directory Integration Base DN subtree (`o=acme` in the examples listed above). In this case, all Domains and Account records will be stored in that `MyDomains` Storage Unit (in a separate local unit or on a remote LDAP server), while all records that do not have the `o=acme` suffix will be stored in other Storage Units:



Note: If you change any Domain Subtree setting, the existing Subtree is not modified. Carefully select the proper values for the Domain Subtree settings before you start any Directory Integration activity. If you need to change these settings later, it is your responsibility to move the existing Domain Subtree to the new location (specified with the new BaseDN) and/or to change RDNs of the existing domain records (if you have changed the Domain RDN Attribute setting).

Custom Account Settings

The CommuniGate Pro Server has a predefined set of Account Settings (see the [Accounts](#) section for more details). The Directory Integration settings include a panel that allows you to specify additional, Custom settings for CommuniGate Pro Accounts:

Custom Account Settings

System	Public Info

You can use these Custom Account Settings to store additional information about your users: locations, phone numbers, demographic data, etc.

To add a Custom Setting, type its name into the last (empty) field and click the Update button.

Additional (custom) Account Settings are stored in Account Directory records (these records have the CommuniGateAccount Object Class).

When you select a name for a new Custom Account Setting, either use a name of an attribute already specified for CommuniGateAccount object class in the [Directory Schema](#), or use the Directory Integration [Attribute Renaming](#) feature and map the new Custom Account Setting name onto a name of any already specified attribute.

Example:

To add the `telephoneNumber` setting to all CommuniGate Pro accounts, add the name `telephoneNumber` to the Custom Account Settings table.

If a Local Storage Unit is used to store CommuniGate Pro Domains and Accounts subtree, no additional action is needed: the `telephoneNumber` attribute is already included into the CommuniGateAccount object class description in all Local Unit Schemas.

Example:

To add the `surname` setting to all CommuniGate Pro accounts, add the name `surname` to the Custom Account Settings table, and add the pair (`surname`, `sn`) to the Attribute Renaming table, so the `surname` Account Settings will be stored in Directory records as `sn` attributes.

If a Local Storage Unit is used to store CommuniGate Pro Domains and Accounts subtree, no additional action is needed: the `sn` attribute is already included into the CommuniGateAccount object class description in all Local Unit Schemas.

Example:

To add the `BirthDay` setting to all CommuniGate Pro accounts, add the name `BirthDay` to the Custom Account Settings table.

If a Local Storage Unit is used to store CommuniGate Pro Domains and Accounts subtree, add the `BirthDay` attribute to the Local Unit Schema, and add the newly created `BirthDay` attribute name to the list of Optional Attributes of the CommuniGateAccount object class.

If a Remote Storage Unit is used to store CommuniGate Pro Domains and Accounts subtree, update the Directory Schema on the remote LDAP server to allow directory records of the CommuniGateAccount object class to include the `BirthDay` attribute.

Note: account records in the Directory always contain the `sn` attribute to make them compatible with the standard LDAP Directory Schema. If you do not include this attribute into the Custom Account Settings set, CommuniGate Pro stores account records with the `sn` attribute containing an empty string.

After you have specified some Custom Account Settings, their names appear on the Account Settings pages. You can use those pages or [CLI](#) to add and update the Custom Setting values for all CommuniGate Pro Accounts:

Real Name: surname: city: CommuniGate Password: telephoneNumber:	
--	--

Note: if you rename a custom attribute name or remove it, the attribute values are not modified in the Directory - you are effectively changing the Directory Integration parameters, not the Directory data itself. To update the actual Directory data (for example, to remove all `telephoneNumber` attribute values from the Directory), use LDAP utilities and/or applications.

Integrating Regular Domains

Regular CommuniGate Pro Domains do not rely on Directory data. All Domain and Account settings are stored in files inside the CommuniGate Pro file directories and CommuniGate Pro Server reads those files when it needs to retrieve Domain and Account settings. Regular Domains can store copies of **some** Account Settings in Directory records.

The directory record for a Regular Domain is created when the Server needs to store a directory record for any object in that Domain. For example, when the Server needs to create a directory record for the Account `john` in the `dom1.dom` Domain, it creates the `cn=dom1.dom` record first (if it does not exist), and then the Server creates the `uid=john,cn=dom1.dom` record for the Account `john`.

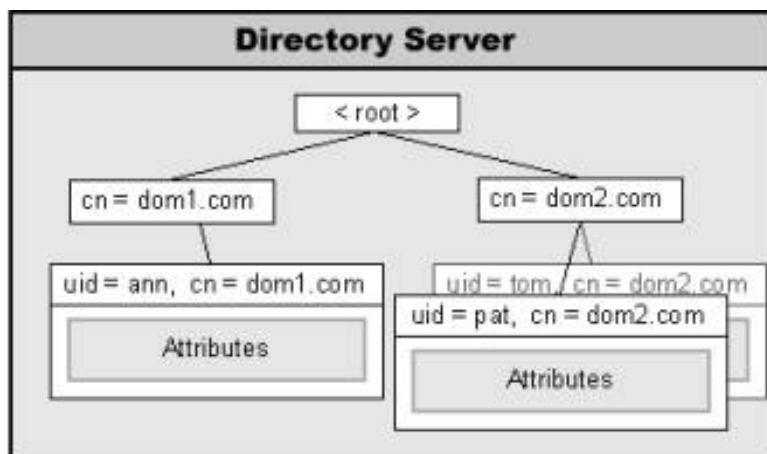
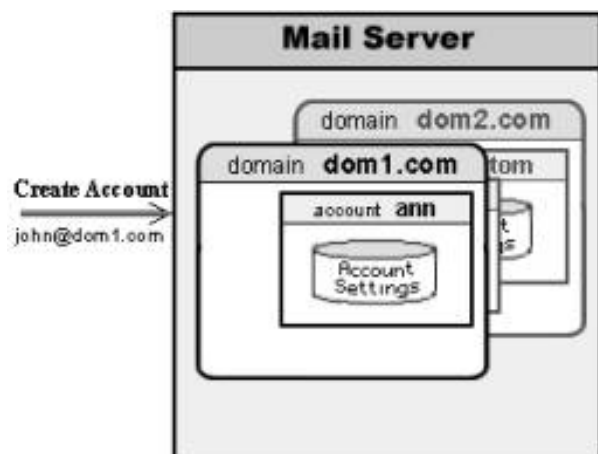
When the [Directory Integration](#) Domain Setting is set to `Keep In Sync`:

- A directory record is created for each object (Account, Group, Mailing List, Forwarder) created in that Domain.
- A directory record is removed when an object is removed from that Domain.
- Directory record DNs are renamed when Domain objects are renamed.
- Directory records are updated when Domain Accounts settings are updated.

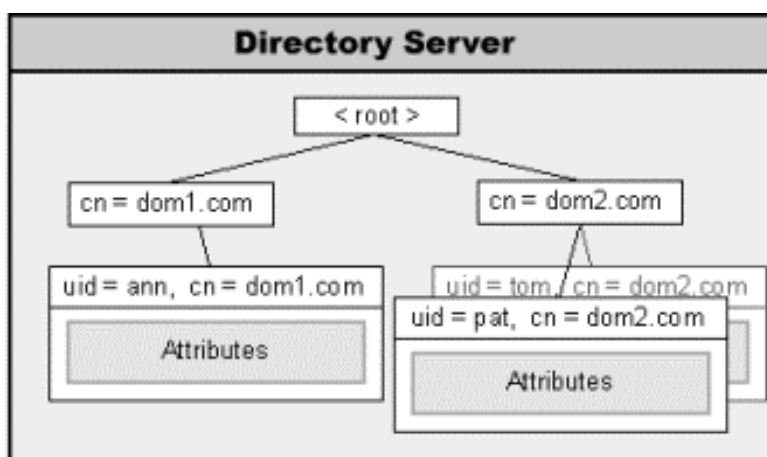
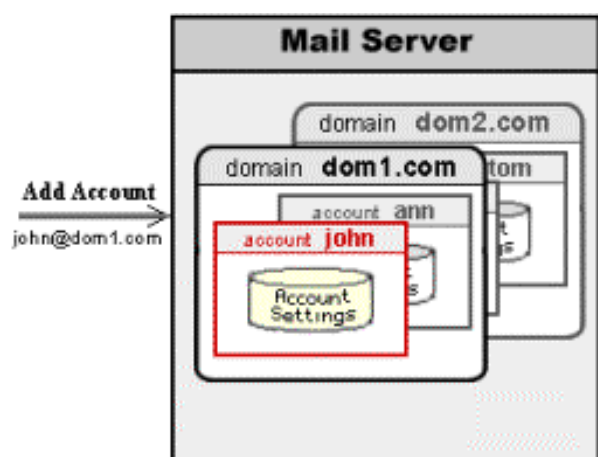
None of these actions takes place when the Domain Directory Integration settings is set to `Disabled`.

The following diagram illustrates what happens when the `dom1.com` Domain has the Directory Integration option set to `Keep In Sync`, and an account is created in that domain:

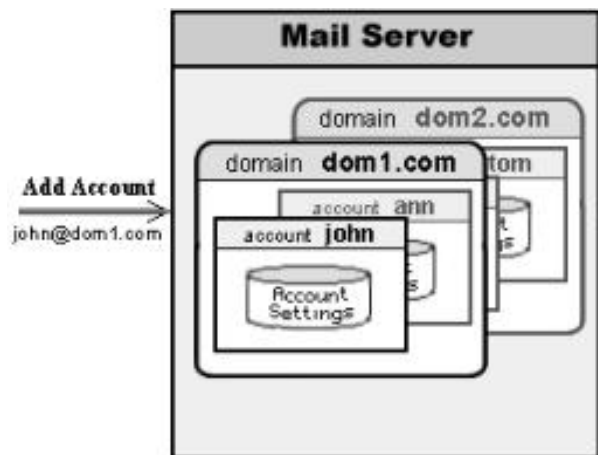
Step 1



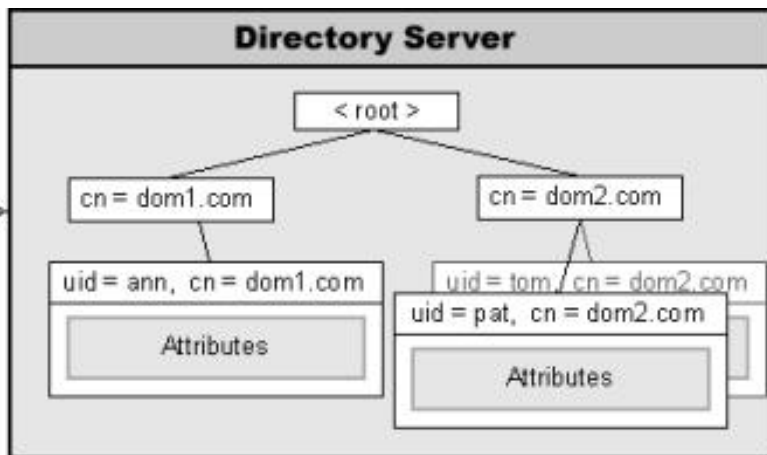
Step 2



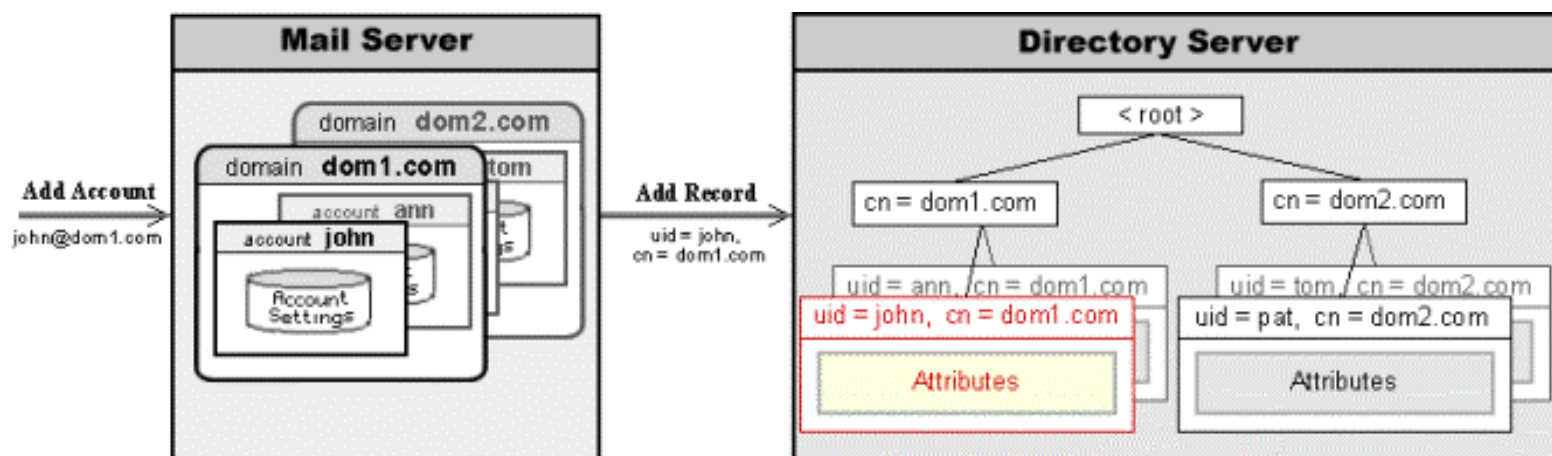
Step 3



Add Record
uid = john,
cn = dom1.com



Step 4



In this example:

- The WebAdmin or CLI interface is used to create the **john** Account in the **dom1 . com** Domain.
 - The Account **john** is created, and the supplied settings (together with the Account Template) are used to compose the initial Account Settings.
 - The Account Manager executes the **AddRecord** Directory operation to create a record in the Directory. The Directory record DN is composed using the global Directory Integration Settings. If the default settings are used, the Directory record DN is **uid=john ,cn=dom1 . dom**.
- Some of the initial Account Settings are converted into Directory attributes and stored into the newly created Directory record.

Now Directory search operations (initiated with an LDAP client or the WebUser Interface) can display the record for the newly created account.

Directory records for Regular Domain Accounts contain the following attributes:

- **uid** - the Account name
- **cn** - the Account "Real Name"
- **sn** - an empty string if this attribute is not included into Custom Account Settings
- **hostServer** - the main domain name of the CommuniGate Pro Server that hosts this Account
- custom attributes (see above).
- **userPasword** - the Account password (*optionally*).
- other standard settings - (*optionally*).

The Regular Domains panel located on the Directory Integration page of the WebAdmin Interface allows you to specify these options:

Regular Domains	
Copy into Account records:	Passwords Standard Settings

Copy Password

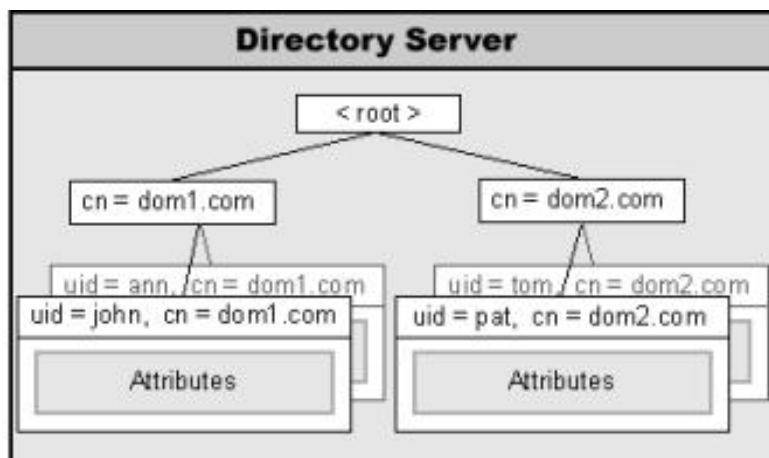
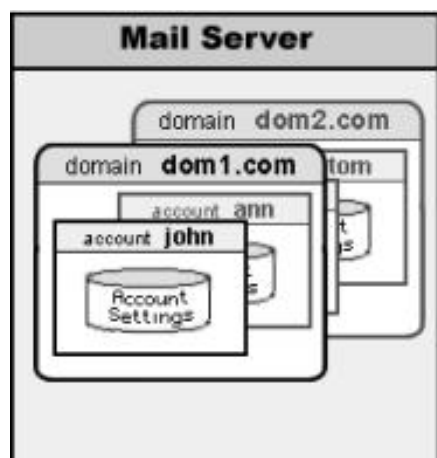
If this option is selected, the Directory records for Regular Domain Accounts will contain the Account Password attribute (usually it is renamed into the userPassword attribute).

Copy Standard Settings

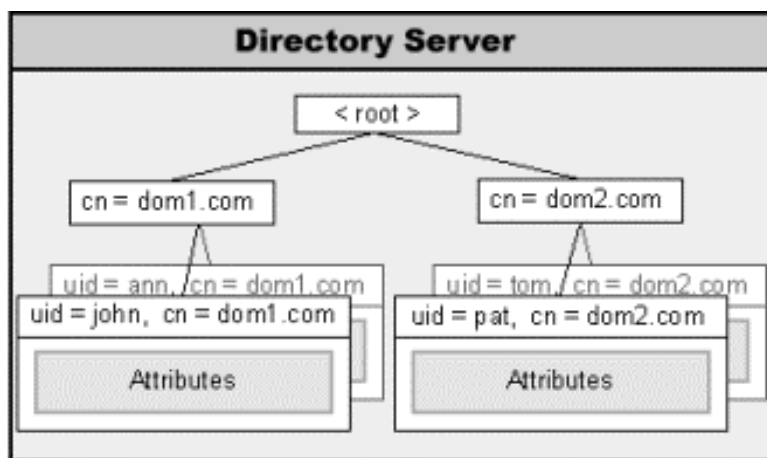
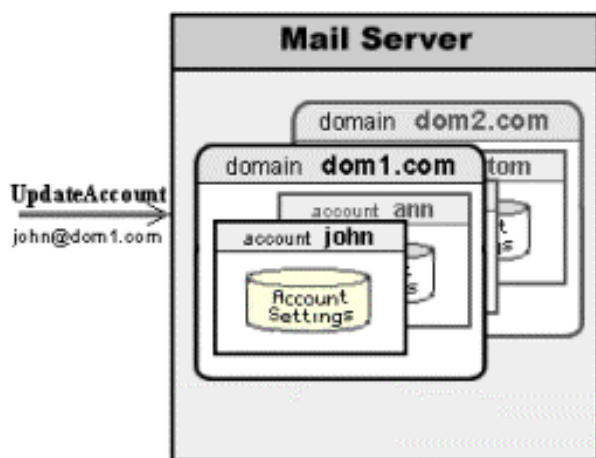
If this option is selected, the Directory records for Regular Domain Accounts will contain all CommuniGate Pro standard Settings for those Accounts (excluding the RealName setting that is always being stored and the Password setting that it controlled using a separate option).

The following diagram illustrates what happens when the dom1 . com Domain has the Directory Integration option set to Keep In Sync, and Domain Account settings are updated:

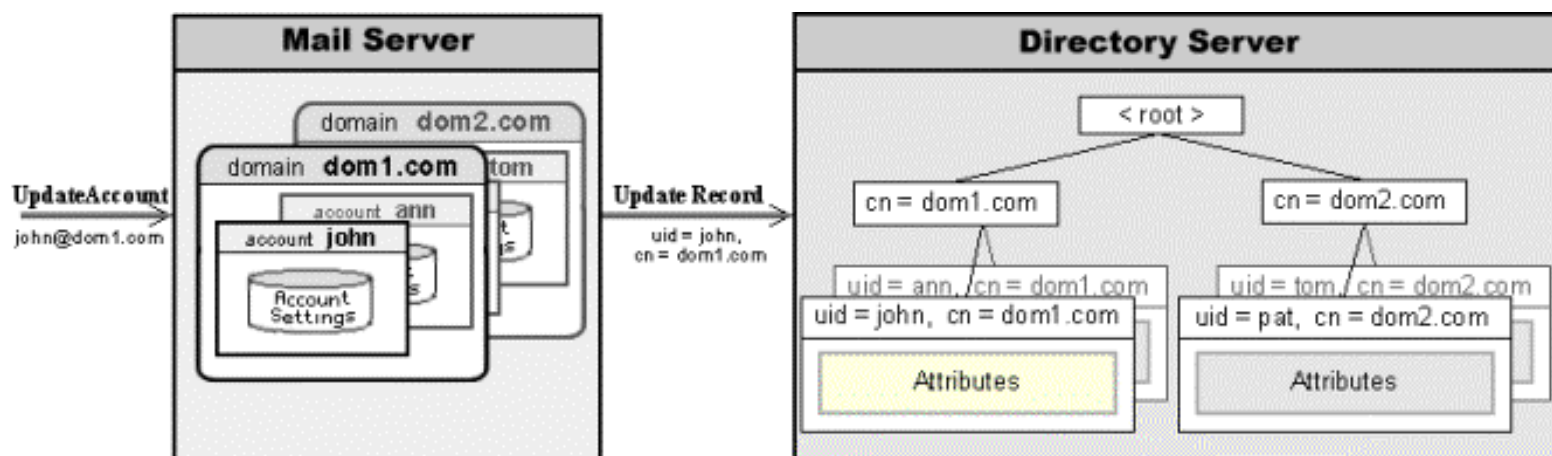
Step 1



Step 2



Step 3



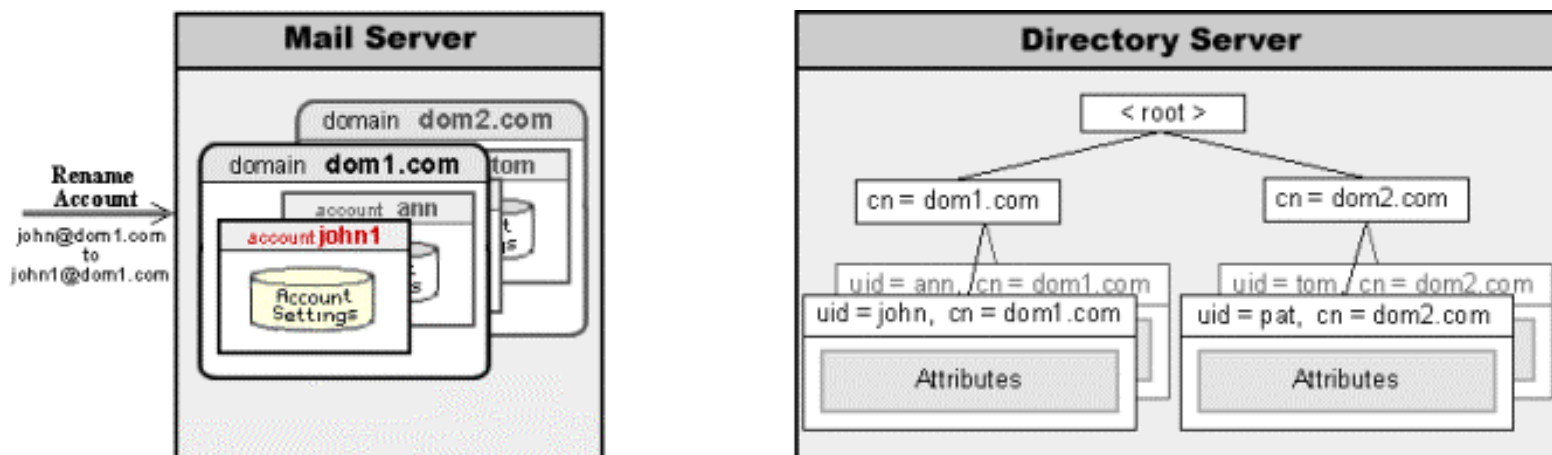
In this example:

- The UpdateAccount operation is initiated with a WebAdmin or CLI Interface command.
- The Account john Settings are modified using the supplied new settings and the updated settings are stored in the CommuniGate Pro Account data files.
- If the dom1 . com Domain Directory Integration setting is set to Keep In Sync, the Account Manager executes the UpdateRecord Directory operation to update the Account record in the Directory.

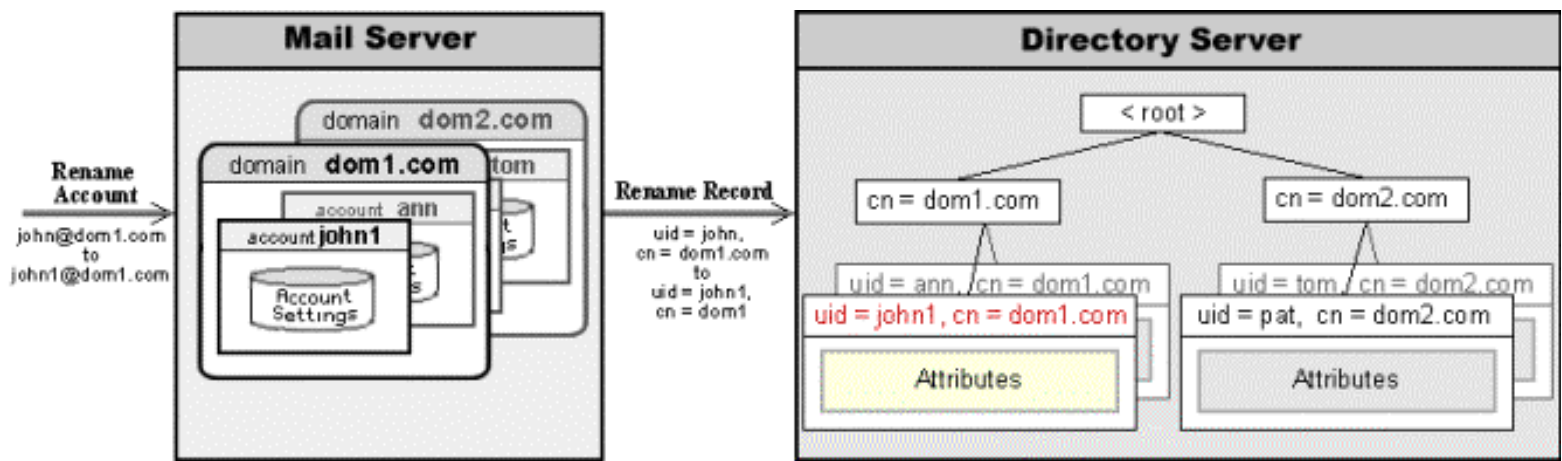
Note: It is important to understand that Directory Integration for Regular Domains is a one-way relationship: if you change attributes of Account records in the Directory (using any LDAP utility), the actual Account Settings will not be modified - CommuniGate Pro always uses data in the settings files, and never reads data from the Directory when it needs to retrieve settings for Regular Domains or settings for Accounts in those Domains. The CommuniGate Pro Manager for regular Domains and Accounts only updates the Directory, but it never reads the Account record data back from the Directory.

The following diagram illustrates what happens when the dom1 . com Domain has the Directory Integration option set to Keep In Sync, and a Domain Account is renamed:

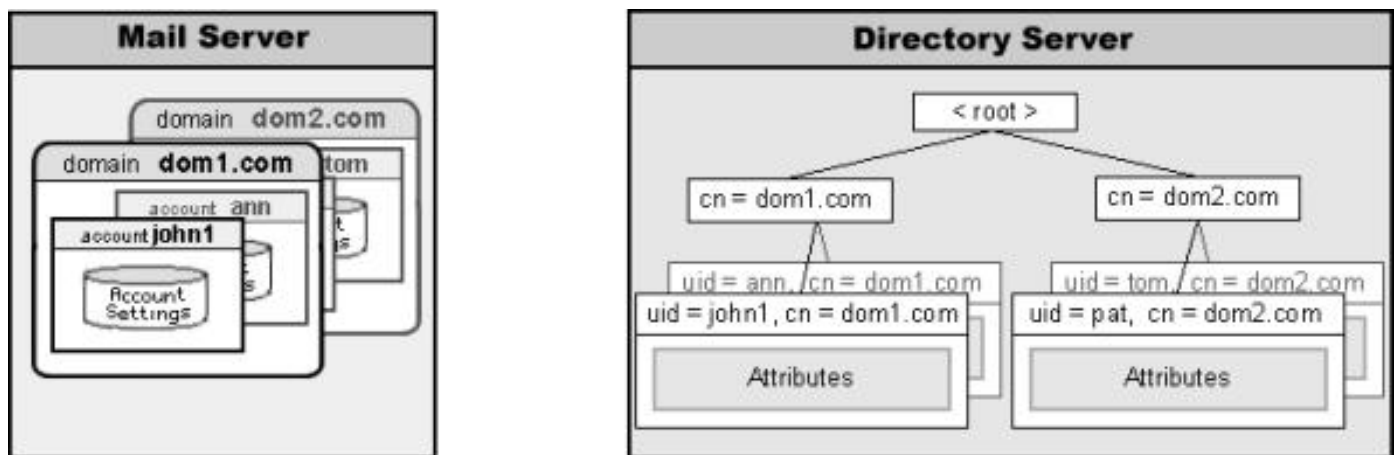
Step 1



Step 2



Step 3

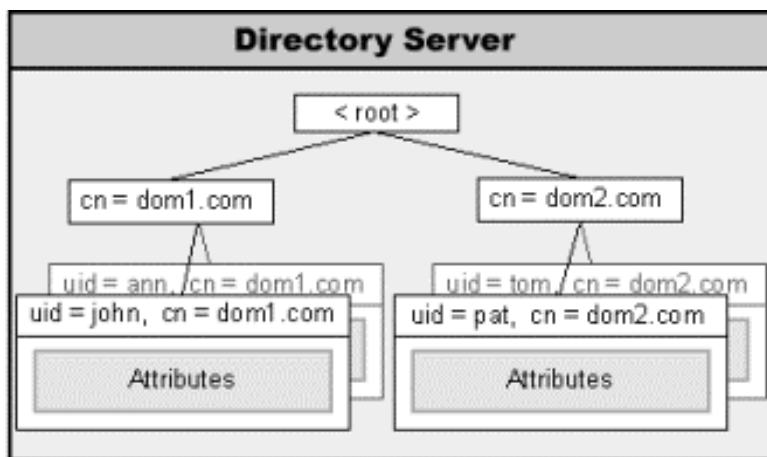
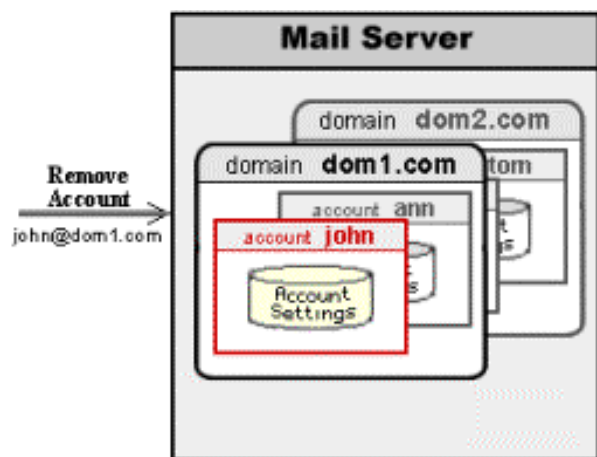


In this example:

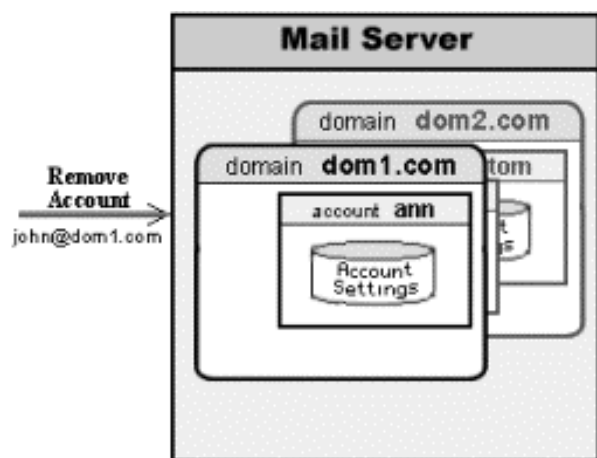
- The RenameAccount operation is initiated with a WebAdmin or CLI Interface command.
- The Account john and its files are renamed.
- If the dom1 . com Domain Directory Integration setting is set to Keep In Sync, the Account Manager executes the RenameRecord (modifyDN) Directory operation to rename the Account record in the Directory.

The following diagram illustrates what happens when the dom1 . com Domain has the Directory Integration option set to Keep In Sync, and a Domain Account is removed:

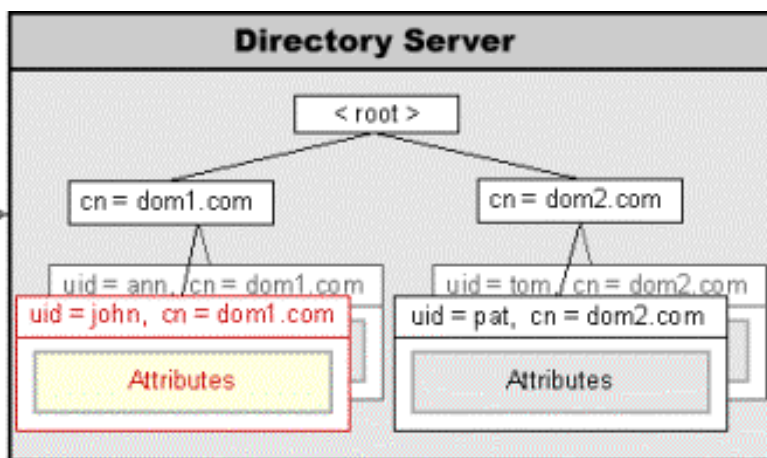
Step 1



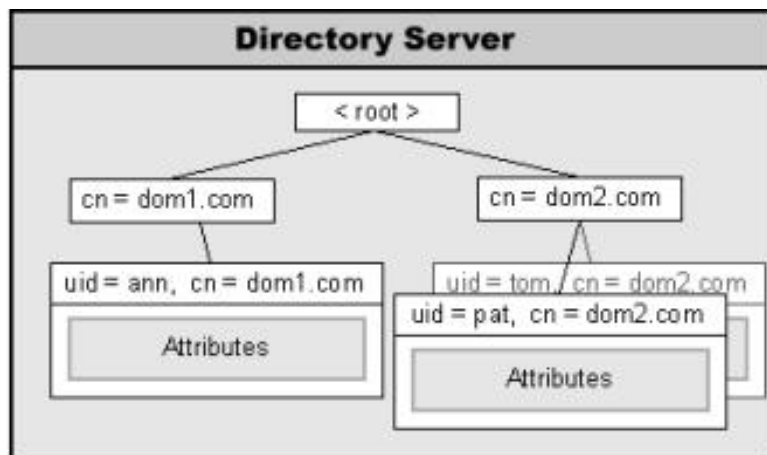
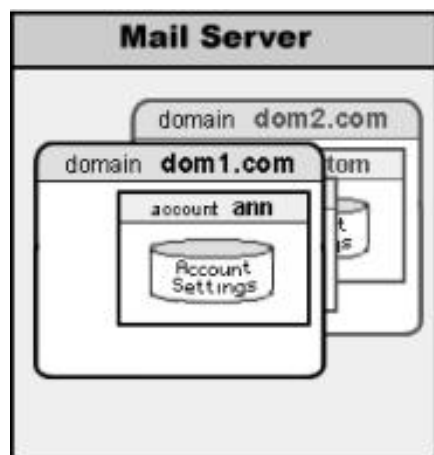
Step 2



Remove Record
uid = john,
cn = dom1.com



Step 3



In this example:

- The RemoveAccount operation is initiated with a WebAdmin or CLI Interface command.
- The Account john and its files are removed.
- If the dom1 . com Domain Directory Integration setting is set to Keep In Sync, the Account Manager executes the DeleteRecord Directory operation to remove the Account record from the Directory.

The Directory Integration panel on the Domain Settings page has the `Delete All` button. Use this button to delete the Domain record and all Domain Object records from the Directory. The operation deletes only those records that contain the `hostServer` attribute and that attribute value is the same as the Main Domain name of this CommuniGate Pro Server.

The Directory Integration panel on the Domain Settings page has the `Insert All` button. Use this button to create a directory record for this Domain and to create directory records for all Domain Objects.

Note: If you have created several Accounts in the regular Domain when its Directory Integration setting was set to Disabled, the Directory does not contain records for those Accounts. If later you switch that setting to `Keep In Sync`, you will see error reports when you try to rename, remove, or update those Accounts: the Server tries to update the Directory records for those Accounts, but the Directory records do not exist.

Before you switch the Directory Integration setting from Disabled to `Keep In Sync`, click the `Delete All` button, and then click `Insert All` button to synchronize the Directory and the current Domain Objects set.

LDAP-based Provisioning

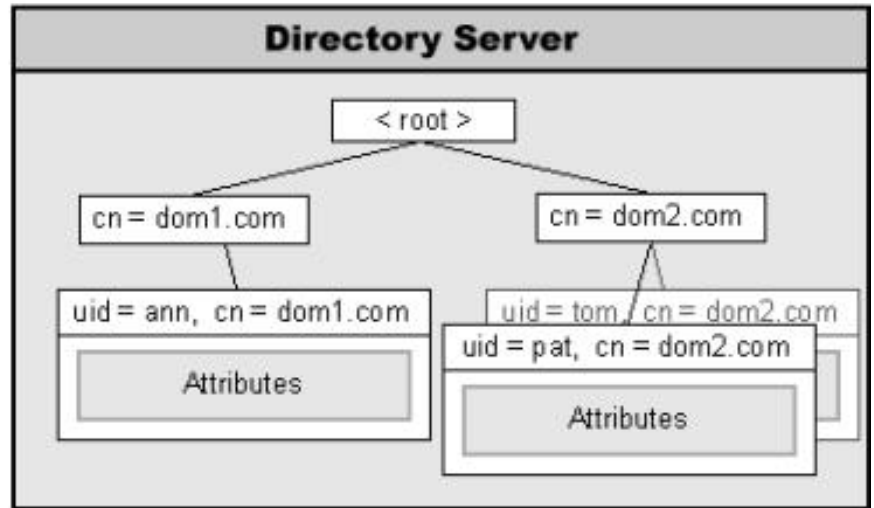
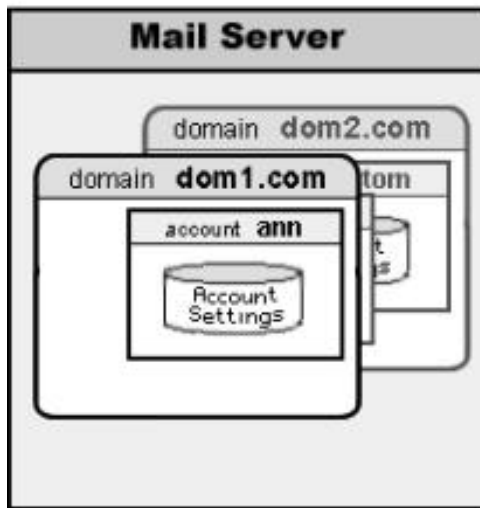
CommuniGate Pro can allow you to use the LDAP protocol for to create, update, rename, and remove accounts:

LDAP direct Provisioning

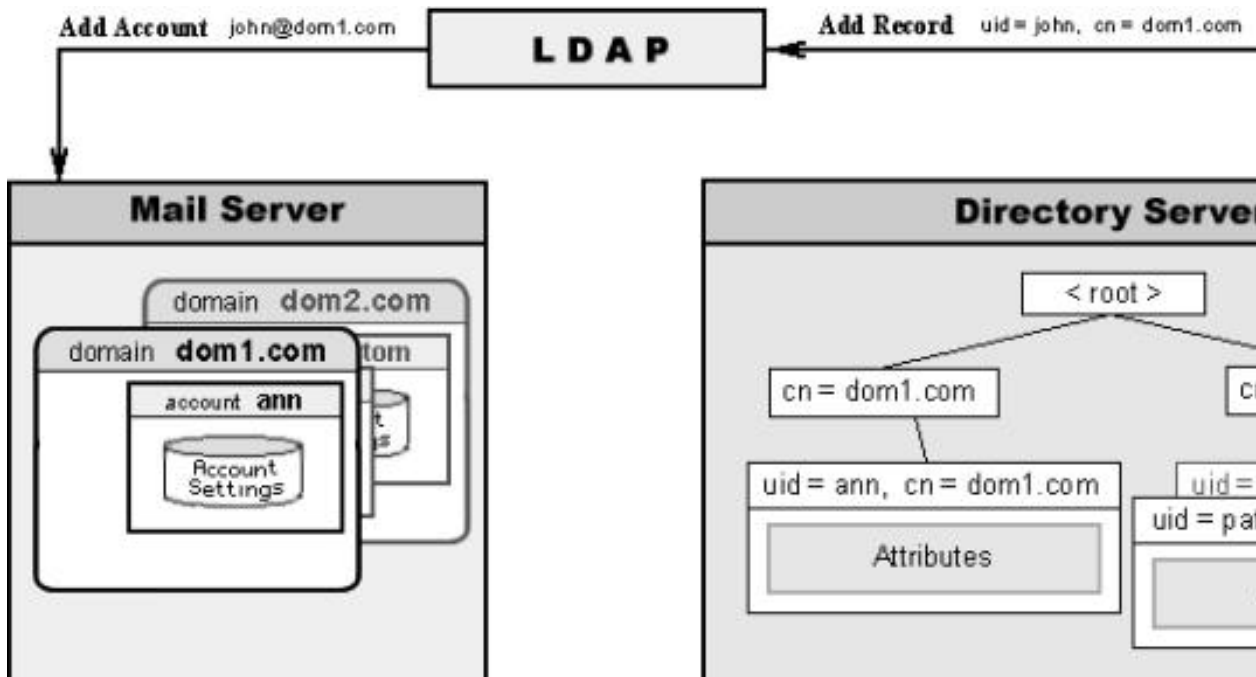
When this option is enabled, the LDAP module check the names (DNs) specified in the update operations. If the DN looks like a DN of a CommuniGate Pro Account, the LDAP module does not perform the request operation with the Directory. Instead, it converts executes the `CreateAccount`, `UpdateAccount`, `RenameAccount`, or `RemoveAccount` operations for the specified Account and Domain.

The diagram below illustrates how the LDAP `AddRecord` operation works in this case:

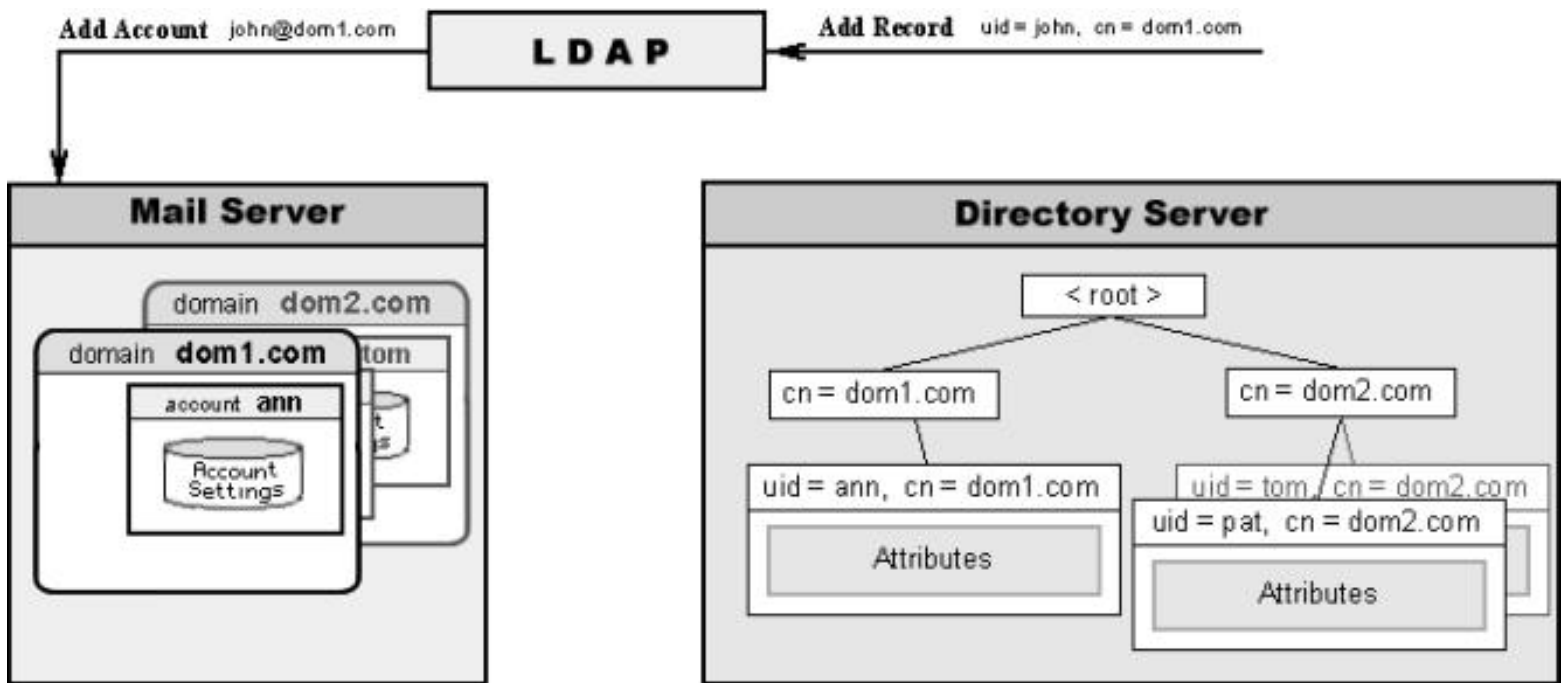
Step 1



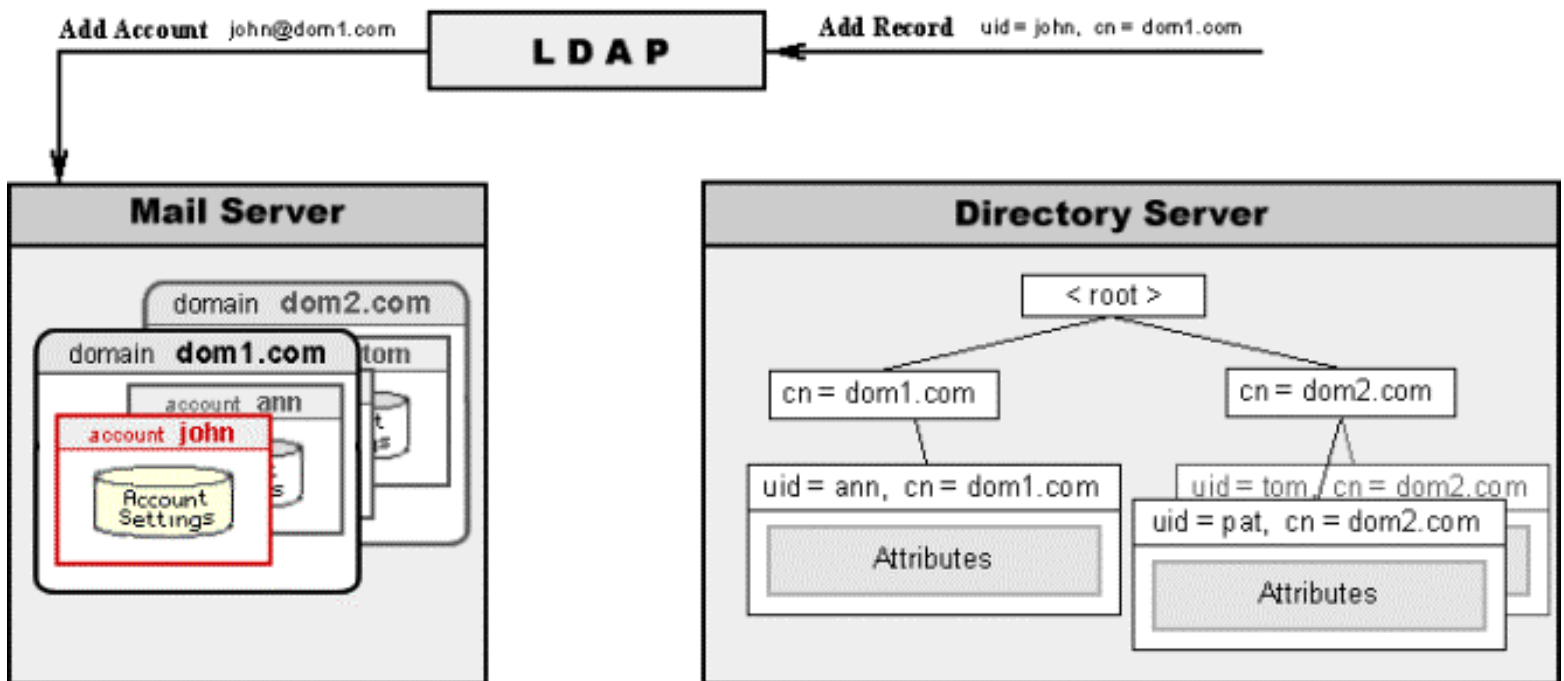
Step 2



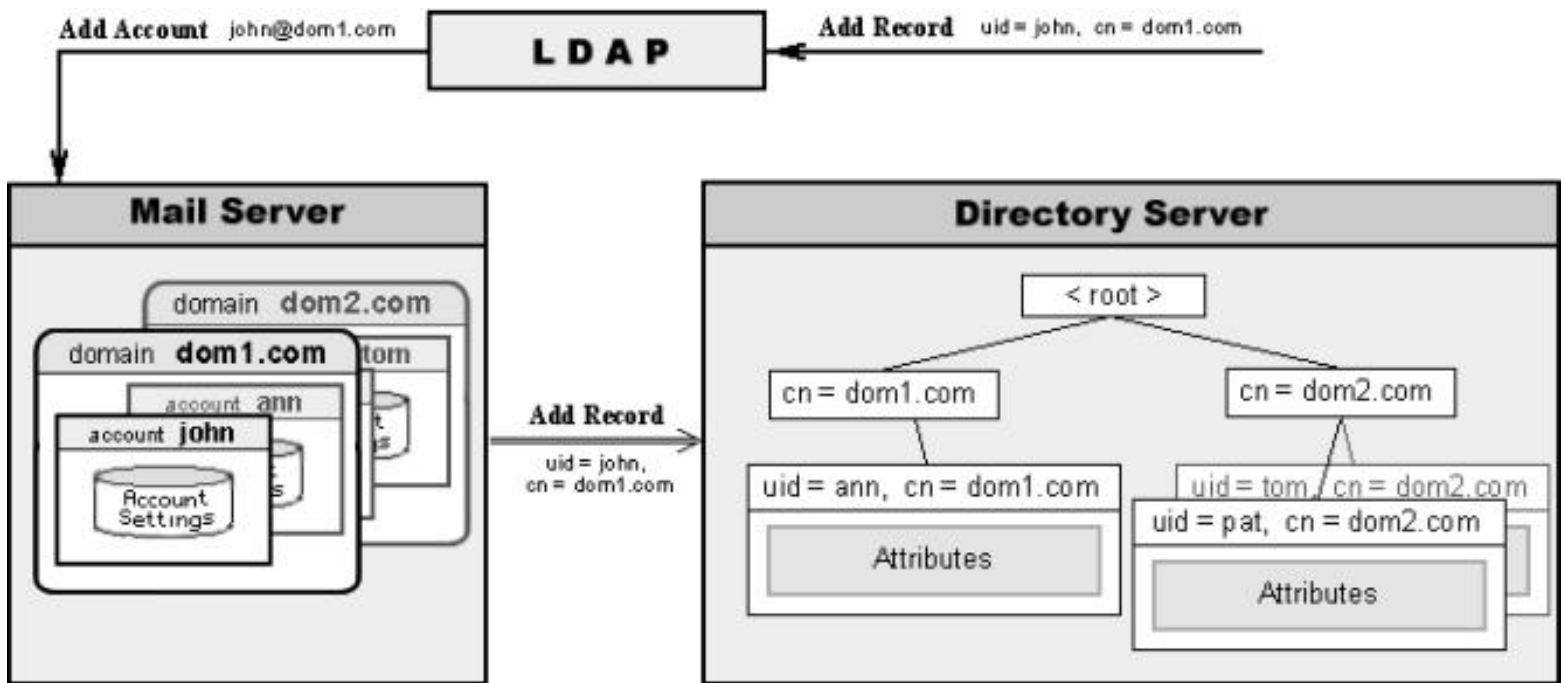
Step 3



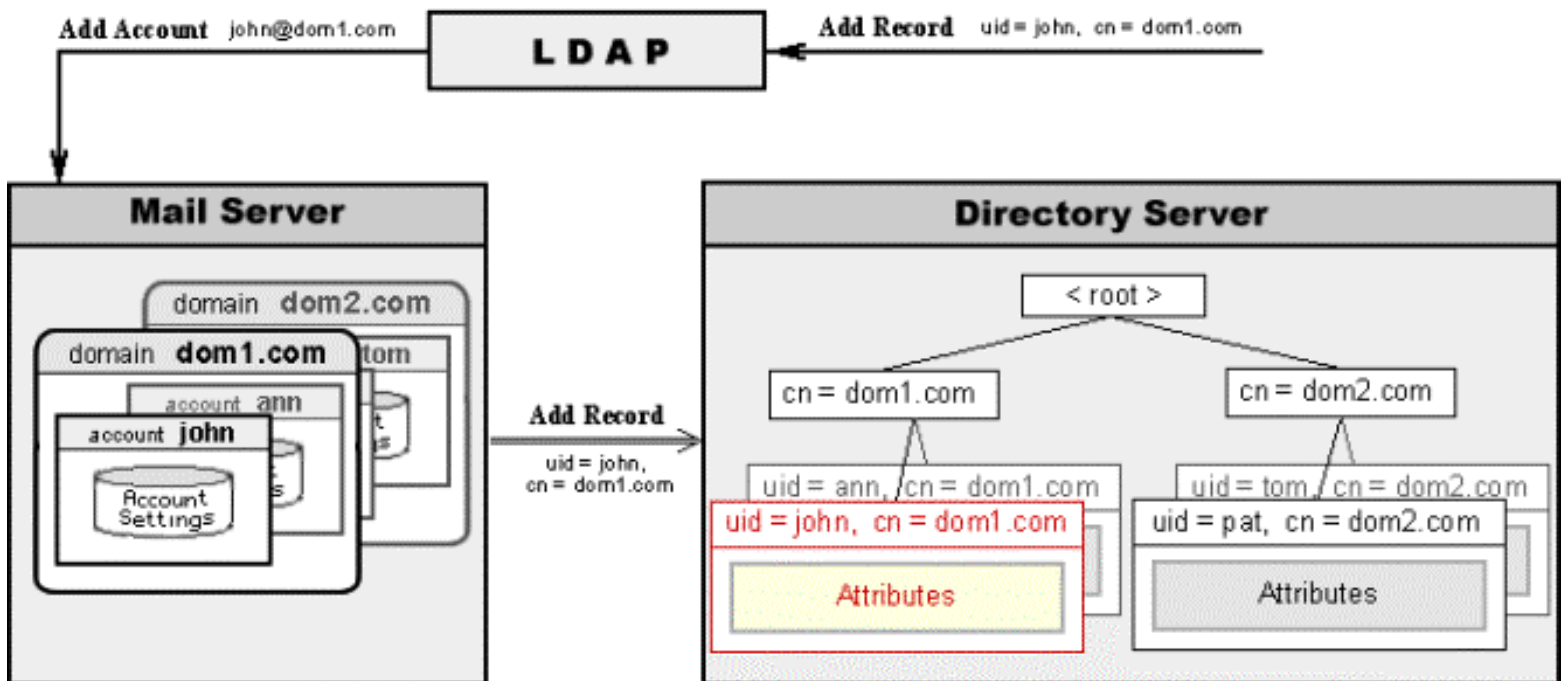
Step 4



Step 5

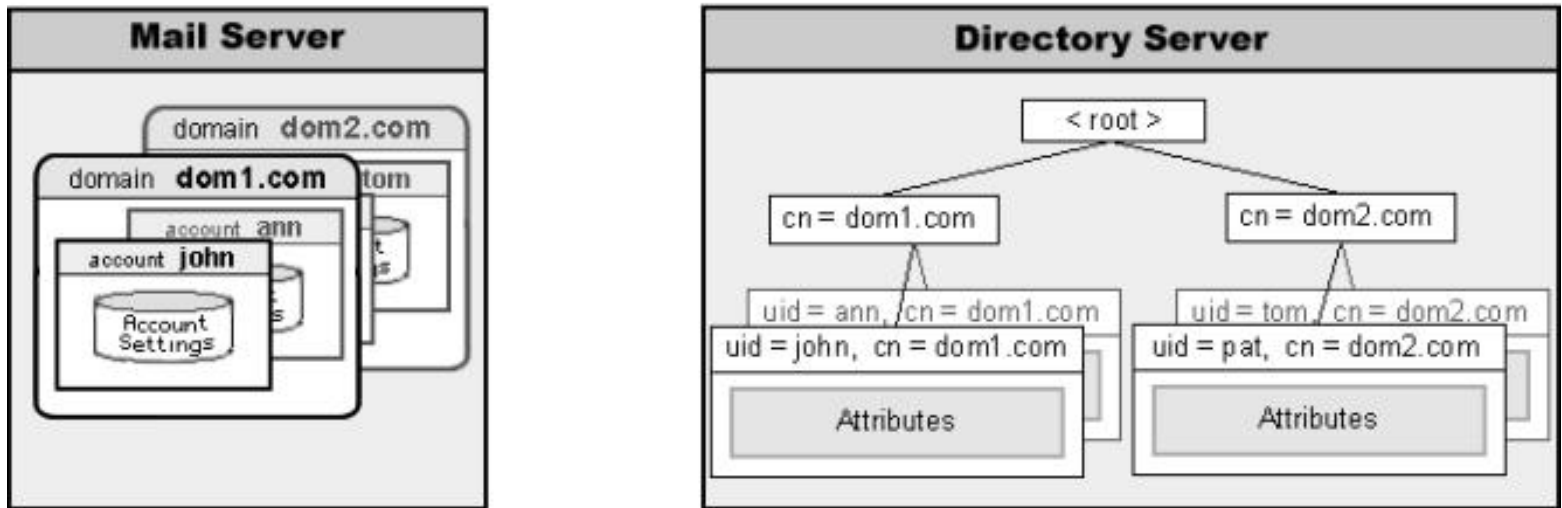


Step 6



Step 7

LDAP



In this example:

- The LDAP module received an AddRecord request from an LDAP client. The client asks the LDAP module to create the record with the `uid=john, cn=dom1.com` DN.
- The LDAP module checks the DN and sees that it looks like the DN record for the CommuniGate Pro Account `john` in the domain CommuniGate Pro Domain `dom1.com`, and the CommuniGate Pro Domain `dom1.com` does exist.
- Instead of performing the requested AddRecord operation with the Directory, the LDAP module executes the `CreateAccount(john)` operation in the `dom1.com` Domain.
- The Account `john` is created, and the supplied LDAP attributes (together with the Account Template) are used to compose the initial Account Settings.
- If the `dom1.com` Domain Directory Integration setting is set to Keep In Sync, the Account Manager executes the AddRecord Directory operation to create a record in the Directory.

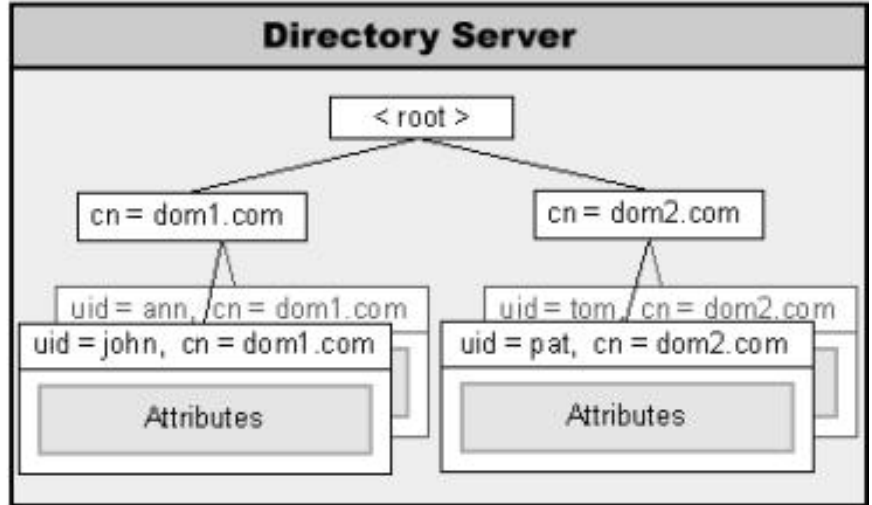
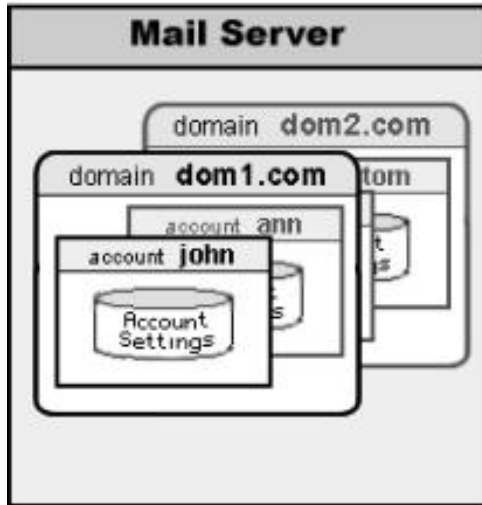
Note: the Directory Integration settings are used to convert LDAP record attribute names into the CommuniGate Pro attribute names. For example, the LDAP AddRecord request can contain the `cn` attribute. This attribute is stored in the Account settings as the Account RealName setting. When the Account Manager adds a record to the Directory, it converts the RealName Account setting back into the `cn` record attribute.

Note: all LDAP AddRecord request attributes will be stored as the Account Settings. But only the attributes specified with the Directory Integration parameters will be copied into the new Directory record. The Directory record will also contain the attributes not included into the original LDAP AddRecord request, but specified in the Account Template.

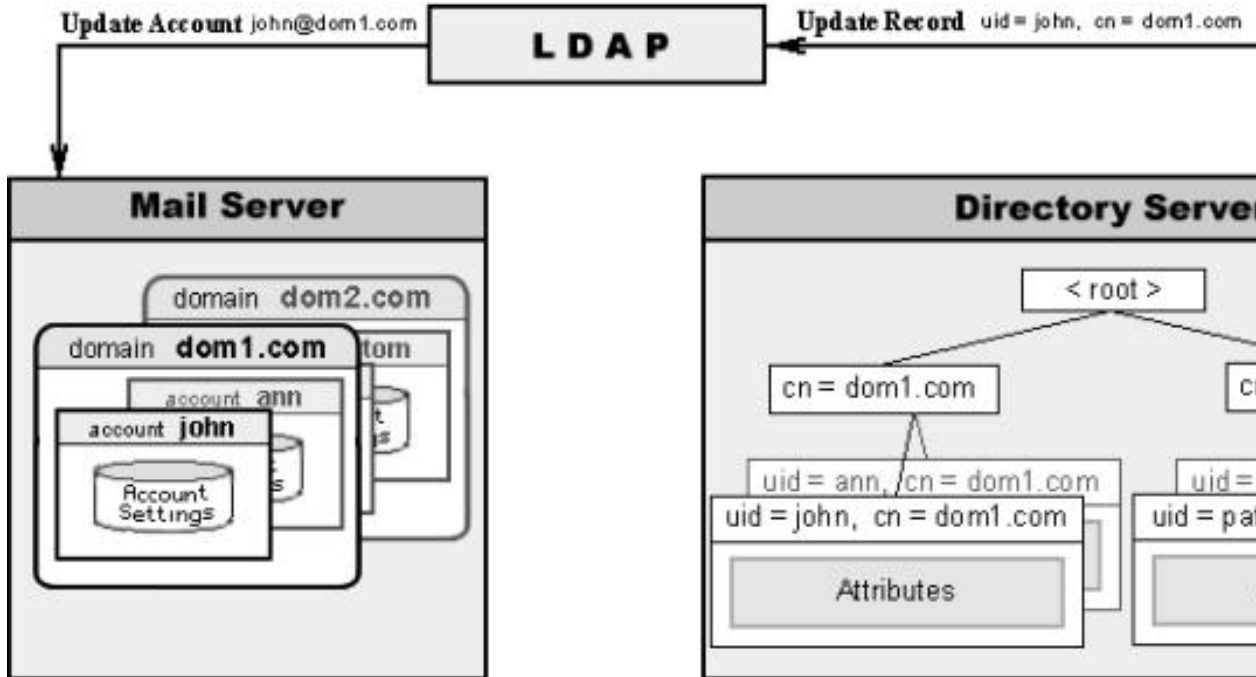
Note: the LDAP Provisioning feature detects the `unixPassword` attributes and converts them into Password settings after adding a leading `0x02` byte. See the [Account Import](#) section for the details.

The following diagram illustrates how the LDAP ModifyRecord operation can be used to modify Account Settings:

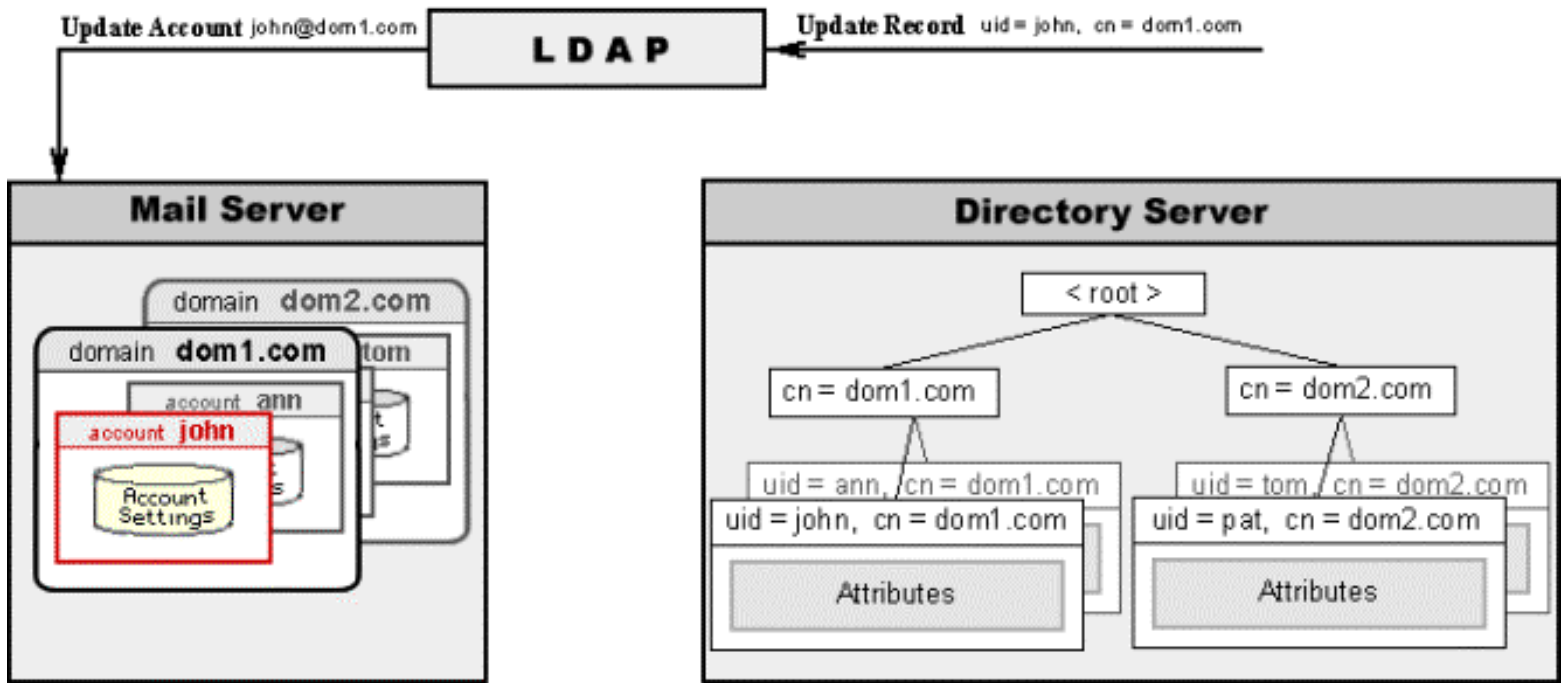
Step 1



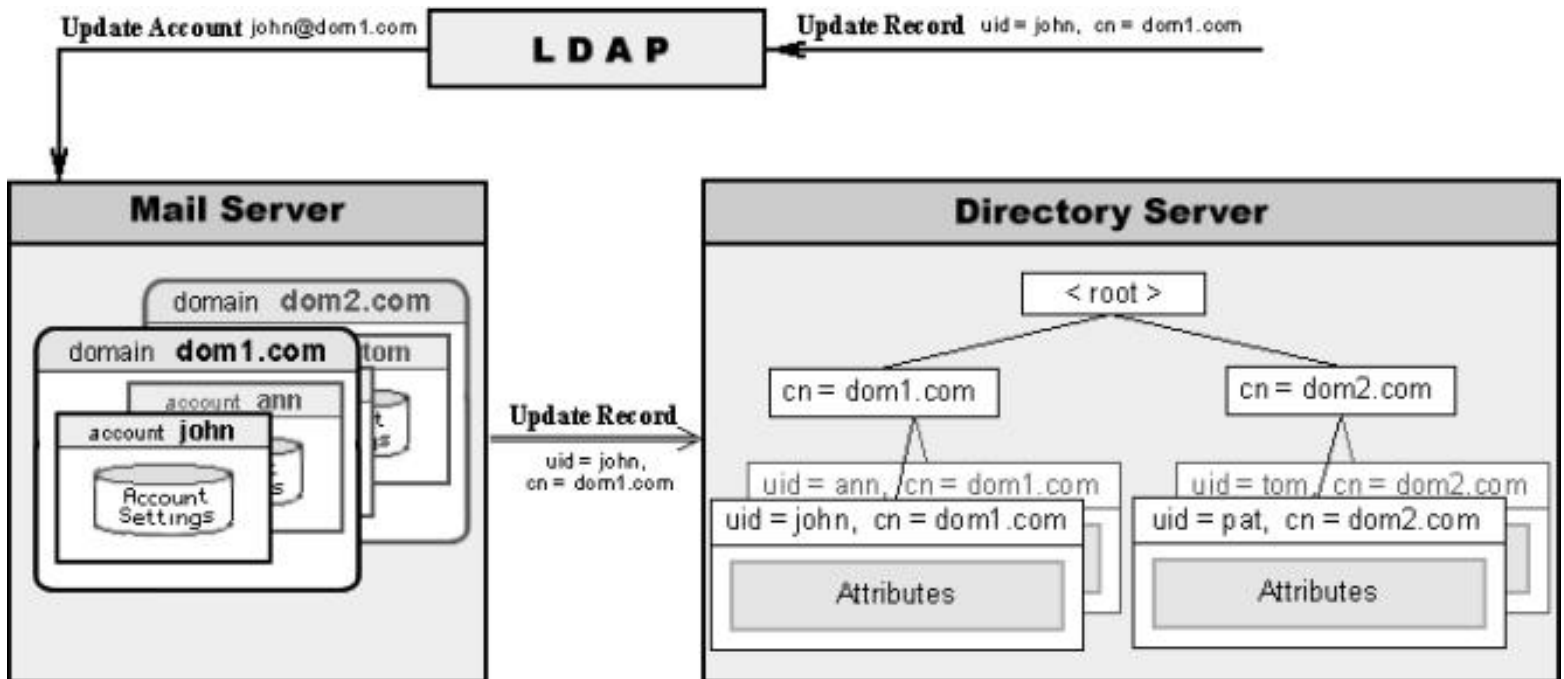
Step 2



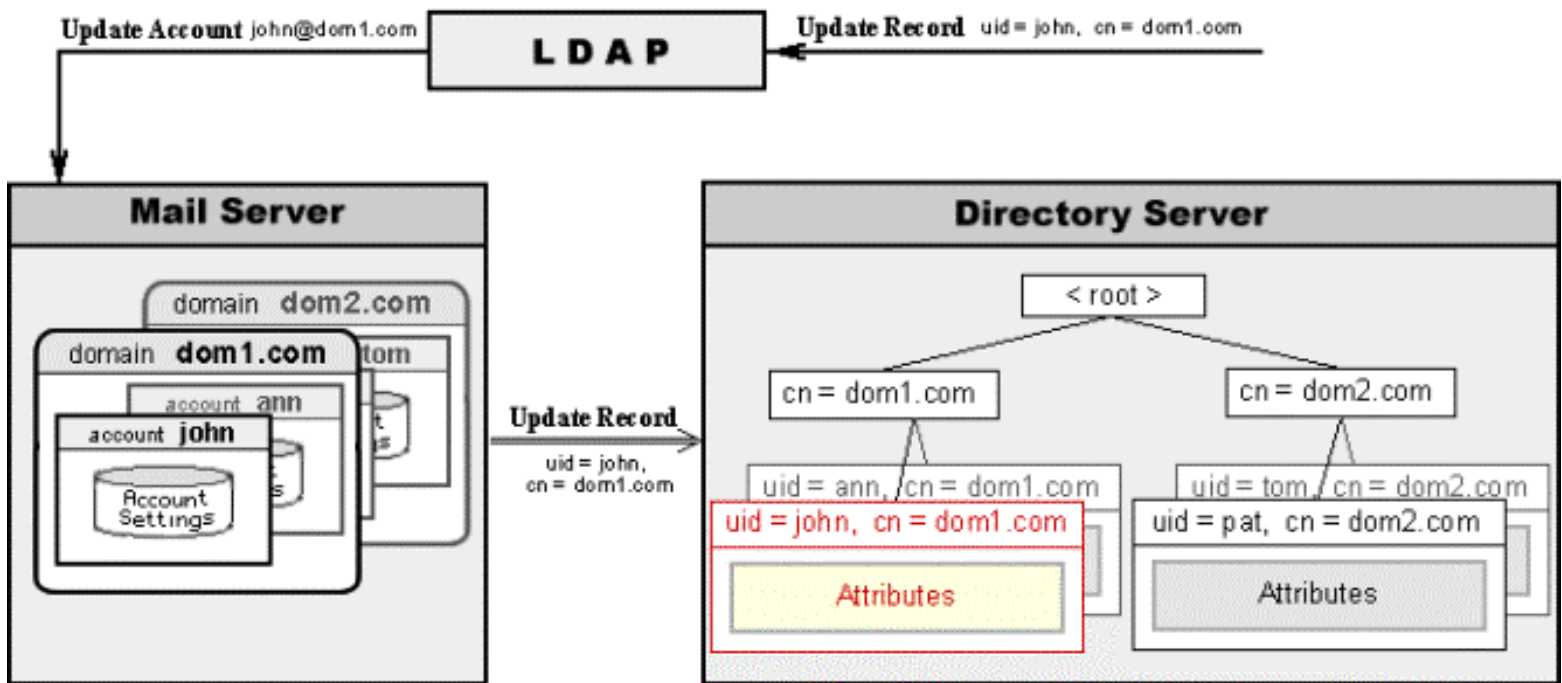
Step 3



Step 4



Step 5

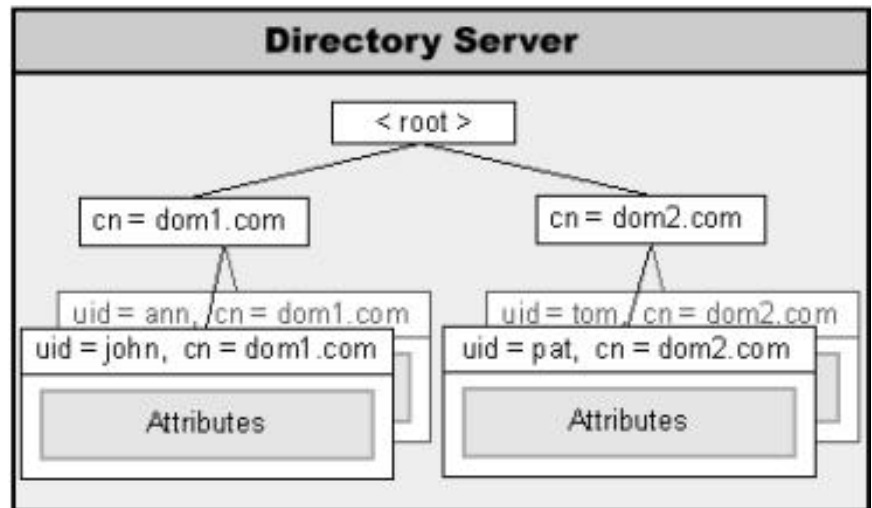
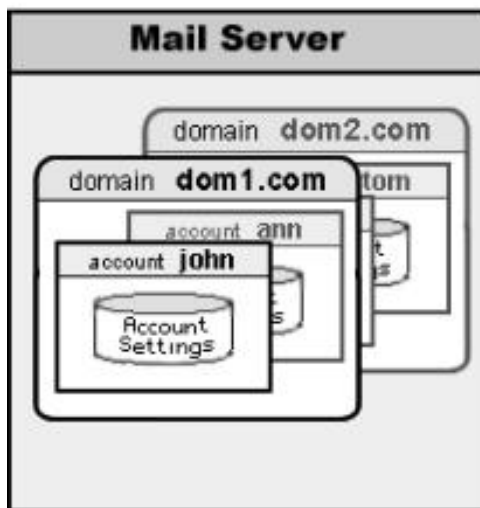


In this example:

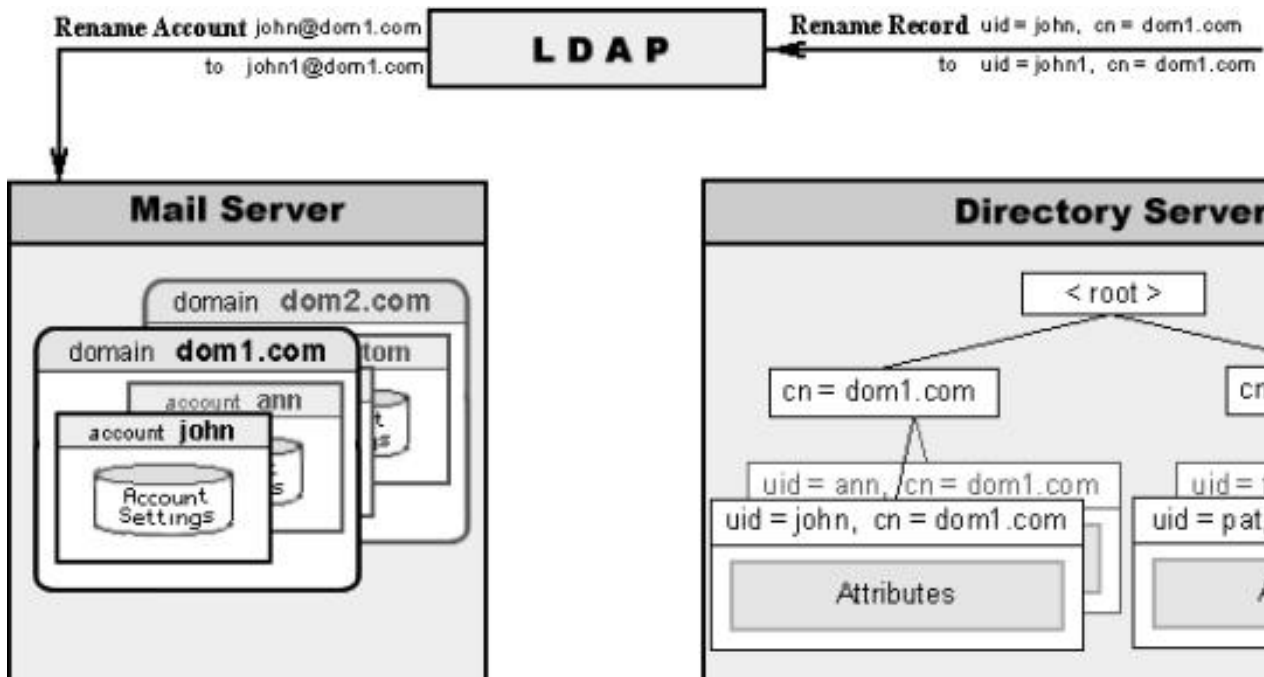
- The LDAP module received a ModifyRecord request from an LDAP client. The client asks the LDAP module to update the record with the `uid=john, cn=dom1.com` DN.
- The LDAP module checks the DN and sees that it looks like the DN record for the CommuniGate Pro Account `john` in the domain CommuniGate Pro Domain `dom1.com`, and the CommuniGate Pro Domain `dom1.com` does exist.
- Instead of performing the requested ModifyRecord operation with the Directory, the LDAP module executes the `UpdateAccount(john)` operation in the `dom1.com` Domain.
- The Account `john` Settings are modified using the supplied LDAP attributes.
- If the `dom1.com` Domain Directory Integration setting is set to Keep In Sync, the Account Manager executes the ModifyRecord Directory operation to create a record in the Directory.

The following diagram illustrates how the LDAP ModifyDN operation can be used to rename Accounts:

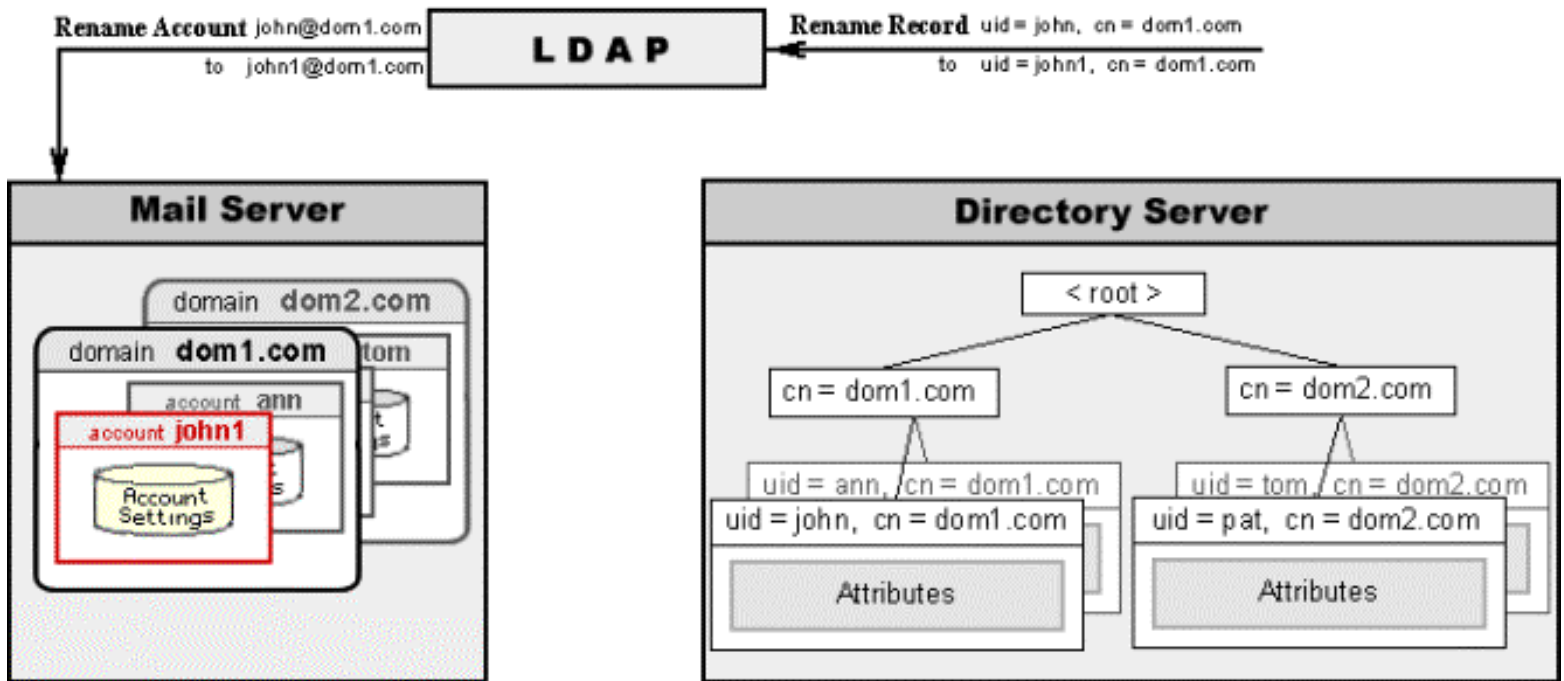
Step 1



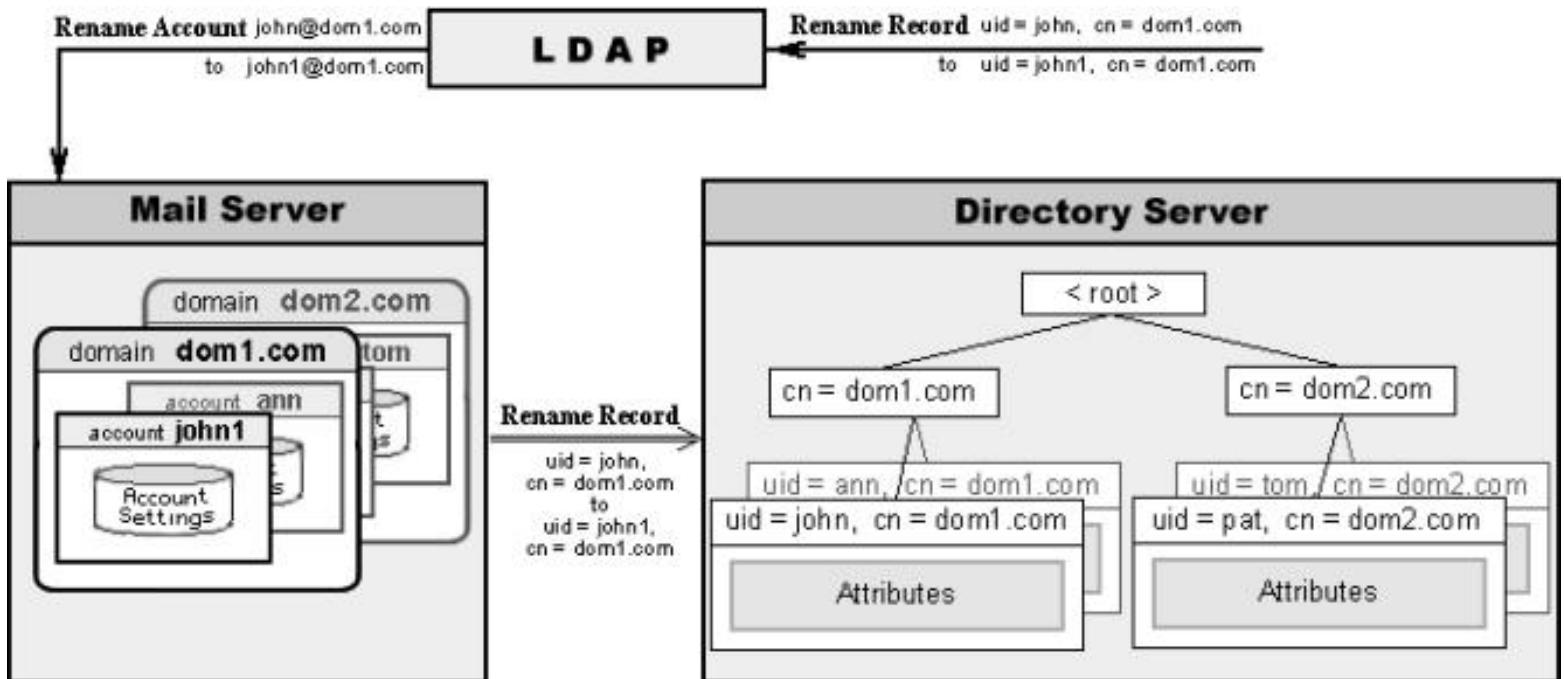
Step 2



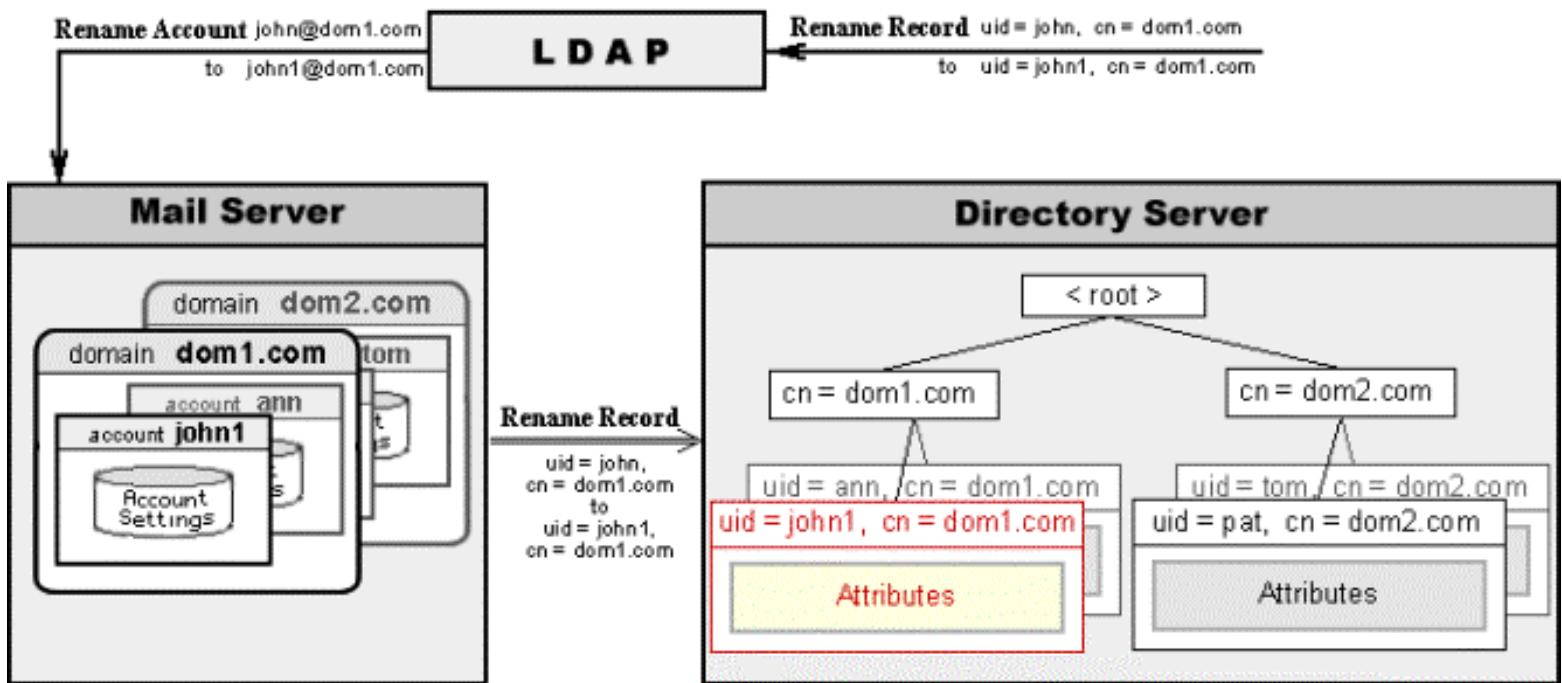
Step 3



Step 4



Step 5

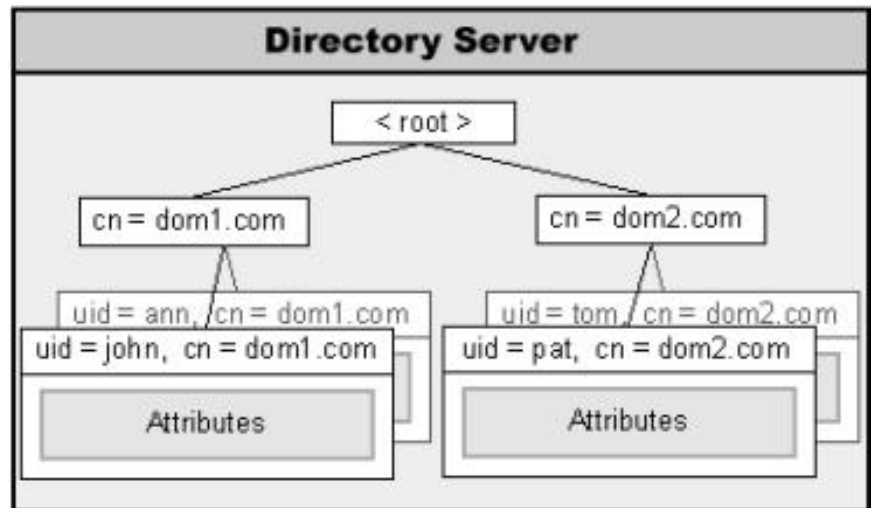
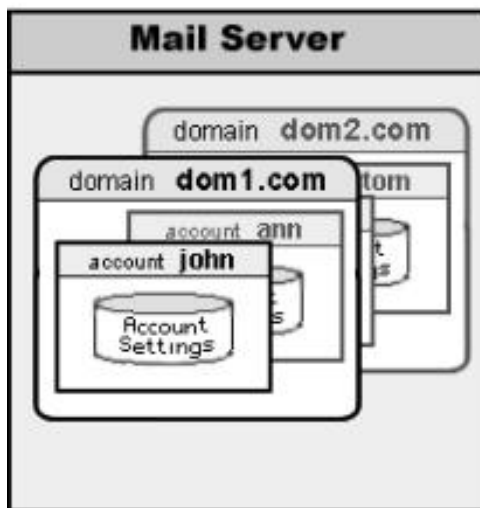


In this example:

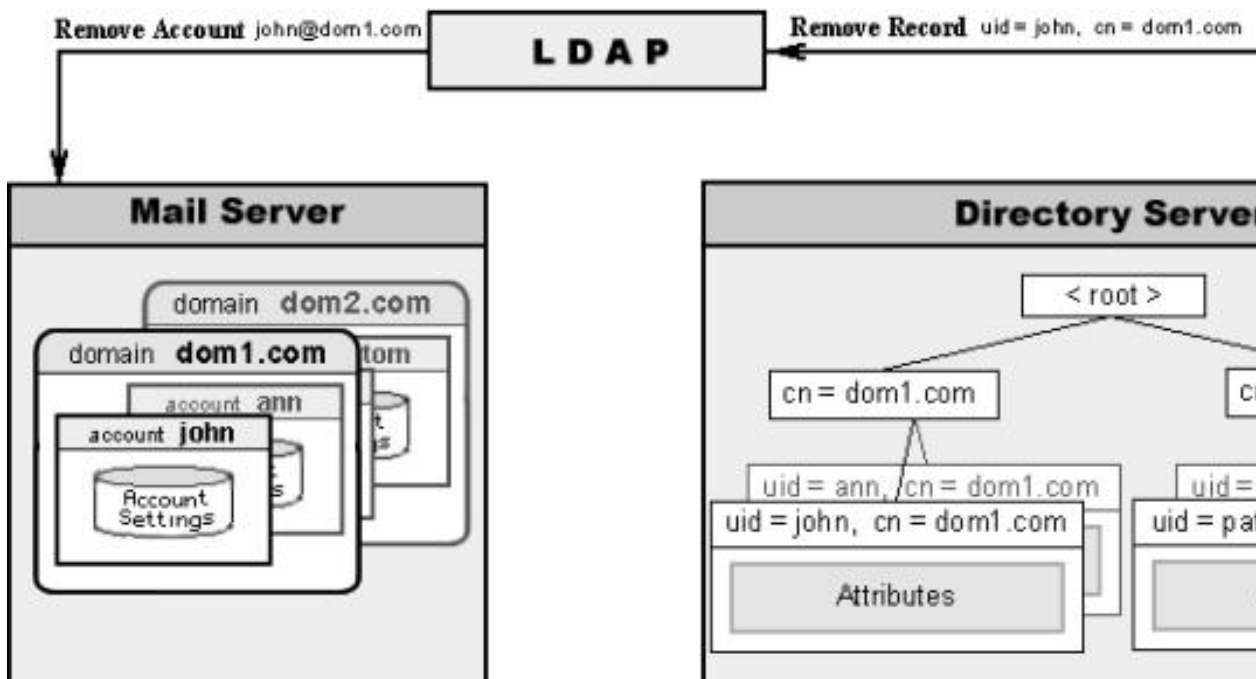
- The LDAP module received a ModifyDN request from an LDAP client. The client asks the LDAP module to change the uid= john , cn=dom1 . com record DN.
- The LDAP module checks the DN and sees that it looks like the DN record for the CommuniGate Pro Account john in the domain CommuniGate Pro Domain dom1 . dom, and the CommuniGate Pro Domain dom1 . com does exist. It also checks that the new name (uid=john1) looks like some CommuniGate Pro Account record DN.
- Instead of performing the requested ModifyDN operation with the Directory, the LDAP module executes the RenameAccount (john , john1) operation in the dom1 . dom Domain.
- The Account john is renamed into john1.
- If the dom1 . com Domain Directory Integration setting is set to Keep In Sync, the Account Manager executes the ModifyDN Directory operation to change the Account record DN in the Directory.

The following diagram illustrates how the LDAP DeleteRecord operation can be used to remove Accounts:

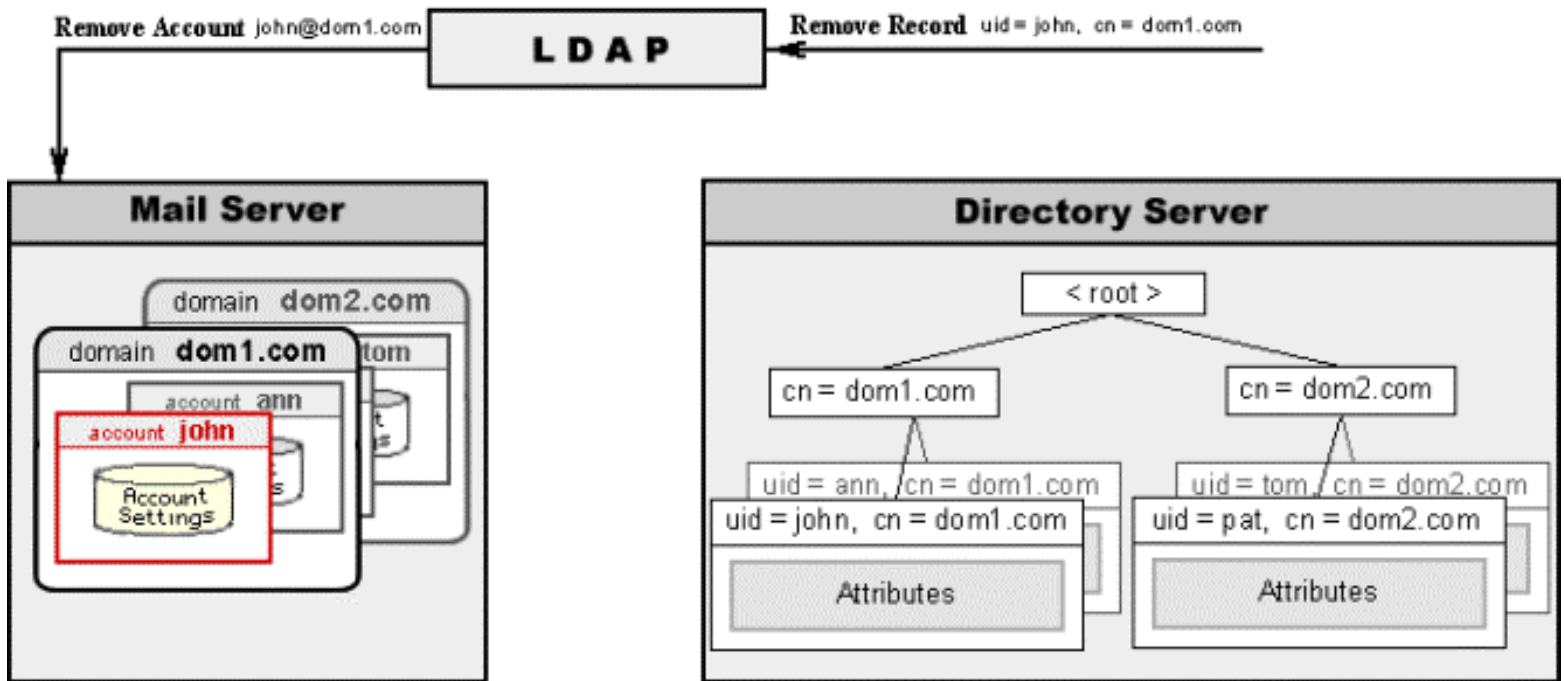
Step 1



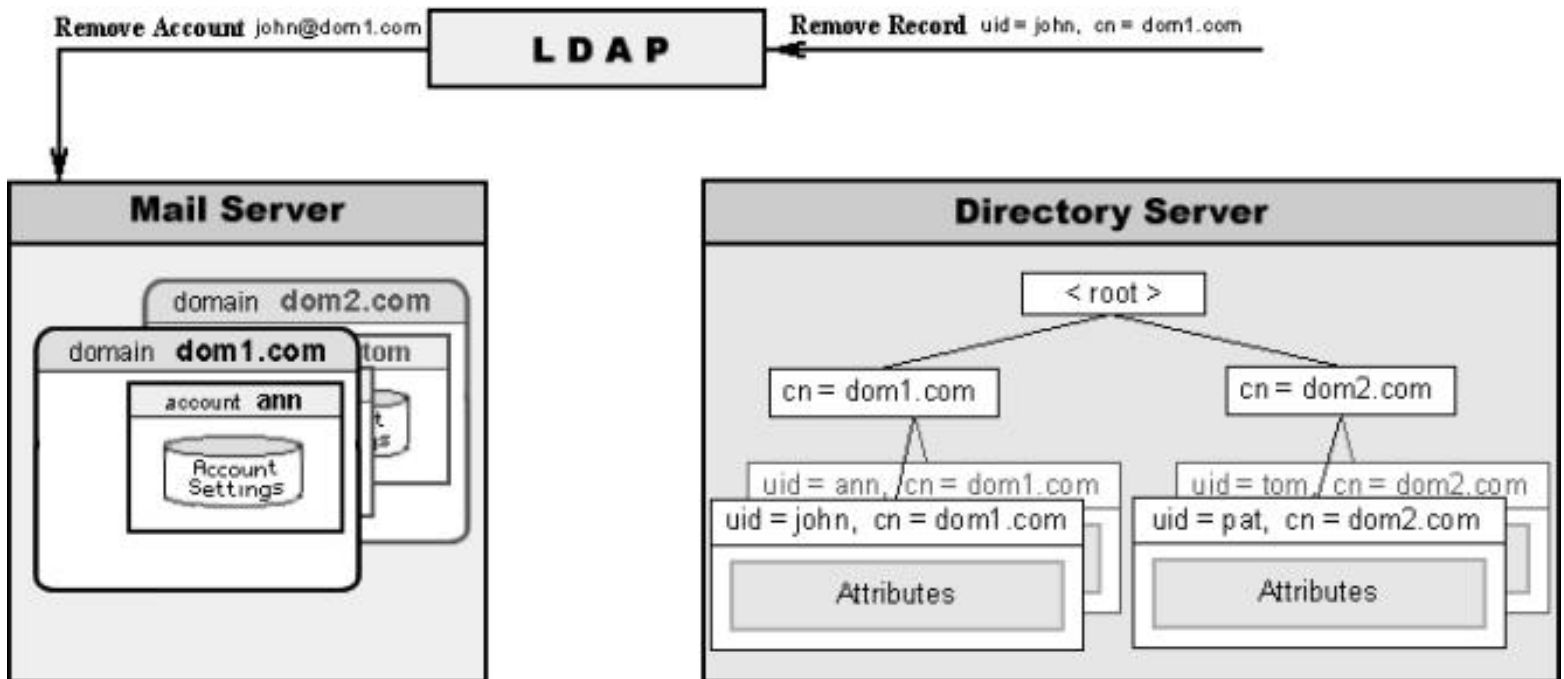
Step 2



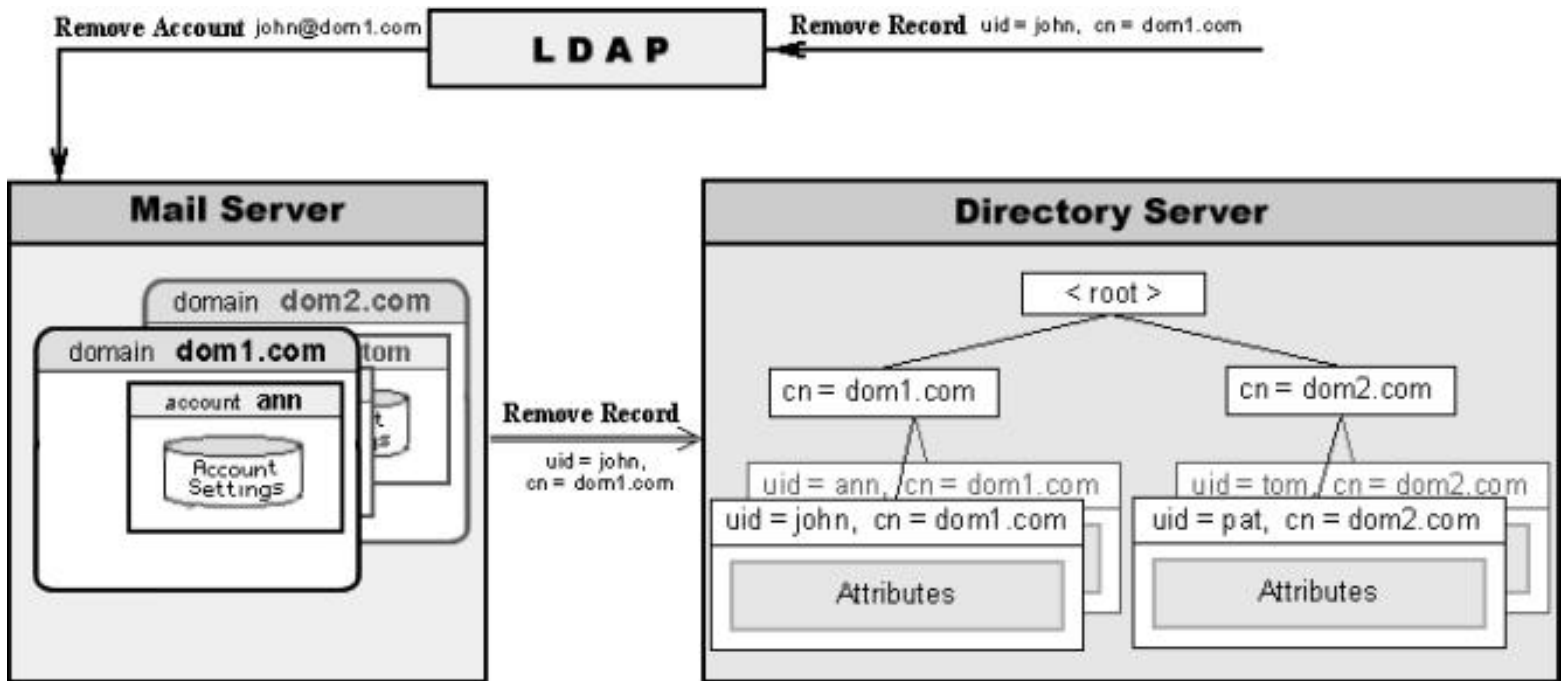
Step 3



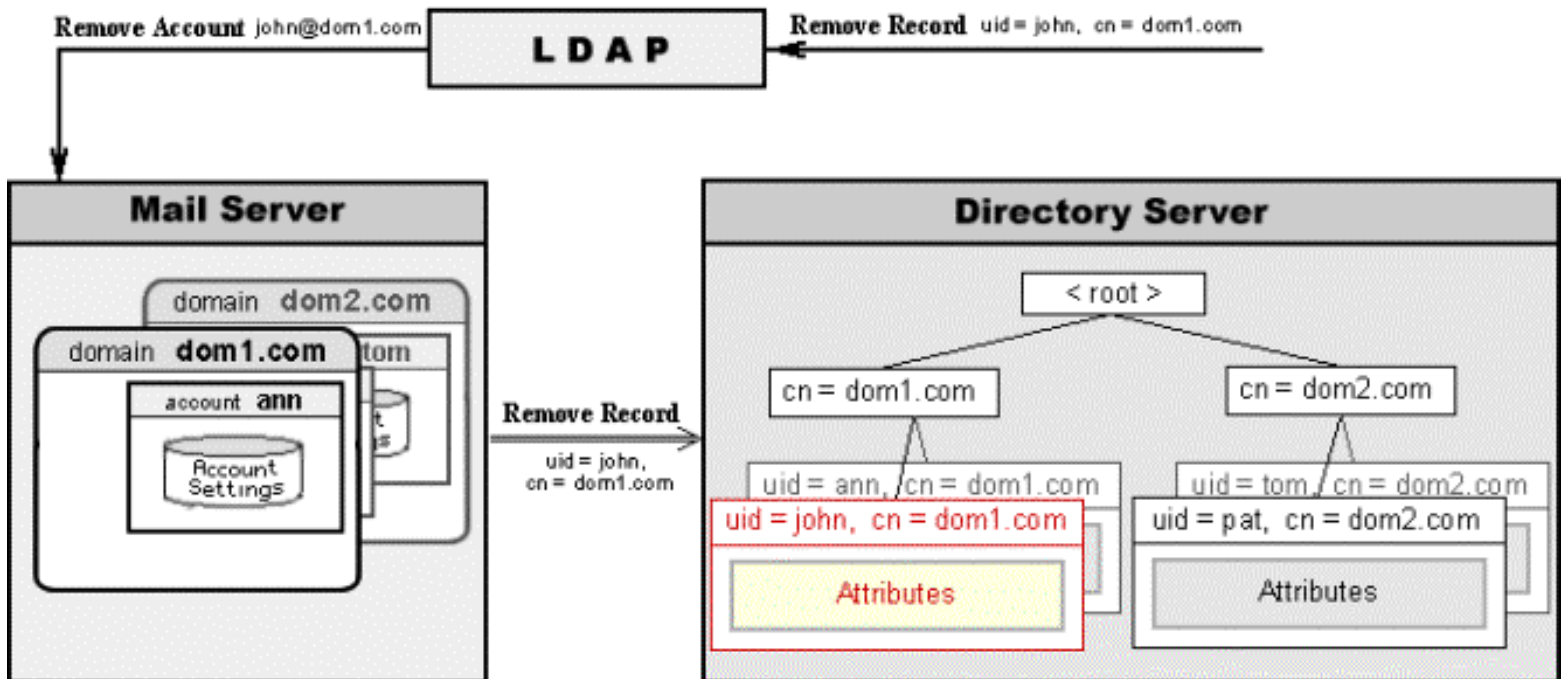
Step 4



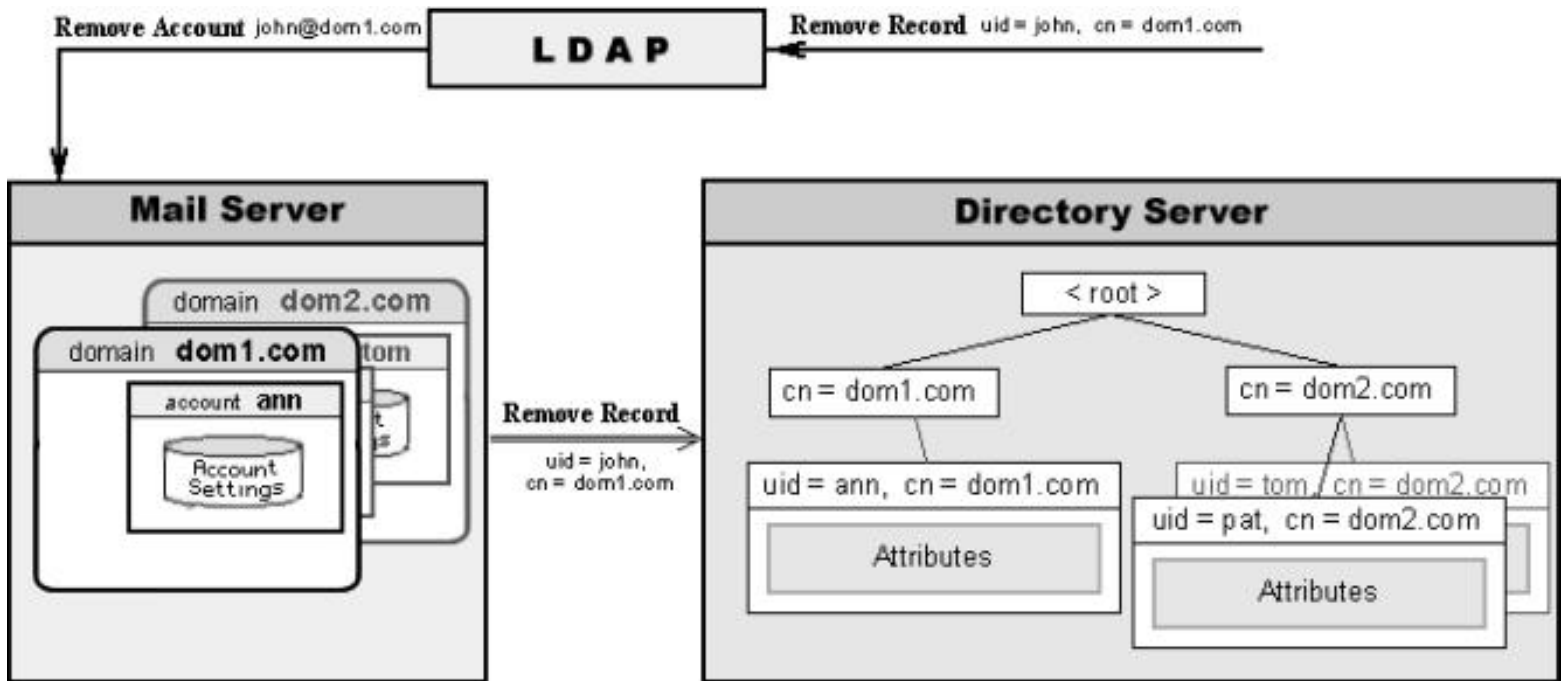
Step 5



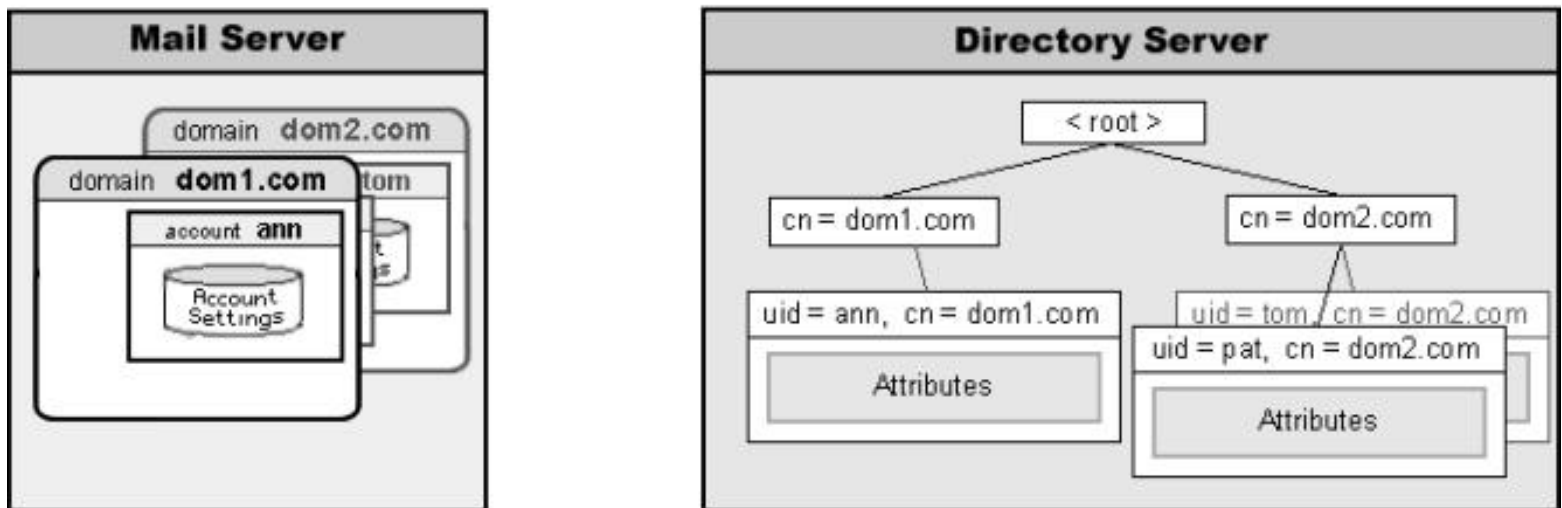
Step 6



Step 7



Step 8



In this example:

- The LDAP module received a DeleteRecord request from an LDAP client. The client asks the LDAP module to delete the uid=john, cn=dom1.com record.
- The LDAP module checks the DN and sees that it looks like the DN record for the CommuniGate Pro Account john in the domain CommuniGate Pro Domain dom1.com, and the CommuniGate Pro Domain dom1.com does exist.
- Instead of performing the requested DeleteRecord operation with the Directory, the LDAP module executes the DeleteAccount(john) operation in the dom1.com Domain.
- The Account john is deleted.

- If the `dom1.com` Domain Directory Integration setting is set to Keep In Sync, the Account Manager executes the `DeleteRecord` Directory operation to remove the Account record DN from the Directory.
-

Directory Integration in a Cluster

The CommuniGate Pro [Dynamic Cluster](#) maintains cluster-wide Directory Integration settings: the cluster-wide Attribute Renaming table, Domain Subtree, and Custom Attributes settings are used with all Accounts in Shared Domains, while "regular" settings are used for all Accounts in "local" Domains.

When you open the Directory Integration WebAdmin page on a Cluster Member, the page contains a link that allows you to switch the Cluster-wide Settings.

Directory-Based Domains

The CommuniGate Pro Server software implements Directory-based Domains. Directory-based Domains and all their Accounts keep all their settings in the Directory - there is no `.settings` files for those domains and accounts.

For each Directory-based Domain a Directory record of the `CommuniGateDirectoryDomain` objectClass is created. This record stores all [Domain Settings](#).

DNs for Directory-based Domains are built in the same way they are built for Regular Domain records.

For each account in a Directory-based Domain a Directory record of the `CommuniGateAccount` objectClass is created. This record stores all [Account Settings](#) (including the Custom Settings).

DNs for accounts in the Directory-based Domains are built in the same way they are built for Regular Domain Account records.

Directory records for Directory-based Domain Accounts must contain the `storageLocation` attribute. This attribute specifies the location of the Account file directory (for the Multi-mailbox accounts) or the location of the account INBOX file (for single-mailbox accounts). The location is specified as a file path relative to the *base directory* of the CommuniGate Pro Server hosting this Account.

If a CommuniGate Pro server has to open an Account in a Directory-based domain, and the account `storageLocation` attribute starts with the asterisk (*) symbol, the CommuniGate Pro Server creates the account file directory (for multi-mailbox accounts) and other required account files and file directories.

- If the `storageLocation` attribute contained only one asterisk, then the new account location path is composed in the same way it is composed for new accounts in the Regular CommuniGate Pro Domains, using the path for the Domain file directory and the Foldering Domain Setting.

Directory records are created for aliases of Directory-based Domain Accounts.

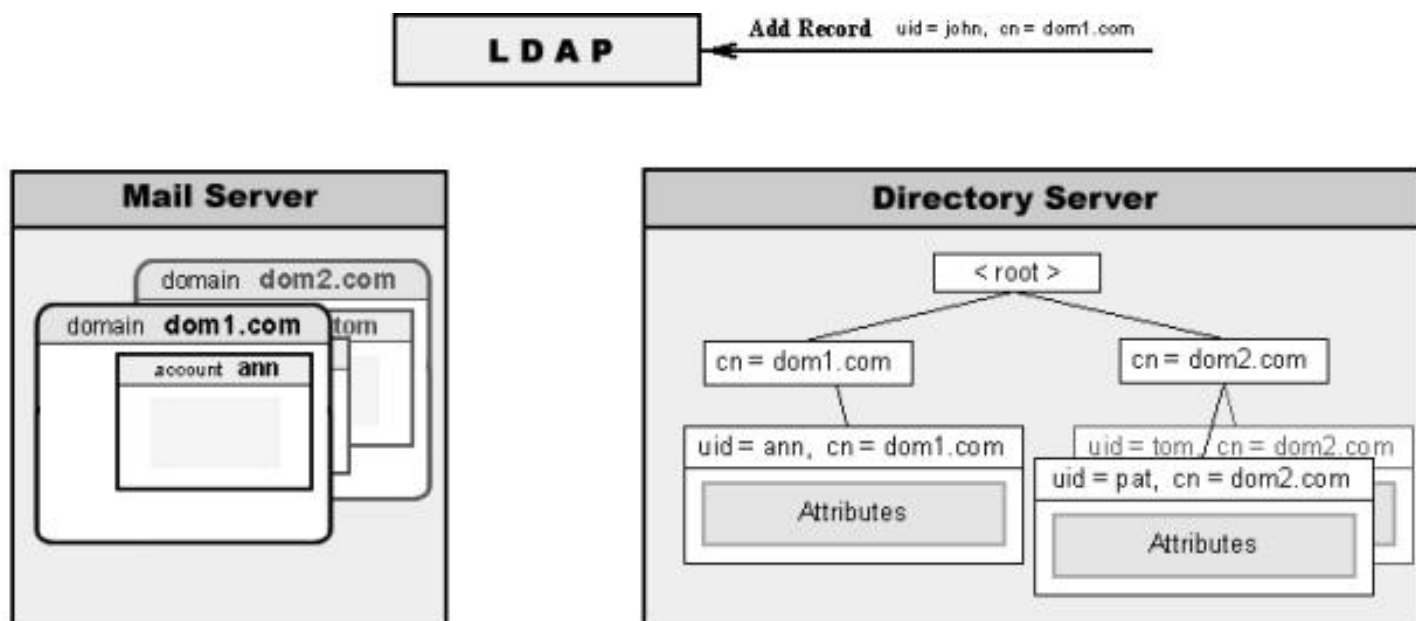
Alias records have the same DNs as Accounts (`uid=aliasname, domain DN`).

Alias records have the standard `alias` objectClass, and their `aliasedObjectName` attribute specifies the DN of the

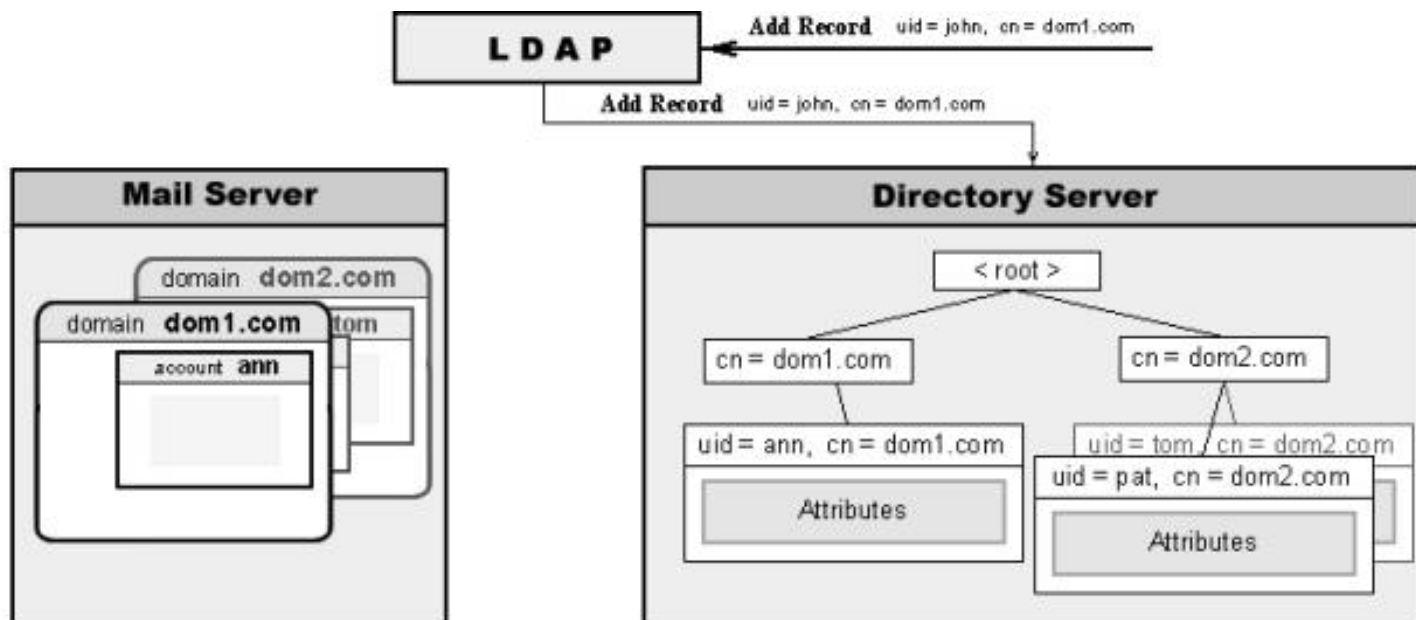
original account record.

The following diagram illustrates how the LDAP AddRecord operation can be used to create an Account in the Directory-based Domain:

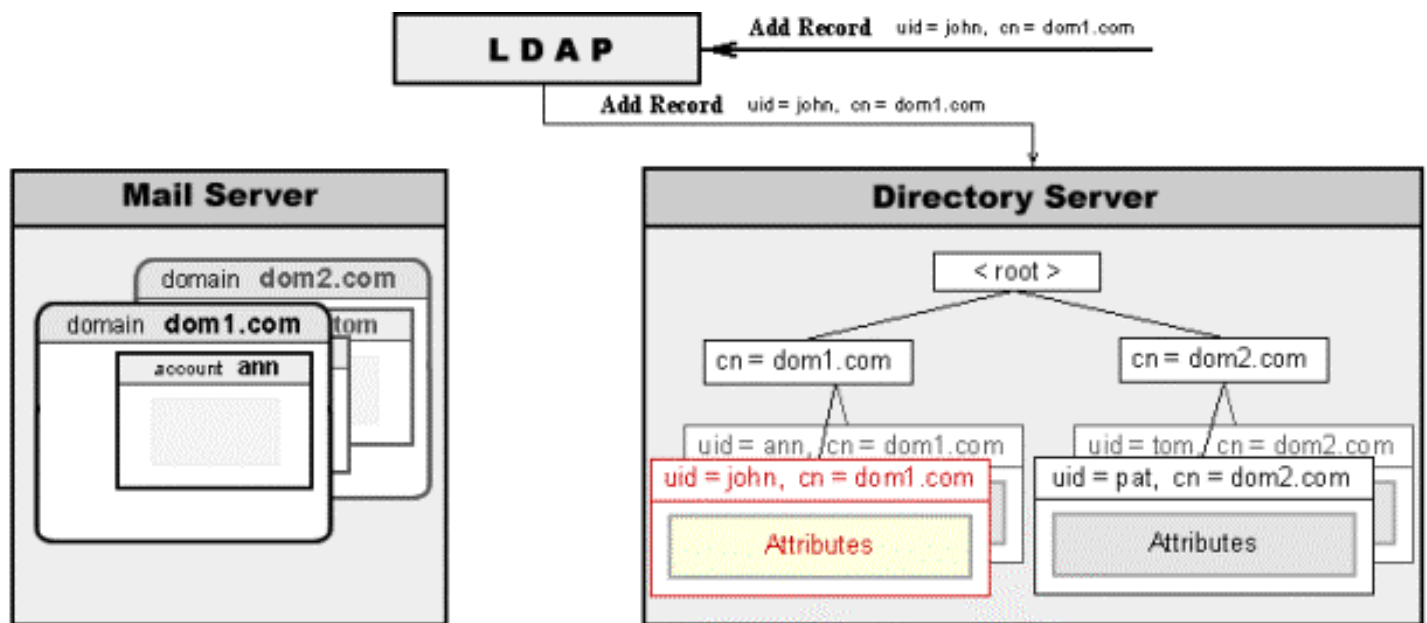
Step 1



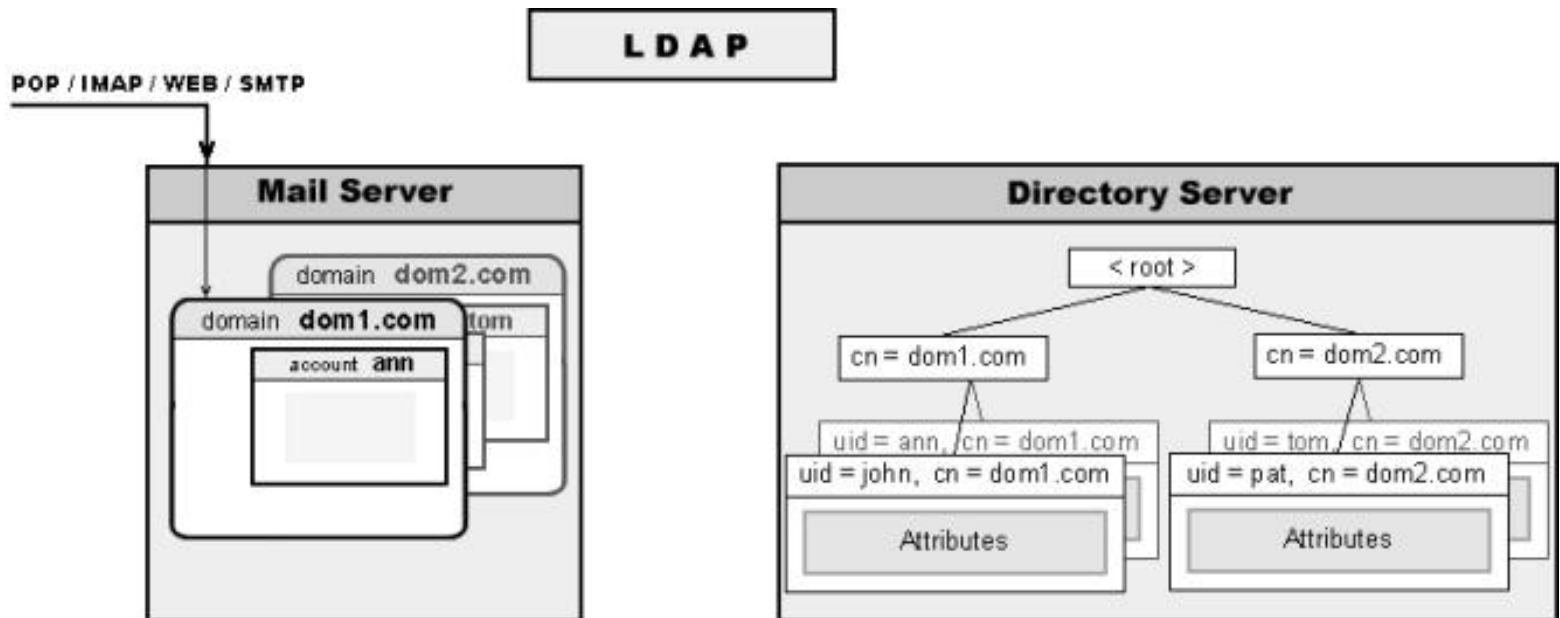
Step 2



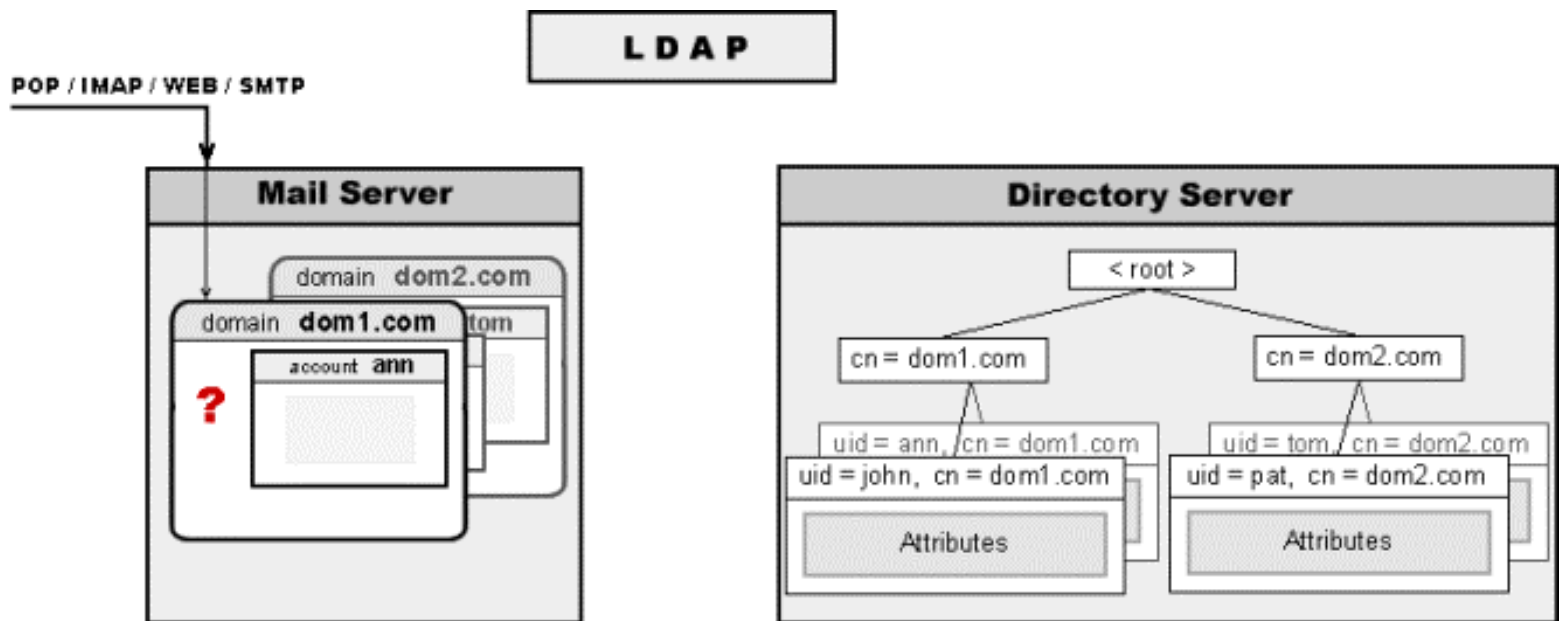
Step 3



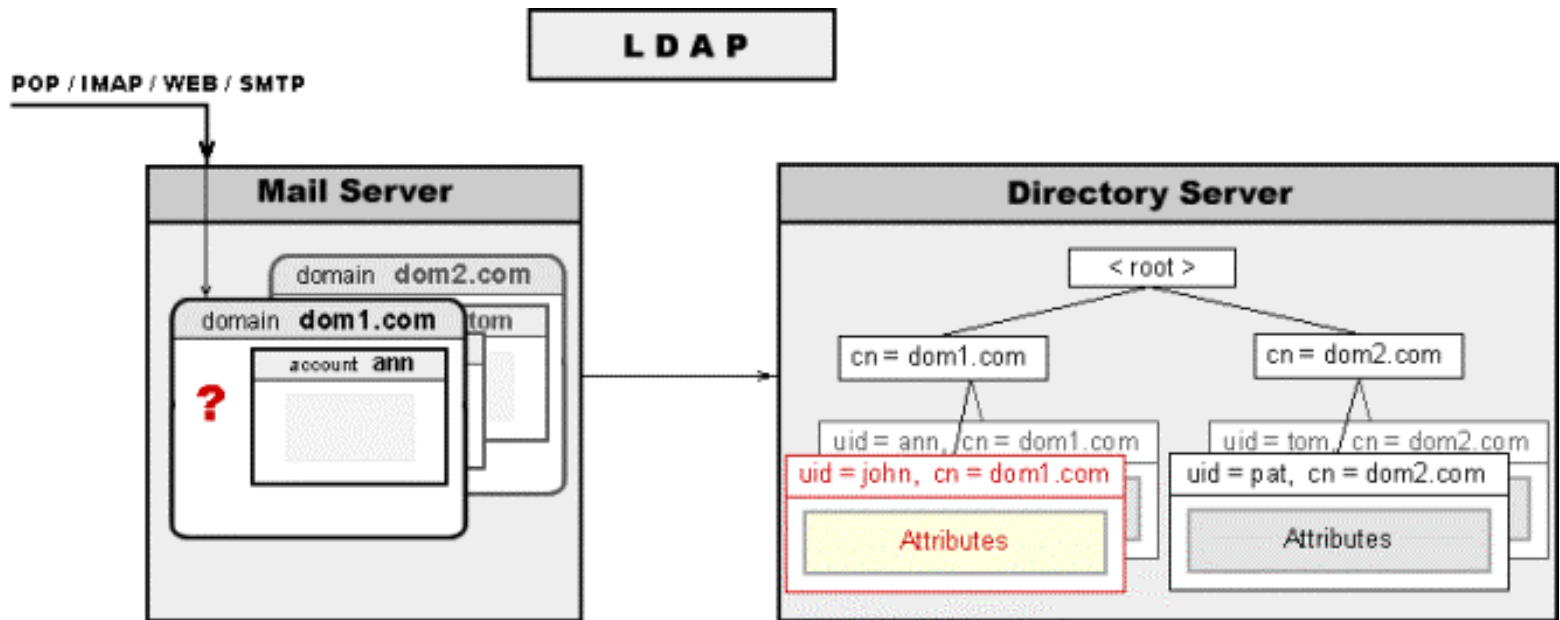
Step 4



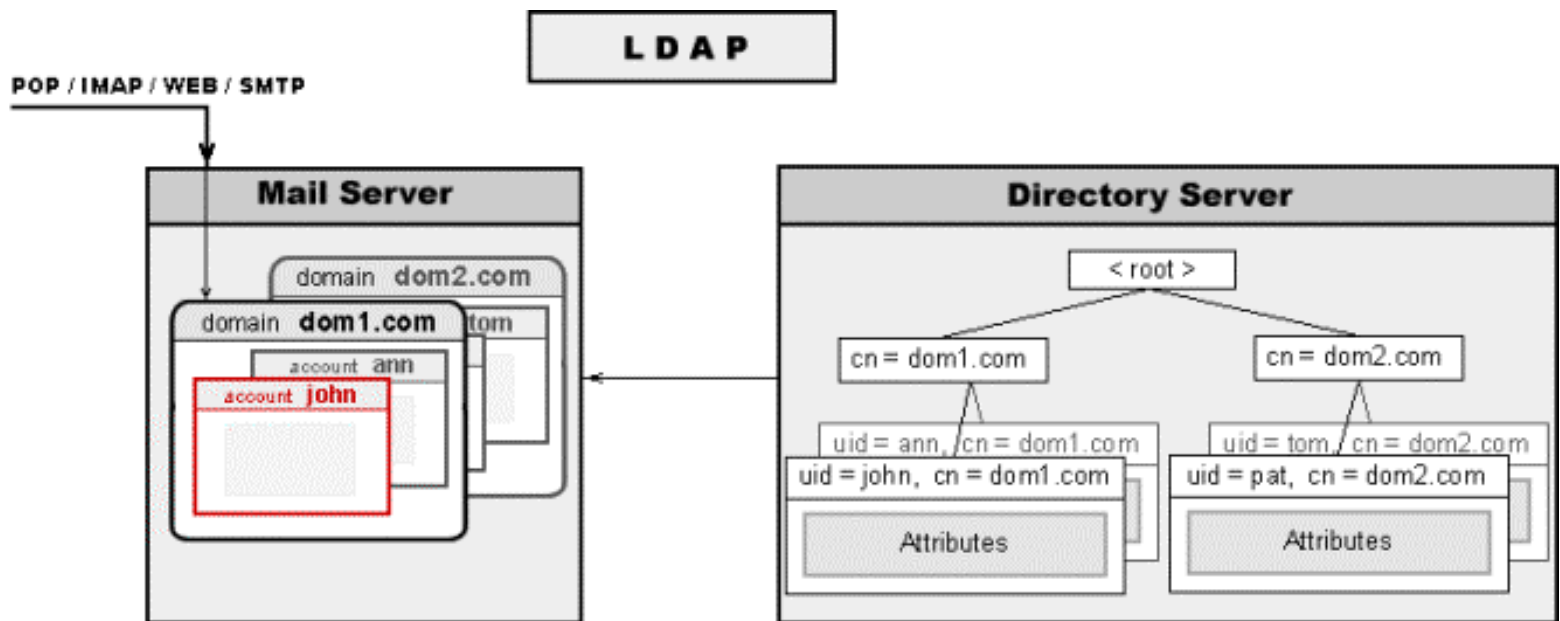
Step 5



Step 6



Step 7

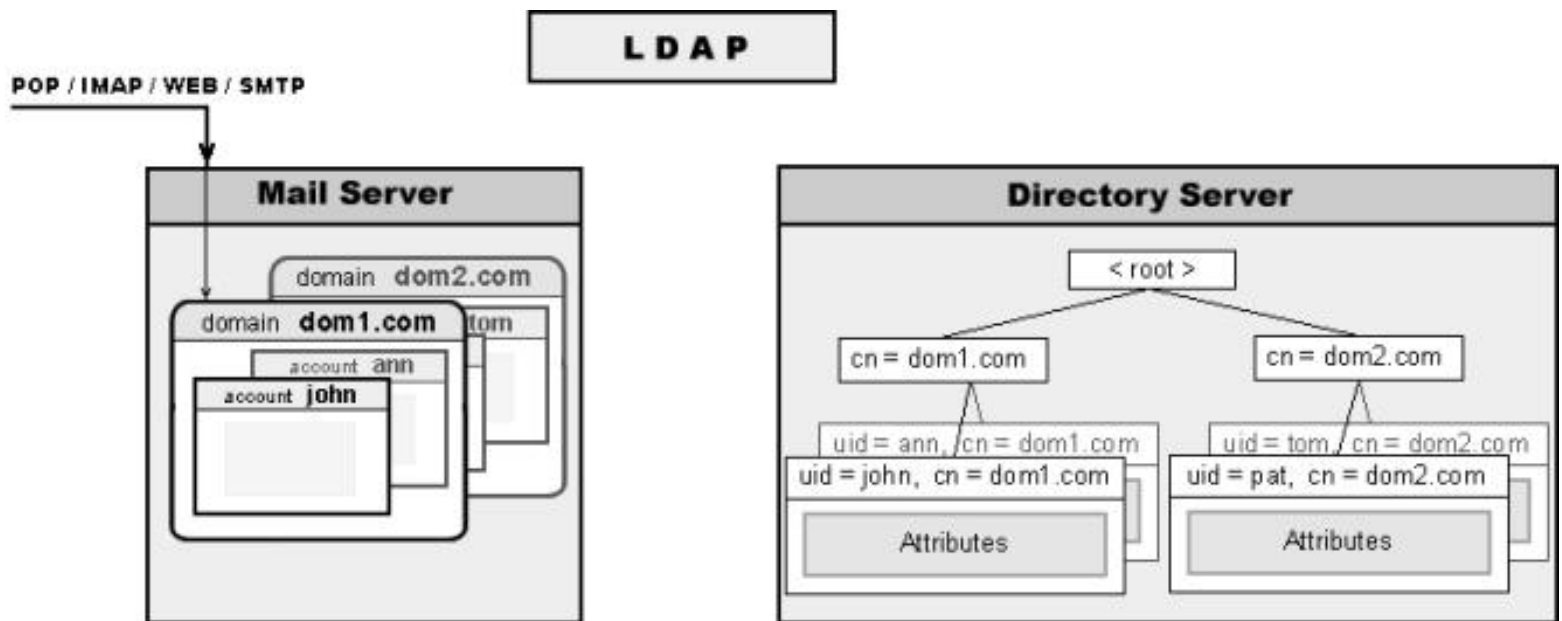


In this example:

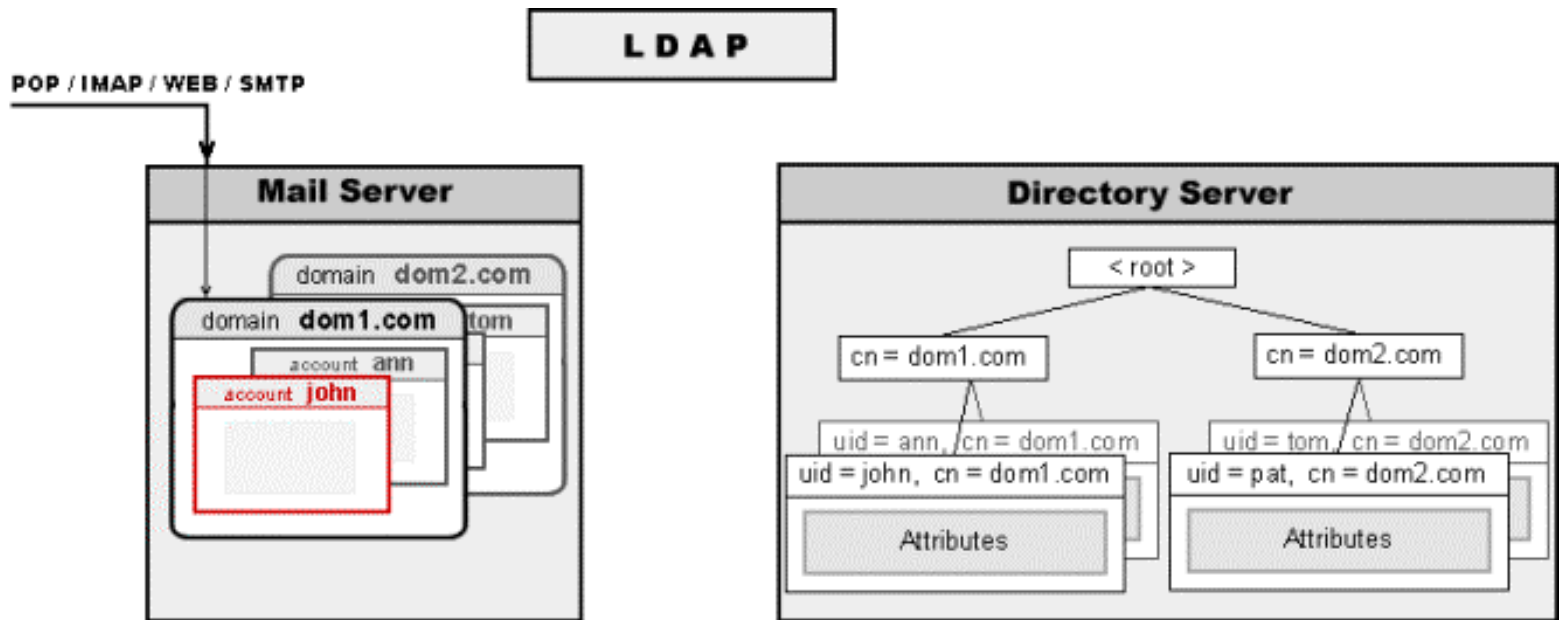
- The LDAP module received an AddRecord request from an LDAP client. The client asks the LDAP module to create a new record with the `uid=john, cn=dom1.com` DN.
- The LDAP module creates a new record in the Directory and stores all supplied attributes in it. The response is sent back to the LDAP client and the operation is completed.
- Any Mail Server component tries to open the account `john` in the Directory-Based Domain `dom1.com`. The request is sent to the Directory and the record with the `uid=john, cn=dom1.com` DN is retrieved.
- The `storageLocation` attribute in the retrieved record contains the asterisk sign. The Mail Server creates the Account files on disk, and updates the Directory record, storing the file path to the Account files in the `storageLocation` attribute.
- The Account `john` is opened and the requested operation is executed.

Since the Account in the Directory-based Domains do not store their settings in the CommuniGate Pro data files, the settings are retrieved from the account Directory record every time an account has to be opened. The following diagram illustrates this procedure

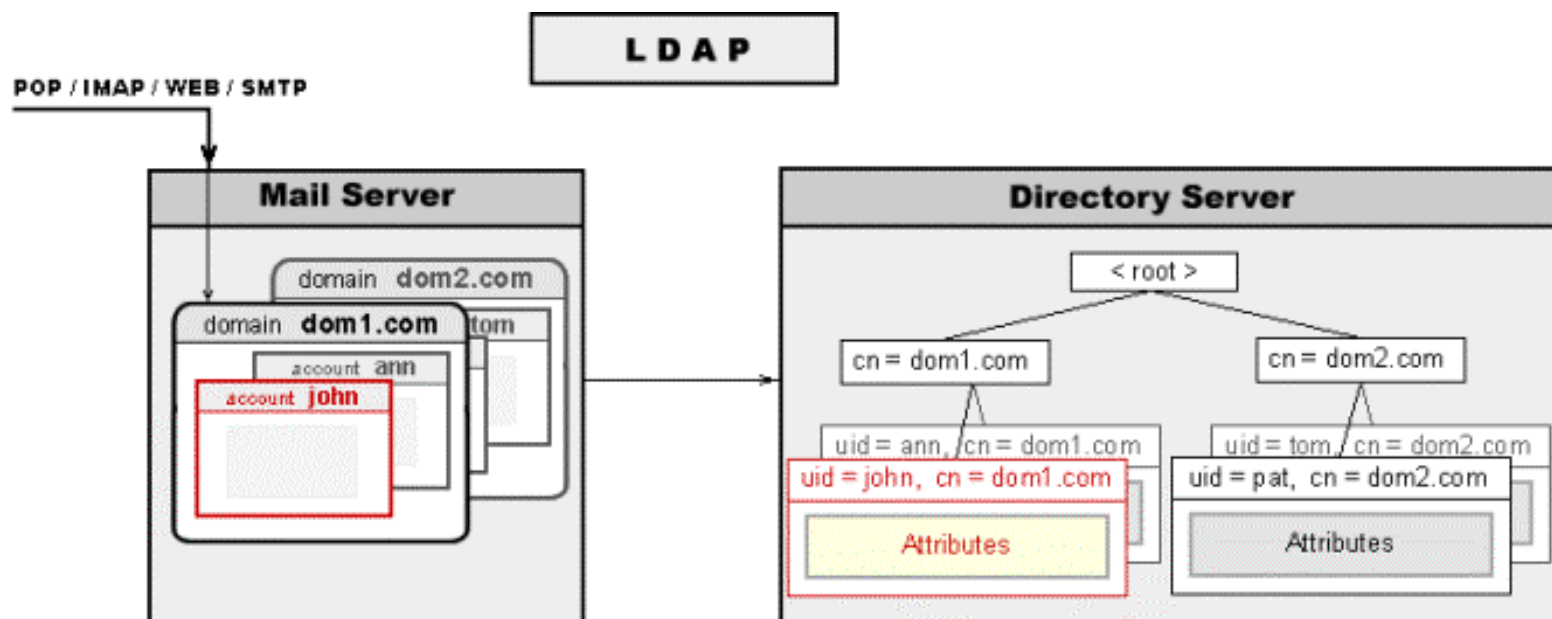
Step 1



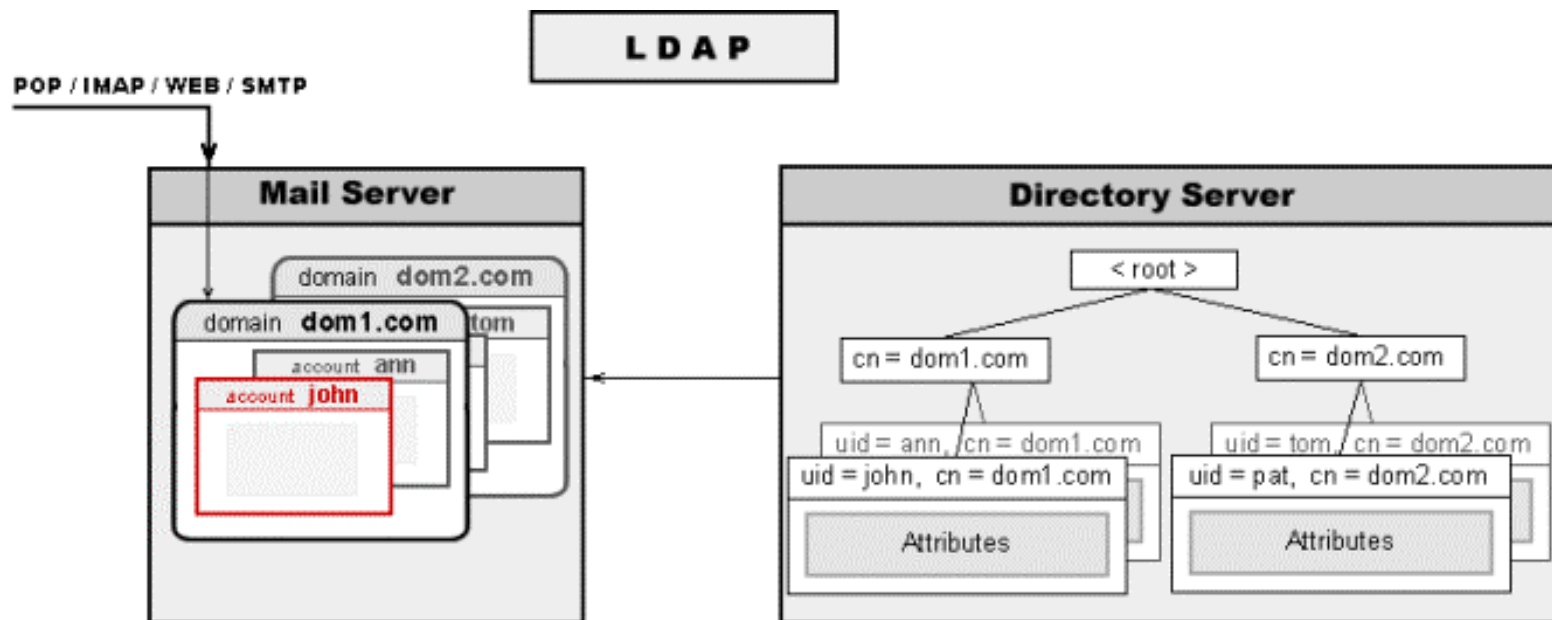
Step 2



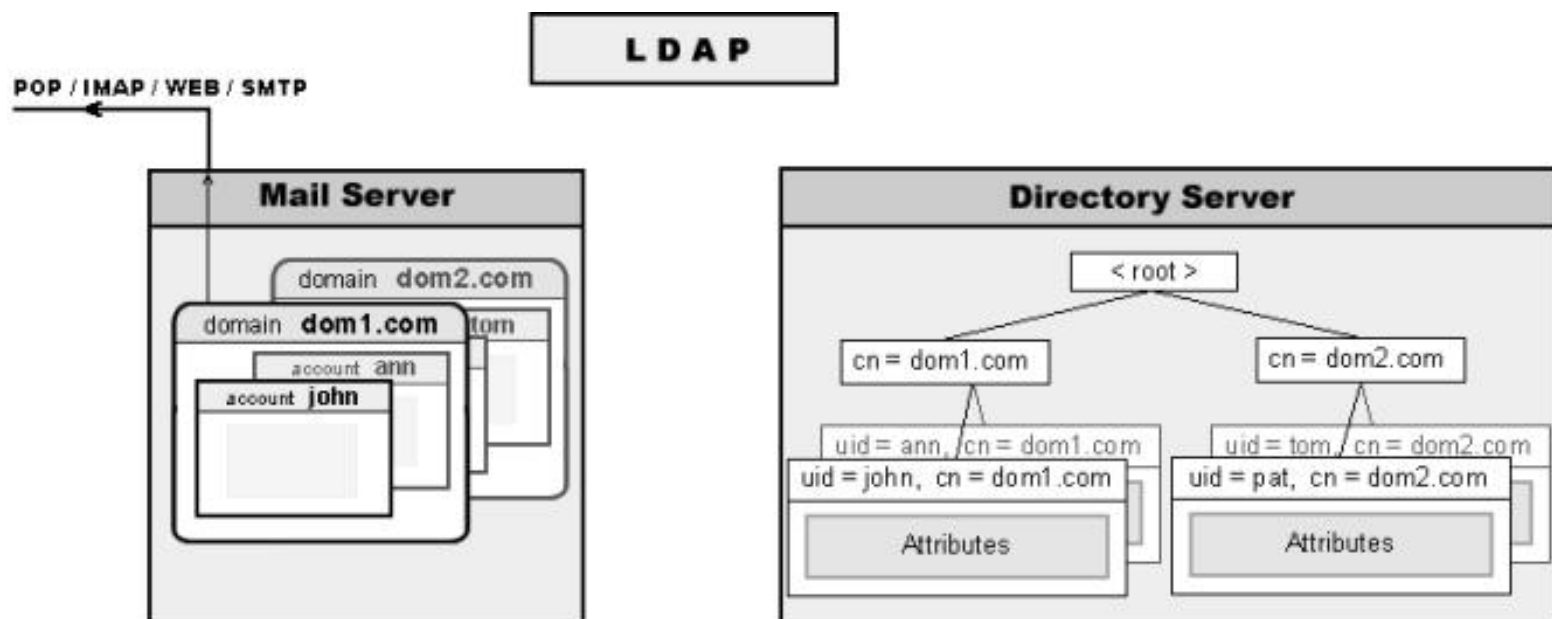
Step 3



Step 4

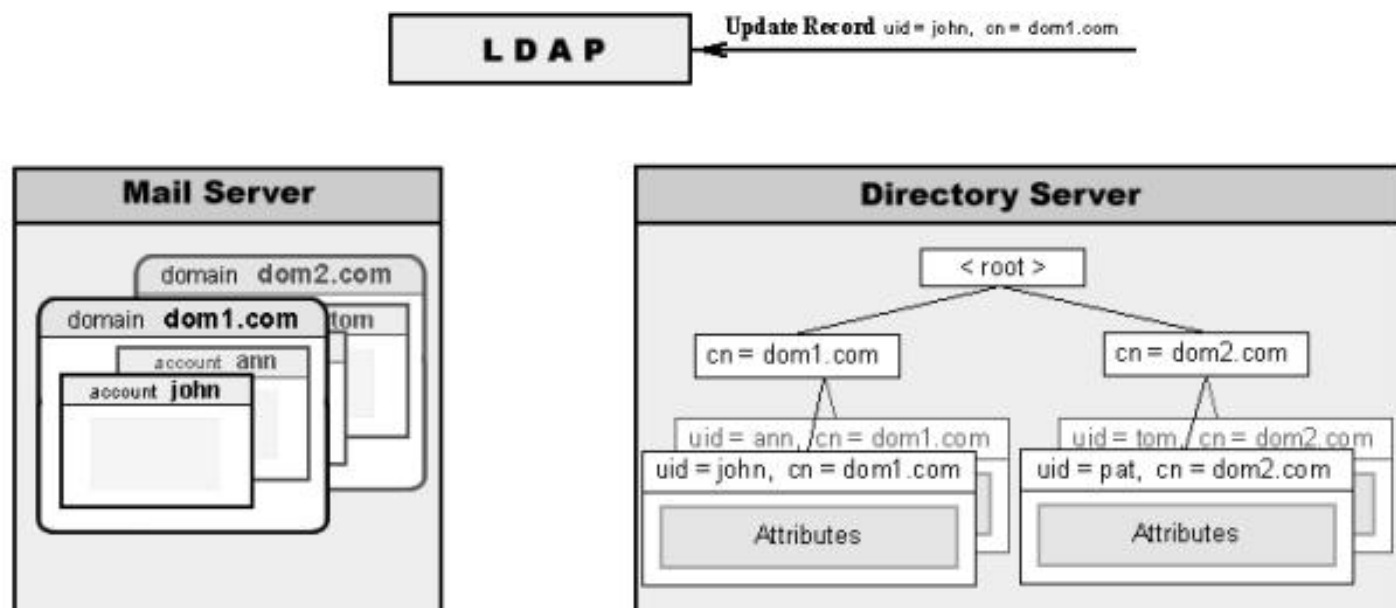


Step 5

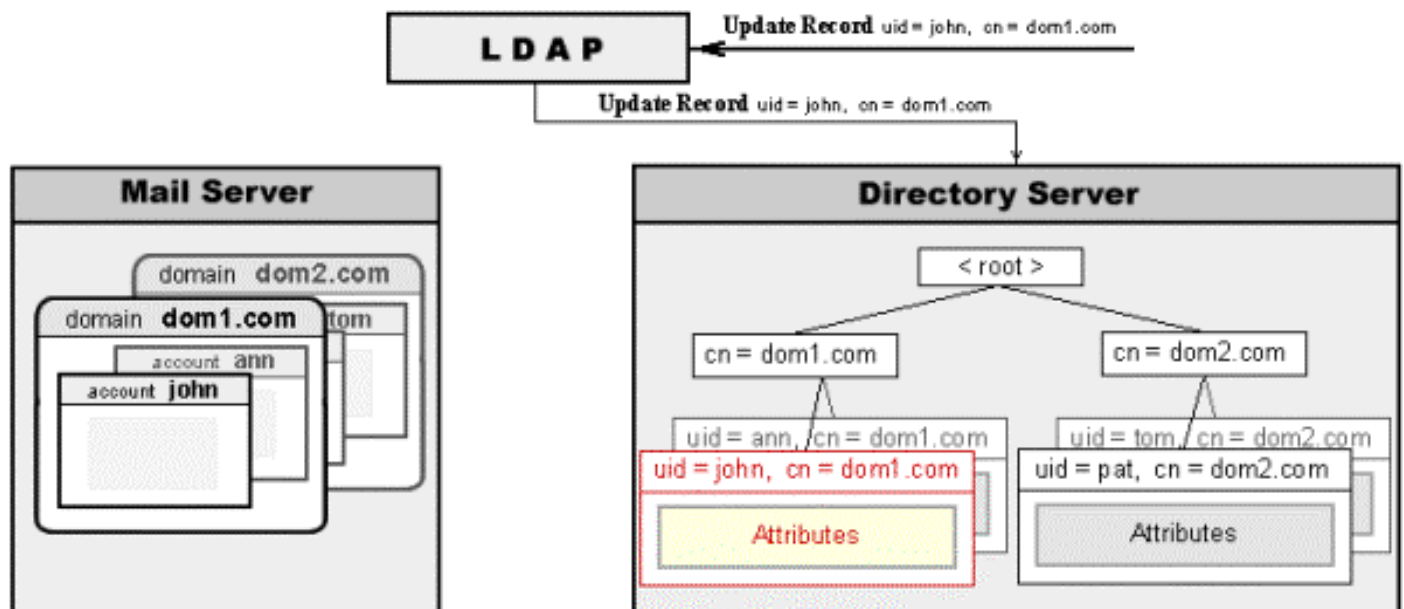


Since the Directory records are the only source of the Account settings, modifying Directory record attributes effectively modifies the Account Settings:

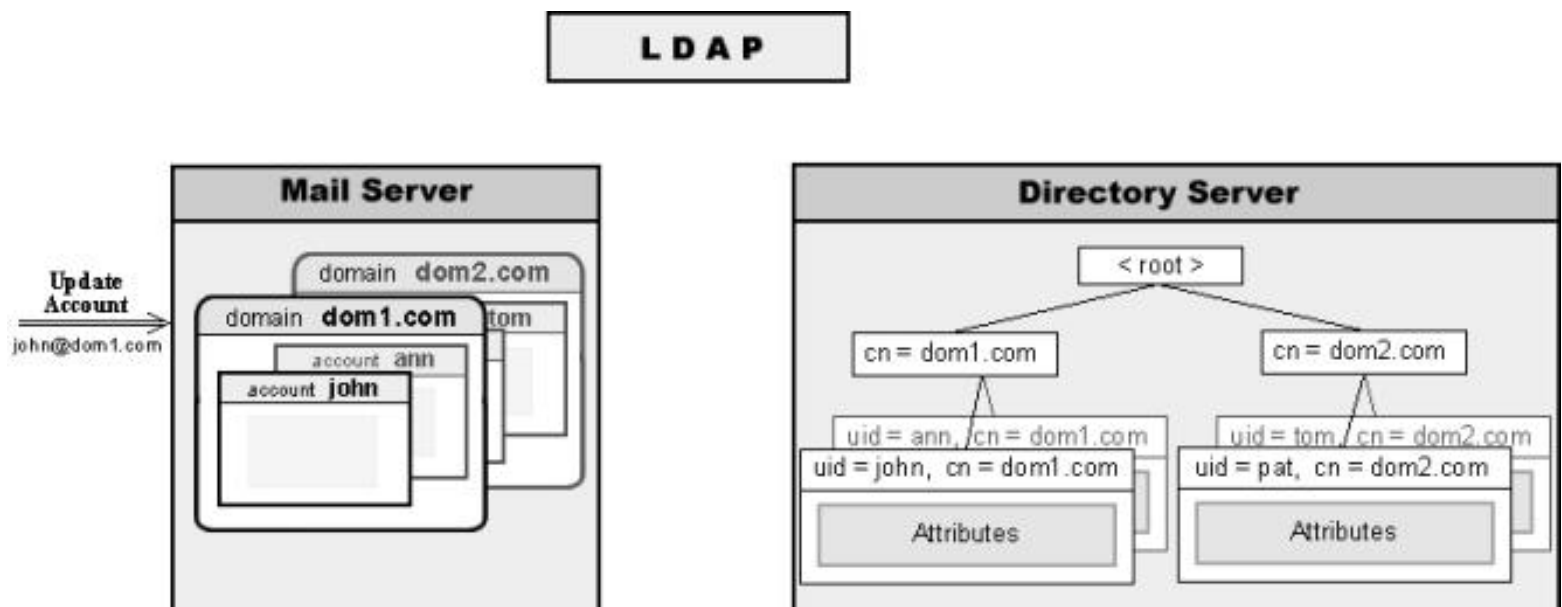
Step 1



Step 2

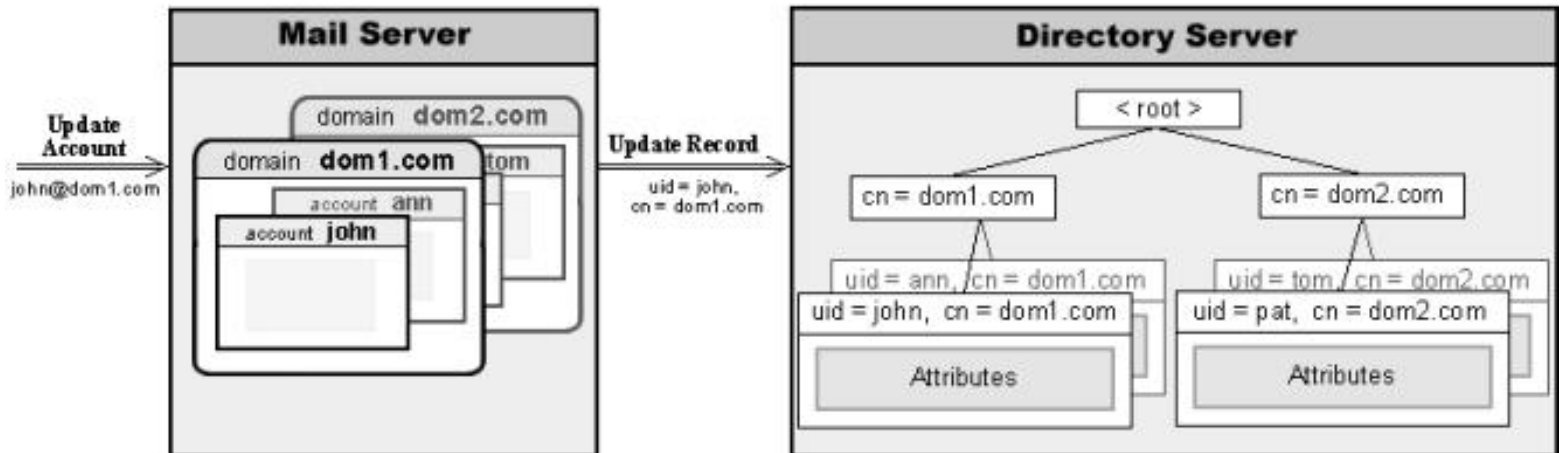


Step 3



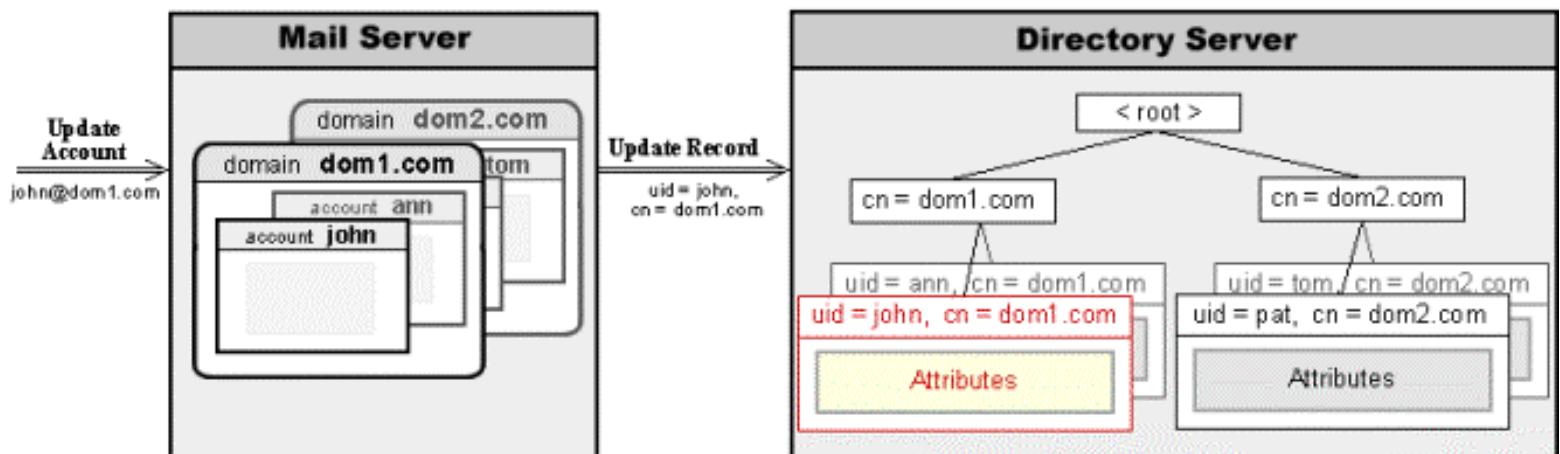
Step 4

LDAP



Step 5

LDAP



In this example:

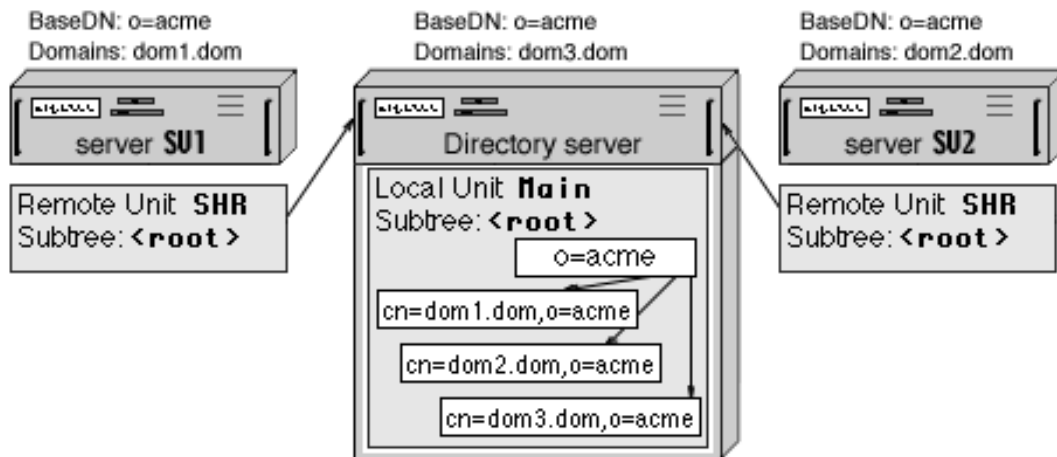
- The LDAP module received a ModifyRecord request from an LDAP client. The client asks the LDAP module to change attributes in the **uid=john, cn=dom1.com** record.
- The LDAP module tells the Directory to modify the specified record. The response is sent back to the LDAP client and the operation is completed.
- Any Mail Server component tries to open the account **john** in the Directory-Based Domain **dom1.com**. The request is sent to the Directory and the record with the **uid=john, cn=dom1.com** DN is retrieved and its attributes are used as Account Settings. Since the attribute value has been modified, the Account Setting set with that attribute is effectively modified.

Several CommuniGate Pro Servers can use the same physical Directory (Directory Unit) to keep all their Domain Integration Records.

The shared Directory Unit can be implemented as a Local Storage Unit on one of the CommuniGate Pro Servers, or it can be hosted on some third-party Directory Server.

- Specify the same [Domains Subtree](#) parameters on all CommuniGate Pro Servers.
- On all CommuniGate Pro Server (except the one that will host the shared Directory) create Remote Storage Units for the same Subtrees. The Remote Storage Unit Subtree parameter should be either the same as the Domains Subtree Base DN parameter, or should be its parent, so the entire Domains Subtree will be stored in those Remote Storage Units.
- Configure all those Storage Units to point to the server that will host the shared Directory.

To simplify the setup, especially if you have many CommuniGate Pro Servers, it is recommended to create the Remote Storage Units for the <root> Subtrees. To create such a Unit, remove the default Main Local Unit first:

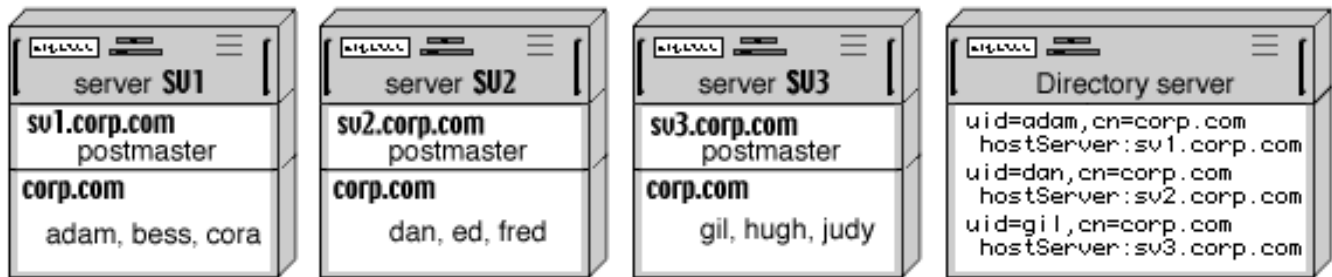


In this example:

- One CommuniGate Pro Server is used to host the Shared Directory.
- The SV1 and SV2 CommuniGate Pro Servers are configured to use that Shared Directory: they both have Remote Storage Units SHR for their <root> Directory Subtrees, and those Units point to the Directory hosting Server.
- All Servers have the same Directory Integration settings - the Domain Subtree Base DN is o=acme for all Servers.
- The actual o=acme record is created in the Main Local Storage Unit on the Directory Hosting Server.
- The Directory records for:
 - the dom1 . dom domain created on the SV1 Server,
 - the dom2 . dom domain created on the SV2 Server, and
 - the dom3 . dom domain created on the Directory Hosting Serverare stored in the Main Local Storage Unit on the Directory Hosting Server

Distributed Domains (Directory Routing)

When several CommuniGate Pro Servers use a [Shared Directory](#) to keep all their Domain Integration Records, these Servers can be used to serve the same domain (or the same domains). Such a Domain is called a Distributed Domain, and each Server hosts a subset of all Domain Accounts. The Shared Domain should not be a Main Domain of any CommuniGate Pro Server:



In this example:

- Three CommuniGate Pro Servers (with the `sv1.corp.com`, `sv2.corp.com`, and `sv3.corp.com` Main Domains) all have the same `corp.com` Secondary Domain. Some Accounts are created in the `corp.com` domain on each Server.
- The Domain Subtree Base DN is set to an empty string (`<root>`) on all CommuniGate Pro Servers.
- The Shared Directory is hosted on a separate device/server, but in reality one of the CommuniGate Pro Servers can act as the Shared Directory Host.
- The Directory Integration option of the `corp.com` Domain is set to `Keep In Sync` on all Servers.

When an Account is created, renamed, removed, or updated on one of the `sv*.corp.com` Servers, the Directory Unit on the Shared Directory Server is updated. As a result, the Shared Directory contains records for all Accounts created on all `sv*.corp.com` Servers.

When any Server creates an Account and places a record into the Shared Directory, it stores the Server Mail Domain name as the record `hostServer` attribute.

The Shared Directory can be used to route Shared Domain mail to the proper location (Server). After you enable the Directory-Based Routing Setting in the CommuniGate Pro General->Cluster Settings, the address routing mechanism is modified:

- When the CommuniGate Pro Server receives a mail for one of its local Domains, it checks if a local domain object (Account, Alias, Mailing List, Group, Forwarded) exists.
- If no local object is found in the addressed Domain, the Server checks the Directory.
- If the Directory contains a record for the specified object (`uid=objectName, cn=domainName`), the record `hostServer` attribute is checked.
- If the `hostServer` attribute is absent, or if it contains the Main Domain Name of this CommuniGate Pro Server, an error message is generated. Otherwise, the `hostServer` name attribute value with added `.smtp` suffix is appended to the original address:

`objectName%domainName@hostserver.smtp`

The [SMTP module](#) accepts such an address and routes it to `objectName@domainName` at the remote host `hostserver`.

This Distributed Domain configuration is useful for multi-location and international organizations and corporations where all employee accounts should be in the same domain, but each organizational unit is served with its own Server. The DNS MX records for the such a Distributed Domain should point to any or to all Servers hosting that domain. When a Server receives mail for a Distributed Domain, it either delivers the mail locally (if the addressed Account is hosted on that Server), or relays mail to Server specified in the hostServer attribute of the Account Directory record.

Usually, one of the Servers (the "main location") hosts most of the Distributed Domain Accounts. It is recommended to host the Shared Directory on that CommuniGate Pro Server to minimize the delays introduced with the Directory lookups. Other CommuniGate Pro Server serving this Distributed Domain can be configured to reroute all mail to non-local objects of the Distributed Domain to that "main location" Server. In the Distributed [Domain Settings](#), set the Mail To Unknown option to

```
Reroute To: *%domain.com@mainserver.smtp
```

This method eliminates a need for "remote location" Servers to communicate with the Directory when they have to route addresses. The "remote location" Servers communicate with the Directory only when Accounts are created in, renamed, or removed from a Distributed Domain, and when a WebMail or LDAP user requests a Directory search operation. This can drastically improve the "remote location" Servers performance if the communication links between them and the Shared Directory Server are slow and/or unreliable.

In asymmetric, "main/remote location" configurations, the high-priority MX records for the Distributed Domain should point to the "main location" Server, while "remote location" Server names can be used for low-priority MX records. It is not recommended to use [Directory-based Domains](#) for Distributed Domains if connections between "remote location" Servers and the Shared Directory are slow and/or unreliable.

The Distributed Domains concept is the foundation of the CommuniGate Pro [Static Clusters](#).

For small Distributed Domains, routing can be implemented using regular CommuniGate Pro [Router](#) records. If the Distributed Domain has the same Accounts as shown in the example above, the SV1 server should have the following records in its Router:



While this method does not require any Directory activity, it is hardly acceptable for Domains with more than few dozen Accounts, unless names of Accounts hosted on different Servers can be easily expressed using the Router wildcard symbols. For example, if all Accounts hosted on the Server SV2 end with the -uk suffix (dan-uk@corp.com, ed-uk@corp.com, fred-uk@corp.com, etc.), routing for all SV2 Accounts can be specified with one Router record:

<*-uk@corp.com> = *-uk%corp.com@sv2.corp.com.smtp



Data Formats

The CommuniGate Pro server uses several formats for the files it creates and maintains. All data is stored in the text form to make it easy to move files between platforms, and to process CommuniGatePro data with external programs.

Strings, Dictionary, and Array Formats

Strings

Strings are the very basic, unstructured data.

A string is either an *atom* - a sequence of letters and digits, or a *quoted string* - a sequence of any printable symbols except the quotation mark and the backslash symbol enclosed into the quotation marks (").

Examples: `MyName My2ndName "My Name with spaces and the . sign"`

If you want to include the quotation mark into a string, include the backslash symbol and the quotation mark, if you want to include the backslash symbol into a quoted string, include 2 backslash symbols.

Examples: `"a \"string\" within string" "Single \\ backslash"`

You can use the `\r` symbol combination to include the *Return* symbol into a string, you can use the `\n` symbol combination to include the *Line Feed* symbol into a string, and you can use the `\e` symbol combination to include the system-independent *End-Of-Line* symbol(s) into a string.

Examples: `"Line1\eLine2" "TEXT3\rTEXT67\nTEXT78"`

Use the `\r` and `\n` combinations to include the return and line-feed characters only when they are NOT used as line separators.

You can use the `\t` symbol combination to include the *Tab* symbol into a string.

Example: `"Line1:\tField1\tField2\eLine2:\tField1\tField2"`

You can use the `\nnn` symbol combination to include any symbol into a string, if *nnn* is a 3-digit

decimal number equal to the code of the desired symbol.

Examples: "Using the \012 - Vertical Tabulation symbol"

Arrays

An array is a set of values, separated with the comma signs (,) and enclosed into the parenthesis.

Example: (Element1 , "Element2" , "Element 3")

An array element can be either a string, or an array, or a dictionary.

Example: (Element1 , ("Sub Element1", SubElement2) , "Element 3")

Any number of spaces, tabulation symbols, and/or line breaks (end-of-line symbols) can be placed between a parentheses and an element, and between an element and a comma sign.

Example:

```
(
Element1  ,
    ("Sub Element1",
    SubElement2  )
,
"Element 3"  )
```

Dictionaries

The Dictionary format is used for most CommuniGate Pro setting files, as well as for some other files and data.

A Dictionary value can be either a string, or an array, or a dictionary, so the format definition is recursive.

A dictionary is a list of key-value pairs. Each key should be unique, and the key names are **case-sensitive**. The equal sign (=) is placed between the key and its value, end the semicolon symbol (;) is placed after a value. The set of key-value pairs is enclosed into the curvy brackets. Each key is a string.

Example: {Key1=Element1; Key2 ="Element2" ; "Third Key"="Element 3"; }

The value element in any key-value pair can be a string, an array, and/or a dictionary.

Example: {Key1=(Elem1,Elem2); Key2={Sub1="XXX 1"; Sub2=X245;}; }

Any number of spaces, tabulation symbols, and/or line breaks (end-of-line symbols) can be placed between a bracket and a pair, around the equal sign, and around the semicolon sign.

Example:

```
{
Key1  =    (Elem1,Elem2)    ;
```

```

Key2 = { Sub1 = "XXX 1";
        Sub2=X245; };
}

```

Syntax Rules

Below is the formal syntax for the Dictionary and Array formats:

```

a-symbol ::= A .. Z | a .. z | 0 .. 9
atom      ::= 1*a-symbol
s-symbol  ::= any printable symbol except " and \ |
               \\      |  \"      |  \r      |  \n      |  \e      |  \nnn
string    ::= " 1*s-symbol " | atom
array      ::= ( [object [, object ...]] )
dictionary ::= { [string = object ; [string = object ; ...]] }
object     ::= string | array | dictionary

```



Clusters

When your site serves more than 150,000-200,000 accounts, or when you expect a really heavy IMAP/WebMail traffic, you should consider using a *Cluster configuration*.

If your site serves many domains, you may want to install several independent CommuniGate Pro Servers and distribute the load by distributing domains between the servers. In this case you do not need to employ the special Cluster Support features. However if you have one or several domains with 100,000 or more accounts in each, and you cannot guarantee that clients will always connect to the proper server, or you need dynamic load balancing and very high availability, you should implement a CommuniGate Pro Cluster on your site.

Many vendors use the term *Cluster* for simple *fail-over* or *hot stand-by* configurations. The CommuniGate Pro software can be used in fail-over, as well as in [Distributed Domains](#) configurations, however these configurations are not referred to as *Cluster configurations*.

A CommuniGate Pro Cluster is a set of server computers that handle the site mail load together. Each Cluster Server hosts a set of regular, non-shared domains (the CommuniGate Pro Main Domain is always a non-shared one), and it also serves (together with other Cluster Servers) a set of Shared Domains.

To use CommuniGate Pro servers in a Cluster, you need a special CommuniGate Pro [Cluster License](#).

Please read the [Scalability](#) section first to learn how to estimate your mail server load, and how to get most out of each CommuniGate Pro Server running your multi-server (Cluster) site.

Cluster Types

There are two main types of Cluster configurations: *Static* and *Dynamic*.

Each Account in a Shared Domain served with a Static Cluster is created (hosted) on a certain Server, and only that Server can access the account data directly. When a Static Cluster Server

needs to perform any operation with an account hosted on a different Server, it establishes a TCP/IP connection with the account Host Server and accesses account data via that Host Server. This architecture allows you to use local (i.e. non-shared) storage devices for account data.

Note: some vendors have "Mail Multiplexor"-type products. Those products usually implement a subset of Static Cluster frontend functionality.

Accounts in Shared Domains served with a Dynamic Cluster are stored on a shared storage, so each Cluster Server (except for Frontend Servers, see below) can access the account data directly. At any given moment, one of the Cluster Servers acts as a Cluster Controller synchronizing access to Accounts in Shared Domains. When a Dynamic Cluster Server needs to perform any operation with an account currently opened on a different Server, it establishes a TCP/IP connection with that "current host" Server and accesses account data via that Server. This architecture provides the highest availability (all accounts can be accessed as long as at least one Server is running), and does not require file-locking operations on the storage device.

Supported Services

The CommuniGate Pro Clustering features support the following services:

- POP3 access
- IMAP access
- WebUser Interface access
- Personal Web Sites (including uploading)
- ACAP access
- PWD access and remote administration
- SMTP mail receiving
- SMTP mail delivery

The WebUser Interface module maintains user sessions even if subsequent page requests come to the Backend Server through different Frontend Servers.

Frontend Servers

Clusters of both types are usually equipped with *Frontend Servers*. Frontend Servers cannot access Account data directly - they always open connections to other (backend) Servers to perform any operation with Account data.

Frontend servers accept TCP/IP connections from client computers (usually - from the Internet). In a pure Frontend-Backend configuration no Accounts are created on any Frontend Server, but nothing prohibits you from serving some Domains (with Accounts and mailing lists) directly on the

Frontend servers.

When a client establishes a connection with one of the Frontend Servers and sends the authentication information (the Account name), the Frontend server detects on which Backend server the addressed Account can be opened, and establishes a connection with that Backend Server.

The Frontend Servers:

- handle all SSL/TLS encryption/decryption operations
- handle most of the SMTP relaying operations themselves
- virtually eliminate inter-server communications between Backend Servers, and (in Dynamic Clusters) provide second-level load balancing
- provide an additional layer of protection against Internet attacks and allow you to avoid exposing Backend Servers to the Internet
- smooth out the external traffic (soften peaks in the site load), and protect the Backend Servers from the Denial-of-Service attacks

If the Frontend Servers are directly exposed to the Internet, and the security of a Frontend Server operating system is compromised so that someone gets unauthorized access to that Server OS, the security of the site is not totally compromised. Frontend Servers do not keep any Account information (mailboxes, passwords) on their disks. The "cracker" would then have to go through the firewall and break the security of the Backend Server OS in order to get access to any Account information. Since the network between Frontend and Backend Servers can be disabled for all types of communications except the CommuniGate Pro inter-server communications, breaking the Backend Server OS is virtually impossible.

Both Static and Dynamic Clusters can work without dedicated Frontend Servers. This is called a *symmetric configuration*, where each Cluster Server implements both Frontend and Backend functions.

In the example below, the domain1.dom and domain2.dom domain Accounts are distributed between three Static Cluster Servers, and each Server accepts incoming connections for these domains. If the Server SV1 receives a connection for the account kate@domain1.dom located on the Server SV2, the Server SV1 starts to operate as a Frontend Server, connecting to the Server SV2 as the Backend Server hosting the addressed Account.

At the same time, an external connection established with the server SV2 can request access to the ada@domain1.dom account located on the Server SV1. The Server SV2 acting as a Frontend Server will open a connection to the Server SV1 and will use it as the Backend Server hosting the addressed account.

In a symmetric configuration, the number of inter-server connections can be equal to the number of external (user) access-type (POP, IMAP, HTTP) connections. For a symmetric Static Cluster, the

average number of inter-server connections is $M*(N-1)/N$, where M is the number of external (user) connections, and the N is the number of Servers in the Static Cluster. For a symmetric Dynamic Cluster, the average number of inter-Server connections is $M*(N-1)/N * A/T$, where T is the total number of Accounts in Shared Domains, and A is the average number of Accounts opened on each Server. For large ISP-type and portal-type sites, the A/T ratio is small (usually - not more than 1:100).

In a pure Frontend-Backend configuration, the number of inter-server connections is usually the same as the number of external (user) connections: for each external connection, a Frontend Server opens a connection to a Backend Server. A small number of inter-server connections can be opened between Backend Servers, too.

Withdrawing Frontend Servers from a Cluster

To remove a Frontend Server from a Cluster (for maintenance, hardware upgrade, etc.), reconfigure your Load Balancer or the round-robin DNS server to stop redirection of incoming requests to this Frontend Server address. After all current POP, IMAP, SMTP sessions are closed, the Frontend Server can be shut down. Since the WebMail sessions do not use persistent HTTP connections, a Frontend Server in a WebMail-only Cluster can be shut down almost immediately.

Access to all Shared Domain Accounts is provided without interruption as long as at least one Frontend Server is running.

If a Frontend server fails, no Account becomes unavailable and no mail is lost. While POP and IMAP sessions conducted via the failed Frontend server are interrupted, all WebUser Interface session remain active, and WebUser Interface clients can continue to work via remaining Frontend Servers. POP and IMAP users can immediately re-establish their connections via remaining Frontend Servers.

If the failed Frontend server cannot be repaired quickly, its Queue can be processed with a different server, as a [Foreign Queue](#).

Cluster Server Configuration

This section specifies how each CommuniGate Pro Server should be configured to participate in a Static or Dynamic Cluster. These settings control inter-server communications in your Cluster.

First, install CommuniGate Pro Software on all Servers that will take part in your Cluster. Specify the Main Domain Name for all Cluster Servers. Those names should differ in the first domain name element only:

back1.isp.dom, back2.isp.dom, front1.isp.dom, front2.isp.dom, etc.

Remember that Main Domains are never shared, so all these names should be different. You may want to create only the Server administrator accounts in the Main Domains - these accounts can be used to connect to that particular Server and configure its local, Server-specific settings.

Use the WebAdmin Interface to open the Settings->General->Cluster page on each Backend Server, and enter all Frontend and Backend Server IP addresses. Backend CommuniGate Pro Servers will accept Cluster connections from the specified IP addresses only. If the Frontend Servers use dedicated Network Interface Cards (NICs) to communicate with Backend Servers, specify the IP addresses the Frontend Servers have on that internal network:

Local Cluster Address	
On restart:	Active: [192.168.0.5]
Backend Server Addresses	
Frontend Server Addresses	

Local Cluster Address

This setting specifies the local network address this server will use to communicate with other servers in the Cluster. Connections to other servers will be established from this IP address. This address is used as this server "name", identifying the server in the Cluster.

If the `first` IP value is selected, the server selects the first address from the list of server Local IP Addresses.

If you change this setting value, the new value will be in effect only after server restart.

Backend Port	Cache	Backend Port	Cache
Cluster:		SMTP:	
POP:		IMAP:	
ACAP:		PWD:	
HTTP User:		HTTP Admin:	

If your Backend Servers use non-standard port numbers for mail services, change the Backend Server Ports values.

For example, if your Backend Servers accept [WebUser Interface](#) connections not on the port number 8100, but on the standard HTTP port 80, set 80 in the HTTP User field and click the Update button.

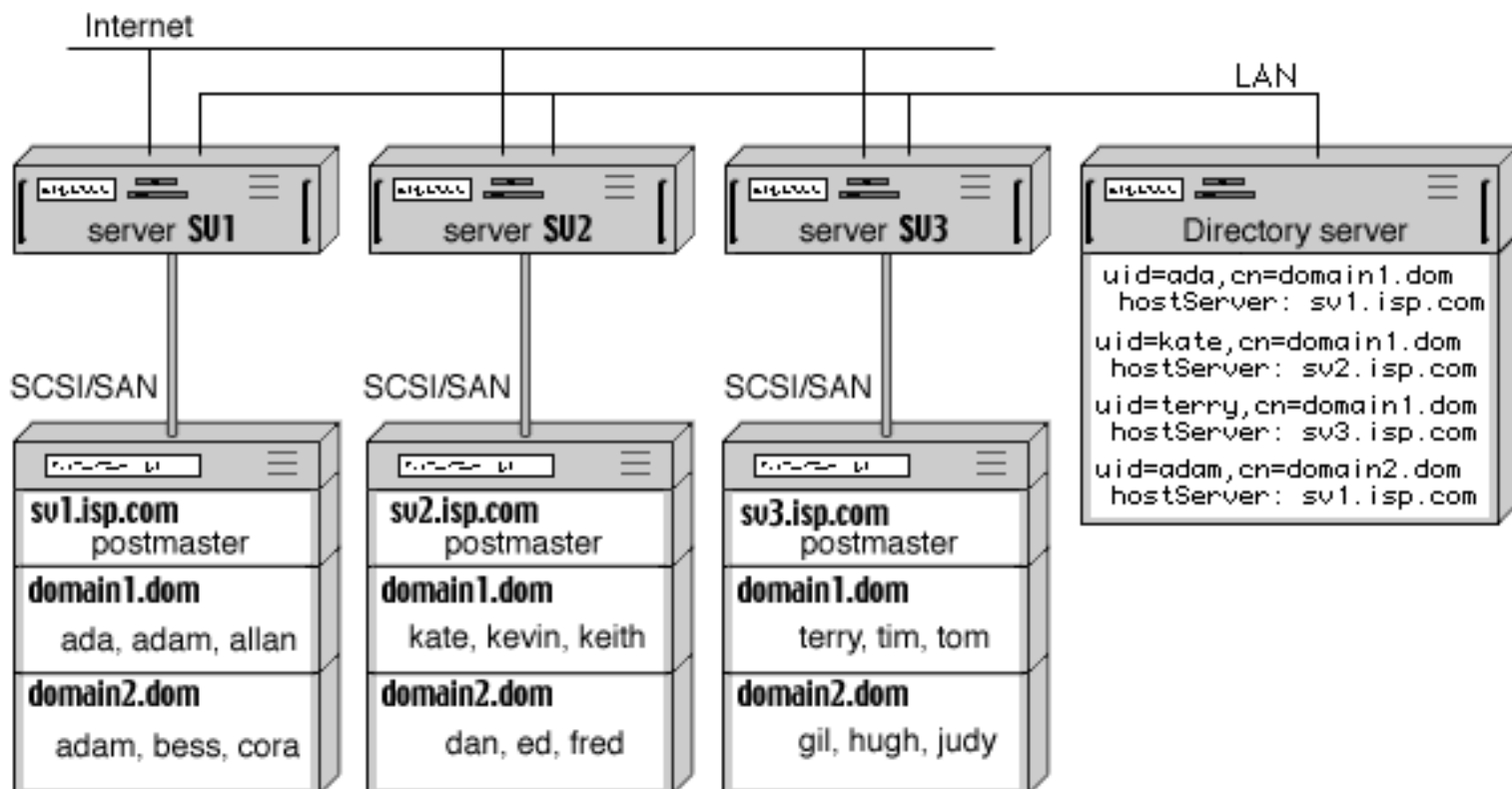
For certain services, CommuniGate Pro can reuse inter-server connections. Instead of closing a connection when an operation is completed, it is placed into an internal cache, and reused later, when this server needs to connect to the same server. The Cache parameter specifies the size of that connection cache. If there are too many connections in the cache, older connections are closed and pushed out of the cache.

Cluster members use the PWD protocol to perform operations remotely, on other cluster members. The port number they need to connect to on other Cluster members is specified with the Cluster port setting. Usually it is the same as the PWD port setting.

Servers in a Dynamic Cluster use other members SMTP modules to do remote message delivery (though the protocol between the servers is not the SMTP protocol). Use the SMTP port setting to specify the port number used with SMTP modules on other cluster members.

Static Clusters

Shared Domains in a Static Cluster are created in exactly the same manner as regular CommuniGate Pro Domains. Each Server in a Static Cluster contains a subset of all Shared Domain Accounts. As a result, each Shared Domain Account has a "Host Server". Only the Host Server needs access to the Account data, so Static Clusters can use regular, non-shared disk storage. Static Clusters rely on some method that allows each Cluster Server to learn the name of the Host Server for any Shared Domain account. This type of routing can be implemented using a shared Directory Server, in the same way it is implemented for [Distributed Domains](#):



Backend and Frontend Server Settings

The CommuniGate Pro Static Cluster setup is an extension of the [Distributed Domains](#) configuration.

- Install and [configure](#) CommuniGate Pro Software on all Servers that will take part in a Static Cluster.
- Configure all Servers to use one [Shared Directory](#) for all Shared Domains.
- Create Shared Domains on all Servers (Backend and Frontend), in the same way regular, non-shared Domains are created.
- Use the WebAdmin Interface to open the Settings->General->Cluster page on each Server, and enter the names (Main Domain Names) of all Backend Servers and the IP addresses of those Servers:

Static Clustering	
Static Member Name	Member Address

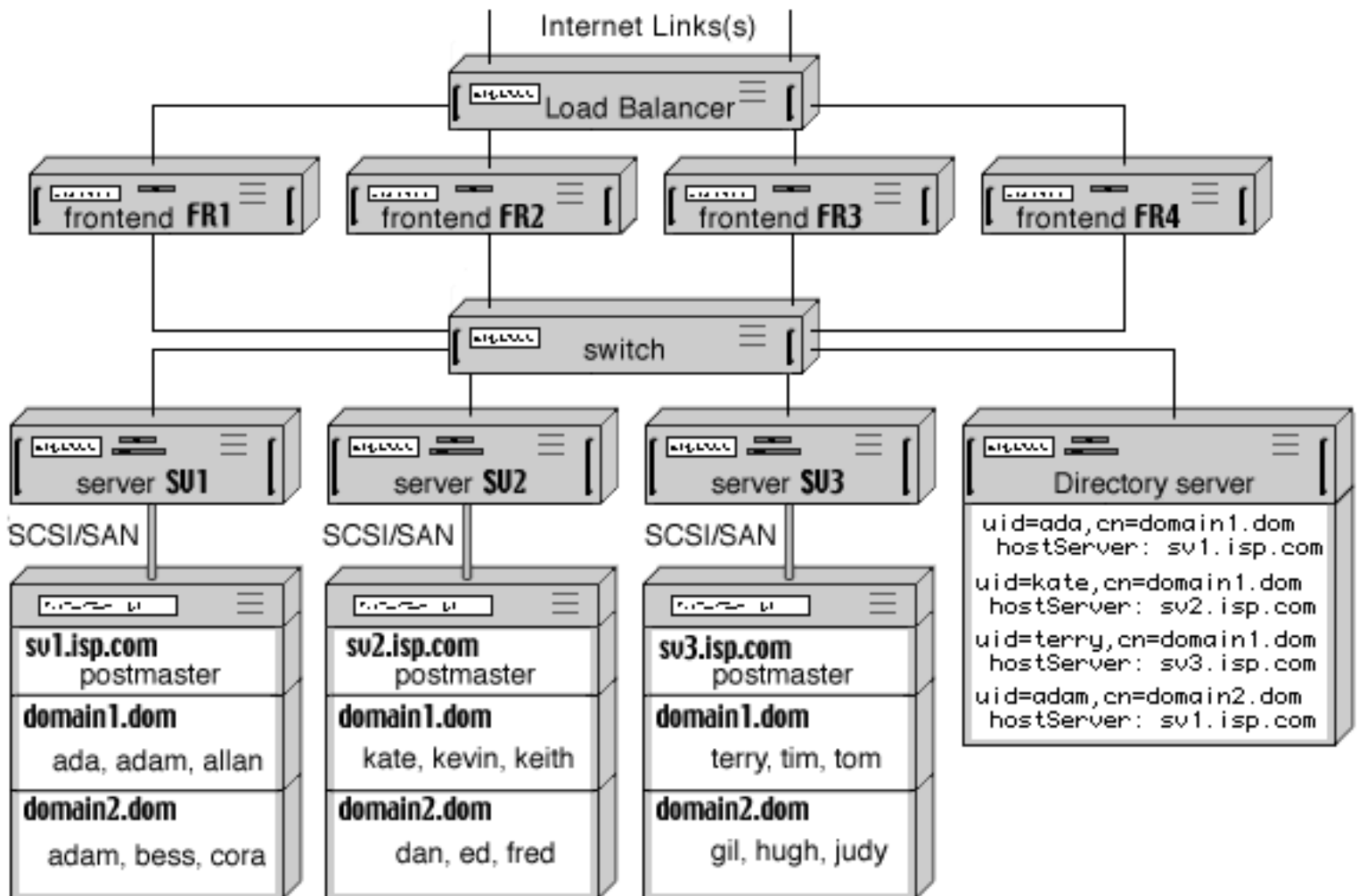
If an address is routed to a domain listed in this table, the CommuniGate Pro Server uses its Clustering mechanism to connect to the Backend server at the specified address and performs the

requested operations on that Backend server.

The logical setup of the Backend and Frontend Servers is the same - you simply do not create Shared Domain Accounts on any Frontend Server, but create them on your Backend Servers.

Computers in a Static Cluster can use different operating systems.

A complete Frontend-Backend Static Cluster configuration uses Load Balancers and several separate networks:



In a simplified configuration, you can connect Frontend Servers directly to the Internet, and balance the load using the DNS *round-robin* mechanism. In this case, it is highly recommended to install a firewall between Frontend and Backend Servers.

Adding Server to a Static Cluster

You can add Frontend and Backend Servers to a Static Cluster at any time.

To add a Server to a Static Cluster:

- Properly configure the Server (see above): configure it to access the Shared Directory, create Shared Domains, and set the Clustering Settings.
- Add the IP address of the new Server to the Backend or Frontend Addresses tables of other Cluster Members (if you have specified proper network address ranges for those tables, this step is not needed).
- If the new Server is a Backend one, add its name and IP Address to the Static Clustering tables on other Servers.

After a new Frontend Server is configured and added to the Static Cluster, reconfigure the Load Balancer or the round-robin DNS server to direct incoming requests to the new Server, too.

After a new Backend Server is configured and added to the Static Cluster, you can start creating Accounts in its Shared Domains.

Withdrawing a Server from a Static Cluster

If you decide to shut down a Static Cluster Backend Server, all Accounts hosted on that Server become unavailable. Incoming messages to unavailable Accounts will be collected in the Frontend Server queues, and they will be delivered as soon as the Backend Server is added back or these Accounts become available on a different Backend Server (see below).

Backend Failover in a Static Cluster

If a Backend Server in a Static Cluster is shut down, all Accounts hosted on that Server become unavailable (there is no interrupt in service for Accounts hosted on other Backend Servers).

To restore access to the Accounts hosted on the failed Server, its Account Storage should be connected to any other Backend server. You can either:

- physically connect the disk storage to some other Backend Server;
- use dual-access RAID devices and tell the sibling Server to take over that device;
- use a file server partition or file directory for each Backend Account Storage, and mount that directory on some other Backend Server in case of a Backend Server failure.

After a sibling Backend server gets physical access to Account Storage of the failed server, you should modify the Directory so all Servers will contact the new "home" for Accounts in that Storage. This can be done by an LDAP utility that modifies all records in the [Domains Subtree](#) that contain the name of the failed Server as the `hostServer` attribute value. The utility should set the attribute value to the name of the new Host Server, and should add the `oldHostServer` attribute with the name of the original Host Server. This additional attribute will allow you to restore the

hostServer attribute value after the original Host Server is restored and the Account Storage is reconnected to it. If the CommuniGate Pro is used as the site Directory Server, 100,000 Directory records can be modified within 1-2 minutes.

Dynamic Clusters

The Static Clusters described above can be used to handle extremely large (practically unlimited) Internet sites, providing 24x7 site access. In a rare case of a Backend Server failure, the Static Cluster continues to operate and access to accounts on the failed Server can be restored within 2-10 minutes (depending on how easily the disk storage can be reassigned and how fast the Routing tables/Directory can be updated).

If it is necessary to provide 100% site uptime and 24x7 access to all Accounts even when some of the Backend Servers fail, the Dynamic Cluster should be deployed.

The main difference between Static and Dynamic Clusters is the account hosting. While each account in a Static Cluster has its Host Server, and only that Server can access the Account data directly, all Backend Servers in a Dynamic Cluster can access the Account data directly. The most common method to implement a Dynamic Cluster shared Account Storage is employing dedicated File Servers or using a Cluster File System.

See the [Shared File Systems](#) document to learn more about various types of Shared File Systems available.

Traditional File-Locking Approach

Many legacy mail servers can employ file servers for account storage. Since those servers are usually implemented as multi-process systems (under Unix), they use the same synchronization methods in both single-server and multi-server environments: *file locks* implemented on the Operating System/File System level.

This method has the following problems:

- Every operation with account/mailbox data should be surrounded with file locking/unlocking operations, and additional File System operations are needed to ensure data consistency. As a result, the number of File System operations increases in 3-5 times, and (since the speed of file operation usually defines the speed of the site) the site performance suffers a lot.
- Modern File Servers either do not support file locking mechanisms at all, or provide severely limited versions of those mechanisms, making the most important site component -

account storage - unreliable and not fault-tolerant.

- Malfunction of one of the servers can bring the entire site down (because of deadlocks), and makes fault recovery extremely painful.
- Simultaneous access to the same account/mailbox by several clients is either prohibited or unreliable.

In the attempt to decrease the negative effect of file-locking, some legacy mail servers support the MailDir mailbox format only (one file per message), and they rely on the "atomic" nature of file directory operations (rather than on file-level locks). This approach theoretically can solve some of the outlined problems (in real-life implementations it hardly solves any), but it results in wasting most of the file server storage: many high-end file servers use 64Kbyte blocks for files, while an average mail message size is about 4Kb, and storing each message in a separate file results in wasting more than 90% of the file server disk space, and overloads file server internal file tables. Also, performance of File Servers severely declines when an application uses many smaller files instead of few larger files.

While simple clustering based on Operating System/File System multi-access capabilities works fine for Web servers (where the data is not modified too often), it does not work well for Mail servers where the data modification traffic is almost the same as the data retrieval traffic.

Simple Clustering does not provide any additional value (like Single Service Image), so administering a 10-Server cluster is more difficult than administering 10 independent Servers.

The CommuniGate Pro software supports the [External INBOX](#) feature, so a file-based clustering can be implemented with the CommuniGate Pro, too. But because of the problems outlined above, it is highly recommended to avoid this type of solutions and use the real CommuniGate Pro Dynamic Cluster instead.

Cluster Controller

CommuniGate Pro Servers in a Dynamic Cluster do not use Operating System/File System locks to synchronize Account access operations. Like in a Static Cluster, only one Server in a Dynamic Cluster has direct access to any given Account at any given moment. All other Servers work through that Server if they want to access the same Account. But this assignment is not static: any Server can open any Account directly if that Account is not opened with some other Server.

This architecture provides the maximum uptime: if a Backend Server fails, all Accounts can be accessed via other Backend Servers - without any manual operator intervention, and without any downtime. The site continues to operate and provide access to all its Accounts as long as at least one Backend Server is running.

One of the Backend Servers in a Dynamic Cluster acts as the *Cluster Controller*. It synchronizes all other Servers in the Cluster and executes operations such as creating Shared Domains, creating and removing accounts in the shared domains, etc. The Cluster Controller also provides the *Single*

Service Image functionality: not only a site user, but also a site administrator can connect to any Server in the Dynamic Cluster and perform any Account operation (even if the Account is currently opened on a different Server), as well as any Domain-level operations (like Domain Settings modification), and all modifications will be automatically propagated to all Cluster Servers.

Note: most of the Domain-level update operations, such as updating Domain Settings, Default Account Settings, WebUser Interface Settings, and Domain-Level Alerts may take up to 30 seconds to propagate to all Servers in the Cluster. Account-Level modifications come into effect on all Servers immediately.

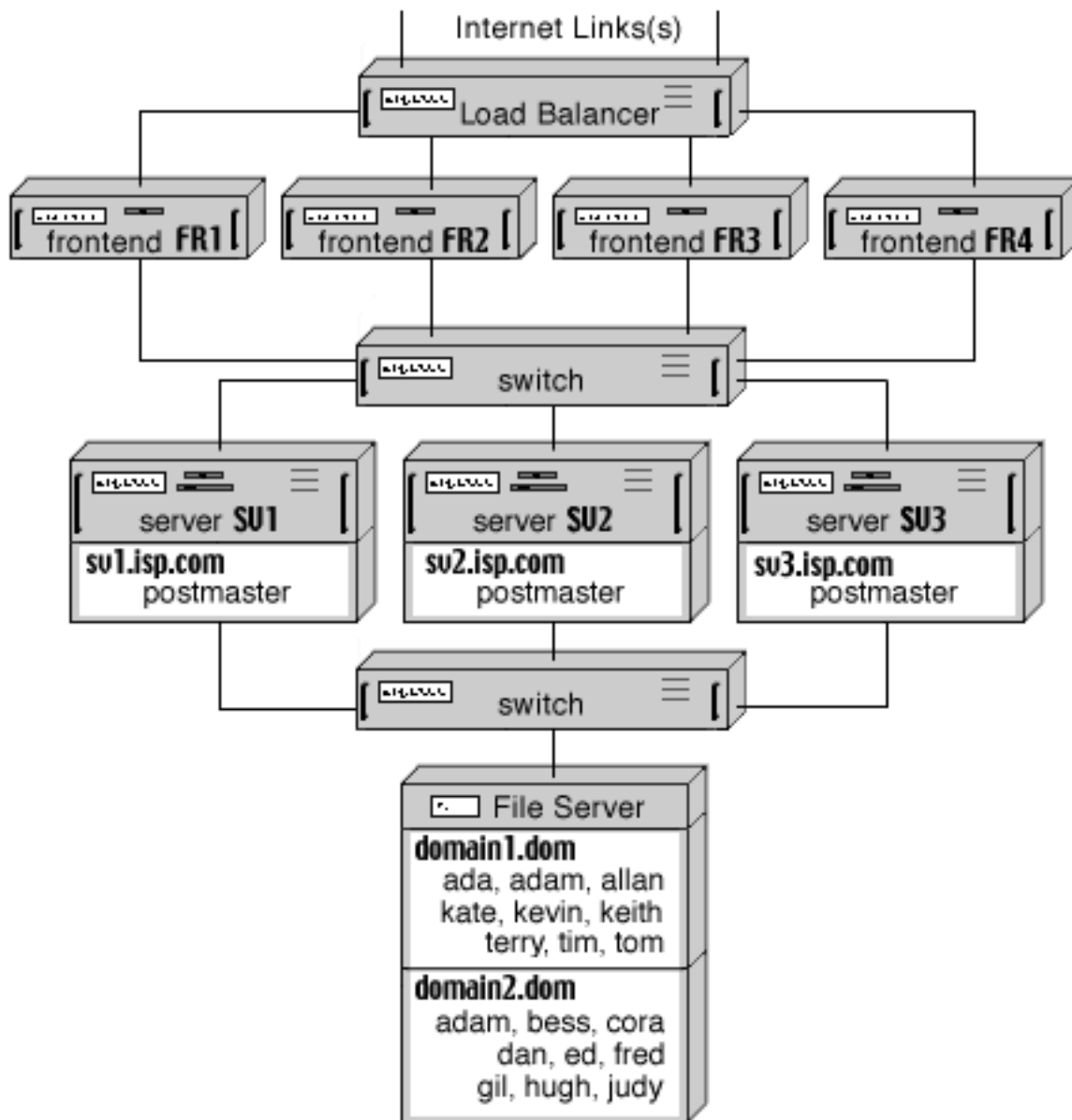
The Cluster Controller collects the load level information from the Backend Servers. When a Frontend Server receives a session request for an Account not currently opened on any Backend Server, the Controller directs the Frontend Server to the least loaded Backend Server. This second-level load balancing for Backend Server is based on actual load levels and it supplements the basic first-level Frontend load balancing (DNS round-robin or traffic-based).

When a Dynamic Cluster has at least 2 backend Servers, the Cluster Controller assigned the Controller Backup duties to one of the other backend Servers. All other Cluster members maintains connections with the Backup Controller. If the Backup Controller fails, some other backend Server is selected as a Backup Controller.

If the main Controller fails, the Backup Controller becomes the Cluster Controller. All Servers send the resynchronisation information to the Backup Controller and the Cluster continues to operate without interruption.

While the Dynamic Cluster can maintain a Directory with Account records, the Dynamic Cluster functionality does not rely on the Directory. If the Directory is used, it should be implemented as a [Shared Directory](#).

A complete Frontend-Backend Dynamic Cluster configuration uses Load Balancers and several separate networks:



Since all Backend Servers in a Dynamic Cluster have direct access to Account data, they should run the operating systems using the same EOL (end-of-line) conventions. This means that all Backend Servers should either run the same or different flavors of the Unix OS, or they all should run the same or different flavors of the MS Windows OS. Frontend Servers do not have direct access to the Account data, so you can use any OS for your Frontend Servers (for example, a site can use the Solaris OS for Backend Servers and Microsoft Windows 2000 for Frontend Servers).

Cluster File Systems and Cluster OSes

Some of the modern Operating Systems (Tru64®, UnixWare®) provide advanced Clustering capabilities themselves. Most of those Cluster features are designed to help porting "regular", non-clustered applications on these Cluster platforms. But some features provided with those Cluster OSes are extremely useful for the CommuniGate Pro Dynamic Cluster implementations:

- Cluster File System
- IP Aliasing

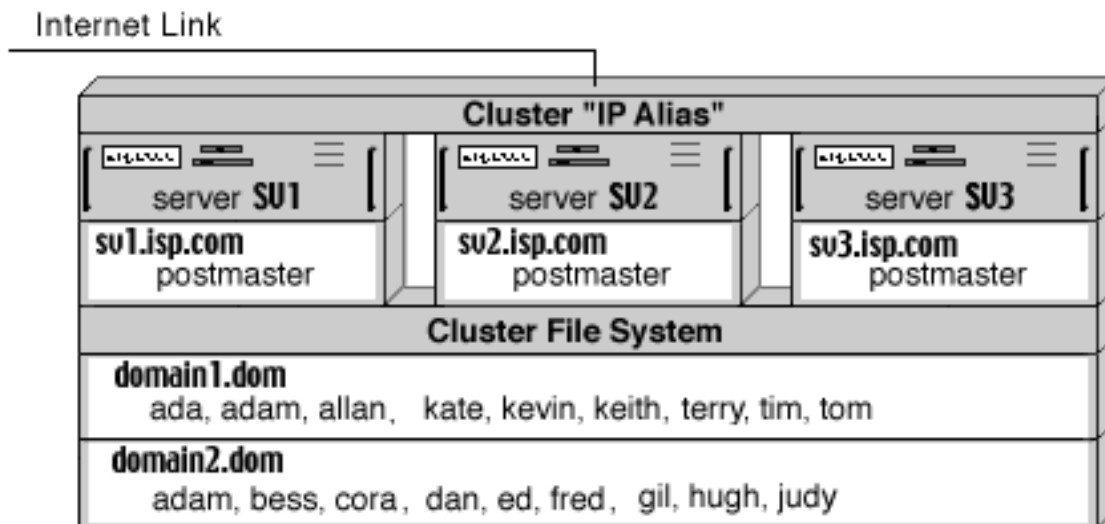
A Cluster File System allows all Servers in an OS Cluster to mount and use the same file system(s) on shared devices. Unlike Network File Systems (NFS), Cluster File Systems do not require a dedicated server on the network. Cluster File Systems can utilize multiple SCSI connections provided with some high-end SCSI storage devices, and they can allow each Server to exchange the data directly with storage devices via a fibre SAN (Storage Area Network). To ensure file system integrity, Cluster File Systems use high-speed server interconnects.

The SAN protocols are very effective for file transfers, and Cluster File Systems can provide better performance than Network File Systems.

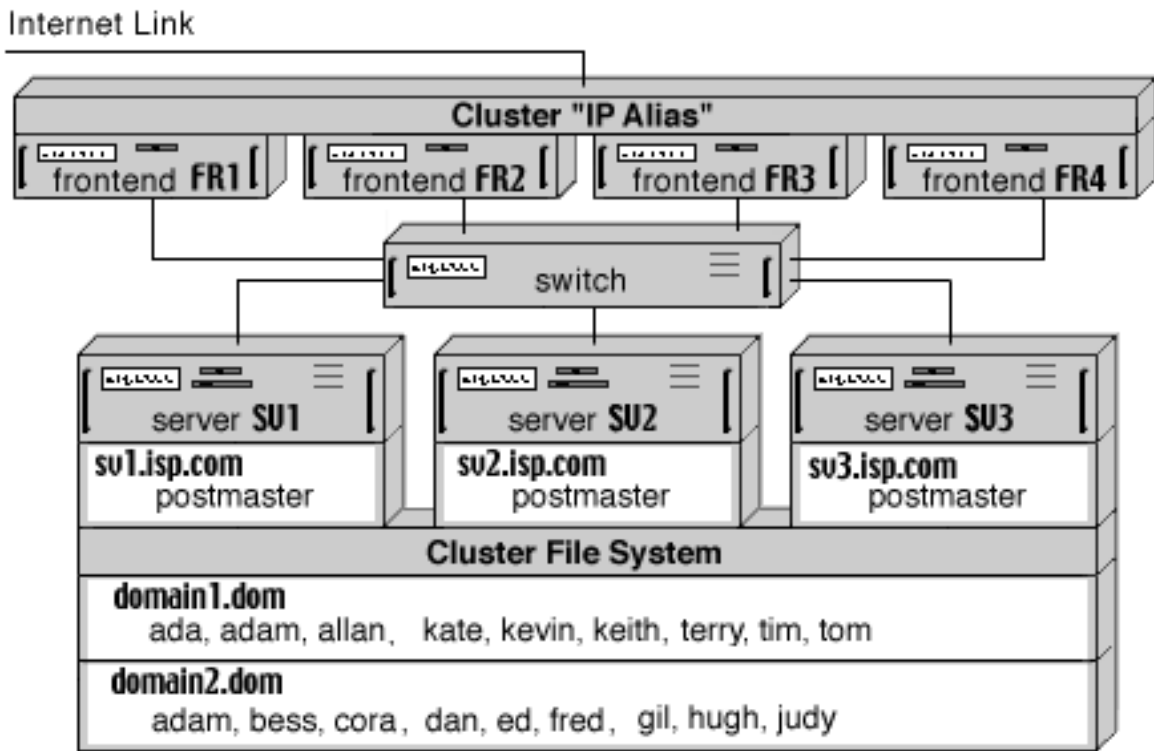
The Cluster File Systems can also provide better reliability than single-server NFS solutions (where the NFS server is a single point of failure).

The IP Aliasing feature allows the Cluster OS to distribute the network load between Cluster Servers without an additional Load Balancer unit.

A "backend-only" CommuniGate Pro Dynamic Cluster can utilize both features of a Cluster OS: the IP Aliasing is used to distribute the load between CommuniGate Pro Server, and CommuniGate Pro Servers use the Cluster File System to store all account data in shared Domains:



A Cluster OS can be used in a frontend/backend CommuniGate Pro Cluster configuration, too. In this case, one OS Cluster is used for CommuniGate Pro frontend Servers, utilizing the IP Aliasing load balancing, and the second OS Cluster is used for CommuniGate Pro backend Servers, where the Cluster File System is employed:



The Configuration of the CommuniGate Pro Dynamic Cluster does not depend on the type of the load balancing used (separate Load Balancers or IP Aliases), or on the type of the shared file system used (Network File System or Cluster File System).

Configuring Backend Servers

- Install and [configure](#) CommuniGate Pro Software on all Servers that will take part in a Dynamic Cluster.
- Open the WebAdmin Settings->Access page and modify the PWD service settings. Each Cluster member (Backend and Frontend) opens 2 PWD connections to the Cluster Controller, so the maximum number of channels should be increased at least by $2 * (\text{number of Backend servers} + \text{number of Frontend servers})$. Since additional PWD connections can be opened by Frontend and Backend servers to serve administrator and user requests, it is better to increase the number of channels by: $5 * (\text{number of Backend servers}) + 3 * (\text{number of Frontend servers})$.
- Open the WebAdmin Settings->General->Clusters page and enter the IP addresses of all backend and frontend Servers in the Cluster.
- Stop all Servers.
- Create a file directory that will contain Shared Domains. You should create that file directory on a storage unit that will be available for all Cluster Backend Servers (on a file server, for example). Place a link to that directory into the CommuniGate Pro *base directory*, and name that link *SharedDomains*. Make sure that all Backend Servers have all file access rights to create, remove, read, and modify files and directories inside the *SharedDomain* directory.

Note: if creating symbolic links is problematic (as it is on MS Windows platforms), you should specify the location of the "mounted" file directory as the `--SharedBase` [Command Line Option](#):

`--SharedBase H:\Base`

- If you are upgrading from a single-server configuration, you may want to make some of your existing domains shared, so they will be served with the entire Cluster. In this case you should move the domain file directory from the `{base}/Domains` file directory into the `{base}/SharedDomains` file directory (located on a shared storage unit).
- Start the first Backend Server. You can use the "controller" parameter for the startup script, or start the server manually, with the `--ClusterController` Command Line Option.

Use the WebAdmin Interface of this first Backend Server to verify that the Cluster Controller is running. Open the Domains page to check that:

- all domains you have placed into the SharedDomains directory are visible;
- the Create Domain button is now accompanied with the Create Shared Domain button.

Use the Create Shared Domain button to create additional Shared Domains to be served with the Dynamic Cluster.

When the Cluster Controller is running, the site can start serving clients (if you do not use Frontend Servers). If your configuration employs Frontend servers, at least one Frontend Server should be started (see below).

Adding a Backend Server to a Dynamic Cluster

Additional Backend Server can be added to the Cluster at any moment. They should be pre-configured in the exactly the same way as the first Backend Server was configured.

To add a Backend Server to your Dynamic Cluster, start it with the `--ClusterMember` Command Line option (it can be added to the CommuniGatePro startup script). The Server will poll all specified Backend Server IP Addresses until it finds the active Cluster Controller.

Use the WebAdmin interface to verify that the Backend Server is running. Use the Domains page to check that all Shared Domains are visible and that you can administer Accounts in the Shared Domains.

When the Cluster Controller and at least one Backend Server are running, they both can serve all accounts in the Shared Domains. If you do not use Frontend Servers, load-balancing should be implemented using a regular load-balancer switch, DNS round-robin, or similar technique that

distributes incoming requests between all Backend Servers.

Adding a Frontend Server to a Dynamic Cluster

You can add additional Frontend servers to the Cluster at any moment.

Install and [Configure](#) the CommuniGate Pro software on a Frontend Server computer. Since Frontend Servers do not access Account data directly, there is no need to make the SharedDomains file directory available ("mounted" or "mapped") to any Frontend Server.

Specify the addresses of all Backend Servers using the Frontend Server Settings->General->Cluster WebAdmin page.

To add a Frontend Server to your Dynamic Cluster, stop it, and restart it with the --ClusterFrontend Command Line option (it can be added to the CommuniGatePro startup script). The Server will poll all specified Backend Server IP Addresses until it finds the active Cluster Controller.

Use the WebAdmin interface to verify that the Frontend Server is running. Use the Domains page to check that all Shared Domains are visible.

When Frontend Servers try to open one of the Shared Domain accounts, the Controller directs them to one of the running Backend Servers, distributing the load between all available Backend Servers.

Shared Settings

The Dynamic Cluster maintains a separate set of "Default settings" for Shared Domains. These settings include:

- Default Domain Settings for all Shared Domains
- Default Account Settings and Default WebUser Preferences for all Accounts in Shared Domains
- Cluster-Wide Alerts - these alerts are sent to all Accounts in Shared Domains

When the Server Administrator uses the WebAdmin Interface to modify these settings, the WebAdmin pages display the links that allow the Administrator to switch between the Server-wide settings (that work for all non-Shared Domains), and Cluster-wide settings. The Cluster-wide settings are automatically updated on all Cluster Members, and they work for all Shared Domains.

The Cluster-wide settings also include:

- [Cluster-wide Rules](#).

- [Cluster-wide Router Table](#).
- [Cluster-wide Directory Integration Settings](#).
- [Cluster-wide Protection Settings](#).
- Default and Named Cluster-wide [WebSkins](#). These WebSkins are used as default Skins for WebSkins in Shared Domains.

Withdrawing Servers from a Dynamic Cluster

If a Backend Server fails, all Shared Domain Accounts that were open on that Server at the time of failure become unavailable. They become available again within 10-20 seconds, when the Cluster Controller detects the failure. A Backend Server failure does not cause any data loss.

Upgrading Servers in a Dynamic Cluster

The Dynamic Cluster is designed to support "rolling upgrades". To upgrade to a newer version of the CommuniGate Pro software, you should upgrade the servers one-by-one: withdraw a server from the Cluster, upgrade the software, and add the server back to the Cluster. This procedure allows your site to operate non-stop during the upgrade.

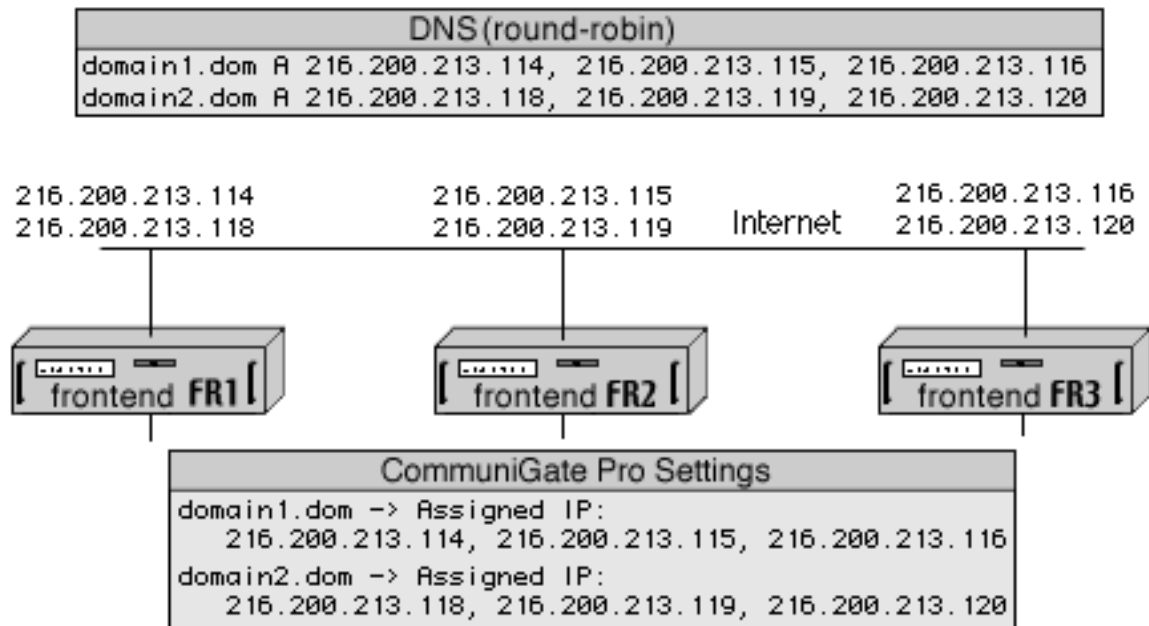
Certain changes in CommuniGate Pro software can impose some restrictions on the "rolling upgrade" process. Always check the [History](#) section before you upgrade your Cluster, and see if any Cluster Upgrade restrictions are specified there.

Assigning IP Addresses to Shared Domains

A CommuniGate Pro Cluster can serve several Shared Domains. If you plan to provide POP and IMAP access to Accounts in those Domains, you may want to assign dedicated IP addresses to those Domains to simplify client mailer setups. See the [Access](#) section for more details.

If you use Frontend Servers, only Frontend Servers should have dedicated IP Addresses for Shared Domains. Inter-server communications always use full account names (*accountname@domainname*), so there is no need to dedicate IP Addresses to Shared Domains on Backend Servers.

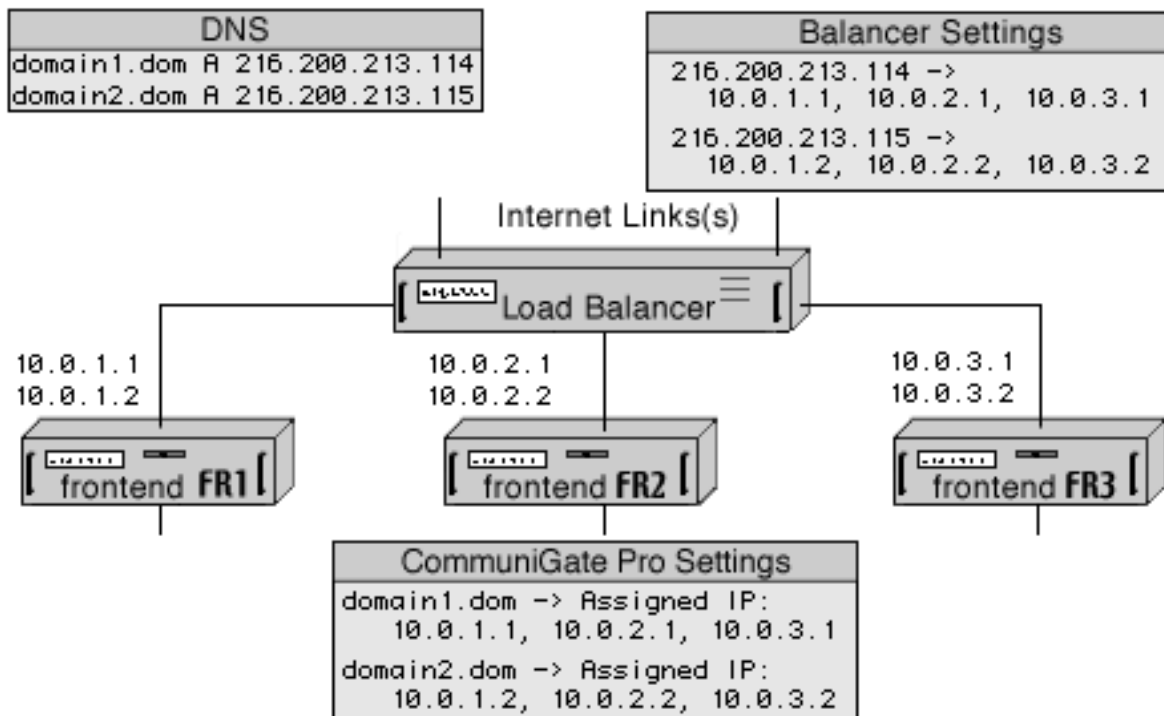
If you use the DNS round-robin mechanisms to distribute the site load, you need to assign N IP addresses to each Shared Domain that needs dedicated IP addresses, where N is the number of your Frontend Servers. Configure the DNS Server to return these addresses in the round-robin manner:



In this example, the Cluster is serving two Shared Domains: domain1.dom and domain2.dom, and the Cluster has three Frontend Servers. Three IP addresses are assigned to each domain name in the DNS server tables, and the DNS server returns all three addresses when a client is requesting A-records for one of these domain names. Each time the DNS server "rotates" the order of the IP addresses in its responses, implementing the DNS "round-robin" load balancing (client applications usually use the first address in the DNS server response, and use other addresses only if an attempt to establish a TCP/IP connection with the first address fails).

When configuring these Shared Domains in your CommuniGate Pro Servers, you assign all three IP addresses to each Domain.

If you use a Load Balancer to distribute the site load, you need to place only one "external" IP address into DNS records describing each Shared Domain. You assign one "virtual" (LAN) IP address to each Shared Domain on each Frontend Server:



In this example, the Cluster is serving two Shared Domains: domain1.dom and domain2.dom, and the Cluster has three Frontend Servers. One IP Addresses assigned to each Shared Domain in the DNS server tables, and those addresses are external (Internet) addresses of your Load Balancer. You should instruct the Load Balancer to distribute connections received on each of its external IP addresses to three internal IP addresses - the addresses assigned to your Frontend Servers.

When configuring these Shared Domains in your CommuniGate Pro Servers, you assign these three internal IP addresses to each Domain.

DNS MX-records for Shared Domains can point to their A-records.

Security Issues

The Frontend-Backend topology allows you to protect the site information and Backend Servers not only when a Frontend Server crashes because of some type of network attack, but even if the Frontend Server OS is "cracked" and an intruder gets the complete ("root") access to the Frontend Server OS using a security hole in that OS.

To protect the site from these "cracks":

- Do not use the Frontend Servers to administer the Shared Domains as a Frontend Server administrator. In this case you can disable the Admin Connections option on the Cluster page of all Backend Servers.

- Enable the Admin Connections option on the current Cluster Controller Server only when you add a new Frontend server to the Cluster. When that Frontend server is up and running, disable the Admin Connections option on the Cluster Controller.

These measures do not cause any problem for your users that have the domain administrator rights and want to administer their Shared Domains (using WebAdmin Interface or CLI). They also do not cause any problem for your regular users that want to use the PWD module to update their passwords.

Cluster Configuration Details

Listeners

To protect your site from DoS attacks, you may want to open SMTP, POP, IMAP, and other [Listeners](#) and limit the number of connections accepted from the same IP address. Set those limits on frontend servers only, since backend servers receive all connections from frontends, and each frontend can open a lot of connections from the same IP address.

WebAdmin

Usually the Backend servers are not directly accessible from the Internet. If you need to change the settings or monitor one of the Backend servers from "outside", you can use the WebAdmin interface of one of the frontend servers, using the following URL:

`http://frontendaddress:8010/Cluster/12.34.56.78/`

where 12.34.56.78 is the [internal] IP address of the backend server you want to access.

SMTP

The outgoing mail traffic generated with regular (POP/IMAP) clients is submitted to the site using the A-records of the site Domains. As a result, the submitted messages go to the Frontend Servers and the messages are distributed from there.

Messages generated with WebUser clients and messages generated automatically (using the Automated Rules) are generated on the Backend Servers. Since usually the Backend servers are behind the firewall and since you usually do not want the Backend Servers to spend their resources maintaining SMTP queues, it is recommended to use the forwarding feature of the CommuniGate Pro [SMTP module](#).

Select the Forward to option and specify the domain name that resolves into the IP addresses of all (or some) Frontend Servers. In this case all mail generated on the Backend Server will be quickly sent to the Frontend Servers and it will be distributed from there. You can also list the IP addresses of all (or some) Frontend Servers, separating them with the comma sign.

RPOP

RPOP activity takes place on backend servers in a Static Cluster, and on the Cluster Controller in a Dynamic Cluster. As a result, it is essential for those servers to be able to initiate outgoing TCP connections to remote servers. If the Backend servers are connected to a private LAN behind a firewall, you should install some NAT server software on that network and configure the Backend servers (using their OS TCP/IP settings) to route all non-local packets via the NAT server(s). Frontend servers can be used to run NAT services.

FTP

The FTP module does not "proxy" connections to backend servers. Instead, it uses CLI to manipulate with Account Personal Web Sites on backend servers. This eliminates a problem of backend servers opening FTP connections directly to clients.

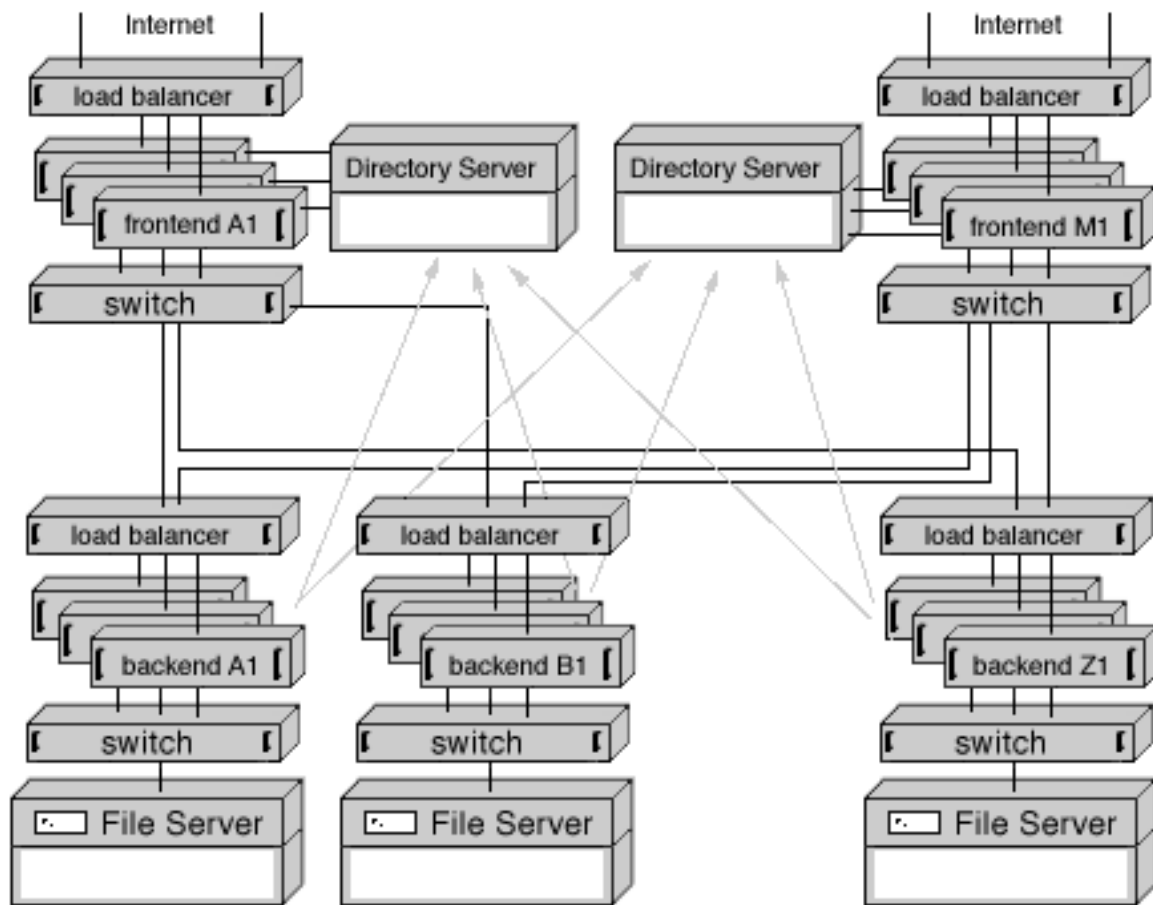
The FTP module running on cluster frontends behind a load balancer and/or a NAT has the same problems as any FTP server running in such a configuration. To support the active mode, make sure that frontend servers can open outgoing connections to client FTP ports (when running via a NAT, make sure that the "address in use" problems are addressed by the NAT software). To support the passive mode, make sure that your load balancer allows clients to connect directly to the frontend ports the FTP module opens for individual clients

Postmaster Account

Do not remove the "postmaster" Account from the Main Domains on your backend servers. This Account is opened "by name" (bypassing the Router) when any other Cluster member has to connect to that backend. You should also keep at least the "Can Modify All Domains and Accounts Settings" [Access Right](#) for the postmaster Account.

Cluster Of Clusters

For extremely large sites (more than 5,000,000 active accounts), you can deploy a Static Cluster of Dynamic Clusters. It is essentially the same as a regular Static Cluster with Frontend Servers, but instead of Backend Servers you install Dynamic Clusters. This solves the redundancy problem of Static Clusters, but does not require extremely large Shared Storage devices and excessive network traffic of extra-large Dynamic Clusters:



Frontend Servers in a "Cluster of Clusters" need access to the Directory in order to implement Static Clustering. The Frontend Servers only read the information from the Directory, while the Backend Servers modify the Directory when accounts are added, renamed, or removed. The `hostServer` attribute of Account directory records contains the name of the Backend Dynamic Cluster hosting the Account (the name of Backend Cluster Servers without the first domain name element).

Frontend Servers can be grouped into subsets for traffic segmentation. Each subset can have its own load balancer(s), and a switch that connects this Frontend Subset with every Backend Dynamic Cluster.

If you plan to deploy many (50 and more) Frontend Servers, the Directory Server itself can become the main site bottleneck. To remove this bottleneck and to provide redundancy on the Directory level, you can deploy several Directory Servers (each Server serving one or several Frontend subsets). Backend Dynamic Clusters can be configured to update only one "Master" Directory Server, and other Directory Servers can use replication mechanisms to synchronize with the Master Directory Server, or the Backend Clusters can be configured to modify all Directory Servers at once.



Web Application Module

The CommuniGate Pro Web Application module provides access to various CommuniGate Pro objects (accounts, messages, mailing lists, web files) via any Web (HTTP/HTML) browser.

The [HTTP module](#) receives HTTP client browser requests that come to the WebUser port(s), and passes those requests to the Web Application module. The Web Application module either retrieves the requested file, or it starts some internal *web application* code and converts the result into the HTML format. The result is returned to the HTTP module that delivers it back to the client browser.

Read this section if you want to customize your CommuniGate Pro Server Web User Interface.

Stateless and Session-based Processing

Regular HTTP servers are *stateless* processors: a user's browser may send several sequential requests, but the HTTP server does not keep any information about the browser or the client between the requests. Each request is processed individually.

The Web Application Server allows users to "log in", providing a name of some CommuniGate Pro Account and the account password. For each successful login, a *Session* is started. The session keeps the information about user actions and requests, so all HTTP requests sent to the same session can share and use the same set of session data. To maintain a session, all session requests have URLs in the following form:

`http://hostname:serverport/Session/sessionID/sessionRequest`

where the *sessionID* string identifies the session, and the *sessionRequest* is the name of the file to retrieve or the application component to run.

The Web Application Sessions have time-out counters. If no HTTP request has been sent to a session during the configurable time-out interval, the Session is closed. The session user can close the session by sending a request for the special *Bye* page.

The *hostname* string specified in a stateless request URL is used to find the CommuniGate Pro [Domain](#). Each Domain may have a *Basic Skin* - a set of files that an HTTP request can retrieve. Since different Domains have different Basic Skins, the same request sent to different Domains can retrieve different data.

If a Domain does not have a Basic Skin, the server-wide Basic Skin is used.

The CommuniGate Pro Server software comes with a Stalker-designed Skin, called the *stock* Skin. This Skin is stored in the *application directory*, it is a part of the software package, and it should not be modified by server administrators.

If a requested file is not found in the Basic Domain Skin, the Web Application module tries to retrieve a file with the specified name from the server-wide Basic Skin. If the file is not found in the server-wide Basic Skin, the module tries to find that file in the Stock Skin.

Initially, when no Domain has a Basic Skin, and the server-wide Basic Skin is empty, all Domains effectively use the Stock Skin. By uploading files into the server-wide Basic Skin, the server administrator can "shadow" the Stock Skin files, and can change the application look and feel for all Domains. By uploading files into the Domain Basic Skin, the Server or Domain Administrator can "shadow" the Stock Skin and server-wide Basic Skin files and change the look and feel for that particular Domain.

The session-based HTTP requests do not use the hostname string specified in the URL. Instead, when a user logs in, and a Web Application session is created for the specified CommuniGate Pro [Account](#), the Account WebUser preferences are retrieved. Those preferences contain the name of the Skin to use. If this name is empty, then the Basic Skin for the Account Domain is used. If that Domain does not have a Basic Skin, the server-wide Basic Skin is used.

If the Account WebUser preferences specify a non-empty Skin name, the Web Application module tries to open a *Named Skin*. Each Domain can have several Named Skins, and if the Domain has the Named Skin with the specified name, it is used. If the Domain does not have the specified Named Skin, the module tries to find the server-wide Named Skin with the specified name. If it cannot find it, the server-wide Basic Skin is used.

The Stateless requests usually use the Basic Skin for the addressed Domain. To retrieve a file from a Named Skin, the request should include the *Skin* parameter that specifies the Skin Name.

When the Web Application module needs to retrieve a file from a Named Skin, and the Domain Named Skin does not contain a file with the specified name, the module looks for a server-wide Named Skin with the same name, and tries to retrieve the file from that Skin. If the file is not found again, the module tries to retrieve the file from the server-wide Basic Skin, and then - from the Stock Skin.

This hierarchy of Skins allows Server and Domain Administrators to provide different, custom look and feel for different Domains and for different users, and simplifies Skin development by providing server-wide Skins used as the source of "default" files.

The [Dynamic Cluster](#) installations have two sets of the server-wide Skins - one set is used for local Server Domains, while the other, Cluster set is used for all Shared Domains in the Cluster. Modifications of these server-wide Skins, as well as modifications of the Shared Domain Skins are automatically distributed to all Cluster Members.

Skin Text Dataset

Each Skin can have a Text Dataset - the `strings.data` text file containing a [dictionary](#). The [WSSP Script](#) pages can use various commands to retrieve data from this Skin Text Dataset.

When the Text Dataset of the selected Basic Domain Skin does not contain the requested data, the Text Dataset of the server-wide Basic Skin is checked. If it does not contain the requested data, the Text Dataset of the Stock Skin is checked.

When the Text Dataset of the selected Named Domain Skin does not contain the requested data, the Text Dataset of the server-wide Named Skin is checked. If it does not contain the requested data, the Text Dataset of the server-wide Basic Skin is checked. If it does not contain the requested data, the Text Dataset of the Stock Skin is checked.

If the strings in the Skin Text Dataset should contain non-ASCII symbols, the UTF-8 character set should be used. Check the <http://www.unicode.org> site to learn more about Unicode and the UTF-8 charset.

Serving Regular Files

When a URL specifies a file with any file name extension other than `.wssp`, the Web Application module retrieves this file from the selected Skin, places it into the internal Skin Cache, and returns that file to the client browser via the HTTP module connection.

The specified file names are always converted into the lowercase letters.

When the Web Application module receives a request for the same file, it is retrieved from the Skin

Cache.

If the file has been requested using a Session-based URL, the session time-out counter is reset. This can be used to create a frame in the client browser window, and make that frame periodically retrieve some file using the session URL. As a result, this session inactivity timer can be reset to keep the session alive as long as this frame is displayed in the user's browser.

System and Domain administrators can upload custom files into server-wide and Domain Skins to modify the Web Application look and feel.

For example, the Stock Skin uses the `Logo.gif` file for most of its pages. By uploading a custom `Logo.gif` file to a server-wide Basic Skin you can change the look of the Web Application pages even without creating and uploading custom page (WSSP) files.

To retrieve a file from the Basic Domain Skin, use the

`HREF="/files/filename.extension"`

URL. The additional `/files/` realm is used to avoid conflicts between file names and names of special objects that are served in the "topmost" realm.

Sessions can use Named Skins, and the session-based pages usually need to refer to regular files in the same Skin. References in the "session realm" (`HREF="filename.extension"` or `HREF="/Session/sessionID/filename.extension"`) work, but they do not allow client browsers to cache these files between sessions, since each session has its own `sessionID`, and file URLs are different for each session. To allow client browsers cache regular files, use the following regular file URLs:

`HREF="/SkinFiles/domanName/skinName/filename.extension"`

To compose these URLs, use the `SESSION(filesRef)` [function](#) in your `.wssp` files.

Serving Web Application (WSSP) Files

When a URL specifies a resource with the `.wssp` file name extension, the Web Application module retrieves the specified WSSP file from the Skin, and Compiles it into some internal code. The module then runs the Web Application code associated with the file name. This code produces a dataset with various string, array, and dictionary data. Then the module runs the WSSP internal (compiled) code to produce an HTML page using this dataset, and returns the resulting HTML page to the browser using the HTTP module connection.

The specified resource names are always converted into the lowercase letters.

The [WSSP Scripting](#) section explains the WSSP file format. System and Domain administrators can create custom WSSP files and upload them to the server-wide and Domain Skins to modify the Web Application look and feel.

The [Code Components](#) section lists the available Web Application code components, defining the set of WSSP pages that this version of CommuniGate Pro server can generate. It specifies how each component processes the form parameters sent to it, and what data is included into the dataset it generates.

Creating and Managing Skins

The same type of WebAdmin Editor is used to manage server-wide and Domain Skins.

To manage the server-side Skins, open the Domains realm of the WebAdmin Interface, and click the Skins link.

To manage the Domain Skins, open that Domain page in the Domains realm of the WebAdmin Interface, and click the Skins link. The Domain Administrator should have the CanModifySkins Access Right to be able to create and modify the Domain Skins.

The Skins Editor page appears. If the Domain Skins Editor is opened, and there is no Basic Skin for that Domain, the page contains the Create Basic Skin button. Click this button to create the Basic Skin for this Domain.

The Skin Editor page contains the list of all files "visible" in this Skin: it lists files directly uploaded to this particular Skin, as well as all files uploaded to the Skins that work as the "default files" source for this Skin:

Marker	File Name	Size	Modified
	Help.gif	155	25-Sep-01
default	addressbook.wssi	1143	23-Sep-01
default	alerts.wssp	1727	26-Sep-01
default	answeredletter.gif	890	27-Feb-99
default	attachedFile.gif	1147	27-Feb-99
	...		
default	mailbox.wssp	5806	28-Sep-01
	mailboxes.wssp	3452	02-Oct-01
	...		
default	website.wssp	592	28-Sep-01

default	websitebody.wssi	2648	28-Sep-01

Files directly uploaded to the Skin have a checkbox in the Marker column. Files from the other skins "visible" in this Skin have the word `default` in that column.

You can download any of the Skin files by clicking the file name.

You can upload a file to the Skin by clicking the Browse button and selecting the file on your workstation, then clicking the Upload button.

You can delete any of the files uploaded to the Skin by selecting the checkboxes and clicking the Delete Marked button.

If you are uploading a `.wssp` or a `.wssi` file, the Editor tries to compile that file first. If the compiler parser detects an error, the file is not uploaded, the source of the file is displayed on the Editor page, with the red `<--ERROR-->` marker indicating the location of the error.

When you upload a file to any Skin, that Skin cache is automatically cleared. If you upload a file to a Shared Domain Skin or to a Cluster-wide Skin, the update automatically propagates to all Cluster Members.

The Editor page for a Basic Skin contains the list of all Named Skins:

Named Skins	
Marker	Skin Name
	German
	IceColdMail
:	

To create a Named Skin, enter its name, and click the Create Skin button.

To remove Named Skins, use the checkboxes to select the Skins, and then click Remove Marked button. Only empty Skins (Skins with any files) can be removed.

To remove the Basic Skin, remove all its files and all Named Skins, and then click Remove Basic Skin button.

To open a Skin, click its name. The Editor will display the Skin Name, and it will provide the UP link to the Basic Skin page.

The Named Skin Editor allows you to rename the Skin by entering a new Skin Name and clicking the Rename Skin button.



WSSP Scripting

The CommuniGate Pro Web Application module processes a request for a WSSP file by calling a [code component](#) that produces a *dataset* - a [dictionary](#) containing text string keys and values, associated with those keys. Values can be text strings, arrays of values, or dictionaries.

For example, when a Domain default page is requested, the code component is called. The component processes request (HTML FORM) parameters and produces a dataset - a dictionary containing keyed values. For example, a dataset produced with the component processing the Login requests may contain `canAutoSignup`, `hasMailLists`, and `hasCertificate` keys.

The Web Application module then uses the script code from a WSSP file to convert this dataset into a markup language (HTML) page.

Scripting Elements

The WSSP file is a markup language (usually - HTML) file with two additional types of elements:

- text elements, started and ended with double percent signs (%%)
- structural elements, started with the `<!--%%` marker and ended with the `-->` marker.

The following is a sample of an WSSP document:

```
<HTML>
<BODY>

<H1>Welcome to %%server%%. Your ID is %%ID%%.</H1>

<!--%%IF EXISTS(lastLogin)-->
Last time you visited us on %%lastLogin%%
<!--%%ENDIF-->

</BODY>
</HTML>
```

This WSSP document contains the `%%server%%`, `%%ID%%`, and `%%lastLogin%%` text elements, and the `<!--%%IF EXISTS(lastLogin)-->` and `<!--%%ENDIF-->` structural elements.

If the WSSP document should contain non-ASCII symbols, the UTF-8 character set should be used. When the WSSP document is being processed, the Web Application module retrieves the `charset` string value from the produced data dictionary. If this value is not UTF-8, then the WSSP text is converted into this *page charset*.

Expressions

The text and structural WSSP elements use expressions - combinations of names and symbols that specify the data to be retrieved from the data dictionary or from other available sources.

The WSSP scripting uses several types of expressions:

- data element
- array scanner
- keyed element
- indexed element
- functions

An alphanumeric string (such as `system` or `id`) is a data element name. The value of such an expression is the dataset value associated with this name. If the dataset does not have a specified key, the expression value is NULL.

Example: the dataset contains the key `system` and its associated value is the "Sun Solaris" string, the value of the expression `system` is the string `Sun Solaris`.

The dataset dictionary is case-insensitive, so the data element names are case-insensitive, too.

An alphanumeric string followed by the `[]` symbols is interpreted as an array scanner name. It can be used only inside the `<!--%%FORALL name...--> ...<!--%%ENDFOR name-->` structure where this array element is defined (see below). The array scanner names are case-insensitive.

An expression followed by the dot symbol `(.)` and an alphanumeric string is a keyed element. The expression before the dot symbol is calculated, and its value should be a dictionary. The alphanumeric string after the dot sign specifies the key to be used to extract the value from that dictionary. If the value of the expression before the dot sign is not a dictionary, or if it does not contain the specified key, the keyed element value is NULL.

Keys can be specified as quoted strings, in this case they can specify non-alphanumeric symbols.

Example: the dataset contains the key `settings` and its associated value is the 2-element dictionary: `{ OS = "Sun Solaris"; CPU="sparc"; }`. The value of the `settings.OS` expression is the `Sun Solaris` string, the value of the `settings."OS1"` expression is NULL.

An expression followed by an *index expression* in square bracket symbols (*[index]*) is an indexed element. The expression before the square bracket is calculated, and its value should be an array. The *index expression* is calculated, and its value should be a string representing a number. This number specifies which array element becomes the value of this indexed expression. The first array element is retrieved if the value of the *index expression* is 0.

If the value of the expression before the bracket symbol is not an array, or if the value of the *index expression* is not a string, or if the value of the *index expression* represents a number that is negative or is equal or greater than the number of array elements, the value of the indexed element expression is NULL.

An alphanumeric string followed by the (symbol is a function call. Elements after the (symbol specify the function parameters, and they are followed by the) symbol.

Function names are case-insensitive.

The following list specifies the available functions and their parameters.

SESSION(*key*)

This function can be used only in Session-based requests. The function value is the session dataset value associated with the string *key*. The *key* parameter can be specified as an alphanumeric string, or as a quoted string.

Example: the SESSION(*accountName*) expression value is the name of the CommuniGate Pro Account this session is opened for.

The session dataset is case-insensitive. It contains the following keys and values:

Key	Value
ID	a string with the unique identifier of this session
accountName	a string with the session Account name
domainName	a string with the name of the Domain the session Account belongs to
filesRef	a string with the URL prefix needed to retrieve files from the session Skin
fullAccountName	a string with the session Account full name: <i>accountName@domainName</i>
loginAddress	a string specifying the network (IP) address the user was using when initiating this session
loginTime	a string with the session start time in the ACAP format
selectedMailbox	a string with the name of the target mailbox for the last Copy/Move operation

EXISTS(*expression*)

The parameter is an expression. Its value is calculated, and the function returns the string YES if the calculated value is not NULL, or the string NO if the returned value is NULL.

DOESNOTEXIST(*expression*)

The parameter is an expression. Its value is calculated, and the function returns the string NO if the calculated value is not NULL, or the string YES if the returned value is NULL.

NOT(*expression*)

The parameter is an expression. Its value is calculated, and the function returns NULL if the calculated value is a string that does not start with one of the symbols N, n, -, or 0, otherwise the function returns the string YES.

`EQUALS(expression1 AND expression2)`

Both expressions are calculated, and if the calculated values match (including the case when both expressions return NULL), the function returns the string YES. Otherwise the function returns NULL.

`EQUALS(expression AND string)`

The value of the expression is calculated and compared with the string, specified as a quoted string. If the value matches the string, the function returns the string YES. Otherwise the function returns NULL.

`ISINDEX(expression IN scanner)`

The *scanner* should be the name of the `<!--%%FORALL . .-->` construct surrounding the current portion of the script code. The expression value is calculated, and if it is a string and its numeric value matches the current index in the array this *scanner* is used for, the function returns the string YES. Otherwise the function returns NULL.

`CHECKED(expression)`

The parameter is an expression. Its value is calculated, and the function returns the string CHECKED if the calculated value is a string that does not start with one of the symbols N, n, -, or 0, otherwise the function returns NULL.

`NULL ()`

The value of this function is NULL.

`EMPTYSTRING ()`

The value of this function is an empty string.

`EMPTYARRAY ()`

The value of this function is an array with zero elements.

`EMPTYDICTIONARY ()`

The value of this function is an empty dictionary.

`STRING(key)`

The value of this function is the object associated with the *key* in the Skin Text Dataset. This object should be a string, otherwise the function returns NULL. The key can be specified either as a quoted string literal, or as an expression - the expression value is calculated and used as the key.

`DICTIONARY(key)`

The value of this function is the object associated with the *key* in the Skin Text Dataset. This object should be a dictionary, otherwise the function returns NULL. The key can be specified either as a quoted string literal, or as an expression - the expression value is calculated and used as the key.

`ARRAY(key)`

The value of this function is the object associated with the *key* in the Skin Text Dataset. This object should be an array, otherwise the function returns NULL. The key can be specified either as a quoted string literal, or as an expression - the expression value is calculated and used as the key.

`TRANSLATE(string USING dictionary)`

The *string* parameter is an expression that should return a string value; the *dictionary* parameter is an expression that should return a dictionary value. If the dictionary contains a string value for the key specified with the first parameter value, the function returns this string. Otherwise the value of the *string* parameter is returned;

Example: the dataset contains the element `boxName` with the string value `INBOX`, and the element `boxNames` with the dictionary value `{INBOX = Incoming; Trash = "Trash Can";}`. The value of the `TRANSLATE(boxName USING boxNames)` expression is the `Incoming` string.

`RANDOMELEMENT(array)`

The *array* parameter is an expression that should return an array value; The value of this function is a randomly-selected element from that array.

Text Elements

Text elements are specified using double percent markers. The body of a text element is an expression with an optional prefix.

`%%expression%%`

The expression is calculated. If the value is not a string, then the entire text element is removed from the resulting markup (HTML) code. If the result is a string, it substitutes the text element in the resulting markup code.

`%%HTML:expression%%`

The expression is calculated. If the value is not a string, then the entire text element is removed from the resulting markup (HTML) code. If the result is a string, the string is converted from the UTF-8 charset into the required charset, and the converted string substitutes the text element using HTML escape symbols.

Example: if the expression result is the `>=GO=>` string, the text element is substituted with the `>=GO=>` string.

`%%URL:expression%%`

The expression is calculated. If the value is not a string, then the entire text element is removed from the resulting markup (HTML) code. If the result is a string, it substitutes the text element using URL escape symbols.

Example: if the expression result is the `Stop It?` string, the text element is substituted with the `Stop+It%3F` string.

`%%MAILBOXRAWNAME:expression%%`

The expression is calculated. If the value is not a string, then the entire text element is removed from the

resulting markup (HTML) code. If the result is a string, it is converted from the IMAP-specified mailbox name encoding into the UTF-8 charset, then it is converted into the required charset, and the converted string substitutes the text element using HTML escape symbols.

`%%MAILBOXNAME:expression%%`

The expression is calculated. If the value is not a string, then the entire text element is removed from the resulting markup (HTML) code. If the result is a string *X*, the `TRANSLATE(X USING DICTIONARY("MailboxNames"))` expression is calculated. Then the prefix works in the same way as the `MAILBOXRAWNAME` prefix.

`%%SIZE:expression%%`

The expression is calculated. If the value is not a string, then the entire text element is removed from the resulting markup (HTML) code. If the result is a string, it is converted into a numeric value (the number of bytes).

The string should contain some number and, optionally, the *k* or *K* suffix (in this case the number is multiplied by 1024), or the *m* or *M* suffix (in this case the number is multiplied by 1048576).

Alternatively, a string can start with a letter *u* or *U*, in this case the converted number of bytes is -1.

The resulting number is converted into a "size string", using the dictionary retrieved with the `DICTIONARY("SizePictures")` expression. If the number is negative, the dictionary is used to translate the string `unlimited`, and the result is used to replace this text element using the same conversions as used for the a text element with the `HTML:` prefix.

The calculated number of bytes is checked to see if it represents an even number of Megabyte or Kilobytes, and that number is greater than one. Then a string value associated with the keys `"M"`, `"K"`, or `" "` (empty string) is retrieved from the dictionary. The string is expected to contain the `^0` symbol combination which is replaced with the number of megabytes, Kilobytes, or bytes specified with the *expression* value. The resulting string is processed with the method used for the `HTML:` text element prefix.

If the `DICTIONARY("SizePictures")` expression result is `NULL`, or this dictionary does not contain a string value for the required key, the resulting string is built using the number and the key name (20M, 1345K, 182345777).

`%%ROUNDSIZE:expression%%`

This prefix works in the same way as the `SIZE:` prefix, but the numeric *expression* value can be modified: if the value is equal or larger than 10000, then it is converted into "Kilo" ($\text{value} = \text{value} / 1024 * 1024$), and if the value is equal or larger than 10240000, it is converted to "Mega" ($\text{value} = \text{value} / 1048576 * 1048576$).

`%%TIME:expression%%`

The expression is calculated. If the value is not a string, then the entire text element is removed from the resulting markup (HTML) code. If the result is a string, it is converted into a numeric value (the number of seconds).

The string should contain some number and, optionally, the s or S suffix, or the m or M suffix (in this case the number is multiplied by 60), or the h or H suffix (in this case the number is multiplied by 3600), or the d or D suffix (in this case the number is multiplied by 86400).

The resulting number is converted into a "time string", using the dictionary retrieved with the `DICTIONARY("TimePictures")` expression.

The calculated number of seconds is checked to see if it represents an even number of weeks, days, hours, or minutes, and if that number is greater than 1. Then a string value associated with the keys `weeks`, `days`, `hours`, `minutes`, or `seconds` is retrieved from the dictionary. The string is expected to contain the `^0` symbol combination which is replaced with the number of weeks, days, hours, minutes, or seconds specified with the *expression* value. The resulting string is converted from the UTF-8 into the required charset and the converted string substitutes the text element using HTML escape symbols.

If the `DICTIONARY("TimePictures")` expression result is `NULL`, or this dictionary does not contain a string value for the required key, the resulting string is built using the number, a space, and the key name (3 weeks, 11 hours, 5 seconds).

Example:

If the data element `elapsedTime` is the 2400 string, then the text element `%%TIME:elapsedTime%%` will be substituted with the 40 minutes string.

If the `DICTIONARY("TimePictures")` exists and contains the string `^0mins` as the `minutes` value, then the text element `%%TIME:elapsedTime%%` will be substituted with the 40mins string.

`%%DATE:expression%%`

The expression is calculated. If the value is not of the "date" type, then the entire text element is removed from the resulting markup code. If the result is a "date" element it is converted into the date string:
DD-MMM-YYYY

where DD is the day of month, MMM is the month name, and YYYY is the year.

The month name is Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

If the `DICTIONARY("DatePictures")` expression result is a dictionary, that dictionary is used to convert ("translate") the month names.

`%%DATETIME:expression%%`

The expression is calculated. If the value is not of the "date" type, then the entire text element is removed from the resulting markup code. If the result is a "date" element it is converted into the time and date string:
HH:MM:SS DD-MMM-YYYY

where HH is 2-digit hour value, MM - minutes, SS - seconds, and the DD-MMM-YYYY is processed in the same way as it is processed for the `DATE:` prefix.

`%%DATETIMESHORT:expression%%`

The expression is calculated. If the value is not of the "date" type, then the entire text element is removed

from the resulting markup code. If the result is a "date" element is compared with the current Universal (GMT) time.

If the time difference is less than 24 hours, the value is shown as:

HH:MM:SS

where HH is 2-digit hour value, MM - minutes, SS - seconds.

If the time difference is larger or equal to 24 hours, the value is shown as:

DD-MMM-YYYY

i.e. it is shown in the same way as with the DATE: prefix.

%%LOCALDATETIME:*expression*%%

Processing is the same as for the DATETIME prefix, but the date value is converted into the local time first.

%%LOCALDATETIMESHORT:*expression*%%

Processing is the same as for the DATETIMESHORT prefix, but the date value is converted into the local time first, and the difference with the current local time defines if the time or date value is to be displayed.

Structural Elements

The structural elements start with the <!--%% symbols and end with the --> symbols. The structural elements themselves are always removed from the resulting markup (HTML) code.

<!--%%IF *expression*-->

This structural element can be followed with the

<!--%%ELSE-->

element, and must be followed with the

<!--%%ENDIF-->

element.

The value of the expression is calculated, and if it is a string and that string does not start with one of the symbols N, n, -, or 0, the portion of the script between the <!--%%IF *expression*--> element and the <!--%%ELSE--> element is processed. If the <!--%%ELSE--> element does not exist, the portion of the script between the <!--%%IF *expression*--> element and the <!--%%ENDIF--> element is processed.

If the *expression* result is not a string, or it is a string that starts with one of the listed symbols, the portion of the script between the <!--%%ELSE--> element and the <!--%%ENDIF--> element is processed. If the <!--%%ELSE--> element does not exist, the portion of the script between the <!--%%IF *expression*--> and <!--%%ENDIF--> elements is removed completely.

Example:

```
<!--%%IF EXISTS(lastLogin)-->We have not seen you since  
<I>%%HTML:lastLogin%%</I>  
<!--%%ELSE-->Welcome, new user!  
<!--%%ENDIF-->
```

In this example, if the dataset contains the `lastLogin` element with the `20-Apr-2001` string value, this script portion will produce

```
We have not seen you since <I>20-Apr-2001</I>
```

If the dataset does not contain the `lastLogin` element, this script portion will produce

```
Welcome, new user!
```

```
<!--%%FORALL scanner in expression-->
```

This structural element can be followed with the

```
<!--%%EMPTYFOR scanner-->
```

element, and must be followed with the

```
<!--%%ENDFOR scanner-->
```

element. All elements should have the same *scanner* alphanumeric string.

The value of the *expression* is calculated. The resulting value should be an array.

The portion of the script between the `<!--%%FORALL scanner ...-->` and `<!--%%EMPTYFOR scanner-->` elements or (if the `<!--%%EMPTYFOR scanner-->` element does not exist) the portion of the script between the `<!--%%FORALL scanner ...-->` and `<!--%%ENDFOR scanner-->` elements is processed repeatedly, for each array element.

Expressions specified in that portion of the script can use the *scanner*[] *array scanner* reference to access the current element of the *expression* array value.

If the *expression* value is not an array or if it is an empty array, and the `<!--%%EMPTYFOR scanner-->` element is specified, the portion of the script between the `<!--%%EMPTYFOR scanner-->` and `<!--%%ENDFOR scanner-->` elements is processed. If the resulting array has at least one element, this portion is not processed.

Example:

```
<TABLE BORDER=1>  
<TR><TD>File Name</TD><TD>File Size</TD></TR>  
<!--%%FORALL elem in fileList-->  
<TR><TD>%%HTML:elem[ ].name%%</TD><TD>%%elem[ ].size%%</TD></TR>  
<!--%%EMPTYFOR elem-->  
<TR><TD colspan=0>&nbsp;</TD></TR>  
<!--%%ENDFOR elem-->  
</TABLE>
```

In this example, the data element `fileList` is expected to be an array of dictionaries.

Each dictionary is expected to contain string values for keys `name` and `size`.

If the `fileList` value is

```
( {name=MyReport; size=2300; } , {name="My Old Report";  
size=4000; } )
```

then this portion of the script will produce the following HTML code:

```
<TABLE BORDER=1>  
<TR><TD>File Name</TD><TD>File Size</TD></TR>  
<TR><TD>MyReport</TD><TD>2300</TD></TR>  
<TR><TD>My Old Report</TD><TD>4000</TD></TR>  
</TABLE>
```

The `<!--%%IF ...--> ...<!--%%ELSE--> ...<!--%%ENDIF-->` constructs and the `<!--%%FORALL ...--> ... <!--%%ENDFOR ...-->` constructs can be nested.

```
<!--%%INCLUDE filename-->
```

The file with the specified *filename* name is retrieved from the same Skin this script is retrieved from. The file should contain some WSSP code. This code is executed within the current context (using the same dataset). The resulting markup (HTML) code is used to replace this structural element.

It is recommended to use the `.wssi` file name extension for files designed to be used with the `INCLUDE` element.

The `INCLUDE` elements can be nested, this means that `.wssi` files can include other `.wssi` files.

Unlike the `#include` operators in the C and C++ languages, this operator is a real, not a pre-processor operator. As a result, if an `<!--%%INCLUDE filename-->` element is used within a `<!--%%FORALL ...--> ... <!--%%ENDFOR ...-->` construct, the *filename* code can be executed several times, once for each element of the array used in the `FORALL` construct.

```
<!--%%NUMERICMENU selected [DEFAULT selectedDefault ] IN  
(number1,number2,...,numberN) [ DISPLAY dictionary]-->
```

This element is substituted with a sequence of the `<OPTION VALUE="value">presentation` string elements.

The number of elements and the *value* used for each element are defined by the list of numeric numbers - *number1,number2,...,numberN*. These numbers should be specified in the ascending order, and these numbers should not be less than -1.

The *selected* expression is calculated, and its value should be a string. The string numeric value is used to add the keyword `SELECTED` to the `<OPTION VALUE="value">` element that has the same *value*.

The `DISPLAY` keyword and the *dictionary* expression can be omitted. In this case, the *presentation* strings are the same as the *value* strings, this means that these strings are numbers.

Example:

The dataset element `sizeLimit` is the 200 string.

The element:

```
<!--%%NUMERICMENU sizeLimit IN (-1,0,100,200,300)-->
```

will be substituted with the following markup (HTML) text:

```
<OPTION VALUE="-1">-1<OPTION VALUE="0">0
<OPTION VALUE="100">100<OPTION VALUE="200" SELECTED>200<OPTION
VALUE="300">300
```

If the `DISPLAY` keyword and the *dictionary* expression are specified, the expression is calculated. If the expression value is a dictionary, then the *presentation* strings are the numeric values "translated" using this dictionary, the results are converted into the required charset and placed using the HTML escape symbols.

Example:

The dataset element `sizeLimit` is the 200 string.

The Skin Text Dataset contains the `Limits` dictionary: `{"-1" = Unlimited; 0 = "Off & Shut";}`.

The element:

```
<!--%%NUMERICMENU sizeLimit IN (-1,0,100,200,300) DISPLAY
DICTIONARY("Limits")-->
```

will be substituted with the following markup (HTML) text:

```
<OPTION VALUE="-1">Unlimited<OPTION VALUE="0">Off & Shut
<OPTION VALUE="100">100<OPTION VALUE="200" SELECTED>200<OPTION
VALUE="300">300
```

If the keyword `DEFAULT` with the *selectedDefault* expression are specified, an additional `<OPTION VALUE="-2">defaultPresentation` string is added before the sequence. If the *selected* expression value is `NULL`, this string element will have the keyword `SELECTED` added.

The *defaultPresentation* string is the `DefaultValuePicture` string retrieved from the Skin Text Dataset. This string should contain the `^0` symbol combination. This symbol combination is substituted with the *selectedDefault* expression value. If the `DISPLAY` dictionary is specified, the *selectedDefault* expression value is translated first.

Example:

The dataset element `sizeLimit` is the 200 string.

The dataset element `defLimit` is the -1 string.

The Skin Text Dataset contains the `DefaultValuePicture` string: `default(^0)`.

The Skin Text Dataset contains the `Limits` dictionary: `{"-1" = Unlimited; 0 = "Off & Shut";}`.

The element:

```
<!--%%NUMERICMENU sizeLimit DEFAULT defLimit IN (-
1,0,100,200,300) DISPLAY DICTIONARY("Limits")-->
```

will be substituted with the following markup (HTML) text:

```

<OPTION VALUE="-2">default(Unlimited)
<OPTION VALUE="-1">Unlimited<OPTION VALUE="0">Off & Shut
<OPTION VALUE="100">100<OPTION VALUE="200" SELECTED>200<OPTION
VALUE="300">300

```

```

<!--%%TIMEMENU selected [DEFAULT selectedDefault ] IN
(number1,number2,...,numberN) [ DISPLAY dictionary ]-->

```

This element is processed in the same way as the NUMERICMENU element. The *numberX* numbers specify time intervals in seconds.

The *selected* and *selectedDefault* expressions should return strings. Each string is converted into a numeric value (the number of seconds) using the algorithm used in the text elements with the TIME: prefix.

Values are translated using the specified DISPLAY dictionary. If the DISPLAY dictionary is not specified or it does not contain a string for the given value, the *presentation* time strings are composed using the same method as the method used for the TIME: text elements.

```

<!--%%SIZEMENU selected [DEFAULT selectedDefault ] IN
(number1,number2,...,numberN) [ DISPLAY dictionary ]-->

```

This element is processed in the same way as the NUMERICMENU element. The *numberX* numbers specify a data size in bytes.

The *selected* and *selectedDefault* expressions should return strings. Each string is converted into a numeric value (the number of bytes) using the algorithm used in the text elements with the SIZE: prefix.

Values are translated using the specified DISPLAY dictionary. If the DISPLAY dictionary is not specified or it does not contain a string for the given value, the *presentation* size strings are composed using the same method as the method used for the SIZE: text elements.

```

<!--%%ENUMMENU selected [DEFAULT selectedDefault ] IN valueSet [ DISPLAY
dictionary ]-->

```

This element is substituted with a sequence of the <OPTION VALUE="*value*">*presentation* string elements.

The number of elements and the *value* used for each element are defined by the value of the *valueSet* expression. This value should be an array of strings. The *value* in each element is the string index in the *valueSet* array result.

The *selected* expression is calculated, and its value should be a string. The keyword SELECTED is added to the <OPTION VALUE="*value*"> element created for the the *valueSet* array element that matches the *selected* expression result.

The DISPLAY keyword and the *dictionary* expression can be omitted. In this case, the *presentation* strings are the same as the *valueSet* result elements.

Example:

The dataset element color is the Green string.

The dataset element colors is the (Blue, Green, Red) array.

The element:

```
<!--%%ENUMMENU color IN colors-->
```

will be substituted with the following markup (HTML) text:

```
<OPTION VALUE="0">Blue<OPTION VALUE="1" SELECTED>Green<OPTION  
VALUE="2">Red
```

If the DISPLAY keyword and the *dictionary* expression are specified, the expression is calculated. If the expression value is a dictionary, it is used to "translate" the *valueSet* array strings before converting them into the required charset and placing into the resulting markup text using the HTML escape symbols.

Example:

The dataset element color is the Green string.

The dataset element colors is the (Blue, Green, Red) array.

The Skin Text Dataset contains the Colors dictionary: {Blue = "Night Blue";
Green = "Grass Green";}.

The element:

```
<!--%%ENUMMENU color IN colors DISPLAY DICTIONARY("Colors")-->
```

will be substituted with the following markup (HTML) text:

```
<OPTION VALUE="0">Night Blue<OPTION VALUE="1" SELECTED>Grass  
Green<OPTION VALUE="2">Red
```

If the keyword DEFAULT with the *selectedDefault* expression are specified, an additional <OPTION VALUE="-1">*defaultPresentation* string is added before the sequence. If the *selected* expression value is NULL, this string element will have the keyword SELECTED added.

The *defaultPresentation* string is the DefaultValuePicture string retrieved from the Skin Text Dataset. This string should contain the ^0 symbol combination. This symbol combination is substituted with the *selectedDefault* expression value. If the DISPLAY dictionary is specified, the *selectedDefault* expression value is translated first.

Example:

The dataset element color is the Green string.

The dataset element defColor is the Blue string.

The dataset element colors is the (Blue, Green, Red) array.

The Skin Text Dataset contains the DefaultValuePicture string:
default(^0).

The Skin Text Dataset contains the Colors dictionary: {Blue =
"Night Blue"; Green = "Grass Green";}.

The element:

```
<!--%%ENUMMENU color DEFAULT defColor IN colors DISPLAY  
DICTIONARY("Colors")-->
```

will be substituted with the following HTML text:

```
<OPTION VALUE="-1">default(Night Blue)<OPTION VALUE="0">Night  
Blue  
<OPTION VALUE="1" SELECTED>Grass Green<OPTION VALUE="2">Red
```

```
<!--%%BOOLMENU selected [DEFAULT selectedDefault ] [ DISPLAY dictionary]-  
->
```

This element is substituted with a sequence of the `<OPTION VALUE="value">presentation` string elements in the same way the ENUMMENU element is processed.

Unlike the ENUMMENU element, this element does not contain the `IN valueSet` part: the built-in array (No , Yes) array is used instead.

```
<!--%%MAILBOXMENU selected [DEFAULT selectedDefault ] IN mailboxList-->
```

This element is substituted with a sequence of the `<OPTION VALUE="value">presentation` string elements in the same way the ENUMMENU element is processed.

The values of the *selected* and *selectedDefault* expressions are converted in the same way as they are converted for a text element with the MAILBOXNAME : prefix, using the MailboxNames dictionary from the Skin Text Dataset, this is why MAILBOXMENU elements do not have the DISPLAY part.



Web Application Code Components

The CommuniGate Pro Web Application module processes a request for a WSSP file by calling a code component that produces a *dataset* - a [dictionary](#) containing text string keys and values, associated with those keys. Values can be text strings, arrays of values, or dictionaries.

For example, when a Domain default page is requested, the code component called and it produces a dictionary that contains keys such as `canAutoSignup`, `hasMailLists`, `hasCertificate`.

The Web Application module then uses the script code from a WSSP file to convert this data into an HTML or other markup language page.

This section lists the available CommuniGate Pro code components, specifies when those components are called, explains how the code components process the `<FORM>` parameters, and specifies the content of the dataset produced by each code component.

Code Components for Stateless Requests

The Web Application module places certain data into datasets produced with all Stateless Requests code components. The following list specifies these "generic" dataset elements that can be used with all Stateless WSSP pages:

`serverName`

This element value is a string with this CommuniGate Pro server name (its Main Domain Name).

`domainName` `type:string`

This element value is a string with the name of the CommuniGate Pro Domain. This element exists only if the Server has succeeded to direct the Stateless Request to one of the server Domains.

`charset`

This element value is the value of the `charset` element from the Domain Skin Text Dataset.

Individual code components may specify other values for the `charset` element (see below).

`secureChannel`

This element exists and has the YES string value if the request has been received via a secure (HTTPS) connection.

The following sections specify the Stateless URLs, the name of the code component called, the actions taken by the component, the dataset produced with that component, and the name of the WSSP file used to produce the HTTP response.

URLs: `/`, `/default.html`

these URLs are used to process Login operations.

Actions

If the HTTP request has the `username` and `password` parameters, the code tries to authenticate the client using these parameter values. If the supplied credentials are correct, a new WebUser Session is created, and a request for the "entry page" is sent to the Session. This request usually returns an HTML "jump page" that is sent back to the user browser. It forces the browser to enter the "Session realm".

Result Dataset

If `username` or `password` parameter has not been specified, or a WebUser Session could not be created, the component generates the following *dataset*:

`autoSignup`

this element (with the string YES as the value) is added to the dataset if the addressed Domain provides the Auto-Signup operation.

`clientAddress`

this string element contains the IP address of the user browser.

`errorCode`

this element is added to the dataset if the Login operation has failed. The value is a string with the error code.

`forgotPassword`

this element (with the string YES as the value) is added to the dataset if the `errorCode` value is "Incorrect Password" or "Unknown Account".

hasCertificate

this element (with the string YES as the value) is added to the dataset if the addressed Domain has a custom [Certificate](#).

hasDirectory

this element (with the string YES as the value) is added to the dataset if the default Skin for the addressed Domain has an array element GuestDirectoryFields in its Text Dataset, and that array is not empty.

hasLists

this element (with the string YES as the value) is added to the dataset if the addressed Domain has at least one Mailing List. This element is always added if the addressed Domain is a Shared Domain.

loginName

this string element is added to the dataset if the user has tried to log in, but failed. The value specified in the username parameter becomes the loginName element value.

WSSP page:

the login.wssp page is used to generate the HTTP response.

URL: /RecoveryPassword.wssp

this URL is used to process Password Recovery operations.

Action

If the HTTP request has the username and Send parameters, the component tries to find the specified Account, retrieves the Account password, and the RecoverPassword Account Setting. If the password can be decrypted and the RecoverPassword is specified, an E-mail message containing the password is composed and sent to the RecoverPassword E-mail address.

Result Dataset

errorCode

this element is added to the dataset if the Password Recovery operation has failed. The value is a string with the error code.

messageCode

this element is added to the dataset if the Password Recovery operation has succeeded. The value is the PasswordSent string.

WSSP page:

the `recoverypassword.wssp` page is used to generate the HTTP response.

URL: /signup.wssp

this URL is used to process Password Recovery operations.

Action

If the HTTP request has the `username`, `password1`, `password2`, and the `realName` parameters, the component tries to create the specified Account. Before it tries to create an Account, it checks if the `password1` and `password2` strings are the same. If a new Account is created, its `UseAppPassword` setting is set to YES, the `Password` and `RealName` settings are set to the specified values. If the HTTP request contains a non-empty `ForgotPassword` parameter, it is used as the `RecoverPassword` Account Setting.

The component checks if the request contains one or more `PublicInfo` string parameters. The value of the parameter must be one of the Public Info Attributes specified in the [Directory Integration](#) settings. The component then checks if there is a non-empty request parameter with this name, and adds the parameter value to the initial Account settings.

Example: to provide a `City` field on the Auto-Signup page, include the `<INPUT type="hidden" name="PublicInfo" value="City">` control and the `<INPUT type="text" name="City" value="" size=30 maxLength=255>` control into the `Signup.wssp` HTML code.

If an Account has been created, a new WebUser Session is created, and a request for the "entry page" is sent to the Session (see above).

Result Dataset

If `username`, `password1`, `password2`, or the `realName` parameter has not been specified in the HTTP request, or a new Account could not be created, the component generates the following *dataset*:

`errorCode`

this element is added to the dataset if the Auto-Signup operation has failed. The value is a string with the error code.

`userName`

this element is added to the dataset if HTTP request contained a non-empty `userName` parameter. The element value is the value of the parameter.

`realName`

this element is added to the dataset if HTTP request contained a non-empty `realName` parameter. The element value is the value of the parameter.

recoverPassword

this element is added to the dataset if HTTP request contained a non-empty recoverPassword parameter. The element value is the value of the parameter.

WSSP page:

the `signup.wssp` page is used to generate the HTTP response.

URL: /List/, /List/default.html

this URL is used to retrieve the list of the browsable Domain [Mailing Lists](#).

Action

The HTTP request parameters are not processed.

Result Dataset

The component generates the following *dataset*:

errorCode

this element is added to the dataset if the list retrieval operation has failed. The value is a string with the error code.

lists

this element contains an array of mailing list descriptors. Each descriptor is a dictionary with the following keys and values:

name

a string with the mailing list name

realName

the mailing list "description" string.

browse

the string describing the mailing list archive browsing policies. The string value can be anyone or subscribers.

WSSP page:

the `listlist.wssp` page is used to generate the HTTP response.

URL: /List/*listname*/, /List/*listname*/List.html

this URL is used to retrieve a portion of the mailing list records for the *listname* [Mailing List](#).

The code component actually uses the generic [Mailbox](#) Code Component.

Action

The component checks if the HTTP request contains the `NextMessage` parameter with a numeric value. If it exists, the value is interpreted as the unique message ID (UID) in the list archive mailbox, and the component tries to find this message in the selected mailbox "view", and tries to find the next message in the view.

The component checks if the HTTP request contains the `PrevMessage` parameter with a numeric value. If it exists, the value is interpreted as the unique message ID in the list archive mailbox, and the component tries to find this message in the selected mailbox "view", and tries to find the previous message in the view.

If the next or previous message is found, its UID is added to the dataset (see below) and the generic Mailbox component is not used for processing.

If no next/previous message is found, the generic Mailbox component is used to process the HTTP request parameters and to compose the resulting dataset.

Result Dataset

`listName`

a string with the Mailing List name.

If a next or previous message is found:

`messageJump`

a string with the found message UID.

If a next or previous message was not requested in the HTTP request parameters or it was not found:

`realName`

the Mailing List Description string

`charset`

the Mailing List Preferred charset string

the generic Mailbox code component is used to generate the rest of the resulting dataset.

WSSP page:

the `listmailbox.wssp` page is used to generate the HTTP response.

URL: `/List/listname/Message/uid.html`

this URL is used to retrieve the message with *uid* unique ID from the *listname* mailing list archive.

The code component actually uses the generic [Message](#) Code Component.

Action

The generic Message component is used to process the HTTP request parameters and to compose the resulting dataset.

Result Dataset

`listName`

a string with the Mailing List name.

`nextMsg`

this element is added if there is a next message in the archive mailbox view. The element value is a string with the next message UID.

`prevMsg`

this element is added if there is a previous message in the archive mailbox view. The element value is a string with the previous message UID.

the generic Message code component is used to generate the rest of the resulting dataset.

WSSP page:

the `listmessage.wssp` page is used to generate the HTTP response.

Error Pages

The WSSP mechanism is used to generate HTTP response body for error responses. The following table lists the HTTP error codes, the situations when this error code is generated, and the WSSP file used to product the HTTP error response body.

Code	Error conditions	WSSP file used
301	the requested resource has been moved	<code>moved.wssp</code>
404	the requested resource does not exist	<code>notfound.wssp</code>
401	the requested page requires the HTTP Authorization (request header)	<code>unauthorized.wssp</code>
401	the credentials in the HTTP Authorization request header are incorrect	<code>denied.wssp</code>
500	generic system error	<code>failure.wssp</code>
501	generic system error	<code>error.wssp</code>

These WSSP pages are processed using datasets generated for all Stateless requests, with the following additional elements:

`errorCode`

this element is added to the dataset if there is an error code to be reported to the user.

`hostName`

this element is added to the dataset if the HTTP request contained the `Host :` field. The element value is that request field parameter.

The `disconnected.wssp` page is used when an HTTP request was sent to a WebUser Session, but the session has not been found. The page is processed using a dataset generated for all Stateless requests.

Code Components for Session Requests

When a new WebUser Session is created, the Skin specified in the account WebUser Preferences is opened and is used as the "Session Skin". The string `StartPage` is retrieved from that Skin Text Dataset, and a jump page is composed and sent to the user browser. The jump page redirects the user browser into the "Session Realm", to the page specified with the `StartPage` string.

HTTP requests to the "Session realm" (requests with URLs started with `/Session/`) are processed as Session Requests. The second component of the Session Request URL is a unique Session ID, the HTTP module uses it to find the WebUser session.

If the specified session is not found, or the Session has the `Fixed Network Address` option set, and the HTTP request did not come from the same IP address as the Login request that started the session, the `disconnected.wssp` page is displayed (see above).

After the Session is found, the Web Application module processes the rest of the request URL as the "request inside this session realm". If the request URL specifies a regular file, that file is retrieved from the Session Skin, and it is sent back to the user browser.

For each `.wssp` request a code component is called. It processes the HTTP request parameters and generates a *result dataset*. Then the `.wssp` file is retrieved from the Skin, and it is used to compose the HTTP response.

If the result dataset does not contain the `blockAlerts` element, the Web Application module checks if there is a pending [Alert](#) for the session user. If one or several pending alerts are found, the

Alerts code component is called, and the Alerts.wssp file is used to compose the HTTP response.

The Web Application module places certain data into datasets produced with all Session Requests code components. The following list specifies these "generic" dataset elements that can be used with all Session WSSP pages:

messageText

This string element is added if the HTTP request contains the messageText parameter. The element value is the same as the HTTP parameter value.

messageCode

This string element is added if the HTTP request contains the messageCode parameter. The element value is the same as the HTTP parameter value.

secureChannel

This element exists and has the YES string value if the request has been received via a secure (HTTPS) connection.

charset

This string element contains the Preferred Charset selected in the User Preferences, if the Use UTF8 mode is set to Never. Otherwise, this element contains the utf-8 string.

Note: it's just a default value, individual code components can specify different values for this dataset element.

If a .wssp request specifies an unknown code component, but the .wssp file with the specified name can be retrieved from the Session Skin, that .wssp file is processed using the dataset with the "generic" elements only, and the result is sent back to the user browser.

Example: The Stock Skin uses Hello.wssp requests. There is no code component with this name, so a dataset with the generic values is composed, and the Hello.wssp file is used to process this dataset.

The following sections specify the names of existing code components (names for .wssp requests), the actions taken by these component, and the dataset produced with these components.

The code component results are processed using the .wssp files with the same names as the code component names.

Name: Mailboxes

Actions

If the HTTP request contains the `Create` parameter and the `NewName` parameter is a non-empty string, the component tries to create a mailbox with the specified name. If this operation fails, the `errorCode` element with the error code text is added to the result dataset. If the mailbox is created, the `messageCode` element with `MailboxCreated` string value is added to the result dataset, and the created mailbox name is added to the list of subscribed mailboxes, if the `Show Subscribed Mailboxes` option is selected in the Account WebUser Preferences.

If the HTTP request contains the `EmptyTrash` parameter, and the mailbox or mailbox alias with the name `Trash` can be with the "Can Delete" Mailbox Access Right, then the code component removes all messages from that mailbox.

Result Dataset

The code component creates a list of all account mailboxes and aliases (if the `Show All Account Mailboxes` option is selected in the Account WebUser Preferences), or the list of all subscribed mailboxes (if the `Show Subscribed Mailboxes` option is selected). If both options are selected, these two lists are merged into one.

If the HTTP request contains the `Filter` parameter, only the mailboxes with names containing this parameter value are included into the list.

`filter`

this string element contains the current value of the HTTP `Filter` parameter.

`newName`

this string element contains the current value of the HTTP `NewName` parameter.

`mailboxList`

this element is an array with one dictionary-type element for each mailbox in the composed mailbox list. Each dictionary contains the following elements:

`mailboxName`

this string element contains the mailbox name.

`nonSelectable`

this string element with the value `Yes` is added if the mailbox is not selectable. If it is added, none of the following elements is added to the dictionary.

`isList`

this element is added to the element if the mailbox is the main mailbox (archive) for a

mailing list (this also means that there is a mailing list with the same name).

`nMessages`

this string element contains the number of messages in the mailbox. If the number cannot be retrieved, the element value is the `???` string.

`nRecent`

this string element contains the number of "Recent" messages in the mailbox.

`size`

this string element contains the "rounded" size of the mailbox. If the mailbox size cannot be retrieved, the element value is the `???` string.

`nSelected`

this string element contains the number of elements in the `mailboxList` array.

`trashSize`

this string element is added only if the account has the Trash mailbox. It contains the Trash mailbox size.

`currentStorage`

this string element contains the "rounded-up" total size of account mailboxes.

`storageLimit`

this string element contains the "rounded-up" account total mailbox size limit. If account does not have the total mailbox size limit, this element contains the `unlimited` string.

Name: Mailbox

The HTTP request must contain the `Mailbox` parameter - the name of the mailbox to be displayed.

Actions

For each mailbox, the module creates a session object that contains the mailbox view parameters. When the module is created, these parameters are initiated with the Web User Preferences values.

The HTTP request `Msg` parameters are interpreted as "message set elements". A request can contain several parameters, and each parameter should have a numeric value - the Unique ID of a mailbox message.

If the HTTP request contains the `Forward` or `Redirect` parameter and the

`RedirectAddresses` parameter is not empty, a "message set" is composed using the `Msg` parameters, and the message set messages are forwarded or redirected to the specified addresses. If the operation has been successful, the `messageCode` element with the `MessagesForwardedInfo` or `MessagesRedirectedInfo` string value is added to the result dataset. Otherwise, the `errorCode` element with the operation error code string value is added to the result dataset.

If the HTTP request contains the `Copy` or `Move` parameter and the `MailboxName` parameter contains a name of some selectable mailbox, a "message set" is composed using the `Msg` parameters, and the message set messages are copied to the specified mailbox. If the `Move` parameter was specified, the message set messages are marked as `Deleted`, `deleted`, or moved to `Trash` - depending on the `WebUser Preferences`.

If the operation has been successful, the `messageCode` element with the `MessagesCopiedInfo` string value is added to the result dataset. Otherwise, the `errorCode` element with the operation error code string value is added to the result dataset.

If the `WebUser Preferences DeleteMethod` option is set to `Move To Trash`, and the HTTP request contains the `Delete` parameter, a "message set" is composed using the `Msg` parameters, the message set messages are copied to the `Trash` mailbox and deleted. If the `Trash` mailbox did not exist, it is created.

If the HTTP request contains the `read`, `unread`, `flag`, `unflag`, `delete`, or `undelete` parameters, a "message set" is composed using the `Msg` parameters, and the flags for the message set messages are modified. The `delete` and `undelete` parameters are processed in this way only if the `WebUser Preferences DeleteMethod` option is not set to `Move To Trash`.

If the operation has not been successful, the `errorCode` element with the operation error code string value is added to the result dataset.

If the `WebUser Preferences DeleteMethod` option is not set to `Move To Trash`, and the HTTP request contains the `Purge` parameter, all mailbox messages with the `Deleted` flag are deleted. If the operation has not been successful, the `errorCode` element with the operation error code string value is added to the result dataset.

If the HTTP request contains the `NextUnread` parameter and the mailbox contains an unread message, the `messageJump` element is added to the result dataset, and the code component stops request processing. The element string value contains the Unique ID of the first unread message.

Then the code component use the generic `Mailbox` component to process the HTTP request parameters and the compose the resulting dataset.

Result Dataset

`mailbox`

a string with the mailbox name.

If a next unread, next, or previous message is found:

`messageJump`

a string with the found message UID.

If a next unread, next, or previous message was not requested in the HTTP request parameters or it was not found:

`refreshTime`

the mailbox view refresh time (in seconds), retrieved from the WebUser Preferences.
the generic Mailbox code component is used to generate the rest of the resulting dataset.

Name: Message

The HTTP request must contain the `Mailbox` parameter (the name of the mailbox containing the messages to be displayed), and the `MSG` parameter - the Unique ID of that message in the mailbox.

Actions

If the HTTP request contains the `Copy` or `Move` parameter and the `MailboxName` parameter contains a name of some selectable mailbox, the message is copied to the specified mailbox. If the `Move` parameter was specified, the message is marked as Deleted, or it is deleted - depending on the WebUser Preferences.

If the operation has been successful, the `messageCode` element with the `MessageCopied` string value is added to the result dataset. Otherwise, the `errorCode` element with the operation error code string value is added to the result dataset.

If the `Move` operation has removed the message, the `backToMailbox` element with the `Yes` value is added to the result dataset, and the code component stops request processing.

If the HTTP request contains the `Redirect` parameter and the `RedirectAddresses` parameter is not empty, the message is redirected to the specified addresses.

If the operation has been successful, the `messageCode` element with the `MessageRedirected` string value is added to the result dataset. Otherwise, the `errorCode` element with the operation error code string value is added to the result dataset.

If the HTTP request contains the `TakeAddress` parameter the message `From:` address is added

to the Account address book.

If the HTTP request contains the `read`, `unread`, `flag`, `unflag`, `delete`, or `undelete` parameters, the message flags are modified.

If the operation has not been successful, the `errorCode` element with the operation error code string value is added to the result dataset.

Then the code component use the generic Message component to process the HTTP request parameters and the compose the resulting dataset.

Result Dataset

`mailbox`

A string with the mailbox name.

If the message has not been removed:

`MSG`

A string with the message UID.

`flagged`, `recent`, `deleted`, `flagged`, `isDraft`

These elements with the Yes value are added based if the message has the corresponding flags.

`status`

This string element has the following values:

- Deleted - if the message has the Deleted flag set, otherwise
- Draft - if the message has the Draft flag set, otherwise
- Redirected - if the message has the Redirected flag set, otherwise
- Unread - if the message does not have the Seen flag set, otherwise
- Answered - if the message has the Answered flag set, otherwise
- Read

`messageBody`

A string with HTML presentation of the message, generated using the generic Message code component.

`charset`

The charset to use for message display. This element can be set with the generic Message code component.

`nextMsg`

this element is added if there is a next message in the mailbox view. The element value is a

string with the next message UID.

prevMsg

this element is added if there is a previous message in the mailbox view. The element value is a string with the previous message UID.

Name: Compose

Actions

If the HTTP request contains the `charset` parameter, the parameter value is used as the *desired* charset - the charset to be used in the composed message.

The optional `Operation` HTTP request parameter specifies the type of the Compose operation and it can have the `Reply`, `ReplyAll`, `Forward`, or `EditDraft` value.

If this parameter is specified, the `OrigMessage` parameter (with the UID of the original message) and the `OrigMailbox` parameter (with the name of the mailbox containing the original message) must be specified.

If the HTTP request contains the `Operation` parameter and it does not contain the `filled` parameter, the original message header fields are used to compose the Subject, To, Cc, and the message body data for the new message.

Otherwise, the `Subject`, `To`, `Cc`, `Bcc`, and `Body` HTTP request parameters are used as the new message data.

If the HTTP request contains the `AddressBook` parameter and it does not contain the `CloseBook` parameter, or if HTTP request contains the `OpenBook` parameter the generic `AddressBook` code component is used to process the request parameters and to form some result dataset elements.

If the HTTP request contains the `Send` parameter, the composed message is submitted to the Server Queue. If the HTTP request contains the `Save` parameter, the composed message is stored as a Draft in the selected Drafts mailbox.

In both cases all HTTP request `Attachment` parameters are added to the message as attachments.

Result Dataset

operation

This string element is added to the result dataset if the HTTP request contains the `Operation` parameter. The element value equals the request parameter value.

origMessage

This element containing the UID of the original message is added to the result dataset if the HTTP request contains the `OrigMessage` parameter.

`origMailbox`

This element with the name of the mailbox containing the original message is added to the result dataset if the HTTP request contains the `OrigMailbox` parameter.

`sentOrSaved`

This element with `Yes` value is added to the result dataset if the Send or SaveDraft operation has completed successfully. If this element is added:

- The `sent` element with the `TT>Yes` value is added if the operation was the Send operation.
- The `messageCode` element with the `MessageSent` or `MessageSaved` value is added to dataset.
- No other element listed below is added to the result dataset.

`Subject, To, Cc, Bcc`

These elements contain strings with the current header fields data.

`From`

This element contains a string with the `From` address specified in the WebUser Preferences.

`addressBook`

This element with the `Yes` value is added to the result dataset if the HTTP request contains the `AddressBook` parameter and does not contain the `CloseBook` parameter or if the HTTP request contains the `OpenBook` parameter.

`body`

This string element contains the current message body text.

`mailerWidth`

This string element contains the `MailerWidth` WebUser Preferences option value.

`forwardedMessage`

This optional string element contains the HTML representation of the original message. This element is added to the result dataset if the HTTP request `Operation` parameter is `Forward`.

`DSN`

This element with the `Yes` value is added to the result dataset if the HTTP request contains the `DSN` parameter.

`SaveSent`

This element with the `Yes` value is added to the result dataset if the `WebUser Preferences` contain a non-empty `SentBox` option and the `HTTP` does not contain the `Filled` parameter or the `HTTP` request contains the `SaveSent` parameter.

`desiredCharset`

This string element contains the name of charset to be used in the composed message.

`charset`

This element is the UTF-8 string if the `Use UTF8 WebUser Preferences` option is set to "for Reading and Composing". Otherwise, this element contains the same value as the `desiredCharset` result dataset element.

Name: MailboxSettings

The `HTTP` request must contain the `Mailbox` parameter - the name of the mailbox to manage.

Actions

If the `HTTP` request contains the `Remove` parameter, the mailbox is removed. If the `HTTP` request also contains the `RemoveSub` parameter, all mailbox submailboxes are removed, too.

If the operation has been successful and the `Show Subscribed Mailboxes` option is selected in the `WebUser Preferences`, the deleted mailbox(es) are removed from the account subscription list. If the operation has been successful, the `removed` element with the `Yes` string value is added to the result data set and the code component stops request processing. Otherwise, the `errorCode` element with the operation error code string value is added to the result dataset.

If the `HTTP` request contains the `Rename` parameter and the `NewName` parameters is not empty, the mailbox is renamed. The `NewName` parameter value is converted into the "UTF-7 Mailbox Name encoding" format and is used as the new mailbox name.

If the `HTTP` request also contains the `RenameSub` parameter, all mailbox submailboxes are renamed, too.

If the operation has been successful and the `Show Subscribed Mailboxes` option is selected in the `WebUser Preferences`, the renamed mailbox(es) are renamed in the account subscription list. If the operation has been successful, the `removed` element with the `Yes` string value is added to the result data set and the code component stops request processing. Otherwise, the `errorCode` element with the operation error code string value is added to the result dataset.

If the `HTTP` request contains the `Update` parameter, the code component retrieves all `Acc` parameters from the request. Each `Acc` parameter should have a numeric value. For each retrieved `Acc` parameter value `nnn`, the `Znnn` parameter is retrieved. If it contains a non-empty string, all

Knnn request parameters are retrieved, where *K* is a [mailbox access right](#) letter.

The list of *Znnn* name strings with their *Knnn* parameter sets are used to form and set the new ACL list for the selected mailbox.

If the ACL update operation has been successful, the `messageCode` element with the `Updated` string value is added to the result dataset. Otherwise, the `errorCode` element with the operation error code string value is added to the result dataset.

Result Dataset

`renamed`

This element with the `Yes` string value is added to the dataset if the mailbox has been renamed.

In this case no other element is added to the result dataset.

`removed`

This element with the `Yes` string value is added to the dataset if the mailbox has been removed.

In this case no other element is added to the result dataset.

`rights`

This array element contains the mailbox ACL (Access Control List) elements. Each array element is dictionary with the following elements:

`ident`

this string element contains the ACL element *name*.

`index`

this string element contains the element number in the ACL set.

`lookup, select, seen, flags, insert, post, create, delete, admin`

these elements with `Yes` string values are added when the ACL element includes these mailbox access rights.

Name: Alerts

This code component can be called implicitly, if the Web Application module has detected a pending [Alert](#) message.

Actions

If the HTTP request contains the `AlertTime` parameter, that parameter should contain a time stamp in the ACAP format. The code component confirms all Alerts older than the specified time.

Result Dataset

alerts

This element is added to the result dataset if there are pending Alerts for the session user. The element value is an array of dictionary elements, each element describing one alert message. Each dictionary contains the following elements:

time

A string element containing the time when the alert was posted.

text

A string element containing the alert message text.

currentTime

This element is added to the result dataset if there are pending Alerts for the session user. Its string value contains the current time in the ACAP format.

Name: Subscription

Actions

If the HTTP request contains the `Open` parameter, the `MailboxName` parameter value is converted into the "UTF-7 Mailbox Name encoding" format, the converted string is added to the result dataset as the `jump` element, and the request processing ends.

If the HTTP request contains the `Update` parameter:

- All `Elem` request parameters are retrieved, converted into the "UTF-7 Mailbox Name encoding" format, and form the new Account Subscription list.
- All `AliasName` request are retrieved, they should contain numeric values. For each retrieved numeric value *nnn*, the parameters pairs *annn* and *mnnn* are retrieved. If both parameters exist and contain non-empty strings, the strings are converted into the "UTF-7 Mailbox Name encoding" format, and are used to form the new set of Account Mailbox Aliases.
- If the Subscription or Mailbox Aliases update operation fails, the `errorCode` element is added to the result dataset. Otherwise the `messageCode` element with the `Updated` string value is added to the result dataset.

Result Dataset

jump

The name of the mailbox to open ("to jump to"). If this element exists, none of the following elements is added to the result dataset.

subscription

This array element contains the Account subscription list. Each array element is a string with some mailbox name.

aliases

This array element contains the Account Mailbox Aliases list. Each array element is a dictionary with the following elements:

index

A string with this Mailbox Alias element index.

name

A string with the Mailbox Alias name.

ref

A string with the name of the Mailbox this Alias points to.

Name: Password

Actions

If the HTTP request contains the `ModifyPassword` parameter, the request should also contain the `OldPassword` parameter, and that parameter should match the current Account password. If the `OldPassword` parameter value is correct:

- The `RecoverPassword` request parameter value is set as the new `RecoverPassword` account setting. The `messageCode` element with the `Updated` string value is added to the result dataset.
- If the Account user is allowed to modify the Account password, the `NewPassword1` and `NewPassword2` parameters are checked. If they are non-empty and match each other, then the Account password is updated using these parameters value.

If the password has been updated successfully, the `messageCode` element with the `PasswordChanged` string value is added to the result dataset. If the password update operation failed, the `errorCode` element is added to the result dataset.

Result Dataset

`RecoverPassword`

This string element contains the Account `RecoverPassword` setting value.

Name: WebSite

The code component uses the generic WebSite component to process the HTTP parameters and to form the result dataset. Before the generic component is called, the following elements are added to the result dataset:

Result Dataset

`fileRef`

The `WebFile/` string.

`pageRef`

The `website.wssp` string.

Name: Bye

Actions

The code component can delete old messages from the Trash (as specified in the WebUser preferences) and closes the session. The session will be destroyed as soon as this HTTP request is processed, so the `bye.wssp` code can use session data, but the produced HTML code should not contain references to session objects.

Result Dataset

`blockAlerts`

This element has the `Yes` string value. It is added to the result dataset to prevent Alert processing.

Generic Code Components

The Web Application module has several generic components used to process both stateless and session requests.

Generic Mailbox component

Actions

If the HTTP request contains `Filter`, `Search`, `Limit` parameters, these parameter values are used to modify the mailbox "viewer" current *Filter*, *Search*, *Limit* values.

If the HTTP request parameter `Skip` exists, it should have a numeric value. This number is used to set the current *first message index* - the number of the first message to be displayed on this page.

If the HTTP request contains the parameter `Next`, then the current *first message index* is increased by the current *Limit* value.

If the HTTP request contains the parameter `Prev`, then the current *first message index* is decreased by the current *Limit* value.

If the HTTP request contains the parameter `Sort`, its numeric value specifies the number of "sorting" column (to sort the mailbox view by the first column, the `Sort` parameter should be 0).

If the HTTP request contains the parameter `SDir`, its numeric value specifies the sorting order: the value 1 requests ascending order, the value 0 - descending order, the value -1 reverses the current sorting order.

Result Dataset

`checkAll`

This element has the `CHECKED` string value. It is added to the result dataset if the HTTP request contains the `MarkAll` parameters.

`filter`

The string value of this element is the current *Filter* string.

`search`

The string value of this element is the current *Search* string.

`limit`

The string value of this element is the current *Limit* value (a number).

`headers`

The array value of this element contains mailbox view column headers. Each array element is a dictionary with the following elements:

`index`

This string element contains the column number.

`name`

This string element contains the column name.

hilited

This element has the YES string value and exists only if this column is the sorting column.

sdir

If this column is not the sorting column, this element contains the current sorting order (0 or 1) If this column is the sorting column, this element contains the reversed current sorting order ($1 - \text{current sorting order}$).

align

This element contains the string LEFT for text columns (From, Subject, Sender, etc.), and the string RIGHT for date- and size-type columns.

messages

The array value of this element contains the mailbox view data. Each array element is a dictionary with message data, and it contains the following elements:

id

this element contains the message Unique ID (UID)

fields

this array element contains message column data. The columns are stored in the same order as columns in the headers result dataset element. Each element is a dictionary. It contains the following elements:

hilited

This element has the YES string value and exists only if this column is the sorting column.

sdir

If this column is not the sorting column, this element contains the current sorting order (0 or 1) If this column is the sorting column, this element contains the reversed current sorting order ($1 - \text{current sorting order}$).

align

This element exists for date- and size-type columns. It contains the string RIGHT.

isRef

This element exists for the selected column and for the first "clickable" column. If it exists, it contains the string YES.

value

This element contains the column data. It exists for all columns except for the Status

column.

`isStatus`

This element exists if the column is the Status column. If it exists, it contains the string YES.

`status`

This element exists if the column is the Status column. If it exists, it contains the one of the following strings:

- If the message has the Deleted flag - Deleted, otherwise
- If the message has the Draft flag - Draft, otherwise
- If the message has the Redirected flag - Redirected, otherwise
- If the message does not have the Seen flag - Unread, otherwise
- If the message has the Answered flag - Answered, otherwise
- Read

`flagged`

This element exists if the column is the Status column and the message has the Flagged flag. If it exists, it contains the string YES.

`recent`

This element exists if the column is the Status column and the message has the Recent flag. If it exists, it contains the string YES.

`firstNumber`

This string element contains the number of the first message in the view.

`firstNumber1`

This string element contains the number of the first message in the view increased by 1.

`lastNumber`

This string element contains the number of the first message in the view increased by the number of the messages array elements if this array is not empty, or increased by 1 if the messages array is empty.

`numTotal`

The total number of messages in this mailbox.

`numUnread`

The total number of unread messages (messages without the Seen flag) in this mailbox.

`numSelected`

The total number of mailbox messages that can be displayed with the current Filter and Search values.

`multiPage`

This element with the YES string value is added if the `nSelected` value is not equal to the number of the `messages` array elements.

`sortColumn`

This element contains the number of the currently selected sorting column.

`sortAscending`

This element contains 1 if the currently selected sorting order is ascending, and 0 if it is descending.

Generic Message component

The generic Message component is used to convert the an RFC822 message into an HTML text. It processes simple and multi-part messages, attachments, digests, inline images and other letter components. To build a HTML presentation, the component uses strings from the Skin Text Dataset.



CommuniGate Pro WebMail

Guide

The CommuniGate Pro Server provides Web (HTTP/HTML) access to user accounts. The WebUser component works via the [HTTP module](#) and allows users to read and compose messages and to perform account and mailbox management tasks using any Web browser.

Even if you prefer a regular POP or IMAP mail client, the WebUser Interface can be used to access the features unavailable in some mailers. For example, the WebUser interface can be used to specify Subscriptions and Access Control Lists for account mailboxes - the features many IMAP clients do not support yet.

The WebUser interface is completely customizable, so the pages shown in this section may look differently on your Server.

WebUser Interface Pages

The WebUser Interface consists of several types of HTML pages that you can access using the controls - links and buttons.

Hello page

This page is displayed when you log into the system. It allows you to switch to other WebUser Interface pages, as well as to other portions of your site.

Mailboxes page

This page lists all mailboxes in your account and allows you to create, rename, and remove mailboxes, and to open mailboxes so you can browse the messages stored in your mailboxes. See the [Mailboxes](#) section for more details.

Mailbox page

This page lists all messages stored in the selected mailbox. You can copy, move, redirect, forward and delete listed messages. You can open and read messages listed on the Mailbox page. See the [Mailboxes](#) section for more details.

Message page

This page presents the content of the selected message. You can read the message, copy, move, delete, redirect, and forward the open message, and you can reply to it. See the [Messages](#) section for more details.

Compose page

This page allows you to compose a new message and send it. See the [Composing](#) section for more details.

Settings page

This page allows you to customize your WebUser Interface.

WebSite page

This page allows you to upload files to your Personal Web Site.

WebUser Account Settings

You can tune the WebUser Interface by modifying settings on the Settings page.

The Settings page contains the options that customize [Accesses to Mailboxes](#), [Mailbox Browsing](#), [Message Browsing](#), and [Message Composing](#). Besides, it contains some generic settings:

- Layout format (frames/no frames).
- The address-controlling option used to protect WebUser virtual sessions.
- The character set options.

The WebUser Interface Settings page contains a link to the account [Mailbox Subscription](#) and [Mailbox Aliases](#) page.

Password Modification

The WebUser Interface Settings page allows you to modify the account password:

Password Modification

Old Password:

New Password:

Reenter New Password:

To update your password, enter your current password, then enter your new password twice, and click the Modify button.

The WebUser Interface Account Settings page contains a link to the account [Public Info](#) page:

Personal Info		
Attribute	Value	What's this?
		country code
		office fax
		any useful info
		city
		home page
		mobile phone
		organization
		pager
		office phone
		job title

Users can update their Publicly Available Information by modifying the data in the value fields. To create a new attribute, a user should enter the attribute name into the empty attribute name field in the line table row. Settings any value field to the empty string deletes that attribute.

The editor always shows a set of well-known (though still not standardized) attributes, but it does not store them if their values are empty.

Public Info Editor

The WebUser Interface Settings page contains a link that allows you to open the Public Info Editor page. You can use this page to modify your Account data that is stored in the Directory. Other users can retrieve this information from the Directory, using any LDAP client, or Web access to the CommuniGate Pro Directory.

The System Administrator defines the set of Account Settings used as user's Public Info. If the Public Info Editor page contains no fields, the System Administrator has not specified any Public Info settings.

Name	Value
Work Phone:	
City:	

You can update the Public Info Account Settings by modifying the data in the value fields. Please note that the CommuniGate Pro Server can 'rename' Account Settings when storing them in the Directory. The City setting may be stored as the standard l directory attribute, while the Work Phone setting can be stored as the telephoneNumber attribute in your Directory record. The Server Administrator specifies the Account Settings <-> Directory Attribute renaming rules.

Automated Rules

The Web User interface provides access to the account [Automated Mail Processing Rules](#). If the Can Modify Account Rules [account option](#) is not enabled, then you can view the account Rules, but you cannot modify them.

You can turn the Auto-Reply option on and you can modify the Auto-Reply message text even if the Can Modify Account Rules option is not enabled for your account.

See the [Automated Mail Processing Rules](#) section to learn how to specify the Rules.

RPOP Accounts

The Web User interface provides access to the list of the [Remote POP Accounts](#) that the system polls on your behalf.

If the Can Modify RPOP Accounts [account option](#) is not enabled, then you can view the list of RPOP Accounts, but you cannot modify them.

See the [RPOP Module](#) section to learn how to specify the Remote POP Accounts to poll.



CommuniGate Pro WebMail: Mailboxes

One of the main functions of the CommuniGate Pro WebUser Interface is Web Access to user mailboxes. You can display the list of mailboxes in your account, create new mailboxes, rename and remove mailboxes, open and view mailbox, search mailboxes for certain data, etc.

Access to Mailboxes

The Mailboxes page displays your mailboxes. It allows the user to open the listed mailboxes and to create a new mailbox:

used 238K of 3M			
Filter:		3 selected	
Mailbox	Size	Messages	New
INBOX	20K	5	2
Friends	200K	56	4
Friends/Family	18K	3	

The mailbox names are links to the WebUser pages displaying the mailbox contents.

You can open the Settings page and specify which mailboxes should be displayed in the Mailboxes page:

Display All Account Mailboxes:	
Display Subscribed Mailboxes :	

Display All Account Mailboxes

If this option is selected, all mailboxes created in your account are listed.
















Display Subscribed Mailboxes

If this option is selected, the Mailboxes page lists all mailboxes your account is [subscribed](#) to (including foreign mailboxes).

If this option is selected, the newly created mailboxes are automatically added to the subscription list.

Mailbox Browsing

You can browse a mailbox by clicking its name (link) on the [Mailboxes](#) page. A mailbox page displays the messages stored in the mailbox, it provides checkboxes to select messages, and the controls for performing operations on the selected messages:

Filter:		Search:	
Status	From	Subject	Size Received
	TestList administration	Welcome!	871 25-Nov-99
	Technical Support	Fwd: [*] CommuniGate Pro 2.8b2 released	4K 25-Nov-99
	Technical Support	TEST - text & 2 gifs	11K 25-Nov-99
	Technical Support	Fwd: TEST - text & 2 gifs	13K 25-Nov-99
	philip@node5.stalker.com (no subject)		5K 25-Nov-99
	Technical Support	FWD:(no subject)	5K 25-Nov-99
	Technical Support	Fwd: HTML letter (alternative) (no subj	6K 25-Nov-99
	John R. Smith	Re: Weird Problems	2015 03-Dec-99
	Douglas M.	bad log file from our server	8K 11-Dec-99
	U&B	LDAP	1575 21-Dec-99
	James Green	design suggestion	3K 23-Dec-99
	Adam Drake	Transmission problems between SIMS and	4K 05-Jan-00
 Read:			
 Flagged:			
 Deleted:			
		Mailbox Management	

For each message in the mailbox, several message header fields are displayed. The messages are sorted by the highlighted field. The field name (link) can be used to highlight a different field and to change the sorting order.

A message can be opened using a link in the first and/or highlighted columns.

Display

This button tells the WebUser module to display not more than the specified number of the mailbox messages. If the Filter field is not empty, only the messages with the highlighted field containing the filter string are displayed. If the Search field is not empty, only the messages containing the search string are displayed.

Read

The Set button can be used to mark the selected messages as "read", the Clear button can be used to mark the selected messages as "unread".

Flagged

The Set button can be used to mark the selected messages with a flag, the Clear button can be used to remove the flag marker from the selected messages.

Copy To

This button can be used to copy the selected messages into the specified mailbox.

Move To

This button can be used to copy the selected messages into the specified mailbox; the original message is deleted or it is marked as deleted (if the WebUser Interface Delete Mode is set to Marked).

Redirect To, Forward To

This button can be used to redirect or forward the selected messages to the specified addresses. The address field below the buttons should contain one or several addresses separated with the comma signs.

Mailbox Management

This link can be used to open the [Mailbox Management](#) page.

The following buttons appear if the WebUser Interface Delete Mode is set to Mark

Delete

The Set button is used to mark the selected messages as "deleted", the Clear button can be used to clear the "deleted" markers.

Purge Deleted

This button is used to remove the messages marked as "deleted" from the mailbox.

If the the WebUser Interface Delete Mode is set to Via Trash or Immediately, the following button appears:

Delete

Click this button to move the selected message(s) to the Trash mailbox (the Via Trash Delete Mode) or to mark all selected messages as "deleted" and remove all marked messages immediately (the Immediately Delete Mode).

You can open the Settings page and specify how mailboxes should be displayed:

Mailbox Viewer	Display:	Refresh Every:
Fields: Status From Subject Size Received		Reverse
Sort:		

Display

This option specifies how many messages should be displayed on one Mailbox page. If a mailbox has more messages than this option specifies, an arrow links appear to allow you to "page" the entire Mailbox.

Refresh Every

This option specifies how often the Mailbox pages should be automatically updated.

Fields

This set of options specifies the message fields to be displayed in the Mailbox pages. Select - nothing- and click the Update button to remove a field.

Sort

This set of radio buttons allows you to select the field for initial (default) mailbox sorting.

Reverse

This option specifies the initial (default) mailbox sorting order.

The WebUser Interface Settings page also allows you to specify the Delete Mode:

Miscellaneous
Message Delete Method:

Move To Trash

The delete operation moves the selected message(s) to the Trash mailbox. This option available

for multi-mailbox accounts only. If the Trash mailbox does not exist, the first delete operation creates it.

Mark

The delete operation marks the selected message(s) as "deleted". The marked messages can be removed using the Purge Deleted operation.

Immediately

The delete operation marks the selected message(s) as "deleted" and then immediately deletes all marked messages from the mailbox.

Mailbox Management

The Mailbox Management page allows you to set the ACL ([Access Control List](#)) settings for the selected mailbox, to rename, and to remove the mailbox.

To rename a mailbox, type the new mailbox name into the New Mailbox Name field and click the Rename Mailbox button. If the Rename Submailboxes option is selected, all submailboxes of this mailbox will be renamed, too. If you are renaming the mailbox `Sent in 2000`, and you also have the `Sent/customers` submailbox, that submailbox is renamed into `Sent in 2000/customers` if the Rename Submailboxes option is selected.

Note: If you rename your INBOX, the new empty INBOX is automatically created.

To remove a mailbox, click the Remove Mailbox button. If the Remove Submailboxes option is selected, all submailboxes of this mailbox will be removed, too. If you are removing the mailbox `Sent`, and you also have the `Sent/customers` submailbox, that submailbox is removed, too - if the Remove Submailboxes option is selected.

Note: You cannot remove your INBOX.

To grant mailbox access rights to a user, enter the user name into the Identifier field, select the desired access rights, and click the Update button. To grant an access right to everybody, use the word `anyone`. To remove certain rights from a particular user, "grant" those rights to the identifier - `username`.

Access Control List						
Identifier	Lookup	Select	Seen	Flags	Insert	Post Create Delete Admin

Mailbox Subscription Management

The Subscription Management page allows you to set the [Mailbox Subscription](#) - the list of your own and foreign mailboxes you want to use.

You can open the Subscription page using the link on the Settings page:

Mailbox Subscription

Type a mailbox name into an empty field and click the Update button to add a mailbox to the subscription list.

To specify a [foreign mailbox](#), type the tilda sign (~), the user name, the slash sign (/) and then the mailbox name. Make sure that user has already granted you the Select access right for that mailbox.

Mailbox Aliases Management

The [Subscription Management](#) page allows you to set the [Mailbox Aliases](#) - the list of simple names for foreign mailboxes. You should use mailbox aliases if you want to access foreign mailboxes via IMAP clients that do not support the foreign mailboxes concept. It is

not recommended to use mailbox aliases with more advanced IMAP clients or with the WebUser Interface itself, since they add unnecessary complexity to mailbox management.

Mailbox Aliases	
Alias Name	Foreign Mailbox Name

Type a simple mailbox name into an empty field in the left column, type the name of a foreign mailbox into the right column field, and click the Update button to create a mailbox alias.

Your mail client software will list the created mailbox aliases as well as the real mailboxes created in your account. You can open a mailbox alias as you open a real mailbox, and the specified foreign mailbox will be opened.

Change the mailbox alias name to an empty string and click the Update button to remove the mailbox alias.

Access to Mailboxes by Name

You may want to access some mailbox without including it into your subscription list.

Open the [Subscription](#) page and type the mailbox name in the Open Mailbox panel and click the Go button:

Open Mailbox



CommuniGate Pro WebMail

Guide

The CommuniGate Pro WebUser Interface allows you to read messages stored in your account mailbox(es). To read ("open") a message, open the [Mailbox](#) page first and click on the message link.

Message Browsing

The WebUser Interface allows you to view messages in your mailboxes. It checks the MIME structure of a message and decodes its MIME parts.

A mailbox message is displayed as an HTML page, containing the important fields of the message header, the decoded message body, and the controls. Text, HTML, and graphics MIME parts are displayed, other parts (attachments) are shown as icon links that allow you to download these parts.

The `multipart`-messages are displayed according to the MIME multipart rules, and the nested messages (forwarded messages, reports, digests) are displayed, too.

The message header (and message headers of all embedded messages) has icon-links that allow you to view the complete header information, and to view the undecoded message body.

You can use the following controls (links/buttons) on Message pages:

Next Unread

Click this control to open the next unread message (a messages without the read marker) in the mailbox.

Back to *mailboxname*

Click this control to close the message and to open the mailbox page.

Close as Unread

Click this control to mark this message as unread (removes the read marker), to close the message page, and to open the mailbox page.

Delete

Click this control to mark this message as deleted (sets the deleted marker), to close the message, and to open the mailbox page.

Undelete

Click this control to remove the deleted marker from the message.

Reply

Click this control to open the [Compose page](#) and to send a reply message.

Reply To All

Click this control to open the [Compose page](#) and to send a reply message. The pre-composed recipients list will include not only the author of the original message, but all message Cc: and To: recipients, too.

Forward

Click this control to open the [Compose page](#) and to forward a message.

Set Flag

Click this control to add the flag marker to the message.

Reset Flag

Click this control to remove the flag marker from the message.

Copy To

Click this control to place a copy of the message into the specified mailbox.

Move To

Click this control to place a copy of the message into the specified mailbox and to remove the original message or to mark the original message as deleted (if the WebUser Interface Delete Mode is set to Marked).

Redirect

Click this control to redirect the message to the specified address(es). If you need to type in several addresses, separate them with the comma sign.

Take Address

Click this button to add the message author (From:) address to the account Address Book.

Edit Draft

This control appears only for *draft* messages; click this control to open the message in the [Compose page](#), so you can complete it and send it to its recipients.

You can open the Settings page and specify how messages should be displayed:

Message Viewer	Show HTML:
Fields:	

Show HTML

This option specifies how the WebUser Interface should process messages in the HTML format:
in frame

In this mode the HTML portions of messages are displayed in an *embedded frame*, providing complete separation of the message HTML code from the WebUser Interface Mailbox page code. If you use a browser that does not support embeded frames (Netscape 4.x), you will have to click a special link to open the HTML portion of the message in a separate window.

inline

In this mode the HTML portions of messages are inserted into the WebUser Interface Mailbox page code. The WebUser Interface checks the message HTML portion code and removes some tags that may distort the entire WebUser Mailbox page.

Fields

The fields listed in this set are displayed in the message headers.



CommuniGate Pro

WebMail: Composing Messages

CommuniGate Pro WebUser Interface allows you to compose E-mail messages and send them to one or several recipients - local and remote. You can use Address Book(s) to select message recipients.

Composed messages can contain one of several file attachments.

Composed messages can be saved as *drafts*, without actually sending them. Draft messages can be opened later, completed, and sent.

All sent messages can be stored in a designated mailbox.

Opening the Composer Page

The Composer page can be opened either directly, by clicking on the Compose link, or as a result of a Reply or Forward command.

The Composer Page contains the following panels:

- The message header panel with the To, Subject, To, and Bcc fields, and option controls.
- The message body text area.
- The attachment controls.

To send a message, fill the header panel fields, type the message text into the message body text area, and click the Send button.

Composer Settings

Each message you send using the CommuniGate Pro WebUser Interface contains your address as the message From address, and it can also contain your signature.

Use the Settings page to specify the options that apply to all messages you compose and send using the CommuniGate Pro WebUser Interface.

Message Composer	Auto Wrap:	Text columns:
From Address:	MIME-encode Headers:	
Signature:		
Store:	Sent Messages in	Drafts in

Text columns

This option specifies the width of the Composer field you use to enter your message texts.

Auto Wrap

This option specifies if the composed text should be hard-wrapped (cut into individual lines) before sending. If you disable this feature, some recipients may see entire paragraphs of your messages as single, very long lines.

From Address

This field allows you to specify the From: address for the messages you send using the WebUser Interface. By default, this address is set to the name of your Account on this Server.

MIME-encode Headers

If this option is set, message header fields containing non-ASCII (national) symbols are sent using MIME encoding.

Signature

This text is automatically added to all messages you compose using the WebUser Interface.

Store Sent Messages in

If this option is selected, a copy of all messages you compose using the WebUser Interface is stored in the specified mailbox.

Store Drafts in

If this option is selected, you can save partially composed messages in the selected mailbox. Later you can open these *draft* messages, complete, and send them.

Replying to Messages

When you read a message stored in your mailbox, you can click the Reply or Reply to All link/button. The Compose page appears and allows you to enter the text of your reply message.

If you click the Reply link/button, the original message Reply-To header field is automatically placed into the To field of your reply. If the original message did not have the Reply-To header, the original message From field is used.

If you click the Reply All link/button, the original message Reply-To/From address is copied to the To field of the reply message. Then all addresses from the original message To fields are added to the reply To field, and all addresses from the original message Cc fields are copied to the reply Cc field.

The text of the original message is formatted as a quotation and copied into the message body text area. The following Settings control the formatting process:

Message Composer	Auto Wrap:	Text columns:
Reply Header:	^T: time, ^F: sender, ^N: new line	
Reply Quoting:		

Reply Header

This string is added in front of the quoted text of the original message. It can contain special symbol combinations which are substituted with the original message data:

- ^T the date and time when the original message was sent
- ^F the From address of the original message
- ^N the EOL (end of line) character(s)

Reply Quoting

Each line of the copied original message text is prefixed with this string.

Note: If you set this option to an empty string, the original message text will not be included into a reply message.

Forwarding Messages

When you read a message stored in your mailbox, you can click the Forward link/button. The Compose page appears and allows you to specify the address(es) to which the message should be forwarded and a comment that will be sent along with the body of the forwarded message.

You can view the original message below the message body text area. The unmodified text of the original message is sent along with your comment, using the standard MIME format for message forwarding.

Attaching Files

You can attach one or several files to a message you want to send. Click the Browse button (or a similar button your browser displays for "file-type" fields) and select a file on your local disk (i.e. on the disk attached to the computer that runs your Web browser software). The name of the selected file appears in the Attachment field. Use other Attachment fields to send several files with your message.

Note: you should attach files after all other message fields are filled. If you click any button (for example, a button that opens the Address Book), the file selection will be cleared and you will have to select the files again.

The Composer page allows a user to specify the message subject, to enter the recipient address(es), to specify if the DSN (Delivery Status Notification) is required, to enter the message text, and to attach files to a message.

Delivery Status Notification

You can request a Notification when your message is delivered to the recipients.

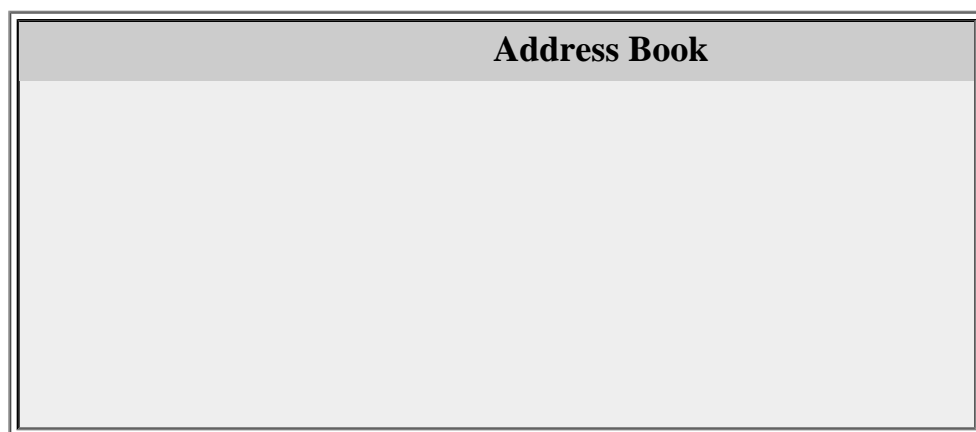
Select the Delivery Notification option to request DSN. DSN messages sent back to your INBOX tell you that your messages have been successfully delivered to the recipient mailboxes. They do not tell you if the recipient has actually seen/read your message.

Note: The DSN is guaranteed to work only when you send a message to other users on the same CommuniGate Pro Server. If a message is sent to a remote recipient, the remote server that serves that recipient account may or may not support the DSN feature. In many cases your CommuniGate Pro server can detect that a remote server does not support DSN. In this case your server will send you a DSN message itself, telling you that your message has been relayed to a remote host.

Address Book

The WebUser Interface allows you to update and use the account Address Book - the [addressbook dataset](#) that can keep individual E-mail addresses, groups, and other (nested) address books.

You can open the Address Book panel in the Composer page by clicking the Address Book button:



Close

Click this button to close the Address Book panel.

To, Cc, Bcc

Select one or several addresses in the list, and click one of these buttons to add the selected address(es) to the message you are composing.

Delete

Select one or several addresses in the list, and click this button to delete the selected address(es) from the address book.

Add New

Type or paste an E-mail address into the field on the right side, and click this button to add the address to the address book.

When you open any message, you can use the Take Address button to add the message author (From:) address to your Address Book.

The WebUser uses the [ACAP](#)-compatible format for the Address Book dataset, so the addresses entered via the WebUser interface can be used in any ACAP-savvy mailer, and the addresses added using such a mailer would appear in the WebUser Address Book panel.



CommuniGate Pro

WebMail: Secure Mail (S/MIME)

Secure Mail (S/MIME) is based on the Public Key technology. Using S/MIME, you can:

- digitally sign your message, so the recipient can:
 - verify that the message has been really sent by you;
 - verify that the message content has not been altered and that it was received in exactly the same form as it was composed by you (the sender);
- digitally encrypt your message, so only the recipients can read it, even if the message has been intercepted while it was being transferred, or if it has been copied from the server files that store the message.

Public Key Infrastructure (PKI)

The Public Key technology implements a so-called asymmetric cryptography. Using a regular, symmetric cryptography, both parties need to know some "key" or "password" (called "shared secret"). Before the parties can exchange data securely, they need to exchange that "shared secret", and this is the main security problem with symmetric cryptography: the "shared secret" can be stolen during the exchange process.

Imagine a spy who needs to exchange information with his center securely, using some secret key. That key must change frequently to ensure that the time needed to "break the key" is much larger than the "lifespan" of the information encrypted with that key. The center has to send those new keys to the spy (or vice versa), but those keys can be intercepted, and anyone who succeeds in intercepting the key will be able to decrypt all messages they send.

The Public Key technology uses pairs of specially generated keys. Both keys are very large numbers: they have 512 bits in length (approximately 60 decimal digits) or more. The special method used to generate those key pairs and the method used to encrypt information with those keys ensures that the message encrypted with one key can be decrypted with the other key. One key is called the "Private Key", the other key is called the "Public Key".

The PKI algorithms ensure that any data encrypted with the Public Key can be decrypted with the Private Key, any data encrypted with the Private key can be decrypted with the Public Key, and that it is extremely difficult to calculate the Private Key if the Public Key is known. Please note that messages encrypted with the Public Key cannot be decrypted with the same Public Key - they can be decrypted only with the Private Key.

Now we can see how this technology can be used by a spy, or any other party that needs to exchange information securely:

- The spy generates a pair of Public/Private Key using the key generation algorithm.
- The Private Key is stored securely at the spy's location.
- The Public Key is sent to the center using any type of communication - even a very "open" one. For example, the Public Key can be posted on the Web.
- When the center receives the Public Key, it uses it to encrypt the information it wants to send to the spy. It then sends it to the spy using any type of communication - again, it can just post the encrypted message on the Web.
- Since only the spy possesses the Private Key, only the spy can decrypt the information the center has sent. All other parties only have the Public Key and the encrypted data, but the Public Key cannot be used to decrypt the information, and it cannot be used to calculate the Private Key.
- The Center can also generate a pair of Private/Public Keys and send its Public Key to all its spies - so all of them can send message to the Center that only Center can decrypt, using its Private Key.

In real applications, PKI is not used to encrypt actual information. Instead, a random "regular key" ("shared secret", "password") is generated, actual information is encrypted using that shared secret, and PKI is used to encrypt that "shared secret" key. The encrypted "shared secret" key is appended to the actual information. The recipient uses its Private Key to decrypt the "shared secret", and then uses that "shared secret" to decrypt the actual data, using regular, "symmetric cryptography".

This method is used to decrease the amount of PKI computations (shared key is usually much smaller than the actual information), since PKI algorithms are much more complex than symmetric-key algorithms.

When it is said that the information is encrypted using 40 bit, or 56 bit, or 128 bit keys, it means that the random "shared secret" key used had this length - the PKI keys have much higher length. It

is much easier to break a 40-bit "shared secret" key used to encrypt data, than to break a PKI key used to encrypt that "shared secret". But "shared secret" keys are generated at random for each transaction, so if someone breaks the "shared secret key" used for any transaction, only that transaction will be compromised (decrypted), because other transactions with the same PKI keys will use different "shared secret" keys.

The method with 2 keys (PKI and "shared secret") allows a sender to send an encrypted message to several recipients at once. A message is encrypted using a random "shared secret" key, and then PKI is used to encrypt that "shared secret" several times, with Public Keys of all recipients. All encrypted "shared keys" are appended to the message, and each recipient can find the "shared secret" encrypted with its own Public Key, decrypt it with its Private Key, and use the decrypted "shared secret" to decrypt the actual information.

Digital Signatures

Encryption alone does not solve all security problems. If we return to our spy/center example, anyone who got the spy's Public Key can send an encrypted message to the spy. The spy needs to verify that the sender is really the center. The "digesting algorithms" and the Public Key algorithms are used to implement digital signatures.

Digest is a relatively short (16-40 bytes) number with a "checksum" of the message. The algorithms used for "digesting" ensure that it is very difficult to compose 2 different messages that would have the same digest values.

To sign a message, the sending software:

- calculates a "digest" for the message;
- encrypts the calculated digest using its own Private Key;
- appends the encrypted digest to the message.

The receiving party uses the sender's Public key (it is known to the receiving party) to decrypt the message digest, calculate the message digest itself, and compare the decrypted and calculated digests. If they match, the message has not been altered, and it was really sent by the party that has the proper Private Key.

In our spy example, a third party won't be able to send a message pretending to be the Center, because it does not know the Center Private Key. And if the third party encrypts the digest with some other Private Key, the signature verification will fail, because the spy will try to decrypt the digest with the Center's Public Key, and the resulting garbage will not match the calculated message digest.

Certificates

The encryption and signing methods assume that parties can freely exchange the Public Key information. The PKI eliminates the risk of "key stealing": the Public Key can be known to anybody (can be "publicly known"). But there is another risk - when a party receives the Public Key, it should verify that it really belongs to the proper entity. Otherwise a third party can generate its own Private/Public Key and send the Public Key to the center, pretending that it is the Public Key of the spy. If the center does not detect that this is a fake, it will use that key to encrypt the information it sends to the spy. The spy will not be able to read it (it is encrypted with the wrong Public Key), but the third party that has issued the key pair will be able to decrypt it, as it possesses the matching Private Key.

To solve this problem, the Public Keys are not distributed in the "raw form". Instead, they are distributed embedded into *Certificates*. A Certificate contains the following data:

- "Subject" - the name of the party the Certificate belongs to.
- Public Key - the Public Key of the "Subject".
- "Issuer" - the name of the party that has issued this Certificate.
- Signature - the digest of the data above, encrypted with the Issuer's Private Key.

Certificates are issued by a Certificate Authority - some party that all parties choose to trust. All parties should know the Public Key of the Certificate Authority. Modern Internet applications (browsers, mailers, etc.) have a built-in list of Trusted Authorities (including VeriSign and other similar companies), and have the Public Keys of those Trusted Authorities built-in.

When a certificate is received, the receiving party can verify if it has been issued by a "trusted authority": it checks if the "issuer" name in the Certificate is one of the "Trusted Authorities", and uses the already known Public Key of that Authority to verify the Certificate signature. If the signature is verified, the party can trust that the Public Key in the Certificate really belongs to the party specified in the Certificate Subject.

Very often an intermediate Certificate Authority is used. For example, a corporation can get a Certificate issued by a Trusted Authority, and then it can act as a Certificate Authority itself, issuing certificates for its employees. To enable verification of such a certificate by any third party, the Certificates issued by an Intermediate Certificate Authority are sent together with the Intermediate Certificate Authority own Certificate. The receiving party first checks that the Certificate is really issued by that intermediate Authority (by using the Public Key from its Certificate to verify the signature in the sender Certificate), and then it checks that the intermediate Authority is what it claims to be (by verifying its Certificate using the known Trusted Authority Public keys).

Private Key and Certificate Storage

In order to use PKI for Secure Mail, an Account should have its own Private Key and a Certificate with its Public Key. The Private Key should be protected as much as possible, while the Certificate should be easily accessible by anyone.

CommuniGate Pro stores the Certificate in the Account Settings (as the "userCertificate" element), and also it copies the Certificate into the Directory - if the Directory Integration is enabled.

CommuniGate Pro stores the Private Key in the Account Settings, but it encrypts the Private Key with a "Secure Mail Password". To use any of the Secure Mail functions, you should enter the "Secure Mail Password" to let the server read and decrypt your Private Key.

Note: The server does not store your Secure Mail Password anywhere. If you forget the password, you will need to obtain a new Private Key and Certificate. This means that your will not be able to decrypt any message encrypted with your old Public Key. Neither your System Administrator nor Stalker Software will be able to help you get those messages back.

Note: While it is very important to remember your Secure Mail Password, it is not too difficult to do: the Secure Mail Password can be a word or a phrase (up to 100 symbols), in any language.

You can use your regular E-mail client (such as Microsoft® Outlook or Netscape® Messenger) to obtain a personal Private Key and Certificate (also called "Digital ID"). You can then export that "Digital ID" to a .pfx or .p12 file - a so-called PKCS#12-formatted file. In order to protect your data, the E-mail client will ask you for a password, and will encrypt the exported information with that password.

Note: while the file format supports non-ASCII symbols in a file password, you should use ASCII symbols only, as many E-mail clients (including Outlook) do not process national symbols correctly.

Connect to the Server using the WebUser Interface, and open the Settings section. Click the Secure Mail link to open the page that contains the following fields:

Import Key and Certificate	
PFX File:	Secure Mail Password:
File Password:	Verify Secure Mail Password:

Note: If you do not see the Secure Mail link on your Settings pages, it means that your Account or Domain has the S/MIME service disabled.

Enter the name of the saved .pfx or .p12 file or use the Browse button to select the file on your workstation disks. Enter the File Password you used when you created that file.

Enter the password that will become your Secure Mail Password - this password will protect your Private Key on the CommuniGate Pro server. Enter this password twice, into two fields, and click the Import File Data button. If you have entered the correct File Password, the Certificate and Private Key information will be stored in your CommuniGate Pro Account settings.

The Secure Mail page now shows your Certificate data and the size of the Private Key.

Modify Secure Mail Password	Remove Key and Certificate
New Password:	Saved PFX File:
Verify Password:	File Password:
Export Key and Certificate	

To change your Secure Mail Password, enter the new password twice into the Modify Secure Mail Password panel fields and click the Modify Password button.

To store your Key and Certificate information in a file on your workstation disks, click the Export Key and Certificate link. A panel will open in a new window:

Export PFX file
File Password:
Verify File Password:

Enter the password to be used to encrypt your Key and Certificate information in the file (you need to enter it twice), and click the Export button. Your browser should ask you where to save the **CertAndKey.pfx** file (you can rename it).

If you decide to remove your Private Key and Certificate, you need to have their copy in a file. This is done to ensure that you can restore this info if you removed the Key by mistake. Remember that if you remove the Private Key completely and do not have a file to restore it from, all encrypted messages sent to you will become completely unreadable.

To remove the Key and Certificate, enter the name of the file that has the your Key and Certificate

and the file password, and click the Compare with File and Delete button. CommuniGate Pro will decrypt the file using the supplied password and it will compare it to your current Private Key. If the Keys match, the Private Key and Certificate are removed from your Account Settings.

Private Key Activation

When the Private Key is placed into the Account Settings, it is activated. The WebUser Interface automatically decrypts all messages encrypted with your Certificate/Public Key, and you can send encrypted and signed messages. In order to protect your sensitive information, your Private Key is automatically deactivated ("Locked") every 3 minutes. If you log out of the WebUser Interface session, and then log in again, your Private Key will not be automatically activated.

To activate your Private Key again, you need to enter the Secure Mail Password on any of the CommuniGate Pro WebUser Interface pages that displays the S/MIME Key Activation panel:

Secure Mail is Locked

SMIME Password:

Receiving Signed Messages

A message soted in your mailbox, or a message part can be digitally signed. When you open such a message, the WebUser Interface component automatically checks the integrity of the signed part. It retrieves the Signers data from the signature data and tries to verify the signature of all signers. It then shows the list of all signers whose signatures match the message content:

Signed Data (Text SHA1)

Dear Sir,

Thank you for your offer. I'm accepting it.

Sincerely yours,
Sample Sender

Content Unaltered as Verified By:

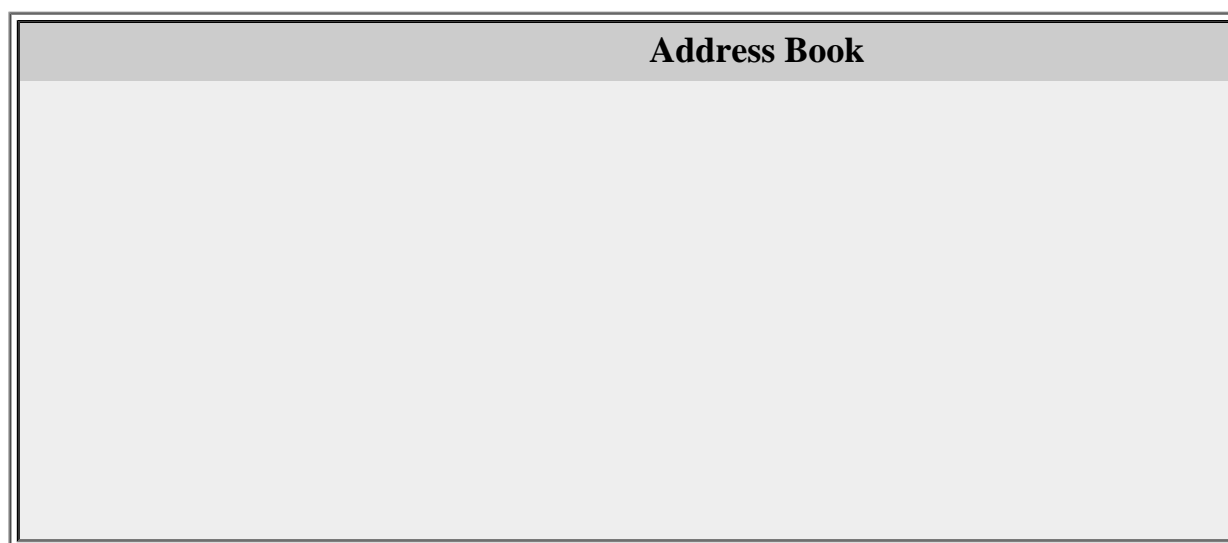
Sample Sender <sender@domain.dom>

If the information cannot be verified with any signature, an error message is displayed.

Recording Certificates

A Signed message contains the Certificate of the signer. The Take Certificate button that appears on the Message page when a Signed message is displayed. By clicking that button you include the E-mail address and the name of the signer (as specified in the Certificate, not in the message headers), and the signer certificate into your Address Book.

When you display an Address Book, the [@] marker indicates the entries that have known (stored) certificates. You can send encrypted messages to those addresses:



Sending Signed Messages

To Send a signed message, make sure that your Private Key is unlocked. If it is unlocked, you will see the Send Signed checkbox on your Compose page. Select this checkbox to sign your message. If you send a message with attachments, the entire content of your message, including all attachments, will be signed with your Private Key and your Certificate will be added to the message signature.

Recipients of your Signed message will be able to verify that the content has not been altered, and they will be able to store your Certificate and later send you encrypted messages.

Sending Encrypted Messages

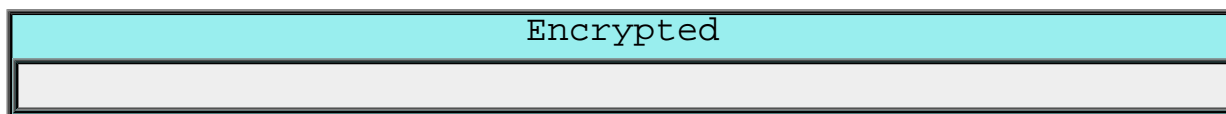
To Send an encrypted message, make sure that your Private Key is unlocked, and that all message recipients are included into your Address Book, and their Address Book entries contain certificates.

If your Private Key is unlocked, you will see the Send Encrypted checkbox on your Compose page. Select this checkbox to encrypt your message. If you send a message with attachments, the entire content of your message, including all attachments, will be encrypted with the recipients Public Keys (taken from their Certificates), and with your own Public Key. As a result, if a copy of the encrypted message is stored in your Sent mailbox, you will be able to read (decrypt) it.

If you select both Send Signed and Send Encrypted options, the message will be composed as a Signed message, and then the entire content (including the message headers and your signature) will be encrypted.

Receiving Encrypted Messages

When you receive an encrypted message, its content is not displayed:



You need to activate (unlock) your Private Key first. With the Private Key unlocked, the WebUser Interface module tries to decrypt all encrypted messages with your Private Key. If it succeeds to decrypt the message, the message content is displayed:





Miscellaneous

This chapter describes various CommuniGate Pro features not mentioned elsewhere.

Return Receipts

Senders can request return-receipts by including the `Return-Receipt-To:` header fields into messages. When a message containing a `Return-Receipt-To:` header field is delivered to a local Account, the Server generates a Delivery Notification message. That message is sent to the Return-Path address of the message, not to the address specified in the message `Return-Receipt-To:` header field.

Address Testing

If a message has the
`X-Special-Delivery: test`
header field, the SMTP and Local Delivery modules do not send the message to its recipients.

The SMTP module connects to all hosts the message is addressed to, then the module sends all recipient addresses to those hosts, but it does not send the message itself.

The Local Delivery module checks if the account exists, but the module does not try to apply the Account Rules to the message and the module does not store the message in the Account Inbox.

This feature can be used to verify addresses on large mailing lists: if an address contains an unknown domain name, or the host is not unreachable, or if the host rejects a user address, an error message is generated in the regular way, and can be used to detect "bad" addresses and to "clean" the mailing list.

Adding Required Headers

If a message does not have a properly composed RFC header part, the Server adds an RFC header to the message. This header contains the required header fields only.

If a submitted message does not have a Date: header field, the Server adds one using the date and time when the message was submitted to the server.

If a submitted message does not have a Message-Id: header field, and the message was received from a "trusted source", the Server adds a Message-Id: header to the message.

Legacy Mail Emulation

The CommuniGate Pro software package includes the command-line program `mail` (`mail.exe` for the Microsoft Windows platforms). You can use this program to submit messages to the CommuniGate Pro system, as you used the legacy `mail` program to submit messages to the `sendmail` MTA.

```
mail [-iInv] [-d base-directory]  
      [-s subject] [-f from-address]  
      [-c Cc-addresses] [-b bcc-addresses] to-addresses
```

`-i, -I, -n, -v`

These options are ignored; they are included for compatibility only.

`-f from-address`

Use the *from-address* as the message From: address. If this parameter is not specified, the current user name is used.

`-d base-directory`

Use the *base-directory* path as the location of the CommuniGate Pro *base directory*.

`-s subject`

Specifies the *subject* (only the first argument after the `-s` flag is used as a subject; be careful to quote subjects containing spaces).

`-c cc-addresses`

Send carbon copies to the *cc-addresses*; *cc-addresses* should be a comma-separated list of e-mail addresses.

`-b bcc-addresses`

Send blind carbon copies to the *bcc-addresses*; *bcc-addresses* should be a comma-separated list of e-mail addresses.

`to-addresses`

A comma-separated list of e-mail addresses.

The CommuniGate Pro software package includes the command-line program **sendmail** (`sendmail.exe` for the Microsoft Windows platforms). You can use this program to submit messages to the CommuniGate Pro system, using the interface of the legacy `sendmail` program.

```
sendmail  [-i] [-t] [-d base-directory]  
          [-f from-address] [-F sender-name] [-V envid]  
          [-Oparameter] [-oparameter] [address, ...]
```

`-d base-directory`

Use the *base-directory* path as the location of the CommuniGate Pro *base directory*.

`-i`

Ignore dots alone on lines by themselves in incoming messages. This should be set if you are reading data from a file.

`-t`

Read message for recipients. To:, Cc:, and Bcc: lines will be scanned for recipient addresses. The Bcc: lines will be deleted before transmission. The addresses listed on the command line will be excluded from the list of the recipients.

`-ffrom-address`

Use the *from-address* as the message From: address. If this parameter is not specified, the current user name is used.

`-Fsender-name`

Set the full name of the sender.

`-V envid`

The Enevelope ID of the message.

`-Oparameter`

`-oparameter`

Option ignored.

addresses

The destination addresses (without the `-t` option) or the addresses to be excluded from the the destination address list (if the `-t` option is specified).

The CommuniGate Pro mail and sendmail commands use the [Submitted folder](#) feature.

To submit messages from your OS/400 (IBM iSeries) programs, check the [CommuniGate Pro Sendmail API for OS/400](#) document.



CommuniGate Pro Licensing

The CommuniGate Pro software is available from Stalker [FTP](#) and [Web](#) sites, and their mirrors, on CD-ROMs, and from other sources. This software is a fully functional, unlimited version of the CommuniGate Pro server.

Until the Server is licensed, it adds a one-line banner to all messages it transfers. There are no disabled features in an unlicensed installation.

After you have installed and configured CommuniGate Pro software, you may run it in the trial mode as long as you need to check all its functionality and features, and to verify its stability in your particular configuration.

Only when the system is up and running to your full satisfaction, should you contact the Stalker Sales staff and purchase the License Keys.

License Keys

The CommuniGate Pro uses numeric License Keys. One Master key is issued to each installation. It encodes the Main Domain Name of the system, but it does not limit the number or names of [secondary domains](#).

Additional Keys enable various features of a licensed CommuniGate Server. All currently available CommuniGate Pro features are enabled for free in both trial and licensed versions, and the keys are needed only to allow you to create a certain amount of user Accounts and to create a certain number of Mailing Lists.

Single-Server Pricing (per one single- or multi-CPU Server Computer)

Basic License	Includes	Price
---------------	----------	-------

Professional	50 accounts, 5 mailing lists	US\$499.00	<i>edu</i>
Corporate	200 accounts, 15 mailing lists	US\$999.00	<i>edu</i>
Enterprise	1,000 accounts, 100 mailing lists	US\$1,999.00	<i>edu</i>
Small ISP	30,000 accounts, unlimited lists	US\$4,999.00	<i>edu, fin</i>
Midsize ISP	200,000 accounts, unlimited lists	US\$29,999.00	<i>edu, fin</i>
Large ISP	Unlimited accounts, unlimited lists	US\$59,999.00	<i>fin</i>
Upgrade	Next-Level Basic License	price difference	
Additional License	Includes	Price	
Mailing Lists	Additional 25 mailing lists	US\$199.00	<i>lst</i>
	Additional 100 mailing lists	US\$499.00	

edu Educational Discounts Available

fin Financing Available

lst Only one license per server

Note: The number of accounts is the number of actual [Accounts](#) created in all CommuniGate Pro [Domains](#). The number of domains, account and domain aliases, groups, and forwarders, as well as the number of mailing list subscribers is not limited.

Cluster Pricing

Entry-Level License	Includes	Price	
Dynamic Entry-Level	100,000 accounts, unlimited lists, 2 backends, 2 frontends	US\$99,999.00	<i>edu</i>
Static Cluster License	Includes	Price	
Static Cluster core	300,000 accounts	US\$39,999.00	<i>fin</i>
Static Cluster core	1,000,000 accounts	US\$69,999.00	<i>fin</i>
Static Cluster core	3,000,000 accounts	US\$119,999.00	<i>fin</i>
Static Cluster frontend	frontend server license	US\$15,000.00	

Static Cluster backend	backend server license	US\$25,000.00	
Static Cluster (sample)	300,000 accounts, 2 backends, 3 frontends	US\$134,999.00	
Static Cluster (sample)	300,000 accounts, 3 backends, 5 frontends	US\$189,999.00	
Static Cluster (sample)	1,000,000 accounts, 3 backends, 5 frontends	US\$219,999.00	
Dynamic Cluster License	Includes	Price	
Dynamic Cluster core	100,000 accounts	US\$99,999.00	<i>fin</i>
Dynamic Cluster core	300,000 accounts	US\$179,999.00	<i>fin</i>
Dynamic Cluster core	1,000,000 accounts	US\$349,999.00	<i>fin</i>
Dynamic Cluster core	3,000,000 accounts	US\$719,999.00	<i>fin</i>
Dynamic Cluster core	unlimited accounts	US\$999,999.00	<i>fin</i>
Dynamic Cluster frontend	frontend server license	US\$15,000.00	
Dynamic Cluster backend	backend server license	US\$30,000.00	
Dynamic Cluster (sample)	300,000 accounts, 2 backends, 2 frontends	US\$269,999.00	
Dynamic Cluster (sample)	300,000 accounts, 3 backends, 5 frontends	US\$344,999.00	
Dynamic Cluster (sample)	1,000,000 accounts, 3 backends, 5 frontends	US\$514,999.00	

Dynamic Cluster (sample)	1,000,000 accounts, 5 backends, 10 frontends	US\$649,999.00	
Special Licenses	Includes	Price	
Cluster of Clusters	Unlimited backends & frontends	US\$CALL	

edu Educational Discounts Available

fin Financing Available

Plugin Pricing (per one single- or multi-CPU Server Computer)

McAfee Anti-Virus Plugin

The McAfee License limits the number of messages the Plugin can scan within any 60 minute period of time. If the E-mail traffic exceeds the licensed limit, the Plugin suspends the CommuniGate Pro Queue processing module.

Note: The McAfee Anti-Virus Plugin is licensed for a **12 months period**. If you choose to provide non-interrupted virus protection for your users, you should re-purchase the Plugin License once a year.

Note: The McAfee Anti-Virus Plugin is **NOT** available for all platforms the CommuniGate Pro can run on. Before you order the McAfee Plugin License, make sure that the Plugin software runs on your CommuniGate Pro Server platform.

McAfee Plugin License	Scanner Performance	Price	
Light Traffic	50 messages/hour	US\$399.00	<i>edu</i>
Office Traffic	200 messages/hour	US\$799.00	<i>edu</i>
Enterprise Traffic	1,000 messages/hour	US\$1,599.00	<i>edu</i>
ISP Traffic	10,000 messages/hour	US\$3,999.00	<i>fin</i>
Unlimited/Cluster	unlimited traffic	US\$CALL	<i>fin</i>
Upgrade	Next-Level Plugin License	price difference	

edu Educational Discounts Available

fin Financing Available

Purchasing the License Keys

When you complete the CommuniGate Pro evaluation and verify that the CommuniGate Pro software operates as required by you and/or your organization, you should purchase the CommuniGate Pro License Keys.

Read, print, and sign the [CommuniGate Pro License Agreement](#), and fax it to

1 415 383 7461

You can call Stalker sales department at:

1 800 262 4722

1 415 383 7164

or E-mail to

sales@stalker.com

or order online at

[Stalker Web Site](#)

Visa, MasterCard, and American Express credit cards are welcome.

Contact Stalker Sales for information about check, wire transfer and purchase order acceptance policies.

Stalker Sales Staff will process your order and fax over the License Keys to be entered into the CommuniGate Pro Software Settings. This will eliminate the Trial Version banners.

A receipt and a copy of the License Keys will be mailed to you for your records.

Support

Basic Support	Includes	Price
Unlimited E-mail technical support	E-mailing to support staff at cgp-support@stalker.com	\$FREE
2 years of Updates	Downloading from ftp://ftp.stalker.com/pub/CommuniGatePro/	\$FREE
Mailing List Access	Automatic Subscribing at CGatePro-on@stalker.com	\$FREE
Knowledge Base Access	Archive search at http://mail.stalker.com/Lists/CGatePro/List.html	\$FREE

In addition to free unlimited E-mail support provided to all CommuniGate Pro customers, premium support packages are available, too.

Premium Support	Includes	Price	
Migration	up to 5 pre-scheduled <i>telesessions</i>	US\$600.00	
Emergency	a <i>telesession</i> within 4 hours (24x7), 5 incidents	US\$2,000.00	
Silver Pack	a <i>telesession</i> within 8 hours (24x7), up to 4 incidents/month	US\$5,000.00/year	
Gold Pack	a <i>telesession</i> within 6 hours (24x7), up to 4 incidents/month	US\$7,000.00/year	
V.I.C. Pack	a <i>telesession</i> within 4 hours (unlimited) (24x7), on-site support within 24 hours	US\$12,000/year	<i>loc</i>

telesession - phone and/or telnet login to the site
loc on-site support is not available in some locations

Training

Stalker Software runs regular training courses at our offices in Mill Valley, CA. Please E-mail your course preferences to training@stalker.com and we will notify you of upcoming dates.

Training	Includes	Price
Training at Stalker <i>location</i> (up to 3 students)	Basic Administrator Course (2 days)	US\$2,400
	Cluster Administrator Course (1 day)	US\$1,500
On-site Training	Basic Administrator Course (2 days)	US\$3,000 + <i>exp</i>
	Cluster Administrator Course (1 day)	US\$2,000 + <i>exp</i>
On-site Launch	installation, migration, testing, training (3+ days)	US\$1,500/day + <i>exp</i>

location Mill Valley, CA
exp Travel & Lodging expenses



CommuniGate Pro: How To

This section explains how you should configure your CommuniGate Pro Server if you have some specific needs.

Routing

How can I gradually migrate accounts from my old server?

In many cases, especially when you migrate users from an old server, you may want CommuniGate Pro to deliver mail to all accounts created in a certain domain, while mail to all accounts that do not [yet] exist in that CommuniGate Pro domain should be relayed to some other [old] server, without any change in the headers and envelope addresses.

Open the Domain Settings for that domain and set the Mail to Unknown option:

Mail to Unknown Accounts
is

Here `domain.dom` is the name of this CommuniGate Pro Domain and `otherserver.dom` is the DNS name of the other [old] server. If the DNS name for the other server does not exist, you can use the IP address instead:

```
*%domain.dom@[11.22.33.44]
```

When the CommuniGate Pro server receives any message directed to `aname@domain.dom`, and the domain does not have an account/group/forwarder/mailling list with that `aname` name, the message is Rerouted (the envelope address is changed) to `aname%domain.dom@otherserver.dom.smtp`. The `.smtp` suffix tell the SMTP module to accept this address, and cut the domain name part from the envelope address, using that part only as a name of the server to connect to (the SMTP module always cuts the IP-address type domain

parts, too). The resulting envelope address (aname%domain.dom) is converted to the standard form (aname@domain.dom) before it is sent to that other server. As a result, the other server receives such a message with the unmodified envelope data and header fields.

As soon the aname account is created in the CommuniGate Pro server domain.dom domain, mail starts to go to that account automatically. You can copy all messages from the aname account on the old server to the aname account on the new server and phase out the aname account on the old server.

SMTP Delivery

How can I relay mail for certain domains?

If you want your Server to act as a back-up mail relay for certain domains, you can enable the Relay to All Hosts We Backup option in the [SMTP module](#) settings. But this is not a perfect solution, since anybody with access to any DNS server would be able to use your server for unauthorized relaying.

To safely back-up the friend.com domain place the following record into the [Router](#) table:

```
Relay: friend.com = friend.com@friend.com.smtp
```

Read the [Protection](#) section to learn the meaning of the Relay: prefix (you can omit it, or you may want to use the RelayAll: prefix instead).

If you want to relay mail for the friend.com domain, but it should go to via a different server firewall.friend.com, use the following Router record:

```
Relay: friend.com = friend.com@firewall.friend.com.smtp
```

If you want to bypass the MX records and relay all mail to a certain IP address (specified explicitly or using a DNS A-record), then see the [Bypassing MX](#) section.

How can I send mail to a remote host bypassing its DNS MX records?

If your server should send mail to a domain target.domain via the relay relay.domain, you

can specify the IP address of that relay in the [Router](#):

```
target.domain = target.domain@[11.22.33.44]
```

You may want to relay mail using DNS A-records instead of explicitly specified IP addresses:

```
target.domain = target.domain@relay.domain.25.smtp
```

The [SMTP module](#) does not look at the MX records if the port number of a remote host is explicitly specified. By specifying the standard (25) SMTP port number, you tell the SMTP module to look for the relay.domain DNS A-record, and ignore its MX records.

Note: You may want to add a `Relay:`, `NoRelay:` or `RelayAll:` prefix

How can I hold all client mail till their servers send ETRN?

If your client has a symmetric dial-on-demand link (i.e. a link that is brought up by the provider when there is any traffic to the client hosts), that client may want:

- to get all mail via your server instead of receiving mail directly, when each incoming message brings the connection link up;
- to receive mail from your server only when the client software issues the ETRN command, so your server will not bring the link up and try to relay the client mail as soon as it is received.

To serve such a customer (the `client.com` mail domain), you should:

- create a DNS A-record for the `mail.client.com` name, pointing to the IP address of the client server;
 - create a DNS MX record for the `client.com` domain pointing to your CommuniGate Pro server; you should NOT include the `mail.client.com` name into the MX records for the `client.com` mail domain.
 - create a record in the CommuniGate Pro Router:
`client.com = mail.client.com.smtp`
 - include the `mail.client.com` name into the [SMTP module](#) Hold Mail for Domains setting.
-

How can I forward mail to the other SMTP MTA on the same server?

You may want to have two different SMTP Servers (MTA) running on the same computer, but listening on either different port numbers or on different IP addresses.

To relay mail to the "sibling" server running on the port 26, you can redirect to the domain other-port if you put the following record into your [Router](#) table:

```
other-port = 127.0.0.1.26.smtp
```

To relay mail to the "sibling" server running on the port 25, but on a different IP address 11.22.33.44, you can redirect to the domain other-ip if you put the following record into your [Router](#) table:

```
other-ip = 11.22.33.44.25.smtp
```

For example, if all mail to the domain client57.com should go to the sibling server running on a different port, place the following records into the Router:

```
other-port = 127.0.0.1.26.smtp  
Relay: client57.com = client57.com@other-port
```

or simply:

```
Relay: client57.com = client57.com@127.0.0.1.26.smtp
```

How can my customer servers receive mail if they have dial-up connections? (ETRN)

Small sites may have dial-up connections only and they can be off-line most of the time. To provide better mail delivery to those sites, you should use your CommuniGate Pro server as their back-up mail relay. You should:

- create 2 MX records (in your DNS server) for the customer domain.dom domain name: a high priority record pointing directly to the customer server and a lower priority record pointing to your CommuniGate Pro server;
- configure your server to let it [relay mail](#) to the domain.dom domain;
- optionally include the domain.dom name into the Hold Mail list of your CommuniGate Pro [SMTP module](#);
- configure the customer server to send the wakeup ETRN commands to your server.

How can I hold all client mail till their servers send ETRN?

If your client has a symmetric dial-on-demand link (i.e. a link that is brought up by the provider

when there is any traffic to the client hosts), that client may want:

- to get all mail via your server instead of receiving mail directly, when each incoming message brings the connection link up;
- to receive mail from your server only when the client software issues the ETRN command, so your server will not bring the link up and try to relay the client mail as soon as it is received.

To serve such a customer (the `client.com` mail domain), you should:

- create a DNS A-record for the `mail.client.com` name, pointing to the IP address of the client server;
 - create a DNS MX record for the `client.com` domain pointing to your CommuniGate Pro server; you should NOT include the `mail.client.com` name into the MX records for the `client.com` mail domain.
 - create a record in the CommuniGate Pro Router:
`client.com = mail.client.com.smtp`
 - include the `mail.client.com` name into the [SMTP module](#) Hold Mail for Domains setting.
-

How can my customer servers receive mail if they have dynamic IP addresses? (ATRN/PROP)

If a customer has a mail server and a dial-up connection with a dynamic IP address, the customer server cannot be listed in the DNS, because DNS records link domain names and fixed (static) IP addresses.

To deliver mail to those sites, you should configure your CommuniGate Pro server as their mail relay. Depending on the customer server capabilities, you can use either the ATRN or the Unified Domain-Wide Account (RPOP) method.

If the customer server supports the On-Demand Mail Relaying (ATRN) method, you should:

- create an MX record (in the your DNS server) for the customer `domain.dom` domain name; this record should point to your CommuniGate Pro server address;
- include the `domain.dom` name into the Hold Mail list of your CommuniGate Pro [SMTP module](#);
- create the `domain.dom` account in your CommuniGate Pro server Main Domain and assign some password to that account;
- configure the customer server to send the ATRN command to your server using `domain.dom` as the login (AUTH) name and the `domain.dom` account password as the AUTH password. If the customer software cannot send the ATRN command, it may send the TURN command, but only after it sends the AUTH command with the proper name and password.

If the customer server supports the Unified Domain-Wide Account method, you should:

- create an MX record (in the your DNS server) for the customer domain.dom domain name; this record should point to your CommuniGate Pro server address;
 - create the dd-customer account (actual name is not relevant) in your CommuniGate Pro server Main Domain and assign some password to that account;
 - add the following record to the CommuniGate Pro [Router](#):
domain.dom = dd-customer.local
 - configure the customer server to poll the dd-customer account on your CommuniGate Pro server;
 - configure the customer server to use the X-Real-To header field (or other field you have specified in the [Local Delivery](#) module settings) as the "special header" containing the mail envelope information.
-

How can my customers release mail to all their domains with one ETRN or ATRN?

Remote servers that use your CommuniGate Pro server as a back-up mail relay can serve multiple domains. Those servers usually send ETRN or ATRN commands specifying only one domain as the command parameter.

To let mail to all customer domains being released with one ETRN or ATRN command, you should enqueue mail sent to the customer "secondary" domains into the customer "main domain" queue.

If the remote server should receive mail for the domain1.dom, domain2.dom, and domain3.dom domains, but it sends ETRN or ATRN commands only for the domain1.dom domain, use the following Router domain-level records:

```
domain2.dom = domain2.dom@domain1.dom.smtp  
domain3.dom = domain3.dom@domain1.dom.smtp
```

Mail to all customer domains will be placed into the domain1.dom queue, and if you want to hold that queue till the ATRN/ETRN command is sent, include the domain1.dom name into the Hold Mail for Domains setting of the SMTP module.

Rules

How can I store all outgoing mail sent by all my users?

In a corporate environment, it may be necessary to store all outgoing mail into a mailbox in a system administrator or a security officer account.

Note: if your company chooses to copy employee mail, it **MUST** notify all server users about this policy.

To copy mail sent from certain domains, use a Server-wide [Rule](#):

Data	Operation	Parameter
Action	Parameters	

The account `security` should already exist in the main domain, and the mailbox `outgoing` should already exist in that account.

How can I restrict to whom my users can send mail?

In a corporate environment, it may be necessary to let certain groups of users send mail only to other members of that group and to only certain addresses outside that group.

The simplest way to implement restrictions is to organize these groups of users into CommuniGate Pro Domains. If all users in the Domain `dept1.company.dom` (except the user `boss`) are allowed to send mail only to the users in the same Domain and to the `supervisor@hq.company.dom` address, then the following Server-wide Rule should be used:

Data	Operation	Parameter
Action	Parameters	

How can I create an autoresponder that sends files or HTML messages?

You can use Rule "Reply" actions or the simplified AutoResponder Rule to generate messages in any MIME format. Just start the Reply text with the plus (+) sign and add all necessary MIME headers. Remember that the Subject field is not autogenerated in this case and that you have to specify the MIME-Version: header field, too.

Auto-Reply

You can use the same method to send non-text attachments:

Auto-Reply

The easiest way to compose such a message is to send the required file to your CommuniGate Pro account using MIME-encoding, and then open the message using the WebUser Interface. After verifying that the message has arrived intact, click the "Undecoded Letter" icon in the message header panel. The undecoded text of the message will be displayed in a new browser window. You can copy the encoded message body text and paste it to the Rule text field.

Mailboxes

How can I create and use Shared Mailboxes?

A shared mailbox is a mailbox in account X that can be used by a user (account) Y. Shared mailboxes can be used for incoming mail processed by a group of people (sales department, support department, etc.). Shared mailboxes can be used as an extremely fast and effective alternative to mail and distribution lists: the announce mailbox in the marketing account can be used to store all company announcements. If all employees have read access to that mailbox, a single copy of each announcement becomes available to everybody.

To use a Shared Mailbox, two steps must be taken: first, potential users of the shared mailbox should be granted access rights for that mailbox. On the second step the user mailers should be configured to access shared mailbox(es). Since these shared mailboxes belong to a different account, they are called *foreign* mailboxes.

First, the owner of the shared mailbox should create a regular mailbox within his/her account. It is useful to create a special account `public` and create shared mailboxes in that account. To grant others access rights to the shared mailbox, the account owner should use either a decent IMAP client that can deal with ACL (Access Control Lists) or the WebUser Interface. The [WebUser Interface](#) section describes how you can set the desired [Mailbox Access Rights](#).

If a shared mailbox is created inside the `public` account, it is useful to grant all Mailbox Access Rights to the real shared mailbox owner, so the owner can perform all operations with that mailbox without logging in as the user `public`.

To access shared mailboxes, user mailers should be configured to display both the user account's own mailboxes, and the available shared (foreign) mailboxes. The most universal method is to use the account [Mailbox Subscription](#) list. This list is a simple set of mailbox names, and both account's own mailbox and foreign mailbox names can be included into that list.

Many IMAP clients can only use the Mailbox Subscription list, but they cannot modify that list, or they do not allow a user to enter a foreign mailbox name into that list. In this case IMAP users should use the [WebUser Interface](#) to fill their subscription lists. If a shared mailbox announce has been created in the account `marketing`, users should put the `~marketing/announce` foreign mailbox name into their subscription lists.

The domain administrator can use the [Account Template](#) to specify the initial Mailbox Subscription list, so all new accounts automatically get subscriptions to some shared mailboxes.

When shared mailboxes are included into the Account Subscription List, the users should configure their mail clients to display all mailboxes listed in the Subscription List:

- WebUser Interface users should check that the Show All Subscribed Mailboxes [Setting](#) is selected.

- Microsoft® Outlook Express users should open the IMAP account Properties panel and enable the Advanced setting called `Only Show Subscribed Folders`. Since in this mode the Outlook Express mailer shows ONLY the mailboxes listed in the account Mailbox Subscription list, the users should include their own mailboxes (Sent, Drafts, etc.) into their Subscription lists.
- Netscape® Messenger users should open the IMAP Mail Server Properties panel and enable the Advanced setting `Show only subscribed folders`. Since in this mode the Messenger mailer shows ONLY the mailboxes listed in the account Mailbox Subscription list, the users should include their own mailboxes (Sent, Drafts, etc.) into their Subscription lists.

The Messenger automatically scans the `public` account and displays its shared mailboxes made available for the Messenger user. As a result, if all shared mailboxes are created in the `public` account, Netscape Messenger users should not do anything with the Mailbox Subscription Lists.

Some clients (including Microsoft Outlook and Outlook Express) cannot display foreign mailboxes even if those mailbox names are included into the account subscription list. Users of these mailers can access foreign mailbox via [mailbox aliases](#). They should use the [WebUser Interface](#) to specify aliases for foreign mailboxes they want to access. If a shared mailbox announce has been created in the account `marketing`, users should create the `mkt-announce` mailbox alias for the `~marketing/announce` foreign mailbox. Their IMAP clients will display the `mkt-announce` name and will provide access to the `~marketing/announce` mailbox messages.

The domain administrator can use the [Account Template](#) to specify the initial Mailbox Aliases, so all new accounts automatically get a predefined set of mailbox aliases for the specified shared mailboxes.

How can an Administrator clean User Mailboxes?

Sometimes a Server or Domain Administrator should be able to check user mailboxes to clean or file user messages. This can be done without actually logging to the Server under that user name.

The Server Administrator with the `All Accounts` [access right](#) has unlimited access rights to all mailboxes in all accounts on the Server. The Domain Administrator with the `CanAccessMailboxes` [access right](#) has unlimited access rights to all mailboxes in that domain accounts.

Administrators can use any decent IMAP client to access user mailboxes. That client should be able to let users enter the mailbox name directly. To open the INBOX in the `username` account, administrators should log in under their own names and tell the IMAP client to open the

~username/INBOX mailbox.

The [WebUser](#) Interface can be used for the same purpose. Administrators can log in under their own names, open the Subscription page and type the user mailbox name in the Open Mailbox panel.

How can I provide username.domain.dom personal Web Sites?

The standard URL for Personal Web Site of the username@domain.dom account is
`http://domain.dom/~username`.

You may want to provide more "nice-looking" `http://username.domain.dom/` URLs for your account Personal Web Sites. This feature is based on the method the CommuniGate Pro server uses to process [HTTP requests](#) sent to the WebUser port(s).

For users in a secondary domain domain.dom, add the following records to the [Router](#):

```
*.domain.dom = *@domain.dom
<LoginPage%*@domain.dom> = *@domain.dom
```

If the domain.dom is your Main Domain, then add the following records:

```
*.domain.dom = *@fict
<LoginPage%*@fict> = *
```

These records route the `LoginPage@username.domain.dom` addresses to `username@domain.dom` addresses (or `username` addresses if domain.dom is the main domain).

Finally, you have to update your DNS server to ensure that all `username.domain.dom` names point to your server IP address. You may want to use wildcard records (`*.domain.dom CNAME domain.dom`) if your DNS server supports them.



CommuniGate Pro: Help Me

This section lists the most common problems with the CommuniGate Pro installations, and it provides the suggestions that should help you to solve those problems.

WebAdmin

I have rerouted the Postmaster account and now I cannot log in as the Postmaster.

CommuniGate Pro applies routing rules not only to addresses in incoming messages, but to all addresses it processes. If you have rerouted the `postmaster` account to some other account `abc`, then all attempts to log in as the `postmaster` will cause the Server to try to open the `abc` account. If you provide the correct password (i.e. the `abc` account password), you will be able to log in, but you will have the access rights granted to the `abc` account, not to the `postmaster` account.

You still can log into the `postmaster` account even if the `postmaster` name is redirected to a completely different address. Use the following name instead of the `postmaster` name:

```
abcd@postmaster.local
```

This address is always routed to the account `postmaster`. Use the regular `postmaster` account password with this string.

For more details on the `.local` routing, check the [Local Delivery Module](#) section.

I have deleted the Postmaster account.

If you have deleted the `postmaster` account, stop the Server and start it again.

If the CommuniGate Pro Server does not find the `postmaster` account during the startup process, it creates a new one. Check the `postmaster` account files to get the new `postmaster` password, in the same way you used when you [installed](#) the CommuniGate Pro Server.

I have created a secondary Domain and now I cannot log into WebAdmin.

When you connect to CommuniGate Pro via a browser, the Server checks the domain name you have specified in the browser URL. If that name matches the name of one of your Secondary Domains, the WebAdmin Interface of that Domain is opened, rather than the Server WebAdmin Interface.

To open the Server WebAdmin Interface, use the Main Domain Name in your browser URL. If that name does not have a DNS A-record or its record points to a different server, use the Server IP Address in the browser URL.

If all Server IP Addresses were assigned to secondary Domains, you can try to use ANY domain name that points to the CommuniGate Pro Server, and does not match any of the Secondary Domain names.

If all Server IP Addresses were assigned to secondary Domains and all DNS domain names pointing to your server are names of your secondary Domains or secondary Domain Aliases, then use the following URL:

```
http://sub.domain.com:8010/MainAdmin/
```

```
https://sub.domain.com:9010/MainAdmin/
```

where *sub.domain.com* is any name pointing to your server computer or any of its IP addresses.

When I try to log in, I get the "access from your network is denied" error.

You have selected the Reject all Non-SMTP connections from Non-Client Addresses option on the Protection page. Now you can connect to the Server only from the addresses listed in the Client IP Addresses field (on the same page). If that field was left empty, you still can connect to the Server if you launch your browser on the Server computer itself, and connect locally.

If you have not entered anything into the Client IP Addresses field, or if you cannot connect from the IP Addresses listed in that field, then:

- stop the CommuniGate Pro Server;
- open the `{base}/Settings/IPAddresses.settings` file and change the `ClientOnly` option from YES to NO, and save the updated file.
- start the CommuniGate Pro Server again.

SMTP Receiving

My Server does not accept mail from my Web script/applet.

When the SMTP module receives messages, it tries to route the address specified in the Mail From command (the message 'Return-Path' address). If the domain name in that address is a name of the Server local Domain and the specified Account (or other Object) is not found in that Domain, the Router returns an error code and the SMTP module refuses to accept the message.

You should reconfigure your script/applet to use either an empty Return-Path (<>) for generated messages, or to use an E-mail address of some existing Account. If the script/applet cannot be reconfigured, you can create an Alias for any existing Account.

If, for example, your script/applet submits messages to your server with the <webform@mydomain.com> Return-Path address, and you do not have the webform Account in the mydomain.com Domain, you may want to create the webform alias for the postmaster Account. If delivery of a submitted message fails, the error report will be sent to the postmaster Account.

SMTP Sending

My Server cannot send mail to some host using SSL/TLS.

When the CommuniGate Pro SMTP module connects to a mail host/relay and tries to establish a secure (SSL/TLS) connection, it receives the host Certificate and check the name in that certificate. That name should match either the name of the domain the mail should go to, or the MX relay name for that domain.

When a remote server hosts several domains on the same IP address, it always sends out only one certificate, because the server cannot learn to which domain the incoming messages will go to and thus it cannot present the Certificate for that particular domain. As a result, your (sending) server may refuse to proceed.

If the server mainhost.com also hosts client1.com and client2.com domains, and the MX records for all 3 domains point to the same name and to the same IP address on that server, the server will always present only one Certificate - usually, the mainhost.com Certificate.

To allow your CommuniGate Pro server to send mail securely to client1.com and client2.com domains, you should specify 2 Domain-level [Router](#) records:

```
client1.com = client1.com@mainhost.com.smtp
```

```
client2.com = client1.com@mainhost.com.smtp
```

These records will place mail to client1.com and client2.com domains into the mailhost.com SMTP queue. You should place the mainhost.com name into the Send Encrypted list of the SMTP module, and the server will connect to the mailhost.com server, check its certificate (it should contain either the mailhost.com name or the name of the relay the SMTP module connected to), and then the SMTP module will establish a secure (SSL/TLS) connection with that server and it will send mail to recipients in the client1.com and client2.com domains via that secure connection.

Access

WebUser connections return the pink page saying "we do not provide Web Access to this domain"

It is very important to understand that the domain name `something.com` and `mail.something.com` are completely different domain names. If your CommuniGate Pro Server has the main domain `mycompany.dom`, and you are trying to connect to it by typing `http://mail.mycompany.com:8100` in your Web browser, you will get the page saying that the CommuniGate Pro Server does not provide access to the `mail.mycompany.com` domain.

In most cases, you want the domain names `mail.mycompany.com`, `webmail.mycompany.com`, etc. to be just other names (aliases) of the `mycompany.com` CommuniGate Pro Domain. To specify this, open the `mycompany.com` Domain Settings page and find the Aliases table. In an empty field, enter the `mail.mycompany.com` name and click the Update button. Now the CommuniGate Pro Server will know that `mail.mycompany.com` domain name is just a different name for the `mycompany.com` Domain it serves. Connection requests specifying the `mail.mycompany.com` domain name will connect to the `mycompany.com` CommuniGate Pro Domain, and messages sent to a `username@mail.mycompany.com` address will be delivered to the account *username* in the `mycompany.com` domain.

Note: The WebAdmin interface defaults to the main domain if the name specified in the browser URL is not a CommuniGate Pro Domain name. This is why connections to the WebAdmin port (8010) can work, while the connections to the WebUser port (8100) return the "pink page".

WebUser sessions are disconnected almost immediately after login.

When a user connects to your server via a "multi-homed HTTP proxy" (used by large ISPs such as AOL), TCP connections come to the CommuniGate Pro Server from several different IP addresses

of those proxy servers. If the `Require Fixed Network Address` option is enabled in the Account WebUser Preferences, user browser connections can be rejected. Disable the `Require Fixed Network Address` option for those users that connect via "multi-homed proxy" servers. If most of your users connect via those proxy servers, you may want to disable this setting in the Domain Account Defaults or in the All-Server Account Defaults.

What does the "unassigned local network address" error mean?

Your CommuniGate Pro server computer has one or several IP (network) addresses assigned to it. Those addresses can be assigned to CommuniGate Pro Domains, and the Domains WebAdmin page shows all Domains with the IP addresses assigned to them.

Usually, the Main Domain has the IP Addresses setting set to "All Available", so all addresses not assigned to secondary domains are automatically assigned to the Main Domain. If none of your Domains has the IP Addresses setting set to "All Available", then some of your server IP addresses may be not assigned to any Domain.

When a user connects to the server using a POP or IMAP client and provides just the account name (without the domain name), or when a secure (SSL/TLS) connection has to be established, the CommuniGate Pro Server takes the local IP address the user has connected to and tries to find the Domain that address is assigned to. If that IP address is not assigned to any CommuniGate Pro Domain, then the "unassigned local network address" error is generated.

Open the WebAdmin Settings->General page to see all the Local IP Addresses of your Server. You may have to click the Refresh button to see all addresses. The unassigned addresses are displayed in red.

Directory

Microsoft LDAP (Outlook and Outlook Express) users cannot find Directory records.

Most of LDAP clients (including the Microsoft Outlook products) contain a setting specifying the Directory subtree that should be used for search operations. In Outlook Express, this setting can be found in the Directory Account Properties, on the Advanced tab. It is called Search Base and it should contain the DN for the user domain (by default, that DN is `cn=domainname`).

If this setting field is left empty, Outlook products silently replace it with the `c=country_code` string, and search operations fail (unless your Directory has the `c=country_code` subtree).

If you do want to search the entire Directory with an Outlook product, enter the word `top` into the

Search Base setting field.

Attempts to update Account Settings result in the directory record with the specified DN is not found error.

This error appears when the [Directory Integration](#) option is enabled. This option tells the CommuniGate Pro Server to update the Account record in the Central [Directory](#) every time the Account Settings are updated. If the Directory does not contain a record for that account, the error message is returned. Account records may be missing in the Directory if the Accounts were created when the Directory Integration option was disabled.

To fix the problem, open the Domain Settings and find the [Directory Integration](#) panel. Click the Delete All button. It will remove all Domain object records from the Directory. Then click the Insert All button. The CommuniGate Pro Server will create a Directory record for the Domain, and then it will create Directory records for all Domain Objects (Accounts, Groups, Mailing Lists).

Note: if the Domain contains more than 100,000 Accounts, the Insert All operation can take several minutes.

Date and Time

Time stamps in messages sent or received with CommuniGate Pro are several hours off.

This problem is caused by an incorrect Time Zone setting on the server and/or on the client machines. To check the Time Zone setting value on the server machine, open the General page in the Settings realm of the CommuniGate Pro WebAdmin Interface. The Server Time field should contain the correct Date and Time values **and** the correct Time Zone value: -0800 means '8 hours behind the GMT', +0800 means '8 hours ahead of GMT'.

If the Time Zone value is incorrect, fix the OS settings that specifies that value, and re-open the General page to verify the Time Zone value.

Logs

Every time I access the WebAdmin interface, a Failure-type ROUTER record appears in the Log

.

The WebAdmin interface adds the `LoginPage@` string to the domain name you specify in your browser URL field and tries to route the resulting address as any other E-mail address. If routing fails, the WebAdmin Interface defaults to the main domain and to the Server WebAdmin Interface, but the failure record appears in the Router:

```
ROUTER failed to route 'LoginPage@mail'
```

Usually this happens when you use a non-qualified domain name (like `mail`) instead of the qualified domain name (`mail.mycompany.com`). You should either use the qualified domain name in your browser URLs, or you should add the `mail` Domain Alias to the `mail.mycompany.com` CommuniGate Pro Domain.

What do these DNR-16538(xxx.xx.x.xx.rss.mail-abuse.org) A:host name is unknown records mean?

When your SMTP module uses RBLs to check the IP address of the server that tries to send any mail to your server, it converts that server `aa.bb.cc.dd` IP Address into the `dd.cc.bb.aa.rbl-server-name` domain name, and tries to resolve this name using the DNS system. If the sending server is not a known offender, and its address is not included into the RBL database, this composed domain name will NOT exist in the DNS system, and the DNR module will report this with a Problem-level Log record.

If you use RBL servers, you may want to restrict the DNR module Log Level to Major & Failures events only.

Misc

What is that UDP port the CommuniGate Pro Server opens on my system?

This is a DNR (Domain Name Resolver) socket. The port number is selected by the OS, and it can change if you restart the CommuniGate Pro Server. This socket is used to send requests (UDP packets) to DNS servers and to receive responses from those servers.

Other applications (servers, browsers, etc.) use the same type of sockets to resolve domain names, but they usually open and close those UDP sockets quickly, so you may not notice them in your `netstat` output. CommuniGate Pro opens the DNR UDP socket when it starts, and uses that socket for all DNR requests, closing the socket only when the Server shuts down.

How can I make my `formmail`-type CGI work with CommuniGate Pro?

`Formmail` and similar CGIs are used to send E-mail messages from regular Web Server HTML forms. Implemented in the form of a [Perl](#) script, these CGIs use the legacy `sendmail` program to

send the composed messages.

On most platforms, CommuniGate Pro software installer does not replace the legacy `sendmail` program, though the package does contain the `sendmail` replacement program. In order to use that program, you should modify your Perl script: you should find all references to the `sendmail` program (usually the default path used is `/usr/sbin/sendmail`), and replace them with the `{application directory}/sendmail` references.

For example, if CommuniGate Pro and your CGI are installed on a MacOS X system, where the CommuniGate Pro *application directory* is `/usr/sbin/CommuniGate/`, the CGI script `/usr/sbin/sendmail` strings should be replaced with the `/usr/sbin/CommuniGate/sendmail` strings.



Appendix A: Supported Standards

The CommuniGate Pro is based on Internet standards (RFCs). Additionally, it has many unique capabilities that have quickly become the "must-have" features for modern industrial-strength messaging systems.

Kernel

Supported Standards	
RFC2387	The MIME Multipart/Related Content-type. E. Levinson. August 1998.
RFC2183	Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field. R. Troost, S. Dorner, K. Moore. August 1997.
RFC2076	Common Internet Message Headers. J. Palme. February 1997.
RFC2047	Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text. K. Moore. November 1996.
RFC2046	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N. Borenstein. November 1996.
RFC2045	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. N. Freed & N. Borenstein. November 1996.

RFC2044	UTF-8, a transformation format of Unicode and ISO 10646 Yergeau, F. October 1996.
RFC1894	An Extensible Message Format for Delivery Status Notifications. K. Moore & G. Vaudreuil. January 1996.
RFC1892	The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages. G. Vaudreuil. January 1996.
RFC1711	Classifications in E-mail Routing. J. Houttuin. October 1994.
RFC1700/STD0002	ASSIGNED NUMBERS. J. Reynolds,J. Postel. October 1994.
RFC1321	The MD5 Message-Digest Algorithm. R. Rivest. April 1992.
RFC1123	Requirements for Internet Hosts -- Application and Support R. Braden, Editor. October 1989.
RFC0822/STD0011	Standard for the format of ARPA Internet text messages. D.Crocker. Aug-13-1982.

Security

Supported Standards	
RFC2831	Using Digest Authentication as a SASL Mechanism. P. Leach, C. Newman. May 2000.
RFC2595	Using TLS with IMAP, POP3 and ACAP. C. Newman. June 1999.
RFC2585	Internet X.509 Public Key Infrastructure. Operational Protocols: FTP and HTTP R. Housley, P. Hoffman. May 1999.
RFC2459	Internet X.509 Public Key Infrastructure. Certificate and CRL Profile. R. Housley, W. Ford, W. Polk, D. Solo. January 1999.
RFC2246	The TLS Protocol. Version 1.0 T. Dierks. C. Allen, January 1999.
RFC2222	Simple Authentication and Security Layer. J. Myers. October 1997.
RFC2195	IMAP/POP AUTHorize extension for Simple Challenge/Response. J. Klensin & others. September 1997.
RFC2104	HMAC: Keyed-Hashing for Message Authentication. H. Krawczyk, M.Bellare, R.Canetti. February 1997.
RFC1731	IMAP4 Authentication Mechanisms. J. Myers. December 1994.

International

Supported Standards	
RFC2279	UTF-8, a transformation format of ISO 10646. F. Yergeau. January 1998.
RFC2278	IANA Charset Registration Procedures. N. Freed, J. Postel. January 1998.
RFC2184	MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations. N. Freed, K. Moore. August 1997.
RFC1642	UTF-7 A Mail-Safe Transformation Format of Unicode. D. Goldsmith. M. Davis. July 1994.
RFC1522	Korean Character Encoding for Internet Messages. U. Choi. K. Chon,H. Park. December 1993.
RFC1489	Registration of a Cyrillic Character Set. A. Chernov. July 1993.
RFC1468	Japanese Character Encoding for Internet Messages. J. Murai, M. Crispin, E. van der Poel. June 1993.

SMTP

RFC2920/ RFC2197	SMTP Service Extension for Command Pipelining. N. Freed. September 2000.
Supported Standards	
RFC2645	ON-DEMAND MAIL RELAY (ODMR) SMTP with Dynamic IP Addresses R. Gellens. August 1999.
RFC2554	SMTP Service Extension for Authentication. J. Myers. March 1999.
RFC2505	Anti-Spam Recommendations for SMTP MTAs. G. Lindberg. February 1999.
RFC2487	SMTP Service Extension for Secure SMTP over TLS. P. Hoffman. January 1999.
RFC1985	SMTP Service Extension for Remote Message Queue Starting. J. De Winter. August 1996.

RFC1891	SMTP Service Extension for Delivery Status Notifications. K. Moore. January 1996.
RFC1870	SMTP Service Extension for Message Size Declaration. J. Klensin, N. Freed, & K. Moore. November 1995.
RFC1869	SMTP Service Extensions. J. Klensin, N. Freed, M. Rose, E. Stefferud & D. Crocker. November 1995.
RFC1652	SMTP Service Extension for 8bit-MIMEtransport. J. Klensin, N. Freed, M. Rose, E. Stefferud & D. Crocker. July 1994.
RFC0974	Mail routing and the domain system. C. Partridge. Jan-01-1986.
RFC0821/STD0010	Simple Mail Transfer Protocol. J. Postel. Aug-01-1982
Additional Features	
Relay Restrictions	
IP-based Blacklisting	
RBL-based blacklisting	
Return-Path Verification	
Message format correction	
Multi-channel delivery	
Automated Wake-ups	

IMAP

Supported Standards	
RFC2971	IMAP4 ID extension. T. Showalter, October 2000
RFC2683	IMAP4 Implementation Recommendations. B. Leiba, September 1999
RFC2595	Using TLS with IMAP, POP3 and ACAP. C. Newman. June 1999.
RFC2359	IMAP4 UIDPLUS extension. J. Myers, June 1998
RFC2342	IMAP4 Namespace. M. Gahrns, C. Newman, May 1998
RFC2221	IMAP4 Login Referrals. M. Gahrns, October 1997

RFC2192	IMAP URL Scheme. C. Newman. September 1997.
RFC2180	IMAP4 Multi-Accessed Mailbox Practice. M. Gahrns. July 1997.
RFC2177	IMAP4 IDLE command. B. Leiba. June 1997.
RFC2088	IMAP4 non-synchronizing literals. J. Myers. January 1997.
RFC2087	IMAP4 QUOTA extension. J. Myers, January 1997.
RFC2086	IMAP4 ACL extension. J. Myers, January 1997
RFC2060	INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. M. Crispin. December 1996.
RFC1731	IMAP4 Authentication Mechanisms. J. Myers. December 1994.
RFC1730	INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4. M. Crispin. December 1994.
Additional Features	
Controlled access to external (shared) mailboxes.	
Public shared mailboxes.	
Support for various Mailbox formats.	

POP

Supported Standards	
RFC2595	Using TLS with IMAP, POP3 and ACAP. C. Newman. June 1999.
RFC2449	POP3 Extension Mechanism R. Gellens, C. Newman, L. Lundblade. November 1998.
RFC2384	POP URL Scheme R. Gellens. August 1998.
RFC1939 / RFC1725 / STD0053	Post Office Protocol - Version 3. J. Myers & M. Rose. May 1996.
RFC1734	POP3 AUTHentication command. J. Myers. December 1994.

Additional Features	
XTND XMIT	Submitting messages via the POP protocol
LAST command	
Multi-Mailboxes	Accessing all Account Mailboxes using any POP mailer
Multi-Mailboxes	Accessing Public and Shared Mailboxes using any POP mailer
Support for various Mailbox formats.	
RPOP Module	
Polling	Automated Mail Retrieving from Remote Accounts
Unified Domain-Wide Accounts	Automated Mail Distribution of Retrieved Mail

HTTP

Supported Standards	
RFC2818	HTTP Over TLS E. Rescorla. May 2000.
RFC2388	Returning Values from Forms: multipart/form-data L. Masinter. August 1998.
RFC2068	Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee. January 1997.
Additional Features	
Page Caching	Internal Cache for Web Pages and Forms

FTP

Supported Standards	
RFC2389	Feature negotiation mechanism for the File Transfer Protocol P. Hethmon, R. Elz. August 1998.
RFC2228	FTP Security Extensions M. Horowitz, S. Lunt. October 1997.
RFC959	FILE TRANSFER PROTOCOL (FTP) J. Postel, J. Reynolds. October 1985.

LDAP

Supported Standards	
RFC2830	LDAPv3: Extension for Transport Layer Security J. Hodges, D. Byrne, B. Blakley, P. Behera. May 2000.
RFC2820	Access Control Requirements for LDAP E. Stokes, R. Morgan, M. Wahl. May 2000.
RFC2256	A Summary of the X.500(96) User Schema for use with LDAPv3. M. Wahl. December 1997.
RFC2255	The LDAP URL Format. T.Howes, M.Smith. December 1997.
RFC2254	The String Representation of LDAP Search Filters. T. Howes. December 1997.
RFC2252	Lightweight Directory Access Protocol (v3). Attribute Syntax Declarations. M. Wahl, T.Howes, S.Kille. December 1997.
RFC2251	Lightweight Directory Access Protocol (v3). M. Wahl, T.Howes, S.Kille. December 1997.

Mailing Lists

Supported Standards	
RFC2919	List-Id: A Structured Field and Namespace for the Identification of Mailing Lists. R. Chandhok, G. Wenger. March 2001
RFC1153	Digest Message Format. F. Wancho. April 1990.
Additional Features	
Index generator	
Subscription Confirmation	
Automated Bounce Processing	

ACAP

Supported Standards	
RFC2595	Using TLS with IMAP, POP3 and ACAP. C. Newman. June 1999.

RFC2244	Application Configuration Access Protocol. C. Newman, J. Myers. November 1997.
-------------------------	---

DNR

Supported Standards	
RFC1035	Domain names - implementation and specification. P.V.Mockapetris. Nov-01-1987.

SNMP

Supported Standards	
RFC1907	Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 Working Group & others. January 1996.
RFC1906	Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 Working Group & others. January 1996.
RFC1905	Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 Working Group & others. January 1996.
RFC1904	Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 Working Group & others. January 1996.
RFC1903	Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 Working Group & others. January 1996.
RFC1902	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2). SNMPv2 Working Group & others. January 1996.
RFC1212	Concise MIB Definitions. Rose, M., and K. McCloghrie. March 1991

WebUser Interface

Supported Standards	
RFC2646	The Text/Plain Format Parameter. R. Gellens. August 1999.
RFC2557	MIME Encapsulation of Aggregate Documents, such as HTML (MHTML). J. Palme, A. Hopmann, N. Shelness. March 1999.



Appendix B:

CommuniGate Pro History

4.0.5 10-Jan-03

- Rules, PIPE: the addresses added to the command line with the [RETPATH] and [RCPT] headers are enclosed in double quotes now.
- WebUser: WML-related sections were added to .wssp files and wssp processors.
- Bug Fix: Rules: 4.0.4: the charset verification utility could enter an infinite loop.

4.0.4 08-Jan-03

- HTTP: CGI: the HTTPS environment variable (with the value "on") is added if the connection is made via the SSL/TLS protocol.
- Directory: RDNs containing the quote mark and comma symbols are supported in Local Units now.
- WSSP: the HTMLUTF8 : prefix is implemented.
- WebAdmin: now domain administrators can login using non-qualified aliases.
- MAPI: MAPI Connection v1.0.43 is released.
- Bug Fix: HTTP: CGI: if the port number was not specified in the URL coming via an HTTPS connection, the SERVER_PORT envr-element was set to "80" (instead of "443").
- Bug Fix: Skins: the "stock" Rules.wssp page did not contain the value "9" in the Rule Priority menu.
- Bug Fix: LIST: 4.0b-4.0.3: moderated subscriptions did not work (because the X-LIST-Report: header was added to pending requests).
- Bug Fix: Admin: 4.0.3: the Reject Queued Message function could crash the server if the message was enqueued into several queues.
- Bug Fix: WebUser: 4.0b2-4.0.3: viewing a letter with a zero-length text/plain part could crash the server.
- Bug Fix: WebUser: clicking Update on the Rules.wssp page removed rules with non-ASCII names if the page was not displayed in the UTF-8 charset.
- Bug Fix: WebUser: S/MIME: importing a PKS file w/o a signature could crash the server.
- Bug Fix: External Filter: the server reported its supported Interface version as "1", while it should be "2".

4.0.3 16-Dec-02

- WebAdmin: the Reject Queued Message function is implemented.
- IMAP: the nesting depth of SEARCH expressions is now limited (to 30) to avoid stack overflows

when processing extremely complex SEARCH expressions.

- IMAP: mailbox access mode for the FETCH operations has changed to avoid resource locking when an IMAP connection is broken.
- Bug Fix: WebAdmin: 4.0.2: the Domain List limit and Object List limit settings always had the same value.
- Bug Fix: WebAdmin: 4.0-4.0.1: the Domain Admin entrance page displayed only one "Other Domain".
- Bug Fix: CLI: 4.0-4.0.2: the GETSUBSCRIBERINFO command did not return the "mode" data.
- Bug Fix: SECURITY: 4.0b-4.0.2: WebUser Interface: some file-path check routines incorrectly checked for the ".." path elements thus providing access to files outside the target directory.

4.0.2 26-Nov-02

- Security: the MSN SASL method is implemented.
- WebAdmin: the Admin Preferences for the Domain List page are implemented.
- WebUser: RC2-CBC is now used as the default S/MIME encryption cipher (to support older versions of Netscape).
- Charsets: ISO-8859-10 and ISO-8859-15 charsets are supported now.
- MAPI: the basic "off-line mode" is implemented.
- Bug Fix: WebUser: 4.0-4.0.1: the webSiteEnabled session element was not set if the WebSite quota was set to "unlimited".
- Bug Fix: WebAdmin: 4.0b4-4.0.1: the Refresh Rate for WebUser Sessions monitor could not be set via WebAdmin.
- Bug Fix: WebUser: 4.0b8-4.0.1: embedded images were not displayed in the HTML texts inside multipart/related messages.
- Bug Fix: CLUSTER: specifying '*' as the forwarding host (to relay mail via frontends) did not always work (the port number was not set).
- Bug Fix: Log: 1.0-4.0.1: certain "unusual" data elements could cause the Log engine crashes.
- Bug Fix: WebUser: 3.0-4.0.1: the "message open" procedure could cause deadlocks.

4.0.106-Nov-02

- Foundation: condition locks are re-implemented for all platforms. Old implementation could cause excessive thread waiting times on an overloaded system.
- Protection: DNR error codes can be processed with the Blacklisting by DNS option (making it possible to blacklist all hosts that do not have reverse DNS records).
- EventHandlers: the Frequency parameters have been added.
- EventHandlers: now events are recorded in the CommuniGate Log.
- Skins: the DeliveryReport code and the .wssp file have been added.
- Skins: the Notify when Read option has been added to the Compose code and .wssp file.
- Mailbox Manager: the RENAME operation now automatically creates all "outer" mailbox folders needed to create the target mailbox.
- MAPI: many changes and bug fixes (see the MAPI status page).
- LIST: mail header composing routines have been changed.
- CLUSTER: now the cluster-wide RFC822 Filters are not applied when a message is transferred

between cluster members for local delivery.

- WebAdmin: the internal Domain Administration realm has been changed to "/DomainAdmin/" to fix various URL access problems.
- Bug Fix: S/MIME: WebUser Interface refused to send an encrypted message to a recipient with a longer public key.
- Bug Fix: SNMP: HTTPAdmin module wrote its stat information into the HTTPUser data structures.
- Bug Fix: SNMP: 64-bit COUNTER data in responses had the APP 1 type instead of the APP 6 type.
- Bug Fix: RULES: the Remember From action could crash the server if no parameter was specified.
- Bug Fix: MAILBOXES: in some rare situations concurrent access to a mailbox could crash the server.
- Bug Fix: DEQUEUEUR: if a message had a pending request to generate a "delayed warning" report, and the delayed address failed/succeeded at the same time, the server could crash.
- Bug Fix: Rules: ROUTE conditions could crash the server if an incoming message was rejected because it had "too many hops".

4.0 18-Oct-02

- CLI: the GETSUBSCRIBERINFO command is implemented.
- LDAP: binary data are supported now.
- LDAP: if the "Substitute with uid in conditions" option is enabled, the "equals to" search operation ignores the "@domain" part of the search string.
- Directory: Remote Units: binary data are supported now.
- Directory Integration: now the userCertificate attribute is automatically stored in the Directory (together with the RealName/cn attribute).
- CLUSTER: extremely large responses that cannot be read by a requesting cluster member are replaced with an error code on the serving member.
- TLS: a workaround for programs that incorrectly send TLS data in the block-cipher mode (such as Exim MTA) is implemented.
- EXTFILTER: the version 2 API is implemented: the ADDHEADER response code is supported now.
- WebMail: Secure MIME certificate and key page has been reimplemented and documented.
- WebAdmin: the Show Aliases option is implemented.
- WebSites: now the default.html file is retrieved for all types of "directory links" (<http://server/~user/dir1/dir2/>).
- WebSkins: several new SESSION dataset elements have been added.
- WebSkins: the EmptyTrash HTTP parameter has been renamed into EmptyTrashNow; now it is processed for all session requests.
- WebSkins: default charset processing has been changed.
- WebSkins: .wssp pages for stateless requests can (and should) use the %%filesRef%% variable to form file reference URLs.
- WebUser: the "use Letter charset" option is implemented.
- WebUser: the Store Attachments (Files) function is implemented (the message.wssp page has been extended).
- Kernel: Ukrainian KOI8-U charset is supported now.
- Statistics: the "Dynamic Cluster requests" SNMP elements have been added.
- WebAdmin: Directory: the Browser can display non-ASCII data now.

- WebAdmin: "disabled" RPOP periods are supported now.
- CLI: the LISTADMINDOMAINS command is implemented.
- MAPI: support for Outlook Rules has been added.
- Bug Fix: LIST: 4.0b9: digest generator did not calculate the collected message size correctly, and could include too few messages in each digest.
- Bug Fix: Domains: 4.0b7-b9: the Index.data file could fail processing the "removed" (tagged with "-") records.
- Bug Fix: Skins: 4.0b9 "message part" application codes incorrectly processed charsets.
- Bug Fix: CLUSTER: the CREATEWEBUSERSESSION, GETWEBUSERSESSION, KILLWEBUSERSESSION CLI commands could fail in a cluster.
- Bug Fix: External Program parameters were parsed with a wrong parser (adjacent quoted strings were catenated).
- Bug Fix: S/MIME: signatures did not include all attributes required by some mailers.
- Bug Fix: S/MIME: some signatures could not be verified because of an incorrect buffering technique used.
- Bug Fix: S/MIME: signed and encrypted messages larger than 4K were composed incorrectly.
- Bug Fix: images downloaded from Personal WebSites via WebAdmin Interface could be damaged.
- Bug Fix: MAPI: Windows spooler could crash if the CommuniGate MAPI Connector was used.
- Bug Fix: MAPI: embedded/nested Calendaring messages were processed incorrectly.
- Bug Fix: MAPI: some composed headers were not MIME-encoded.
- Bug Fix: MAPI: drag-and-drop text-type attachments did not work correctly.

4.0b7 22-Aug-02

- New Platform: HPUX/Itanium version is released.
- RULES: "string lists" are implemented. The Auto-Reply Rule has been modified to send only one reply to each sender.
- SMTP: the Recipients/Message setting has been implemented.
- CLUSTER: SMTP: the "*" forward to option (meaning "to all frontends") is supported now.
- Directory: the DN's of found records are now checked against the Access Rights. If the RDN is not allowed to be read, the record is not returned to the client.
- Directory: to comply with RFC2256, the cn and dc attributes have been moved from the organization objectClass to the communiGateDomain objectClass.
- CLI: the SETPOSTINGMODE command is implemented.
- LOCAL: now distribution to all@domain is prohibited for the messages with an empty return-path (to stop the "bounce mail distribution" attacks).
- Admin: the MAPI Connections setting has been added to the Limits panel on the Domain Settings page.
- WebAdmin: now the Server Up-Time is displayed on the Settings -> General page.
- MAPI: if a newly created message does not have a plain-text part, the RTF part is used (these RTF-only messages were sent as empty messages before).
- MAPI: if a message has an HTML part, other parts are "disabled" (these messages were copied as empty messages before).
- MAPI: non-system locales are supported now.
- MIME: strings: the '\' symbol processing has been improved.

- Skins: the default Skin HTML code has been "cleaned".
- RPOP: internal scheduling mechanisms have been re-designed.
- Platform: the MacOS X package now includes the Uninstall.sh script
- Bug Fix: MAPI: there could be a deadlock in the spooler, causing Outlook to hang when the Send button was pressed.
- Bug Fix: MAPI: ITable interface bugs could crash Outlook.
- Bug Fix: DOMAINS: the Index.data files have been processed incorrectly if there were account names starting with the "-" sign.

4.0b6 30-Jul-02

- LDAP: SASL authentication for mail=accountName BindDN is implemented.
- Charsets: data encoded using Korean charsets can be converted to and from the UTF-8 charset.
- WSSP: the DATE:, DATETIME:, LOCALDATETIME:, DATETIMESHORT:, LOCALDATETIMESHORT: prefixes are implemented.
- WebApp: the "mailbox" processing code has been changed, so the Sent and Received column values are now returned as "date"-type elements.
- WebApp: the "mailboxes" processing code now adds the "parent" element to the mailbox elements.
- WebAdmin: now administrators without the CanCreateGroup access right cannot modify any group data.
- WebAdmin: GROUP: several member addresses can be entered in one field now.
- WebMail: now moving a message between "own" mailboxes (including moving a message to the Trash) temporarily disables storage quota checks.
- HTTP: CGI: the SCRIPT_NAME parameter does not contain the "additional path" data now.
- HTTP: CGI: now an external program receives the PATH_INFO and REQUEST_URI parameters.
- IMAP: RFC3348 (CHILDREN extensions) is implemented.
- MAPI: the version-checking schema has been changed.
- MAPI: ReadReceipts are working now.
- MAPI: now messages are always sent via the spooler.
- MAPI: Win95 systems are supported now.
- MAPI: Unseen message number calculation is improved.
- MAPI: non-default text encodings are supported now.
- MAPI: X-Mailer header is inserted into the composed messages.
- MAPI: "simple" calendar requests are stored and transferred in the Outlook-compatible form.
- MAPI: now the EXPUNGE command is sent to the selected mailbox.
- Platform: the "application directory" for Linux systems has been changed from /usr/local/sbin to /opt
- Foundation: the cryptography routines have been optimized to increase SSL/TLS performance.
- Bug Fix: 4.0b: CLUSTER: failure to open an account on a backend server could crash the frontend server.
- Bug Fix: the "copyMessages" Mailbox Manager routine ignored errors when it was copying large (over 1MB) messages followed by a smaller message.
- Bug Fix: the "oldestMessage" info was not updated when a new message was added to a mailbox.
- Bug Fix: SMTP: 4.0b5 could not open the Listener WebAdmin page.

- Bug Fix: WebUser: the text/enriched converter did not replace single EOL with a white space symbol.
- Bug Fix: MAPI: the "Remember password" setup option did not work correctly.
- Bug Fix: MAPI: the Date: field is now composed correctly, using the current daylight saving time settings.
- Bug Fix: MAPI: generated messages could contain empty bodies.
- Bug Fix: MAPI: RTF EOL processing has been fixed.
- Bug Fix: MAPI: iCalendar data now contains the UID attribute.
- Bug Fix: MAPI: in some situations the Inbox mailbox was displayed twice.

4.0b5 08-Jul-02

- Events are implemented (see the SysAdmin->Events section for more details)
- HTTP: the module has been divided into two modules (HTTP Admin and HTTP User).
- Admin/Directory Integration: the Public Info Custom Settings are implemented.
- Admin: the MaxMailboxes Account Setting is implemented.
- WebAdmin: the Threads Monitor has been implemented.
- WebAdmin: the Access Monitor pages have been re-implemented. LDAP, FTP, ACAP, PWD monitors are implemented.
- SMTP: the actual address of the blacklisted host is now included into the negative (591) SMTP server response.
- SMTP: now the Send Secured and Hold Mail names can include the wildcard (*) symbols.
- SNMP: the "AUTH" group of statistic elements has been implemented.
- WebAdmin: name filtering is implemented on the SNMP Element List Monitor page.
- WebUser: HTTP responses for .wspp requests now have explicitly specified charsets in the Content-Type headers.
- WebAdmin: HTTP responses now have explicitly specified charsets in the Content-Type headers.
- WSSP: now the Auto-signup component code can put Custom Attribute values into the new Account settings.
- FTP: LIST and NLST command parameters (strings starting with the "-" signs) are skipped.
- FTP: the CWD "/" command is allowed now.
- CLUSTER: the account info is now displayed on Account list and Account Settings WebAdmin pages of all Cluster members.
- MAPI: processing has been changed to utilize only one IMAP connection per client.
- MAPI: new Read Receipts (MSN) are generated and sent.
- MAPI: now the Setup program has the Install/Upgrade/Uninstall options.
- MAPI: the Outlook Setup mode (Corporate/Internet) is detected now and the user is warned if the Corporate mode is switched off.
- MAPI: Address Book: apostrophes are removed from recipient names
- MAPI: now the Contacts info is stored in the standard vCard format, too
- MAPI: the X-MAPI-Message-Class header is now added to "special-class" message objects.
- MAPI: Free-Busy information processing has been changed to support shared Accounts.
- Bug Fix: EXTERNALFILTER: in 4.0b4: the INTF command replies were not processed correctly.
- Bug Fix: the MIME parser could return negative part size values for malformed MIME messages.
- Bug Fix: MAPI: message header encoding has been fixed.

- Bug Fix: MAPI: quoted-printable message body encoding did not work correctly;

4.0b4 18-Jun-02

- Foundation: Windows: OS privileges needed to enable external program termination are enabled explicitly now.
- External Filtering: the API has been extended (the INTF command added, the DISCARD response is supported now).
- Mailbox: the "oldest message" info is now stored in the mailbox meta-data dictionary.
- MAPI: several minor bugs are fixed.
- Admin: the Rename In Folder option is implemented. The "renameInPlace" attribute has been added to the Directory Schema.
- WebAdmin: Login page for Domain WebAdmin Interfaces now lists "Other Domains" that this Domain Administrators can open.
- Bug Fix: 4.0b3 could crash when checking mailbox access rights.

4.0b3 16-Jun-02

- The Linux/Itanium version is released.
- MAPI Connector is released.
- Admin: the MAPI and TLS Account and Domain "Enabled Service" options are implemented.
- CLUSTER: Virtual Mailbox manager did not read the UID information for appended messages. As a result the IMAP APPEND and COPY operations applied to virtual mailboxes did not return the extended UIDPLUS codes.
- Security: the WebAdmin "Security" page now accepts multi-certificate CA Chains.
- Security: the Advertise NTLM SASL Method option has been added to the Obscure Settings page.
- SMTP: the Advertise NTLM AUTH option has been removed from the SMTP module settings.
- Rules: now the Subject and From Name fields are MIME-decoded and converted into UTF-8 before processing.
- WebAdmin: the Server-Wide and Cluster-Wide "default WebAdmin" pages are supported now.
- LIST: now the ^I macro can be used in the Hello and GoodBye messages.
- ALERTS: Account-level Alerts have been implemented.
- CLI: the GETACCOUNTALERTS, SETACCOUNTALERTS, POSTACCOUNTALERT, and REMOVEACCOUNTALERT commands have been implemented.
- CLI: the GETSNMPELEMENT command has been implemented.
- WebUser: now Named Skins can be used in Stateless requests (such as login.wssp requests).
- WebMail: the GBK charset is supported now.
- Bug Fix: CLUSTER: SMTP: successful client authentication did not reset the "blacklisted IP" flag.
- Bug Fix: MAILBOX: the "Recent" flag was not always processed correctly in the concurrent access environments.
- Bug Fix: the Big5->UTF-8 conversion routine did not process all Big5 symbols correctly.

4.0b2 02-Jun-02

- CLUSTER: the GetAccountInfo CLI command is "clusterized" now.
- MIME: search for body strings now works for non-ASCII strings specified in various charsets.
- Security: the Advertise secure SASL Methods option has been added to the Obscure Settings page.
- Directory: new Unit names are now checked for "bad symbols".
- IMAP: now the GETQUOTA/GETQUOTAROOT commands can be used to retrieve mailbox store information for foreign mailboxes.
- Protection: now "Unblacklisted" (White Hole) addresses can be specified using their DNS Names.
- PIPE: now the module does not try to "fix" multi-line From: and Sender: header fields in submitted messages.
- Security: the External Authentication API has been changed (see the Security section for more details).
- Domains: now the Index.data file is kept in sync with the Domain Accounts (rather than created on server shutdown).
- Domains: now the Index.data file can be stored in the "Index" subdirectory (to simplify symlink processing).
- Foundation: Tru64: the system mode allowing 65000+ file descriptors is enabled now.
- Bug Fix: IMAP: the QUOTA STORAGE responses now specify sizes in 1K units.
- Bug Fix: IMAP: UID-based non-range messagesets were formed incorrectly if a message with the specified UID did not exist.
- Bug Fix: FTP: the SIZE command could crash the server.
- Bug Fix: FTP: the QUIT command was not process correctly after an unsuccessful login attempt.

4.0b1 02-May-02

- FTP: the FTP module is implemented. See the Access->FTP section of the manual for details.
- WebMail: the WML requests are supported now.
- MAILBOX: the information about the number of Unseen messages is cached now.
- WebUser: the information about the number of Unseen messages in now available on the mailboxes.wssp page.
- MAILBOX/IMAP: additional extensions are implemented to simplify communication with the MAPI Connector.
- CLUSTER: now users with the Require Secure Login setting enabled can login using secure connections to frontends.
- RULES: the ^S and ^F macros are now implemented as ^S/^s and ^F/^f macros.
- Router: update procedue has been modified to avoid crashes during Router Table updates under heavy load.
- LOCAL: the Always Add the Envelope Addresses Field option is implemented.
- CLI: the RELEASESMTPQUEUE command is implemented.
- CLUSTER: SMTP: ETRN commands and Wake-up E-mails release queues on all Dynamic Cluster members.
- HTTP: the OPTIONS method is implemented.
- WebUser: the text/enriched format is supported now.
- RPOP: the secure (TLS) option is implemented. The WebSkin, WebAdmin, WebUser RPOP files are updated to accomodate the new option.
- Directory: LDAP provisioning now detects the unixPassword attribute and stores its value as a

3.5.9 05-Apr-02

- POP: the '#' symbol processing has been changed to allow re-routing of 'user#domain' account names into 'user@domain' names.
- MAILBOXES: MailDir mailboxes under Windows are now opened in the Shared mode, allowing several clients to read the same message at the same time w/o the --SharedFiles option.
- Bug Fix: MAILBOXES: deleting messages from MailDir mailboxes used by several clients could crash the server.
- Bug Fix: PIPE: the 3.5.8 version incorrectly enqueued messages directed to the PIPE module.
- Bug Fix: on some platforms timezone switches (such as daylight saving time switches) have not been detected correctly.

3.5.8 23-Mar-02

- LIST: now the administrative message subjects can be specified in the List Settings. The Stock Skin listsettings.wssp file has been modified to include these new settings.
- LIST: if a posted message does not have the charset specified, the Feed mode header and trailer are added to the distributed message in the List Preferred charset.
- LIST: newly created lists now have the same preferred charset as the effective WebUser preferred charset for the list owner account.
- Bug Fix: CLUSTER: the HTTP Keep-Alive option enabled on backend servers could cause user disconnects.
- Bug Fix: CLUSTER: SSL connection proxying could pass incorrect data to the server, causing (mostly) IMAP APPEND errors.
- Bug Fix: IMAP: under certain OS'es (AIX) the fetch command could return "syntax error" when it was issued for an empty mailbox.
- Bug Fix: message batch enqueueing routines could cause crashes during the server shutdown procedure.

3.5.7 11-Mar-02

- Protection: the Blacklisting by IP Domain Name feature has been implemented.
- CLI: the LISTSUBSCRIBERS command now needs the FILTER keyword.
- CLUSTER: now cluster members do not disconnect from the backup controller even if the failover transition takes more than 3 minutes.
- CLUSTER: now the LIST and LISTSUBSCRIBERS commands can be used on any cluster member and they can deal with mailing lists in shared domains.
- MIME: MIME header decoder now removes unencoded white spaces between encoded atoms.
- HTTP: optional support for the Keep-Alive method is implemented.
- HTTP: a workaround for Microsoft Internet Explorer browser bug is implemented.
- LIST: now if the posting policy is set the "Moderate all", non-subscribers can submit their posts, too.

- MAILBOXES: now the TextMailbox scanner can scan several large mailboxes at the same time.
- Bug Fix: 3.5.6: MIME header encoding module did not encode spaces separating encoded atoms.
- Bug Fix: 3.5.3-3.5.6: the "host queue splitting" operation did not properly release error message object and could cause memory leaks.

3.5.6 18-Feb-02

- Bug Fix: 3.5.5: mailbox size check routine could crash the server if a message was being added to an empty mailbox in the "parsed" state.

3.5.5 14-Feb-02

- LIST: settings are stored in the UTF-8 charset now.
- CLUSTER: message object attributes ("trusted source", "auth-ed") are now sent to backends.
- SMTP: low-level logging for incoming connections is improved.
- Bug Fix: WebAdmin: the Server-wide defaults were displayed instead of Cluster-wide defaults for the shared Domain WebUser Prefs defaults.
- Bug Fix: Directory: if no Unit existed, the Browser could crash the server.
- Bug Fix: Skins: skin file uploading can cause crashes on heavily loaded systems.
- Bug Fix: WebUser: the "HTML message cleanup" module could enter a loop blocking a WebUser session and its open mailbox.

Note: In a dynamic cluster, each frontend must be restarted (during or after its upgrade) AFTER all backends are upgraded.

3.5.4 03-Feb-02

- HELPERS: time-out and auto-restart settings are implemented, the HELPER object internals have been re-written.
- Skins: file upload algorithms has been changed, now CLI commands clear caches.
- CLI: the VERIFYACCOUNTPASSWORD command is implemented.
- LIST: if a list subscriber is a local account, the account password can be used to browse "subscriber-only" list archives.
- WebAdmin: the Admin Prefs now can specify the default limit for the number of List Subscribers to display.
- SMTP: reversing the channel after ATRN is sent now works even when the connection is secured using TLS/SSL.
- SMTP: ATRN 4xx response codes have been changed to the values specified in RFC2645.
- Bug Fix: SMTP: receiving ATRN over TLS could crash the server.

3.5.3 17-Jan-02

- New Platform release: BeOS/PowerPC.
- CLUSTER: algorithms have been improved to better handle situations when an account create/rename/remove operation is being executed at the same time when the account list is being built. For domains with more than 200,000 users these situations could block access to hash tables for several minutes, causing cluster break-ups.
- LOCAL: distribution rights for the all@ addresses are now checked before the account list is being built.

- LIST: the "listserver" address processing has been changed. The new processing method does work in the Dynamic Cluster environment, too.<http://mail.stalker.com/Session/162-pQpgs0tIm63GfmZoVVtM/Mailbox.wssp?Mailbox=INBOX&MSG=29468&Delete=&>
- SMTP: the Relay to Client IP Addresses = simple option processing has been changed to check the original addresses, not the resulting, re-routed addresses.
- SNMP: WebAdmin pages now display 64-bit numeric values.
- Bug Fix: LIST: the "cleanup procedure" closed list owner accounts on the Cluster Controller, allowing them to migrate to other cluster members and lose connections with their lists.

3.5.2 05-Jan-02

- Bug Fix: IMAP: 3.5b-3.5.1: the CAPABILITY response did not have a space before the first AUTH= parameter.
- Bug Fix: Account: when a mailbox without submailboxes was removed using 3.5x versions, the mailbox size was not subtracted from the account total message storage size.
- Bug Fix: McAfee License Limit Expiration was not always calculated correctly.
- Big Fix: LIST: subscription via the "listserver" address could fail because of incorrect confirmation string processing.
- Bug Fix: WebUser: the last symbol of the "From" address was not stored if the MIME Headers option was switched off.
- Bug Fix: IMAP: 3.5x versions did not support the UID EXPUNGE command.
- Bug Fix: CLI: 3.5x versions could return improperly terminated strings in GetAccountLocation responses.

Bug Fix: Directory-Based Domains did not "see" their domain Skins after a server restart.

3.5.1 18-Dec-01

- WebUser: the Expire header with negative date (added in the 3.5 version) has been removed, since it caused problems for old Netscape browsers.
- Lists: the Dynamic Cluster Controller now tries to move the owner account to itself before creating a mailing list.
- RULES: the Vacation Rule priority is not set to 2, to make the server apply it before the Redirect All Rule.
- CLUSTER: the IMAP and ACAP backend login responses now carry the "Relay" flag.
- Bug Fix: 3.5: the Directory Integration option "generate mail attribute" did not work on non-Cluster systems.
- Bug Fix: WebUser: the Auto-wrap algorithm for "flowed" texts could cause crashes on some platforms.
- Bug Fix: HTTP: the Redirect operation placed "http://" into the Location header even if the current connection was an https one.
- Bug Fix: MacOS X (Darwin) package had a syntax bug in the "post-install" script.
- Bug Fix: Viewing the "Queue" Monitor page could cause Queue deadlock.

3.5 11-Dec-01

- POP: access to empty INBOX mailboxes has been optimized.
- PIPE: Foreign Queue processing is implemented.
- PIPE: [STDERR], [FILE], [RCPT], and [RETPATH] tags are implemented.
- IMAP, POP: the STARTTLS/STLS option is seen in the CAPABILITY response only if the addressed Domain has the Security Certificate option enabled.
- Account Templates: the Initial Message text now can start with the [charset] prefix.
- Rules: now Reply and React texts can start with a [charset] prefix.
- WSSP: the ROUNDSIZE: prefix is implemented, the mailboxes.wssp page has been changed.
- WebUser (Skins): the directory processing has been changed, the directory.wssp and the sessiondirectory.wssp pages have been modified.
- Directory-based Domains: Skin support is implemented.
- CLUSTER: header fields added with the frontend Server Rules are now stored by backends.
- Bug Fix: 3.5b9 incorrectly processed server-wide WebUser Preferences.
- Bug Fix: 3.3-3.5b9 a rare deadlock situation (in all prior versions) could stop ENQUEUEER and DEQUEUEER processors if an administrator opened the Message Monitor page.
- Bug Fix: 3.2-3.5b9 message file stored in the MDIR mailboxes by Cluster backends could be improperly replaced with file links if the Reuse Temp Files option was enabled.
- Bug Fix: 3.5b5-b9: header composing algorithm for non-ASCII data could enter an infinite loop, blocking that thread and consuming CPU.
- Bug Fix: 3.4-3.5b9: supplementary send-phase relay checking algorithms were not the same as the input-phase algorithms, causing relay refusals in some rare situations.
- Bug Fix: 3.3-3.5b9: Directory: Local Unit: the delete record operation incorrectly checked if record children existed.
- Bug Fix: 3.4-3.5b9: CLUSTER: the POP3 "relaying enabled" flags were not processed correctly on frontends.

3.5b9 17-Nov-01

- LDAP/Admin: LDAP-based provisioning for regular domains is implemented (see the Directory Integration section of the manual).
- DNR: the "search PTR records" operation is implemented.
- SMTP: when message sending fails because the receiving host drops the connection, the message is re-enqueued (to avoid queue blocking for hosts that violate standards and just drop connections when they do not want to accept certain messages).
- SMTP: input messages exceeding the size limit are received, but they are not stored in files.
- Protection: the Banned Header and Body line settings are implemented. See the Protection section of the manual.
- Protection: the Client By Name option is implemented.
- Protection: the Unblacklistable (WhiteHole) Addresses list is implemented.
- Security: now Certificate Signing Requests can be generated and new Certificates can be set without prior removing of an existing certificate.
- CLUSTER: the cluster-wide Protection settings are implemented.
- CLI: Banned Header, Protection, and Cluster Protection commands are implemented

- CLI: GETACCOUNTSUBSCRIPTION and SETACCOUNTSUBSCRIPTION commands are implemented.
- CLI: GETMAILBOXALIASES and SETMAILBOXALIASES commands are implemented.
- WebUser: more tags and tag parameters are cleaned out from HTML message portions now.
- WebAdmin: additional Monitor Access Rights are implemented.
- DNR: additional settings are implemented.
- DNR: requests to RBL servers are sent "quickly", so if an RBL server is down incoming SMTP connections do not time-out.
- WebUser: now when Sent and Drafts mailboxes are auto-created, they are auto-subscribed to and the mailbox list is refreshed.
- Bug Fix: 3.5b6-b8 versions crashed if an incorrect "LIST operation" address was used.
- Bug Fix: 3.5b6-8: WebSkins: password recovery E-mail address was not updated if password modification was disabled.
- Bug Fix: 3.5b6-8: WebSkins: custom Message header fields were not decoded from UTF-8 when a message page was being composed.
- Bug Fix: 3.5b8: External Helpers: crashed Helper could cause Server crash if the Helper log was enabled.
- Bug Fix: 3.5b7-8: LOCAL Delivery: some error codes could be lost, so no error reports were sent back when delivery to an account failed.

3.5b8 03-Nov-01

- SMTP: the Advertise NTLM AUTH option is implemented.
- SMTP: now when the module sends the STARTTLS command, it uses the SSLv3 (rather than SSLv2) "hello" operation.
- DEQUEUEUR: more SNMP statistics elements have been implemented.
- DEQUEUEUR: minor internal algorithm changes.
- DIRECTORY INTEGRATION: Server-Wide and Cluster-Wide settings are implemented.
- CLI: the [GET|SET][CLUSTER]DIRECTORYINTEGRATION commands are implemented.
- CLI: the Skin Administration commands are implemented.
- CLI: the GETCLIENTIPS and GETBLACKLISTEDIPS commands are implemented.
- CLI: the GETWEBUSERSESSION and KILLWEBUSERSESSION commands are implemented.
- LOCAL: if a message is delayed by a Cluster backend, the entire account queue is suspended now.
- LOCAL: the SNMP statistics elements have been implemented.
- CLUSTER: slave startup procedure has been changed to avoid problems on systems with a large number of shared domains.
- SNMP: processing of "not-found" elements have been changed to match the SNMPv2 specs.
- HTTP: the "SkinFiles" realm is implemented to allow Skin File retrieval without using a WebSession URL.
- WebUser: auto-wrap algorithms have been changed.
- WebUser: the autoWrap "flowed" option has been added.
- WebUser: the charset parameter of message parts containing the ASCII-only symbols is ignored now.
- Queue: the Web Monitor modules have been changed to display not more than 1000 items in huge queues.

- IMAP: non-standard parameters of the Content-Type and Content-Disposition fields are retrieved now.
- Foundation: thread priority routines are implemented.
- Foundation: Linux: the STTask routines have been changed to avoid leaving "zombies" of killed processes.
- Foundation: BSD: the STDictionaryEnumerator routines have been changed to avoid directory-detection problems on NFS filesystems.
- Bug Fix: Foundation: in 3.5b7 the setInetAddress routine incorrectly formatted IP addresses as numeric strings. This could cause problems in specifying IP addresses on the Listener, Cluster, and SNMP WebAdmin pages.
- Bug Fix: SNMP: the value of parameters that changed their types in 3.5b7 from INTEGER to COUNTER was sent incorrectly via SNMP.

3.5b7 24-Oct-01

- SNMP: the WebAdmin Monitor interface to SNMP data is implemented.
- SNMP: the "total number of jobs"-type parameters now have the Counter data type.
- SNMP: HTTP monitoring elements have been added.
- SMTP: the Force AUTH option is implemented.
- WebAdmin: Admin account Preferences processor has been changed.
- Directory: the "search"-type operations now enter the subtrees stored on different Units.
- CLUSTER: the Cluster-Wide Rules are implemented.
- CLI: the GETCLUSTERRULES and SETCLUSTERRULES commands are implemented.
- CLUSTER: the Cluster-Wide Router Table is implemented.
- CLI: the GETCLUSTERROUTERTABLE and SETCLUSTERROUTERTABLE commands are implemented.
- CLI: the GETACCOUNTSEFFECTIVESETTINGS and GETDOMAINEFFECTIVESETTINGS commands are implemented.
- CLI: the [GET|UPDATE|SET][ACCOUNT|DOMAIN] commands have been renamed into the [GET|UPDATE|SET][ACCOUNT|DOMAIN]SETTINGS commands. Old names continue to work, too.
- WSSP: the EQUALS operation with a quoted-string argument is implemented.
- Migration: the MoveAccounts program and its parameters have been changed.
- LDAP: the modifyDN "newRDN" parameter was processed incorrectly.
- AIX: build parameters have been modified to support 6000+ threads.
- Bug Fix: WebUser: the number of selected messages for the mailbox.wssp page was calculated incorrectly.
- Bug Fix: Directory: Remote Units: the Search operation returned "unstripped" DN's when the "Server Base" setting was non-empty.

3.5b6 18-Oct-01

- Security: the login-disabling options protecting Accounts from Password Attacks are implemented.
- RPOP: when retrieving mail from Unified Domain-Wide Accounts without using Special Headers,

the module now checks that To:/Cc: addresses can be routed to a Local account (rather than just checking that they are directly addressing the Main Domain).

- CLUSTER: Cluster-wide Default Domain Settings, Default Account Settings, Alerts, and WebSkins are implemented.
- CLI: the Cluster-wide versions of the commands dealing with Default Domain and Default Account settings are implemented.
- CLI: the Alert Administration commands are implemented.
- WebUser: URLs for non-ASCII attachments are now composed in the UTF8 charset (to work with Windows Internet Explorer).
- WebUser(WebSkins): the Address Books records are sorted now.
- WSSP: the RANDELEMENT function is implemented.
- Domains, WebUser Settings: non-ASCII attributes are supported now.
- MoveIMAPMail: the --target option has been implemented.
- Bug Fix: new (Skin-based) List Archive Browser pages did not have the correct charset specified.
- Bug Fix: old (Web-User and WebAdmin) List Subscribers pages could crash the server if the list had the Require Confirmation option disabled.
- Bug Fix: Skins: the default rules.wssp and rule.wssp files did not have the closing </SELECT> tag, causing problems for Netscape browsers.
- Bug Fix: Skins: the webUserSiteIndex internal code routine misplaced the account name (was shown as an error code).

3.5b5 07-Oct-01

- SMTP: the Wake up Now button has been added to the SMTP Settings WebAdmin page.
- DNR: the new Custom setting allows an administrator to specify DNS Server addresses explicitly. See the SysAdmin section for the details.
- WebAdmin: Preferences now work for Secondary Domain Administrators, too.
- WebAdmin: the Charset parameter has been added to the Administrator Preferences.
- WebAdmin: Alerts now stored using the UTF-8 charset.
- Directory: spaces around the comma signs are removed from the DN strings
- LIST: when archive mailboxes are swapped, the newly created archive mailbox gets the ACLs of the old archive mailbox
- Mailboxes: the "Redirected" message status is implemented.
- Groups: non-ASCII "real names" are supported now.
- Accounts: non-ASCII "real names" and "custom attributes" are supported now.
- Account Templates: non-ASCII "real names", "custom attributes", and mailbox names are supported now.
- WebUser: the "Redirected" message status is set when a message is redirected or forwarded.
- WebUser: renaming and remove mailboxes when the Show Subscribed option is enabled now renames/removes the mailbox and submailboxes from the Subscription list.
- WebUser: now the Composer encodes non-ASCII attachment file names.
- Bug Fix: the Save Sent Messages mailbox could not be set in the 3.5b4 Skins interface
- Bug Fix: the 3.5b4 version incorrectly processed most non-cluster License Master Keys.
- Bug Fix: Rules: if the Reply/Reply All operation parameter was specified with the "+" sign and additional headers, the Cc: headers were not processed at all, and Bcc: headers caused parsing

errors.

- Bug Fix: Mailbox renaming for non-top level mailboxes did not work with the "rename Submailboxes" option.

3.5b4 04-Oct-01

- WebSkins are implemented. Now all domains that do not have files in the Account subfolder of the old custom WebUser Interface directory use the new Skins Interface.
- LIST: additional subscriber address checks are implemented to avoid self-subscribing of the special mailing list addresses (-on, -subscribe, -off, etc.)
- CLI: the SetAccountPassword command is implemented.
- SMTP: the "send encrypted wherever possible" option is implemented.
- RBL: RBL responses in the 127.1.x.x range are now recognized as "blacklist it" responses.
- Migration: the utility to simplify migration from the Post.Office product has been implemented.
- WebAdmin: the /MainAdmin/ realm is implemented (see the HTTP section of the manual).
- Directory: the CACHain attribute has been added to the default Schema.
- LDAP: the Compare operation is implemented.
- ROUTER: the parser has been changed to allow special symbols in the left part of the alias records (records like <FAX=*> = * can be used now).
- ROUTER: now account alias records can be used for the Central Directory-based Routing.
- Bug Fix: WebUser/WebAdmin: in 3.4b3 version custom file uploading could fail in regular (non-directory-based) Domains.
- Bug Fix: In 3.5b2-3 the "Insert All" Directory Integration operation could crash the server if the Domain had at least one Group.
- Bug Fix: CLUSTER: renaming a shared domain did not rename Domain Aliases on all cluster members.

3.5b3 25-Jul-01

- Many internal file-handling routines have been changed.
- WebAdmin: Log settings can now be changed only if the Administrator has the CanModifySettings Server Access rights.
- RPOP: now the Leave On Server option works with any types of UIDs remote servers present and the list of retrieved UIDs is preserved between Server restarts.
- RPOP: the APOP option is added to individual RPOP records.
- Local: the X-Special-Delivery: test header field now works with the Local Delivery module, too: messages with that field are not stored in the Account mailboxes.
- Forwarders and Groups: processing algorithms have been changed.
- Directory-Based Domains: Groups are supported now.
- Directory: the mailListSettings and groupOptions attributes have been added to the default Directory Schema.
- Directory-Based Domains: Custom WebUser and WebAdmin Interfaces are supported now.
- Directory: the fileData attribute and the CommuniGateWebInterface objectClass have been added to the default Directory Schema.

- Directory: the mailListSettings and groupOptions attributes have been added to the Directory Schema.
- SMTP: a workaround for sending mail to buggy firewall relays is implemented.
- List: the "Hide From Address" option is implemented.
- PWD: when a user with the Server Settings access right logs in, the max input buffer size is increased to 1MB (to allow for larger data in the Router operations).
- Bug Fix: WebUser: non-standard MIME content types and subtypes could be returned corrupted.

3.5b2 15-Jul-01

- DEQUEUEUR: algorithms have been changed to avoid crashes when multiple DEQUEUEUR threads process the same message.
- IMAP: the FETCH algorithms have been changed to avoid loading large messages into memory.
- CLUSTER: the "virtual mailbox" code has been redesigned.
- SECURITY: the SASL NTLM method now works with Macintosh versions of Microsoft products (Outlook Express, Entourage)
- WebSite: nested folders are supported now. The WebAdmin WebSite.html and WebUser WebSite.html and UserSiteIndex.html files have been updated.
- ADMIN: now Domain Administrators can control several Domains. See the SysAdmin section of the manual for more Details.
- CLUSTER: now Mailing List Archives can be browsed via any cluster member.
- CLUSTER: now the GETLIST and UPDATERLIST commands work with any Cluster member.
- CLUSTER: now the Create/Rename/remove operations for Forwarders and Groups work with any Cluster member.
- STATIC CLUSTER: Accounts in Shared Directory-based Domains can be administered from any Cluster member.
- WebUser: attachments in the AppleDouble format are displayed correctly.
- SMTP: the TURN command is implemented to support dial-up client sites running Microsoft Exchange servers.
- DIRECTORY: the folderIndex and adminDomainName attributes have been added to the CommuniGateDirectoryDomain objectClass in the default Schema.
- Bug Fix: CLUSTER: WebAdmin: Personal WebSite administration did not work across Cluster members.
- Bug Fix: ROUTER/DNR changes in 3.5b1 effectively disabled the SMTP RBL feature.
- Bug Fix: RULES: the Reply/React operations in 3.5b1 incorrectly inserted the From:/Sender: header field.

3.5b1 06-Jul-01

- Security: the SASL NTLM authentication method is supported now (this method allows you to use the "Secure Password Authentication" option in Microsoft products).
- POP: the parameterless AUTH command is supported now (for MS Outlook compatibility).
- Startup: Unix startup scripts have been modified to support a custom Startup.sh file in the {Base} directory.

- SMTP: now the ATRN command is always accepted from non-client addresses (after the AUTH command).
 - Initialization: now garbage collector is activated during domain initialization, so systems with several thousand domains and 10,000+ domain aliases do not abuse VRAM on restart.
 - SMTP, RPOP, LDAP, Router: all settings that should contain an A-record domain name (such as the forwarding server name) can contain several explicitly specified IP addresses, separated with the comma sign.
 - LDAP: Start TLS (RFC2830) is supported now.
 - LIST: RFC2919 (LIST-ID header field) is implemented.
 - Accounts: the Date: header field is now added to the Initial Message when it is stored in the INBOX of newly created accounts.
 - CLI: the LISTSUBSCRIBERS command is implemented.
 - CLI: the WRITELOG command is implemented.
 - WebUser: the "Trash" mailbox can now be replaced with a mailbox alias (useful to create a "shared Trash" mailbox for several accounts).
 - Bug Fix: Cluster: the "addMessages" operation did not work correctly with Virtual Mailboxes.
-

3.4.8 26-Jun-01

- Bug Fix: Cluster: certain Rule operations could cause the Local delivery protocol de-synching and backend crashes.

3.4.7 23-May-01

- LOCAL: delivery is now repeated in 1 minute if an External INBOX is locked with some other application.
- CLUSTER: backend Local Delivery now checks the Mail Disabled and Account is Full conditions.
- WebUser: the "Send as HTML" option is added to the Compose.html page.
- WebUser: the converter now displays embedded objects in HTML messages [incorrectly] created with Lotus Notes.
- WebUser: now the Drafts mailbox is auto-created during the "Save as Draft" operation.
- SMTP: the "STARTTLS" EHLO response is now presented only if the target domain has an active Certificate.
- Bug Fix: WebUser: updating mailbox ACLs from a non-owner account could clear the ACL list.
- Bug Fix: The process environment variables were not correctly passed to external programs.
- Bug Fix: SMTP: the ATRN command was issued instead of ETRN if the ATRN 'loginname' was entered into the module settings.
- Bug Fix: WebUser: large Subject lines could result in insertion of a header string with just one space symbol.
- Bug Fix: WebUser: incorrect customization of the Compose page could cause server crashes.

3.4.6 05-May-01

- Windows NT/2000: OS User Names that contain the '%' sign can now be used to explicitly specify the Windows Domain that should be used for Authentication.
- Directory: Local Units can now insert records that have multiple objectClasses specified (if one specified class is a child of all other specified classes).
- Directory: Local Units now ignore spaces in the first parts of the DN elements.
- Bug Fix: CLUSTER: The Incoming flow control operation could cause a crash of a backend server.
- Bug Fix: IMAP: the FETCH BODY[] operation could return message text without headers if the same FETCH operation included the BODYSTRUCTURE keyword.
- Bug Fix: SMTP (3.4.5) failure to create an output stream could destroy the SMTP module queue and crash the server.
- Bug Fix: RULES: the Each/Any ROUTE conditions were not available on all OS platforms (compiler-related error).

3.4.5 22-Apr-01

- Windows NT/2000: CGStarter application now can accept and remember parameters set in the Services control panel.
- SMTP: now secure connections can be used to connect to backup (ETRN/ATRNL) servers.
- SMTP: now the HELO and Return-Path parameters are not verified for connections coming from Dynamic Cluster members.
- Foundation: On MS Windows platforms NICs with more than 30 network aliases (IP addresses) are supported now.
- Bug Fix: SMTP: Send Encrypted option did not work for the messages that had to be sent via a specific Local IP address.
- Bug Fix: SMTP: the Send Encrypted feature did not re-send the EHLO command causing problems for some freeware MTAs.
- Bug Fix: SMTP: the DSN parameters were incorrectly used when sending messages with empty return paths.
- Bug Fix: WebAdmin: the Account Import function did not check the Domain Administrator access rights.
- Bug Fix: External Filtering caused a memory leak.

3.4.4 09-Apr-01

- LOCAL: the Flow Control settings are implemented.
- Listener: the Maximum Connections from the Same Address setting is implemented.
- SMTP: the module can now send the AUTH command to the Forwarding Server.
- Rules: messages generated with the Reply and React operations now use <Mailer-Daemon@maindomain> return-path address.
- SNMP: local IP address selection and remote IP address restriction settings are implemented.
- The --SharedFiles command line option is now supported for MS Windows platforms.
- AUTH: the DIGEST-MD5 SASL method is correctly re-implemented now.
- CLI: the SetClientIPs and SetBlacklistedIPs commands are implemented.
- SMTP: messages with extremely long lines (more than 100K) do not result in connection

terminations now.

- Logs: The Auto-delete setting can be set to 1 year.
- Rules: now the [RETPATH] prefix can be used with the Execute actions.
- External Filtering: License Key management mechanism has been modified to avoid processing delays after server restart. A new version of McAfee Plugin is required to benefit from this modification.
- WebUser: the content of "message/delivery-status" MIME part is decoded and displayed now.
- WebUser: the Compose function now tries to break long To/Cc/Bcc/Subject header lines into multiple shorter MIME lines.
- Bug Fix: Account: attempts to create foreign mailboxes with empty names could crash the server.
- Bug Fix: SMTP: the AUTH command sent before the ATRN command could incorrectly form the password string.
- Bug Fix: Domains could not be renamed ("domain is in use" error) if they had at least one mailing list.

3.4.3 25-Mar-01

- SMTP: the release queue methods (ETRN, wakeup-email) now release both the generic domain queues and the 'send-via-this-ip' domain queues.
- Directory-based Domains: the WebAdmin interface now implements the Load New Domains operation.
- CLI: the CREATEDIRECTORYDOMAIN and RELOADDIRECTORYDOMAINS commands are implemented.
- Foundation: support for both process-level and system-level thread scheduling is implemented.
- ADMIN: the --ThreadScope command line parameter has been added.
- RULES: Reply With operation not adds a space to the "Re:" subject prefix.
- ACCOUNT: FreeBSD-style MD5-encrypted passwords are supported now.
- CLUSTER: time-out values in Dynamic Cluster operations have been modified to ensure proper failover when the Controller network connection fails.
- WebUser: the Forgotten Password page now passes the current Domain Name to Cluster Backends.
- WebUser: the Mailbox.html page now supports the NextMessage=msgid and PrevMessage=msgid parameter to provide the "next/prev" messages switches. The NextPrevJump.html file is added.
- WebUser: processing of multipart/alternative messages has been improved (added support for Apple's Mail.app message formatting).
- Bug Fix: Mailboxes: the LIST command incorrectly checked the access rights for nested submailboxes in foreign accounts (some foreign mailboxes could be invisible even if the user had the Lookup right for them).
- Bug Fix: The MoveIMAPMail utility did not copy empty mailboxes from some 3rd party IMAP servers.
- Bug Fix: LIST: when the number of LIST processors was decreased using CLI, the excessive processors did not disappear.
- Bug Fix: WebAdmin: updating the Obscure page settings could change the Helpers page settings.
- Bug Fix: Directory Based Domains: if the automatic account creation procedure failed, the cleanup algorithm could end up in a deadlock.
- Bug Fix: WebUser: the Login page did not show the "forgotten password" link is the anti-

harvesting option was enabled.

3.4.2 06-Mar-01

- U-crpt password encoding now works on AS/400 and BeOS platforms.
- The Dynamic Cluster Monitor page is added to the WebAdmin Interface.
- CLUSTER: the Dynamic Cluster Controller fail-over algorithm is improved.
- Bug Fix: the 3.4.1 version did not properly accept the License Keys.
- Bug Fix: CLUSTER: attempts to rename or remove an account in a shared Domain after a Controller fail-over could crash the new Controller.
- Bug Fix: CLI: the RenameDomain command did not work properly in 3.3/3.4b versions.
- Bug Fix: KERNEL: delivery to groups improperly decreased the Domain usage counters. This bug produced the "open counter < 0" messages in the Server and system logs.
- Bug Fix: LIST: the text/alternative Posting Format restriction option incorrectly checked the message structure.

3.4.1 01-Mar-01

- Group and Forwarder creating/renaming algorithms have been changed.
- Shared Domains: special mailing list addresses (-report, -on, -off, etc.) now work in Shared Domains.
- TLS: max output block size is decreased to provide a workaround for products using Microsoft SSL libraries.
- LIST: the HELP command is implemented.
- IMAP, MIME: the Content-Description field is processed now.
- WebUser: the Reply operation now uses a smarter algorithm to find plain text message portions.
- WebUser: the Strings.data file now contains the "Translator" dictionary that can be used to translate pop-up menus like those used in the Rule composer.
- WebUser: the ^B and ^C macros are removed from the Rules.html and Rule.html pages (the buttons can be customized now).
- HTTP: the Personal Web Site prefix is now detected after URL decoding.
- Mailboxes: now all 'outer' mailboxes are automatically created when a sub-mailbox is created.
- Bug Fix: WebUser: the default reply text routine could incorrectly insert quotation marks. In some rare cases it could cause a system crash.
- Bug Fix: On Unix platforms the external programs did not get all environment parameters.
- Bug Fix: some Server Settings updates could cause false 'main domain renaming' operations.
- Bug Fix: 'mailbox list' routine incorrectly capitalized names of 'xxxx/inbox' sub-mailboxes.
- Bug Fix: Log recording stopped if the total size of all logs generated since the last server restart exceeded 4GB.
- Bug Fix: Domains could not be removed if the domain Default WebUser Preference set has been used at least once.
- Bug Fix: Server-Wide Rules updates could crash the server when a very long rule operation (as a suspended virus scan) was in progress.

3.4 18-Feb-01

- Directory Integration: the `mail` attribute processing can be fine-tuned now. See the LDAP Module chapter for the details.
- CLI: the `StatReset` keys were added to the Domain and Account Statistics data.
- CLI: the `CREATEWEBUSERSESSION` command is implemented.
- Forwarders: addresses without '@' and '%' symbols are qualified using the forwarder Domain name.
- WebUser: the "Print-friendly" message link is implemented, the `Strings.data "MessageHeaderEnd"` string has been modified.
- Bug Fix: MAILBOXes: in 3.4b9 'rollbacking' the target mailbox after failed copying operation could crash the server.
- Bug Fix: WebUser: not all special symbols in mailbox names were properly URL-encoded.

3.4b9 04-Feb-01

- Directory Integration: the 'Store Passwords for Regular Accounts' option is implemented.
- LIST: the Content-Transfer-encoding header field is removed from distributed 'feed' messages only if its value is '7bit' (previous versions removed the header with the value of '8-bit', too).
- IMAP: the `MULTIAPPEND` extension is implemented.
- CLUSTER: now errors on inter-server SMTP connections do not turn on delays and other anti-attack mechanisms.
- Passwords: the `--BatchLogon` command line parameter is documented.
- Passwords: now the U-crypt password encryption can be used on Windows-based servers, too.
- Bug Fix: Domains: when the Main Domain was renamed, the Domain Directory Subtree was not renamed.
- Bug Fix: WebAdmin: the size-type parameters could not be set to 'Default (XXX)' values.
- Bug Fix: Messages with several 'Subject:' headers were processed incorrectly.
- Bug Fix: LIST: the 'banned' mode could not be set for lists with disabled archiving.
- Bug Fix: WebUser: if a custom WebUser Interace file was empty, the server could crash.

3.4b8 06-Jan-01

- Directory Integration: the "UID Subtree" setting is implemented.
- SMTP: Relay To Client settings can be set to "no".
- Domains/External AUTH: the Consult External Authenticator Domain Settings is implemented.
- Admin: the "Drop Server Root privilege" options are implemented (Unix only). See the SysAdmin section for the details.
- DEQUEUEUR: dequeuer messages can be customized now (using the General->Strings page).
- POP: the message size has been added to the `RETR` command response (to make Netscape mailer show its progress bar correctly).
- Account: the Collect Account Statistics setting is implemented (the Obscure page).
- Domain: statistics on received messages is collected now.
- CLI: `GETACCOUNTSTAT`, `RESETACCOUNTSTAT`, `GETDOMAINSTAT`, and `RESETDOMAINSTAT` commands are implemented.

- MAILBOX: the "~username" mailbox aliases are implemented now - they provide access to all shared mailboxes in the specified account.
- Foundation: the OS Password checking routines now check for the OS Account and Password expiration dates.
- External AUTH: Logging changed.
- Bug Fix: INFOWEBFILES renamed into GETWEBFILESINFO, output format of GETWEBFILE and LISTWEBFILES was incorrect.

3.4b7 03-Dec-00

- Mailbox: the mailbox view creation algorithm has been changed.
- Directory: Remote Units: the Server Subtree setting is implemented.
- Directory: many attributes and objectClasses have been added to the Local Unit built-in Schema.
- Directory: the Import LDIF and Import LMOD functions are implemented in the WebAdmin Interface.
- WebUser: the X-UUEncode MIME encoding is supported now.
- MIME: UUencoded files embedded into plain text messages are recognized now.
- CLI: GETWEBFILE, LISTWEBFILES, INFOWEBFILES, PUTWEBFILE, DELETEWEBFILE, and RENAMEWEBFILE commands are implemented.
- WebUser: the UUencoded files embedded into plain text messages are displayed now.
- Account: the new method is used to recalculate the MailStore size stored in the .info files.
- Account Admin: now Account Rules can be specified in the Account Import file.
- Directory-based Domains: Personal WebSite support is implemented.
- Directory-based Domains: Account Removal is implemented.
- Directory-based Domains: Account Renaming now renames the Account files, too - if they reside within the DirectoryDomains file directory.
- Directory-based Domains: Forwarders are implemented.
- Template: Initial (default.html) page for Personal WebSites can be specified now.
- WebUser: the keywords INBOX and Trash can be "translated" in the Strings.data file now, so these "fixed-name" mailboxes can be "renamed on screen".
- WebUser: the HeaderNames dictionary is added to the Strings.data file. It can be used to change the names of the RFC822 header fields used on the Mailbox and Message pages.
- Router: local routing algorithms have been changed to restore compatibility with the old router.
- POP: the login response line contains the total size of all messages in the mailbox (to make the Netscape e-mail client happy).
- CLUSTER: Personal Web Site support now works in both Static and Dynamic Clusters w/o exceptions.
- CLUSTER: Domain Aliases are now properly initiated during the frontend and "slave" backend startups, and the domain aliases are properly removed from all servers when the domain is removed.
- CLUSTER: non-local IP Addresses can be assigned to shared domains in Static Clusters.
- Bug Fix: TLS: the 'exportable' TLS 1.0 methods were implemented incorrectly in 3.4b6.
- Bug Fix: WebAdmin/WebUser: the default numeric settings could be displayed as "-2" strings.

3.4b6 24-Nov-00

- Directory: the Search Results Limit setting is added to the Local Unit settings.
- WebUser: the Session Time Limit setting is implemented.
- LIST: the "Special" posting mode processing has been changed to make it possible to subscribe lists to other lists.
- Rules: the [RCPT] Execute operation prefix is implemented.
- RPOP: processing is re-scheduled when an RPOP record changes its poll period value.
- Directory-based Domains: when account is auto-created on first access, the INBOX mailbox and other account details (suppl. mailboxes, mailbox aliases, subscriptions, etc.) are automatically created, too.
- Template: Initial (Greeting) message can be specified now.
- CLI: GETACCOUNTLISTS and GETDOMAINLISTS return value types are changed to *dictionary*.
- TLS: TLS 3.1 is implemented.
- Queue: delivery delay warnings are implemented.
- SMTP: relaying settings have been modified.
- SMTP: the Send Warnings setting is implemented.
- Local Delivery: the Send Warnings setting is implemented.
- Bug Fix: 3.4b4-b5 versions did not store the updated Queue page Settings on disk.
- Bug Fix: 3.4b versions could show duplicate IP Addresses assigned to Domains.
- Bug Fix: TLS sessions interrupted in the negotiation phase could crash the server.
- Bug Fix: in 3.4b5, mail sent to an incorrect mailing list address could crash the server.

3.4b5 18-Nov-00

- Domains: the global DomainAliases.tdb file is phased out. Now the DomainAliases.data file in the Settings directory of the Domain subdirectories is used to store this domain aliases. The content of DomainAliases.tdb is moved automatically to those files, and the DomainAliases.tdb file is renamed into DomainAliases.tdb.unused.
- Dynamic Cluster: the Domain Aliases now work for Shared Domains, too.
- Kernel: the old DataBase Managers (handling the .tdb and .ldb files) are phased out.
- ROUTER: algorithms used for Local Domain Routing are redesigned.
Note: this version can act as a Cluster Controller for 3.4b3-3.4b4, but not vice versa. Complete Cluster Upgrade is required.
- Static Cluster: Directory-based Domains can be used for Static Clustering now.
- Domains: the Send To Forwarders option has been added to the Mail to All panel.
- WebAdmin: Administrator Preferences are implemented (see the SysAdmin->HTTP section of the manual for more details).
- WebAdmin: the Preferences settings are implemented for POP,IMAP,ACAP,LIST,SMTP,LOCAL,PIPE, and RPOP Monitors.
- WebAdmin: the Preferences settings are implemented for the Domain Account List page.
- WebAdmin: all numeric settings can now be set to some "other" (unlisted) value.
- IMAP: RFC2971 (the "ID" extension) is implemented.
- LISTS: the GETACCOUNTLISTS command is documented.
- CLI: the GETDOMAINLISTS is now the preferred name for the LISTLISTS command. This

command now works in the Dynamic Cluster environment.

- CLI: the MAINDOMAINNAME command is implemented.
- LIST: delivery to Lists now works without exceptions in both Dynamic and Static Clusters.
- LOCAL: delivery to "all" now works without exceptions in both Dynamic and Static Clusters.
- PIPE: the Queue (Wait) page is added to the WebAdmin Monitors.
- Bug Fix: the default Directory Schema did not contain the "cn" attribute for the "organization" objectClass. In 3.4b4 version, this bug made domain Directory record creation impossible. On fresh installations, the "postmaster" account was not created.

3.4b4 09-Nov-00

- Domains/SMTP: the Local IP Address for outgoing SMTP connections can be specified now. See the SMTP and Domain Settings sections for the details.
- CLI: the DELETEMAILBOX, RENAMEMAILBOX commands are implemented.
- IMAP, WebMail: foreign mailboxes can be renamed and deleted now.
- QUEUE: the Copy Failure Reports option is implemented.
- ACAP: the PREFIX, SUBSTRING, and SUFFIX filters are implemented.
- SMTP: the server now reports the "DNS Loop" situation only if a remote host name resolves into an IP Address the SMTP Listener is enabled for.
- SMTP: the domain name used in the ATRN command must be included into the Hold Mail for Domains list.
- Central Directory: the hostServer attribute can be renamed now.
- CLI: the RefreshOSData, GetRouterTable, and SetRouterTable commands are implemented.
- Rules: The Redirect All simplified Rule now has the "Preserve To/Cc fields" option.
- Central Directory: forwarder records are now included into the Domain directory subtree.
- Directory: Local: the Enforce Schema setting is implemented (it is now enabled by default).
- WebUser: if the Save a Copy option is selected on the Compose page and the mailbox with the specified name does not exist, a new mailbox is automatically created.
- Streams: the PLAIN authentication method now uses an empty challenge string.
- Migration: the MoveIMAPMail and MovePOPMail utilities now support the --noTimeout flag.
- Bug Fix: ACAP/IMAP: if the AUTHENTICATE command was interrupted, the NO response was returned instead of the BAD response.
- Bug Fix: Directory: Local Units incorrectly processed some update requests ("add" attribute sets were processed as "replace" sets).
- Bug Fix: OS Passwords did not work on the AS/400 platform.

3.4b3 30-Oct-00

- CLUSTER: a separate Domain Controller is implemented.
- CLUSTER: the Account Controller is completely redesigned.
- CLUSTER: the Account and Domain Controllers now automatically move to a different backend server if the current Controller backend server fails.
- CLUSTER: error codes reported by backends are now transparently relayed via frontends to client mailers.

- CLUSTER: POP Alert messages generated on backends are now relayed via frontends to client mailers.
- CLUSTER: backend IMAP alerts issued at the login time are now passed to the client.
- DIRECTORY: the "cn=schema" subtree is implemented. The Local Unit schema can be retrieved and extended using the 'cn=schema' record.
- Foundation: the STSkipList data structure is implemented.
- Several internal routines switched to the STSkipList structures to improve performance on large systems (1,000,000+ accounts, 10,000+ domains in queue, etc).
- SECURITY: the Hide Unknown Account Error option is implemented and it is enabled by default. See the Security section for more details
- SMTP: the Advertise 8BITMIME option is implemented.
- Bug Fix: the server could crash if someone modified the Domain Settings while a domain Private Key was being generated.
- Bug Fix: ACAP: data strings with special symbols are returned as literals now.

3.4b2 06-Oct-00

- EXTFILTERING: McAfee scanning plugin is implemented.
- SNMP: the CGatePro-MIB.txt file now includes the absolute OIDs.
- WebUser: The Thai charsets support is implemented.
- TLS: the Generate Key option is added to the Domain Security page.
- TLS: support for 2048-bit keys and long Certificates is added.
- TLS: the Certificate Authority Chain option is implemented.
- TLS: the nested TLS negotiations are supported now. They are used to implement strong (128-bit) SSL encryption with weak (40-bit), "export-legal" products.
- Shared Domains: if LDAP connections to the Central Directory fail, the SMTP module now returns a non-fatal error code for unroutable addresses.
- Bug Fix: Directory: Multi-Level searches in Local Units could return incomplete DNs. This could also cause the LDAP module to stop on-the-fly creation of the "mail" attributes.
- Bug Fix: incorrect parallel initialization could cause the PIPE module to crash on startup.
- Bug Fix: the RPOP module could accept To: and Cc: addresses as "trusted" when working without the "Special-header" option (this could result in unwanted relays).
- Bug Fix: ENQUEUEER: the default value for ENQUEUEER threads is 1 now.

3.4b1 07-Aug-00

- QUEUE: ENQUEUEER design is multi-threaded now.
- HELPERS: Content Filtering/Anti-Virus API is implemented (see the Rules section).
- CLUSTER: "Virtual" Mailbox objects are implemented (allowing an account opened on one server to access mailboxes in an account opened on a different server).
- CLI: LISTMAILBOXES, CREATEMAILBOX, GETMAILBOXINFO, GETMAILBOXACL, SETMAILBOXACL, and GETMAILBOXRIGHTS commands are implemented.
- Mailboxes: ACL subsystem has been redesigned.
- Mailboxes: MDIR mailboxes are now parsed correctly even if the message internal date was set to

0.

- CLUSTER: PWD module now returns different codes for some key error messages.
 - Accounts: excessive file operations are removed from the account opening procedure.
 - WebUser: Japanese (ISO-2022-JP) characters are now correctly processed in the message header fields on the Compose page.
 - WebUser: The UTF8 mode for the Japanese (Big5 and GB2312) encodings is supported now.
 - WebUser: The UTF8 mode for the Japanese (ISO-2022-JP) encodings is supported now.
 - Notifier: the Log Level and Queue size can now be specified using the Obscure page.
 - Security: External Authenticator internals are redesigned, its settings are moved to the Helpers page.
 - Bug Fix: Personal WebSite: URLs for site files did not contain URL escape symbols, some of the access utilities did not remove the URL escape symbols.
 - Bug Fix: POP, IMAP: SASL AUTH methods incorrectly supported the "short-form" syntax.
 - Bug Fix: CLUSTER POP login could fail on backends.
-

3.3.2 02-Oct-00

- Security: the Certificate processing buffer size has been increased from 1K to 4K.
- Bug Fix: SMTP: Dynamic Cluster backends could crash when the mailbox STORE operation failed.
- Bug Fix: incorrect parallel initialization could cause the PIPE module crash during startup.

3.3.1 07-Aug-00

- BeOS version is released.
- Bug Fix: WebUser: processing format=flowed texts could cause crashes on some platforms (AS/400).
- Bug Fix: SMTP: malformed ETRN could cause crashes.
- Bug Fix: CLUSTER SLAVE: the controller response parser could crash the server.
- Bug Fix: LIST: automatically-generated messages were processed incorrectly.
- Bug Fix: UTF8/Unicode decoding procedure did not work correctly for several charsets.

3.3 16-Jul-00

Update Note: the 3.3 version uses a completely new Directory Manager. If some of your Domains had the Directory Integration setting set to Keep In Sync, open those Domain Settings in the newly installed 3.3 version, and click the "Insert All" button in the Directory Integration Panel.

- Domains: Mailing lists and Groups are now automatically added/updated in the Directory if the Domain Integration setting is set to Keep In Sync
- Domains/Accounts: now OS Names can be explicitly set for individual Accounts.
- HTTP: CGI programs now inherit the environment variables of the Server (under Windows, this is needed to open TCP/IP sockets in CGIs).
- Rules: the "[FILE]" and "[STDERR]" Execute command tags are implemented.

- IMAP: the APPEND command now checks if the message text lacks the trailing EOL and fixes it. This is a workaround for the Netscape Messenger bug.
- Groups: the Remove Author Address option is implemented.
- Groups: sending to a non-empty group with all group addresses removed is processed as normal (final) delivery now.
- ACAP: datasets entry names are case-insensitive now.
- WebUser: the Sent and Draft mailbox names are properly "defaulted" now. The WebUser Settings.html page has been changed.
- WebUser: the simplified Chinese (GB2312) charset is supported now.
- SNMP: 64-bit Counters are implemented, some MIB elements have been switched to the COUNTER format.
- Bug Fix: Directory: Browser: URL escape symbols were not removed correctly, causing problems for Netscape browsers.
- Bug Fix: POP: the CAPA command was not processed in the TRANSACTION state.
- Bug Fix: binary zeros in message headers could crash the server.

3.3b9 06-Jul-00

- Manual: the Directory, Directory Integration and Clusters pages are updated.
- WebAdmin: Obscure: the Central Directory settings are phased out.
- Directory: File(Local Units): the updates merging daemon is implemented.
- Directory: Storage Unit deletion is implemented.
- Directory Integration: the Delete All operation now removes only the records for accounts created on this Server.
- WebUser: Korean (ISO-2022-KR) and Chinese (Big5) letters are correctly converted into HTML code.
- WebUser: the Certificate link is displayed only if the domain has a Custom Certificate.
- WebUser: the Mailing Lists link is displayed only if the domain has some mailing lists.
- Foundation: a faster version of 'write to file' method is implemented (now used to store aliases, forwarders, groups, and account info).
- LDAP: case-insensitive dictionaries are now used in modify-type operations.
- HTTP: CGI environment variables HTTP_AUTHORIZATION and HTTP_REFERER are added.
- HTTP: CGI program name in a URL can now be followed by '/' and some URL string.
- Bug Fix: Mailboxes: 3.3b6-b8 versions might not show INBOX in the mailbox list, if INBOX was an external mailbox.
- Bug Fix: Routing: Directory-based Routing did not work in 3.3b3-b8.
- Bug Fix: under IRIX, AIX, HP/UX daylight saving times for the local time zones was not detected correctly.
- Bug Fix: Rules: Rule Editor could crash the server if some of the condition or action fields were missing in the (customized) form.

3.3b8 21-Jun-00

- LDAP: the 'mail' attribute is now composed on-the-fly for records of the CommuniGateAccount

objectClass.

- LDAP: the authentication methods are improved and documented.
- WebUser: the Directory Search page can now use the "internal CGatePro" names instead of the standard attribute names (i.e. `RealName` instead of `cn`).
- WebUser: Security: the Security Certificate (RFC2585) link is added to the Login page. See the Security section of the manual.
- WebUser: Non-ASCII mailbox names are supported now.
- WebUser: format=flowed processing (RFC2646) is implemented.
- Router: the Add *name* to Non-Qualified Domain Names option is implemented.
- LOCAL: the Account Detail addressing Routing options are implemented.
- TLS: session recycler is implemented.
- TLS: the SSL 3.0/3.1 interaction is improved (interoperability with both Microsoft products and open source utilities).
- SNMP: TLS monitoring agents are implemented.
- Mailboxes: .mdir (MailDir) format is redesigned to store the number of message text lines in message file names. This should help some mailers (such as Netscape) correctly process messages with attachments retrieved from Unix servers.
- LIST: feed headers and trailers are correctly inserted into base64-encoded messages.
- Transfer: message with extremely long header fields (>100K) are now rejected with the SMTP, RPOP, PIPE, or POP (XTND XMIT) modules.
- Bug Fix: WebMail: in 3.3b7 2-byte charsets could result in infinite loops.
- Bug Fix: SMTP: the "relayHost" field was not always filled correctly resulting in garbage on the SMTP Monitoring pages.
- Bug Fix: IMAP: in 3.3b6-b7 the LIST "%" command might not list folders that were not mailboxes at the same time.

3.3b7 12-Jun-00

- SMTP: RFC2645 (ATRN) is implemented in both server and client modes to support mail delivery to hosts with dynamic IP addresses.
- WebUser: texts using japanese ISO-2022-jp charsets should be displayed correctly now.
- LOCAL: all-domain aliases are case-insensitive now.
- LOCAL: the new Alert Text option allows you to specify the "over the quota" alert message text.
- TLS: the SSL 3.0/3.1 interaction is improved (interoperability with open source utilities).
- Domain: the 2-Letter 2-Level Domain Hashing now provides a workaround for accounts with 1-letter names and for accounts that have the dot symbol as the second symbol of their names.
- Bug Fix: IMAP: in 3.3b6 the LIST commands with non-empty prefixes returned incorrect results.
- Bug Fix: IMAP: in 3.3b6 the SELECT command could improperly capitalize the inbox mailbox name, resulting in duplicated (INBOX and inbox) records in the account.info file.
- Bug Fix: OS/400 version improperly passed parameters to external tasks.

3.3b6 01-Jun-00

- Linux/StrongARM version is released.

- Directory: Access Rights (ACLs) are implemented.
- Directory: Browser and Access Right WebAdmin pages are implemented.
- Alerts: automatic "account is over quota" alerts are implemented. See the SysAdmin->Alerts section of the manual.
- LIST: the Digest generator has been modified to fit the RFC1153 requirements.
- SMTP: secure connections with forwarding servers are supported now.
- Mailboxes: mailbox aliases are transparent now (included into mailbox hierarchy views).
- WebUser: longer, alpha-numeric "session passwords" are used now.
- WebUser: the DirectoryFields arrays are added to the Strings.data file.
- IMAP: SNMP monitoring agents are implemented.
- WebUser Interface: the Files realm is implemented to provide access to arbitrary files in the WebUser directory.
- SysAdmin: the --noLockFile option is implemented.
- CLI: GetAccountLocation, GetServerRules, and SetServerRules commands are implemented.
- Bug Fix: IMAP: 3.3b5 did not place the quote marks around the "boundary" parameter value in the BODYSTRUCTURE response.
- Bug Fix: Directory: in 3.3b3-5 local Storage Units could generate the "non-text data" errors.
- Bug Fix: CLI: GetAccountAliases command could crash the server if the specified domain did not exist.
- Bug Fix: ACAP: multi-level searches in Dictionary DataSets could cause synchronisation deadlocks.
- Bug Fix: ACAP: MODTIME responses for the SEARCH command were returned untagged.
- Bug Fix: ACAP: the optional metadata list in the RETURN clause of the SEARCH command was not properly parsed.

3.3b5 01-May-00

- Mailboxes: Mailbox Aliases are implemented (see the Objects->Mailboxes section of the manual). Mailbox aliases can be used to provide access to foreign mailboxes for IMAP clients (such as MS Outlook / OE) that cannot process foreign mailbox names in the Mailbox Subscription lists.
- WebAdmin: the Account Template page now includes the Initial Mailbox Aliases panel.
- Directory: Remove and Relocate Storage Unit operations are implemented.
- IMAP, ACAP: the output buffering method has been changed.
- Domains: the Generate Index option is implemented. It can be used to decrease the restart time for domains with 100,000+ accounts.
- Rules: the Current Day conditions are implemented.
- Bug Fix: SMTP: the 3.3b4 version crashed if the Send Encrypted to Domains setting contained an empty list.

3.3b4 24-Apr-00

- Security: SSL/TLS client-side connections are implemented.
- SMTP: the Send Encrypted option is implemented to support server<->server encrypted message transfer.

- WebUser: attachment file names and HTML text portions are MIME-decoded and (optionally) converted to UTF-8 now.
- WebUser: the Use MIME for Headers option is implemented. When this option is selected, the Subject, To, Cc, and Bcc header fields containing non-ASCII symbols are MIME-encoded.
- WebUser: Alerts, Bye, Hello, List, ListArchive, Public, Rules, Rule, RPOP, Subscribers, Subscription pages now include the '=' (charset) macro symbols.
- WebUser: the '^\$' macro (domain name) now works in all Account pages.
- WebUser: all ISO-8859-x and windows-125x charsets are now supported in the UTF-8 mode.
- Foundation: DNS addresses are now correctly retrieved from the Windows 2000 Registry.
- WebAdmin: clicking the Refresh button on the General Settings page tells the Server to re-read the DNS addresses from the OS.
- LIST: the First Digest At setting processing has been changed (see the LIST module manual).
- Rules: the Current Date and Time of Day Rule conditions are implemented.
- Rules: the Write To Log action is implemented.
- Bug Fix: WebUser: the "New" counters on the Mailboxes page were not updated properly.

3.3b3 17-Apr-00

- Directory: multiple Sub-tree storage units are implemented.
- Directory: schema editor is implemented.
- WebUser: UTF8 Mode Setting is implemented. The Settings.html and Compose.html pages have been modified.
- WebUser: the Settings.html, RPOP.html pages have been changed (^m has been added).
- WebUser: Compose: now the From, To, Cc, and Reply-To original letter headers are MIME-decoded first.
- Mailboxes: BSD Mailbox Manager now checks the size of text lines in new messages.
- Admin: the WebSite, BasicAccountSettings, and WebUserSettings Domain Access Rights are implemented.
- SNMP: WebUser monitoring agents are implemented.
- SNMP: POP module totaling-type reporting is implemented.
- Directory Integration: the special-case dc RDN attribute is supported now.
- Rules: comparison operations now ignore the surrounding angle brackets in the envelope (Recipient, Return-Path) addresses.
- Manual: the WebMail section is created to contain all user-level information about the WebUser Interface.
- Manual: the HTTP and HowTo sections are updated to provide the information about username.domain.dom Personal Web Site URLs.
- Bug Fix: the Header-field Rule condition did not process wildcard symbols correctly.

3.3b2 02-Apr-00

- Linux/Sparc version is released.
- SNMP: StalkerMIB ObjectID has been changed to 5678 - the IANA-registered Stalker ObjectID.
- Directory: the Directory Manager is implemented.

- Directory: the File-based Directories are implemented.
- Directory: the LDAP-based Directories are implemented (with caching)
- Directory: the .tdb and .ldb DataBase Managers are being phased out.
- Domains: the Directory Integration Settings are implemented.
- LDAP: implementation is based on the Directory Manager now.
- LDAP: RFC2254 is implemented.
- Account/Domain: the Directory-based Domains are partially implemented.
- RPOP: support for remote POP servers that do not return the number of messages in the initial UIDL response.
- Account/Domain: the "Mobile" setting is enforced now: domains and account w/o that Access Mode cannot be accessed from non-client IP addresses.
- SMTP: Log reporting of relayed messages is improved.
- HTTP: WebUser port handling has been changed to support domain-style Personal Web Sites (i.e. http://user.domain.com can be processed as http://domain.com/~user/ now).
- WebUser: Korean, Chinese, and UTF-8 encodings are supported now.
- WebUser: the Mailbox page now remembers the filter and search strings and the position in the selected message set.
- WebUser: the Empty Trash button is now implemented on the Mailboxes Page.
- DEQUEUEUR: engine is redesigned and converted into a multi-threaded one.
- SMTP: statistical data (available via SNMP) is extended.
- SMTP: the SIZE EHLO response does not include the '=' sign now.
- IMAP: some responses are sent in batches now. This should improve performance over SSL connections.

3.3b1 03-Mar-00

- SNMP agent is implemented.
- Groups are implemented. GROUP-related CLI commands are implemented.
- Forwarders are implemented. FORWARDER-related CLI commands are implemented.
- LIST: the Special posting mode is implemented to allow subscribing of a mailing list to some other list.
- SMTP: the AUTH=address Mail From parameter is now supported.
- SMTP: batch-splitting for messages with multiple recipients in one domain is implemented.
- Bug Fix: CLUSTER: slave nodes refused to connect to the controller if one of the shared domains had the WebCache option disabled.
- Bug Fix: SMTP, RPOP: if a remote server was misconfigured, garbage left in the response buffers could cause problems when connecting to other servers serving that remote domain.

3.2.4 14-Feb-00

- HTTP: document name extensions are now converted into lowercase before checking them against the file extensions specified in the MIME Type table.
- The sendmail and mail utilities now use the backslash separators to compose files names on Win32 platforms.

- CLUSTER: SMTP AUTH is now supported for accounts in shared domains.
- Security: serverKeyExchange protocol is implemented to let "exportable" browsers connect to the domains that have strong (>512 bit) keys.
- Security: now domain aliases can be specified for Certificates and CSRs.
- Bug Fix: updating account settings via the WebAdmin Interface did not always update the Central Directory.
- Bug Fix: WebSites: references to Personal Web sites for users that had dot (.) symbols in their account names did not work without a trailing slash.
- Bug Fix: IMAP: COPYUID reporting was incorrect when mailbox messages were copied into the same mailbox.
- Bug Fix: in 3.2.3 the Mailbox#Username scheme did not work for POP APOP login.

3.2.3 28-Jan-00

- RULES: the Additional headers set with the Server-Wide Rules are now used with the LOCAL delivery and SMTP modules (before, those headers could be used only for Store and Redirect operations started from the same set of Server-Wide Rules).
- Foundation: socket-reading routines have been changed to avoid race conditions.
- WebUser: the Inbox name in the Mailbox Subscription lists is now processed with case-insensitivity.
- CLUSTER: the Authenticate then Relay feature now works for POP and PWD clients connecting via front-end servers.
- Migration: the MoveIMAPMail program now understands literals in mailbox names and can copy mailboxes with names containing space characters.
- Bug Fix: CLUSTER: the front-end addresses were used instead of the client addresses when a client was closing connections and the client address was being removed from the TempClients table.
- Bug Fix: WebUser: some of the mailbox view parameters were not processed correctly if they were set to the "default" values.
- Bug Fix: WinNT: passwords verification using NT accounts (LogonUser function) could crash the server or result (later) in various errors if an incorrect password was supplied.

3.2.2 18-Jan-00

- SMTP: the Advertise AUTH option can be set to Non-Clients now.
- LIST: the Who can browse option can be set to "Clients" now.
- Bug Fix: CLUSTER: the PWD module could not connect to back-end servers.

3.2.1 14-Jan-00

- Security: the TLS module 128-bit encryption is enabled.
- Bug Fix: CLUSTER: some admin operations did not convert account names to lowercase resulting in Controller de-synchronisation and auto-shutdown.

3.2 12-Jan-00

- Additional Mailing List Licenses are supported now.
- Bug Fix: certificates with long serial numbers were not accepted.

3.2b9 26-Dec-99

- Foundation: passing environment parameters now works on the WIN32 platforms, too.
- HTTP: CGI applications can receive POST/PUT request data now;
- HTTP: Settings->WebUser page now allows to enter "starter" programs for cgi applications with specified extensions (so Perl scripts can now be used on Win32 platforms, too).
- Admin: the "Mobile" access mode is renamed into the "Relay" (existing settings should be updated automatically).
- Admin: the new "Mobile" access mode is added.
- Admin: the CertificateType Domain access right is added (when disabled, the domain admin cannot modify the Domain Security settings).
- SMTP: the alternative delay interval settings are implemented.
- IMAP: the size limitation for the APPEND command has been removed.
- LOCAL: the All-Domain Aliases are implemented (see the Local Delivery section of the manual).
- The {NS-MTA-MD5} -type digested passwords are supported to simplify migration from Post.Office servers. See the Migration section for the details.
- WebUser: the https:// URLs are now detected in plain-text message bodies and converted into clickable links.
- WebUser: the speed of composing HTML pages for large mailboxes is increased.
- Bug fix: HTTP: a channel could crash when returning CGI results longer than 1 pipe block.

3.2b8 15-Dec-99

- Foundation: the synchronisation routines have been modified on all platforms to improve performance under a heavy load.
- Foundation: optimized timer in 3.2b6-7 could result in queue processing delays on multi-processor systems.
- Obscure: the Use Conservative Info Updates option is implemented. Can be disabled on heavily loaded systems, especially if accounts are stored on an NFS server.
- The Do Not Redirect Automatic Messages option is added to the Simplified Redirection Rule.
- The {MD5} and {SHA} digested passwords are supported to simplify migration from Netscape and software.com servers. See the Migration section for the details.
- Secondary Domain names are explicitly converted to small letters now.
- The QUEUE Log records for new messages now include the message Return-Path.
- WebUser: when an open message is moved to a different mailbox and the account Delete mode is not "Mark", the original message is deleted.
- Rules: the Each/Any Recipient conditions are renamed into the Each/Any Route conditions, and the new Each/Any Recipient conditions are implemented (see the Rules section for the details).
- Bug Fix: domain renaming involving Central Directory updates could cause crashes.
- Bug Fix: in 3.2b6-7, changing the poll period of an RPOP record could result in RPOP record

disappearing.

3.2b7 07-Dec-99

- Many routines are switched from linear search algorithms to binary search ones.
- Security: the per-Domain Custom certificates are implemented.
- WebUser: the "Recover passwords via E-mail" functionality is implemented: the Password.html page is added to retrieve forgotten passwords. A link to that page (appearing on the "Incorrect Password" error) is added to the Login (default.html) page. The Settings.html page has been modified to let users change the Recovery Address setting.
- Admin: the speed of importing accounts with aliases is improved.
- SMTP: AUTH command is accepted from Client IP Addresses only if the Grant Access to Clients Only option is enabled.
- RFC2595 is implemented: IMAP, ACAP and POP modules support the STARTTLS/STLS command now.
- Bug Fix: Rules: the Return-Path condition did not work in 3.2b6.
- Bug Fix: RPOP: double-unlocking of the RPOP queue could generate exceptions in 3.2b6
- Bug Fix: the 3.1-3.2 versions did not properly clear the WebUser cache.

3.2b6 29-Nov-99

- The Dynamic Cluster code is included into the mainstream releases.
- SMTP: recipient address checking has been changed to accomodate advanced Router techniques (routing to non-standard ports, bypassing MX records, etc.)
- DNR: now TCP connections are used to read extra-large DNS responses.
- CLI: the GETDOMAINALERTS and SETDOMAINALERTS commands are implemented
- CLI: the SETDOMAIN, SETACCOUNT, SETACCOUNTDEFAULTS, SETACCOUNTTEMPLATE commands are implemented.
- CLI: the GETACCTNTRPOP and SETACCTNTRPOP commands are implemented
- CLI: the CREATESHAREDDOMAIN command is implemented
- CLI: the GETWEBUSER, SETWEBUSER, GETWEBUSERDEFAULTS, and SETWEBUSERDEFAULTS commands are implemented
- CLI: the GETWEBUSERINTERFACE, PUTWEBUSERINTERFACE, DELETEWEBUSERINTERFACE, LISTWEBUSERINTERFACE, and CLEARWEBUSERCACHE commands are implemented
- CLI: the LISTMAILBOXES command is implemented
- CLI: the ROUTE command output format has changed;
- ROUTER: the ".here" domain names suffix is supported now.
- Admin: the Refresh button is added to the General Settings page.
- WebAdmin: the WebAdmin interface and WebUser interface Editors pages have been moved inside the WebAdmin/Accounts directory.
- WebUser: custom domain files cache speed is improved.
- Rules: the "in" and "not in" condition operations are implemented.
- Bug Fix: PIPE: module routing was broken in 3.2betas.

- Bug Fix: the General Settings WebAdmin page did not update all its settings in 3.2betas.
- Bug Fix: the mobile client support (Temp Addresses) was broken since 3.2b2.
- Bug Fix: SMTP: the module could crash if the highest priority MX record for a recipient domain had a priority value ≥ 32768 .
- Bug Fix: IMAP: the BODYSTRUCTURE response was not composed correctly for Content-type: message subparts.

3.2b5 04-Nov-99

- IMAP: the fld-disp BODYSTRUCTURE extended data field output has been modified to provide compatibility with legacy imap clients.
- IMAP: the Literal format is used for ENVELOPE and BODYSTRUCTURE fields containing non-ASCII or special characters.
- SMTP: AUTHorization can be used to clearup the "blacklisted" host flag.
- LIST: moderated subscribing mode is implemented
- LIST: messages stored in the *listname*/request mailbox now get the X-List-Report header.
- Admin: the CanAccessMailboxes domain administration Access Right is implemented.
- WebUser: letters/parts in the multipart/related format are supported now.
- WebUser: the "Open Mailbox" feature is added to the Subscribed Mailboxes page.
- Router: the default Relaying mode for Router records has been changed to Relay and the NoRelay tag is implemented. See the Router section for the details.
- Rules: the Any Recipient and Each Recipient conditions are now supported in the Server-Wide Rules.
- Bug Fix: very long input lines (>65K) sent to the service streams could cause server crashes.
- Bug Fix: memory leak in Domain Routing is fixed.
- Bug Fix: memory leak in WebUser Interface (Message viewer) is fixed.
- Bug Fix: WebUser: the Mailbox Refresh settings was not working correctly.

3.2b4 25-Oct-99

- RBL: now supporting RBL servers such as ORBS that return non-standard responses.
- WebSite: CGI scripts are now supported (use the Settings->WebSettings page to specify the directory where the the /cgi-bin/ program files are located).
- WebSite: the banner insertion routine is improved.
- WebUser: the Show HTML inline/in frame option is implemented.
- WebAdmin: the Domains page processing is improved (aliases, sorting).
- Bug Fix: if a Secure (SSL/TLS) connection failed on an early stage of negotiation, the server could crash.
- Bug Fix: in 3.2b1-3 domain aliases for the main domain could not be removed.
- Bug Fix: IMAP: the BODYSTRUCTURE response had a space between the "multipart" part descriptors and did not have spaces between multiple part-body attributes. This could cause a problem to the "pine" IMAP client.

3.2b3 18-Oct-99

- Domains: Reroute Mail to Unknown: the "*" symbol processing is improved to put quotes back into the original "quoted" addresses.
- WebUser and Rules: if a recipient E-mail address is specified without the domain part, the domain name of the current account is added to the envelope address.
- Security: the U-crpt password encryption is supported for Unix platforms. This is useful for moving account passwords from old servers. See the Security section for the details.
- Security: now empty CommuniGate Passwords cannot be used for logging in, but they can be updated if users have logged in using OS or External Authentication.
- WebUser: attachments processing on the Compose.html page has been changed (to allow localization and modification of max. number of attachments). Update your customized Compose.html pages.
- Admin: the UnixPassword column is now supported in the Account Loader import files.
- Bug Fix: in 2.3b1-2, the SMTP AUTH workaround for Netscape Mail servers was not working.
- Bug Fix: in a client using SMTP AUTH could crash the server.
- Bug Fix: an uninitialized variable problem could crash the PWD server (accessed via its TLS port).

3.2b2 10-Oct-99

- Account Management: storing Account Settings in Central Directory database is implemented.
- Cluster: frontend-backend Clustering for SSL/TLS connections is implemented.
- Cluster: the Directory-based Cluster Routing is implemented.
- RPOP: the Leave on Server option is implemented.
- LDAP: array-type database data elements can be returned now (as multi-value LDAP SET elements).
- DataBase: TextManager (.tdb) databases now support complex (array, dictionary) data elements.
- DataBase: LDAPManager (.ldb) databases now support array-type (multi-value) data elements.
- SMTP: the "Use Forwarding Server" processing mode has been changed.
- SMTP: the '#' and '@' ETRN parameter prefix symbols are accepted (and ignored) now.
- MIME: headers with "folded strings" are processed correctly now.
- Account, Domain, SMTP, POP: the Mobile "Account Service" setting is implemented. Check that this option is enabled for all your mobile users. This is an anti-spam measure useful for sites providing free WebMail services.
- BSD Mailbox: message flags updating is delayed now. This should increase performance on under heavy load, especially POP3 load.
- Temp File: the Recycle Temp File option is implemented (see Obscure settings); this should increase performance under heavy load.
- The "Mail to Unknown" setting now applies only to mail routing, not to account name (Access) routing.
- WebSite Uploading: application/x-machbinary format is decoded now, so MS Explorer on Mac can be used to upload Web files.
- Bug Fix: LDAP: in 3.2b1 BIND-ing as "anybody" (using an empty username string) could crash the server.
- Bug Fix: 3.2b1 Account Rules Web interface failed to create a new Rule.
- Bug Fix: SMTP: the "Relay for hosts we backup" option did not work correctly.

- Bug Fix: IMAP: the LIST "" "name/%/%" command did not work correctly (Netscape could not see nested mail folders)
- Bug Fix: the Central Directory "Who can Browse" option was reset to "Anyone" after server restart.
- Bug Fix: message envelope addresses containing 8-bit characters inside a q-string were rejected.

3.2b1 17-Sep-99

- SMTP, POP, IMAP prompt strings and responses can be customized now. See the SysAdmin section of the manual for the details.
- LDAP: the AddRecord, DeleteRecord, ModifyRecord, and ModifyDN operations are supported now; the LDAP user should be authenticated and should have the All Domains server access right to be able to use these operations with the Central Directory database.
- The LDAP-based DataBase manager is implemented. See the Data section of the manual for the details.
- ACCOUNT: now if the Secure Authentication is set to Required, clear text passwords are still accepted, if they are passed via a secure (SSL/TLS) connection.
- ACCOUNT: an external authenticator can be used for SASL-type authentication.
- WebAdmin: access to account WebUser settings is now provided via a link on the Account Settings page.
- WebUser: the WebUser Settings now have the sever-level and domain-level Defaults (same mechanism as used for the Account Settings). The Defaults can be set by a server/domain administrator using the links on the Default Account Settings pages.
- WebUser: the Settings page: the ^A macro has been changed to ^b and the ^L macro has been removed.
- WebUser: Personal WebSite management has been moved inside WebUser Session processing (no additional authentication required).
- WebAdmin: access to the Personal WebSite management is now provided from within the WebAdmin "Domains" realm (no additional authentication required).
- WebUser: the SaveDraft and Save Sent operations now save messages with the "seen" flag pre-set.
- RULES: additional headers specified for React/Reply messages can now contain macro symbols.
- RULES: additional headers specified for Reply messages can now contain To/Cc header fields.
- RULES: additional headers specified for React/Reply messages can now contain Bcc header fields.
- CLI: the GetAccountRules and SetAccountRules commands are implemented.
- Bug Fix: LIST: "account closing" house-keeping procedure could cause a deadlock in the entire kernel (via locking the Router).
- Bug Fix: WebAdmin: WebSession Monitor processor generated the "STObject refCount != 0" messages in the OS system Log. This bug should not cause any problem.
- Bug Fix: the 3.1 LIST "cleanup" fix was not correct, unconfirmed subscribes still could be removed too early.
- Bug Fix: sometimes when the INBOX mailbox was renamed, a new INBOX was not created.

3.1 01-Sep-99

- CLI: the UpdateModule command is added.
- CLI: the GetAccountInfo command is added.
- LIST: refs to the list Web Archive now point to the secure port if the unsecure one is not enabled.
- WebAdmin: the Account Settings page now contains a link to the user Personal Web Site.
- Bug Fix: LIST: unconfirmed subscription requests were deleted from the subscriber list during each "cleanup" job, without the 2 days waiting interval specified in the manual. As a result some new subscribers had to send confirmation requests twice.

3.1b9 25-Aug-99

- WebUser/WebAdmin: image buttons are now accepted in multipart/form-data forms.
- WebAdmin: an administrator can now remove Alert messages.
- Domain: security options can be specified for "all"-type message distribution.
- WebUser: the mailing lists page now displays the list names in the sorted order.
- WebUser: the Via Trash and Immediate delete modes are implemented. The Delete Method setting is added to the WebUser Settings page.
- WebUser: several additional message strings were placed into the Strings.data file.
- WebUser: the Baltic, Arabic, and Chinese charsets are added.
- Rules: the Header Field condition operation is implemented.
- Bug Fix: Accounts Importing did not work in 3.1b8.
- Bug Fix: the 3.1b7-b8 version did not place the "trusted" flag into the Mirror-ed messages.
- Bug Fix: updating the RPOP accounts could crash the 3.1b8 version.
- Bug Fix: the WebAdmin SMTP Read Monitor could crash if it saw a stream going through a clean-up procedure.

3.1b8 23-Aug-99

- MIME headers decoder now converts the underscore sign into the space symbol.
- Accounts: the External Authenticator is implemented (see the Security section of the manual).
- WebUser: if the Subject field of the composed messages contains 8-bit characters, it is sent in the MIME-encoded form now.
- PIPE: message delivery is multi-threaded now.
- PIPE: the module now processes the Envelope-Ignore and the Envelope-ID fields in submitted messages.
- LIST: the Insert Feed Prefix after Reply Prefix option is implemented.
- Kernel: the thread-killing mode is implemented.
- Rules: the Redirect/Mirror/Forward operations now put "additional headers" into the generated messages.
- sendmail: the --, -t, and -V parameters are supported now; processing of the -f and -F parameters has been improved.
- Bug Fix: the server could crash when displaying the Queue WebAdmin Monitor page on a heavily loaded system.
- Bug Fix: the server could crash when trying to access a just-deleted mailbox message if the message was the only one in that mailbox.

- Bug Fix: 3.1b7 direct mailbox addressing could crash the server.
- Bug Fix: 3.1b7-b6 could crash when inserting "hot links" into message HTML representations.
- Bug Fix: ACAP: comparison operations with the NIL operand could crash the server.

3.1b7 06-Aug-99

- Protection: multiple RBL Servers are supported now.
- SMTP: the name of the RBL server is now included into the rejection response.
- SMTP: unqualified recipient names are now qualified with a secondary domain name, if the SMTP connection is established to an IP address assigned to a secondary domain.
- SMTP: the initial prompt timeout for not-last-chance relays increased to 2 minutes.
- SMTP/LOCAL: the Host Monitor pages now allow a system operator to specify the error code for the Reject operation.
- The Mirror-to operation (applied to queued messages) now preserves the original Return-Path envelope address.
- Account: the New MailBoxes Type setting is implemented.
- Account: Mailbox case sensitivity is enforced when running under MS Windows.
- LOCAL Delivery: Direct Mailbox Addressing is implemented.
- POP: Direct Mailbox Addressing is supported.
- Mailbox storage: "common base" storage is implemented for "Directory" mailboxes (when a queued message is being delivered to many directory-type mailboxes, only one OS file is created).
- WebUser: the AutoWrap option is implemented. The "wrap=virtual" option is added to the Compose.html page.
- WebUser: the charset (^=) macro combination is added to service pages (mailboxes.html, settings.html, etc.)
- Redirecting/Forwarding: when multiple addresses are used, all addresses are included into the "To" header.
- Bug Fix: the new HTTP Redirect realm did not always set the text/html Content-type for its response.
- Bug Fix: in 3.1b6, updating the Router table on a Unix server inserted "return" symbols into the Table text.
- Bug Fix: WebUser: references to foreign mailboxes (return jumps) were composed incorrectly.

3.1b6 02-Aug-99

- Central Directory: the Who Can Browse setting is implemented.
- Access: the Server-wide and Domain-Wide Alerts are implemented.
- LIST: the Keep To And Cc option is implemented.
- WebUser: web access to Central Directory is implemented (for both Logged in and Guest users).
- WebUser: the mailto "hot links" seen within a WebUser session open the WebUser Composer now.
- WebUser: now http "hot links" are processed via a Redirecting page to prevent Referer-based account cracking.
- the sendmail legacy mailer emulator is added.
- Bug Fix: account caching mechanism had a synchronization bug, which could cause memory

corruption problems, especially under a heavy load.

- Bug Fix: on some PPC platforms the domain alias data was not processed correctly.
- Bug Fix: WebUser: references to foreign mailboxes were composed incorrectly.

3.1b5 19-Jul-99

- Domain: the Server OS User Name picture setting is implemented.
- Account: now OS passwords can be used for accounts in all domains (using the OS User Name picture settings).
- Rules: the FingerNotify action (useful for the NotifyMail® users) is implemented.
- HTTP: now the Date and Last-Modified HTTP headers are returned with static objects.
- WebUser: file extensions -> MIME association table is implemented (WebSession Settings)
- WebUser: now all message text components are displayed in the "wrappable" format, rather than in the "preformatted" format.
- WebUser: the Received header with the user current IP address is included into all composed messages.
- WebUser: the URLs in message bodies are presented as active URL links.
- WebUser: an alternative WebUser Session access method (/Session/ID/xx instead of /Account/xx?SID=ID) is implemented; the netrance and several default pages have been switched to use this method, and the "Message.html" portion processing is switched to use this method to avoid problems with file downloading using MS Internet Explorer.
- WebUser: Greek and Central European charsets are added. Charsets are now displayed by name rather than by their MIME codes.
- SMTP: messages received via authenticated connections are marked as "submitted by authenticated users".
- SMTP: the maximum number of parallel sending channels is automatically opened when the domain queue contains too many messages.
- LIST: duplicate envelope addresses are processed as one address now.
- LIST: the Verify Owner setting is implemented.
- LIST: the current number of subscribers is displayed on the List Monitor and Account Setting pages (when that number is available).
- LIST: the Notify Owner option is implemented.
- LIST: now the message body charset is specified in List service messages (Warnings, Welcome, etc.).
- SMTP: the ETRN command now applies the server Router to its argument.
- SMTP: the Hold Mail for Domains option is implemented.
- Account: the Add Mail Trailer and Add Web Banner settings are implemented.
- Rules: when a Reaction/Reply message is composed, the original subject text is MIME-decoded before substituting.
- Bug Fix: modifying Server-Wide Rules on a heavily-loaded server could result in a crash.
- Bug Fix: if the Personal Web Site size setting was set to Unlimited, the Web Site current file number was calculated incorrectly.
- Bug Fix: the 3.1b4 version failed to route addresses with IP-type domain parts.

3.1b4 07-Jul-99

- Kernel: the Crash Recovery General Setting is implemented (obsoletes the --NoCatch and --NoExceptions command line options).
- WebInterface: the CommuniGate Pro Guide are now shipped with the software and all the links in WebAdmin and WebUser pages are changed to the local server copy of the Guide.
- Rules: the Redirect All Mail simplified Rule is implemented.
- Accounts: files used for New Accounts Import can specify all account settings, as well as Central Directory supplementary fields.
- WebUser: access to the Index.html Personal Web Site file requires authorization now.
- WebUser: the Index.html Personal Web Site file format is customizable now; the page form fields allow an authorized user to upload files to that Personal Web Site, and to remove files from that site.
- WebUser: caching is optimized.
- WebUser: form-based page Editor for WebUser pages is implemented.
- WebAdmin: HTTP and form-based page Editor for WebAdmin pages is implemented.
- WebUser/Domain: The WebSite Banner Domain settings can be used to specify some HTML codes to insert into all HTML files retrieved from the domain Personal Web Sites.
- WebUser/WebAdmin: graphical (image) buttons are supported now.
- WebUser: "window" browsing for mailboxes and mail list archives is implemented (i.e. when there are more messages that can be displayed on the page, the user can use the next/previous buttons).
- CLI: the `GetAccount *` command (available to all users) is implemented.
- BSD Mailbox: a mailbox with a damaged last line (w/o EOL) cannot be opened now (manual mailbox correction is required).
- Logs: time stamp composing algorithm has been changed.
- Router: module-level routing is optimized.
- Bug Fix: account passwords set via the WebAdmin Interface were encrypted even if the encryption setting was set to "clear text".
- Bug Fix: MailDir-type accounts were not processed correctly after server restart.
- Bug Fix: the SSL/TLS module could report "wrong version number" because of a buffer processing bug.

3.1b3 30-Jun-99

- Kernel: X.509 certificate routines are implemented.
- Kernel: BigNumber library is implemented.
- Kernel: Cryptography library is implemented.
- Kernel: SSL 3.0 and portions of TLS 1.0 security protocols are implemented.
- Listner: Local Address binding is implemented (to allow the server to accept connections on a selected local IP address only).
- Listner: multi-socket listeners are implemented (the server can now accept connections on several port numbers and/or several local IP addresses).
- Listner: individual Remote IP Address restrictions are implemented.
- SMTP: the STARTTLS extension (secure communication) is implemented.
- SASL: the LOGIN method accepts initial parameters now (should eliminate the AUTH problem

when getting mail from Netscape Servers).

- LDAP: during an initial install, the LDAP module is configured to accept clear text connections on the TCP port 389, and secure connections - on the TCP port 636.
- LDAP: the BIND operation processing has been changed to provide compatibility with Netscape and Microsoft clients.
- IMAP: during an initial install, the IMAP module is configured to accept clear text connections on the TCP port 143, and secure connections - on the TCP port 993.
- IMAP: the UID messages set specifications n:* where n is larger than the largest UID in the mailbox return empty sets now.
- MIME: searching inside MIME-encoded headers is implemented.
- PIPE: support for Envelope-To: headers is implemented.
- RPOP: retrieving UDWA messages has been improved.
- WebUser: the Cache Files option and the Flush Cache button are added to the Domain Settings.
- WebUser: the MailboxViewRegularHeader and MailboxViewHilitedHeader elements are added to the Strings.data file.
- Bug Fix: the Reply/React Rule operation could place garbage into the From: field of the generated messages.
- Bug Fix: 3.1b2 versions could crash when logging a Rule redirect/forward./mirror action.
- Bug Fix: 3.1b1-2 versions could crash when redirecting messages using a Server-wide Rule.

3.1b2 09-Jun-99

- WebAdmin: Account Settings, Template and Defaults have separated HTML page files now.
- Admin: the Account Template can now specify the additional mailboxes to create for all new accounts and the initial mailbox subscriptions for new accounts.
- WebUser: LIST browser: next/previous message switching mechanism has been changed to avoid an intermediate jump page. The old mechanism is still supported for compatibility with old customized List Browser pages.
- WebUser: the Cache-Control: no-cache HTTP attribute is added to various WebUser interface pages.
- WebUser: the FORM element in the Mailbox.html page has been modified to avoid various problems (like inability to undelete a just-deleted message).
- WebUser mailbox browser: the From field is substituted with the To field for the mailboxes specified as Sent and Drafts mailboxes.
- WebUser: the reply/forwarded message Composer MIME-decodes the original message Subject.
- LIST: message subjects are MIME-decoded and excessive Re:/Re> prefix strings are removed when messages are inserted into a digest and/or distributed in the feed mode.
- Bug Fix: in 3.1b1 on most Unix platforms the account Rules page always showed the actions allowed for the Filter Only level.
- Bug Fix: in 3.1b1, generating a digest for a list without any digest subscriber could crash the server.
- Bug Fix: WebUser: the Save As Draft operation could place garbage into the message return-path field.

3.1b1 07-Jun-99

- Admin: the Account password encryption setting is implemented.
 - Rules: the multi-level Rule specifying rights are implemented.
 - WebUser: all attachments (including text files) are sent encoded now.
 - HTTP: the Roaming support for Netscape is implemented.
 - HTTP: the MOVE HTTP method is implemented for personal Web sites.
 - LIST: Settings Web pages have been changed: the Archive/Digest settings have been moved to a separate page.
 - LIST: The Digest Format setting is implemented.
 - LIST: postings from "non-human" sources are rejected now.
 - LIST: the confirm operation is implemented.
 - Bug Fix: WebUser: displaying a Message page for a just deleted message could crash the Server.
-

3.0.1 30-May-99

- Relay restrictions: Mobile users support is improved (see the Protection section).
- SMTP: relay rejection code is changed to 4xx if the Mail From: address is routed to a local account address (this makes the "use POP, then send mail" method more useful for mobile users).
- Rules: the messages generated with automatic Redirect/Forward/Mirror rules get an additional Received header now; this stops "inside-the-server" mail redirection loops.
- MIME: charset name processing is case-insensitive now.
- WebUser: the MarkAll button is added to the mailbox pages.
- WebUser: the Messages To Display option is added to the settings (it is used for account mailing lists, too).
- WebUser: attachment file name processing is improved.
- WebUser: text attachments are displayed as regular attachments, not as inline components.
- WebUser: the Info parameter can be used on the mailbox pages now.
- LDAP: the Make LDAP Server Slower option is implemented (to avoid Netscape crashes).
- Bug Fix: LDAP server could crash when processing malformed search requests.
- Bug Fix: MIME: on RISC platforms, the destructor could cause stack overflow when releasing MIME structures with several thousand elements (as in WebAdmin HTML forms for large subscriber lists).

3.0 24-May-99

- Migration: the MovePOPMail, MoveIMAPMail, and MoveAccounts programs are included into the package. See the Migration section for the details.
- IMAP: the Envelope fields Sender and Reply-To are now defaulted to the From field as required by RFC2060.
- LIST: the unsubscribe and mode change operations can be now performed "silently". The "posting prohibited" policy can now be set as default for new subscribers.
- LDAP: if the "search root" includes the c=country string and the Central Directory does not contain the "c" data, the "c=country" filter is removed (to avoid problems with MS Outlook).

- WebUser: the "Edit Draft" operation is implemented.
- CLI: the ROUTE command is implemented.
- Bug Fix: LDAP: the "strictly equals" string operation parser could crash the server.

3.0b8 16-May-99

- IMAP: the Fetch BODYSTRUCTURE responses now include the Content-ID and the Disposition extension fields.
- Account: the Accept Mail To All option is implemented.
- WebAdmin: the WebAdmin directory in a secondary domain directory can be used to customize that domain administrator WebAdmin Interface.
- Bug Fix: IMAP: Fetch ENVELOPE parsing routine could address out-of-buffer data.
- Bug Fix: BSD Mailboxes: message saving and mailbox parsing routines could read data beyond buffer boundaries, causing crashes on some platforms (FreeBSD).
- Bug Fix: the Mail for Unknown parameter did not work in 3.0b7 version.
- Bug Fix: changing the RPOP module settings could crash the server in 3.0b5-7 versions.

3.0b7 10-May-99

- Domain/Accounts: the concept of Default Settings is introduced (see the Accounts and Domains manual sections for the details).
- DomainAdmin: now the system administrator can specify all operations the domain administrator is allowed to do.
- Admin: now the system administrator should have only the "AllDomains" right, not the Master Right to specify access (domain administration) rights for the accounts in non-main domains.
- WebAdmin: the Domain Administration entrance page (accesses via <http://domainname:8010>) is unprotected now, and it contains a link to the protected domain administration realm.
- CLI: SetAccountTemplate command is changed into UpdateAccountTemplate.
- CLI: GetAccountDefaults/UpdateAccountDefaults operations are implemented.
- CLI: now the special value `default` can be used in the UpdateAccount and UpdateDomain commands.
- PIPE: the Processing Time Out option is implemented.
- LOCAL: the RFC822 field name for storing envelope recipients can be customized now.
- WebUser: the Composer component now selects the proper charset and adds the charset parameter to composed messages.
- PWD: prompt format and login processing has been changed to support the APOP authentication method.
- Bug Fix: QUEUE: Non-ASCII characters in envelope addresses are processed correctly now.
- Bug Fix: Domains: a domain with non-flat foldering could not be removed.
- Bug Fix: WebUser: the Refresh Mailbox rate was not properly stored in the Viewer settings.
- Bug Fix: DOMAIN: domain aliases were not processed correctly when a domain was renamed or removed.

3.0b6 28-Apr-99

- Domain/Account: the Enabled Services options are implemented.
- LOCAL: mail to domains/accounts with the disabled Mail service is suspended in queue now.
- LOCAL: now mail to "almost full" accounts can be suspended in Queue.
- LOCAL: WebAdmin monitor for waiting accounts and for account queues is implemented.
- WebAdmin: now administrators can use the HOST Monitor pages to reject all messages waiting in the Host queue.
- HTTP: internal caching scheme has been changed.
- Domain: RPOP accounts limit is implemented.
- Domain: zero limits can be specified for domain accounts, mailing lists and RPOP accounts.
- WebAdmin/WebUser: Mailing Lists pages are customizable now.
- WebAdmin: now domain administrator access is available for domains without A-records (and for domains that have A-records pointing to a different server). See the HTTP module section for the details (new URL schema).
- WebUser/Archive: preferred charset is used to view messages w/o any explicitly specified charset.
- CLI: the DLACNT alias of the DeleteAccount command did not work.
- CLI: the RenameAccount, RenameList, and RenameDomain command syntax has been changed.
- CLI: the ListLists command has been added.
- WebUser: the name of the HTML page to display when a user logs in is customizable now (stored in the Strings.data file).
- HTTP: a new WebUser setting allows you to specify a prefix for user personal Web Site URLs.
- Bug Fix: search operations in the domain alias database were case-sensitive.

3.0b5 16-Apr-99

- Security: the CommuniGate Pro passwords are stored in an encrypted format now.
- CLI: mailing list manipulation commands have been added.
- WebUser/WebAdmin: Routing of Domain names has been changed (see the HTTP/Routing section for the details).
- RPOP: the Last info field has been added to individual RPOP setting tables.
- RPOP: the Allow Self-Poll option is implemented.
- RPOP: WebAdmin Monitoring is improved.
- RPOP: Retrieval via non-standard ports (servername:port) is implemented.
- LIST: the Preferred charset and Reject Non-Matching Charset options are implemented.
- LIST: the fields in a digest Table of Contents are MIME-decoded.
- LIST: the charset is now specified for the TOC part.
- WebUser: the Preferred charset option is implemented.
- WebUser: mailbox displaying algorithms/formats have been improved.
- Web/MIME: Message header fields encoded in the MIME format are decoded now.
- SMTP: relay selection algorithm is changed to provide better connectivity to slow/remote sites.
- SMTP: a space after ":" is removed from the Mail From: and Rcpt To: commands. Some broken mail servers do not accept SMTP commands with a space after the colon sign.
- SMTP: sending to non-standard SMTP ports is implemented.
- POP: locking model has changed to avoid problems with broken connections.
- The Aliases database is renamed into the DomainAliases database.

- WebUser: some strings have been moved to the String.data file.
- WebUser: an option specifying the default sorting order is added.
- WebUser: an option specifying the mailbox refresh rate is added.
- WebAdmin: last access IP address is displayed in the Account List now.
- Monitor synchronization algorithms have been optimized.
- Windows NT: both local OS accounts and accounts in the trusted domains can now be used for OS-based authentication.

3.0b4 09-Apr-99

- Security: if a password check fails, client processing is suspended for 2 seconds now.
- CLI: GetAccountTemplate and SetAccountTemplate operations are implemented.
- CLI: GetAccountRights and SetAccountRights operations are implemented.
- Foundation: external task starting is modified to address thread-safety issues on the Solaris, FreeBSD, and Linux platforms.
- Bug Fix: routing of listname-request and listname-admin addresses was broken in 3.0b1-3 versions.
- Bug Fix: LDAP "substring searches" could crash the 3.0b1-3 versions.
- Bug Fix: the 3.0b3 version allowed unrestricted access to accounts that had the "use OS password" option set, if no user with the account name was registered with the server OS.

3.0b3 07-Apr-99

- WebAdmin: WebUser Monitor and WebUser Settings pages are implemented.
- WebAdmin: PIPE Monitor is implemented.
- IMAP: message-set processing changed (for Eudora compatibility).
- Kernel: backslash processing in RFC822 comments has been fixed.
- Foundation: a workaround for Solaris DST (Daylight Saving Time) bug.
- Bug Fix: IMAP processing in 3.0b2 could enter an infinite loop.
- Bug Fix: 3.0b1-b2 on some platforms failed to rename/remove accounts.
- Bug Fix: launching external tasks (Rules, PIPE) could lock.

3.0b2 02-Apr-99

- Major Foundation Library redesign, many components modified to take the advantage of the new Foundation classes.
- SMTP: the Message Recipients Limit option is implemented.
- SMTP/Router: the spamtrap address processing is implemented.
- SMTP: envelope processing is improved.
- SMTP: the ORCPT parameter is passed/generated when mail is relayed.
- RPOP: the Minimum Poll Period for Users option is implemented.
- RPOP: the Maximum Poll Accounts per User option is implemented.
- WebUser: address book entries sorting is implemented.
- WebUser: mailing list archives URL schema has changed.
- LIST: the Only Subscribers list archive Browse mode is implemented.

- CLI: GetAccountAliases and SetAccountAliases commands are implemented.
- CLI: GetDomainAliases and SetDomainAliases commands are implemented.
- Bug Fix: the WebUser time-outer thread was not correctly synchronized with access threads; this could cause server crashes.
- Bug Fix: account removal operations could fail because the Personal WebSite files were not removed automatically.
- Bug Fix: Web pages uploading was broken in 3.0b1 .

3.0b1 26-Mar-99

- Cluster: Cluster support Settings are implemented.
 - Cluster: POP, ACAP, IMAP, and PWD cluster support is implemented.
 - Cluster: Personal Web Site cluster support is implemented.
 - Cluster: WebUser Interface cluster support is implemented.
 - Account Prefs: Address Books are implemented.
 - WebUser: access to account AddressBooks is implemented.
 - WebUser: Save Sent Messages and Save as Draft options are implemented.
 - Domain Aliases are implemented.
 - PIPE: non-empty stderr output is now used in the error messages generated when processing ended with a non-zero code.
 - Redirect: the automatically (via Rules) redirected messages have non-NULL Return-Paths now (instead, the notify=never DSN is used for all generated recipient addresses).
-

2.9.1 09-Mar-99

- Bug Fix: the new Router scanner did not cut trailing spaces from the parsed names.

2.9 08-Mar-99

- PIPE: delivery to external applications is implemented.
- WebUser/Domains: The Add Trailer Domain Option is implemented.
- Router: the parser is modified to accept quotation marks in addresses.
- WebUser: the In-Reply-To header is no added to reply messages composed using the WebUser Interface.
- IMAP: part filename and charset data are now included into the BODYSTRUCTURE responses.
- LIST: now existing subscribers can always confirm their subscriptions and can unsubscribe.
- Bug Fix: the All Domains And Account Settings access right worked as Master Right in 2.9b3-2.9b5 versions.
- Bug Fix: Redirect/Forward/Mirror operations in the Account-Level Rules used the empty (<>) address instead of the full account name.

2.9b5 27-Feb-99

- WebUser: personal account Web Sites are implemented. Users can build their Web sites using any HTML editor that uses the PUT/DELETE/MOVE HTTP methods to upload pages (Netscape Composer and others).
- Admin: the MaxWebSiteSize and MaxWebSiteFiles settings (limiting the personal Web site size) are added to the Account Settings.
- The Central Directory can now contain any number of fields. These fields can be set in Account Settings and in the Account Templates.
- The maildir mailbox type is implemented. See the Account -> Data section of the manual.
- WebAdmin: server and domain administrators can now retrieve and update the WebUser Interface files via HTTP using the products like the Netscape Composer.
- WebUser: when displaying a message, the charset of the HTML page is set to the first non-ascii charset used in the message (making it possible to view non-roman letters w/o manually selecting the charset in the browser).
- WebUser: if the Display Subscribed option is enabled, the account subscription is updated when mailboxes are renamed and/or removed via the WebUser interface.
- Rules: the Each To and CC condition is implemented.
- LOCAL: Unified Domain-Wide Accounts can be created in Secondary Domains.
- SMTP: the module does not try to resolve non-qualified HELO/EHLO names now.
- Security: the "Plain" method has been updated to support all versions of Netscape Messenger.
- Bug Fix: if an HTTP connection broke while the Server was receiving a non-form POST request, an exception was raised.
- Bug Fix: Auto-Replying and Reacting to messages w/o the Subject field crashed the server.
- Bug Fix: the LDAP server did not interpret zero time-out value as "unlimited".
- Bug Fix: the LDAP server crashed when generated time-out reports.
- Bug Fix: 2.9b3-4 versions interpreted the "Assign IP Addresses by MX record" setting as the "By A-Record" setting.

2.9b4 18-Feb-99

- The first BSDI BSD/OS version is released.
- SMTP: a new option tells the SMTP module not to advertise the SMTP AUTHentication feature (to avoid problems with Netscape 4.x Messenger).
- SMTP: Verify Return-Path processing has been changed.
- POP: Access to Individual Mail in Unified Accounts is implemented.
- CLI: the GetModule, SetModule, GetDomain, and UpdateDomain commands are implemented.
- HTTP: request Content-Length: values are validated now.
- WebAdmin: all size settings can be set precisely now, using the Others menu option.
- WebUser: the Strings.data dictionary file is added to keep customizable HTML elements used in WebUser Interface.
- WebUser: the Public Info editor is implemented.
- Domain Admin: the Per-Domain Accounts and Lists Number limits are implemented.
- Rules: processing of address strings (conditions) has been changed.
- Rules: Web editing has changed to avoid problems with mixed links.
- The --NoExceptions option is implemented (was set by default in all 2.8-2.9 versions).

- Bug Fix: POST HTTP requests w/o the multipart encoding resulted in memory leaks.
- Bug Fix: a Server-Wide Rule could crash a server if it discarded or rejected a message.

2.9b3 05-Feb-99

- The "Execute" Rule action (allows users to start external programs) is implemented.
- The DNR listener now logs all error conditions instead of shutting down the server (these situations can be met rather often on Linux systems).
- The SMTP module now detects a single '%' sign in an E-mail address and changes it into the '@' sign before sending to a remote host (see the Router section of the SMTP module guide).
- The SMTP module can now accept remote queue starting commands for the Client Hosts only.
- RPOP: Special Headers are removed from the received messages.
- Bug Fix: not all local IP address were detected on some Unix platforms.
- Bug Fix: if a Content-type or Content-Disposition message header had an incorrect format, the MIME parsing engine could crash.

2.9b2 22-Jan-99

- The ACAP module is implemented.
- The DataSet manager (used with ACAP) is implemented.
- The Server-Wide Rules are implemented.
- The SASL Authentication methods (RFC2222) are implemented. See the Security section.
- SMTP: SASL authentication is implemented; this can be used to allow relaying for mobile users and to submit "trusted" messages. The later can be used for LIST approval via E-mail and other operations.
- WebAdmin: the domain-level administration is implemented. See the SysAdmin section.
- WebUser: the IP-address controlling mechanism can be disabled - useful for users accessing the server via a multi-homed proxy.
- WebUser: if the Display Subscribed option is enabled, the account is automatically subscribed to all new mailboxes created via the WebUser interface.
- WebUser: the width of the Message field can be specified in the User settings now.
- IMAP: the "CREATE mailbox/" operation now creates all intermediate folders (needed for to support the Eudora mailer in the IMAP mode).
- RPOP: the polling scheduler internal design has been modified.

2.9b1 20-Dec-98

- ITU "BER" decoder/encoder is implemented.
- LDAP server is implemented (provides read-only access to the "Central.tdb" database).
- SMTP, POP, IMAP, PWD, and LDAP port numbers can be specified now.
- SubFolder support for large domains is implemented.
- the AllDomains virtual account/address is implemented. A message sent to that address in the main domain is stored in all accounts in all server domains.
- Bug Fix: if a message was rerouted/forwarded to an empty set of valid addresses, the operation was

not rejected; instead, an incorrectly formed message was submitted and then moved to the Bad Files.

2.8 06-Dec-98

- Bug Fix: the long-standing bug that first showed up on MacOS and recently reported on some Linux and Solaris systems was finally found and fixed. The bug affected all Unix-based systems, it appeared in rather specific situations, and it could (later) cause various problems - the server stopped to respond, Web clients saw the "Status :1" browser errors, queued message files were stored as empty files, etc.
- The first Digital Unix (OSF) version for Alpha® processors.
- POP: the CAPA operation (RFC 2449) is implemented.
- Mailbox Management: a "quick check" is now performed before a message is stored into a unparsed mailbox. These decreases the probability of storing messages with duplicate UIDs, especially when working with external mailboxes.
- LIST: NULL and BANNED subscription types are implemented.
- LIST: the Archive Swap option is implemented.
- Bug Fix: LIST: the `first digest` and `cleanup period` options were not restored correctly after a restart; this could cause digests not being generated in time.

2.8b3 29-Nov-98

- CLI: the administrator command line interface (CLI) is implemented (see the API/CLI section of the Guide).
- Account/Mailboxes: External Mailboxes providing legacy Unix mailer compatibility are implemented.
- WebUser: an empty WebUser folder is now created inside domain folders. HTML and other files placed into that folder override the files in the main WebUser folder. This allows the system administrator to create different WebUser interfaces for different domains.
- The ProcessID lock-file is implemented (stored in the base directory). Unix start-up scripts now can use that file to stop the Server.
- The `--NoCatch`, `--LogAll`, and `--Daemon` command line options are implemented.
- SMTP: the leading @ sign is ignored in the domain names specified with the ETRN command (needed to serve Lotus Notes client systems).
- Domain service files (settings, templates, rpop accounts, and aliases) have been moved into the special Settings subfolder. This version of the server should move your files into the new locations automatically.
- Bug Fix: distributing mail to all domain accounts via the `all@domainname` address was broken.

2.8b2 17-Nov-98

- Domains: the Assigned IP Addresses option is implemented.
- Kernel: the TCP Activity Scheduler is implemented (restricts SMTP sending and RPOP polling

activity).

- Access: the Grant Access to Client Hosts Only option is implemented.
- LIST: bounce processing has been improved.
- LIST: the `listserver` address for List Server requests is supported now.
- POP: access to all account mailboxes is implemented.
- POP: access to public and shared (foreign) mailboxes is implemented.
- HTTP: image files are retrieved with 24 hours expiration period to avoid unnecessary requests.
- Bug Fix: the HTTP module could duplicate empty lines in form data.

2.8b1 08-Nov-98

- Access: the local domain resolver uses MX records now (see the Access section of the Guide)
 - Access: the "Connections to unassigned IP addresses" option is implemented; it simplifies setup for single-domain systems.
 - Web User: both Frames and No Frames interfaces are implemented (user-selectable).
 - Web User: Web pages are cached now; if you change Web User pages and want the new pages to be used immediately, start the server with the `--NoWebCache` option.
 - LIST: the Enable Archiving & Digesting option is implemented.
 - Bug Fix: the LIST module could crash if a message did not have a [valid] From address.
 - Bug Fix: the INBOX mailbox was not visible in non-multimailbox accounts.
 - Bug Fix: "Authenticated Users Become Clients" processing is fixed and improved.
-

2.7 26-Oct-98

- Kernel: access to foreign (shared) mailboxes is implemented.
- IMAP: operations with foreign mailboxes are implemented.
- IMAP: RFC2086, ACL (access control lists) are implemented.
- IMAP: RFC2342, NAMESPACE request command is implemented.
- IMAP: RFC2359, UIDPLUS protocol extension is implemented.
- Web User: Access to foreign (shared) mailboxes is implemented.
- Web User: Access to subscribed mailboxes is implemented.
- Web User: Modification of the subscribed mailbox list is implemented.
- Web User: Mailbox management (mailbox renaming/removing and ACL management) is implemented.
- LIST: the Silent, Send Welcome, and Ask Confirmation subscribe operations are implemented.
- LIST: The digest size limit can be set to zero to force digest distribution in the semi-feed mode.
- LIST: max archive size can be set to zero to disable list archiving
- LIST: the confirmation ID is generated only once now, (not for each new request). The old scheme created too many problems when several warning/confirmation messages were sent.
- LIST: "text alternative" as available as an allowed format (a messages should be either a text, or multipart/alternative with the first alternative being a text).
- LIST: a bug that prevented archive clean-up has been fixed
- RULES: the COPY command can use the foreign mailbox (`~username/mailbox`) name.

- RULE: the Reject Action is implemented
- RULE: the Add Header action is implemented
- Secondary Domain Access improved, the Unknown Network Address error should not appear in most situations.

2.7b2 20-Oct-98

- Auto Sign-up Domain option is implemented.
- Web User:Auto Sign-up interface is implemented.
- IMAP: the \Deleted flag is processed in accordance with IMAP4rev1 standard now.
- IMAP: the FETCH RFC822.HEADER.LINES and RFC822.HEADER.Lines.NOT commands are implemented to support old IMAP4 clients.
- IMAP: RFC2087 (QUOTA extension) is implemented.
- IMAP: RFC2221 (Login Referrals) is implemented.
- LIST: digest separator is shorten to 70 symbols so it will not create problems for some mail clients.
- Router: IP address to local domain name conversion has been moved from the SMTP router to the kernel router, so it can be applied when routing addresses for Access operations.
- ACCOUNT: the Can Modify Password option is implemented. The PWD module checks this option now.
- Web User: password modification is implemented.
- WebUser: the maximum and used account storage is displayed.
- WebUser: the "From Address" setting is implemented.
- SMTP: the "white hole" processing is implemented.

2.7b1 16-Oct-98

- Account aliases are implemented.
 - Account routing (see the Access section of the manual) is implemented.
 - A lot of internal changes and code clean-ups. Recovery procedures. 64-bit-clean code.
 - The Return Failed option is implemented (in the Obscure settings).
 - The Return-Path and MessageID Rule conditions are implemented.
 - Bug Fix: messages rejected with non-fatal error code could be suspended for a huge period of time.
-

2.6.4 07-Oct-98

- The socket "send" call processing has been changed. This should eliminate some problems with FreeBSD and slow links.

2.6.3 01-Oct-98

- The statically-linked (.tgz) Linux/Intel version for old and non-RedHat systems.
- DNR is improved and fixed (the DNR timeout thread is retired).
- The resolv.conf nameserver address 0.0.0.0 is not rejected now.

- Several fixes/improvements in Linux and Solaris installation procedures.
- Bug Fix: Thread implementation on Mach fixed (this bug could crash MacOS X Servers)
- Bug Fix: Rerouted addresses processing fixed (this could treat redirected mail as illegal relaying).
- Bug fix: the List Manager could crash when unsubscribing users via Web.
- Bug fix: on several platforms List Manager could enter a deadlock state when subscribing users via Web.

2.6.1 19-Sep-98

- The packaged (.rpm) Linux/Intel version.
- The "To", "Cc" and "To or Cc" rule conditions are implemented.
- Bug Fix: some message/digest letters were parsed incorrectly.

2.6 03-Sep-98

- **The first commercial release**
- The first Solaris/Intel version ("pkg_add" format).
- The packaged FreeBSD/Intel version.
- Socket diagnostics are improved on all platforms.
- IMAP mailbox name processing changed to ignore leading "/" symbols (required for the Solaris mailer and some other mailers).
- Bug Fix: the LIST module did not store the number of processors in the settings file.

2.6b4 31-Aug-98

- Account cache is implemented.
- Account hashing tables are implemented.
- Mac OS X (aka Rhapsody) version is now one Installer.app package with fat (PPC + Intel) applications.
- User Web access: the specified domain name is processed with Router domain records.
- Account access: account names are processed with the Router table.
- Local Delivery: the wildcard (*) character is supported in the Reroute-to Domain Options setting.
- IMAP: the List command is made case-insensitive on servers using case-insensitive file systems.
- RULES: string comparisons are made case-insensitive.
- Some default records are stored in the Router table when the system is installed for the first time.
- Bug Fix: IMAP SEARCH BODY bug introduced in 2.6b2 is fixed.
- Bug Fix: LIST module could crash if a new subscriber had no 'real name' in the E-mail address.
- Bug Fix: Web Admin: Router Settings were not updated properly on Unix systems.

2.6b3 25-Aug-98

- RULES: Messages generated with Redirect, Forward and Mirror operations now include the X-Autogenerated header.
- LIST: distribution messages are sent with the Precedence: list RFC field.

- LIST: distribution messages are sent with the Sender RFC field.
- LIST: subscriber address search is case-insensitive now.
- LIST: X-ListServer: field is added to warning, confirmation and other service messages.
- FOUNDATION: implemented a workaround for the Rhapsody/MacOS X bug, so all local IP addresses are retrieved correctly now.
- Bug Fix: the APPEND IMAP command could crash on Win32 platforms.
- Bug Fix: if a message was in the non-multipart text/html format, the User Web interface did not display it correctly.
- Bug Fix: RULES: the Mirror-To operation was not recognized.

2.6b2 20-Aug-98

- LIST module: Web interface to mailing lists is implemented.
- LIST module: Web Interface to subscription lists is implemented.
- Search operations for mailboxes and mailing lists made available via Web Interface.
- DNR is changed to use variable-length time-outs.
- DNR is closed earlier when the server shuts down to abort pending DNR requests.
- Stream management is changed to provide quicker and reliable server shutdown.
- Account caching is implemented.
- QUEUE (for queued messages) Log records are separated from the SERVER Log records.

2.6b1 10-Aug-98

- The LIST module is implemented.
- The Listener module changed to make listening sockets stay alive even after network errors.
- DNR datagram socket buffer has been increased to avoid DNS packet loss under a heavy load (when sending mail to large mailing lists).
- Web Viewer: now "multipart/digest" messages are displayed correctly, as a set of RFC822 messages.
- the Mirror To operation is added to the Automatic Rules.

2.5b3 31-Jul-98

- Vacation Message processing is implemented now.
- The LOCAL Delivery module is multi-threaded now.
- The LOCAL Delivery module monitor is implemented.
- The Original Recipient (ORCPT) option is supported in the SMTP module and Dequeueer
- The HTTP "Host" header is now used for multi-domain Web access.
- The "X-Listserver", "Precedence: bulk" and other special headers are now detected.
- The ""Human Source" condition has been added to the Rules.
- Bug Fix: Transferring messages between mailboxes using Web Interface could crash.
- Bug Fix: On Unix systems the Web Interface did not process long data (message bodies, etc.) correctly (the last symbol was multiplied sometimes).
- Bug Fix: The POST forms were stored corrupted, thus the first parameters could be lost (was seen

as incorrectly processed From/Cc addresses in the Web Interface Composer).

2.5b2 23-Jul-98

- Bug Fix: Rule Editor could crash when editing Action parameters.
- Bug Fix: Message Viewer did not set the proper content-type for non-text and non-image message components (this could corrupt attachments when they were downloaded via the Web Mail interface).

2.5b1 22-Jul-98

- Web-based Mail Access is implemented.
- Automated Mail Processing (Rules) is implemented.
- HTTP "post" requests are supported and used now.
- HTTP MIME-mode parameters are supported and used now.
- Message Redirection/Forwarding is implemented.
- Rule Editor is implemented.
- Bug Fix: SMTP module did not reschedule a message if connection dropped after one of recipients had been rejected.
- Bug Fix: IMAP "LITERAL+" processing is fixed.
- Bug Fix: MAIL-DEAMON is renamed into MAILER-DEAMON and messages to that address are discarded as those sent to the NULL address.

2.2b1 20-Jun-98

- Many internal changes in queue processing and error reporting.
- The SMTP Module, RPOP Module, Host monitor and message Monitors are implemented.
- The DNR channels number restriction is implemented (can be used on slow links).
- "White-Hole" processing is implemented, error reporting for blacklisted addresses is improved.
- The "Relay to backed-up hosts" option is implemented.
- The "*-wakeup" and "/*.smtp" special addresses are implemented
- The "Authenticated Users become Clients" option is implemented
- Settings updates are recorded in the Log now.
- SMTP host delays are restored after a server restart.

2.1b3 10-Jun-98

- The first FreeBSD® version.
- The Text Mailbox format has been changed to make it compatible with legacy "mail" programs.
- The PIPE module is implemented.
- The "Use System Password" option is implemented on Unix platforms (employing the "passwd" authentication).
- The Account Templates are implemented.

2.1b2 27-May-98

- The first Solaris® version.
- The "Use System Password" option is implemented for the WindowsNT platform (using the LogonUser call).
- The MS Windows Installer is implemented.
- The CGStarter.exe is implemented to enable the CommuniGate Pro Server to run as a "service" under WindowsNT.
- Bug Fix: the POP/IMAP modules could crash during multi-access sessions if one of the sessions has emptied the mailbox.

2.1b1 25-May-98

- The RPOP module is implemented (retrieving mail from external accounts via POP protocol).
- The XTND XMIT extension is implemented in the POP module. It can be used with Eudora to submit mail via POP connections.
- Bug Fix: the Domain options "Mail to Unknown" and "Mail to All" did not work properly on all Unix platforms.

2.0b3 11-May-98

- The first Linux version.
- The first Rhapsody DR2 version.
- Rhapsody: the default location for the Server software changed to /Local/Servers, the default location of the "base folder" is changed to /Local/CommuniGate. If you used the older versions under Rhapsody, move the "base folder" and delete the old Server software folder from the /System folder.
- Processing of the IMAP \Deleted flag is changed to confirm the IMAP standard. Non-standard, advanced processing is disabled, and it will be made available as an option later.
- Under Rhapsody DR2, the nameserver addresses are retrieved from the NetInfo database if the /etc/resolv.conf file does not exist.
- On all Unix platforms, the panic (STLog) messages are recorded in the OS "mail" syslog.
- The Web interface HTML pages are updated.
- Bug Fix: the HTTP processor could crash if the very first request resulted in an empty response.
- Bug Fix: deleting of folder accounts, domains and mailbox subfolders did not work in 2.0b2.
- Bug Fix: the "domain does not exist" DNR error was not processed correctly on all Unix platforms.

2.0b2 05-May-98

- Minor fixes in the Web interface. The first Rhapsody version.

2.0b1 27-Apr-98

- **The CommuniGate Pro software is rewritten from Objective C to C++.**
 - **The CommuniGate Pro software is not based on Apple/Next Foundation framework (Yellow Box) any more, the Stalker Portable Foundation Framework is used instead.**
 - Several minor changes in the IMAP protocol.
 - Improved performance for extra-large (50MB and more) text mailboxes.
 - Socket library improvements for MS Windows platforms.
-

1.0b4 08-Mar-98

- Multiple Account Domains are implemented.
- Hierarchical mailboxes are implemented.
- RFC1870 - SMTP "SIZE" extension support is implemented.
- RFC2180 - IMAP multi-access details are corrected.
- RFC2088 - IMAP non-synchronizing literals are implemented.
- RFC1077 - IMAP IDLE command is implemented.
- The Account Info databases are implemented.
- The Last Login Time, Last Login IP is stored in the Account Info.
- The IMAP UIDValidity / UIDNext functionality is implemented.
- The IMAP RECENT functionality is implemented.
- The POP LAST command is implemented.
- Independent UIDs for every mailbox are implemented.
- Sizes of folder-type accounts are calculated dynamically now.
- The marked/unmarked mailboxes status is indicated in IMAP now.
- IP multihoming is supported.
- Multi-domain support based on mutihoming is implemented.
- Multi-domain support based on @ and % symbols is implemented in POP, IMAP, PWD and HTTP modules.
- Bug fix: when the SMTP module was receiving a letter with a line starting with "." followed with an empty line, the Server crashed.

1.0b3 20-Feb-98

- The Rhapsody/Intel version is released.
- Now the SMTP module can open several channels when sending messages to one host.
- Monitor access privileges are added. Logs and Queues panels require the Can Monitor privilege now.
- Queue monitor panel is implemented. It shows all the messages in the Server queue.
- Now the DNR module can repeat requests to Domain Name Servers.
- Now the DNR module can use several DNS servers.
- The POP and IMAP modules now remember the IP addresses used to make authenticated connections. The SMTP module considers those IP addresses as "Client Hosts" for 30 seconds.
- Bug Fix: the DSN messages did not have the "trusted source" marker.

1.0b2 09-Feb-98

- Various low-level issues corrected to fix the Yellow Box for Windows version.
- Bug Fix: the IMAP BODY[] commands (used with MS Exchange) were not processed correctly.
- Bug Fix: IMAP SEARCH date did not work correctly with MS Outlook clients.
- Bug Fix: IMAP module did not correctly processed UID-prefixed commands causing problem for some mail clients (including one from Netscape).
- Bug Fix: storing messages in an open mailbox that had been cleared caused a crash.

1.0b1 03-Feb-98

- The SEARCH BODY/TEXT IMAP command is implemented.
- Now non-EXPUNGE IMAP mailbox updating information can be returned in response to FETCH, STORE and SEARCH commands.
- Bug Fix: IMAP EXPUNGE responses could be returned for the STORE command.
- Bug Fix: The SEARCH KEYWORD IMAP command was not implemented correctly.
- Bug Fix: Account Enabled and Login Enabled options were not set correctly via the HTTP interface.

1.0b0 02-Feb-98

- The first public release.